2023

The use of deep learning to solve the inverse problem of ground penetrating radar and its application to potato farming.

Charles Lees

A thesis submitted in partial fulfilment of the requirements of the University of the West of England, Bristol for the degree of Doctor of Philosophy, in Deep Learning.

This research programme was carried out in collaboration with B-Hive.

College of Arts, Technology and Environment

May 2024

Dedicated to my wife Shira, for her continued patience and support.

Abstract

This thesis develops a novel algorithm to solving the inverse problem for Ground Penetrating Radar when used to detect potatoes during the growing phase. It does so by reviewing current analytical approaches to solving the inverse problem. It then explains the creation of a 2D image creation method with the aim of demonstrating how deep learning can solve the inverse problem. It then uses a similar methodology to map between a single radar scan and usable 3D images with the intention to demonstrate that the technology is viable in both applications, and that by using a single scan it is possible to reduce the investment and computational costs required for the hardware and overall system solution. The 2D results demonstrate that it is possible to utilise the 2D mapping approach to create images based on a single radar scan. The 3D results, while not perfect, demonstrate that there is a potential in the method contained within this thesis. Overall, the results show that there are severe limitations which still need to be overcome in the current commercial scanning hardware available and in the computational resource. Some solutions to these limitations have been suggested and should be incorporated in future developments of the system.

1 Introduction

The agricultural sector is under constant economic strain due to market pressures to keep food plentiful and cheap in the context of an increasing global population, environmental awareness and the uncertainties of climate change. The introduction of technology into the farming industry, such as will be investigated within this project, has the potential to address these challenges, help improve crop yield forecasts, and get crops to the marketplace at the optimum time. Other secondary benefits include improving crop health monitoring.

This research builds on prior field work already performed on 3D tomography of subsurface objects. Its principal aim is to develop a system for the visualisation of plant tubers and root stock in 3D during the growing phase to monitor plant health and make informed interventions prior to the harvesting phase to maximise yield. Imaging will be coupled with state-of-the-art deep learning algorithms to process the signal, using image analysis to identify shape, size and condition of the root stock and tubers in real time. In this manner, the principal technology will be using deep learning algorithms to achieve the project's aim to produce results in the field and in as close to real time as possible.

A wide range of subsurface imaging techniques, for example Electro-Resistance Tomography, Seismic Attenuation Profile, and Ground Penetrating Radar are currently employed across a variety of sectors such as archaeology, forensic science, civil engineering, geology, and unexploded ordinance detection. Therefore, it seems reasonable to also expand the application of these technologies into the agriculture sector, where only limited progress was made to date.

The major problem with subsurface imaging techniques is that the results are highly influenced by the content of the soil, its structure, organic material, and different levels of moisture. These factors affect the strength of the response and the speed at which the signal propagates through the soil, leading to confusing results and false positives. Other complicating factors include the amount of foliage limiting access to the ground, and reluctance by the farmers to unnecessarily disturb the fields during the growing phase.

These factors can result in erroneous distance measurements. Machine learning will be employed to reduce the shortfalls in the Ground Penetrating Radar (GPR) hardware. GPR utilises electromagnetic waves to penetrate the surface. These waves then reflect off objects to a receiver. GPR traditionally creates a 2D slice, which shows subsurface objects in the form of a hyperbolic curve.

The inverse problem is where a known output exists, and the aim is to find what caused the output. Over the years, many empirical solutions to the inverse problem were proposed, some of which are discussed in section 2.2, however they mainly focus on a 2D results. The purpose of this thesis is to create an alternative solution to the inverse problem using deep learning to map between two solution spaces.

This thesis will take the reader through an attempt to create an empirical solution to the inverse problem in Section 3 and highlight the problems associated with this solution. It will then give a review of GPR and the use of numerical simulation software to create both 2D and 3D potato simulations to create a dataset for the deep learning training which is discussed in detail in chapter 4. The data set will be created using a commercially available antenna. It will explain the creation of a 2D-to-2D solution to the inverse problem using deep learning in section 5.3 and finally

detail a 2D to 3D solution in section 5.4 to realise a proposed system capable of accurately imaging tubers in the ground.

The main contributions that this work provides are as follows:

- Develop a data driven approach to the inverse problem.
- An investigation into the use of deep learning to speed up the solution of the inverse problem.
- Investigation into the possibility of creating 3D images from a single GPR scan, and to show that accurate 3D representations of sub-surface tubers can be produced from GPR data.

1.1 Problem Statement and Method

This section is split into three: aims, objectives and the approach taken. The principal problem that this work attempts to tackle is that it is difficult to see what is happening subsurface without disturbing the ground, and disturbing the ground changes the measurements, not to mention can harm crops.

A non-intrusive technology that presents itself is GPR, but mapping the scan results to the actual locations and sizes of object that caused the observed scan is not straightforward. This is a classical inverse problem, and this research will explore the implementation of an analytical solution to the inverse problem as well as a data driven approach. It will also highlight that the need for high computation times and inability to transform between the 2D and 3D spaces are significant drawbacks of the analytical solution.

1.1.1 Aims

The aim of this research is to develop a system for the visualisation and data extraction from plant tubers and root stock in 3D during the growing phase of potatoes or other subsurface organic produce to monitor crop health prior to the harvesting phase, which will allow growers to take appropriate actions to maximise yield. This will be coupled with deep learning algorithms to process the signal, using image analysis to identify the shape, size and overall condition of the root stock and tubers in real time.

1.1.2 Objectives

The objectives of this research are:

- Identify and size individual tubers using existing geophysics techniques.
- Demonstrate that deep learning can significantly improve the accuracy of quantification (count, size, and location) of tubers whilst below ground.

1.1.3 Approach

The approach adopted in this work is to use simulation software to generate a representative GPR scan dataset for both 2D and 3D geometries, which represent known subsurface organic objects. These simulations will be compared to a set of GPR scans and an attempt to correlate between the real world GPR scans and the predicted scans will be undertaken.

The 2D dataset contains 15,998 samples and the 3D dataset contains 16,000 samples. The pilot studies undertaken as part of this PhD showed that there was adequate convergence after 15,000 samples with the 2D dataset.

The deep learning networks are all bespoke for this research project and are developed using Python and the Pytorch framework.

2 Literature Review

*"*Artificial Neural Networks and Machine Learning techniques applied to Ground Penetrating Radar: A review*"* by (Travassos, Avila and Ida, 2018) performs a review of published articles which combine GPR and machine learning. The conclusion of the review is that machine learning has the potential to process GPR signals for object identification. However, two challenges are associated with the use of machine learning in this field. The first is reducing false positives and negatives. The second is improving the object localisation and handling of multiple objects. The paper also notes that the problem is becoming more complicated by the requirement to produce 3D volumes and the additional complexity of weather, vegetation, and terrain. It is also clear from the literature review that very little work has been done using deep learning to solve the inverse problem.

Following on from the work undertaken by (Travassos, Avila and Ida, 2018), this research builds upon peer reviewed literature in the fields of machine learning and signal processing detailed in sections 2.1 to 2.1.3.

2.1 Deep Learning Applied to GPR Signals

2.1.1 Laplace Neural Network Proposal for Processing GPR Signals (Szymczyk and Szymczyk, 2015a).

The paper proposes a method of implementing artificial neural networks to process GPR signals. The approach put forward in the paper is adopted because GPR traces have been historically difficult to interpret automatically due to the large number of variables which are outside of the user's control. The paper outlines a proposal to use a neural network architecture based on Laplace transforms to process GPR signal traces. The proposed architecture uses the dynamic nature of the waveforms to solve a given problem. The proposed architecture for the Laplace Neural Network (LNN) is shown in Figure 2-1.

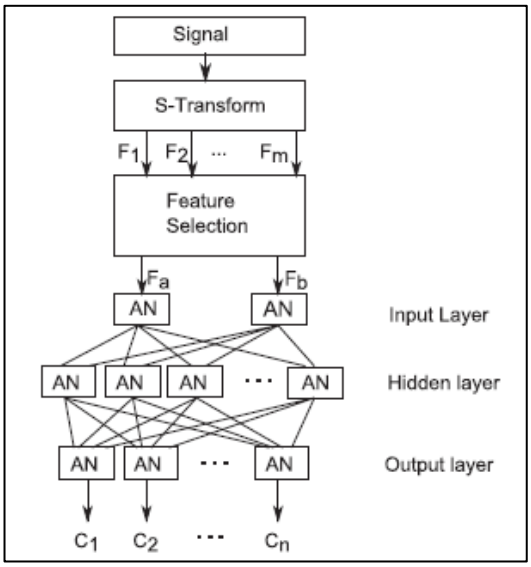


Figure 2-1 Laplace Neural Network Architecture (Szymczyk and Szymczyk, 2015a)

As with all neural networks, there is a requirement to include a learning algorithm which minimises the error by adjusting the weights on each node in the fully connected layer. As seen in Figure 2-2, in this architecture this is done by using the output signal $Y(s)$ in response to input signals $X(s)$. The learning algorithm

proposed is based on the Widrow-Hoff proposal which leads on to the back-propagation algorithm.

The fully connected layer is replaced with the Laplace artificial neuron, while the weights and linear activation functions are replaced with transmittances. In this case the outputs are differences between a known signal and the supplied signal, and these differences can be used as classification.

A subsequent paper (Szymczyk and Szymczyk, 2015b) was published by the same authors which shows how the Laplace Neural Network has been used with GPR signals. In the second paper (Szymczyk and Szymczyk, 2015b) the learning method proposed is with a known input signal $Z(s)$, which is compared to the output signal that creates the error. The basic learning method is shown in Figure 2-2. It aims to minimise the difference between Y(s) and Z(s). In their subsequent paper, the learning algorithm was expanded as shown in Figure 2-3, there the feedback is identified in red.

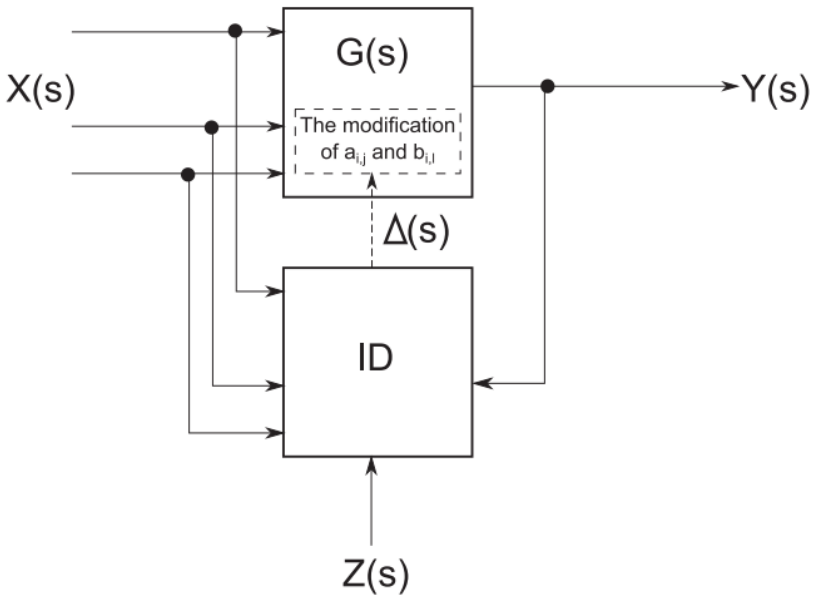


Figure 2-2 Supervised Learning Method (Szymczyk and Szymczyk, 2015a)

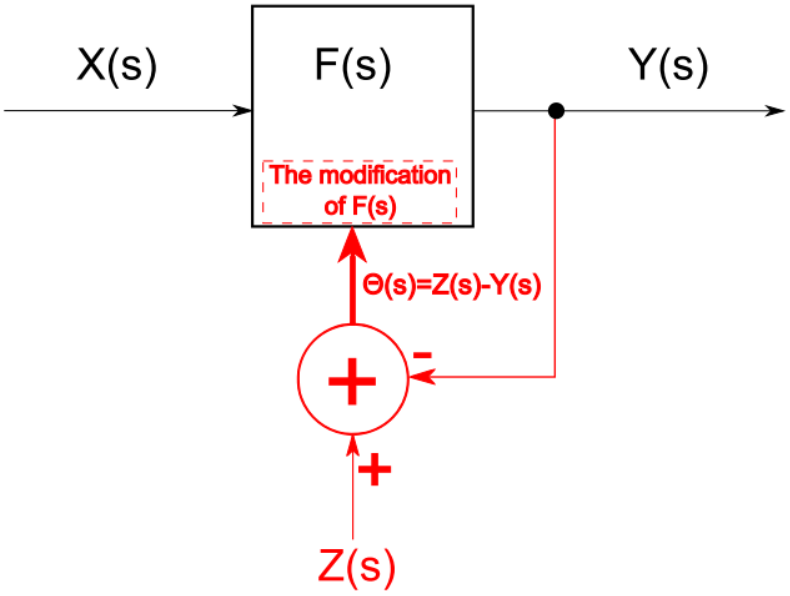


Figure 2-3 Revised and Updated Learning Method (Szymczyk and Szymczyk, 2015b)

In the approach used in both papers the transmitted signal is modelled as a Laplace transform of sine wave and the Heaviside function whilst the received signal is modelled as a Laplace transform of a large order polynomial. From this the transfer function can be calculated.

$$Transfer\ Function = \frac{Recieved\ Signal}{Transmitted\ Signal}$$

The transmittance variables F(s) are calculated for known signals (a sinkhole, loose zone, and no object), these are then compared to the received signal and from this a classification can be deduced.

The paper does not extensively detail how successful this architecture is, and a review of citations does not reveal any works which compared the results from this paper to alternative architectures. This therefore leads to some scepticism about the success in real world applications. However, using the ideas in this paper, it may be possible to use the wavelet domain to create a filter bank such that the filter constants are optimised to extract the most information from the signal, as well as

compressing the signal which would help to maintain the efficiency of the system. It is possible to then use alternative deep learning techniques to classify the signal and generate a point cloud, from which a useful image can be generated.

In Summary, the paper proposes a novel approach for processing the GPR signal using Neural Networks.

2.1.2 Detection and Classification of Landmines Using GPR and R-CNN's (Kafedziski, Pecov and Tanevski, 2018)

The main theme of this paper is the use a faster Region-Convolutional Neural Network (R-CNN) based on the Inception-v2 architecture which delivered accurate results. The input data was a mixture of real-world scans and simulated results. The authors used the numerical Ground Penetrating Radar simulation tool GPRMax to simulate 48 B-scans of various anti-personnel and anti-tank mines that were combined with 109 real world scans of mines, which resulted in a total of 157 B-scans, containing 351 objects. The test data was split into 75 random object hyperbolic images, 24 anti-tank mine signatures and 25 signatures of empty space.

It is difficult to comment on the approach used without fully understanding how much variation there is in the B-Scan simulations. The obvious comment is that the dataset is rather small, which would impact on the overall generalisability of the approach.

The paper does not discuss how well the GPRMax models correlate to the real-world scan. If this was explored in more detail, the approach would be easier to comment on as well as giving more confidence about the validity of this research. However, the approach itself is sound, and a mixture of simulated data and real-world data will allow the generation of a dataset efficiently and, in the case of unexploded ordnance (UXO), safely.

As a result of this article, in Appendix A, it has been demonstrated that it is possible to count the number of items using pretrained deep learning networks with a limited dataset with some success, albeit with a significantly larger dataset.

Figure 2-4 is the confusion matrix for the two classes: objects and Anti-Tank mines, where background is added to create false positives and negatives (FP & FN).

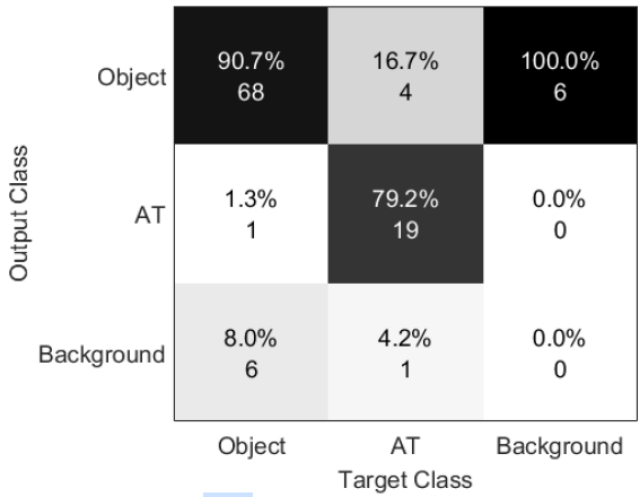


Figure 2-4 Confusion Matrix for the Classification Model (Kafedziski, Pecov and Tanevski, 2018)

The results in Figure 2-5 are further refined by summing various components of Figure 2-4, for example the true positive (TP) values are obtained using the sum of the values in the four cells (it does not say which four cells). The FP entry is obtained by summing the first two entries in the last column of Figure 2-4, and the FN value is the sum of the first two entries in the last row. The TP entry in Figure 2-5 is overstated, since there are five miss-classification (FN) and as a result the true value for FN is 85% and TP (correct object classed as correct object is 68% and AT classed as AT is 19%) should be 87.88%.

TP 92.9% 92	**FP** 100% 6
FN 7.1% 7	**TN** 0.0% 0

Figure 2-5 Detection Confusion Matrix (Kafedziski, Pecov and Tanevski, 2018)

This was then used to a retrain a pretrained Regional CNN. The results are not overly clear as to the success level, however when dealing with highly dangerous objects such as UXO a 92.9% (or 87.88%) does seem low. More information on the application of this approach to organic material is in Appendix A.

2.1.3 GPRInvNet: Deep Learning-Based Ground-Penetrating Radar Data Inversion for Tunnel Linings (Liu *et al.*, 2021a)

This is a highly relevant paper because the methodology is similar to the one employed in this thesis. The main differences are firstly that the authors utilise a MATLAB simulation to create a large 2D dataset (more than 400,000 files), with an architecture based on an encoder and decoder deep learning model. The second key difference is that they are not trying to recreate an image of the subsurface feature, but instead calculate the permittivity (the ability of a substance to store

energy in the electrical field) map. The model is run using a batch size of 12 with a learning rate of $5e^{-5}$ and run for 100 epochs.

The loss function used is a combination of the L2 (Mean Squared Error – MSE) loss and a multiscale structural similarity (MS-SIM) loss. The main benefit of the MS-SIM loss function is that it allows for a method of predicting the perceived quality of an image by creating parameters which define the relative importance of different scales and was originally proposed in (Wang, Simoncelli and Bovik, 2003)

The results in this paper are very encouraging when compared to actual measured data, even though no absolute error margin is shown and no statistical comparison of the results against actual data carried out. Furthermore, the authors do not appear to attempt an analytical solution, or any reasoning as to why the data driven approach is preferred. One possible answer for this is explored in section 3 of this report.

There is also no validation of the MATLAB code within the paper. That said, are several options are freely available. One such was explored as an alternative to GPRMax by (Irving and Knight, 2006) but it was found to be limited in its application.

2.1.4 DMRF-UNet: A Two-Stage Deep Learning Scheme for GPR Data Inversion under Heterogeneous Soil Conditions (Dai *et al.*, 2022)

This paper proposes the use a two-stage Deep Neural Network consisting of a UNet and a Deep Multi-Receptive Field (DMRF) with the aim of reconstructing the permittivity distributions of buried objects of interest. The first stage uses a DNN to remove clutter caused by an inhomogeneity of the soil. The second stage uses the cleaned B-scan from the first stage along with noisy B-Scan data to learn the inverse relationship and then reconstructs the permittivity of the subsurface objects. The

system employs an end-to-end training regime with the aim of minimising information loss. The architecture is shown in Figure 2-6 below.

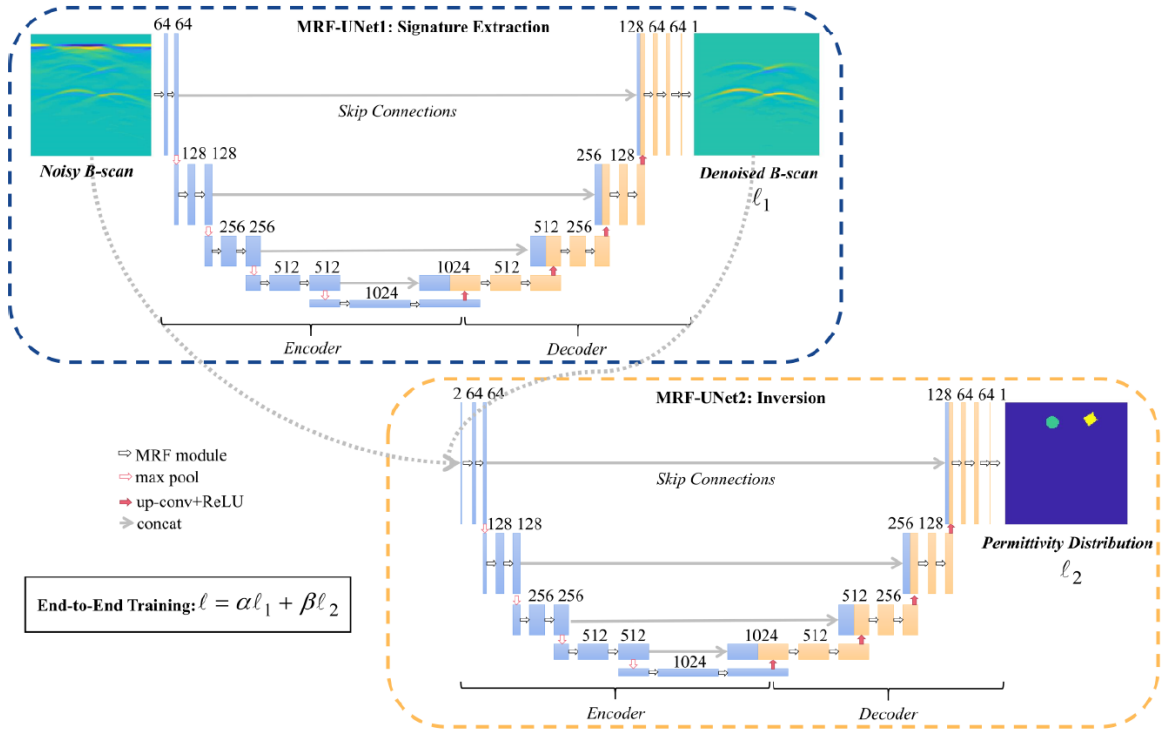


Figure 2-6 The Structure of the Proposed DMRF-Unet. 'MRF Module', 'Max Pool', 'Concat', 'Up-Conv', 'Conv', and 'Relu' Represent The Multi-Receptive-Field Module (Dai *et al.*, 2022)

An example of the output from the model for a single case is shown in Figure 2-7 below and across all test cases summarised Table 2-1. The results show that the approach performs better for single object detection than for two separated or two interfaced scenarios.

