Engineering Applications of Artificial Intelligence 166 (2026) 113670

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Developing fully convolutional networks with permittivity-based class mapping for tunnel lining defects detection in ground penetration radar scan data

Viet Dinh Le [a], Gyu-Hyun Go [a,*], Sayali Pangavhane [a], Chang Kyoon Yoo [b]

[a] School of Architecture, Civil and Environmental Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk, 39177, Republic of Korea
[b] Chungcheong Headquarters, Korea Authority of Land and Infrastructure Safety, Cheongju, Chungcheongbuk-do, 28537, Republic of Korea

ARTICLE INFO

Keywords:
Ground penetrating radar
Fully convolutional network
Local defects
Tunnel lining

ABSTRACT

Non-destructive testing using ground penetrating radar (GPR) plays a vital role in identifying defects in tunnel linings, such as cavities, delamination, and interlayers, to ensure the safety and longevity of underground structures in mega-cities. However, the absence of automated tools for generating realistic simulation datasets and the limitations of existing methods in handling complex defect scenarios, such as cavity and delamination, pose significant challenges for accurate detection. This study addresses these gaps by introducing KIT-GPR, a numerical simulation tool, and a fully convolutional network (FCN) for automated defect detection. KIT-GPR has employed the Finite-Difference Time-Domain method to simulate electromagnetic wave propagation through multi-layered tunnel structures, resulting in high-fidelity B-scan data. Its innovative application of the Perlin noise algorithm, commonly used in game design, generates realistic interlayers between grout and rock layers, thereby improving the accuracy of the simulation model. A modified FCN, designed with a 320x320 square input size to accommodate versatile data and a 256-class output corresponding to dielectric constant ranges, was trained using 1000 paired B-scan and permittivity datasets. This approach achieves a Root Mean Square Error of approximately 0.8 in most scenarios, including interlayers with delamination. Furthermore, the predicted delamination locations using FCN from B-scan data of a real tunnel closely align with the results from endoscopic imaging. This demonstrates that the FCN prediction model has significant promise as a scalable and efficient solution for early defect detection, thereby greatly enhancing the safety of underground infrastructure.

1. Introduction

Urbanization, particularly the rise of megacities, has led to an increase in tunnel construction, making underground structures an essential component of transportation infrastructure. These structures meet the growing demands and ensure the convenience of urban mobility in different areas. However, underground construction often faces unique challenges owing to its subsurface locations. The geological conditions may pose risks such as fluctuation in groundwater levels, leading to local defects that can combine and threaten the normal functioning of the structures. For instance, grout layer damage may result in cavities, whereas interlayer between the grout and concrete layers may cause delamination. Fig. 1 illustrates common local defects in the tunnel lining structure. Such damage often facilitates structural corrosion through groundwater infiltration. If these defects expand their combined effect, can compromise operational safety and pose significant risks, especially for older underground structures with extended service lives.

Non-destructive testing (NDT) technologies play a vital role in maintaining tunnel infrastructure, as highlighted by an early study from Haack et al. (1995). Among these, ground penetrating radar (GPR) stands out as a leading method for detecting defects in tunnel linings. Recent advancements have further enhanced GPR's capabilities: Zhang et al. (2019) improved defect detection accuracy by integrating full waveform inversion (FWI) and Reverse Time Migration (RTM) techniques, while Huang et al. (2024) introduced a self-supervised deep learning framework to refine GPR data analysis, enabling precise flaw identification. Dawood et al. (2020) also demonstrated GPR's effectiveness in inspecting subway infrastructure. Beyond defect detection, effective geological prospecting is essential for mitigating risks during

* Corresponding author.
E-mail address: gyuhyungo@kumoh.ac.kr (G.-H. Go).

https://doi.org/10.1016/j.engappai.2025.113670
Received 1 April 2025; Received in revised form 19 December 2025; Accepted 24 December 2025
Available online 27 December 2025

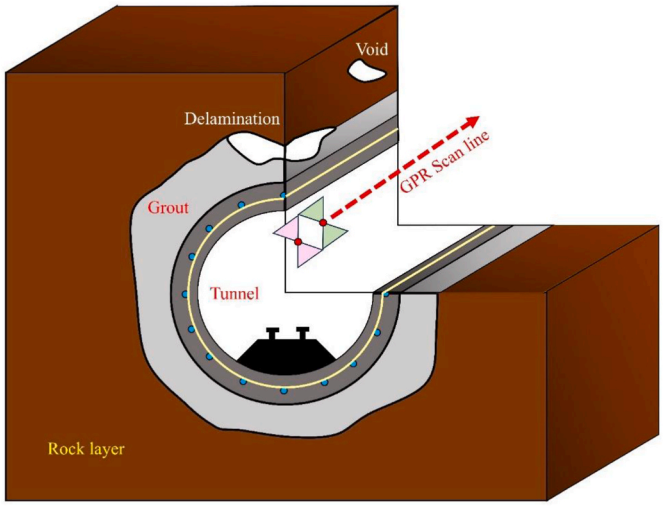

Fig. 1. Detection of local defects in tunnel lining using GPR.

tunnel construction. Li et al. (2017) outlined detailed methodologies for identifying geological hazards, crucial for ensuring safety. Subsurface imaging technologies have similarly evolved, with Wang et al. (2015) proposing forward-looking synthetic circular arrays to enhance tunnel boring machine (TBM) operations. The NeTTUN project (Simi and Manacorda, 2016) further advanced this by developing specialized GPR antennas for TBMs to better detect subsurface anomalies. Complementing these innovations, Zhang et al. (2022) employed wavelet transforms for quantitative, real-time recognition of filler materials in three dimensional (3D), reducing uncertainties in geological assessments.

In recent years, data-driven methods have gained popularity owing to advancements in deep learning technologies. A convolutional neural network (CNN) is a specialized form of feed-forward neural network that incorporates regularization through filter (or kernel) optimization to automatically learn features. This deep learning architecture has been widely used for processing and making predictions across various data types such as text, images, and audio (LeCun et al., 2015). To detect local defects in tunnel linings, Qin et al. (2021) developed a CNN-based system to automate the identification of tunnel lining elements from GPR images using data augmentation to enhance performance. Zhou et al. (2018) introduced an automated interpretation model for GPR B-scan images that reduced the reliance on manual analysis. Montero et al. (2015) emphasized the role of robotic inspection technologies in streamlining maintenance processes, which are vital for ensuring the long-term integrity of tunnel infrastructure. However, their performance depends heavily on the quality and quantity of the input data, which poses challenges, especially in inspecting structural linings where defects are unevenly distributed and lining structures are influenced by diverse geological conditions. To address data scarcity in GPR inspections, machine learning techniques and numerical simulations have been implemented (Veal et al., 2018; Qin et al., 2021; Yue et al., 2021; Huang et al., 2024; Zhu et al., 2024). For example, Veal et al. (2018) utilized generative adversarial networks (GANs) to detect explosive hazards using handheld GPR systems, achieving higher accuracy and reducing false positives. Yue et al. (2021) applied LS-GANs to generate high-precision GPR images, significantly improving subsurface feature recognition. Zhu et al. (2024) utilized deep learning and BIM to enhance the data-driven maintenance of tunnel linings. Among these methods, artificial intelligence (AI)-driven approaches have demonstrated the potential of machine learning in overcoming the limitations of traditional GPR analyses. In addition, data synthesis through GPR simulations has proven to be particularly effective.

Several authors have developed machine learning models to identify subsurface objects beneath underground structures from B-scan data. Liu et al. (2021) developed a deep neural network (DNN) model named GPRInvNet, with a specially designed trace-to-trace transformer to directly convert B-scan data into a dielectric permittivity map. The GPRInvNet model was compared with FWWI and Encoder-Decoder (Enc-Dec) methods. The results demonstrated that GPRInvNet significantly outperformed the other methods in reconstructing dielectric permittivity maps. The input database for GPRInvNet was generated by Liu et al. (2021) using the Finite-Difference Time-Domain (FDTD) simulation method in MATLAB. The authors employed 432,000 data pairs—a vast amount of data requiring extensive training time. Additionally, the authors used rectangular input data (HxW), leading to difficulties in applying the model to data with varying width-to-height ratios.

A more recent study by Liu et al. (2023) proposed a multitask deep neural network method (M-YOLACT). This model combined with post-processing via curve fitting to simultaneously identify the shape, type, and depth of defects inside tunnel linings, as well as the thickness of the lining, from GPR images. This system incorporates attention mechanisms, background suppression, and multitask semantic segmentation to improve the accuracy of defect reflection signal recognition, lining boundary detection, and hyperbolic feature extraction. Moreover, the study introduced an automatic in-situ dielectric constant estimation method, enabling precise calculation of defect depth and lining thickness. Liu et al. (2023) validated this approach through numerical simulations, sandbox experiments, and field tests. His model demonstrated high effectiveness in identifying defects and lining boundaries from GPR B-scan images. Notably, Liu et al. (2023) converted simulated and measured GPR data into images with a resolution of 533×533 pixels, allowing the trained model to predict any GPR data from either simulations or measurements.

Zhu et al. (2024) developed the TunGPR model, which offers significant advantages in applying digital technology to tunnel maintenance. This system integrates Building Information Modeling (BIM) with GPR, creating an advanced tunnel lining assessment method by leveraging digital data and geological models. A key advantage is the construction of a comprehensive database using FDTD simulations and domain randomization techniques. This model enhances dataset diversity and improves deep learning model accuracy. The system also employs a CNN with a dual-rotational bounding box design, enhancing the detection of hyperbolic features in GPR B-scan data and increasing automation in data analysis. To detect defects such as voids, cracks, delamination, and water infiltration, TunGPR integrates a risk assessment matrix into BIM, visually representing damage severity using a color-coded scheme. This method help tunnel managers prioritize maintenance measures. However, certain limitations remain, particularly the dependency on the quality of the synthesized dataset, which may not fully reflect actual tunnel and geological conditions. Although the authors used Bézier curves and random points to generate delamination interfaces, real-world delamination patterns are often more complex. Furthermore, while CNN improves hyperbola detection accuracy, the model may still be affected by background noise and irregular defect shapes in real-world conditions. Additionally, FDTD simulations and deep learning model training require significant computational resources, increasing deployment costs.

Thus, key challenges in applying machine learning to defect prediction from GPR data include the quality of the training dataset, model adaptability to arbitrary input data sizes, and automation in dataset generation via FDTD simulations. To address these challenges, this study aims to developing the KIT-GPR, an automated graphic user interface simulation tool based on the FDTD method, and a modified Fully Convolutional Network (FCN) model designed to predict localized defects in tunnel linings efficiently. Furthermore, the trained FCN model is validated on real-world B-scan data collected from a tunnel in South Korea exhibiting delamination damage, demonstrating its practical applicability.

A simulation program, named KIT-GPR, is developed based on the FDTD method. Several representative scenarios are designed to verify the accuracy of KIT-GPR, and the A-scan data obtained are compared with data from the open-source gprMax software (Warren et al., 2016). To create interlayer objects, the Perlin noise algorithm (Perlin, 2002a) is employed to define the shapes of cavities and delaminations. This algorithm, widely used in graphics programming to simulate geological layers, water surfaces, smoke, and fog, enables more natural delamination contours. Additionally, KIT-GPR integrates an automated algorithm to identify appropriate delamination positions along interlayer curves, reducing time consumption and increasing automation in dataset generation for prediction models. Four defect scenarios were developed for an FCN prediction model: interlayer only, interlayer with cavity, interlayer with delamination, and a combination of all three. Each scenario includes 250 models, totaling 1000 input models. The FCN uses VGG-16 pre-trained weights to create an initial feature matrix and defines 256 classes based on dielectric constant values (0–25.5), with rebar set at a maximum of 25.5. Input data consists of B-scan data (synthesized from A-scan data) and dielectric permittivity maps (generated from geometry and random material values). In addition, smoothing algorithms average transition regions at material boundaries in numerical model.

The B-scan data without gain factor is used as input for the FCN model in this study. The original B-scan data allows the FCN model to better understand electromagnet wave dispersion with depth. The KIT-GPR program automatically generates random geometries according to predefined scenarios and outputs B-scan data along with dielectric constant maps for FCN training. The dataset is divided into training, validation, and test sets to evaluate the model. The FCN prediction results are analyzed and compared with the geometry ground truth in the test set, which was previously separated and never used in FCN training.

The paper is divided into five sections. Section 2 presents the literature review, including numerical analysis model, the fundamental of Perlin noise and the FCN model. Section 3 introduces the application of the Perlin noise algorithm to generate geometries of simulation model. Four scenarios of tunnel defects are described and simulated by the KIT-GPR program, and then the A-scan results compared to those from the open-source software gprMax (Warren at el., 2016). Section 4 discusses the FCN model and evaluation process. The FCN is developed to predict defect locations in tunnels. The FCN is evaluated with synthetic scenarios and measured B-scan from one tunnel in Korea. Finally, Section 5 presents the research findings and key conclusions.

2. Literature review

2.1. Fundamental of numerical analysis model

Numerical models of ground penetrating radar (GPR) are frequently simulated using various numerical methods, including ray-based, frequency-domain, integral, pseudo-spectral, and Finite-difference time domain (FDTD) methods. Of these methods, the FDTD method is experiencing growing utilization in the numerical simulation of electric field waves in general as well as ground-penetrating radar problems.

Some authors have developed GPR simulation software based on the FDTD method, such as FDTD-based MATLAB (Irving and Knight, 2006) and gprMax (Warren et al., 2016). Following the FDTD method, physical parameters related to electromagnetic waves including conductivity σ, dielectric constant ε, and magnetic permeability μ. These parameters are input into the calculation model. For example, metals generally exhibit very high conductivity σ, typically ranging from 1000 to 10,000 S per meter (S/m) or even higher. In contrast water has a considerably lower conductivity, approximately between 0.005 and 0.05 S/m, while air possesses very low conductivity, functioning almost as an insulating medium due to its near-zero conductivity. Another critical parameter is the dielectric constant ε, which illustrate the relationship between electric flux density **D** and electric field intensity **E**. In addition, magnetic permeability μ, another parameter reflecting the relationship between the magnetic field **H** and magnetic flux density **B**, are typically constant for most materials on Earth. Fig. 2 shows the electric and magnetic fields for a transverse electric model formulation.

In general, the GPR numerical model is interpreted using Maxwell's formula in the frequency domain and the basic equations are expressed as follows:

$$\nabla_e \times \mathbf{E} = -i\omega\mu\mathbf{H}, \tag{1}$$

$$\nabla_h \times \mathbf{H} = i\omega\varepsilon\mathbf{E} + \sigma\mathbf{E}, \tag{2}$$

where $i = \sqrt{-1}$; ω is the angular frequency.

In addition, the operators ∇_e and ∇_h are defined as follows:

$$\nabla_e = \hat{x}\frac{1}{e_x}\frac{\partial}{\partial_x} + \hat{y}\frac{1}{e_y}\frac{\partial}{\partial_y} + \hat{z}\frac{1}{e_z}\frac{\partial}{\partial_z}, \tag{3}$$

$$\nabla_h = \hat{x}\frac{1}{h_x}\frac{\partial}{\partial_x} + \hat{y}\frac{1}{h_y}\frac{\partial}{\partial_y} + \hat{z}\frac{1}{h_z}\frac{\partial}{\partial_z}. \tag{4}$$

where e_k and h_k $(k = x, y \text{ or } z)$ are the variables for complex coordinate transformation of the $\hat{x}$, $\hat{y}$ and $\hat{z}$ for ∇_e and ∇_h, respectively.

In the numerical GPR model, the geometry is divided into a grid of points, which includes points inside the area and a boundary region known as the Perfectly Matched Layer (PML) (Gedney, 2011). The PML is a special material layer used in electromagnetic wave simulations to absorb waves without allowing reflections back into the interior of the model. It is designed to absorb waves perfectly, helping to simulate an infinite space within a finite region. The calculated points within the interior region are considered as material properties defined by the user, whereas the computational points within the boundary condition region are assigned PML material properties, which are typically designed to absorb outgoing waves and minimize reflections. These PML parameters are based on default settings but can often be customized depending on the specific simulation requirements (Irving and Knight, 2006). Therefore, the calculated points in Equations (3) and (4) can be interpreted according to the general formula as follows:

$$\nabla = \hat{x}\frac{1}{s_x}\frac{\partial}{\partial_x} + \hat{y}\frac{1}{s_y}\frac{\partial}{\partial_y} + \hat{z}\frac{1}{s_z}\frac{\partial}{\partial_z}, \tag{5}$$

As suggested by Irving and Knight (2006), the functions $s_k (k = x, y \text{ or } z)$ can be expressed as

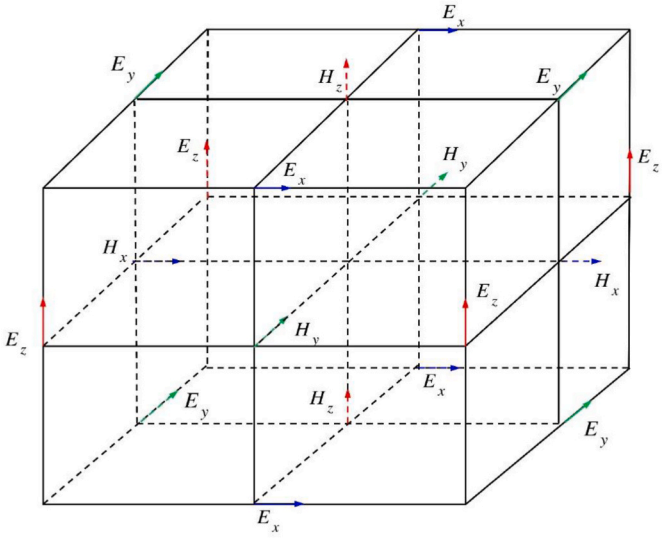

Fig. 2. Schematic of electric and magnetic fields for numerical model (Warren et al., 2016).

$$s_k = \kappa_k + \frac{\sigma_k}{\alpha_k + i\omega\varepsilon_0}, \tag{6}$$

where $k = x, y$ or z and $\varepsilon_0 = 8.8542 \times 10^{-12}$ is the absolute permittivity of free space.

The parameters κ_k, σ_k, and α_k are used to express the attenuation of electromagnetic waves from the computational region to the boundary region known as the PML. The value of α_k, in various directions, is generally selected to be greater than zero to improve the absorption of electromagnetic waves by the PML layer (Roden and Gedney, 2000). However, for the simplifying computations, this value is set to zero as a default in the GPR numerical model (Irving and Knight, 2006).