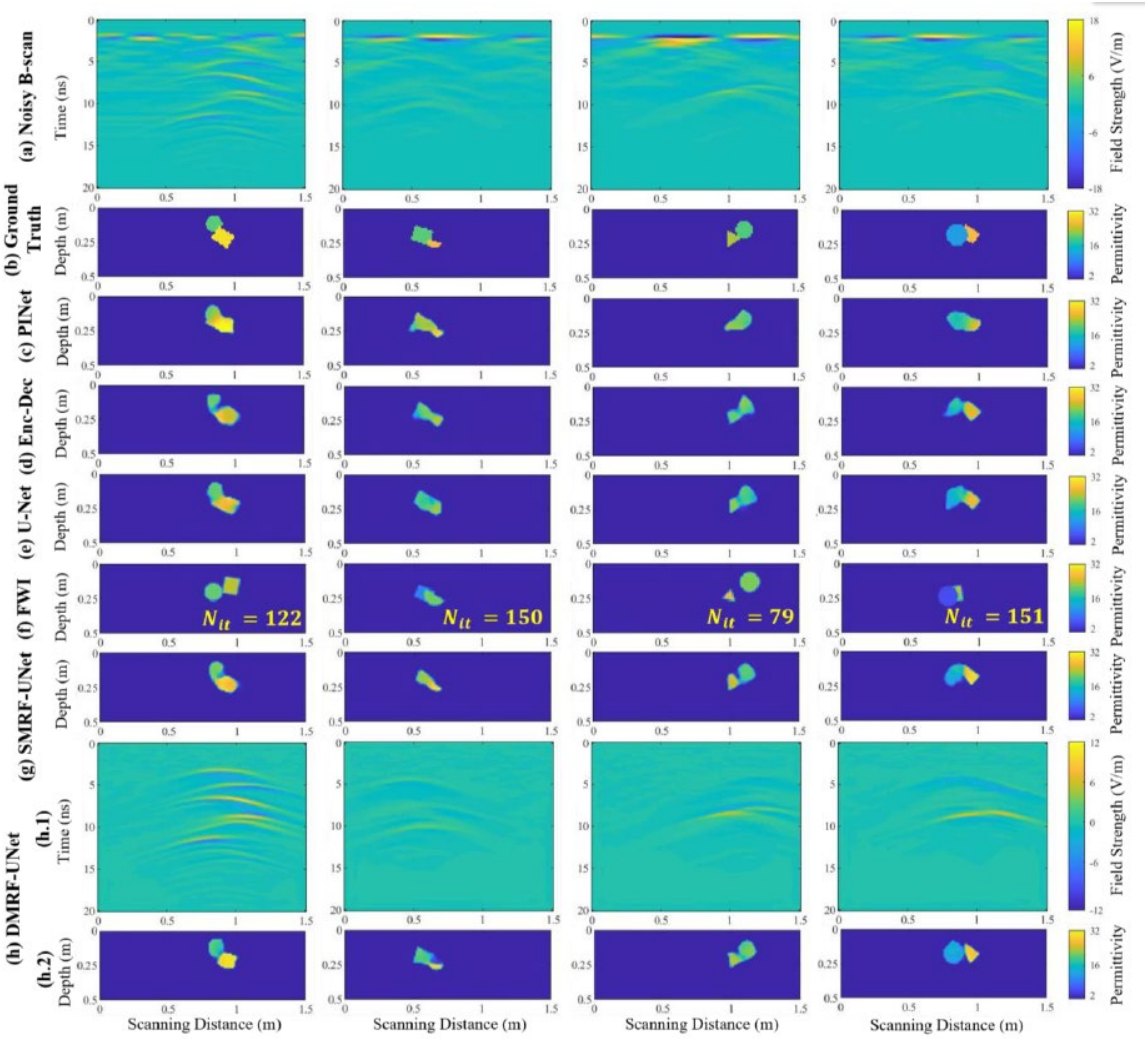


Figure 2-7 Inversion Results Comparison when Two Interfaced Objects are Buried. (Dai *et al.*, 2022)

COMPARISON WITH FWI ALGORITHM ON EVALUATION METRICS OF THE 12 TESTING DATA SHOWN IN FIGS. 5-7

Scenario Type	MSE (↓)			MAE (↓)			MRE (%) (↓)			SSIM (↑)		
	FWI	SMRF-UNet	DMRF-UNet	FWI	SMRF-UNet	DMRF-UNet	FWI	SMRF-UNet	DMRF-UNet	FWI	SMRF-UNet	DMRF-UNet
One Object	1.5693	0.2225	**0.1804**	0.0950	0.0239	**0.0196**	0.6598	0.1736	**0.1558**	0.9709	0.9868	**0.9875**
Two Separated Objects	7.0984	0.8116	**0.5544**	0.3044	0.0675	**0.0596**	1.0541	0.2139	**0.1847**	0.9379	0.9693	**0.9713**
Two Interfaced Objects	3.8661	0.9687	**0.8579**	0.2089	0.0714	**0.0571**	0.7721	0.2656	**0.2116**	0.9590	0.9799	**0.9816**
Ave.	4.7119	0.6676	**0.5298**	0.2028	0.0543	**0.0412**	0.8287	0.2177	**0.1840**	0.9559	0.9787	**0.9827**

Table 2-1 Summary Of Results Across All Test Scenarios In The Paper (Dai *et al.*, 2022)

The paper comments that it takes about 1.5 hours to complete a single inversion and 24 iterations for the FWI approach. This approach takes 0.01 seconds to generate an image, while training the model takes 16 hours. The results from the model were compared against real-world test data and it was found to outperform existing approaches as shown in Table 2-2 below.

Method		SSIM (↑)	MSE (↓)	MAE (↓)	MRE (%) (↓)
FWI		0.9412	10.5406	0.3449	0.8371
PINet		**0.9654**	2.4186	0.1503	0.5290
Enc-Dec		0.9478	0.9227	0.0932	0.3456
U-Net		0.9506	0.6564	0.0820	0.3135
SMRF-UNet		0.9579	0.5044	0.0786	0.3037
DMRF-UNet	#1	0.9820	1.3265	0.7082	1.0962
	#2	0.9589	**0.3881**	**0.0537**	**0.1914**

Table 2-2 Comparison On Evaluation Metrics Of Real Measurement Data (Dai *et al.*, 2022)

2.1.5 Geophysical model generation with generative adversarial networks (Puzyrev *et al.*, 2022)

In this paper, the authors have proposed the use GANs to generate a 2D subsurface dataset which can be used in future applications for deep learning inverse problem applications. The approach is to use 2D models extracted from five 3D stratigraphic models generated using an open-source modelling code called Badlands. Each model is between 8km and 16km long with a depth range between 2km and 4km. The approach used to generate the training data assigns different lithological types representing shale-sand proportions. The data is created using the standard GAN configuration of generator and discriminator which is derived from StyleGAN2 (Karras *et al.*, 2020). It creates a realistic generator with either 24.03 and 24.85 million parameters and an output size of 1x512x512. The training results are shown in Table 2-3.

GAN	Models	Total	Time (s/kimg)			FID
		kimg	1080 Ti	1 V100	2 V100	
StyleGAN2 config-e	Density 512 x 512	7500	156.2	95.4	58.7	35.62
StyleGAN2 config-e	Stratigraphy 512 x 512	6000	155.8	95.7	58.9	22.05
StyleGAN2 ADA	Density 512 x 512	1000	292.1	191.3	156.9	36.03

Table 2-3 Training Statistics and FID scores from the paper (Puzyrev *et al.*, 2022)

The quality of the generated image is measured using the Fréchet Inception Distance (FID) score where a lower score is better. The paper concludes that the

approach adopted has the ability to generate sufficiently detailed and varied artificial models which have features that are comparable to those used in the training dataset in real-time. This allows the artificial dataset to be generated in a cost-effective manner. This is a flexible approach that can be applied to several different data types.

Future development of the method has included the use of conditional GANs allowing the generation of data based on class labels.

2.1.6 Deep Learning for Geophysics: Current and Future Trends (Yu and Ma, 2021)

This review paper has looked at applications of deep learning models to geophysical applications. The conclusion is that most of the work has been centred around application in exploration geophysics, earthquakes, and remote sensing. Several deployments have used a u-net to predict the velocity model (see Figure 2-8 below)

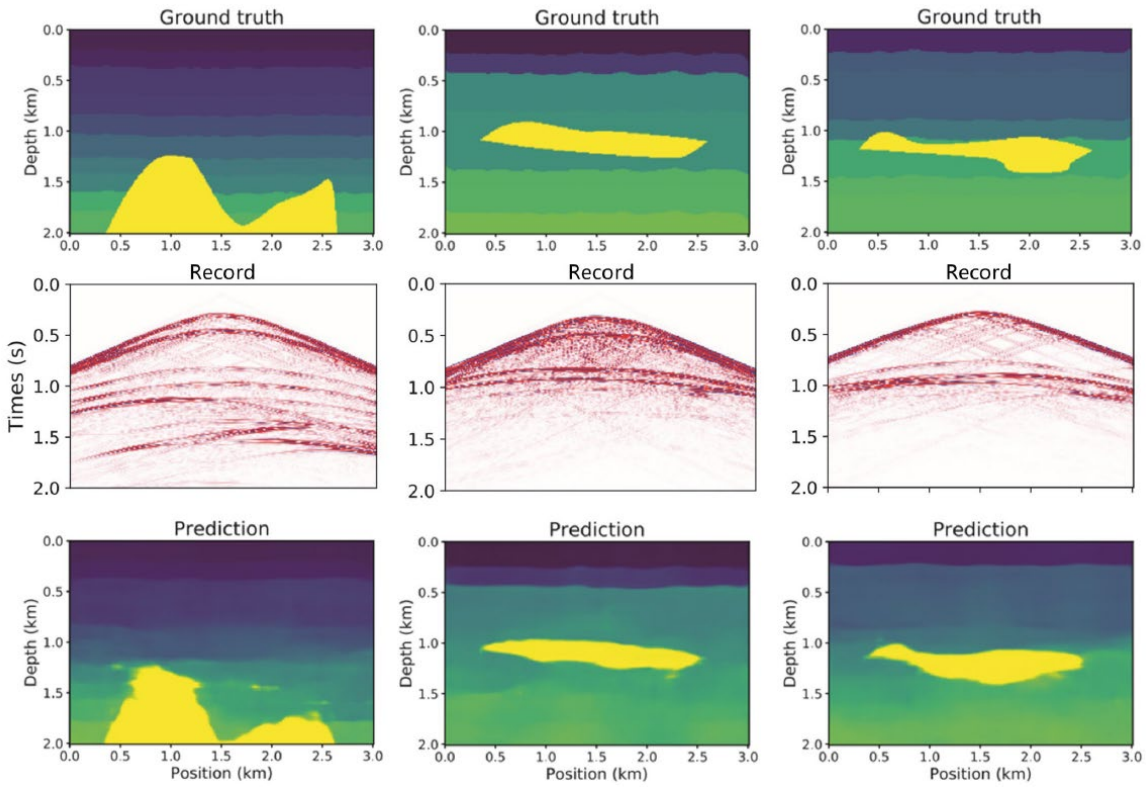


Figure 2-8 Predicting the velocity model with U-Net from raw seismological data (Yang and Ma, 2019)

The paper also summarises how the use of semi-supervised and unsupervised approaches like CAEs and GANs have a great benefit to offer the geophysics community as a way of mitigating the time consuming and expense of labelling datasets.

2.1.7 How Deep Learning Can Help Solving Geophysical Inverse Problems (Picetti, 2023)

This paper provides a summary of the advantages and limitations found of deep learning on seismic survey results from the author's PhD studies. The particular focus was using Deep Convolutional Neural Networks. The author uses GANs to improve the quality of the reflectivity of images from surveys, then applies a technique called deep priors which uses a CNN to precondition the inverse problem. The final stage uses feature extraction techniques developed from work using GPR in landmine detection.

In the final stage the author uses a convolutional autoencoder on patches of mine free B-scans and uses a CNN on the output as shown in Figure 2-9 below, where v is the input to the autoencoder with $\hat{v}$ being the output from the encoder. The CNN then classifies any objects of interest as a '1' and otherwise as '0'.

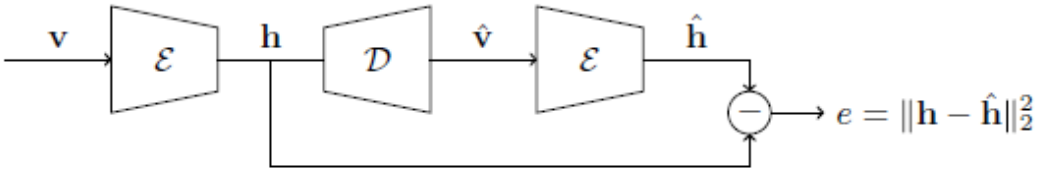


Figure 2-9 Proposal for Anomaly Detection Scheme (Picetti, 2023)

Once all the results are processed, they are merged into a mask which has the same size as the input, and then the final label is computed by hard thresholding the maximum value in 'e'.

The approach uses the maximum value in 'B' scan energy which is a constant false alarm rate technique to compare against other methods. The results were presented using the Receiver Operating Characteristic (ROC), which represents the probability of correct detection and is shown in Figure 2-10 below. It can be seen that the

approach presented in the paper achieves the best Area Under Curve when compared to competitive solutions.

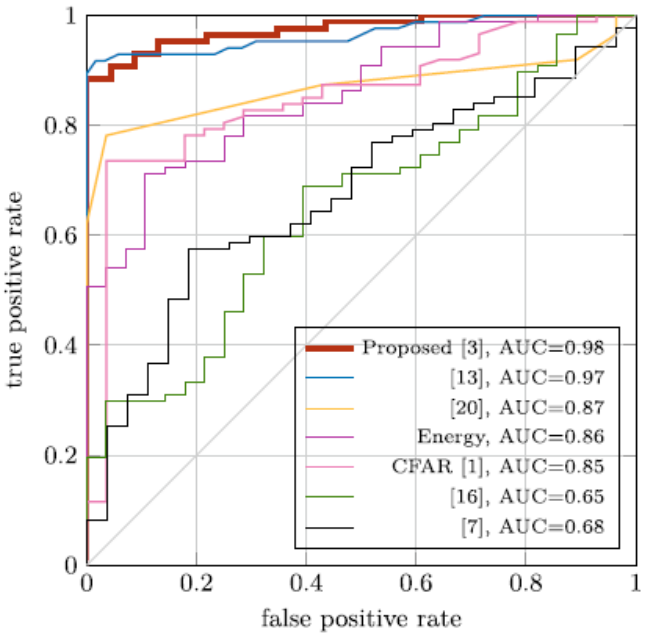


Figure 2-10 ROC Curves for the Landmine Detector when Compared to other Solutions (Picetti, 2023)

2.1.8 A Deep Learning Approach for Urban Underground Objects Detection from Vehicle-Borne Ground Penetrating Radar Data in Real Time (Zong *et al.*, 2019)

This paper proposes a method of detection and classification of objects found by GPR systems in 2D. The approach uses a two-stage process as shown in Figure 2-11 below. The first step is to use the Darket53 and train it on a 2D set of images taken from the ImageNet, COCO, and PASCAL VOC datasets, thus allowing classification. The second step is a two-step process, the first freezes the first 50 layers and uses transfer learning with a GPR based dataset, the second step is to unfreeze these layers to allow the model to predict the location of multiple bounding boxes.

Once training has been completed, the inference is used to obtain both the category and the location of objects detected.

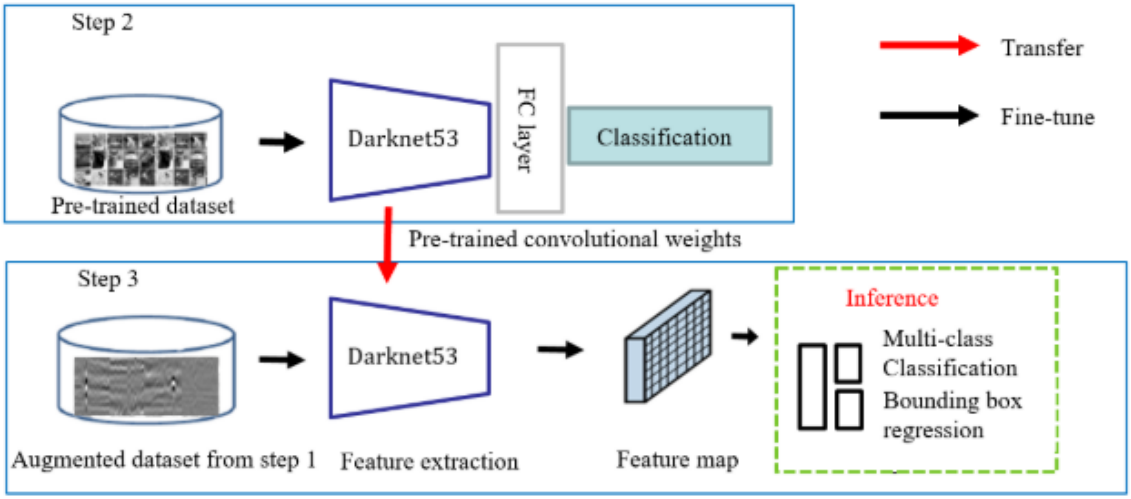


Figure 2-11 The Proposed Method for Training of the CNN (Zong *et al.*, 2019)

The GPR hardware for this was a 400MHz vehicle-based system and was used to identify rainwater wells, sparse/dense steel mesh, bridges, pipelines and cables over a 4.5km scanning length. These detected items were then augmented to create the full dataset of 3033 items.

The results from the testing are shown in Figure 2-12 below. Interestingly, the traditional confusion matrix is not used. Precision terms, recall and the F1 score are shown instead. In the paper, the formula for the recall and the precision are the same, however in the table the values are different. The results do show that the ability to detect objects correctly is relatively high (89%) while the accuracy of the model is slightly lower (87%). The approach seems to have the most success classifying rainwater wells and having the lowest success in detecting voids. Whereas the paper does not explain the reason for the apparent lack of success on nonconductive items (nonmetal pipes and voids), it can be assumed that the signals were not dominant enough compared to the metallic and water-based signals.

Classes	Metrics		
	Recall	Precision	F1score
Rainwater wells	0.912	0.954	0.932
Cables	0.923	0.857	0.889
Metal pipes	0.852	0.920	0.885
Non-metal pipes	0.833	0.833	0.833
Sparse steel reinforcement	0.942	0.867	0.903
Dense steel reinforcement	0.969	0.839	0.900
Voids	0.816	0.816	0.816
Average	0.892	0.869	0.880

Figure 2-12 The Results from Testing of the Network (Zong *et al.*, 2019)

2.1.9 Deep Learning Based Subsurface Target Detection from GPR Scans (Hou *et al.*, 2021)

This paper proposes an improvement over existing Regional Convolutional Neural Networks (R-CNN) as a method of automation in object detection of GPR scans. This paper enhances the Masked Scoring method, achieved by the introduction of new anchoring schemes.

The proposed method aims to address several challenges when adapting deep learning to GPR based problems. The identified issues are firstly the physical attributes of the scanning hardware and the objects being detected. The second issue is the properties of the medium being scanned (moisture content, homogeneity, etc.). The final issue is that when passing GPR scans into a model, the model may lose useful information and lead to redundant processing. The proposed architecture is shown in Figure 2-13 below.

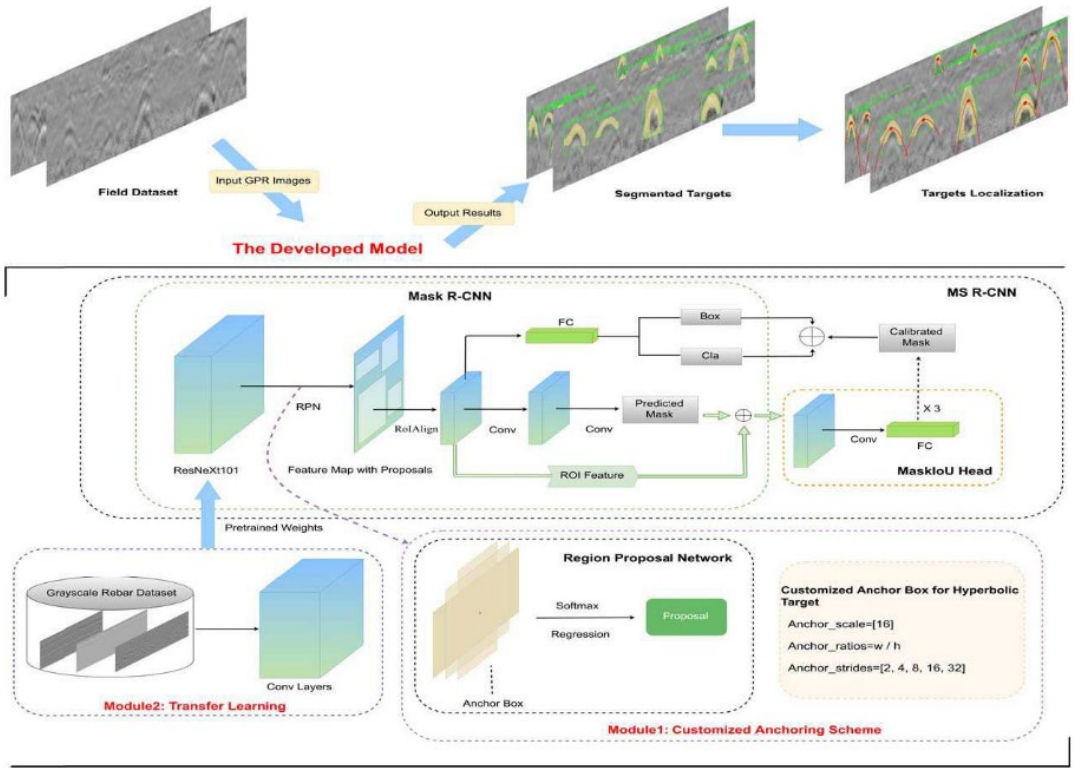


Figure 2-13 The Proposed GPR Framework (Hou *et al.*, 2021)

The proposed methodology can be considered in 2 stages. The first stage of a traditional mask scoring approach is modified by first analysing the hyperbolic signatures to develop a customisable anchoring scheme which improves the object detection performance. The second stage is extracting the signatures from a cluttered background and then the hyperbolic fitting is performed to find a peak, thus identifying the approximate location.

For real world data collection, the GSSI 2GHz antenna coupled with an SIR-4000 was used and had a predicted penetration depth of around 0.5m (depending on moisture content).

In Figure 2-14 below, a sample of the results are shown for the new improved methods. The first step is to identify and extract the hyperbolic signatures (figures b & c) and then these hyperbolic signatures are identified and using the method to

predict the peak of the hyperbola. The method also demonstrates a resilience to small and dense hyperbola.

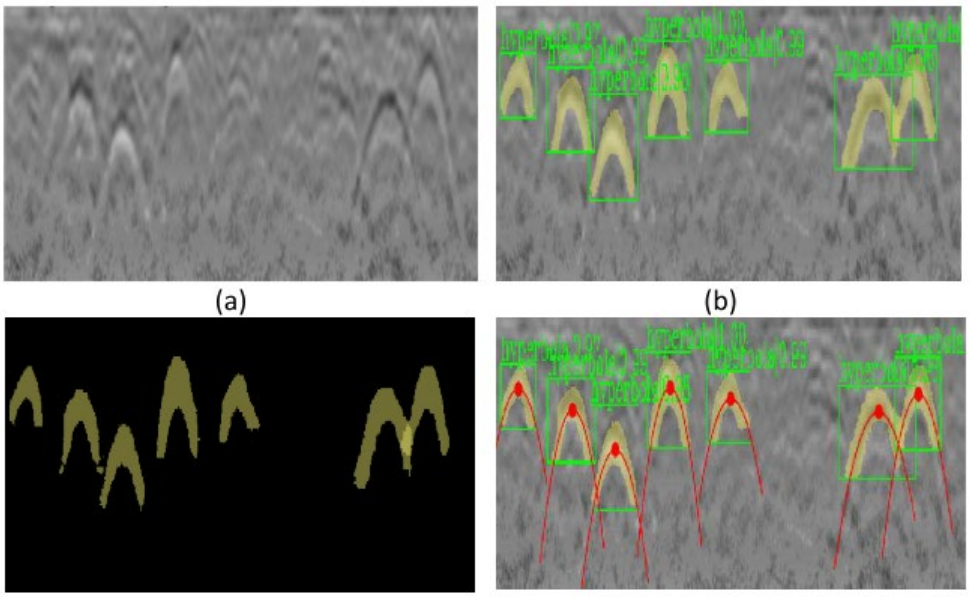


Figure 2-14 (a) Pre-processed root GPR scan. (b) Detected and Segmented Results. (c) Only Picked Mask Clusters. (d) Hyperbolic Fitting Results with Peaks (red point) (Hou *et al.*, 2021)

A comparison of the average precision (AP) for different intersection over union values of 50 & 75 and for small and medium objects are shown in Figure 2-15 below.

Framework	AP	AP_{50}	AP_{75}	AP_S	AP_M
Mask R-CNN	27.9	31.4	21.2	27.7	27.8
Cascade Mask R-CNN	34.3	38.2	21.7	32.6	29.5
MS R-CNN	33.6	35.4	26.9	30.3	36.5
Improved MS R-CNN	**35.0**	**38.6**	24.5	**32.7**	**41.5**

Figure 2-15 Comparison Of Segmentation Results Via Different Instance Segmentation Frameworks (In Percent) (Hou *et al.*, 2021)

The paper has proposed an improved method for automated signature detection and segmentation using deep learning technique, however there are several limitations of the work identified, which include:

- The size of the datasets used.
- The deep learning model needs to be designed to accept the input characteristics.
- The root dataset was dependent on the soil parameters, which, in wet conditions could result in missing some plant root images.
- The recognition of roots is inferior when compared to linear objects such as rebar.

2.1.10 A Novel Method of Hyperbola Recognition in Ground Penetrating Radar (GPR) B-Scan Image for Tree Roots Detection (Zhang *et al.*, 2021)

This paper proposes an alternative method to the root detection problem in the previous paper. In this paper, the team proposes to use both multidirectional features as well the symmetry of the hyperbola in the B-scans. A faster RCNN was trained on a mixed set of GPR B-Scans in order to locate any potential hyperbolas. The peak area was then identified from any connected data from the 4 directional gradient graphs. Any intersecting hyperbolas were separated using the symmetry hypothesis. The final stage was to use a Hough transform to detect the lines and coordinates of the hyperbolas. From this data it would be possible to identify root radius and position.

The dataset was made by combining synthetic data from GPRMax with field data from 35 ancient tree root systems and fresh grapevine.

In the GPRMax Simulation the antenna frequency was 900MHz. The actual hardware used in the field was the GSSI SIR3000T tree radar and a 2D domain was used. The Architecture used for the RCNN is shown in Figure 2-16 below, with the backbone using VGG, Resnet50 and Resent100 for feature extraction. The output from this was sent into a region proposal network and a regression network.

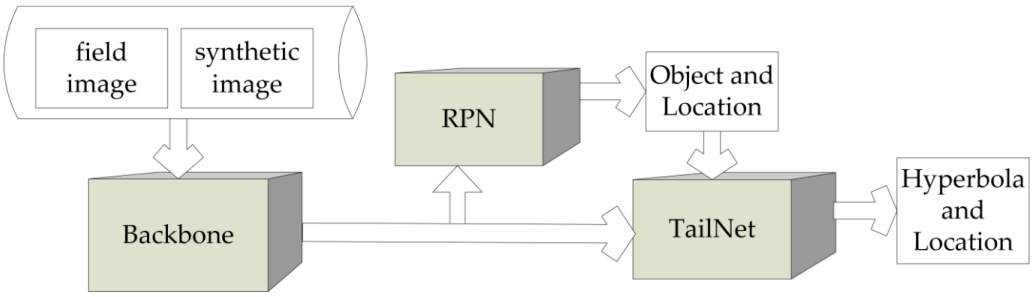


Figure 2-16 The framework of hyperbola region detection (Zhang *et al.*, 2021)

The results of the feature extraction are shown in Figure 2-17 below and show that the ResNet101 has the best Average Prediction). When the paper compares the results for the hyperbolic extraction (see Figure 2-18 below), the relative error is inversely proportional to the diameter of the root and is 60% in the multi root assessment.

Backbone Network	VGG16	ResNet50	ResNet101
mAP	84.94%	86.67%	89.71%
TPI	0.987 s	1.273 s	1.662 s
FPS	1.02	0.79	0.62

Figure 2-17 The Comparison for the mAP, TPI And FPS of Three Different Backbone Networks. (Zhang *et al.*, 2021)

	Radius (m)			Depth (m)			Offsets (m)		
No.	Real	Pro	RE	Real	Pro	RE	Real	Pro	RE
1	0.015	0.019	26%	0.265	0.266	0.4%	0.60	0.595	0.8%
2	0.025	0.027	12%	0.255	0.225	11.0%	0.60	0.60	0%
3	0.035	0.029	17%	0.245	0.232	5.3%	0.60	0.590	1.6%
4	0.10	0.103	3%	0.38	0.377	0.8%	0.60	0.60	0%
5	0.15	0.134	10%	0.33	0.33	1%	0.60	0.595	0.8%
6	0.20	0.189	5.5%	0.28	0.29	3.6%	0.60	0.595	0.8%

Figure 2-18 Single Root Radius and Location of Single Root

The paper concludes that the method adopted shows a significant effect in tree root detection as well as position and size estimation. In a real-world situation this method could not guarantee capture of vertical roots because the amount of water content in the soil has a detrimental effect on the detection system.

2.2 Inverse modelling

The papers mentioned in this section demonstrate the complexities of solving the inverse problems using numerical techniques when applied to RADAR sources.

2.2.1 Reconstruction of 3D objects from multi-frequency experimental data with a fast DBIM-BCGS method (Yu, Yuan and Liu, 2009)

This paper is applied to airborne RADAR, which is less complicated due to the removal of impurities contained within the transmitted medium (air in this case and soil in GPR).

This paper presents a solution to the inverse problem using the Distorted Born Iterative Method – Stabilised Biconjugate Gradients (DBIM-BCGS). The approach used in the paper has great potential as it is based on the Born series expansion, which is explained in section 3.5.1 in this thesis. The paper is generally light on technical detail, and only includes a rudimentary explanation of the theory used. It also doesn't include a discussion of the results which leaves the reader to infer the quality of the reconstruction using this method based on a single graph at the end. The images in this paper show that there is a reasonable degree of reconstruction accuracy, and this is validated by the graph at the end of the paper though the reader is left to assume that the accuracy is 0.33% and not 33%.

DBIM is an iterative approach that implements the Distorted Wave Born Approximation by updating the Green's functions in every iteration. In the case of

noiseless solutions DBIM has been demonstrated to be superior to the Born Iterative Method. The approach taken in this paper is to further enhance the capability of the DBIM approach.

The paper has no recommendations of where the theory can be improved and what possible avenues can be explored. There is no dimensional accuracy comparison between the calculated data and actual data and, while the 3D reconstructions are plotted against the XYZ Cartesian coordinates, the actual output from the method has not been produced.

Several things are left unexplained. They include the choice of a frequency band of 3 to 8 GHz, the effects if this frequency band is changed, and what if any affect it has on the results.

In addition, the rig setup is very specific, and no explanation as to the reasoning behind the setup has been given, e.g., as to why the angular increments or the radial distance of the antennas have been chosen and what are the impacts on the results if these are changed.

The paper suggests that the use of the Born Method is suitable to the solution of the inverse GPR problem.

2.2.2 Two-Dimensional Linear Inversion of GPR Data with a Shifting Zoom along the Observation Line (Persico, Ludeno and Lambot, 2017)

This paper presents an alternative method to the linear inverse scattering GPR problem. One of the main difficulties with the inverse problem is that the size of matrices can become intractably large or too large for computational analysis. This in turn leads to a large expenditure in both computational hardware and time.

The proposed method in this paper is a way of minimising this expenditure using a shifting zoom approach. This approach uses is a modified windowing technique, which effectively slides over the data allowing only a section of data to be processed at a time. There is no comparison of the results from this approach against a more traditional approach, so it is very difficult to make a judgement on the effectiveness of this approach.

Besides the shifting zoom approach, the inverse problem is solved using a traditional Born approximation of the solution, so it is difficult to say whether the novel introduction of a shifting zoom or the well-trodden path of the Born approximation method of solving the linear inverse problem is responsible for the accuracy of the results.

Neither of these the two papers in sections 2.2.1 and 2.2.2 make any reference to the inherent issues with solving the inverse problem or discuss how the boundary conditions at the antenna have been derived. The simplest way of calculating this would be to use a Green's function (these are discussed in detail section 3.4.1). This then allows an estimation of the internal behaviour of a system based on the external known properties. As can be deduced, the derivation of this is therefore critical to the success of the approach.

From both papers, it seems that the Born approximation or a derivative is the most suitable empirical process to be used in solving the inverse problem, and as a result this is going to be employed in the inverse problem solution section of this thesis.

2.2.3 Interpreting complex, three-dimensional, near-surface GPR surveys: an integrated modelling and inversion approach (Millington *et al.*, 2011)

The paper presents results for the development of integrated Finite Difference Time Dependent (FDTD) and a linear tomographic inversion method in order to interpret near surface 3D GPR data. The method uses the Born approximation and truncated Single Value Decomposition (SVD) to create the reconstructions. The approach adopted is very similar to the method described in chapter 3 of this thesis. The results are shown in Figure 2-19 below.

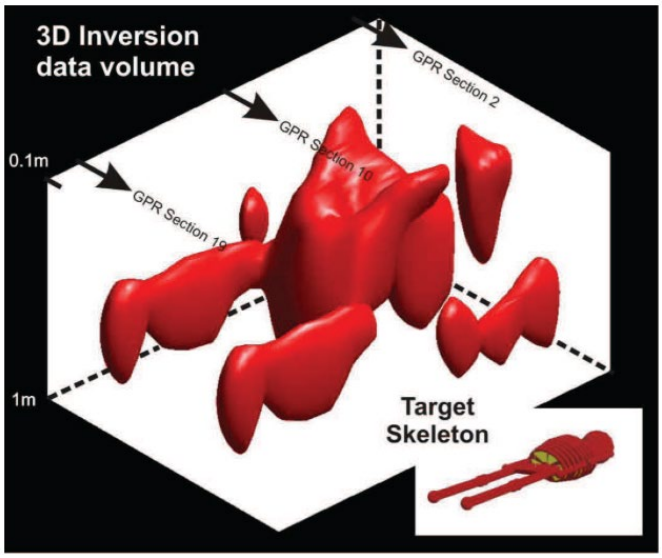


Figure 2-19 3D Inversion Results for a Clandestine Burial Target (Millington *et al.*, 2011)

This method works by performing an inversion model on each individual B-Scan, which are then sewn together to create a 3D iso-surface with a threshold value of approximately 70%. It is noted in the report that the relative depths of the upper parts of the skeleton have been maintained, and even individual limbs have been

reconstructed. However, there are also smearing effects which produced false dimensionality as well as issues with the reconstruction of the shallowest parts of the skeleton.

The paper goes on to add clutter to the signals, in an attempt to assess the robustness of the algorithm. It is found that while the legs and pelvis are resolved albeit more loosely than without the clutter. The level of smearing is also worse in the cluttered model.

The paper concludes that the approach has overcome a problematic closely spaced objects area of the Born approximation and that the approach is invariant to initial selection parameters.

2.3 Image Mapping

This section reviews two papers taking different approaches to the mapping of 3D images. The first utilises an approach based on the use of spherical wavelets and included in this section as a potential solution to hardware limitations in both the GPR and deep learning training. The second paper utilises a Deep Learning approach.

2.3.1 Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (Zhu *et al.*, 2017)

Cycle Generative Adversarial Networks (GANs) as described in (Zhu *et al.*, 2017) have demonstrated some success in mapping between two different images, for example mapping an image of horse colours onto a zebra, which results in the Zebra losing its stripes and vice versa (resulting in the horse gaining stripes). If it is possible to map from one latent space, a hidden space which is a dimensionally reduced representation in which similar data points are close together, and is explained in

more detail in section 5.2.6, to another, then this is the key to mapping from the GPR domain into the image domain and solving the inverse problem. The mapping can also be employed by mapping from a 2D domain into a 3D domain. Further explanation on how this relates to this research is in section 5.3.

The approach adopted in this paper is to utilise unpaired mapping, which consists of a source set and a target set with no information provided as to which item in the source set matches the target set. This approach ignores the supervision at the item level. However, it does require some level of supervision at the set level, in other words:

$$\hat{y} = G(x), x \in X \quad \textbf{2.1}$$

And this is identical to images that belong in the mapped to set Y. This is achieved using adversarial training to classify $\hat{y}$ apart from any member of set $Y(y)$. This approach is not without difficulties and using the adversarial approach on its own has led to mode collapse (one of the main issues of GANs, discussed in section 5.2.5) resulting in all input images mapping directly to the same output image. This is tackled by using cycle consistency. In this case the practise of mapping from $G: X \rightarrow Y$ and then using another translator $F: Y \rightarrow X$.

The results are very promising when mapping between 2D spaces, in this case from a zebra to a horse or between 2 different styles of painters. The authors did compare the output from their proposed architecture to several existing architectures and demonstrated that this approach leads to a significant improvement in the mapping between the two spaces as shown in Figure 2-20.

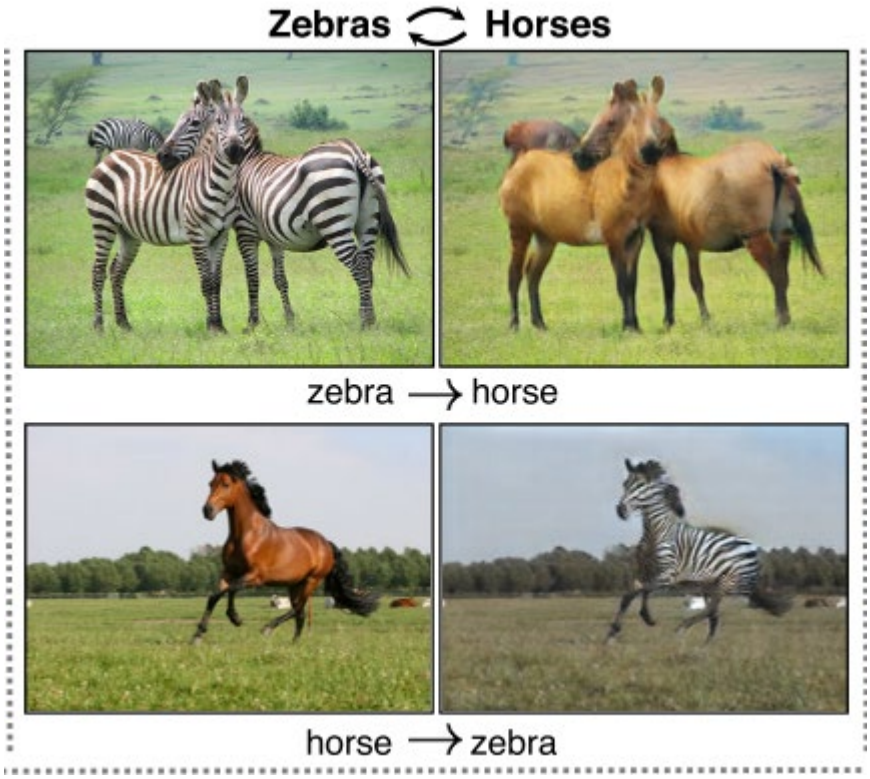


Figure 2-20 Showing the Results of a Cycle GAN on Images of Horses and Zebras (Zhu *et al.*, 2017)

The authors also compared the image quality of the output using the FCN metric, which evaluates how interpretable the generated images by a Fully Connected Network (FCN) to compare a predicted label of a generated image, and have found that the data quality is high (as shown in Figure 2-21).

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [32]	0.40	0.10	0.06
BiGAN/ALI [9, 7]	0.19	0.06	0.02
SimGAN [46]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [22]	0.71	0.25	0.18

Figure 2-21 FCN Scores for Different Methods Evaluated on Cityscapes (Zhu *et al.*, 2017)

A section of the paper identifies several limitations to this approach:

1. There is little success in tasks which require geometric changes, for example the task of dog to cat transfiguration.

2. There is a gap between what can be achieved by using paired data and unpaired data.

2.3.2 This paper does seem to push the boundaries forward of what can be achieved through using unpaired data in an unsupervised setting. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modelling (Jiajun *et al.*, 2016)

The paper (Jiajun *et al.*, 2016) uses a 3D Generative-Adversarial Network (GAN) combined with a Variational Auto Encoder (VAE) in order to generate high resolution images of objects from 2D probabilistic space. This allows objects to be explored without the underlying Computer Aided Design (CAD) models. The input data used consists of 3D voxel elements (a voxel is, in effect a volumetric representation of a Pixel) which represent the object, in what may be considered an alternative to the marching cube approach. This work shows with success that it is possible to map from a 2D image into a 3D voxel representation. The paper also demonstrates that the size of the latent space between the encoder and the decoder has a direct effect on the quality of the output, the smaller the latent vector, the lower the resolution, and larger the latent vector, the higher the resolution.

Overall, the paper provides a good description of the architecture, but the mapped 3D images are in some cases less than finalised, with no suggestions as to how to improve the quality of the mapping. The paper gives some description of the training parameters used, however fails to mention the dataset size and the number of epochs used in training.

In addition to the mapping between 2D and 3D spaces, there is a classification section which shows that the unsupervised model approach adopted is better than other networks currently available.

The size of the 2D image used in the input is unclear and never fully specified, however it is deduced that the size is at least 257 pixels. The paper also suggests that multiple images are taken at different viewpoints. The main purpose of the 3D GAN paper is to create a method of creating 3D images based on a latent vector supplied, whereas in this work the intention is to recreate a 3D image from a 2D array of values which represent the GPR response.

The paper does not fully show how well the 2D image is mapped to the 3D space for any given image of an object. In other words, what is the accuracy of the recreation. The focus of this paper seems to be about how well the classification is achieved.

2.4 Geophysics in Agriculture

2.4.1 Ground penetrating radar (GPR) detects fine roots of agricultural crops in the field (Liu *et al.*, 2018)

(Liu *et al.*, 2018) outline their attempts to detect roots of agricultural crops in the field using GPR. The study was conducted in 4 locations with different soil types, moisture content and several different types of cereal crops and scanned using a 1.6GHz GPR system.

The study sites were spread across Texas in different types of soil (silty clay loam, clay, sandy clay, and sandy soils). The crops used for the study were winter wheat and Bio energy sugar cane. The team have selected an off-the-shelf GPR system with a frequency of 1.6GHz. At each location the team measured the wave speed

velocity, with the dielectric constant being determined by the relationship between the wave speed and soil dielectric permittivity (Al Hagrey, 2007).

The soil surface was cleared of debris before each scan, which is unrealistic in a real-world application, and the scan was performed between the two middle rows of each test plot. After scanning took place, soil cores were collected to measure the root parameters. An example of a scan is shown in Figure 2-22 below.

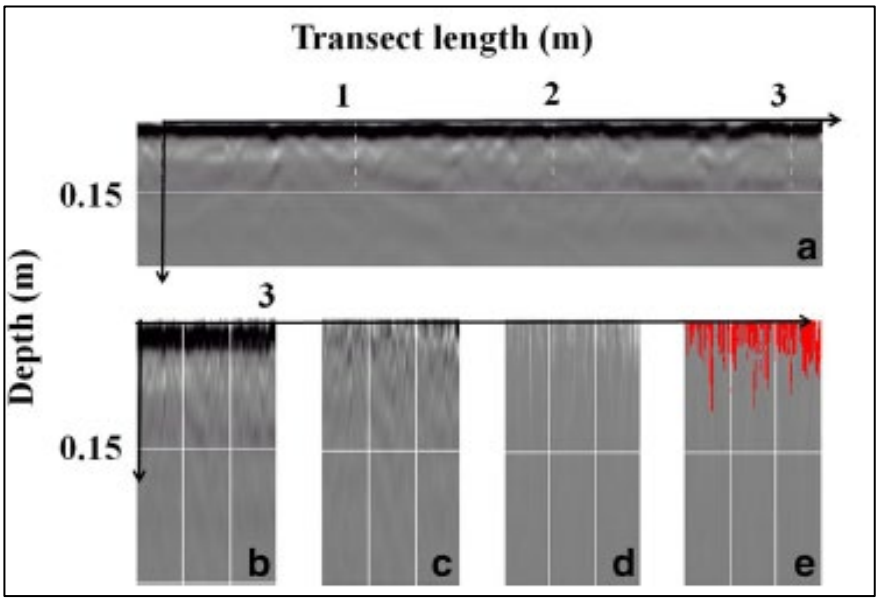


Figure 2-22 Example GPR Scan of Root Biomass Taken Within 15cm of the Surface (Liu *et al.*, 2018)

Statistical analysis and linear regression models were used to verify that GPR could detect root radius and mass. The results show that there are significant relationships between the signal and the root characteristics. The results lead to 3 general conclusions:

1) Root characteristics of wheat can be detected in a wet field.

2) The use of average pixel intensity without a threshold compared to intensities threshold range is better to reflect the information on root mass.

3) Negative correlations were detected in dry fields. This is because high levels of water content within the soil have a beneficial effect on the ability to assess root characteristics.

The paper concludes that there is significant potential in the use of GPR to detect fine roots in cereal crops. However, the relationship between root characteristics and GPR signal varies significantly and is generally better in moist soils than in drier soils.

The results from this paper are concerned with the prospect of detecting the biomass of the root cluster and not with any image generation of the root cluster. According to the paper only one type of crop was planted in each location. Perhaps it would have been better to grow each of the 4 types of crops in each of the 4 locations, thus confirming if the relationship is linked to crop or soil and moisture content. The report comments on the effect of water within the soil, however there is no mention of water content within the root system and the effect that this may have on the quality of the returned signal. This is significant for this research since a potato tuber contains a large proportion of water.

GPR has been chosen for this research project as opposed to other imaging techniques such as ERT and Seismic Attenuation Profile, because it is a non-invasive approach, with little health and safety concerns. It is beneficial to consider alternative imaging technologies.

2.4.2 3D X-Ray CT Phenotyping of Plant Roots (Xu, Valdes and Clarke, 2018)

The paper on 3D X-Ray CT Phenotyping of Plant Roots (Xu, Valdes, and Clarke, 2018) reviews existing and potential approaches for the analysis of 3D X-ray CT phenotyping images of plant root structures. An example of a typical output from a CT scan of a root system is shown in Figure 2-23. The paper also mentions the use of deep learning algorithms in the form of a 3D recursive Convolutional Neural Network (CNN) used in the medical industry. However, it has not been used in root imaging in this case. It is noted that the main detriment to using deep learning is that CNNs require large volumes of training data, which at present are not always available. The main bottleneck to using this technology is the existence of established commercial grade software.

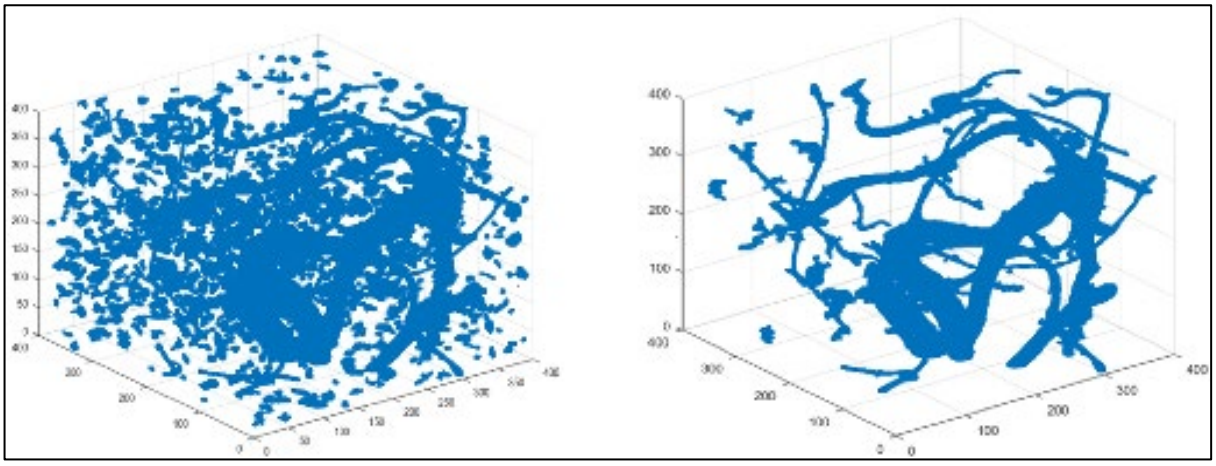


Figure 2-23 CT Scanned Root System, Before and After Filtering (Xu, Valdes, and Clarke, 2018)

The CT scanning technology used is an offshoot of the medical imaging technique. In medical imaging, the tubular structures of interest are highlighted using higher greyscale colours than the surrounding structure, due to different X-Ray absorption rates. Roots do not possess this characteristic and as a result the largest source of error is in the root-soil boundary.