The coefficients κ_k and σ_k are estimated based on the position of the calculated point. Within the computational region, $\kappa_k = 1$ and $\sigma_k = 0$, whereas for the points in the PML boundary region, the coefficients κ_k and σ_k are expressed as follows:

$$\kappa_k = 1 + \left(\frac{d}{\delta}\right)^m (\kappa_{\max} - 1), \tag{7}$$

$$\sigma_k = \left(\frac{d}{\delta}\right)^m \sigma_{k_{\max}}, \tag{8}$$

where d denotes the distance from the calculated point within the PML boundary region to the interface between the internal computational domain and the PML region.

In addition, δ is the thickness of the PML region. According to Irving and Knight (2006), the exponent value m = 4 and the coefficient $\kappa_{\max} = 5$ are selected in the x, y, and z coordinate directions. Meanwhile, the maximum values of coefficient $\sigma_{k_{\max}}$ is defined according to the following condition (Gedney, 2011).

$$\sigma_{k_{\max}} = \frac{m+1}{150\pi\sqrt{\varepsilon_r}\Delta k}. \tag{9}$$

where ε_r is the relative dielectric permittivity at the interface between the computational region and PML boundary. The ε_r value is the normalized permittivity of material nearest the interface to the PML in free space.

Equations (1) and (2) are expressed as partial differential equations in a two-dimensional (2D) model along the x and z axes based on the development in Equation (3), with the components as follows:

$$i\omega\mu H_x = -\frac{1}{s_z}\frac{\partial E_y}{\partial z}, \tag{10}$$

$$i\omega\mu H_z = -\frac{1}{s_x}\frac{\partial E_y}{\partial x}, \tag{11}$$

$$\sigma E_y + i\omega\varepsilon E_z = \frac{1}{s_x}\frac{\partial H_z}{\partial x} - \frac{1}{s_z}\frac{\partial H_x}{\partial z}. \tag{12}$$

According to Irving and Knight (2006), Equations (10)–(12) are utilized to simulate a ground penetrating radar moving on the ground, which is known as a Transverse Magnetic Model. They successfully implemented and demonstrated an algorithm and MATLAB code to solve these equations. However, in this study, the theoretical frameworks and algorithms from their work were adapted and refined to develop a graphic user interface (GUI) GPR simulation program in C#. Further details of the functionality are provided in the following sections.

In summary, the FDTD method is a numerical technique for solving Maxwell's equations in the time domain, which is widely used to simulate the propagation of electromagnetic waves in underground infrastructure. The output simulation data of FDTD is often used in most of research articles as strength field E data in different directions through the collection of A-scan, B-scan or C-scan data files. For A-scan, FDTD divides the computational domain into a grid with spatial and temporal steps, ensuring that the stability condition is satisfied. Following simulation model based FDTD method, a signal source, such as a Gaussian pulse, excites the domain, and the E_y value is recorded over time at a fixed receiver point, generating an A-scan reflecting the reflected waves from material interfaces based on the difference in dielectric constant. B-scan is generated by calculating A-scan at multiple receiver positions along the scanning line. Each A-scan shows the variation of E_y over time at a spatial point, and when combined, they form a 2D image that shows wave propagation in both time and space, highlighting subsurface features such as interlayer, cavities or delaminations. To generate the C-scan, FDTD is extended to include E_x and E_z field components in perpendicular planes, providing a 3D representation of the electromagnetic field. The C-scan provides volumetric information about subsurface structures, supporting the identification of objects in 3D.

Therefore, a combination of FDTD that generates synthetic data with machine learning, such as Fully Convolutional Networks (FCN), enhances the local defects underground analysis capabilities. The A-scan, B-scan, or C-scan data generated by FDTD, labeled with material properties such as dielectric constant, can be used to train a model to predict the relative permittivity map. These prediction models allow the reconstruction of subsurface geometry underground, accurately mapping objects with distinct electromagnetic properties.

2.2. Fundamentals and application of Perlin Noise

Perlin Noise is widely used to generate digital content, such as games or animated movies. The Perlin Noise algorithm developed by Ken Perlin was introduced in the 1980s. However, its application was limited during this period due to the underdeveloped state of computer graphics technology. With the advancements in computing technology, Perlin Noise has become increasingly prevalent during game development. Perlin subsequently improved his algorithm, and published it in 2002 (Perlin, 2002a).

In general, Perlin Noise is used in video game development to generate terrains like mountains, earth surfaces, and lakes, as well as effects such as fire, smoke, and clouds. These applications primarily use 2D or 3D Perlin Noise algorithms but can extend to 4D by incorporating the time axis. In addition, the Perlin Noise algorithm can be reduced to a one-dimensional (1D) form to generate curved terrains or simulate handwritten strokes. In this study, the extended 1D Perlin Noise algorithm is applied and integrated into the KIT-GPR software to create an interlayer between the soil layers, cavities, and delamination.

Among the versions of the Perlin Noise algorithm, the 2D version is the most fundamental, as it can be adapted to 1D, 3D, and even 4D forms. The 2D Perlin Noise algorithm is frequently used to generate pseudo-random terrains or textures (Mount and Eastman, 2018). Assume a given grid of square cells with a size of $n \times n$, where n is a relatively small integer, typically between 2 and 10. Let the height map values $z[i,j]$ at the grid vertices be random scalar values. For a given point with coordinates $I(x,y)$, the corner points of a unit cell containing this point can be identified as $A\left(x_i, y_j\right)$, $B\left(x_{i+1}, y_j\right)$, $C\left(x_{i+1}, y_{j+1}\right)$ and $D\left(x_i,\ y_{j+1}\right)$. These coordinates are determined by the following expressions:

$$\begin{cases} x_i = \lfloor x \rfloor \\ y_i = \lfloor y \rfloor \end{cases} \text{ and } \begin{cases} x_{i+1} = (x_i + 1) \bmod n \\ y_{i+1} = (y_i + 1) \bmod n \end{cases}, \tag{13}$$

In addition, the fractional part of coordinates x and y can be calculated as:

$$\begin{cases} x_{fp} = x_i - \lfloor x \rfloor \\ y_{fp} = y_i - \lfloor y \rfloor \end{cases}. \tag{14}$$

The corner point values of the unit cell are calculated using Perlin's

permutation table, which contains 256 entries (Perlin, 2002b). In general, the permutation table is shuffled values in a calculation step. Therefore, the values corresponding to the corner points of the unit cell are defined as follows:

$$z_A = P[P[x_i] + y_i], \tag{15}$$

$$z_B = P[P[x_i + 1] + y_i], \tag{16}$$

$$z_C = P[P[x_i] + y_i + 1], \tag{17}$$

$$z_D = P[P[x_i + 1] + y_i + 1]. \tag{18}$$

For a given point $I(x,y)$ located within a unit cell, it is necessary to blend the influence of the gradients at the corner vertices. To overcome this issue, for each corner point, a directional vector is computed from the corner vertex to a given point $I(x,y)$. Specifically, they are defined as:

$$v_A = (x,y) - \left(x_i, y_j\right), \tag{19}$$

$$v_B = (x,y) - \left(x_{i+1}, y_j\right), \tag{20}$$

$$v_C = (x,y) - \left(x_{i+1}, y_{j+1}\right), \tag{21}$$

$$v_D = (x,y) - \left(x_i, y_{j+1}\right). \tag{22}$$

Furthermore, using z_k (k = *A, B, C,* or *D*) values ranging from 0 to 255, we can determine four possible directions in space. Accordingly, the corresponding constant vector g_k is selected based on one of the following four cases:

$$z_k \%4 = 0 \Rightarrow g_k = (1,1), \tag{23}$$

$$z_k \%4 = 1 \Rightarrow g_k = (-1,1), \tag{24}$$

$$z_k \%4 = 2 \Rightarrow g_k = (-1,-1), \tag{25}$$

$$z_k \%4 = 3 \Rightarrow g_k = (1,-1). \tag{26}$$

Thus, for a given point with x and y coordinates, the dot product for each vertex of the unit cell can be computed easily. The dot-product values between the directional and constant vectors at each vertex (k = *A B, C,* or *D*) can be calculated as follows:

$$\lambda_k = (v_k \cdot g_k). \tag{27}$$

However, to ensure that the algorithm returns a unique value, the values must be combined through interpolation. Specifically, the interpolation process is performed in two steps: First, the interpolation between the two left vertices (Points *A* and *D*) yields the value g_1; Second, the interpolation between the two right vertices (Points *B* and *C*) produces the value g_2. Interpolation between g_1 and g_2 provides the final output, which is the value returned by the noise function.

The linear interpolation between two values, a_1 and a_2, with the parameter t in the range [0, 1] is expressed as follows:

$$\mathrm{lerp}(a_1, a_2, t) = a_1 + t(a_2 - a_1). \tag{28}$$

where a_1 and a_2 denote the initial and final values, respectively; t indicates the interpolation ratio between a_1 and a_2.

A challenge with these scalar displacement values arises from their dependence on all the vertices of a unit cell. In practice, the farther a point is from the relevant vertex, the greater the displacement value. We aim for the gradient effect to be applied only near the vertices and to diminish rapidly as the point approaches another vertex. That is, the gradient effect at a vertex fades as its distance from the vertex increases. To overcome this problem, Perlin introduced a fading function. This is a function of t that starts at zero when $t = 0$ (no fading) and gradually increases to one when $t = 1$ (complete fading). Initially, Perlin selected a cubic function to implement the fading effect (Mount and Eastman, 2018), expressed as follows:

$$\psi(t) = 3t^2 - 2t^3, \tag{29}$$

This function has a derivative equal to zero at $t = 0$ and $t = 1$, ensuring a smooth interpolation process for the noise algorithm across neighboring unit squares. However, Perlin observed that the second derivative of this function is nonzero at $t = 0$ and $t = 1$. To address this issue, he introduced the following improved fade function (Mount and Eastman, 2018):

$$\psi(t) = 6t^5 - 15t^4 + 10t^3, \tag{30}$$

Because the fade function is applied to both x and y coordinates, the joint fade function can be expressed as follows:

$$\Psi(x,y) = \psi(x)\psi(y), \tag{31}$$

Thus, the noise function for a given point with coordinates x, y is represented by the following formula:

$$\mathrm{noise}(x,y) = \Psi(1-x, 1-y)\delta_A + \Psi(x, 1-y)\delta_B + \Psi(x,y)\delta_C + \Psi(1-x, y)\delta_D, \tag{32}$$

in which

$$\delta_{(A,B,C \text{ or } D)} = z_{(A,B,C \text{ or } D)} + \lambda_{(A,B,C \text{ or } D)}. \tag{33}$$

Based on the noise value function in Eq. (32), the Perlin function is calculated as follows:

$$\mathrm{perlin}(x,y) = \sum_{i=0}^{N-1} \left(a_0 \cdot p^i\right) \cdot \mathrm{noise}\left(x \cdot f_0 \cdot 2^i, y \cdot f_0 \cdot 2^i\right), \tag{34}$$

where a_0 and f_0 denote the initial amplitude and frequency, respectively; p denotes persistence, which reflects the rate of amplitude reduction as the frequency level $i - th$ increases; and N is the number of noise layers (octaves).

For a 1D dimension, the 2D Perlin noise function can be extended and reformulated as shown in the equation below:

$$\mathrm{perlin}(x,y) = \sum_{i=0}^{N-1} \left(a_0 \cdot p^i\right) \cdot \mathrm{noise1d}\left(x \cdot f_0 \cdot 2^i\right), \tag{35}$$

Thus, in the Perlin Noise equation, key parameters such as the number of octaves, persistence (amplitude decay factor), initial amplitude, and initial frequency play critical roles in determining the characteristics and variations in noise. Each parameter has a distinct effect on the noise generation and its properties in practical applications. To examine the influence of these parameters, they must be systematically varied and the resulting outputs analyzed. For example, increasing the number of octaves enhances the level of detail in the signal but also increases the computational complexity. Similarly, adjusting the persistence can affect the smoothness or sharpness of the signal, whereas the initial amplitude and frequency determine the range and scale of the oscillations.

To investigate the impact of the Perlin noise algorithm parameters on a 2D problem, we selected octave values of 1, 4, and 8 while maintaining a constant amplitude of one. The persistence was selected as 0.005, 0.01, and 0.02 while the frequency was selected as 0.005, 0.01 and 0.02. Fig. 3 illustrates how variations in these parameters influence texture, showing the land and ocean formations generated by the Perlin noise algorithm. From Fig. 3, it can be observed that increasing the number of noise layers (octaves) enhances the signal's detail, producing more complex and natural fluctuations. However, increasing the number of octaves may lead to slower computation and unnecessary complexity in certain cases because the smaller noise layers at higher octaves typically contribute very little to the overall value. In the Perlin noise algorithm, the persistence parameter is crucial in determining the rate at which the amplitude of each noise layer decreases as the number of octaves

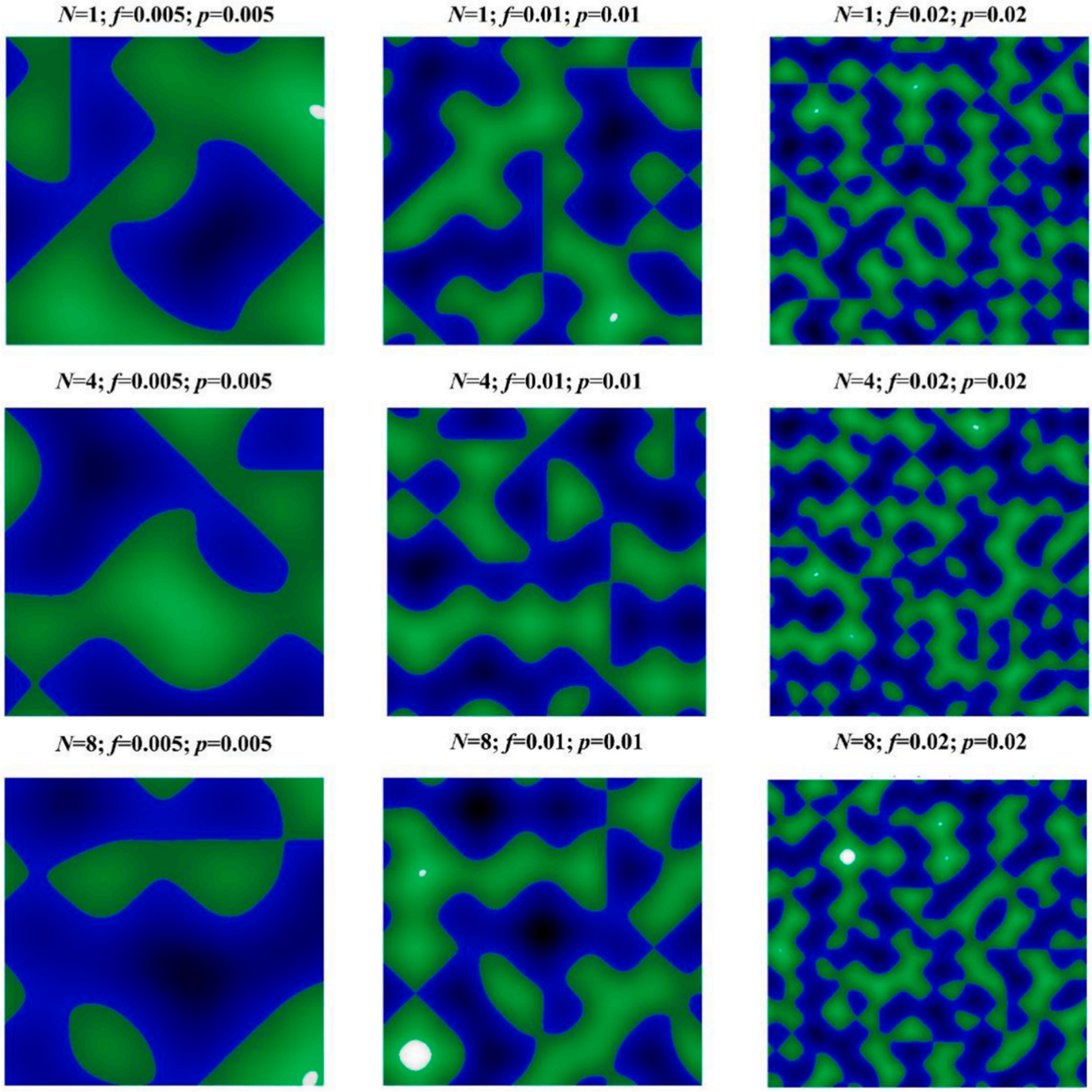

Fig. 3. Height map using Perlin noise 2D with variation its parameters.

increases. Higher persistence results in a more pronounced effect of higher-frequency layers, leading to more detailed noise but potentially reducing its smoothness. Conversely, lower persistence results in smoother but less detailed noise. In addition, the parameters governing the initial noise fluctuation, such as the starting frequency and amplitude, are also significant. A lower initial frequency produces noise with slow and smooth variations. However, a higher initial frequency generates faster oscillations, resulting in the creation of finer and more intricate details within the signal.

In summary, the Perlin noise algorithm is a powerful tool in game graphics due to its ability to create natural phenomena such as terrain surfaces, ripples on the surface of seas or lakes, and even simulate the natural movement of smoke. Therefore, the Perlin noise algorithm has great potential in generating random shapes for interlayer objects between layers of soil, as well as simulating the shape of cavities or delamination phenomena.

2.3. Fully Conventional Network (FCN) neural network model

Fully Convolutional Networks (FCN) have transformed the field of deep learning by enabling dense prediction tasks, such as semantic segmentation, boundary detection, and image restoration to be performed more effectively. Unlike traditional convolutional neural networks (CNNs), which operate on fixed-size inputs, FCN generalize these models to handle inputs of arbitrary size and produce spatially dense outputs. This innovation eliminates the reliance on patch-based

processing or region-based classifiers, thereby marking a significant advancement in deep learning methods. The development of FCN has drawn heavily on the breakthroughs in image classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014) and transfer learning, which have been adapted for dense prediction tasks (Donahue et al., 2013; Zeiler and Fergus, 2014).

The origins of fully convolutional computation were introduced by early works such as Matan et al. (1991), who extended convolutional networks to process one-dimensional input strings. Wolf and Platt (1993) applied these concepts to two-dimensional detection tasks by expanding convnet outputs. Although groundbreaking, these initial implementations relied on heuristic methods such as Viterbi decoding which limited their scalability and flexibility. Contemporary approaches, including those of Pinheiro and Collobert (2014) and Eigen et al. (2013), have significantly advanced the field by integrating all the stages of prediction into unified, end-to-end models. This shift reduces the need for post-processing steps such as super-pixel projection (Farabet et al., 2012) or random field regularization (Ciresan et al., 2012), making FCN more efficient and versatile.

A key innovation of FCN is their ability to adapt traditional classification networks, such as AlexNet and VGGNet, into architectures capable of dense prediction. By reinterpreting fully connected layers as convolutional operations and incorporating techniques such as in-network upsampling, FCN can transform coarse feature maps into detailed, pixel-level predictions. This architectural flexibility allows FCN to train directly on entire images, optimizing spatial loss functions without requiring region proposals or manual feature extraction. Consequently, they achieve exceptional performance in tasks like semantic segmentation on benchmarks such as PASCAL VOC (Everingham et al., 2010) and NYUDv2 (Silberman et al., 2012), while also excelling in boundary detection and image restoration.