CT Scanning is deemed impractical outside of the lab as it will be difficult to place the receivers without disturbing the ground and hence the crops. Other considerations are the effects of X-Rays on crops and the final product. For example, there may be legal requirements that limit the use of X-rays on food. It is unknown whether the use of X-Rays render the crops unfit for human consumption. Such issues will limit the capability of an X-Ray based system.

2.4.3 Data Acquisition Methodologies Utilizing Ground Penetrating Radar for Cassava (Manihot esculenta Crantz) Root Architecture (Delgado, Novo and Hays, 2019)

This paper compares the functionality of two GPR antennas when scanning for Cassava roots. The paper describes in detail the test rig setup, the size, position and orientation of the roots. The distance between the scans was controlled by using a plate with recesses cut out and then the approach adopted to create the 3D iso-surfaces. This approach uses GPRSlice to signal process the GPR signal and then AutoCAD to create the surfaces based on several lateral scans and a significant amount of signal processing. The distances chosen between each radargram were 2.5cm, 5cm, and 10cm and the results show a severe drop off with the increasing space, such that for the medium result, the RMSE and standard deviation are significant.

The paper found that as the roots became obliquely angled, so the error becomes significant. It also found that there were no root features evident when the root ran parallel to the direction of the scan.

This paper is of interest because it uses GPR on sub surface tubers and shows some of the issues with a traditional approach. It does also conclude that current hardware could be used to accurately estimate single root metrics.

2.5 Chapter Summary

In this Chapter, a review of literature related to this thesis has been performed. It evaluated published contemporary papers that use GPR and deep learning, propose analytical solutions to the GPR based inverse problem, image mapping techniques for both 2D-2D and 2D-3D as well as using GANs to generate 3D images. The final group of papers studied imaging techniques applied to agricultural applications, covering the GPR and CT scanning of root systems. The literature review has demonstrated that this is an active area of research, and while there has been significant work on the use of deep learning to process GPR signals, it has been largely confined to munition detection. There have also been attempts to solve the inverse problem using deep learning, but this has focused on civil engineering applications or a general empirical approach.

The following work builds upon existing methods and introduces a novel approach to the inverse problem solution with relation to agricultural applications.

3 Inverse Modelling

In section 2, the literature review has shown that there are two possible solutions to the inverse problem. an analytical solution, and a data-driven solution. This chapter investigates the empirical solution and serves two purposes. The first is to explain in detail what the inverse problem is and why it is so complicated, which would aid the understanding of the data-driven solution. The second is to highlight the rigidity of the mathematical solution, which is unsuitable for real time processing and the reason it was discounted as a solution.

Generally, mathematical models are created when the influence of a known set of inputs have created a measured set of outputs. Examples include:

- Spring-Mass oscillations.
- Thermal conductance.
- Newton's laws of gravity.

In each of these approaches, there is a set of known parameters (the model), and from the understanding of these it is possible to make predictions for a given set of inputs. This is known as the forward problem. There are two steps to solving classical forward problems:

1. Use the physical parameters to define the system.
2. Use the measured data to make a prediction.

There are, however, some cases where instead of wanting to know the output of the model for a set of inputs, the opposite is required, or in other words here is a measured value, but what caused it? The best example of this is the detection of cosmological bodies, e.g. for a set of gravitational measurements and given the

Newtonian gravitational model, what bodies are causing the gravitational changes? This is a classical inverse problem.

Placing the above description into a more general mathematical language gives equation 3.1:

$$r = \mathcal{F}(m) \tag{3.1}$$

where r is the existing set of observed data and ($\mathcal{F}$) is a forward model of known parameters (m).

One important concept in the inverse problem is “residuals”. Residuals are the differences between the forward model prediction and the data, which show how accurate the inverse model is. If the forward problem is linear, then the inverse problem will be linear. Conversely, if the forward problem is nonlinear, then so will be the inverse problem. In general, solutions to the wave and scattering problems are nonlinear. As an interesting aside, in general, linear inverse problems were solved theoretically towards the end of nineteenth century. Of the nonlinear inverse problems, only the inverse 1D scattering and inverse spectral problems were solved before 1970. Due to the relatively simplistic nature of these two problems, they were solved by numerical methods, as opposed to the more complicated problems which are better suited to computational approaches.

This chapter discusses the general approaches used in solving inverse problems in section 3.1, then using this knowledge creates a general solution to Maxwell’s equations in section 3.2. This is then applied to a simplified forward model to show the application of the physics in an ideal solution in section 3.3 and finally to a real-world problem and to show some of the issues with this approach in sections 3.4 and 3.5.

This chapter is included to demonstrate the complexity of existing numerical approaches used to solve inverse problems. These include:

1. The numerical models are computationally complex.
2. Therefore, the models require long processing times.
3. The numerical solutions are sensitive to noise in the data which can cause the convergence of the algorithms to be difficult.

The data driven approaches adopted in chapter 5 aim to address these points and create a more robust approach to this problem.

3.1 General Solution Approach

As stated in the introduction to this chapter, in order to solve a nonlinear inverse problem, the forward problem has to be known, so for a given forward model ($\mathcal{F}$), a noisy measurement set (r) can be reordered based on a set of inputs (m) which can be written as in the simplest form in equation 3.2.

$$\mathcal{F}m = r \tag{3.2}$$

The aim of the inverse problem is to find the set of inputs, based on a forward model and a set of measurement data, or in other words gives equation 3.3.

$$\mathcal{F}^{-1}r = m \tag{3.3}$$

However, solving an inverse problem is rarely simple. There are many issues when performing an inverse problem. The first, and possibly the most significant is that the solution to 3.3 is often ill-posed. An inverse problem is classed as ill-posed when the one or more of the following conditions are met:

- There is more than one possible solution for each data point.
- The solution is not unique.

- The problem is unstable, for example if while solving the problem changes to the inputs (δm) and changes to the measurements (δr) become large.

Regularization is employed to solve ill-posed problems. Regularization involves introducing a parameter into the solution that allows the solution to be changed in a stable way while maintaining an outcome that is close to the original solution as this parameter approaches zero. Three main approaches are adopted for solving linear ill-posed problems.

- Tikhonov Regularization, also known as ridge regression.
- Spectral Cut-off, which approach utilises single value decomposition.
- Iterative Regularization.

The following sub-sections cover these approaches in more detail.

3.1.1 Tikhonov Regularization

This approach is often used in problems where there are large numbers of parameters and offers an improved efficiency solution at the expense of a tolerable amount of bias. The general form is shown in equation 3.4, where y is the input data, X is the design matrix, I is the identity matrix and α is the ridge parameter, which is greater than 0. The reason this is known as the ridge regression is the "**αI"** term adds positive values along the diagonal "ridge" of the covariance matrix $\boldsymbol{X}^T\boldsymbol{X}$.

$$\hat{\beta}_r = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{I})^{-1}\boldsymbol{X}^T y \qquad \textbf{3.4}$$

The main benefit of this approach is that it yields stable solutions by shrinking the coefficients, however it is sensitivity to the input data.

3.1.2 Spectral Cut-off

Spectral cut off uses Singular Value Decomposition (SVD), which is fully explained in section 3.1.3. The main application areas for this type of approach are in small to medium size problems due to the heavy computational cost associated with SVD.

3.1.3 Singular Value Decomposition

Single Value Decomposition is a linear algebraic tool which decomposes a M x N matrix into three. Conventionally, these matrices are called the S, U & V matrices. The S matrix is a diagonal matrix of size M x N, the U matrix is of size M x M and the V is of size N x N. In this application, the outer products of the first column of the U matrix and the first row of the V matrix are taken and this forms a matrix of size M x N which is then multiplied by the first entry in the diagonal matrix S to give it the correct magnitude.

Since the first row and column of the U & V matrices represents the common parts of the original data (which corresponds to the soil and the air/ground boundary), this can then be subtracted from the original matrix thus removing a lot of unwanted background data as shown in section 4.2.2.

3.1.4 Iterative Regularization

In this method, the iteration count is used as the regularization parameter when iterative methods are employed in solving the problem. The approach adopted is shown in equation 3.5 to solve the linear problem in equation 3.2.

$$m^0 = 0$$
$$m^n = m^{n-1} - \tau \mathcal{F}^H(\mathcal{F}x^{m-1} - \bar{y}), \qquad n = 1,2,\dots \tag{3.5}$$

In certain values of τ, the scheme is known as the Landweber iteration and care must be taken in the presence of noise. This is because the error of reconstruction

decreases during initial calculations, but then starts increasing, although the objective function continues to decrease in a process known as semi-convergence.

This approach is more complex than the other two approaches mentioned here.

3.2 Electromagnetic physics

This section walks the reader through a general solution to Maxwell's equations, the equations that summarise the classical properties of an electromagnetic field, which is the underlying principle of GPR. Therefore, they are a fundamental building block, not only of the inverse problem that cannot be solved without them, but of GPR itself. GPR is used to extract data from below the ground's surface, but additional computational process is required to translate it into images that are useful to human operators. These images are used to make informed decisions about the nature of the sub surface area. Then into a derived solution to the forward model and finally into the complexity of building and solving an inverse model. This section is using existing methods to solve the inverse problem.

In order to define the equations, it is important to first build the forward model, and then from this the inverse model can be created. Both models must exist in order to minimise the error between the two, leading to a valid solution.

Ground Penetrating Radar (GPR) is described in two parts. The first is Maxwell's equations and the second is the relationships that quantify the material properties. Maxwell's equations are shown in equations 3.6 thru 3.12:

$$\nabla . \tilde{E} = -j\omega\mu\widetilde{\boldsymbol{H}} \quad \textbf{3.6}$$

$$\nabla . \widetilde{H} = j\omega\epsilon_c\widetilde{\boldsymbol{E}} + J \quad \textbf{3.7}$$

$$\nabla . B = 0 \quad \textbf{3.8}$$

$$\nabla . D = q_v \quad \textbf{3.9}$$

Where:

- ω is the angular velocity
- ε is the emissivity
- μ is permeability
- q_v is the volume electric charge density.
- E is the electric field.
- B is the magnetic field.
- J is the current density.
- D is the electric displacement vector.
- H is the magnetic field intensity.

And the relationship equations are:

$$J = \tilde{\sigma} E \quad \textbf{3.10}$$

$$D = \tilde{\varepsilon} E \quad \textbf{3.11}$$

$$B = \tilde{\mu} H \quad \textbf{3.12}$$

In the above equations:

- $\tilde{\sigma}$ (siemens/metre) is the parameter governing the creation of an electric current in the presence of an electric field.
- $\tilde{\varepsilon}$ (unitless) is the Dielectric Permittivity which governs the amount of energy stored by the material in the presence of an electric field.
- The final parameter $\tilde{\mu}$ (Henries per metre), is the magnetic permeability, which describes how the material responds to a magnetic field.

These three parameters are vector quantities, but for GPR purposes are treated as field independent scalar values.

The following approach is one of many solutions taken from (Ulaby and Ravaioli, 2015) to these equations and is based on the following assumptions to create the model:

1. The permittivity does not vary in space. This allows the use of a mathematical identity that the curl of a vector is equal to the gradient of the divergence of a vector minus the Laplacian of that vector.
2. There is no free charge.
3. The material is not conductive (this will be addressed later in section 3.4).
4. The wave form is a plane wave. This assumption is valid for small distances from the source, however, to make the mathematics easier this will be applied.
5. The wave form is sinusoidal.
6. Scattering is negligible.

Assumption 1 gives rise to the following identity:

$$\nabla \times \nabla \times \vec{E} = \nabla\left(\nabla \cdot \vec{E}\right) - \nabla^2 \vec{E} \qquad \textbf{3.13}$$

And Gauss law from Maxwell's equations:

$$\nabla . D = \mu_v \qquad \textbf{3.14}$$

Which means that 3.13 can be rewritten as:

$$\nabla \times \nabla \times \vec{E} = -\nabla^2 \vec{E} \qquad \textbf{3.15}$$

Using Faraday's law (3.6), gives:

$$\nabla \times \frac{\delta \vec{B}}{\delta t} = -\nabla^2 \vec{E} \qquad \textbf{3.16}$$

Since assumption 1 has been made, and using 3.7:

$$-\frac{\delta}{\delta t}\mu\left(\sigma\vec{E}+\frac{\partial\vec{D}}{\partial t}\right)=\nabla^2\vec{E} \quad \textbf{3.17}$$

Assumption 3 means that σ=0 and using 3.11 means that:

$$-\frac{\delta}{\delta t}\mu\varepsilon\left(\frac{\partial\vec{E}}{\partial t}\right)=\nabla^2\vec{E} \quad \textbf{3.18}$$

And this can be arranged to the wave equation:

$$\nabla^2\vec{E}-\mu\varepsilon\left(\frac{\partial^2\vec{E}}{\partial t^2}\right)=0 \quad \textbf{3.19}$$

Assumptions 4 and 5 allow the following form for a circular polarised, sinusoidal wave to be used for the source (where k is the wave number):

$$\psi(x,t)=A\cos(\omega t-kx) \quad \textbf{3.20}$$

And in complex notation (ignoring the imaginary part):

$$\psi(x,t)=Ae^{j(\omega t-kx)} \quad \textbf{3.21}$$

This ultimately gives the following solution to Maxwell's equation for a plane sinusoidal waveform where z is the distance travelled in space:

$$\vec{E}(z,t)=\overrightarrow{E_0}e^{j(\omega t-kz)} \quad \textbf{3.22}$$

Where:

- $\omega t-kz$ is the representation of sinusoidal waves.

This was a very quick outline solution to Maxwell's equations. Equation 3.22 will be the basis for all the following work. This solution is acceptable if the wave is travelling through space, however when it is through a lossy medium such as soil, losses in the strength of the wave are inevitable. These loses are caused by the heterogeneous nature of the media which the electromagnetic (EM) wave propagates. Each of the heterogeneities that the wave encounters, in some way,

will extract energy from the EM wave and scatter it in multiple directions. It has been proven in (A.P., 2005) that the electric field will attenuate with a coefficient α_s and in distance travelled (r) as shown in 3.23 below.

$$E = E_0 e^{-\alpha_s r} \quad \textbf{3.23}$$

The scattering coefficient is derived using equation 3.24.

$$\alpha_s = \frac{NA}{2} \quad \textbf{3.24}$$

Where A is the scattering cross section and N is the number density, IE atoms or molecules per unit volume.

Other factors affecting the total attenuation of the signal include the scattering attenuation which is frequency dependent (equation 3.25) and must be added to the α_s to get the total attenuation as the wave travels through the heterogeneous lossy medium.

$$A = Ca^6 f^4 \quad \textbf{3.25}$$

Where C is a constant with units $1/m^4\ Hz^4$, a is the sphere radius and f is the frequency.

The total attenuation is the linear sum of the scattering and material attenuation value. The final source of attenuation is volumetric; however, it is only present in ice where the material attenuation is smaller than more common materials for GPR use cases such as soil and rock.

The final source of loss for an EM wave is the effect of relaxation. This is caused by the relationship between the permittivity of the wave and the propagation wave. When a wave interacts with molecules in a medium, energy is transferred to medium in the form of a “separating charge”. This in turn generates a small displacement

current. This current also produces localised EM energy, which acts a brake on the propagating wave if it is out of phase with the wave. In addition to this effect, should these local waves be free to move, such as in water, this movement converts some of the EM energy into heat. This effect is frequency dependent and is demonstrated by the frequency relaxation value, whereby below this threshold the effect is minimal, above this threshold this causes significant losses in the form of heat into the surrounding matrix. This is shown in Figure 3-1 below for an idealised damp lossy material.

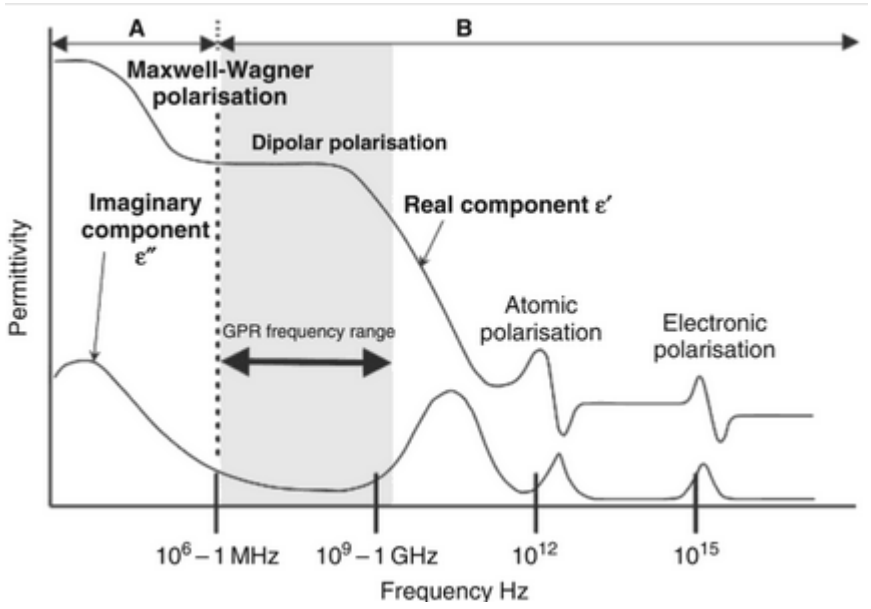


Figure 3-1 Permittivity Response of an Idealised Lossy Dielectric Media (Cassidy, 2009)**.**

It is possible now to describe an electromagnetic sinusoidal plane wave as it passes through the medium, the next thing to assess is the effect of the wave form at boundaries.

Expanding the ideas above into real materials as mentioned in (Cassidy, 2009) gives the equations for α which is the attenuation coefficient (Np/m) and β which is the phase coefficient (rad/m) (relaxation) in 3.26.

$$\alpha = \omega\sqrt{\mu\varepsilon'}\sqrt{\frac{1}{2}\left[\sqrt{1+\left(\frac{\sigma}{\omega\varepsilon'}\right)^2}-1\right]}$$

$$\beta = \omega\sqrt{\mu\varepsilon'}\sqrt{\frac{1}{2}\left[\sqrt{1+\left(\frac{\sigma}{\omega\varepsilon'}\right)^2}+1\right]} \quad \textbf{3.26}$$

Where:

ω is the angular frequency (rads/sec)

ε' is the absolute permittivity

σ is the conductivity

μ is the relative permeability

Incorporating the equations in 3.26 into the equations 3.22 gives the following equation by using an exponential decay term $e^{-\alpha z}$ with k being replaced by the term β so:

$$\vec{E}(z,t) = \overrightarrow{E_0}e^{-\alpha z}e^{j(\omega t-\beta z)} \quad \textbf{3.27}$$

Like any other type of waveform, when EM waves hit a boundary, they obey the laws of conservation of energy. Depending on the solidity of the boundary, some energy is transferred across the boundary, while the remaining energy is either absorbed or reflected at the boundary wave. The amount of energy absorbed is often negligible so this can be ignored. The amount of energy reflected and transferred is governed by Fresnel's Laws. The reflection portion is governed by Snell's law. If it is assumed that the waveform is polarized perpendicularly to the plane of incident, then the following holds true:

$$\overrightarrow{E_{r\perp}} = r_{\perp}E_{i\perp} \text{ and } \overrightarrow{E_{t\perp}} = t_{\perp}E_{i\perp} \quad \textbf{3.28}$$

Where $r_{\perp}$ and $t_{\perp}$ are related to the angle of incident (θ_i) the angle of transmission (θ_t), the index of refraction for both the initial medium (n_i) and the transmission medium (n_t) as follows.

$$r_{\perp} = \frac{n_i \cos\theta_i - n_t \cos\theta_t}{n_i \cos\theta_i + n_t \cos\theta_t} \text{ and } t_{\perp} = \frac{2n_i \cos\theta_i}{n_i \cos\theta_i + n_t \cos\theta_t} \quad \textbf{3.29}$$

It is possible for both $r_{\perp}$ and $t_{\perp}$ to be negative, which would mean a reversal in the phase of the wave occurred.

Using the equations in 3.23, 3.24, 3.25 and 3.29 it is possible to start constructing the forward model of a GPR wave propagating through a lossy medium and hitting a target. As stated above, the plane wave is an oversimplification as the wave form is in fact a circular wave and emits from a source in a manner like ripples in a pond and would form in accordance with equation 3.30.

$$u(\vec{r}, t) = \frac{A\cos\left(k\left(\left|\vec{r} - \vec{r}_j\right| - \omega t + \varphi_j\right)\right)}{\sqrt{\left|\vec{r} - \vec{r}_j\right|}} \quad \textbf{3.30}$$

Where:

- $u(\vec{r}, t)$ is the displacement of the surface from a datum
- A is the amplitude
- k is the wave number
- $\vec{r}$ is the position of the wave front
- $\vec{r}_j$ is the position of the wave source
- φ_j is the phase angle of the wave

The approach detailed above is a simplified solution to the forward model for an electromagnetic wave propagating through a solid medium such as soil. Should the

analytical based approach to the inverse problem be adopted, then the techniques mentioned above will form the core of any solution.

3.3 Simplified GPR Model

The most common technology used in subsurface imaging is that of GPR, as will be discussed in Chapter 4. In this section, the theory from the previous section will therefore be applied to the GPR scenario. Initially it is best to consider the case where an antenna is above the ground and interrogate the air/ground interface as shown in Figure 3-2 below.

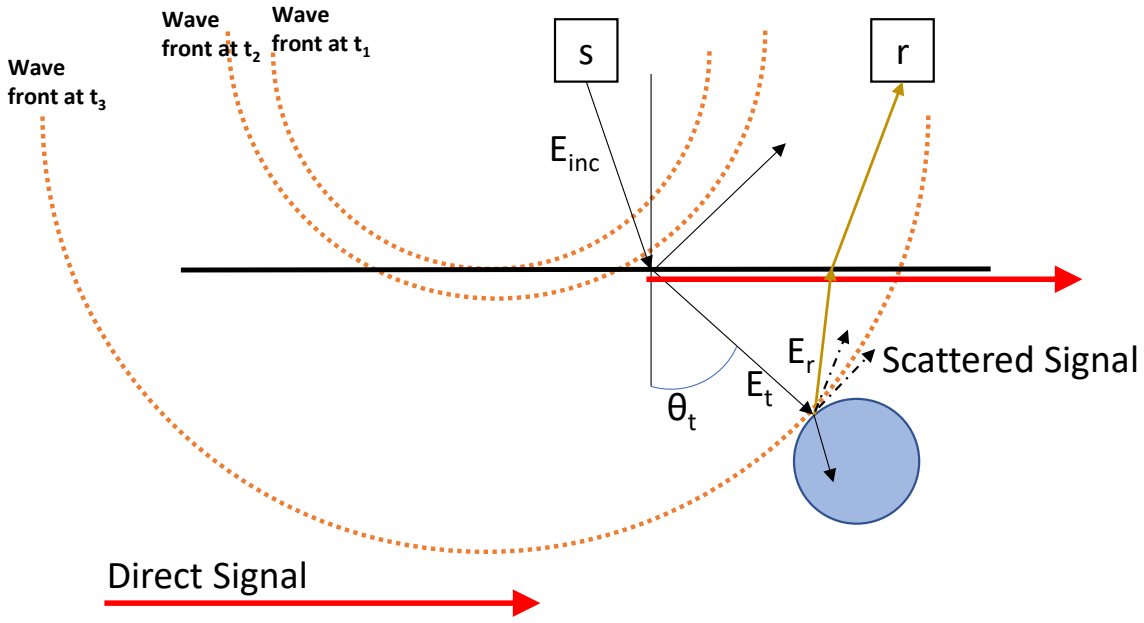


Figure 3-2 Diagram of the Propagation of an EM Wave Through a Medium with the Return Signal and Ground Interactions Labelled.

In the diagram above, "s" is the source and "r" is the receiver. A significant amount of energy is transmitted parallel to the ground, called the direct signal. The wave then propagates through the medium until it interacts with an object, at which point there is some reflection back towards the source, shown in brown dotted line in the Figure 3-2, as well as transmission and reflection into the body. Using Snell's law to find θ_t and equations 3.28 and 3.29 will allow a very simplistic method for calculating

the amount of energy that is reflected and the amount of energy transmitted into the medium.

The same approach can be adopted when the wave hits an object of interest. The wavefront in 2D is circular and the wave will continue propagating through a medium until it runs out of energy. Assuming that there are 3 locations that can monitor the wave represented by 3 discrete time points, t_1, t_2 & t_3 then it is possible to construct a forward map based on the equations in the previous section while maintaining the assumption that it is a plane wave as long as all wave paths and interactions are taken into account.

In the first instance (t_1), the angle of incidence is 0 and both the observed values at the other stations are 0. As the wave propagates so the values of the observed values increase while the values at the other stations decrease. This is a very simple solution and gives the following solutions for a plane boundary of infinite length and is given in 3.31.

$$\begin{aligned} E_1^r &= \Gamma E_1^i, \text{ where } \frac{\eta_2-\eta_1}{\eta_2+\eta_1} = \Gamma \\ E_2^r &= \Gamma E_2^i, \text{ where } \frac{\eta_2-\eta_1}{\eta_2+\eta_1} = \Gamma \\ E_3^r &= \Gamma E_3^i, \text{ where } \frac{\eta_2-\eta_1}{\eta_2+\eta_1} = \Gamma \end{aligned} \qquad \textbf{3.31}$$

Considering the case for oblique angles the refraction and transmission coefficients at a boundary are independent of the polarisation of the incoming wave. A wave form of arbitrary polarisation is described by using superposition back to the perpendicular and parallel polarisation models. The approach taken here is to decompose the wave into perpendicular and parallel polarised waves for both the transmitted and the reflected wave forms.

Figure 3-3 shows a simplified model of how a wave is reflected and refracted upon contact with a medium boundary. This diagram is then used to define the angles used in the following equations.

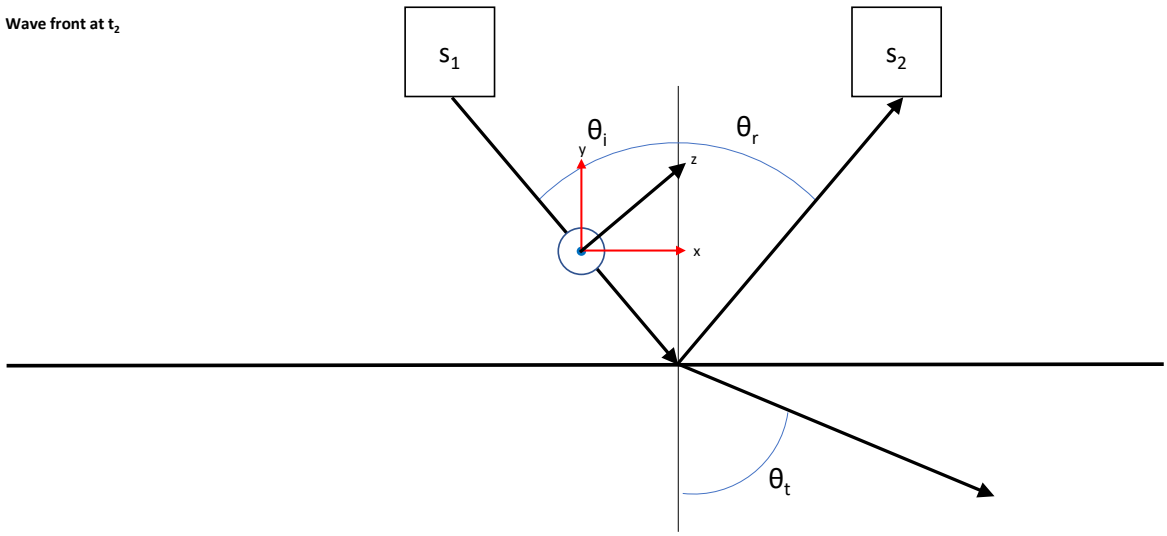


Figure 3-3 Diagram Showing the Reflected Plane Wave

$$x_i = x\,sin\,\theta_i + z\,cos\,\theta_i$$

$$\hat{\boldsymbol{y}}_i = -\hat{\boldsymbol{x}}\,cos\,\theta_i + \hat{\boldsymbol{z}}\,sin\,\theta_i \qquad \textbf{3.32}$$

When 3.32 is substituted into the general wave equation, this gives for the incident, reflected and transmitted wave the following:

$$\vec{E}_{\perp}^{i} = \hat{y}E_{\perp}^{i}e^{-jk(xsin\theta_i + ycos\theta_i)}$$

$$\vec{E}_{\perp}^{r} = \hat{y}E_{\perp}^{r}e^{-jk(xsin\theta_r - ycos\theta_r)} \qquad \textbf{3.33}$$

$$\vec{E}_{\perp}^{t} = \hat{y}E_{\perp}^{t}e^{-jk(xsin\theta_t + ycos\theta_t)}$$

Using Snell's equations, it is possible to reformat 3.33 into coefficients $\Gamma_{\perp}$ and $\tau_{\perp}$ as shown in 3.35. The coefficients in equation 3.35 are applied as scalar values to the wave equation.

$$\theta_i = sin^{-1}\left(\frac{r}{X}\right)\ and\ \sin\theta_t = \frac{n_i}{n_t}\sin\theta_i \qquad \textbf{3.34}$$

$$\Gamma_{\perp} = \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t}$$

$$\tau_{\perp} = \frac{2n_2 \cos\theta_i}{n_2 \cos\theta_i + n_1 \cos\theta_t} E_{i\perp} \quad \textbf{3.35}$$

Putting everything together for the reflected waves, the following hold true:

- S_{ii} is given by perpendicular incident case $E_1^r = \Gamma E_1^i$, where $\frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} = \Gamma$
- S_{ij} and S_{ij} are given by the case $\vec{E}_{\perp}^r = \hat{y} E_{\perp}^r e^{-\alpha z} e^{-jk(x sin\theta_r - y cos\theta_r)}$

While the transmitted waves are given by:

- S_{ii} is given by perpendicular incident case $\tau = 1 + \Gamma$
- S_{ij} and S_{ij} is given by the case $\vec{E}_{\perp}^t = \hat{y} E_{\perp}^t e^{-jk(x sin\theta_t + y cos\theta_t)}$

Similar equations can be derived for the magnetic field.

Where X is the distance between the points, r is the distance from the source to the point of interaction and n_i and n_t are the index of refraction of the two mediums either side of the boundary.

Now that Maxwell's equations have been applied to GPR, we will move to the next step, introducing real-world complexities to the mathematical solution.

3.4 Realistic Forward Model

The approach adopted in section 3.3 is intentionally very simplistic. However, in the real-world soil is not homogenous, and as a result scattering occurs when a wave interacts with non-uniformities. The effects of scattering have been a source of study in physics over the years and several approaches are available in literature. The maths can be complicated depending on the nature of the medium and the geometry of the boundary. This section demonstrates some of the complexities and offers resolutions.

The most widely adopted approaches for the creation of the forward model is the Lippmann-Schwinger equation (Lippman and Schwinger J, 1950) (3.36) which describes the wave–scatter interaction in the 2D domain of interest (D). The second equation describes the scattered field in terms 3.36 of the induced contrast current (3.37) taken at the surface (S).

$$E^t(\boldsymbol{r}) = E^i(\boldsymbol{r}) + i\omega\mu_0 \int_D^{\square} g(\boldsymbol{r},\boldsymbol{r}')[-i\omega\epsilon_0\,(\epsilon_r(\boldsymbol{r}') - 1)E^t(\boldsymbol{r}')d\boldsymbol{r}' \text{ for } \boldsymbol{r} \in D \qquad \textbf{3.36}$$

$$E^s(\boldsymbol{r}) = i\omega\mu_0 \int_D^{\square} g(\boldsymbol{r},\boldsymbol{r}')[-i\omega\epsilon_0\,(\epsilon_r(\boldsymbol{r}') - 1)E^t(\boldsymbol{r}')d\boldsymbol{r}' \text{ for } \boldsymbol{r} \in S \qquad \textbf{3.37}$$

These two equations form the cornerstone to the remainder of this section, although in slightly different formats.

The best solution to 3.36, is the Born approximation (Born, 1926), which takes the form of equation 3.38:

$$E^s(x_s,\omega) = k_s^2 \int_D^{\square} G(x_0,\omega,x',z')\,E_{inc}(x_s,x',z';\omega)\chi_e(x',z')dx'dz' \qquad \textbf{3.38}$$

Where:

- k_s is the wave number.
- G is the Green's Function (see section 3.4.1).
- E_{inc} is the incident wave (see section 3.4.2).
- X_e is the permittivity (see section 3.5).

3.4.1 Green's Functions

Green's functions are a method for solving ordinary differential equations with initial value conditions. They can also be used in the solution of partial differential equations with known boundary conditions. In (Persico, 2014), the rather intense algebraic derivation of a set of Green's functions for above and below the surface have already been derived, and these will be used in this section (equations 3.39 &

3.40). It is worth noting that the derivation of the equations is performed in the Fourier domain, and at the end the inverse Fourier transform is taken.

$$G_i(x,x',z,z') = \frac{-j}{4\pi}\int_{-\infty}^{\infty} \frac{\exp\left(-ju(x'-x)\right)}{k_{zs}}\Big[\exp(-jk_{zs}|z-z'|) + \frac{\mu_0 k_{zs}-\mu_s k_{z0}}{\mu_0 k_{zs}+\mu_s k_{z0}}\exp\left(-jk_{zs}(z'+z)\right)\Big]du\ ,\ (x,z)\ (x',z') \in D \quad \textbf{3.39}$$

$$G_e(x_0,x'z_0,z') = \frac{-j\mu_0}{2\pi}\int_{-\infty}^{\infty} \frac{\exp\left(-jk_{zs}z'\right)\exp(jk_{z0}z)\exp\left(-ju(x'-x_0)\right)}{\mu_0 k_{zs}+\mu_s k_{z0}}du, \quad (x_0,z_0)\ \in \Sigma\ ,\qquad (x',z') \in D \quad \textbf{3.40}$$

Where:

- $(x_0, z_0) \in \Sigma$ is the observation point.
- $k_{zs} = \sqrt{k_s^2 - u^2}$ where $k_s^2$is the wave number of the propagation medium
- $k_{z0} = \sqrt{k_o^2 - u^2}$ where $k_o^2$is the wave number at the origin
- z' depth
- x' lateral distance
- x_0 origin location
- μ_0 is the above ground permeability
- μ_s is the below ground permeability

3.4.2 Incident Field

The incident field is critical in this solution, and again in (Persico, 2014) the significant derivation of the incident fields (equations 3.41 and 3.42) have been performed and are going to be used in this section. In the derivation, the impulsive source in air has been considered. The equations can be further enhanced by

inserting into them a characteristic equation of a source more similar to the one used in the real world.

$$E_{inc}(x,z;\omega) = \frac{-j\omega I_0 \mu_0}{4\pi}\int_{-\infty}^{\infty} \frac{\exp\left(-jv(x_s - x)\right)}{k_{z0}(v)}\Big[\exp(-jk_{z0}(v)|z - z_s|) + \frac{\mu_s k_{z0}(v) - \mu_0 k_{zs}(v)}{\mu_0 k_{zs}(v) + \mu_s k_{z0}(v)}\exp\left(-jk_{zs}(v)(z' + z_s)\right)\Big]dv\,,\ 0 < z \quad \textbf{3.41}$$

$$E_{inc}(x,z;\omega) = \frac{-j\omega I_0 \mu_0 \mu_s}{2\pi}\int_{-\infty}^{\infty} \frac{\exp\left(-jk_{zs}(v)z\right)\exp(jk_{z0}(v)z_s)\exp\left(-jv(x_s - x)\right)}{\mu_0 k_{zs}(v) + \mu_s k_{z0}(v)}dv, \quad z < 0 \quad \textbf{3.42}$$

Where:

- $I_0$is the plane wave of a filamentary current (Persico, 2014)

In the previous few sections, the forward models have been discussed, first in terms of a simplistic solution to the Maxwell's equations and then an in-depth review into a more realistic solution for the forward model that is going to be used in the remainder of this chapter.

3.5 Distorted Born Iterative Method (DBIM)

One of the most commonly used numerical approaches used in inverse modelling is the DBIM approach as discussed in section 2.2.1.

Several approaches are available to solving the Born approximation: Born Approximation Inversion Method, Rytov Approximation Inversion Method, Extended Born Approximation Inversion Method are all useful when using weak scatters. Here, an iterative solution to the Distorted Born Iterative Method will be used as proposed in (Chew and Wang, 1990).

The solution iteratively applies the distorted Born wave approximation and starts with equations 3.36 and 3.37. The equations are simplified using Green's functions for the surface and for the domain of interest and yield equation 3.43.

$$E^s = G_S \chi (I - G_D x)^{-1} E^i \quad \textbf{3.43}$$

Where:

- G_D will be below the surface in the forward problem, in the inverse problem it is the location where the scatterer sits.
- G_S will be the surface on which the receivers exist, in this case the area above ground.
- χ is the contrast
- E^i incident wave

The scatter field consists of two parts, the first is the background scatterer χ_0 and the second part is the perturbation field due to the presence of an inhomogeneous medium (δX). The next step is to follow the approach listed in the previous text:

1. Define an initial value for χ.
2. Then calculate the G_s for χ using:

$$G_s^{\chi}(J) = k_0^2 \int_D g^{\chi}(r, r') J(r)\, dr \quad \textbf{3.44}$$

Solve the forward problem for each incident wave using the single step Born approximation.

$$E^t = E^i(r) + G_D \chi E^t \quad \textbf{3.45}$$

The next thing is to assume that the object creates a small perturbation $\delta\chi$ and solve the following equation for all incidences. Regularization will be required due to the nonlinearity the problem.

$$E^s = E^s + G_S \delta\chi E^t \quad \textbf{3.46}$$

Update the solution for $\chi = \chi + \delta\chi$

3. Check the error and return to step 2 or terminate.

The DBIM method is a well-trodden path for the solution of the inverse problem and is reliant on several key assumptions, all of which will affect the outcome of the solution. These include the quality of the Green's functions, the amount of noise and the ability of the optimiser to avoid local minima. In addition, there is a great deal of computer hardware investment in the solution of the optimization.

3.5.1 Theoretical solution of the Born Approximation

This section creates the solution to the inverse problem for the case shown in Figure 3-4 below. Following the path of the wave forward, that is from the source to the object, two equations are in effect. The first covers the transmitted portion across a boundary (E_t) and the second is the reflected or scattered portion (E_r). Both values are related to the incident wave form.

$$E_{inc} = E_t + E_r \quad \textbf{3.47}$$

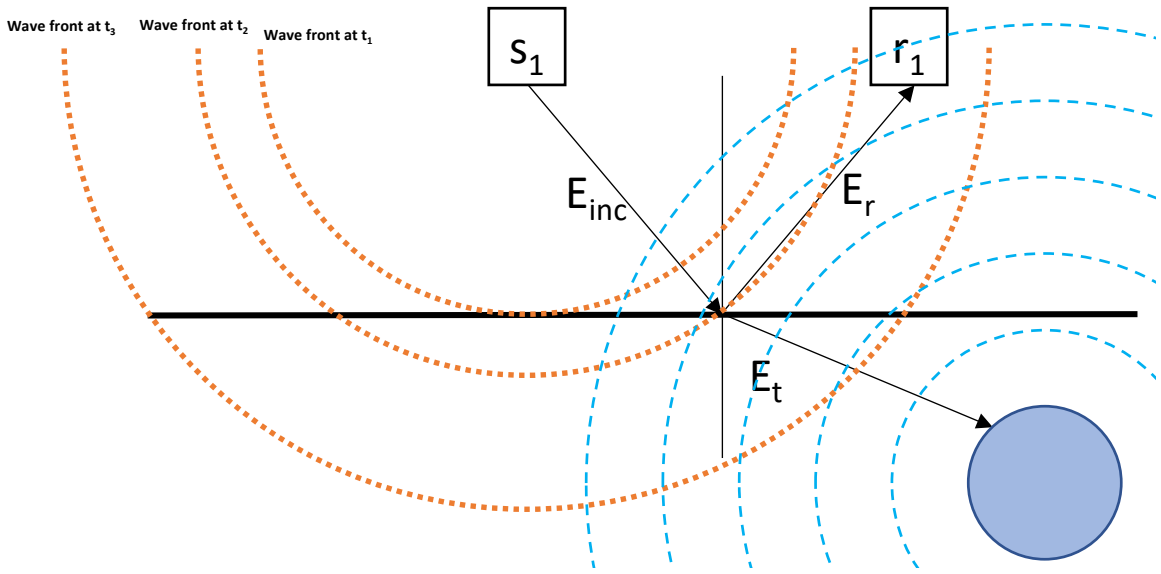


Figure 3-4 Diagram of the 2D GPR Problem Setup

In the inverse problem, the aim is to derive the dielectric constants from the scattered field measured at the antenna across the domain of interest. As will be explained in section 3.5.2, the actual solution is highly nonlinear due to the amount of noise in the system, and therefore the solution approach is to minimise the error in the dielectric constant matrix.

The traditional solutions to the DBIM are derived for a situation where there are multiple reception surfaces around the domain of interest as shown in Figure 3-5. In this picture the object of interest is shown in yellow, with the domain of interest (D) in blue. The surfaces over which the receiving points exist (S).

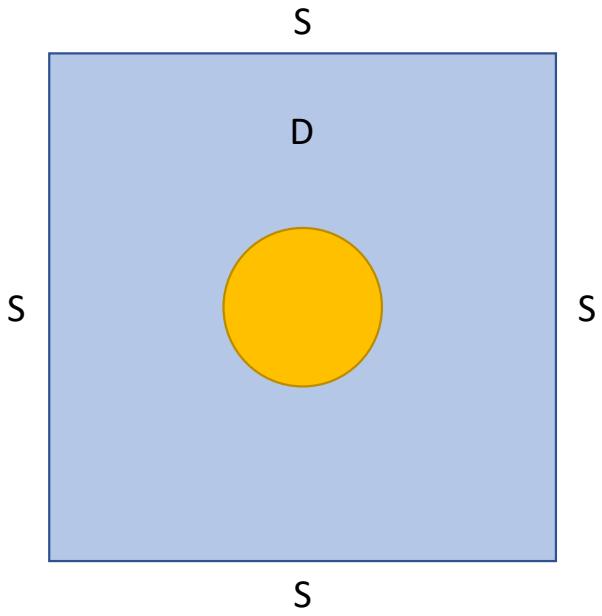


Figure 3-5 Traditional DBIM Model Setup

In the traditional GPR model there are in effect two Surfaces and two Domains (labelled S1 and S2 and D1 an D2 in Figure 3-6). The first domain is where the receiver is located and the second is the Domain of interest. The impact of this requirement is that the forward problem becomes more complicated and as a result there is a requirement for two separate Green's functions: one on the surface and one inside the domain of interest (denoted as G_i and G_e in equations 3.39 and 3.40). The transmitted wave considers the Domain properties, while the scattered field considers the contrast of the Surface and Domain properties.

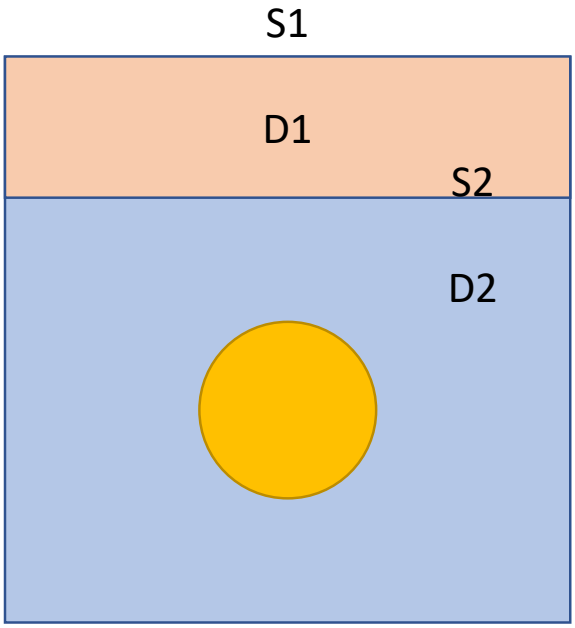


Figure 3-6 GPR DBIM Model Setup

This means that in the DBIM, there are now two values of Z to find (one for the air and one for the soil), suddenly the solution has got even more complicated.

As highlighted, the Green's functions (3.39 and 3.40) are in the Fourier domain It is therefore possible to use an empirical inverse Fourier transform, such as the Inverse Fast Fourier Transform to convert the solution to the Green's functions back into the geometry base, as long as the Fourier equation is solved using the correct values for u, which in this case will be taken as X^{-1}.

Since the transmitter and receiver points move in unison, the incident field is only calculated at a single point in space and again these functions (3.41 and 3.42) are in the Fourier domain. Using the same approach as for the Green's function, it is possible to convert the incident wave back into the geometry base by adopting the transform for v as Z^{-1}.

The wave numbers are calculated using the following equations.

$$k_0 = \frac{2\pi}{\lambda} \text{ where } \lambda \text{ is the wave number}$$

$$k_s = k_0\sqrt{\mu_s \varepsilon_s}$$

$$k_{z0} = \sqrt{(k_0 - u^2)} \qquad \textbf{3.48}$$

$$k_{zs} = \sqrt{(k_s - u^2)}$$

In 3.48 $\mu_s \; and \; \varepsilon_s$ are known dielectric properties (magnetic permeability and permittivity) of the background medium. The magnetic properties are considered to be uniform and negligible, however in most real-world cases this is not the case. The approach adopted to solve this equation is to discretise the domain of interest using the following sign convention shown in Figure 3-7. Here, anything above the Air/Ground boundary is considered as Z<0, and below this Z>0, with X_0 and Z_0 being the top right corner of the domain of interest, thus x increases from left to right.

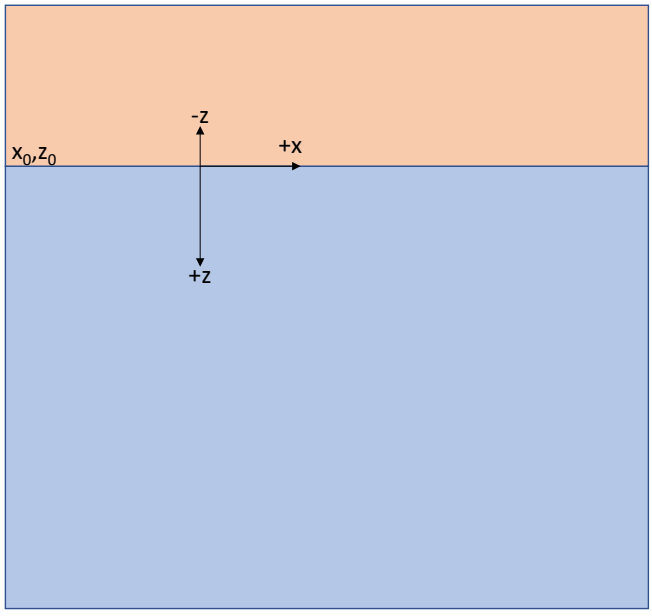


Figure 3-7 Sign Convention Used in Setting up the Inverse Problem

Using a combination of equations 3.39, 3.40 and 3.49 it is possible to solve for the Green's function, while 3.41, 3.42 and 3.49 can be used to solve the incident field equations. The next stage is to take all these equations and use the DBIM model to try and find a value for X.