Compared with hybrid models that combine region proposals with classification, such as R-CNN (Zhang et al., 2014), FCN offer a simpler, end-to-end solution. Although hybrid approaches deliver competitive results, their reliance on pre-processing and multi-stage pipelines limits their efficiency and scalability. By contrast, FCN streamline the process by leveraging convolutional architectures to directly predict dense outputs, making them particularly suitable for large-scale and real-time applications. By integrating all the stages of prediction into a single framework, FCN represent a paradigm shift in dense prediction tasks. Their success highlights the potential of convolutional architectures to address increasingly complex challenges in computer vision, paving the way for future innovations in multi-modal data integration, real-time systems, etc.

U-Net (Ronneberger et al., 2015) uses a symmetric encoder-decoder architecture, with a global information-reducing encoder and a resolution-recovering decoder, combined via a skip connection to preserve local detail. U-Net achieves high accuracy in segmentation due to its ability to reconstruct details, but the large number of parameters increases the computational cost, and it may be less flexible for unstructured data such as B-scan data. In addition, SegNet (Badrinarayanan et al., 2017) also uses an encoder-decoder architecture, but optimizes efficiency by storing the max-pooling index in the encoder for reuse in decoding, reducing memory and accelerating inference. SegNet is well suited for 2D segmentation, so it may be suitable for real-time applications that require fast speed.

Among the state-of-the-art image segmentation models discussed in Table 1, each model has its advantages and disadvantages. This study proposes FCN due to its benefits and its ability to map input to output, which is well-suited for creating relative permittivity maps using B-scan data from GPR. Unlike R-CNN, which focuses on detecting objects in areas, FCN maps input to output pixel by pixel, allowing it to show material distribution in underground structures. Compared to U-Net, FCN is more flexible and uses skip connections to combine features from different levels, improving accuracy for blurry or noisy GPR data. In addition, fully connected layers are removed in FCN architecture,

Table 1
Comparison of FCN and other models.

Model	Fundamental	References
FCN	- Using fully convolutional layers, eliminating fully connected layers, allowing for arbitrary sized inputs. - Skip connections combine coarse and fine features to increase segmentation accuracy. - End-to-end training, mapping input pixels to outputs.	Long, J. et al. (2015)
R-CNN	- Based on region proposals to detect objects, then classify or segment. - Focus on instance detection, not pixel-wise prediction.	Girshick et al. (2014) Zhang et al. (2014)
U-Net	- Encoder-decoder architecture with bypass connections, focusing on local and global details. - Optimized for structured data.	Ronneberger, O. et al. (2015)
SegNet	- Encoder-decoder architecture, using max-pooling to reduce memory footprint in decoding. - Optimized for fast and efficient inference on 2D data.	Badrinarayanan, V. et al. (2017)

resulting in the processing of images of any size, a distinct advantage over other models that often struggle with varying resolutions. While U-Net, SegNet, and R-CNN each have their outstanding features, the FCN is simple and conserves memory while maintaining effective segmentation. Furthermore, the FCN can be estimated with end-to-end training in a single streamlined process. These characteristics make FCN a strong and efficient choice for mapping GPR data to predict relative permittivity of material based on the characteristics of reflected waves.

In FCN, the *convnet* are organized into three-dimensional data layers of size $h \times w \times d$ in which h and w are the spatial dimensions, and d is the number of channels (Long et al., 2015). The first layer typically corresponds to the input image with d channels, such as a standard RGB image with three color channels. Higher layers in the network connect to receptive fields, which represent the regions in the input image that contribute to the features in the higher layers. Receptive fields play a crucial role in mapping spatial relationships between features in higher layers and corresponding areas in the input space. Convnet are designed to work locally and exhibit translational invariance by using convolutions, pooling, and activation functions. The output value at the position (i,j) in the next layer can be computed as

$$y_{ij} = f_{k,s}\left(x_{si+u,sj+v}\right), \tag{36}$$

where $x_{si+u,sj+v}$ is the input data in the particular layer; k and s are the kernel size and step size of the kernel moving across the input matrix, respectively. Following Long et al. (2015), $f_{k,s}$ is a function that determine the layer type, such as convolution or average pooling, and the partial maximum for the max *pooling* or activation function.

The output of an FCN often takes the form of a coarse feature map with a reduced spatial resolution compared with the input. To predict the labels for each pixel in the input image, these coarse outputs must be up-sampled to match the original input resolution. A common approach is deconvolution, which is also known as transposed convolution. The deconvolution operation can be expressed as:

$$y_{ij}^{c} = \sum_{m=1}^{M}\sum_{u=1}^{H}\sum_{v=1}^{W} w_{uv}^{m,c} \cdot z_{i-u,j-v}^{m}, \tag{37}$$

where y_{ij}^{c} is the output value at position (i,j) for channel c; $z_{i-u,j-v}^{m}$ is the input at position $(i-u.j-u)$ from channel m and $w_{uv}^{m,c}$ is the filter weight connecting the input channel m to the output channel c.

In addition, the loss function in an FCN is often defined to optimize the entire spatial output. The loss function is the sum of the losses at each spatial location and can be expressed as:

$$L(x;\theta)=\sum_{i,j}\ell\left(x_{ij};\theta\right), \tag{38}$$

where x_{ij} is the output at spatial location (i,j) and $\ell\left(x_{ij};\theta\right)$ is the loss at each location such as the cross-entropy or the mean squared error. The loss gradient is computed as the sum of the gradients at each spatial location and it can be expressed as:

$$\nabla_\theta L=\sum_{i,j}\nabla_\theta\ell\left(x_{ij};\theta\right). \tag{39}$$

With overlapping receptive fields, Long et al. (2015) emphasized that whole-image training is faster and equally effective as patch-based sampling, making FCN powerful tools for complex image-processing tasks such as semantic segmentation. Therefore, whole-image training was used in the present study.

3. Evaluation of KIT-GPR program using Perlin Noise for creating interlayer, cavity, and delamination patterns

3.1. Model configuration

Four synthetic models were constructed to validate the computational results of KIT-GPR program. The simulation models for the four cases are shown in Fig. 4. Each model consists of three main layers: a reinforced concrete layer, grout layer, and rock layer. The concrete layer includes rebars with a spacing of 400 mm, and the rebar layer is positioned 150 mm below the top surface of the concrete layer. The material properties of the various materials are listed in Table 2.

Table 2
Material properties for synthetic scenarios.

Material	Relative permittivity ε_r	Conductivity σ (mS/m)
Air (for delamination, cavity)	1	0
Concrete	8	0.0001
Rock	6	0.001
Grout	12	0.1
Rebar	$+\infty$	$+\infty$

In the first case, only the interlayer line is designed, based on the Perlin noise curve. The Perlin noise algorithm parameters selected for this case are as follows: amplitude = 0.3, frequency = 1, persistence = 0.5, and octave = 4. The second case involves both the interlayer and cavity. Meanwhile, the third case includes both the interlayer and the delamination. In this case, the interlayer is initialized between the grout and rock layers, and delamination is automatically created by defining the start and end points as vertices on the interlayer curve. Finally, the fourth case is a composite scenario that includes all three objects: interlayer, cavity, and delamination.

The four computational models were built using KIT-GPR software, and the results were subsequently compared with those obtained from gprMax (Warren et al., 2016). This open-source software can simulate GPR wave propagation through the ground. In addition to the numerical model and material information, the antenna wave parameters are critical. Although the KIT-GPR supports multiple wave types, Gaussian waves are typically used in 2D simulation. Therefore, Gaussian waves were selected for the four synthetic models in this study. The distance between the transmitter and receiver was set to 76 mm, which aligns

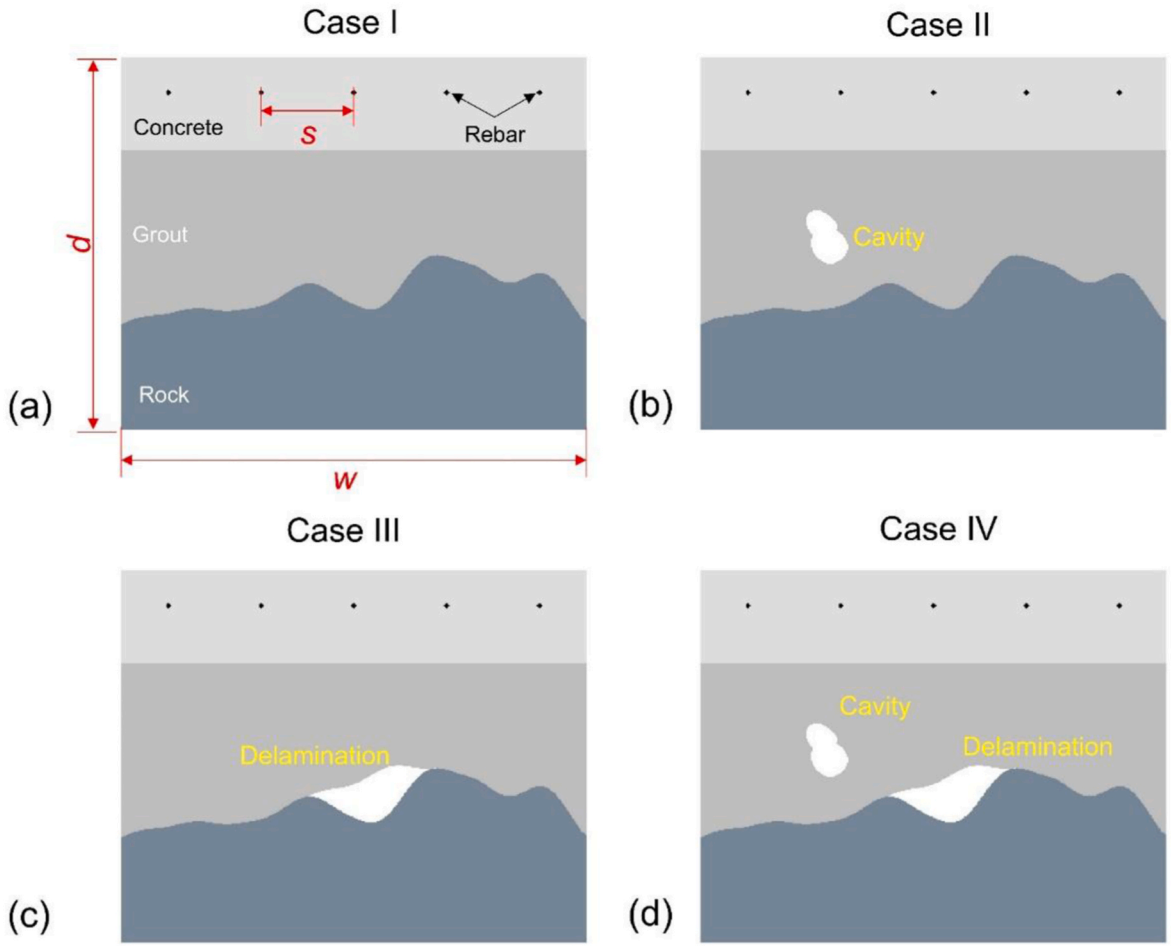

Fig. 4. Synthetic scenarios: (a) Interlayer only, (b) Interlayer with cavity, (c) Interlayer with delamination, (d) Interlayer with cavity and delamination.

with the GPR Antenna MALA 1.2 GHz, a type widely used in South Korea. The excitation frequency is also important and often differs from the nominal frequency of GPR devices. According to the gprMax antenna library (Warren et al., 2016), the optimal excitation frequency is typically determined experimentally in laboratories or through optimization problems based on Taguchi's method (Warren and Giannopoulos, 2011). Specifically, the optimal excitation frequency for the MALA 1.2 GHz antenna, as provided by the gprMax library, is 978 kHz (Warren and Giannopoulos, 2011). Thus, the Gaussian wave, coordinates of the transmitting and receiving points of the GPR antenna, and the optimal excitation frequency were defined in both KIT-GPR and gprMax.

The meshing of the model elements plays a vital role in simulation problems. According to Irving and Knight (2006), the maximum element size in a simulation problem is often determined using the following:

$$\Delta x = \Delta z = \frac{1}{5 \cdot f_{\max} \cdot \sqrt{\varepsilon_{\max} \cdot \mu_{\max}}}, \tag{41}$$

where $f_{\max}$ is the maximum frequency of the source signal, determined by finding the highest value in the frequency spectrum obtained by applying a Fast Fourier Transform (FFT) to the antenna wave; $\varepsilon_{\max}$ and $\mu_{\max}$ denote maximum relative permittivity and permeability, respectively.

This formula was integrated into the KIT-GPR software to assist users in selecting an appropriate element size. For the simulation of the four theoretical cases, the element size in the x- and z-directions was set to 5 mm. With this element size, the maximum time step of the model was calculated using the following formula (Georgakopoulos et al., 2002):

$$\Delta t = \frac{6}{7} \cdot \sqrt{\frac{\varepsilon_{\min} \cdot \mu_{\min}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta z^2}}}. \tag{42}$$

where $\varepsilon_{\min}$ and $\mu_{\min}$ are the minimum relative permittivity and permeability, respectively.

The above equation is also embedded in the KIT-GPR software, enabling users to determine the maximum computational time step and make reasonable choices. Given an element size of 5 mm in both the x and z directions for all four models, the computational time step was set to 1×10^{-11} sec.

Additionally, the scan step of the antenna in the x-direction of the structure must be selected to adequately construct the B-scan data. If the scan step is too large, the B-scan resolution will be poor, whereas a very small scan step improves the resolution but significantly increases the computation time. Therefore, the scanning step must be selected carefully. For a geometry of width 2 m, the scan step selected for this study was 0.01 m. Another critical consideration is the distance between the antenna's emitting point and the surface of the structure, which typically follows the coupling air–concrete principle. Hence, the distance from the emitting point of the antenna to the top surface of the structure was set to 0.2 m. After completing the model setup, A-scan analysis for KIT-GPR and gprMax were compared.

3.2. Evaluation simulation model

This section compares the A-scan and B-scan results obtained from the gprMax (Warren et al., 2016) and KIT-GPR programs, focusing on various subsurface conditions such as uniform interlayers, cavities, and delamination. This aim was to evaluate the effectiveness of both programs (KIT-GPR and gprMax) for modeling and detecting electromagnetic wave propagation through these different materials.

As shown in Fig. 5(a), the cross section with a uniform structure and having two distinct subsurface layers: concrete and rock. The red dashed line represents the propagation path of the electromagnetic waves analyzed in the A-scan. Waveform analysis demonstrates that gprMax and KIT-GPR yielded nearly identical results, showing a strong agreement in modeling the propagation of electromagnetic waves through the uniform interlayer structure. In contrast, the cross section including a cavity within the interlayer is shown in Fig. 5(b). The upper layer consists of concrete, whereas the lower layer is composed of rock and is interrupted by a cavity. The waveform results reveal a notable deviation in the signal corresponding to the cavity region, effectively differentiating it from the surrounding materials. This anomaly is accurately detected using the KIT-GPR program, demonstrating its capability to identify subsurface irregularities. The upper inset highlights the signal from the concrete layer, demonstrating the close alignment between gprMax and KIT-GPR. The lower inset focuses on the rock and cavity signals, confirming that KIT-GPR accurately models the rock layer while effectively identifying the presence of the cavity. The cross section clearly shows the cavity as a space between the concrete and rock layers, which alters the electromagnetic response and produces distinct signals in the A-scan. The KIT-GPR program successfully detected and distinguished the signals from those of the surrounding layers.

As shown in Fig. 6, a comparison of the A-scan results of the gprMax and the KIT-GPR programs for Case III, featuring an interlayer with delamination evaluated at two points, *A* and *B*, in the subsurface structure. At Point *A*, the cross section shows a uniform interlayer structure comprising an upper concrete layer and a lower rock layer without significant delamination. Waveform analysis demonstrated that gprMax and KIT-GPR provided closely matching results, confirming

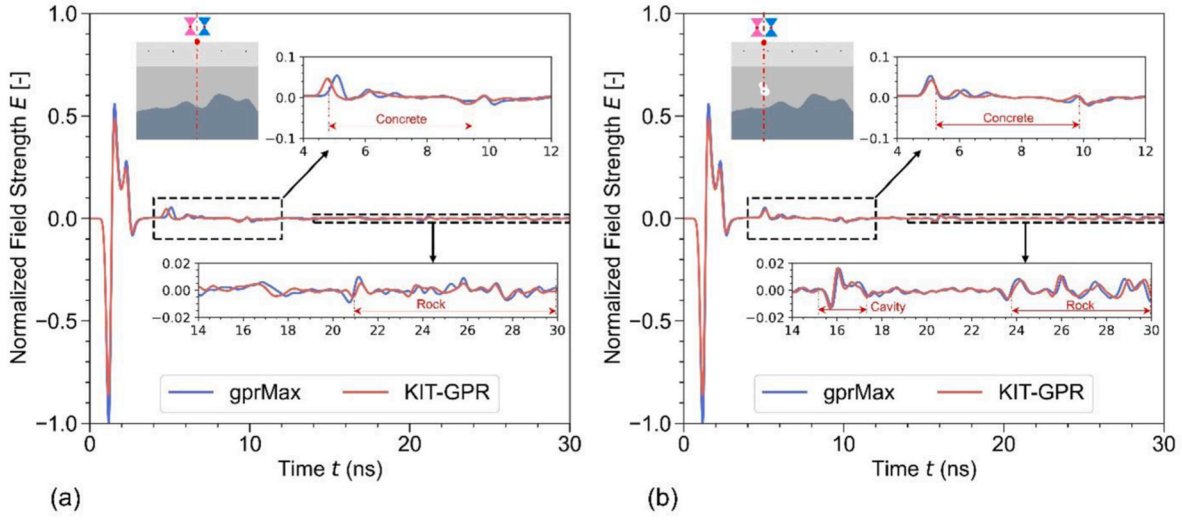

Fig. 5. Comparison of A-scan between KIT-GPR and open-source gprMax program: (a) Case I: Interlayer only, (b) Case II: Interlayer with cavity.