In order to use the DBIM it is necessary to calculate G_S^{χ} for each of the iterations and for each incident wave the forward problem needs to be solved to calculate the total field across 'D' as well as the scattered field (step 2 in the DBIM algorithm). From this step changes in X can be found, and hence X can be updated.

3.5.2 Known Issues with the Inverse Scattering Problem

This section will expand on the main characteristics of the DBIM approach, highlighting the reasons that the mathematical theory described in this chapter is difficult to employ in a real-world scenario due to the iterative approach.

It has been demonstrated in (Jin *et al.*, 2017) that the inverse scattering problem is ill-posed. This means that the solution does not depend continuously on the data. As a result, regularisation techniques are often implemented. Another complication in the solution of inverse problems is nonlinearity. Nonlinearity in the solution of an inverse problem can often lead to false results due to the presence of local minima in the convergence process. To minimise this obstacle, the solution can be started from several points and the results compared.

In order to improve the empirical solution accuracy of the inverse problem the optimisation has been modified to use a gradient descent approach, as this involves a momentum term which "carries" the solution over some local minima. The nonlinearity of the solution (Yu, Yuan and Liu, 2009) is caused due to the impurities contained within the soil and hence the internal field changes.

Based on the papers in section 2.2, it is applicable to use the Born approximation in the numerical solution of the inverse scattering problem. The Born approximation is based on expressing the total wave field as the sum of the incident field plus a small

perturbation. There are several limitations to this approach, chiefly that it is only valid when the scattered field is significantly weaker than the incident field.

In order to solve the Born approximation, it is often coupled with the use of Green's functions. As shown in section 3.4, these are a method of solving the differential equations by introducing a function which allows the prediction of what is happening inside a system based on the boundary values (and is often written as $g(\boldsymbol{r},\boldsymbol{r}')$). In general terms, the Born series is defined in 3.49.

$$|\psi\rangle = |\phi\rangle + [G_0(E)V]|\phi\rangle + [G_0(E)V]^2|\phi\rangle + \ldots [G_0(E)V]^n|\phi\rangle \quad \textbf{3.49}$$

In the above equation, G_0 is the Green's function, ψ is the total wave field and V is the interaction potential. Applying the Born series in 3.49 to the inverse problem takes the form in equation 3.50, with k_s being the spatial circular frequency.

$$E_{(1)} = E_{inc}$$

$$E_{(2)} = E_{inc} + k_s^2 \iint_{-1\,0}^{1\,1} G_i E_{(1)} \bar{\chi}_e dx' dz' \quad \textbf{3.50}$$

$$E_{(n)} = E_{inc} + k_s^2 \iint_{-1\,0}^{1\,1} G_i E_{(n-1)} \bar{\chi}_e dx' dz'$$

The Green's functions in 3.50 need to be derived based on the response of the receiving antenna which are a function of the reflected wave, and since the total wave energy is known, the quantity of the transmitted wave across the first boundary can be derived, which then means that for an object in a medium E_{inc} is the E_T. The major issue with including the antenna is that this has the effect of pushing the solution from the 2D domain into a 3D domain as the antenna has current flow in the third dimension.

3.6 Chapter Summary

This chapter discussed the overall theoretical approach to solving a nonlinear inverse problem. The chapter explained in detail what inverse problems are and how they are generally solved by using a mixture of forward problems and regularization to iterate the solution until the error is minimised.

The next step was to show the general solution to Maxwell's equations which are fundamental in understanding how GPR, or any radar system, work and how with the help of Snell's equations they can be applied to an idealised forward model for a GPR system. This simplified model was then built on to create a more representative case for a useable forward model, which can predict the behaviour of the GPR.

The final sections built on all the knowledge gained in the previous sections to use existing solutions to the Born scattering approximation, the DBIM model, and then to show the approach adopted in a solution of this model.

The Inverse model is not being pursued as a method because it requires a significant amount of processing resource to solve each problem. Therefore, it would be unrealistic to use when mapping into the 3D domain. However, the examination and inclusion of the mathematical model and its application to GPR provide the theoretical background for the data-driven solution.

The next chapter will discuss how GPR systems work and highlight some of the problems associated with the technology when used for scanning potatoes underground. Chapter 5 will then provide a walkthrough of the data-driven solution to this problem.

4 GPR Hardware, Simulation Validation and Data Collection

This chapter is split into 3 sections building on the GPR physics discussed in the inverse problem description presented in chapter 3.

Section 4.1 describes how the theory presented in Chapter 3 is applied to GPR Systems and some common issues with this application.

Section 4.2 is a review of the commercial GPR system used in this project manufactured by GSSI (GSSI, 2022) which, when compared to other systems, not only proved to be the most reliable but also has validated antenna models in the simulation software used.

Finally, section 4.3 details the rig testing performed for the purpose of this project in an attempt to validate the simulated signals to the real-world signals.

Section 4.4 reviews the numerical simulation code used to solve the equations governing the real-world physics of GPR. As mentioned in section 1.1.3, the main paradox arising from trying to correlate the GPR signals to the real-world data is the need to disturb the tubers during the growing phase to validate the accuracy of the signals generated, thus rendering the plant unusable in the future. To get around this, the use of simulation code has been chosen as a way of generating the ground truth data for the deep learning data. Examples of other research that has been performed to validate the GPRMax results against actual test data are included in section 4.4.1.

4.1 Introduction to GPR

This section will describe how the theoretical properties of Maxwells equations described in chapter 3 are applied to real world application of a GPR system.

Depending on the relationship between energy loss and energy storage ($\tilde{\sigma}$ and $\tilde{\varepsilon}$) an object will either diffuse or propagate Electro-Magnetic (EM) waves and the waves amplitude will decay with time and distance. An in-depth review of GPR Physics is made in section 3.2 and its application to the inverse problem solution.

The next key property is how the waveform changes across a boundary. The GPR approach used in this study is looking at reflections as opposed to direct transmission between emitter and receiver. Direct transmission occurs where the receiver and transmitter are both buried in separate boreholes at a predefined distance apart. The signal at the receiver is degraded by the transmitted medium and objects in the ground, as opposed to the signal being reflected by objects in the medium.

The behaviour of EM waves across the boundary is governed by two factors. The first factor is the Fresnel reflection coefficients, which dictate how the magnitude of an EM wave varies across a boundary of two materials. The second factor is Snell's law which shows how the waveform's direction of travel is changed as detailed in section 3.3.

The penultimate properties of interest are how well objects can be detected and what information can be recovered, as captured by the resolution. The resolution of a GPR system consists of two parts: depth and lateral resolution. For the depth resolution, if two responses are received simultaneously, they will be determined to be the same and be combined. However, if the responses overlap, then they are

only considered separate if they overlap by more than half the width of the waveform at 50% of its amplitude as shown in Figure 4-1Figure ‘A’. The amount of dispersion and attenuation of the pulse will also affect the depth resolution. Lateral resolution is governed by the velocity, pulse width and the distance between the object and the system. In effect the lateral resolution can be described as a function of the wavelength and the distance.

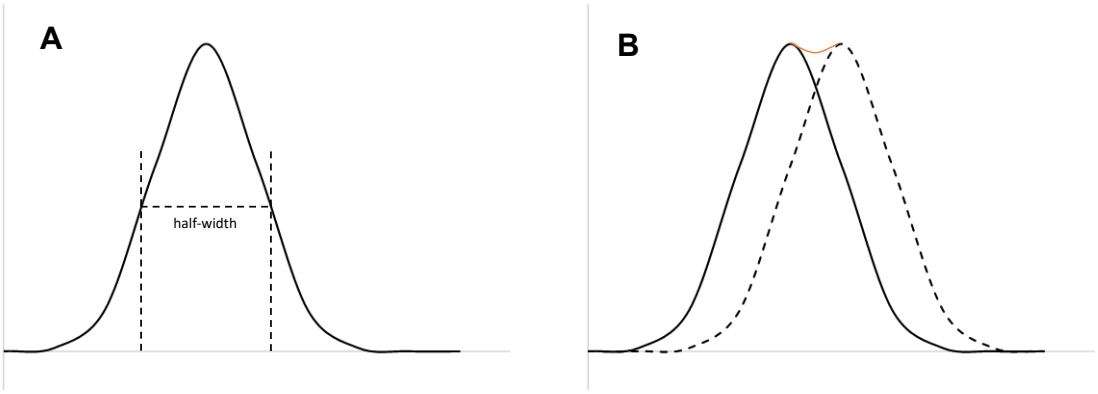


Figure 4-1Figure ‘A’ Describes the Half Width and Figure ‘B’ Demonstrates how Signals may Combine

The final parameter of interest is scattering. The signal strength decreases as a function of distance, and the rate of decay is governed by impurities. Therefore, while a wave propagates through a heterogenous medium such as soil, the wave form loses energy as it encounters impurities because the signal strength decreases as a function of distance, the rate of decay is governed by impurities. The scattering attenuation is dependent on the frequency of the signal and is measured using Rayleigh’s formula (equation 4.1) (Annan, A.P., 2009), where C is a constant with units $1/m^4Hz^4$ a is the sphere radius and f is the frequency. Once the scattering attenuation and the resistance scattering are known, these are simply added together to calculate the total scattering loss value.

$$A = Ca^6 f^4 \quad \text{4.1}$$

Taking all these parameters into consideration allows the accurate prediction of the behaviour of an EM wave as it propagates to, from, and through an object below the surface. This will be discussed in the next section.

4.2 GSSI GPR System

Despite including the same components, each commercial hardware system has a different approach for processing the data, and what data it stores and displays to the user. At the start of this project, experiments were conducted with three different hardware systems. However, as the project progressed, it became clear that the best results we obtained using GSSI Structure Scan Pro, which will be discussed in section 4.2.1. The data driven solution was then developed using the GPRMax simulation of this GRP system.

4.2.1 GSSI Structure Scan Pro

The Structure Scan Pro System comprises of the scanning unit and the SIR4000 data logger and signal processing unit, as shown in Figure 4-2. In addition to the items shown, a signal generator creates the pulses for the system to function.

Figure 4-2 GSSI Structure Scan Pro Unit (GSSI, 2022)

A second, horn antenna, was supplied from GSSI which are often used in remote sensing applications (not ground coupled). No GPRMax model exists for it, but it would be possible to create a model if required. This was deemed out of scope for this project. The main advantage of a horn antenna is that it allows the system to be mounted at a distance from the ground, which in this application would allow the unit to be placed above the plant canopy thus minimising damage during the growing phase. Future development of this work could focus on this aspect if desirable.

The GPRMax software offers two models based on commercially available antennas. While these models are not exact replicas of the commercially available antennas due to intellectual property constraints, they have been validated so that they behave in a manner that is consistent with the commercial antennas (Warren and Giannopoulos, 2011) and (Giannakis, Giannopoulos and Warren, 2019). It was decided that the antennas which would be used are the 1.5GHz bowtie palm antenna from GSSI, which is one of the models available in GPRMax. While it is

acknowledged that the antennas are different in power rating, after discussions with the UK distributor for GSSI equipment, it is expected that the frequency pulse will be the same, and since the frequency drives the antenna, the antenna dimensions should be the same assuming that the antennas are both straight bowties and not a derivative of the bowtie, for example paddle or rounded.

The simulated results for a traverse over 3 hollow PEC spheres are shown in Figure 4-3 and in the field are shown in Figure 4-4. When a GPR signal interacts with a metallic object, 100% of the signal is reflected back, hence there is only weak responses from the subsurface materials and instead the output is dominated by echo effects from the metallic target, which is known as ringing, as shown in Figure 4-3.

This is in part due to the additional, unknown signal processing adopted by the system. Eventually, the model is predicting the antenna response, while the commercial display is processing that signal. As with all signal processing activities, the readability of the output is always at the detriment of the resolution, which has been the toughest hurdle to overcome in this package of work.

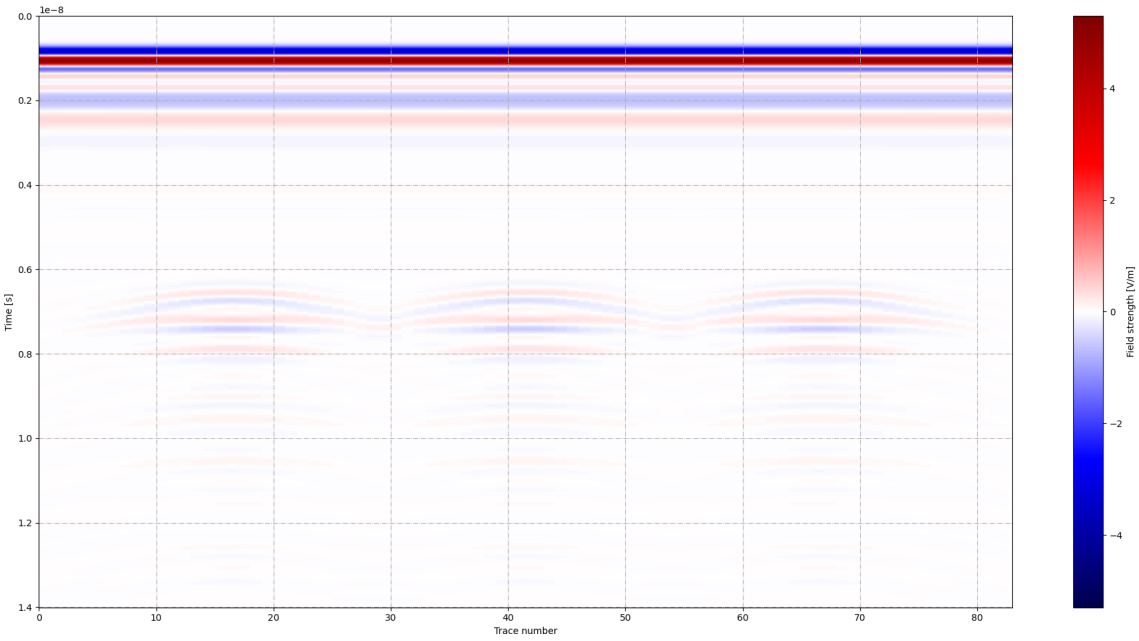


Figure 4-3 3 Hollow Spheres in a Lossy Medium – GSSI Antenna Simulation

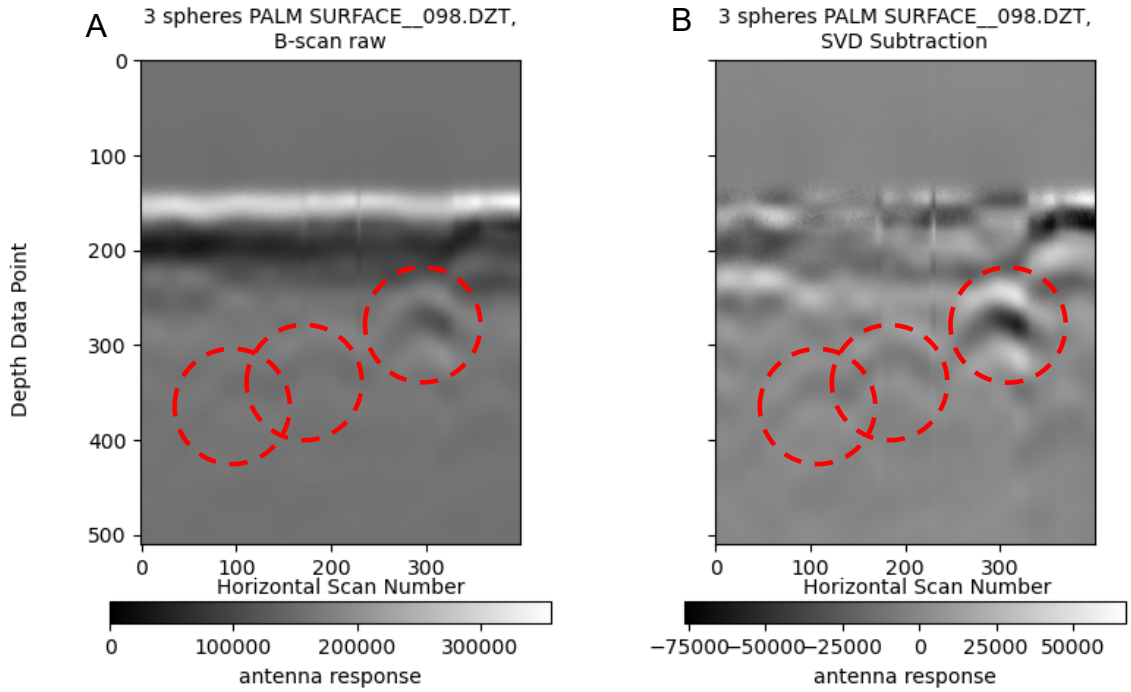


Figure 4-4 3 Image 'A' Shows Hollow Spheres (circled in red) in a Lossy Medium Image 'B' Shows Hollow Sphere After SVD is Applied – GSSI Antenna Real World (B-Hive and Lewis, 2021)

4.2.2 GSSI performance in the real world

The GSSI antenna was taken to the tuber field after the field had been desiccated (the removal of the foliage to prevent additional tuber growth prior to harvest). It is at this point that the tubers are hardened prior to harvesting.

Figure 4-5 shows the effects of different processing techniques aimed at bringing out the scan details of interest, for example individual parabolic curves which represent an object of interest below the surface. The options selected were:

1. Raw data + linear time varying gain
2. Normalised + linear time varying gain
3. 25x25 Median filter + linear time varying gain
4. 9x9 Median filter + linear time varying gain
5. Mean value removal + linear time varying gain
6. SVD + linear time varying gain

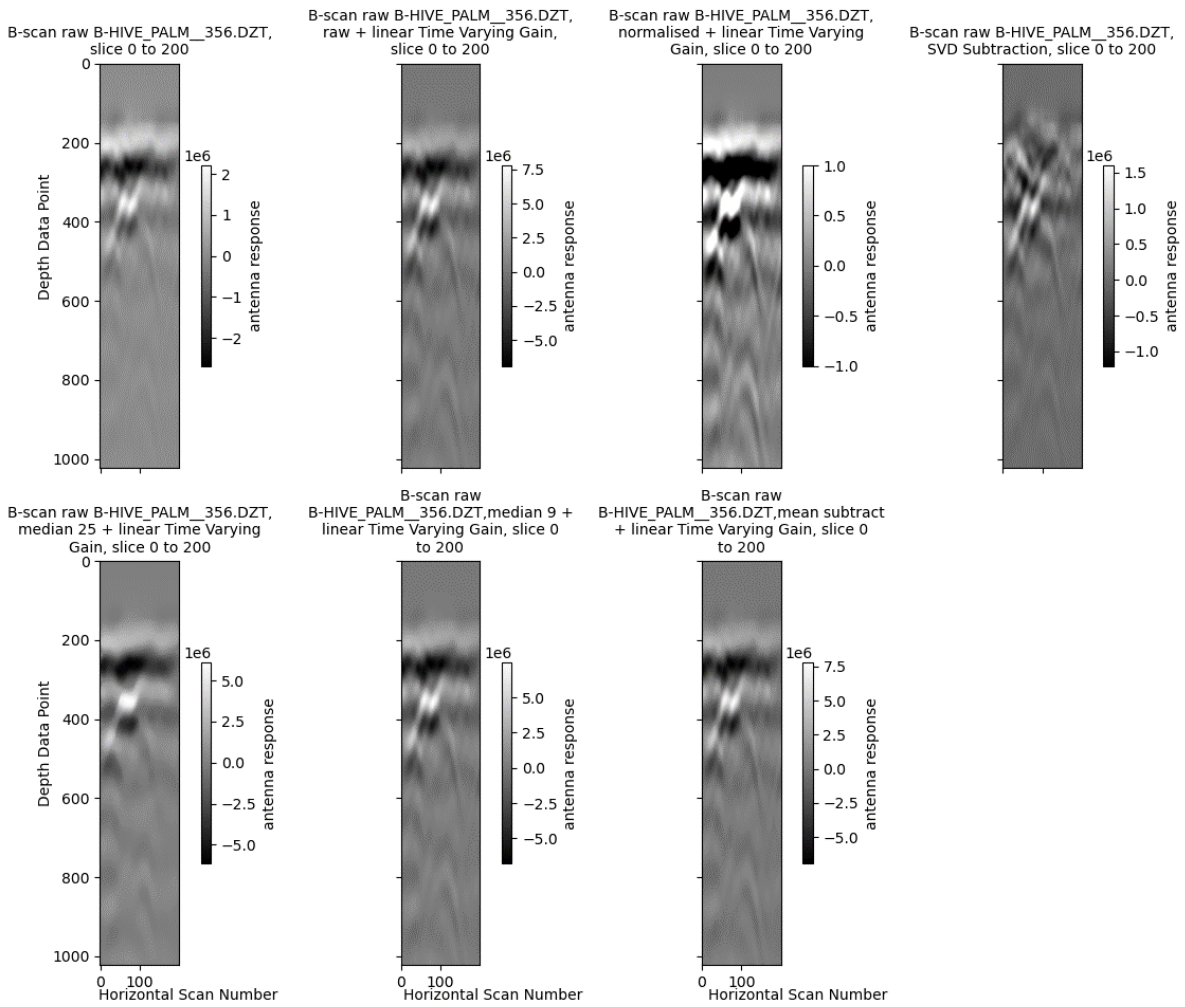


Figure 4-5 GSSI Scan of Bulk Sites of Tubers (B-Hive and Lewis, 2021)

A linear time varying gain was added across all the plots as a relatively simple method of counteracting the dissipation of the radar wave as discussed in section 3.2, other more complicated approaches are available. The data was normalised to a mean value of 0 and with a standard deviation of 0.5, the data was then clipped to [-1,1]. The two sizes of median filter were tested to see if there is any benefit in reducing noise. If this demonstrated success, then it would be further explored with either a band pass filter or a combination of low and high pass filters. Another simple method of trying to reduce the background noise in the signal is to take the mean of the signal and then subtract it from every data point within the signal. The final option was to perform SVD. When looking at the options listed above across the supplied

scans from B-Hive, it was clear that the SVD method results in the clearest signal, which can be used in the trained network, which is described in section 3.1.3.

4.3 Rig Test

While providing useful data, the work performed on all the antennas in the field has shown that there is a need to create a controllable space, which will allow repeatable experiments to be performed to assess the capabilities of the antennas.

The GPR hardware used in this section is as described in section 4.2.

4.3.1 Design

Several considerations were taken when designing the rig including:

- The need to create a space which has sufficient depth that the interactions of the floor can be removed.
- Due to the cost of lining the rig space with radar absorbing material (RAM), the scanning space must be sufficiently large to reduce the boundary effects of the sidewalls.
- Soil selection - topsoil was chosen as it tends to have very small levels of contaminants like rocks, organic debris, clay and sand and is more homogenous than other types of soil.
- The rig therefore had to be strong enough to contain the volume of topsoil.
- The topsoil medium needs to be as homogenous as possible with no significantly large contaminants (rocks, clay, sand, etc). While this is unrepresentative of the real world, it removes noise from the scans and would generate a cleaner signal than testing in the real world where there may be pockets of water and the general inhomogeneity in the

environment. However, for this very reason, it is useful for the proof of concept.

Since the antenna has a maximum scanning depth of 0.7 metres, the depth of the medium was set to 0.9 metres, to minimise the side wall interactions. The width and length were both set to 1.4 metres. To keep the costs as low as possible, the sidewalls were made from 9mm MDF board and screwed onto a frame made from 70mm square posts. The rig is shown in Figure 4-6 with dimensions in Figure 4-7 and a diagram of the subframe in Figure 4-8.

Figure 4-6 Showing the Final Rig with Topsoil in (Before Levelling)

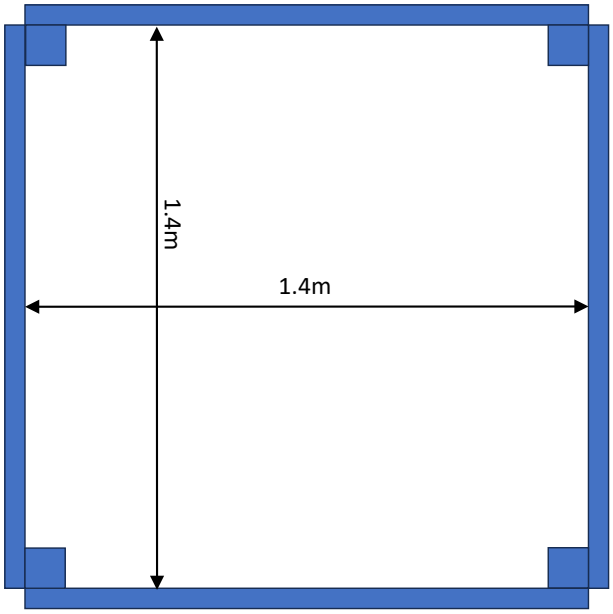


Figure 4-7 Basic Dimensions of the Test Rig

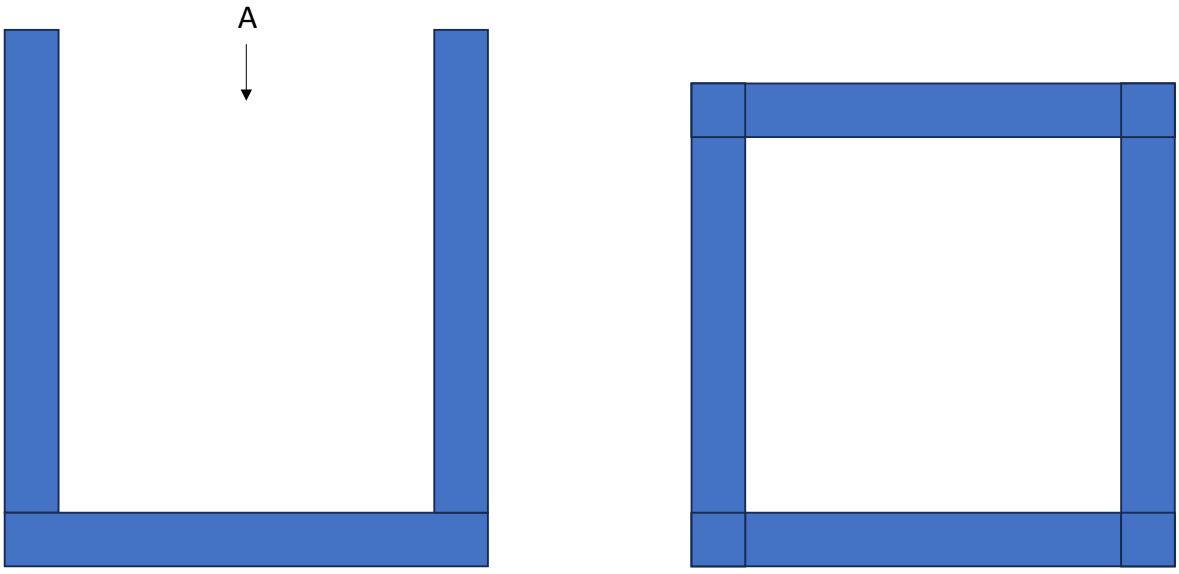


Figure 4-8 Image Showing the Subframe with a View on 'A' to the Right

To facilitate the smooth running of the scanner unit across the top of the rig, a sheet of Perspex (of depth 2mm) was placed on top the soil before each scan (as shown in Figure 4-9). Perspex was used because it has a negligible EM footprint, and therefore would not affect the results. The orientation of the antenna is shown in Figure 4-10 As the antenna traversed across the top of the soil readings were taken every 20mm, which was the smallest distance resolution before the antenna would

beep to warn of reading failure. The depth resolution was set to 4096 with a sampling frequency of 19kHz.

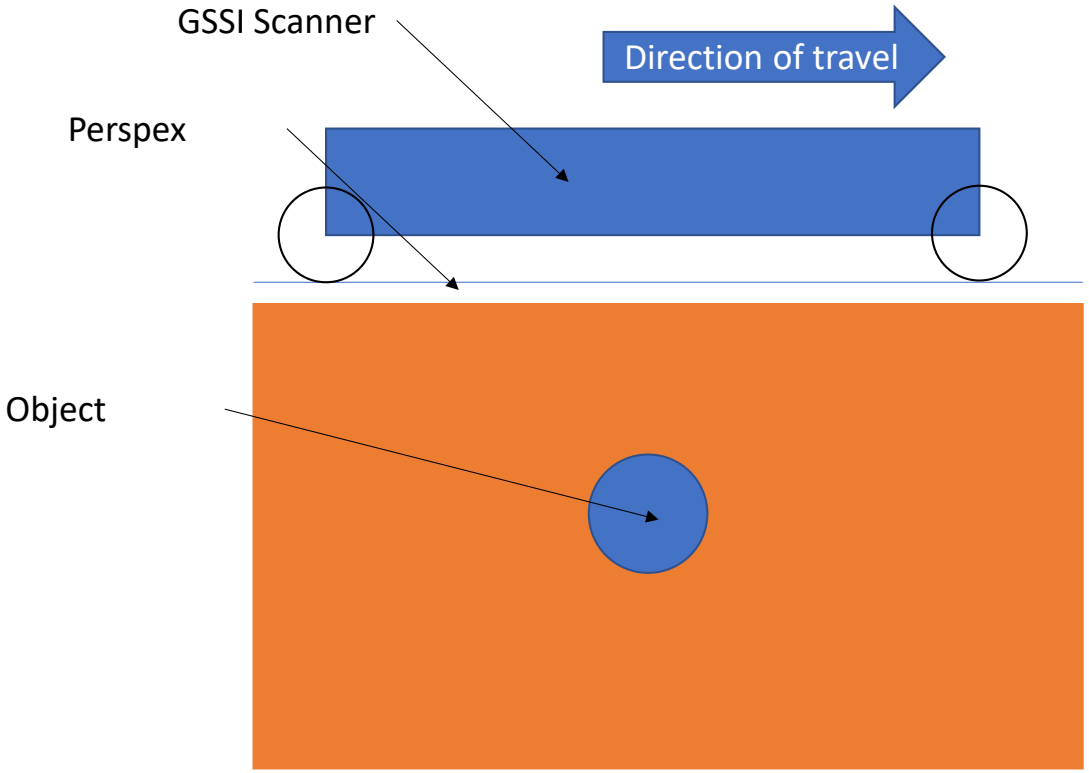


Figure 4-9 Illustration Demonstrating the Rig Testing Approach

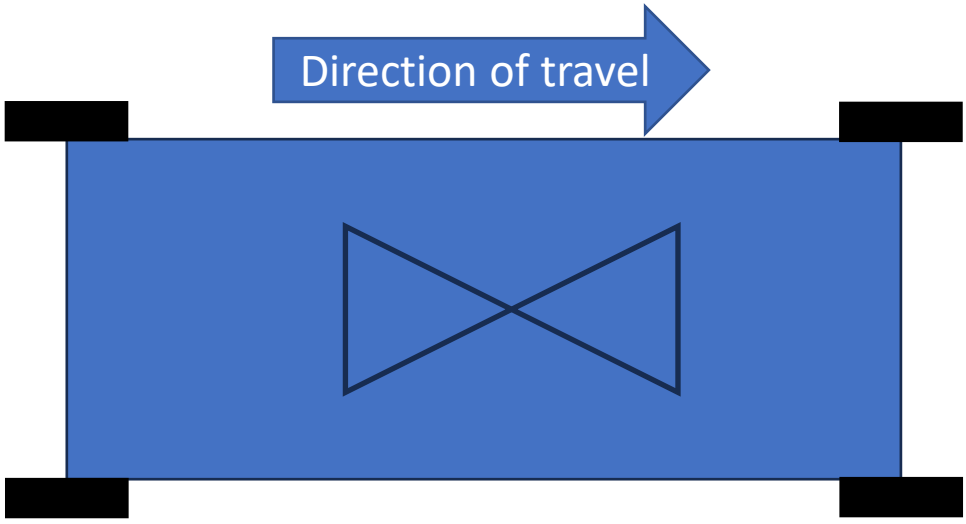


Figure 4-10 A Plan View of the GSSI Antenna Cart Showing the Antenna Orientation in Terms of Direction of Travel

4.3.2 Testing Approach

This section describes the methodology adopted for data collection in a controlled environment. The testing approach adopted for the scans was that there would be an initial scan of a known metallic object at a measured depth to act as a calibration (shown in Table 4-1). The aim of this is to ensure that the hardware is working correctly while being able to generate data which can be used to correlate the simulation results as described in section 4.4. This would be performed at the start of each day and through the course of the day to ensure that parameters are not changing.

The next stage is to use groupings of potatoes which are of known volumes and sizes. The potatoes are measured and recorded in Table 4-2 and Table 4-3. The next step is to perform tests comprising of both individual and combinations of potato groupings to generate data which will be used for deep learning.

Calibartion	Outer Radius (mm)	Wall Thickness (mm)	Volume (mm^3)
Sphere	120	3	529409

Table 4-1 Geometry of the Calibration Sphere

Potato	Volume (mm^3)	Length (mm)	Width (mm)	Height (mm)
1	403.8	108	85	65
2	373.9	105	90	67
3	373.9	115	85	65
4	403.8	115	88	65
5	284.2	9.5	80	65
6	329.1	120	76	63
7	314.1	10	85	60
8	344.0	125	80	65

Table 4-2 Singular Potatoes Geometry Data

Potato group	Volume (mm^3)	Number in Group
1	785.3982	4
2	596.9026	5
3	1005.31	6
4	862.141	5

Table 4-3 Group Potato Geometry Data

The scanning was performed as a forward and reverse traverse across the top of soil, with the objects buried at specific depths.

4.3.3 Rig-Test Results

The stored B-scan result for a single sphere is shown in Figure 4-11. The image on the left is the raw forward and reverse traverse, and clearly shows the metallic sphere in each traverse. However, it also became clear that the back side wall was having an impact on the scans which is indicated by the blue 45° line in the left image of Figure 4-11, though interestingly enough the front side wall remains invisible. To reduce the effect of this, the empty soil box was scanned (in the middle of image of Figure 4-11) and the two files were subtracted in the image on the right.

It is clear that the GSSI pro scan antenna is capable of detecting a sphere in soil with some success.

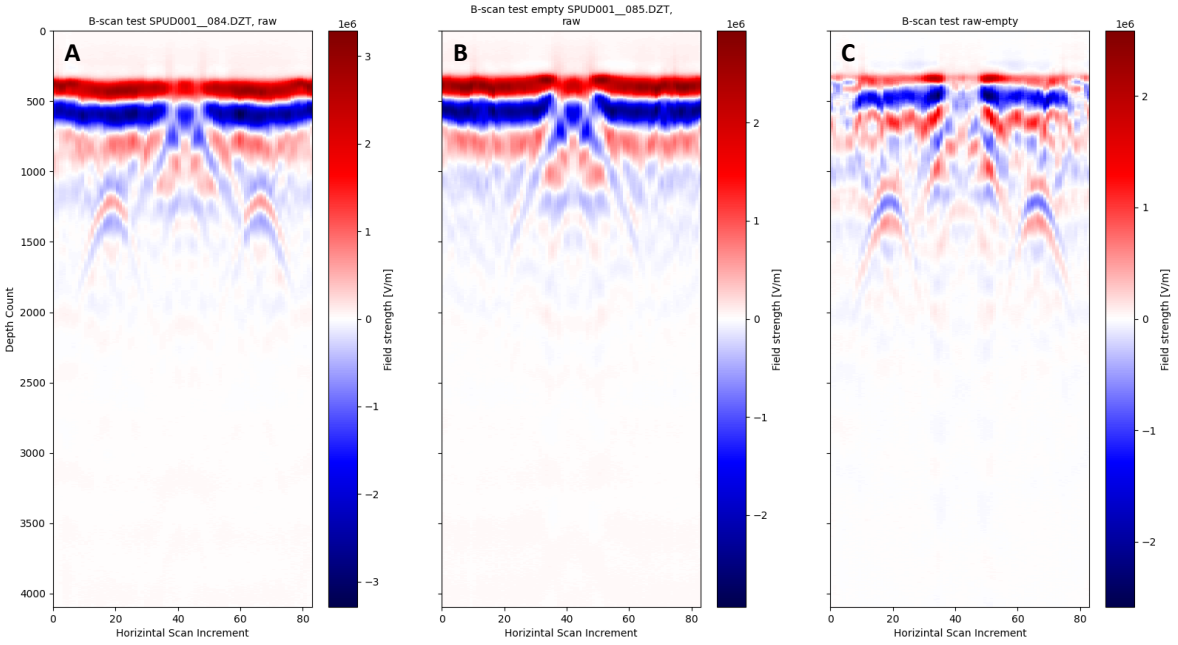


Figure 4-11 The Results of Scanning a Single metallic Sphere (Image 'A' is the Forward and Reverse Scan, Image 'B' is Forward only, and the Image 'C' is the Processed Forward Scan)

Taking a closer look at the final image in Figure 4-11 and highlighting 4 zones of interest in Figure 4-12 listed as Zones A to D below:

Zone A – The air to soil boundary.

Zone B - The back wall reflection.

Zone C – Interaction between objects, an example can be seen in Figure 4-13.

Zone D – The object of interest (in this case the sphere) which always forms a hyperbolic curve as described in in introduction.

The next step is to try and detect actual potatoes in the soil box. The first objects buried were single potatoes (1, 2, 3 and 4 from Table 4-2). The results are shown in Figure 4-14, with images of the potatoes shown in Figure 4-15.

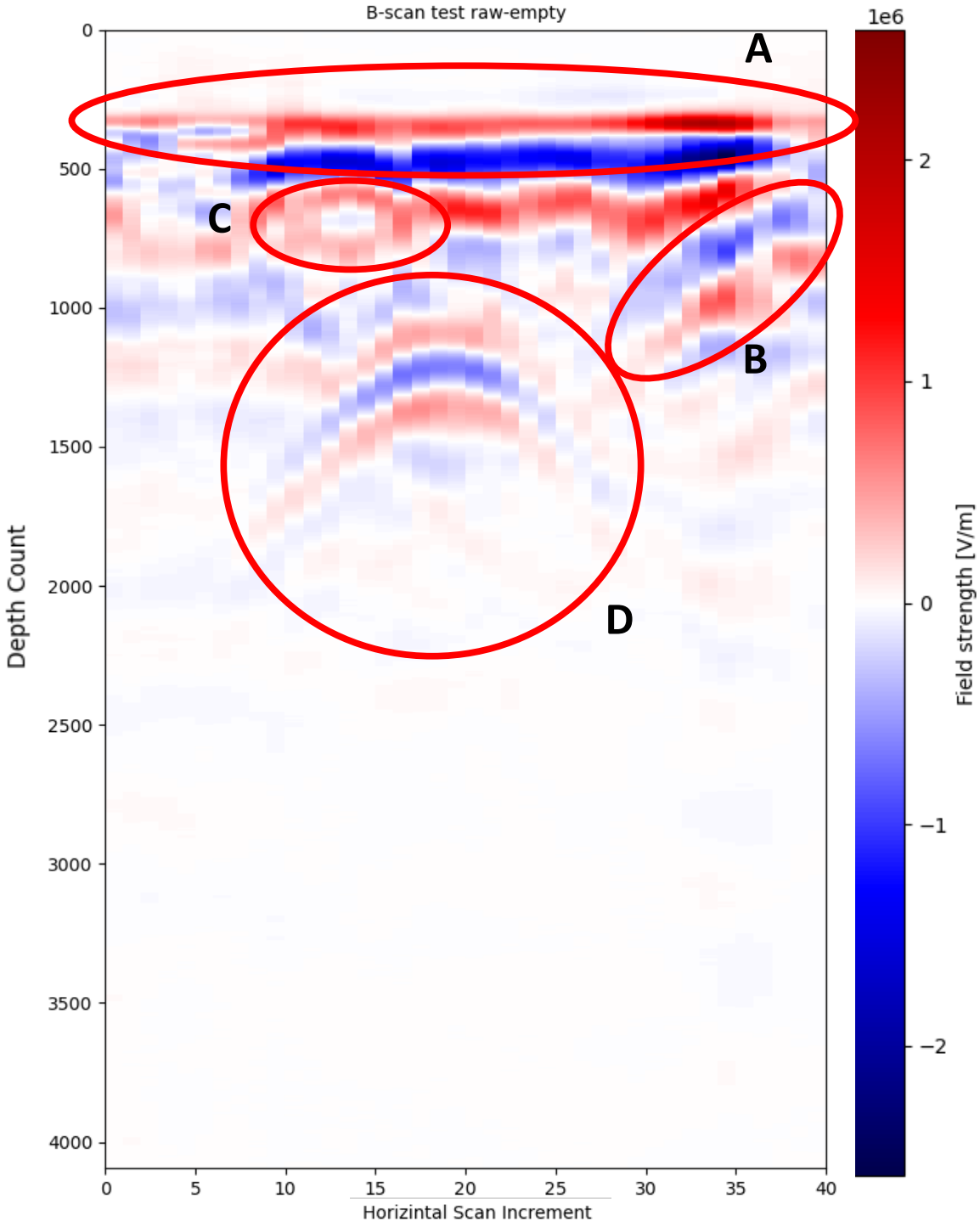


Figure 4-12 Explaining What the Important Features are in the GPR Scan

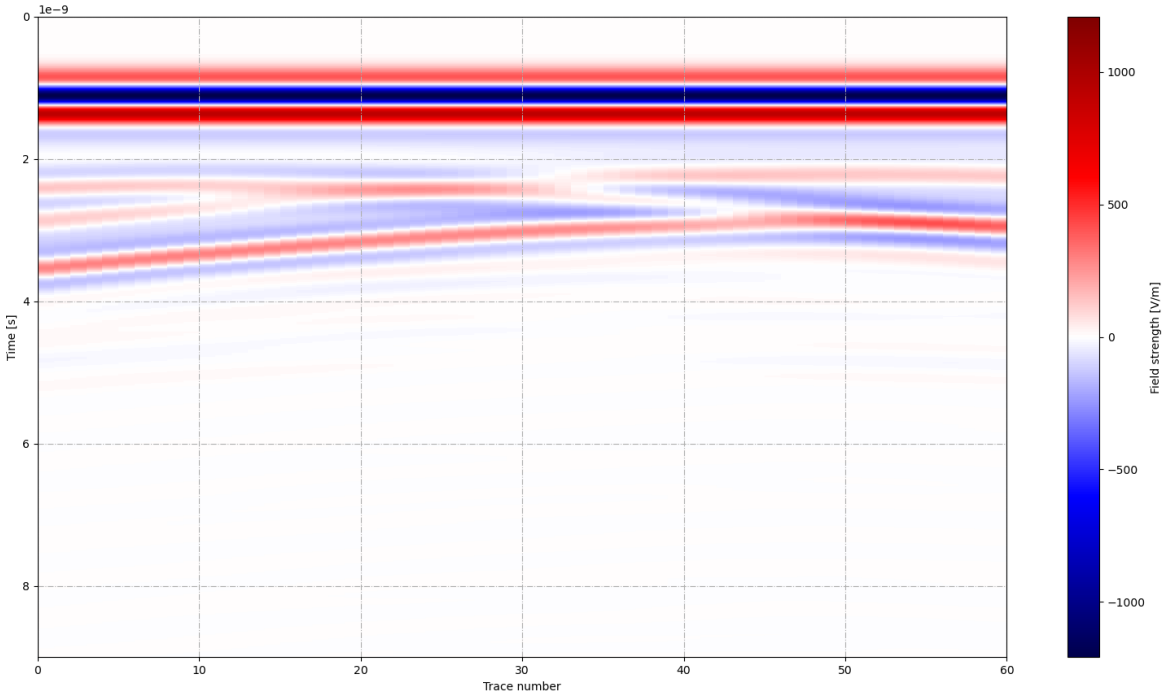


Figure 4-13 2D GPRMax Simulation of Two Cylinders

The next set of experiments was to bury a mixture of large potatoes and smaller potatoes, some of the results of this are shown in Figure 4-16 and Figure 4-17. The groups buried were group 1 and large potato number 5 from Table 4-2 and Table 4-3.