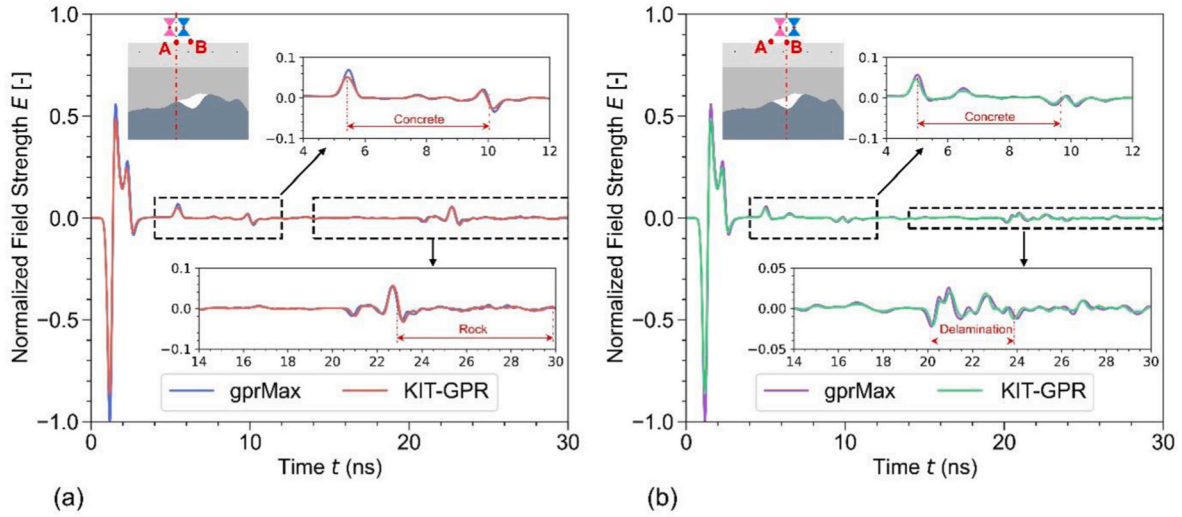

Fig. 6. Comparison of A-scan between KIT-GPR and open-source gprMax program for case III – Interlayer with delamination: (a) A-scan at Point A, (b) A-scan at Point B.

their reliability in modeling electromagnetic wave propagation through homogeneous subsurface materials. At Point B, the cross section introduces delamination within the interlayer materials, creating a separation between the layers and altering the electromagnetic response. The waveform analysis reveals a deviation in the signal corresponding to the delamination region, which was effectively detected using the KIT-GPR program.

According to Fig. 7, a comparison of the A-scan results from gprMax and KIT-GPR for Case IV, featuring an interlayer with a cavity and delamination, evaluated at two points, *A* and *B*, in the subsurface structure. At Point *A*, the cross-section displays a cavity embedded within the rock layer beneath the concrete, disrupting the continuity of the rock material and causing a distinct electromagnetic wave response. KIT-GPR effectively detects this anomaly, with results closely matching those of gprMax. The upper inset highlights the concrete layer, showing consistent signal alignment between the two programs, whereas the lower inset focuses on the cavity, where KIT-GPR accurately models the anomaly and differentiates it from the surrounding rock layer. At Point *B*, the cross section reveals delamination characterized by a separation between the concrete and rock layers, altering the electromagnetic response and generating a unique signal. KIT-GPR successfully identifies and distinguishes the delamination from the regions that are intact, showing strong alignment with the output of gprMax. The upper inset shows the signal from the concrete layer, revealing minimal deviation between the two programs, whereas the lower inset captures the delaminated region within the rock layer, where KIT-GPR precisely detects and models the anomaly. The cross-sections above the plots illustrate the paths of electromagnetic wave propagation, marked by red dashed lines, reflecting the distinct responses of the cavity at Point *A* and delamination at Point *B*.

In the analysis and evaluation of GPR results, the B-scan plays a crucial role in visualizing data and extracting information from geological cross-sections. The B-scan is formed by compiling A-scan data, each representing a measurement at a specific location. As illustrated in Figs. 5–7, the amplitude of the direct wave traveling from the transmitter to the receiver is typically significantly greater than that of the reflected waves, particularly when the waves traverse the boundary between two different materials. This makes it challenging to identify the reflected waves from small objects or layers deep within the material. In addition, radar waves tend to decrease in amplitude over time owing to energy attenuation as they propagate through material layers, rendering the raw data less clear.

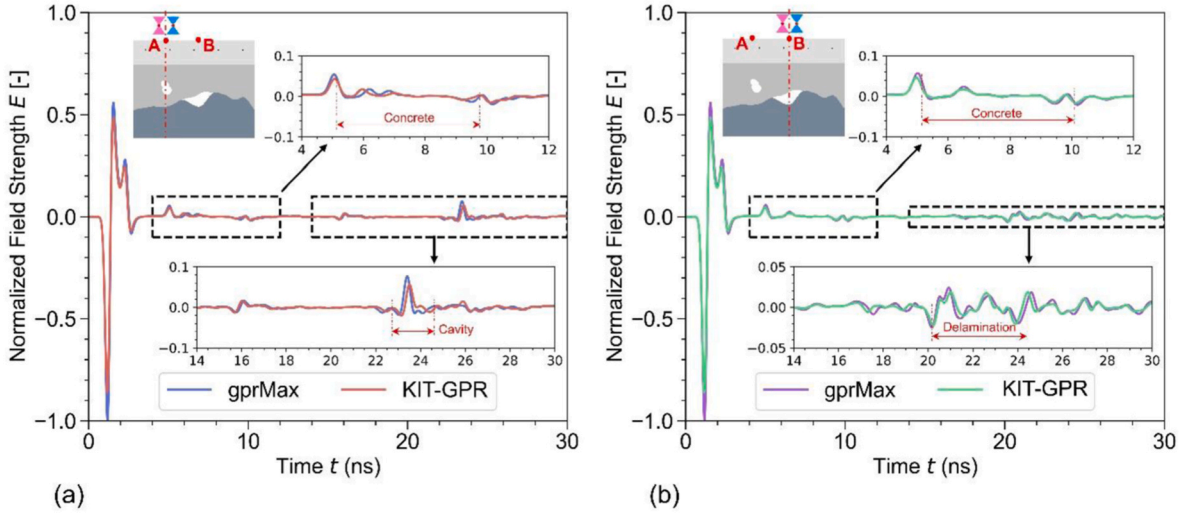

Fig. 7. Comparison of A-scan between KIT-GPR and open-source gprMax program for case IV – Interlayer with cavity and delamination: (a) A-scan at Point A through cavity, (b) A-scan at Point B through delamination.

Owing to the weak amplitude of reflected waves and their being overshadowed by the direct waves, B-scans generated from the original A-scan data often struggle to clearly display critical features such as reflected waves from objects such as rebars, cavities, and delaminations. As shown in Fig. 8 (a), a typical example of this issue, in which the reflected features are obscured by the magnitude of the direct wave. To address this, a gain factor function is applied to enhance the amplitude of the radar reflections over time, thereby highlighting the critical features in the data. In contrast, the B-scan in Fig. 8(b) is enhanced with the gain factor, while Fig. 8(c) clearly depicts the configuration of the gain factor function. This function typically starts at a value of one and gradually increases over time, helping balance the wave amplitude relative to the propagation distance.

The application of the gain factor enables regions with significant reflection spikes to become more recognizable. This is particularly important for detecting objects such as rebars, cavities, and delaminations, which often exhibit weak signals in the original data. For example, Fig. 8 (b) illustrates a theoretical scenario with these objects and demonstrates how the gain factor accentuates the reflected waves at relevant locations. Without applying a gain factor, identifying these objects becomes challenging because the amplitudes of the reflected wave at these points are much smaller than the direct waves emitted from the GPR antenna.

As shown in Fig. 9, the effectiveness of the gain factor for four different theoretical cases. In the first case (interlayer-only, Fig. 9(a)), the peaks and troughs of the interlayer curve clearly display reflections, allowing for a straightforward identification of the material layer boundaries. In the second case (interlayer and cavity, Fig. 9(b)), the reflections at the cavity exhibit a pattern similar to that of the interlayer peaks and troughs, but with more successive wave peaks owing to the strong reflection and reduced attenuation caused by the air inside the cavity. The third case (delamination, Fig. 9(c)) demonstrates significant differences at the start and end points of delamination, where interactions among the grout, rock, and air occur. These points are easily identifiable because of the strong variations in the amplitude and shape of the reflected waves. In the fourth case (interlayer, cavity and delamination, Fig. 9(d)), the reflections at the delamination show a higher amplitude and reach their peaks faster than those at the cavity. Simultaneously, the cavity, with its homogenous air composition, produces more successive wave peaks owing to lesser energy attenuation compared with solid materials.

In summary, applying the gain factor not only enhances the visualization quality of B-scan data, but also improves the detection and analysis of critical objects, such as rebars, cavities, and delamination. This underscores the importance of signal processing in GPR applications to optimize the accuracy and efficiency of studying complex geological structures.

4. Development of FCN model for predicting cavity and delamination positions

4.1. Fully conventional network architecture

In this study, an FCN model was designed and developed to predict the locations of tunnel damages. As depicted in Fig. 10, the architecture of the FCN framework is designed to predict the locations where damages within in tunnel lining. The input to the model is a numerical matrix consisting of multiple columns, with each column representing an A-scan dataset at a specific point along the scanning path of the GPR antenna. These data are initially calibrated to values between 0 and 255, corresponding to the grayscale intensity levels. Before being fed into the FCN model, the matrix is normalized to a range of [0; 1] and interpolated into a square matrix of size 320 × 320. Interpolation is performed along both the X- and Y-dimensions of the input matrix because the number of scanning steps is typically much smaller than the number of time steps in the A-scan data.

To enable the FCN model to identify correlations between the input matrix and the predicted geometry, a parameter matrix of permittivity values is defined as the target matrix for training the FCN. This target matrix has a size of 320 × 320, adhering to the fundamental rail-to-rail operational principle of the FCN. The permittivity matrix is generated using the KIT-GPR software based on the user-defined material matrix. The input and target matrices share the same 320 × 320 size for several reasons. First, the selection of a square matrix ensures symmetry in the corresponding weight matrices, thereby facilitating training. Furthermore, square matrices simplify the interpolation of any B-scan data with asymmetric dimensions, such as cases where the scan width exceeds the depth, or vice versa. Because the FCN model does not strictly require the original input image dimensions, the use of square matrices simplifies the computation of weight matrices during training, which relies on

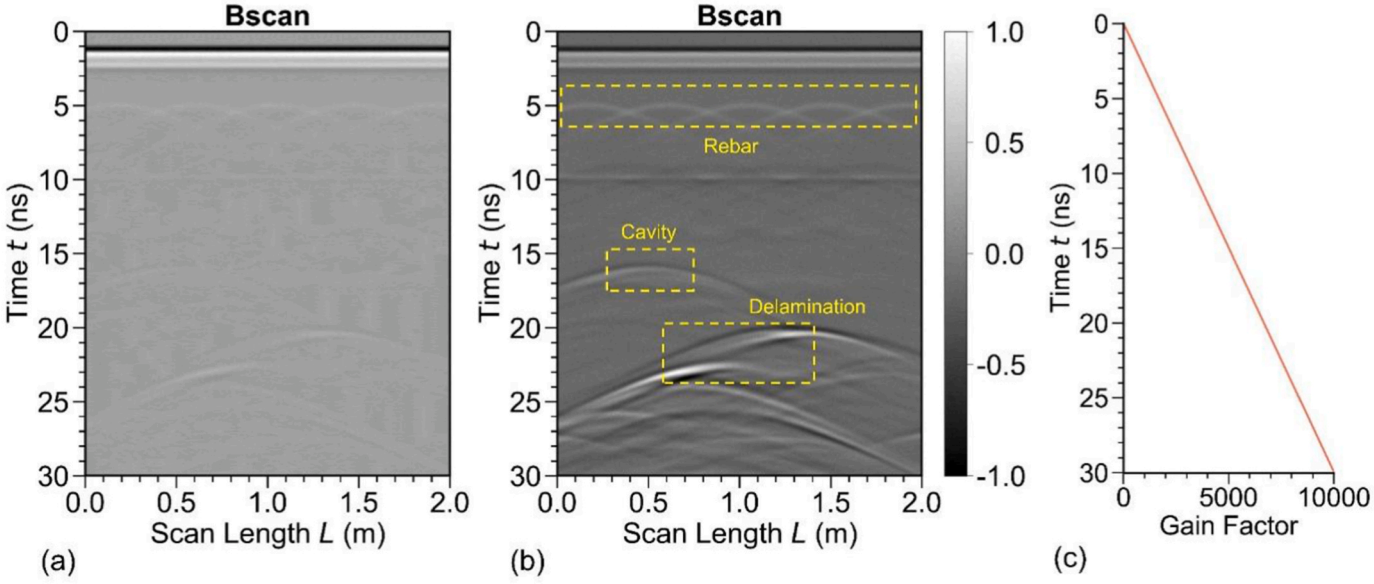

Fig. 8. Example of B-scan analysis and gain factor visualization for Case IV: Interlayer with cavity and delamination. (a) B-scan without gain factor, (b) B-scan with gain factor, (c) Gain factor over time.

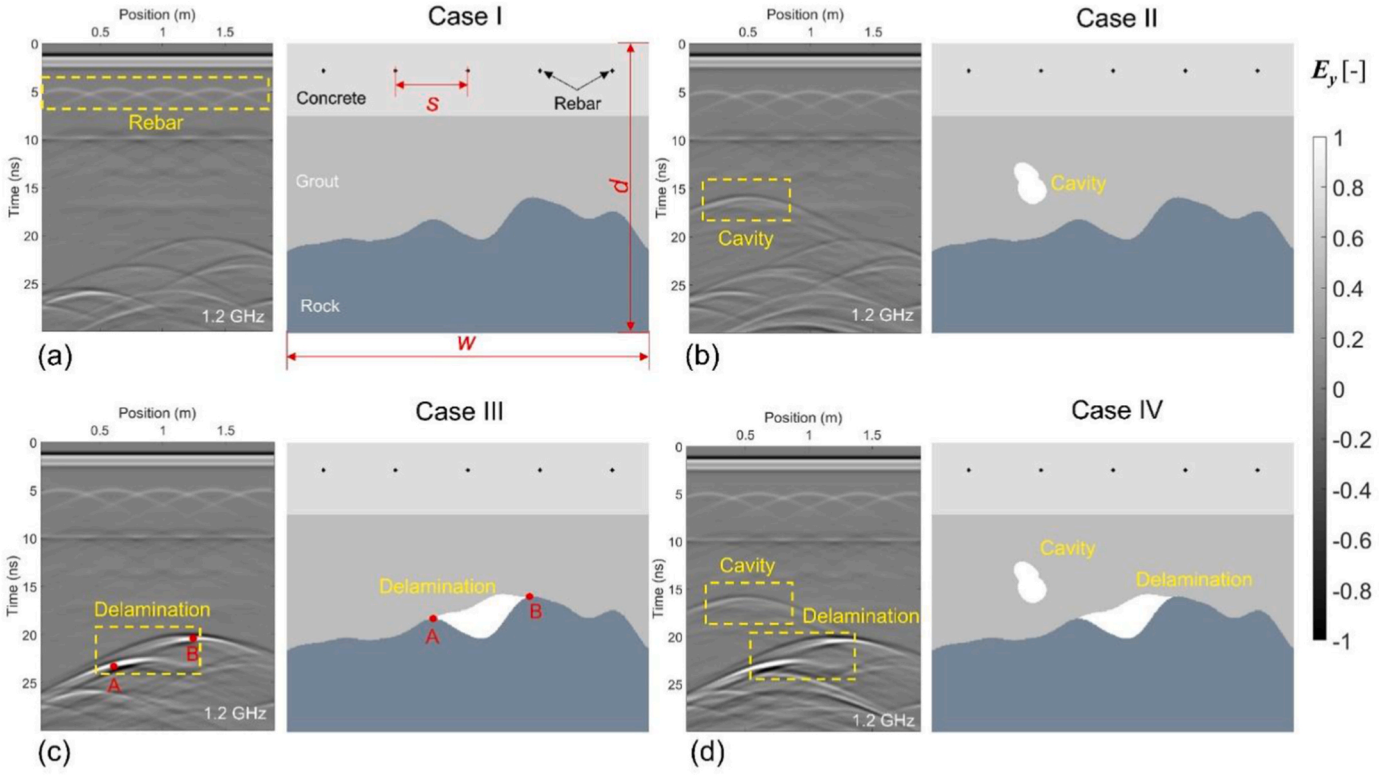

Fig. 9. B-scan with gain factor visualization for synthetic scenarios: (a) Case I – Interlayer only, (b) Case II – Interlayer with cavity, (c) Case III – Interlayer with delamination, (d) Case IV – Interlayer with cavity and delamination.

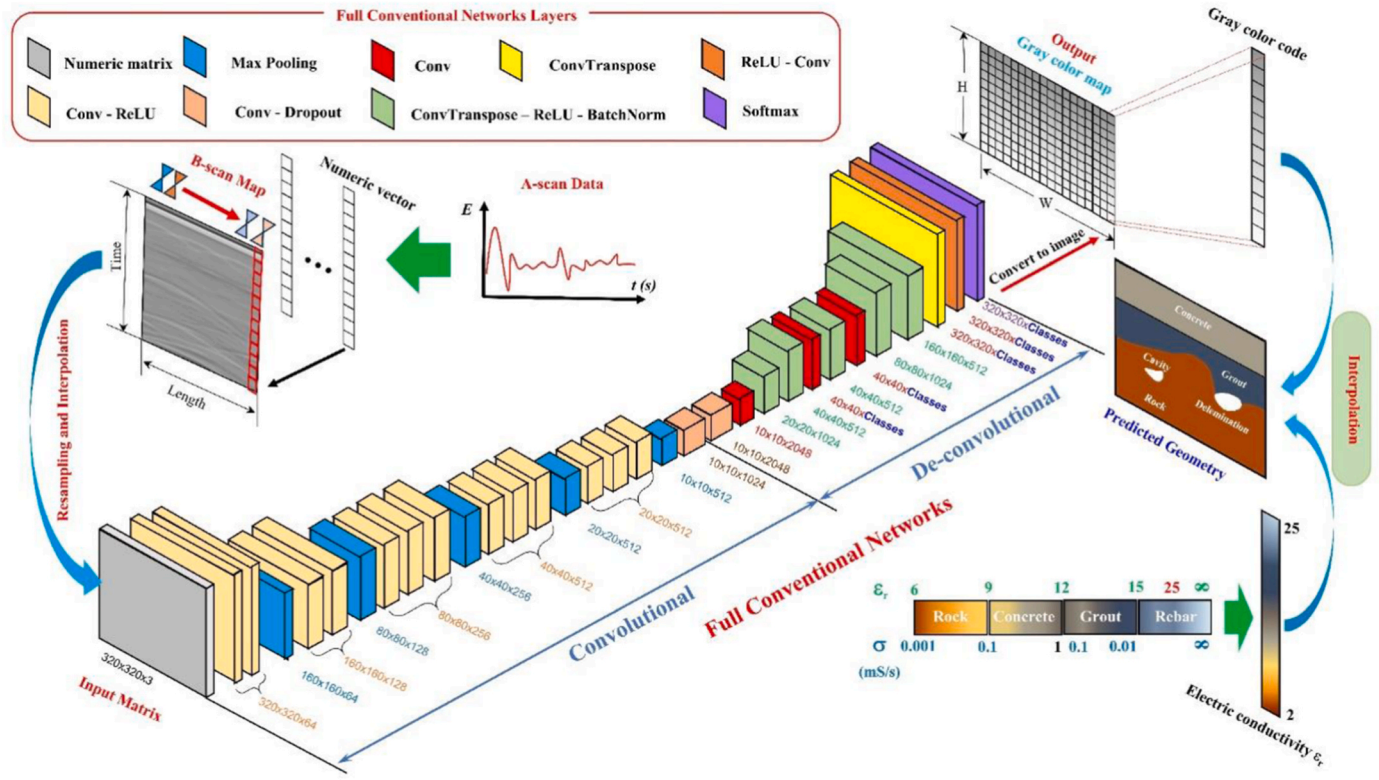

Fig. 10. Schematic diagram of Fully Conventional Networks for predicting local defects in tunnel lining.

paired input and target data. Moreover, selecting a matrix size of 320 × 320 optimized the computational efficiency during model training.

Four characteristic defect scenarios were developed: interlayer only, interlayer with cavity, interlayer with delamination, and a combination of interlayer, cavity, and delamination. Each scenario comprises 250 models, yielding a total of 1000 input models for the FCN prediction model. The FCN model utilizes pre-trained weights from VGG-16 to generate an initial feature matrix. Instead of using text-based labels, this study proposes 256 classes in depth dimension of model, corresponding to the 0–255 encoding of input images and representing dielectric constant values ranging from 0 to 25.5. Notably, rebar typically has a very high dielectric constant; thus, in this study, rebar positions are assumed to have a maximum dielectric constant of 25.5. Consequently, the input data for FCN training consists of B-scan data and dielectric permittivity maps. The B-scan data are synthesized from A-scan data collected at antenna receiver points, while the dielectric permittivity maps are generated based on geometry and randomly assigned material data from user-defined value ranges. Smoothing algorithms are also applied to create transition regions at material boundaries using an averaging method in numerical model.

To build an FCN model, selecting an appropriate architecture is crucial for influencing the performance and accuracy of the training process. According to Long et al. (2015), FCN architectures are often adapted from previously validated classification networks such as AlexNet (Krizhevsky et al., 2017), VGG (Simonyan and Zisserman, 2014), and GoogleNet (Szegedy et al., 2015). First, a validated classification model is converted into an FCN by removing the final classification layer and replacing the fully connected layers with convolutional layers. Next, an additional convolutional layer is appended to predict the class scores (including the background class) at the raw output locations. To increase the resolution from raw outputs to pixel-dense predictions, an upsampling mechanism is introduced through a deconvolution layer. The initial weights of the deconvolution layer are set to perform bilinear interpolation and are subsequently optimized during the training process.

After conversion, the network is fine-tuned for segmentation tasks using a pixel-wise multiclass logistic loss function. Skip connections were employed to enhance the spatial accuracy by integrating information from the hierarchical layers into three segmentation variants: FCN-32s, FCN-16s, and FCN-8s. The indices "32," "16," and "8″ represent the pooling layers (down-sampling layers) in the network and the degree of resolution reduction of features during training. The FCN-32s model represented the architecture with the most aggressive downsampling, in which the resolution of the output was significantly lower than that of the input. The FCN-16s produces higher-resolution outputs than FCN-32s by reducing the dimensionality to a lesser extent. FCN-8s achieved the highest resolution among the three models by preserving the most detailed information.

Specifically, predictions from the stride-16 layer (pool4) were combined with those from the stride-32 layer (conv7) through an upsampling and addition mechanism to form the FCN-32s segmentation model. For the FCN-16s, information from pool4 (stride 16) and conv7 (stride 32) is integrated. Furthermore, FCN-8s incorporates additional information from pool3 (stride 8), resulting in finer details. As described by Long et al. (2015), the models (FCN-32s → FCN-16s → FCN-8s) are trained sequentially.

4.2. Fully conventional network configuration

In the FCN model, the enhancements introduced by skip connections help preserve detailed structures and improve the spatial accuracy of the predictions. The training parameters were optimized using Stochastic Gradient Descent (SGD) with hyperparameters, such as batch size, learning rate, and momentum, chosen experimentally. A dropout is also applied to prevent overfitting. In this study, the PyTorch library is a powerful tool within the Python programming language is utilized, enabling flexible and efficient implementation of the model. The loss function selected is the Binary Cross-Entropy Loss (BCELoss) from the *torch.nn* library, which calculates the difference between the predicted probabilities and ground truth labels at each pixel. This function is particularly suitable for binary segmentation tasks and requires the output of the network to pass through a sigmoid activation function to ensure that the values are within the range [0,1]. To optimize the model, the Adam optimizer from the *torch.optim* module is used with an initial learning rate of $l_r = 0.0001$, which was carefully chosen to ensure stability during convergence. Adam is an efficient algorithm that combines the advantages of SGD and RMSProp, allowing for fast parameter updates even on large and complex datasets. Additionally, the Exponential Learning Rate Scheduler from *torch.optim.lr_scheduler* with a decay factor of $\gamma = 0.985$ is applied, ensuring that the learning rate decreases smoothly over epochs. This aids the model in fine-tuning the parameters during the later stages of training, improving stable convergence, and reducing the risk of overfitting. The selection of a proven classification model is crucial prior to building a model. In this study, the VGG-16 model was selected and adapted to a segmentation architecture for FCN tasks.

The detailed architecture of the FCN along with its input and output matrix dimensions is shown in Fig. 10. In addition, the FCN is divided into two main components: convolutional and deconvolutional. In the first part of FCN, the input data passes through a series of convolutional layers such as the Conv-ReLU layer, Max Pooling layer, and Conv-Dropout layer. The Conv-ReLU and Max Pooling layers are arranged in an interleaved manner with decreasing matrix size and increasing depth. The purpose of this arrangement is to extract features of the input data at decreasing spatial resolution but representing features at increasing depth. The last layer of the series of convolutional layers in FCN is the Conv-Dropout layer to prevent overfitting. Thus, the convolutional stage compressed data information from larger spatial dimensions (320 × 320 × 3) to smaller and deeper feature maps (10 × 10 × 2024). The ReLU layer applies a nonlinear activation function to enhance feature nonlinearity and suppress negative values. Meanwhile, the max pooling layer reduces the spatial dimensions of the feature maps to summarize critical information and decrease the number of parameters.

The deconvolutional part reconstructs the output image from the extracted features using key layers such as the deconvolutional layer and skip connections. The deconvolutional layer performs the inverse operation of convolution, expanding spatial dimensions to restore information, whereas skip connections combine detailed information from the convolutional part with semantic information from the higher layers, thereby enhancing the pixel-wise segmentation accuracy. The first part of the deconvolutional layers is the Conv-Dropout layer, which keeps the dimension but increases the depth of the output of the convolutional layers. The Conv and Conv-Transpose-ReLU-BatchNorm layers are combined to increase the spatial size and decrease the depth. Among these layers, the Conv layer has the same number of layers as the number of classes at the output of the FCN model. The purpose of these layers is maintain the learned features for segmentation with high accuracy. When the data passes through the deconvolutional, it is converted back into a high-resolution feature map with a separate channel for each output layer. The last layers include 3 layers: Conv-Transpose, ReLU-Conv, and Softmax. The combination of these three layers with the depth corresponding to the number of output classes creates a probability map for every spatial location in the form of a grayscale image. Following predicted data, the electric conductivity value is determined, helping the visualization of defects in the tunnel lining, such as interfaces, cavities, and delamination.

Furthermore, the output of the FCN is a 320x320 pixel-by-pixel classification map, where each pixel encodes the predicted electricity conductivity onto the grayscale image. The interpolation technique is applied in post-processing to ensure a smooth and accurate correspondence with the predicted geometry of the tunnel lining.

4.3. Automatic data generation for FCN training using KIT-GPR

Data preparation plays a crucial role in training FCN models. Each data pair consists of an input B-scan and an output permittivity map. The permittivity map is generated using the KIT-GPR software based on user-defined geometry. A smoothing interpolation method is applied to create transition zones between the materials using user-provided material parameters. The B-scan is formed from a collection of A-scans calculated at various scan steps using the numerical model in KIT-GPR. These data pairs are used as inputs for training the FCN model.

Although the KIT-GPR software allows users to define geometry and material properties. However, constructing a large dataset requires automation tools to diversify geometries while ensuring the presence of interlayers between the grout and rock layers, delamination formation on interlayers, and cavities in the grout layer. Using computational functions in KIT-GPR, a training datasets generation tool for the FCN model was developed. This tool randomly generates interlayers and cavities using the Perlin noise algorithm. However, creating a delamination requires an automated algorithm. To this end, an automated procedure for identifying positions and generating delamination based on interlayer curves was developed, as illustrated in Fig. 11. First, convex and concave points (referred to as corner points) on the interlayer curve are identified. Next, the upper triangles (triangles with bases on the curve) are determined. The area of each triangle is calculated, and the one with the largest area is selected. The two points on the base of this triangle are designated as the start and end points of the delamination on the interlayer curve.

Subsequently, a 1D Perlin noise function is applied to draw a curve connecting these two points, incorporating part of the interlayer curve between them to generate the delamination. Notably, the shape of the interlayer curve is produced by randomly varying the parameters of the Perlin noise algorithm (Table 3 and Fig. 12(a)). By contrast, the fixed parameters listed in Table 4 are used to draw the cavity curves (Fig. 12 (b)). The cavity positions are randomly generated but must reside within the grout layer and maintain a minimum distance of 0.5 m from the left and right boundaries. Similarly, the Perlin noise parameters are fixed to draw curves connecting the start and end points of the interlayer curve during automated delamination generation. While generating training data for the FCN model, the KIT-GPR program randomly selects the relative permittivity and conductivity values of the materials within a predefined conventional range. The parameter ranges of these materials are listed in Table 5.

Table 3
Range values of Perlin noise's parameters for creating interlayer curve.

Parameter	Minimum value	Maximum value
Amplitude of curve (m)	0.1	0.5
Frequency	0.01	0.1
Persistence	0	1
Octaves	1	5

In this study, 250 geometries were automatically created for each case: interlayer only, interlayer with cavity, interlayer with delamination, and interlayer with both cavity and delamination. A total of 1000 models were generated, corresponding to the 1000 data pairs used for FCN training and validation. Fig. 13 illustrates some examples of geometries for these cases.

4.4. Training process of FCN model

Upon completing the creation of 1000 models, an automated tool of KIT-GPR was developed to read the model data files, perform calculations, and export the CSV files containing the B-scan data and permittivity maps. These synthetic datasets comprise various models of cavities, delamination, and interlayers between grout and rock layers. The simulation models use random input parameters like relative permittivity, conductivity, cavity positions, and changes in the Perlin noise algorithm to create varied interlayers. In addition, the automatic finding of delamination positions is adapted to improve the accuracy of the model. This approach can generate diverse datasets with many defect types of tunnel lining, allowing the FCN model to train on a theoretical database to handle all possible real-world scenarios. The CSV files were subsequently converted and loaded into the FCN model using a custom Python program. The dataset was divided into three subsets: training set (80 %), validation set (10 %), and testing set (10 %). The data were randomly shuffled 100 times before partitioning to ensure

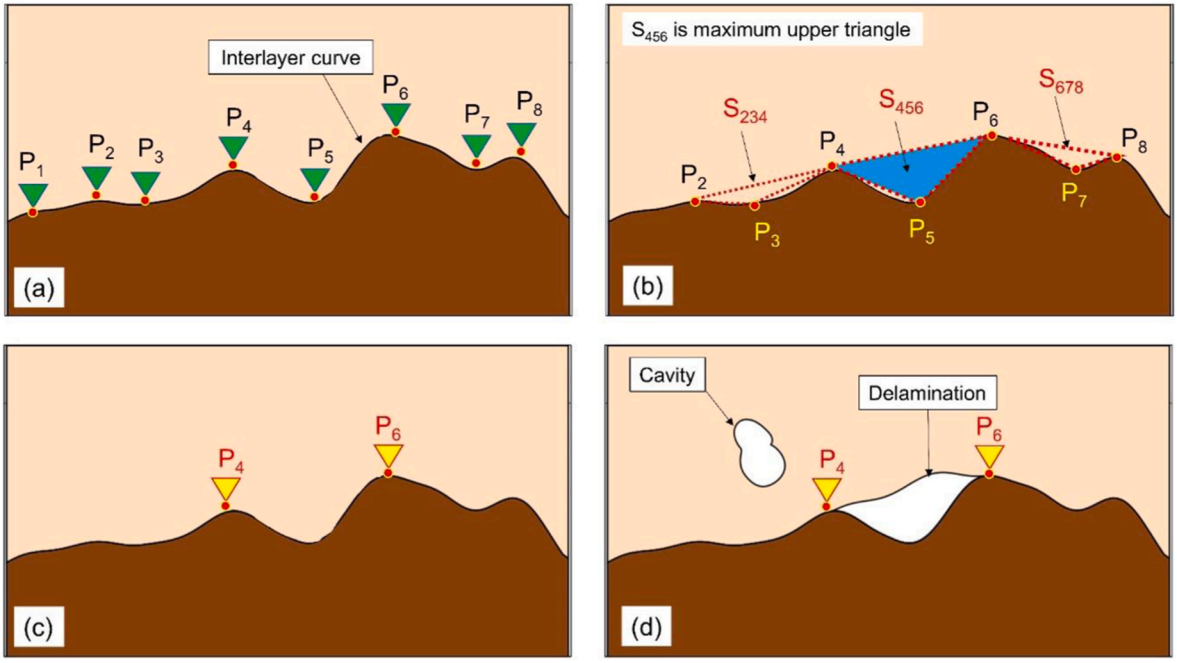

Fig. 11. Steps on automatic delamination detection based on the interlayer curve: (a) Initial corner points, (b) Largest upper triangle, (c) Two points on the upper base of the largest triangle, (d) Drawing the remaining curve for delamination.

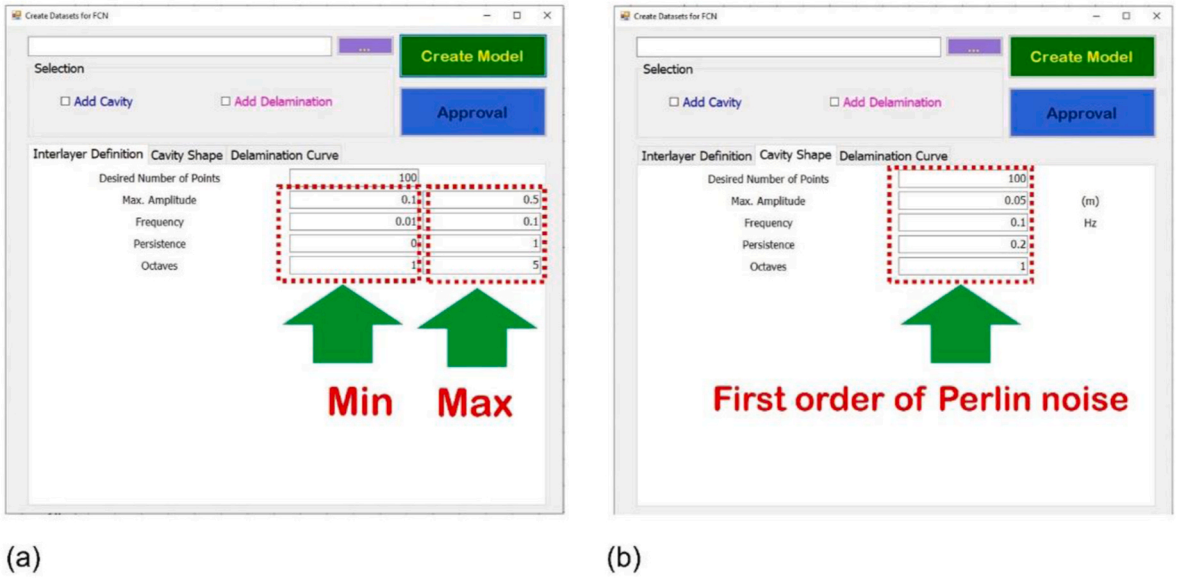

Fig. 12. Creating datasets for FCN in the KIT-GPR program: (a) Interlayer definition, (b) Cavity definition.

Table 4
Parameter of Perlin noise for creating delamination and cavity.

Material	Value
Amplitude of curve (m)	0.05
Frequency	0.1
Persistence	0.2
Octaves	1

Table 5
Range values of material properties for creating dataset.

Material	Relative permittivity ε_r		Conductivity σ (mS/m)	
	Minimum	Maximum	Minimum	Maximum
Rock	6	9	0.001	0.1
Concrete	9	12	0.1	1.0
Grout	12	15	0.01	0.1

balanced distribution across sub-datasets. The training and validation sets were used during model training, whereas the testing set was reserved to evaluate the model's prediction performance. The FCN model is trained through 1000 epochs using the cross-entropy loss function and the Adam optimizer (learning rate 0.0001). In addition, the Exponential learning rate scheduler ($\gamma = 0.985$) helps to fine-tune the parameters, reducing the risk of overfitting. Data augmentation techniques (random flipping, rotation) and GPU system are used to enhance the training efficiency. The process of data generation, training, and validation of the FCN model performance is shown in Fig. 14.

The loss function values for the training and validation datasets is shown in Fig. 15. Overall, the loss values for both sets exhibited a decreasing trend as the number of training steps increased, with the gap between these values stabilizing after the 400th step. Notably, overfitting occurred during the first 50 training steps when the validation loss was significantly smaller than the training loss. Conversely, underfitting occurred between the 50th and 300th steps, as the validation loss significantly exceeded the training loss. After the 400th step, the performance of the model became more stable, with a consistent gap between the loss values of the validation and training sets, although the validation loss fluctuated more than the training loss. As the gap between these values remained unchanged, the weight matrix from the 1000th training step was saved and used to evaluate the performance of the model based on the predictions for the testing set.

4.5. Evaluation of FCN model

To evaluate the performance of the trained FCN model, a test set was used to assess its predictive capabilities. The testing set comprised 100 samples. However, for a comprehensive evaluation, a few randomly selected cases were chosen to cover diverse scenarios, including interlayer only, cavity only, delamination only, and cases involving both cavity and delamination. These results are shown from Figs. 16–18. Each case is represented by four images: the input B-scan data without the gain factor applied, ground truth data, permittivity map predicted by the FCN model, and a profile plot displaying the values of the dielectric constant by depth at specific locations, facilitating the identification of material layer thicknesses. In the profile diagram, two lines are depicted: a solid line representing the dielectric constant predicted by the FCN model, and a dotted line indicating the initial those of value from ground truth geometry.

As shown in Fig. 16(a), the predicted permittivity map for the interlayer-only case. The results reveal challenges in accurately reconstructing the interlayer curve as per the ground-truth geometry. However, the model predicted key points such as the highest peaks and lowest troughs of the interlayer curve relatively accurately, although segments with smaller amplitudes were largely undetectable in the predictions. The profile at Line I highlight the variations in the dielectric constant with depth, enabling the estimation of the layer thicknesses and material types based on the dielectric constant ranges of the corresponding materials. In contrast, the predicted permittivity map for the case with interlayer, cavity, and delamination are shown in Fig. 16(b). Notably, the cavity and delamination in the ground truth are positioned close to each other, making this a unique scenario for examining their mutual influence. Based on the FCN-predicted permittivity map, the cavity and delamination positions were identified with reasonable accuracy. However, the predicted delamination size was significantly smaller than that of the ground truth geometry. Profile analysis at Line II revealed the thickness and positions of the concrete, grout, and rock layers with relative ease, despite minor noise. The position of the cavity, which caused abrupt changes in the dielectric constant, was clearly identified within the grout layer. The delamination position was also detected as it lies between the grout and rock layers. The proximity of the cavity and delamination positions indicates that, while the FCN

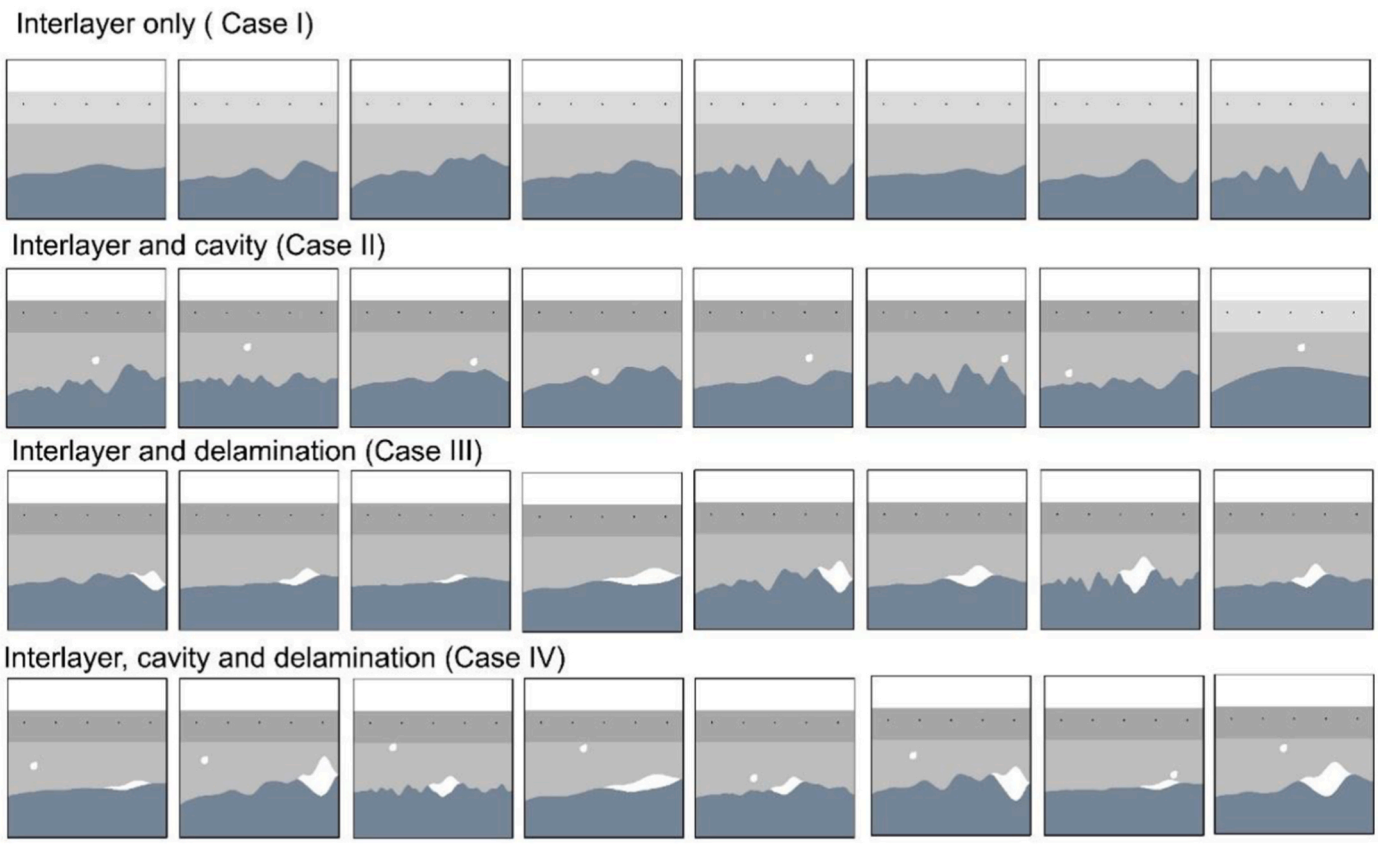

Fig. 13. Example of synthetic scenarios from KIT-GPR program.

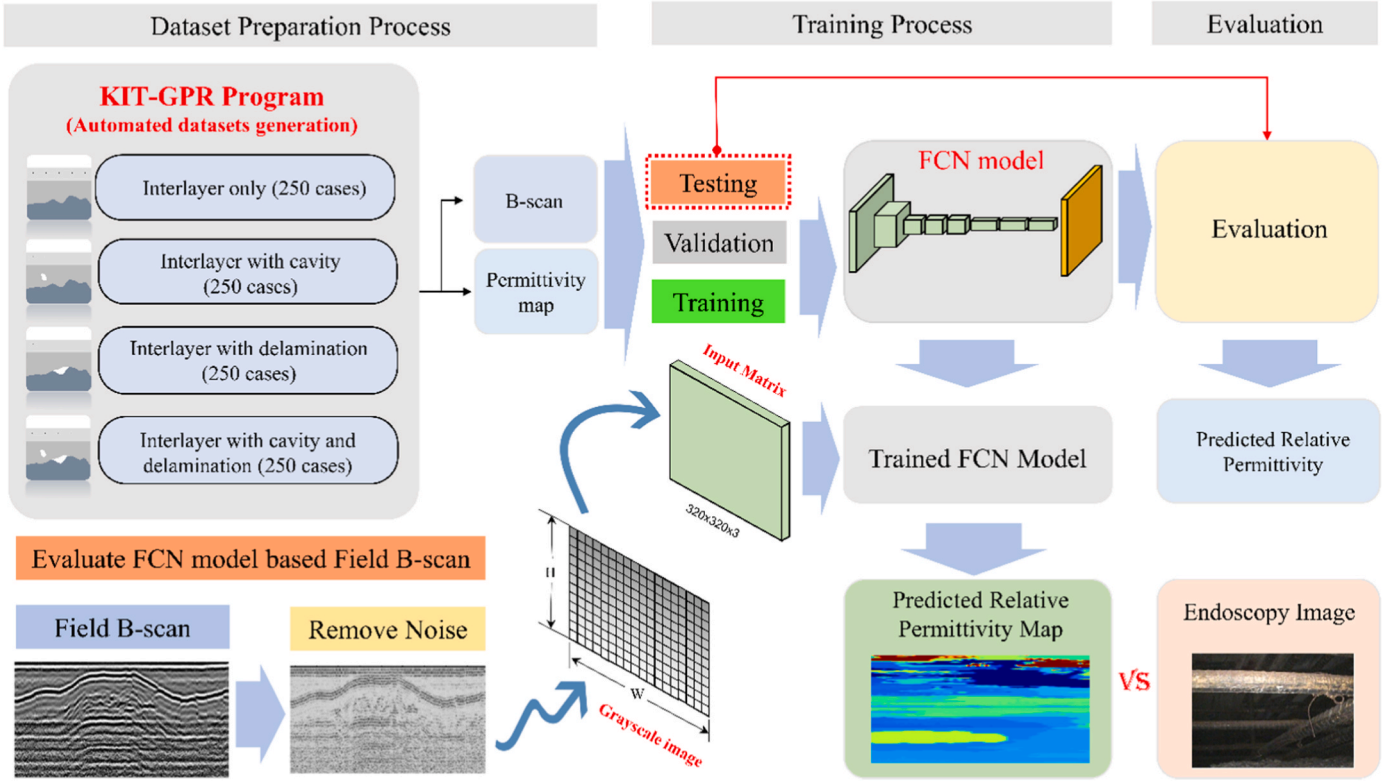

Fig. 14. Training and Evaluation Processes of the FCN model.

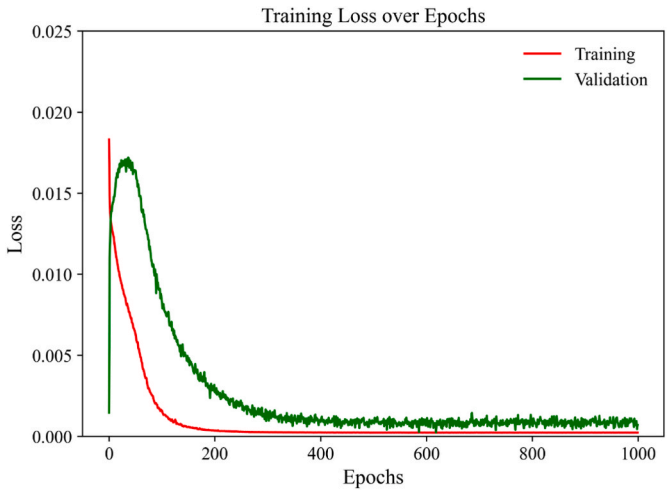

Fig. 15. Training performance metrics of the FCN model.

model can predict their locations accurately, the cavity shape is more precisely identified than the delamination. The smaller delamination shape in the predictions is likely due to wave interference effects when the cavity is near the delamination.

The case with interlayer, cavity, and delamination, where the cavity is located within the grout layer, is illustrated in Fig. 17(a). The profiles at Line III and IV show that the cavity and delamination positions were readily identifiable based on the dielectric constant variations with depth. Remarkably, the predicted permittivity map closely matched the ground truth, demonstrating a strong model performance when the cavity positions are unaffected by delamination. The case, where the cavity is located near the interface between the grout and reinforced concrete layers, is shown in Fig. 17 (b). An additional artifact structure appeared near the cavity position in the prediction owing to boundary effects. Moreover, the interlayer position and shape were accurately predicted, closely resembling the ground-truth geometry.

The two remaining cases—one with only a cavity and one with only delamination—are demonstrated in Fig. 18. The predicted permittivity maps indicate that the cavity positions aligned well with the ground truth, whereas the interlayer shape between the grout and rock layers deviated from the ground truth. Based on the profile at Line VII, the cavity position and material layer thicknesses were relatively straightforward to determine.

The Root Mean Square Error (RMSE), which quantifies the error between the dielectric constant value from ground truth and that predicted by the FCN model, is statistically evaluated for each case presented in Table 6. Based on the results in Table 6, the FCN model demonstrates varying prediction accuracy for dielectric values, as measured by the RMSE, ranging from 0.814 to 2.993, depending on the scenario. The model achieves the lowest error (RMSE = 0.814) in the case of an interlayer with delamination (Line VIII), indicating strong predictive capability in scenarios where the dielectric value is less distorted. It also performs well (RMSE = 1.456) when the cavity is positioned in the middle of the grout layer combined with delamination (Line III). Simpler scenarios, such as the interlayer only (Line I, RMSE = 1.655) or with a cavity in the middle of the grout layer (Line III, RMSE = 1.456), yield reasonably good results. Conversely, performance drops significantly in more complex scenarios, such as when the cavity is under the concrete layer and near delamination (Line II, RMSE = 2.706; Line VI, RMSE = 2.993), likely due to weakened dielectric signals or interactions between material layers complicating the prediction. Comparing similar cases, RMSE is generally lower when the cavity is positioned at a detectable depth (e.g., in the middle of the grout layer) rather than near delamination, where signal interference may occur, though the error notably increases when the cavity is obscured by concrete.

Overall, the FCN model excels in less distorted scenarios but struggles in complex scenarios where cavities are obscured by concrete or located near delamination. This may be due to the attenuation of the

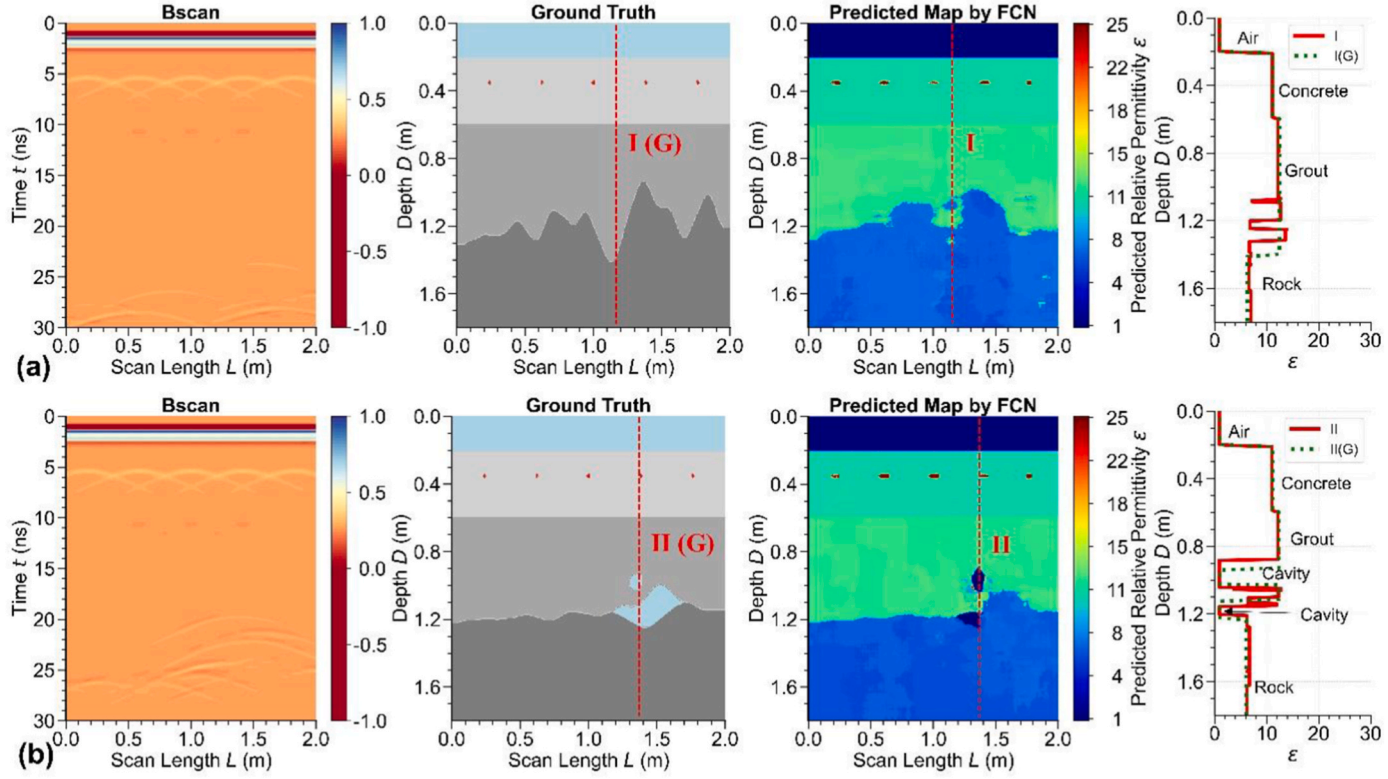

Fig. 16. Predicted relative permittivity map by FCN for tunnel lining defects: (a) Interlayer only, (b) Interlayer with cavity and delamination.

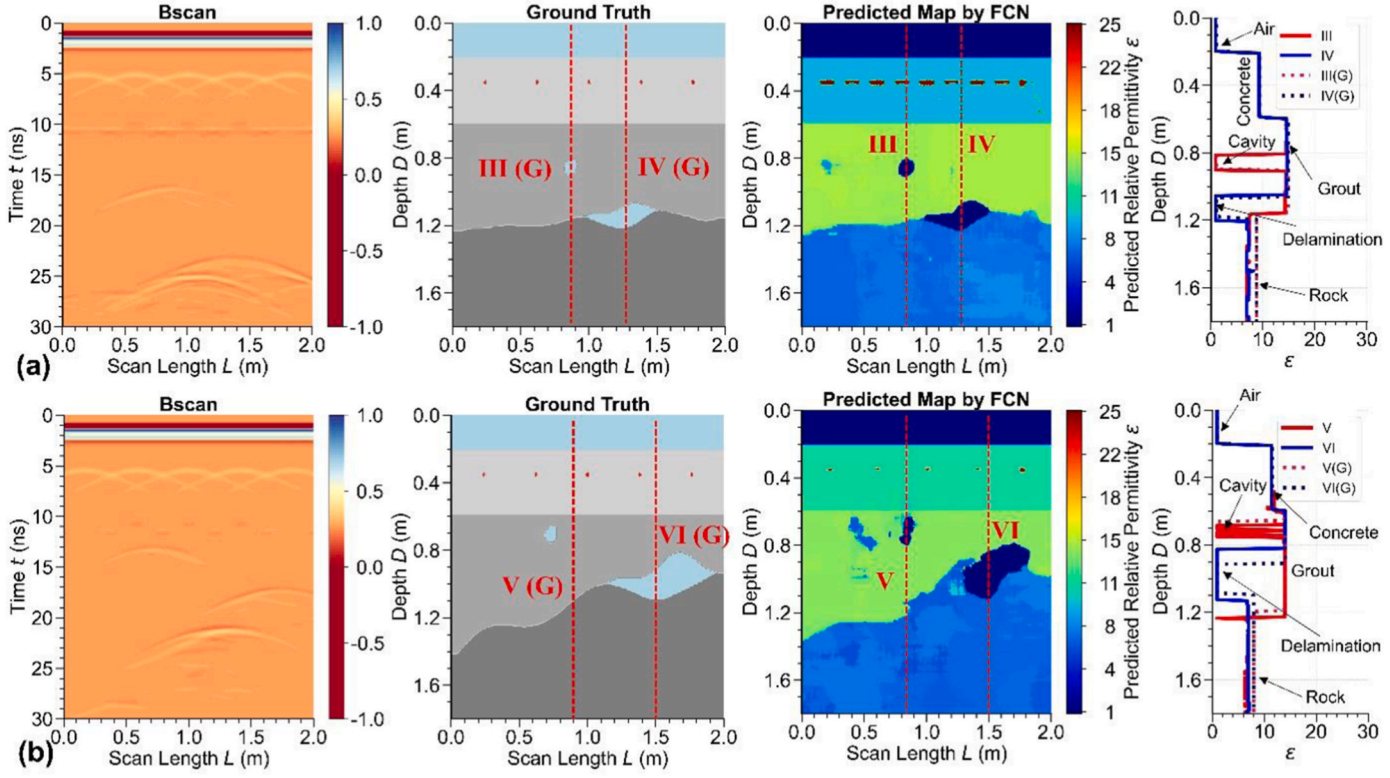

Fig. 17. Predicted relative permittivity map by the FCN model for tunnel lining defects in the case of interlayer with cavity and delamination.

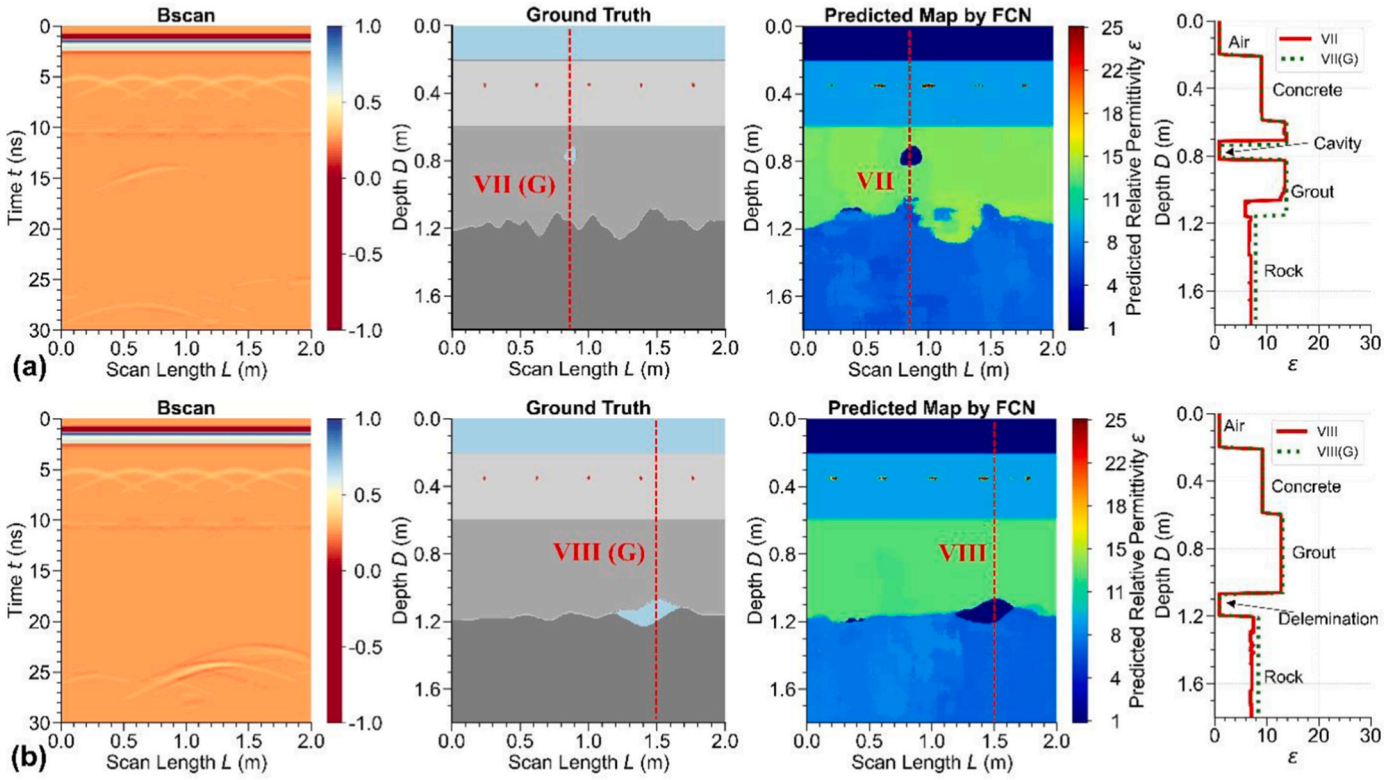

Fig. 18. Predicted relative permittivity map by FCN for tunnel lining defects: (a) Interlayer with cavity, (b) Interlayer with delamination.

dielectric signal that occurs when an electromagnetic wave passes through dense materials such as concrete, which severely reduces the reflection of the cavity or delamination wave near the interlayer between the concrete and the grout. Furthermore, interactions between material layers such as reflection and scattering of the wave at the interface between the concrete and the grout create signal noise, making

Table 6
Statistical analysis of RMSE for FCN-predicted dielectric values vs. Ground Truth.

Sections	RMSE
Line I at the middle of the model (Interlayer only)	1.655
Line II (Interlayer with cavity and delamination, where the cavity is near the delamination)	2.706
Line III at the vertical axis through the cavity (Interlayer with cavity and delamination, where the cavity is at the middle position of the grout)	1.456
Line IV at the vertical axis through the delamination (Interlayer with cavity and delamination, where the cavity is at the middle position of the grout)	1.607
Line V at the vertical axis through the cavity (Interlayer with cavity and delamination, where the cavity is under the concrete layer)	2.196
Line VI at the vertical axis through the delamination (Interlayer with cavity and delamination, where the cavity is under the concrete layer)	2.993
Line VII (interlayer with cavity)	2.373
Line VIII (interlayer with delamination)	0.814

it difficult for the FCN model to distinguish individual features of the cavity or delamination. These factors increase the complexity of the input data, requiring the FCN model to be able to handle noise and analyze more sophisticated signal patterns.

4.6. Case study

The B-scan data collected from a confidential tunnel site in the Gangwon region, South Korea, were used to evaluate the FCN model under real-world conditions. The B-scan data used in this study were acquired by Korea Authority of Land and Infrastructure Safety (KALIS) from the S293 section of a confidential tunnel site in South Korea. In this segment, KALIS used a GPR antenna to determine the delamination location between the concrete layer and the grout layer. Following the B-scan, they manually determined the local delamination. To examine the delamination position, they used the endoscopic image technique through a borehole. As a result, the delamination had a length of 4 m in the horizontal direction of the tunnel cross-section, a width of 2 m in the vertical direction, and a height of the delamination space of 0.3 m (Fig. 19).

B-scan data from field scans often had a lot of noise, so it was processed to remove noise before analysis. Several noise removal methods existed, and the Eigenvalue-based technique was commonly used (Mojahid et al., 2025). The aim of removing background noise was to eliminate unwanted reflections, as GPR radargrams from tunnel scans included signals from the target and other objects. Signals were also affected by direct waves, antenna coupling, and background noise (Mojahid et al., 2025). To extract the target signal accurately, the Eigenvalue technique was applied to reduce these noise effects (Khan & Al-Nuaimy, 2010). Then, a filter was used to enhance the radargram quality by reducing various noise types. The processed B-scan data was fed into the trained FCN model. Before input, the B-scan data was normalized to a 0–255 grayscale range and interpolated to form a 320x320 matrix, suitable for the FCN model (Fig. 14). The FCN model can be predicted a permittivity distribution map, which clearly showed the locations and areas of materials like air, concrete, grout, or rock using a color scale for relative permittivity. This allowed accurate identification of defects such as cavities, delamination, or interlayers. The predicted permittivity map using the FCN model is shown in Fig. 20.

The B-scan data, after removing the background noise using the Eigenvalue technique, and the relative permittivity map are presented in Fig. 20 (a), (b), (c), respectively. The B-scan data show the change in the reflected wave signal between the concrete layer and the grout layer, especially at the position from 7 to 8 m, where the most reflected waves are recorded due to the presence of delamination. The delamination region contains air and steel bars in the reinforced concrete, resulting in many reflected wave patterns, as shown in Fig. 20 (a). After applying the background noise removal technique, the reflected wave frequencies are more clearly separated in the processed data (Fig. 20 (b)). Based on the denoised B-scan data, the FCN model predicts a map of the relative permittivity distribution with a value scale from 1 to 25. By convention, regions with a permittivity greater than 25 correspond to rebar, while regions with values close to 1 represent air, from 6 to 9 are rock, from 9 to 12 are concrete, and from 12 to 15 are grout. These value domains, referenced from Table 3, are used as random initial data to train the FCN model. Based on the reference value scale, the material regions in Fig. 20 are identified: region A is air, region G is grout, and region R is rock. However, the region marked C, representing concrete, has a dielectric constant value from 2 to 6, which is inconsistent with the value range from 9 to 12 in Table 5. This deviation comes from the fact that the outermost concrete layer is directly exposed to air, humidity, exhaust gas, dust, and water vapor in the tunnel. These factors change the dielectric parameters, including relative permittivity and conductivity (mS/m), leading to errors in wave and amplitude characteristics, causing the FCN model to predict the relative permittivity value incorrectly. Furthermore, relying only on relative permittivity for prediction is not enough, which is a limitation of the FCN model. To address this issue, the FCN model needs to be improved to simultaneously predict the distribution maps of dielectric constant and conductivity or combine other parameters to fully cover the characteristics of the material. Future research needs to focus on addressing these challenges to improve the predictive performance of the FCN model.

In this study, the FCN model is optimized to predict relative permittivity maps, which support the identification and localization of characteristic defects such as cavities or delamination. The relative

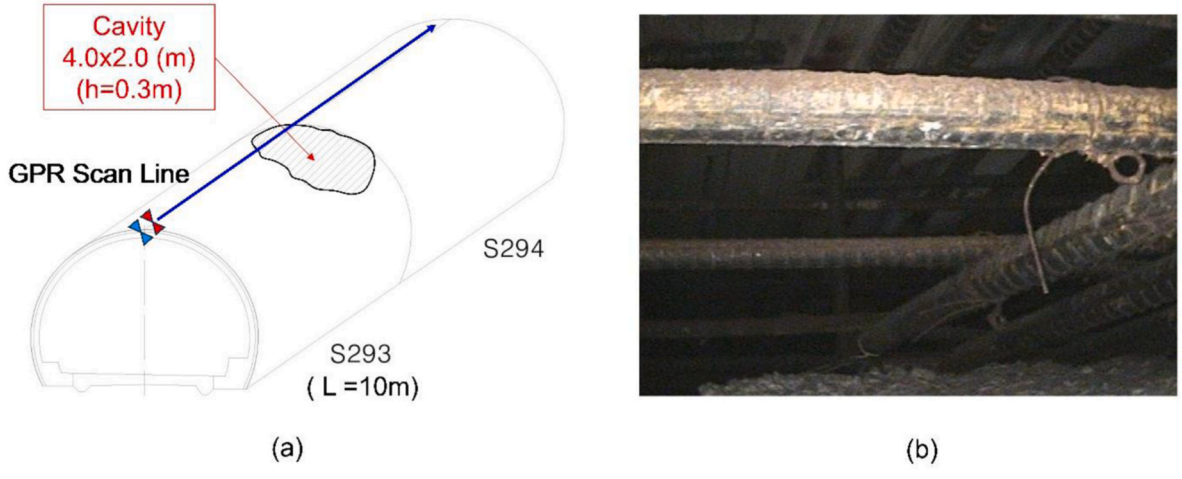

Fig. 19. Illustration of delamination location: (a) Diagram of delamination dimensions; (b) Image obtained from endoscopic imaging technique.

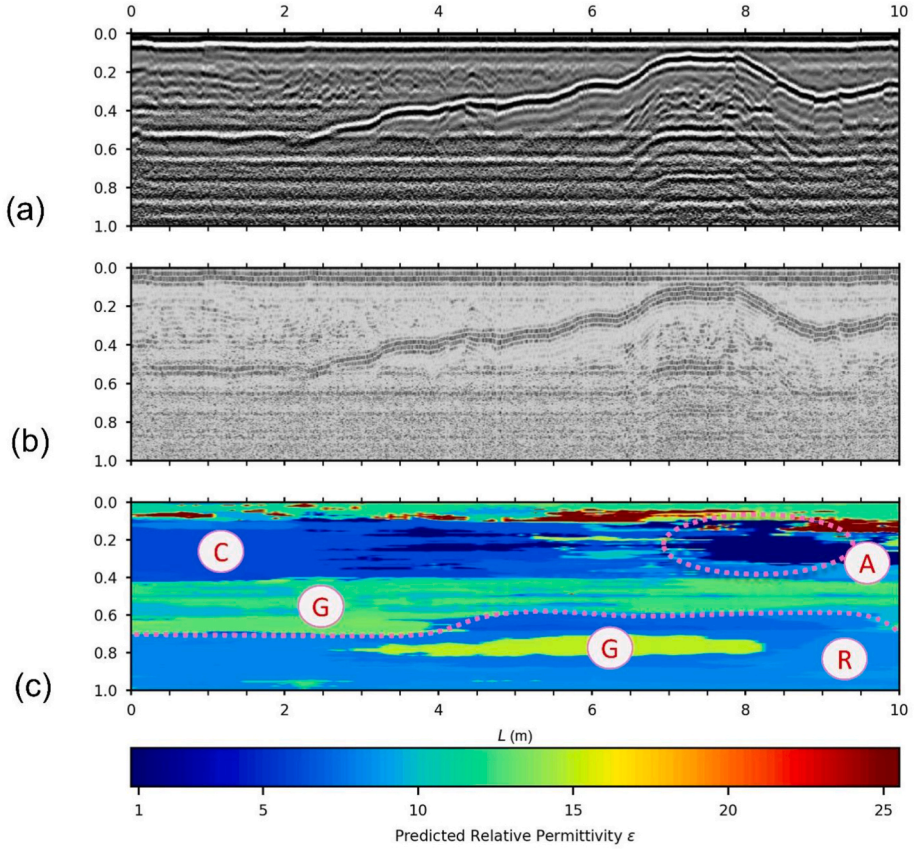

Fig. 20. Evaluate FCN model using actual B-scan: (a) Field B-scan, (b) Filtered B-scan using Eigenvalues, (c) Predicted relative permittivity map by FCN model.

permittivity map also provides information on the thickness of material layers and interlayers between grout and rock layers in a specific field B-scan data. With a 320x320 square matrix input layer, the model is capable of processing B-scan data of any size scale. This opens up the potential application of the FCN model not only in assessing the health of tunnel lining but also in determining the working thickness of concrete layers in slab track, as well as detecting cavities or delamination between the subgrade and slab track concrete. Furthermore, the FCN model can be extended to apply to ballast track structures or to inspect the health of road pavement structures.

However, the predictive results of the FCN model still exhibit errors, particularly concerning concrete structures, when the estimated relative permittivity values fall outside the conventional range established for concrete materials in the initial database. This issue may stem from the fact that the FCN model has only been trained on synthetic data, without incorporating actual B-scan data. To address this limitation, future research should focus on increasing the number of data samples, especially by constructing scale models associated with known defect locations in the laboratory. Additionally, utilizing GPR devices with varying frequencies to collect real data will be beneficial. By combining experimental GPR data with synthetic data, the FCN model can encompass a wider range of real-world scenarios and reduce the effects of signal noise, which significantly impacts the accuracy of relative permittivity maps. These limitations underscore the need for further investigation in future studies.

To address the limitations of the FCN model in this study, integrating robust optimization techniques and fuzzy modeling can minimize the uncertainty in prediction model, particularly for dielectric constant values affected by environmental noise or material variation. Currently, there are many existing researchs, such as the fuzzy regression strategies for gene-environment networks (Kropat et al., 2020) provide tools for handling imprecise input data, potentially refining the training process of mixed-real-genome datasets. Similarly, stochastic control methods during mode transitions (Savku & Weber, 2018) may be feasible in modeling perturbation effects in GPR signals, improving accuracy across different frequencies. These improvements underscore a multidisciplinary path toward more sustainable GPR applications.

5. Conclusion

This study presents a comprehensive approach combining a GPR numerical simulation and FCN to determine the local defects of tunnel lining based on GPR data.

The main findings are summarized as follows.

1. A GPR simulation software, KIT-GPR, has been successfully developed in C# to facilitate flexible and structured modeling of tunnel lining. This software enables users to effortlessly select antenna types, geological features, and defect shapes such as interlayer, cavities and delamination.
2. Four defect scenarios—namely interlayer, interlayer with cavity, interlayer with delamination, and a combination of all three—were simulated using KIT-GPR program and compared with the results obtained from the gprMax software. The high degree of similarity in the A-scan results validates the accuracy of the developed tool and highlights its effectiveness in generating training data for the FCN model.
3. The FCN model shows high accuracy in cases where the defect is single or clearly isolated, with low RMSE errors and the relative permittivity map that closely reference B-scan (Ground truth). However, in complex situations—for example, when cavities and delamination regions overlap or are located near concrete boundaries—the model performance deteriorates significantly. This is mainly due to the attenuation of the reflected signal by material

interactions, resulting in higher RMSE errors and the appearance of image noise.

4. The FCN model predicts the relative permittivity distribution quite well with B-scan input data from a real tunnel site in Korea. Based on the relative permittivity prediction map from the FCN model, the locations of cavity, delamination, grout and rock layers can be easily located. However, the relative permittivity prediction results of the concrete layer are not correct. This may be due to the change in dielectric constants of the concrete layer, so the wave characteristics change compared to the synthetic models.

Although there are several drawbacks, the FCN model offers several advantages over conventional methods. It is worth noting that the FCN model is ability to predict defect locations at the pixel level and capture the spatial variation of dielectric properties. It is particularly effective in identifying the location and presence of cavities and delamination regions—important factors for maintenance and early detection of structural damage underground.

In summary, the FCN model demonstrates great potential for automating the process of determining defects in tunnel lining based on B-scan data. Future works could enhance the FCN model by integrating hybrid models that incorporate physics knowledge or the use of multi-scale attention mechanisms to improve accuracy in difficult scenarios. Moreover, the combination of numerical simulation and deep learning proposed in this study has laid a foundation for the development of a data-driven and automated tunnel inspection system.

CRediT authorship contribution statement

Viet Dinh Le: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Gyu-Hyun Go:** Writing – review & editing, Supervision, Funding acquisition, Formal analysis, Conceptualization. **Sayali Pangavhane:** Writing – review & editing, Validation, Software. **Chang Kyoon Yoo:** Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1C1C1006507).

Appendices.

A.1. KIT-GPR Program

The GPR simulation software KIT-GPR was designed with a strong focus on user experience (as shown in Fig. A.1). KIT-GPR was developed based on the C-Sharp language and consists of five main modules: material declaration, geometry construction, antenna parameter declaration, analysis, and result export. The material declaration module allows users to define the material parameters or select characteristic materials from the built-in library. For each material definitions for display colors and specific parameters such as electrical conductivity (σ), dielectric constant (ε), and magnetic permeability (μ) are included. In the geometry construction module, users can define specific regions in rectangular, circular, or polygonal shapes using a simple user-friendly declaration menu. For each declared region, users must assign the corresponding material, and the color of the region in the model must match the predefined color in the material declaration module. After constructing the model, users define the antenna and other parameters, including the type of simulation wave form, calculation time, time step, mesh size, scan step, and length of the scanning line in the antenna parameter declaration module. The KIT-GPR program integrates several antenna types for user reference, with the primary parameters cited from the open-source library gprMax (Warren et al., 2016), as the MATLAB source code by Irving and Knight (2006) employs only the Harris wave type. In addition, KIT-GPR provides suggestions for the time step and mesh element size, aiding users in a quick selection to minimize errors during subsequent calculations.

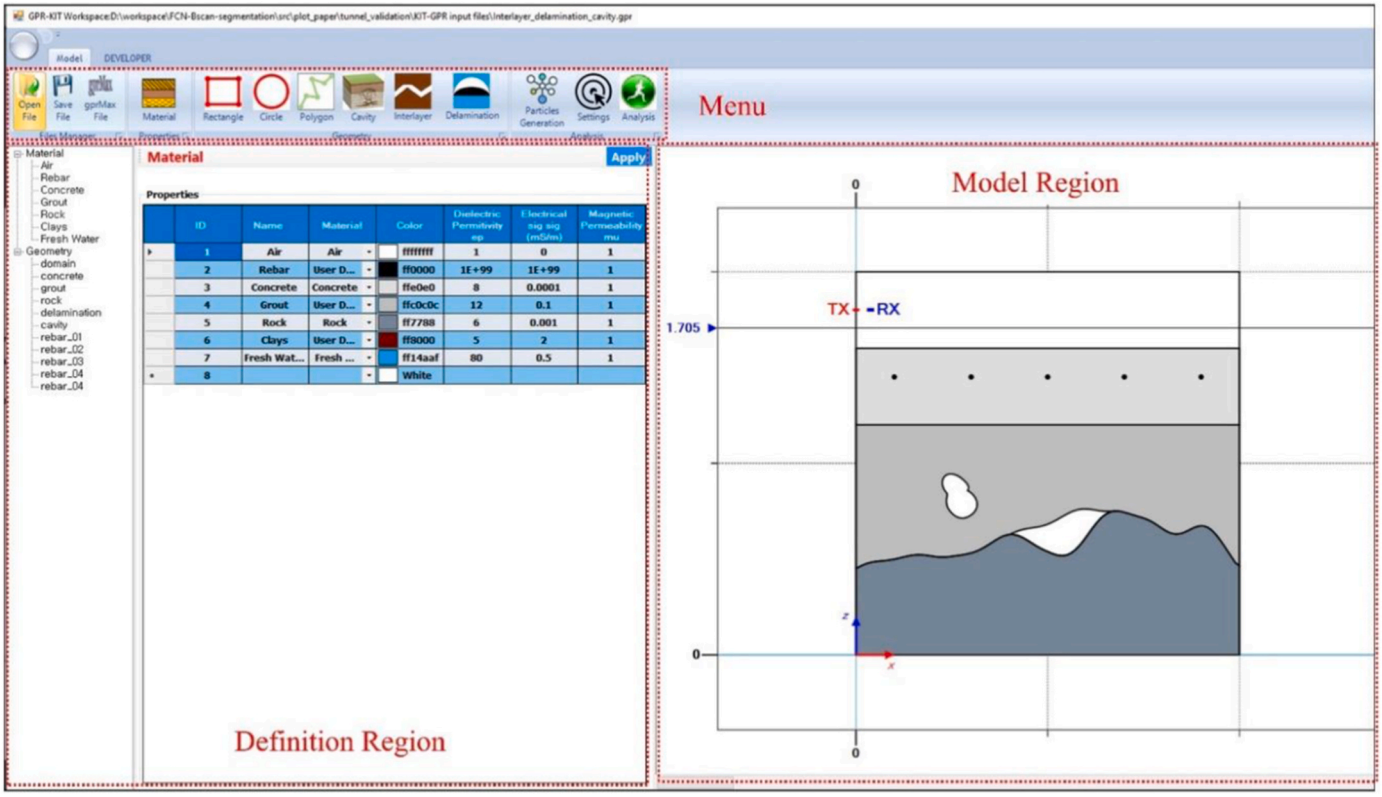

Fig. A.1. Graphic User Interface of KIT-GPR program.

Upon completing the declarations in the above modules, users set output parameters, such as the data storage directory, snapshots of the E_y electromagnetic field over time, and other parameters. KIT-GPR performs automatic error checking before execution and notifies users of any detected errors for review purposes. During the calculation process, KIT-GPR displays real-time A-scan results at the current calculated position of the scanning line and the dynamic motion of E_y electromagnetic field waves, allowing users to observe wave trends and identify potential anomalies. After completing the calculation process, users can export data such as A-scans at various positions, B-scans, and snapshots of the electromagnetic field as image and CSV data format files.

In addition to the basic features of a GPR wave simulation program, KIT-GPR supports the construction of special objects, such as irregular surfaces between soil layers, called interlayers, cavities, or delamination on the surface between soil layers, using the Perlin noise algorithm.

A.2. Application Perlin Noise for Creating Interlayer, Cavity, and Delamination Patterns

In the KIT-GPR program, functions for creating the interlayer, cavity, and delamination are designed to facilitate the initialization of these objects for users. The interlayers between the two material layers are constructed using the Perlin noise 1D algorithm. Users must define the parameters, including the position of the interlayer line and the parameters of the Perlin noise algorithm. In addition, users can plot graphs representing the Perlin noise curve. Fig. A.2 illustrates the interface window forms used to define the interlayers in the KIT-GPR program.

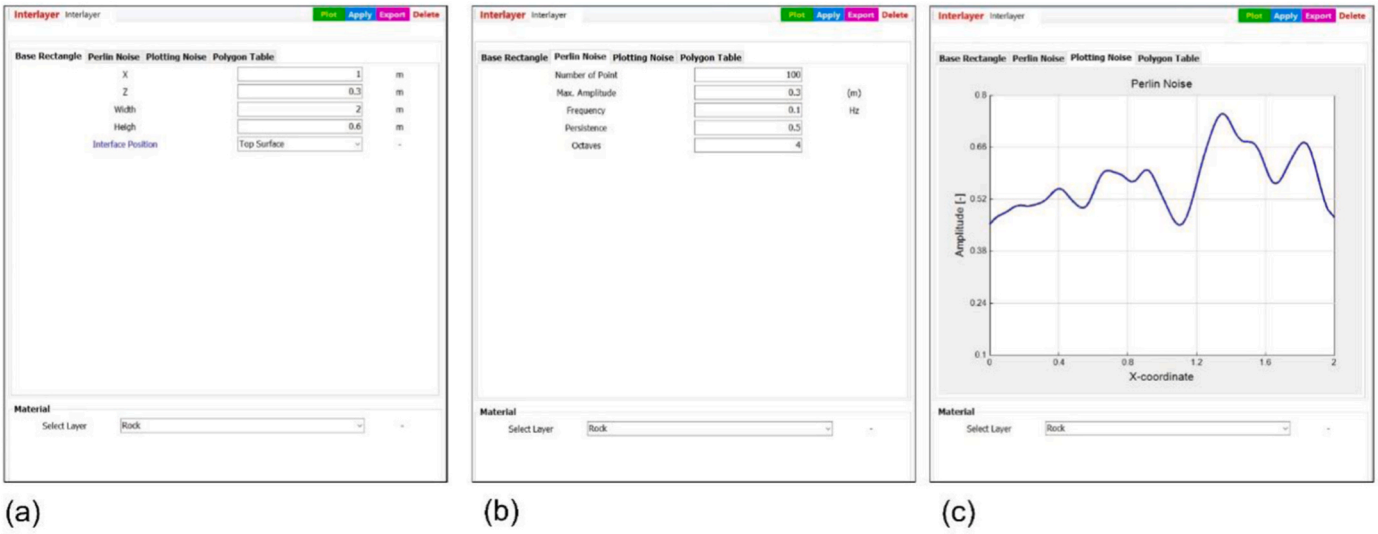

Fig. A.2. Interlayer Definition: (a) Position definition, (b) Perlin noise parameter, and (c) Perlin noise curve.

Users must define a basic rectangle in the model and specify whether the Perlin noise curve should be defined at the top or bottom edges of the rectangle. Additionally, users must define the material type of the layer based on predefined material parameters. For the basic rectangle, users must specify its center, width, and length. Subsequently, users must select the material type through a combination box linked to the predefined material parameters. Once the basic rectangle is defined, users must set the parameters of the Perlin noise algorithm, including the number of points on the curve, amplitude, frequency, persistence, and octave. Then, users can press the "*Plot*" button to preview the Perlin curve. Users can adjust the Perlin noise algorithm parameters to obtain desired curves. After completing the setup, users press the "*Apply*" button to create a geometrical region in the model with the interlayer line. A delamination object is often defined based on the interlayer line because delamination typically occurs between two material layers. Therefore, users must define an interlayer line before defining the Delamination object. After defining the interlayer lines, users select the desired interlayer line and then declare two points, typically corner vertices, on the interlayer line. The program connects these points with a Perlin noise curve, which typically has default first-order parameters: octave = 1, amplitude = 0.05 m, frequency = 0.1, and persistence = 0.2. For the Cavity object, users can easily define any coordinates and then adjust the default parameters of the Perlin noise curve to create a Perlin noise curve around the center coordinates of the cavity, as defined by the user.

Data availability

Data will be made available on request.

References

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Ciresan, D., Giusti, A., Gambardella, L., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. Adv. Neural Inf. Process. Syst. 25, 2852–2860.

Dawood, T., Zhu, Z., Zayed, T., 2020. Deterioration mapping in subway infrastructure using sensory data of GPR. Tunn. Undergr. Space Technol. 103, 103487. https://doi.org/10.1016/j.tust.2020.103487.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T., 2013. A deep convolutional activation feature for generic visual recognition. UC Berkeley and ICSI, Berkeley, CA, USA 1 (2). https://doi.org/10.48550/arxiv.1310.1531.

Eigen, D., Krishnan, D., Fergus, R., 2013. Restoring an image taken through a window covered with dirt or rain. Proceedings of the IEEE International Conference on Computer Vision 633–640. https://doi.org/10.1109/ICCV.2013.84.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vision 88 (2), 303–338.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2012. Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1915–1929. https://doi.org/10.1109/TPAMI.2012.231.

Gedney, S.D., 2011. The perfectly matched layer (PML) absorbing medium. In: Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics. Springer International Publishing, Cham, pp. 113–135. https://doi.org/10.1007/978-3-031-01712-4_6.

Georgakopoulos, S.V., Birtcher, C.R., Balanis, C.A., Renaut, R.A., 2002. Higher-order finite-difference schemes for electromagnetic radiation, scattering, and penetration. IEEE Antenn. Propag. Mag. 44 (1), 134–142. https://doi.org/10.1109/74.997945.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Paper presented at the 580–587. https://doi.org/10.1109/CVPR.2014.81. Retrieved from. https://ieeexplore.ieee.org/document/6909475.

Haack, A., Schreyer, J., Jackel, G., 1995. Report to ITA working group on maintenance and repair of underground structures: State-of-the-art of non-destructive testing methods for determining the state of a tunnel lining. Tunn. Undergr. Space Technol. 10 (4), 413–431. https://doi.org/10.1016/0886-7798(95)00030-3.

Huang, J., Yang, X., Zhou, F., Li, X., Zhou, B., Lu, S., Slob, E., 2024. A deep learning framework based on improved self-supervised learning for ground-penetrating radar tunnel lining inspection. Comput. Aided Civ. Infrastruct. Eng. 39 (6), 814–833. https://doi.org/10.1111/mice.13042.

Irving, J., Knight, R., 2006. Numerical modeling of ground-penetrating radar in 2-D using MATLAB. Comput. Geosci. 32 (9), 1247–1258. https://doi.org/10.1016/j.cageo.2005.11.006.

Khan, U.S., Al-Nuaimy, W., 2010. Background removal from GPR data using eigenvalues. Paper presented at the, 1–5. https://ieeexplore.ieee.org/document/5550079.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25. https://doi.org/10.1145/3065386.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. https://doi.org/10.1145/3065386.

Kropat, E., Weber, G.-W., Tirkolaee, E., 2020. Foundations of semialgebraic gene-environment networks. J. Dynam. Game 1–17. https://doi.org/10.3934/jdg.2020018.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. https://doi.org/10.1038/nature14539.

Li, S., Liu, B., Xu, X., Nie, L., Liu, Z., Song, J., Fan, K., 2017. An overview of ahead geological prospecting in tunneling. Tunn. Undergr. Space Technol. 63, 69–94. https://doi.org/10.1016/j.tust.2016.12.011.

Liu, B., Ren, Y., Liu, H., Xu, H., Wang, Z., Cohn, A.G., Jiang, P., 2021. GPRInvNet: deep learning-based ground-penetrating radar data inversion for tunnel linings. IEEE Trans. Geosci. Rem. Sens. 59 (10), 8305–8325. https://doi.org/10.1109/TGRS.2020.3046454.

Liu, B., Zhang, J., Lei, M., Yang, S., Wang, Z., 2023. Simultaneous tunnel defects and lining thickness identification based on multi-tasks deep neural network from ground penetrating radar images. Autom. ConStruct. 145, 104633. https://doi.org/10.1016/j.autcon.2022.104633.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.

Matan, O., Burges, C.J., LeCun, Y., Denker, J., 1991. Multi-digit recognition using a space displacement neural network. Adv. Neural Inf. Process. Syst. 4, 488–495.

Mojahid, A., Ouai, D.E., Amraoui, K.E., El-Hami, K., Aitbenamer, H., 2025. Intelligent recognition of subsurface utilities and voids: a ground penetrating radar dataset for deep learning applications. Data Brief 59, 111338. https://doi.org/10.1016/j.dib.2025.111338.

Montero, R., Victores, J.G., Martinez, S., Jardón, A., Balaguer, C., 2015. Past, present and future of robotic tunnel inspection. Autom. ConStruct. 59, 99–112. https://doi.org/10.1016/j.autcon.2015.02.003.

Perlin, K., 2002a. Improved noise reference implementation. Retrieved from. https://mrl.nyu.edu/~perlin/noise/.

Mount, D., Eastman, R., 2018. Procedural generation: 2D Perlin noise. CMSC 425. Lecture 13.

Perlin, K., 2002b. Improving noise. Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, pp. 681–682. https://doi.org/10.1145/566654.566636.

Pinheiro, P., Collobert, R., 2014. Recurrent convolutional neural networks for scene labeling. International Conference on Machine Learning, pp. 82–90.

Qin, H., Zhang, D., Tang, Y., Wang, Y., 2021. Automatic recognition of tunnel lining elements from GPR images using deep convolutional networks with data augmentation. Autom. ConStruct. 130, 103830. https://doi.org/10.1016/j.autcon.2021.103830.

Roden, J.A., Gedney, S.D., 2000. Convolution PML (CPML): an efficient FDTD implementation of the CFS-PML for arbitrary media. Microw. Opt. Technol. Lett. 27 (5), 334–339. https://doi.org/10.1002/1098-2760(20001205)27:5.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. Lect. Notes Comput. Sci. 9351, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

Savku, E., Weber, G.W., 2018. A stochastic maximum principle for a markov regime-switching jump-diffusion model with delay and an application to finance. J. Optim. Theor. Appl. 179, 696–721. https://doi.org/10.1007/s10957-017-1159-3.

Silberman, N., Hoiem, D., Kohli, P., Fergus, R., 2012. Indoor segmentation and support inference from RGBD images. Computer Vision–ECCV 2012, 746–760. https://doi.org/10.1007/978-3-642-33715-4_54.

Simi, A., Manacorda, G., 2016. The NeTTUN project: design of a GPR antenna for a TBM. 2016 16th International Conference on Ground Penetrating Radar (GPR), pp. 1–6. https://doi.org/10.1109/ICGPR.2016.7572648.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/ARXIV.1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A., 2015. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

Veal, C., Dowdy, J., Brockner, B., Anderson, D.T., Ball, J.E., Scott, G., 2018. Generative adversarial networks for ground-penetrating radar in handheld explosive hazard detection. Detection and Sensing of Mines, Explosive Objects, and. Obscured Targets XXIII (10628), 306–323. https://doi.org/10.1117/12.2307261.

Wang, J., Cetinkaya, H., Yarovoy, A., Vermesan, I.I., Reynaud, S., 2015. Investigation of forward-looking synthetic circular array for subsurface imaging in tunnel boring machine applications. 2015 8th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), pp. 1–4. https://doi.org/10.1109/IWAGPR.2015.7292673.

Warren, C., Giannopoulos, A., 2011. Creating finite-difference time-domain models of commercial ground-penetrating radar antennas using Taguchi's optimization method. Geophysics 76 (2), G37–G47. https://doi.org/10.1190/1.3548506.

Warren, C., Giannopoulos, A., Giannakis, I., 2016. gprMax: open source software to simulate electromagnetic wave propagation for ground-penetrating radar. Comput. Phys. Commun. 209, 163–170. https://doi.org/10.1016/j.cpc.2016.08.020.

Wolf, R., Platt, J., 1993. Postal address block location using a convolutional locator network. Adv. Neural Inf. Process. Syst. 6, 745–752.

Yue, Y., Liu, H., Meng, X., Li, Y., Du, Y., 2021. Generation of high-precision ground-penetrating radar images using improved least square generative adversarial networks. Remote Sens. 13 (22), 4590. https://doi.org/10.3390/rs13224590.

Zhang, F., Liu, B., Liu, L., Wang, J., Lin, C., Yang, L., Yang, W., 2019. Application of ground-penetrating radar to detect tunnel lining defects based on improved full waveform inversion and reverse time migration. Near Surf. Geophys. 17 (2), 127–139. https://doi.org/10.1002/nsg.12032.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer International Publishing, Cham, pp. 818–833.

Zhang, N., Donahue, J., Girshick, R., Darrell, T., 2014. Part-based R-CNNs for fine-grained category detection. Computer Vision–ECCV 2014, 834–849. https://doi.org/10.1007/978-3-319-10590-1_54.

Zhang, S., Zhang, L., He, W., Ling, T., Deng, Z., Fu, G., 2022. Three-dimensional quantitative recognition of filler materials ahead of a tunnel face via time-energy density analysis of wavelet transforms. Minerals 12 (2), 234. https://doi.org/10.3390/min120.

Zhou, X., Chen, H., Li, J., 2018. An automatic GPR B-scan image interpreting model. IEEE Transact. Geosci. Remot. Sens. 56 (6), 3398–3412.

Zhu, H., Huang, M., Zhang, Q., 2024. TunGPR: enhancing data-driven maintenance for tunnel linings through synthetic datasets, deep learning and BIM. Tunn. Undergr. Space Technol. 145, 105568. https://doi.org/10.1016/j.tust.2023.105568.