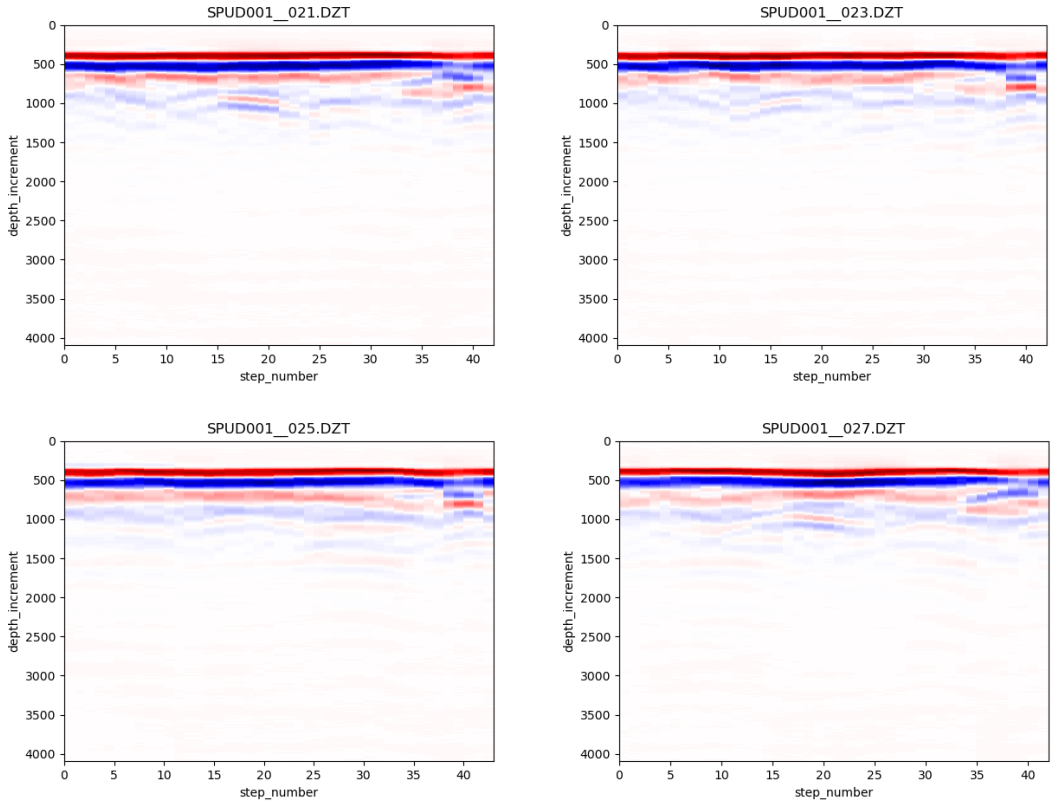


Figure 4-14 GPR Scans of Single Large Potatoes in the Test Rig

Figure 4-15 Showing Actual Buried Potatoes for Figure 4-14 From the Test Rig

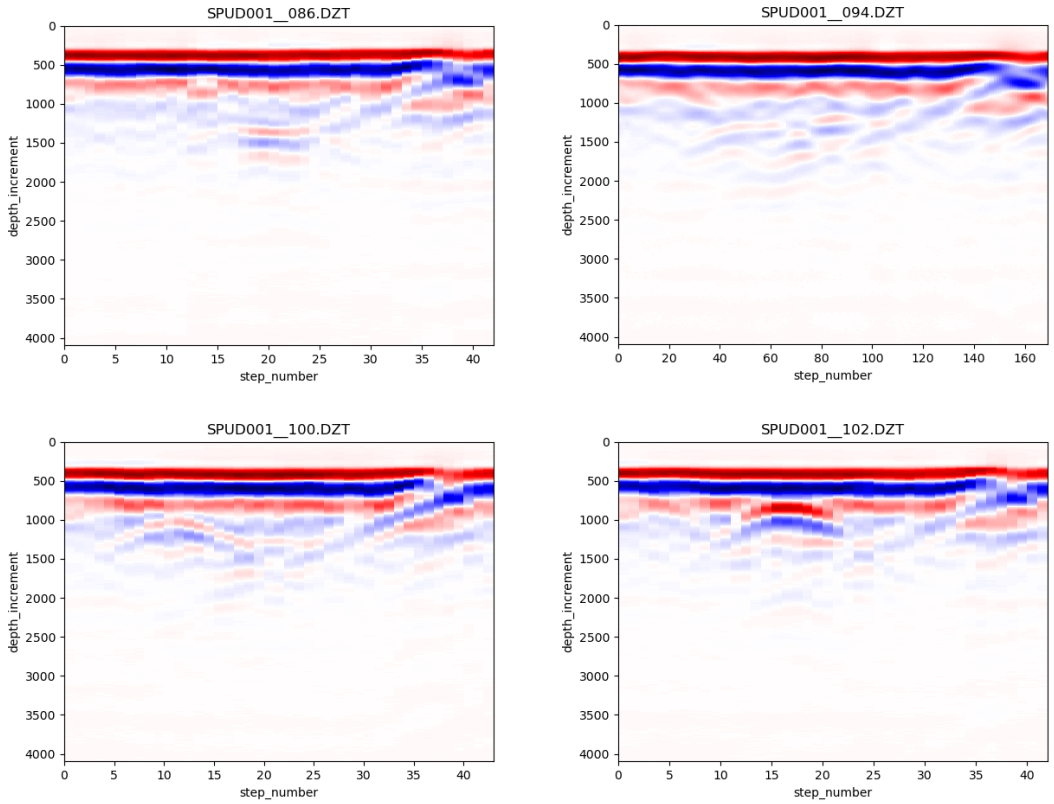


Figure 4-16 Showing GPR Scans of Clustered Potatoes in the Test Rig

Figure 4-17 Showing Actual Image of Clustered Potatoes for Figure 4-16 From the Test Rig

There are minimal configurable parameters available to increase the resolution to the user and increasing the number of scans per cm to 2 scans every cm has the benefit of increasing the resolution, however, as can be seen in Figure 4-18 this still fails to show the wave reflection from individual potatoes.

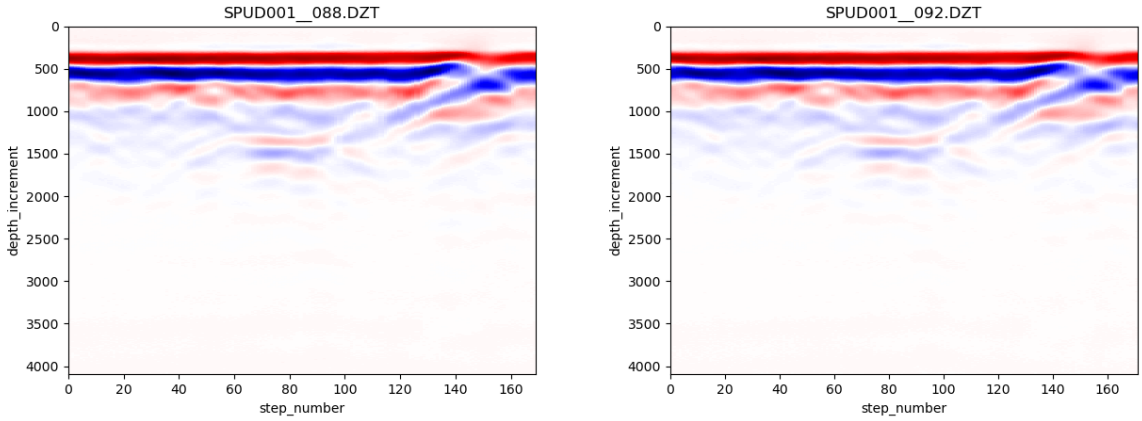


Figure 4-18 Increased Number of Scans per cm

This work demonstrated that in all the images it is possible to detect a buried tuber, however the issue is that commercial software is decreasing the resolution when compared to the simulation predictions and it is difficult to see individual potatoes. This is the main issue when using hardware and software which is not designed for the specific purpose. The system is more than capable of detecting subsurface defects in concrete and building materials, however the detection of individual organic matter in soil is beyond its capability. As a result, for this approach to succeed it will be required to create an additional map, between the high-resolution simulation results and the lower resolution. The mechanism to achieve this, using GPRMax, is explained in the next section.

4.3.4 Correlation With GPRMax

A GPRMax model has been created with a single sphere in the middle of the medium and the GSSI simulated antenna was passed across the model is shown in Figure 4-19.

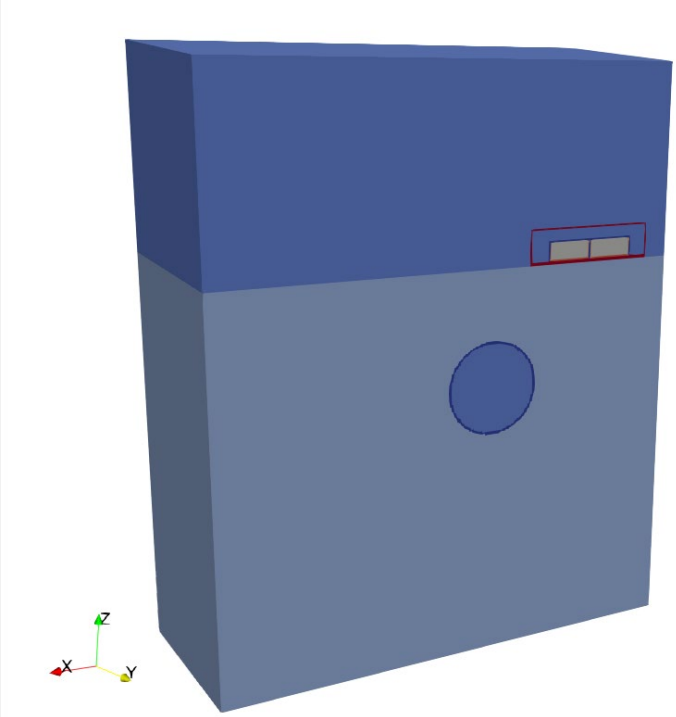


Figure 4-19 The 3D GPRMax Model of a Single Hollow Metal Sphere from the Test Rig

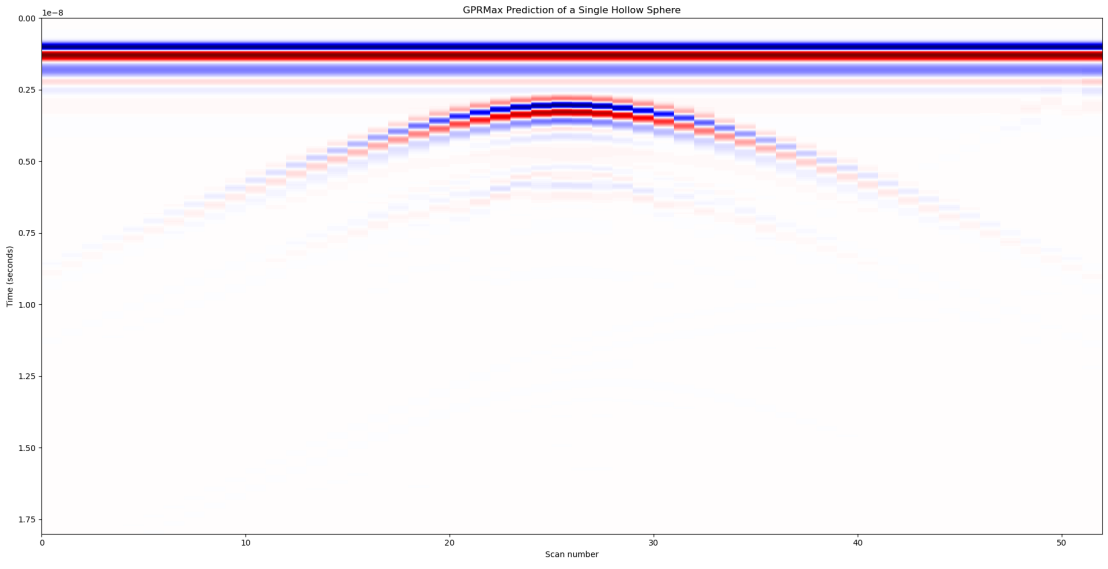


Figure 4-20 GPRMax Predicted Antenna Response B-Scan from the Test Rig

The model was created in 3D (1.2m by 0.5 by 0.5 with a cell dimension of 0.002m) using the inbuilt GPR max model for the GSSI 1500 MHz bowtie antenna (Warren and Giannopoulos, 2019), orientated such that it moved along the x-axis and the target object had dimensions and properties shown in Table 4-4.

Object	Diameter (meters)	material
Outer Shell	0.12	PEC
Inner Shell	0.114	Free space

Table 4-4 Metal Sphere Properties for GPRMax Validation Study

The predicted response at the antenna is shown in Figure 4-20. This response is the raw analogue and is what would be expected if an analogue measurement would be taken at the antenna output. The most noticeable difference is that the thin spherical wall can clearly be seen in the analogue data, as well as the ringing of the sphere (section 4.2.2), these are more clearly seen in the A-Scan taken from position 26. In Figure 4-21, the following zones can be clearly seen.

1. Zone 1 is the Air/Soil boundary.
2. Zone 2 Outer shell of the Sphere.
3. Zone 3 GPR Ringing (as discussed in section 4.2.2).
4. Zone 4 GPR Ringing (as discussed in section 4.2.2).

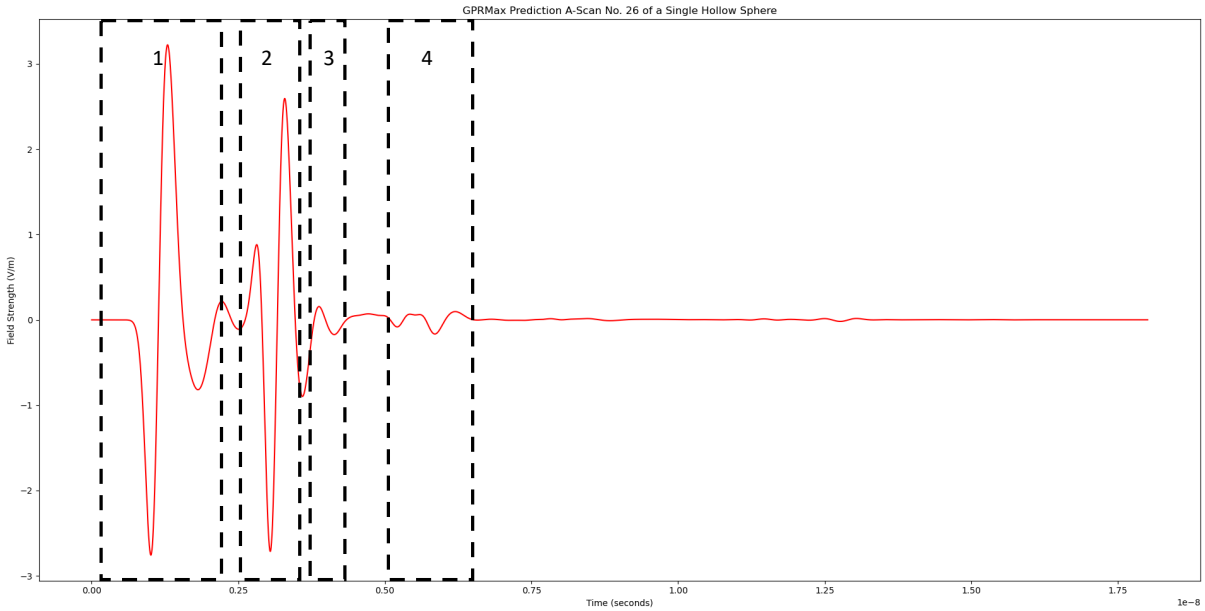


Figure 4-21 Section 26 of A-Scan of a Hollow Sphere from the Test Rig with Zones 1-4 Labelled.

Even a cursory glance at the results (e.g. Figure 4-5 and Figure 4-18) from the GSSI equipment shows that there is a significant reduction in resolution of the signal, which is going to have a substantial impact on the testing results of clustered objects. The signal has to undergo significant signal conditioning and processing for it to be recorded in digital format. Just applying the best guess of processing filters on the data, without the Analogue to Digital and low/high pass filters etc, begins to improve the correlation between the simulation and the GSSI data, this can be seen in section 5.5.

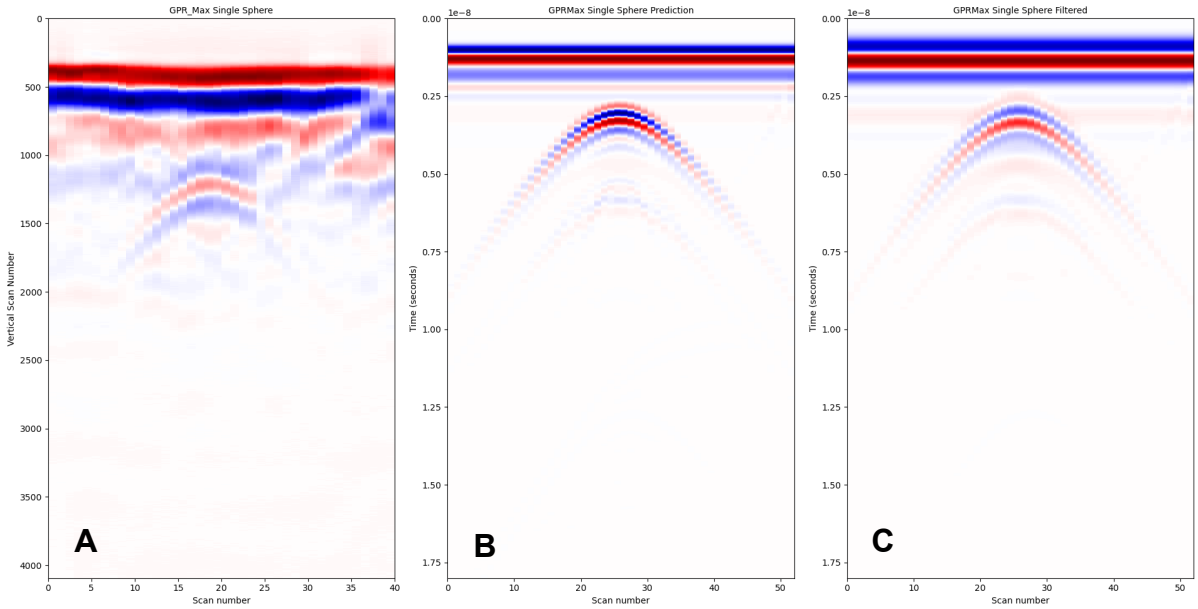


Figure 4-22 Image 'A' Shows the Test Rig Hollow Sphere B-Scan. Image 'B' is the Simulated GPRMax Image and Image 'C' is the Simulated GPRMax Image Along with a Filtering Applied

The predicted image in Figure 4-22 is still showing the ringing effect of the hollow sphere, however the correlation between the two images is improved. Since it has been shown that the output from GPRMax can be adapted to simulate the GSSI measured data, the next stage is to look at the digital conditioning of the signal.

The first step is to make an approximation of the number of bits used in the Analogue to Digital Convertor (ADC), taking the maximum and minimum values from the testing performed. The most likely is a 24-bit encoder, which gives a range of between ±8388608. The sampling rate will be assumed to be the same as an off the shelf ADC, which gives 5 million samples per second and has a differential voltage input. Once the sample has been converted, the A-scan signal needs to be up sampled to 4048 samples long.

It is clear that the system is capable of detecting a single metallic sphere, and it is possible to detect to the bulk response to the potatoes, however the issue remains that it is very difficult to identify individual potatoes within the soil medium using a

commercial antenna system. The reason for this is down to a combination of frequency of the signal and also the discrimination of the signal as a result of limited survey data. If the system was to be optimised in both frequency and antenna arrangement, then the results will be closer to the GPRMax prediction.

4.4 Simulations

GPRMax allows virtual environments to be created. An example of the output can be seen in Figure 4-23 which shows the input model and the numerical solution for the GPR responses (the vertical depth and the horizontal is the scan number – or axial distance). The software allows the user to create different soil distributions and to add roots and moisture contents.

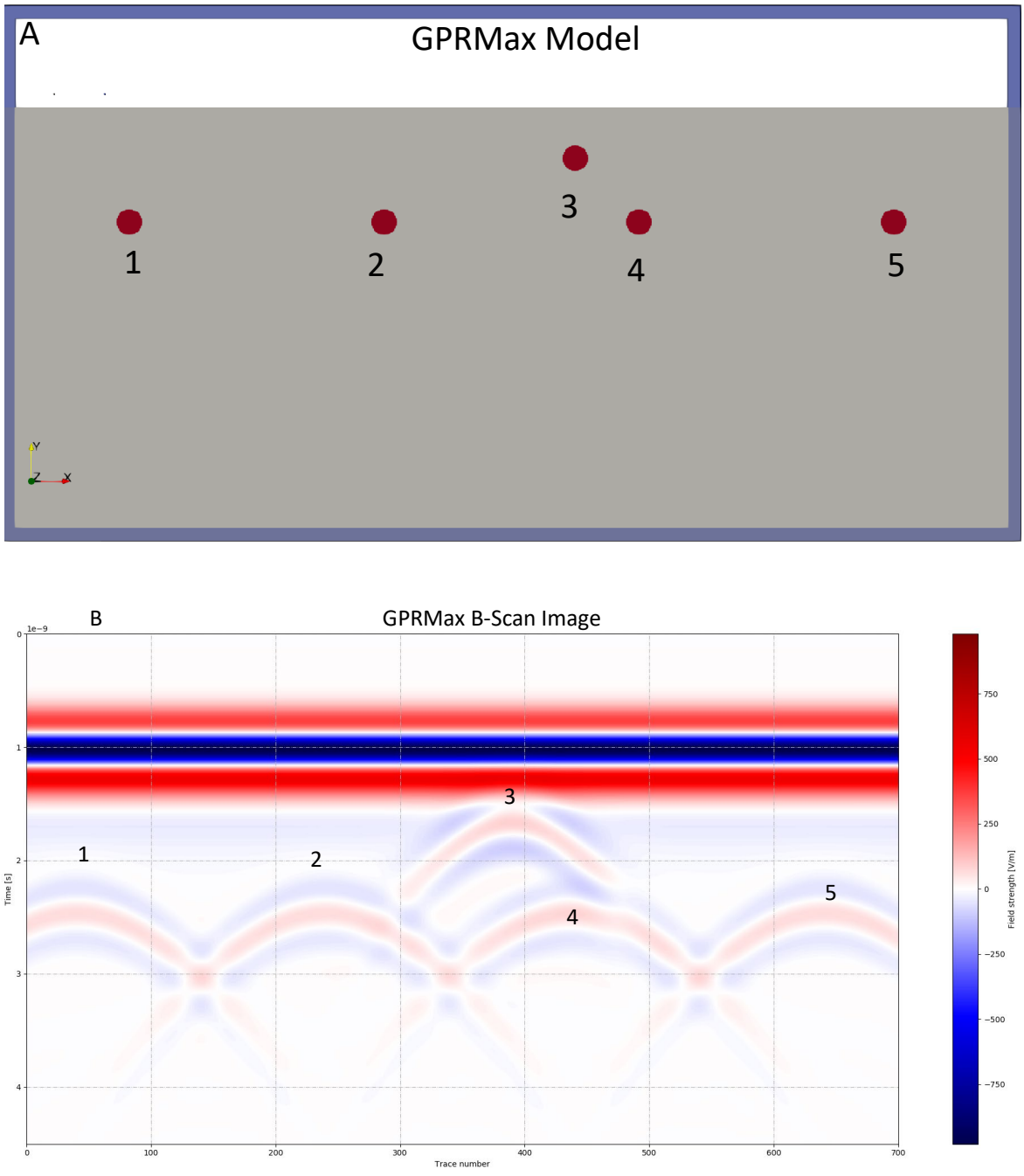


Figure 4-23 GPRMax Physical Model (Image 'A') and the GPRMax Simulated B-Scan Results ('B') numbers 1-5 in image 'A' match the GPRMax responses 1-5 in 'B'

The simulation code (Warren and Giannopoulos, 2019) generates analytical predictions for the Maxwell parameters by using Finite Difference Time Domain (FDTD) methods. This approach allows the prediction of the signal for a given object and is employed in software (Warren and Giannopoulos, 2019).

By using the GPRMax software, it is possible to select and optimise any signal processing requirements, train the deep learning architecture, and verify the feasibility of creating virtual 3D representations. A sample of the properties applied to the simulation models are shown in Table 4-5.

MATERIAL	ε_r	CONDUCTIVITY (mS/m)
Air	1	0
Water Fresh	81	0.5
Soils		
Clay (dry)	2-5	2-100
Clay (wet)	8-40	20-1000
Agricultural Land	15	
Pastoral Land	13	
Soil (fine)	41-49	40
Average Soil	16	5
Rocks		
Limestone (dry)	4 to 8	
Limestone (wet)	6-15	10-100
Sandstone (dry)	4-7	10^{-3}-10^{-7}
Sandstone (wet)	6	10^{-2}-10^{-3}
Shale (wet)	6 to 9	10 to 100

Table 4-5 Properties for Common Geological Materials (Reynolds, 2011)

4.4.1 GPRMax Validation Studies

Significant validation of the software has been performed independently and is available on the provider's website (Warren and Giannopoulos, 2019). An example is found in (J. Ježová , L. Mertens, 2016) who compared 2D simulations to laboratory experiments on a circular tree trunk and found the predicted results correlated to the measured results. A second report, (Sonoda and Kimoto, 2019) whilst not mentioning GPRMax specifically, does reference using Finite Difference Time Domain (FDTD) software to generate a deep learning training database. In

(Kafedziski, Pecov and Tanevski, 2018) a mixed training dataset of 109 anti-tank mines was created along with a number of simulated results using GPRMax for use as a deep learning training set.

Additional validation of the application of GPRMax within a botanical setting was performed in (Guo *et al.*, 2013). In this paper, the authors were interested in learning how the spacing between roots and soil water content affect the root investigation using GPR. The method adopted was to create a forward simulation model in GPRmax which were then compared to in situ collected radargrams, which were found to be of similar resolution along with similar trends between radargrams and root water content, root spacing, root depth and antenna frequency. The correlation coefficients range from between 0.87 and 0.96 between the GPR data taken from the simulated data and the field collected data. As a result, the paper concludes that using the simulation is an effective approach to assessing the limiting factors on root detection and quantification.

A comparison of the simulation results and collected data is shown in Figure 4-24, where figure ‘a’ is the location of and diameter of the samples, ‘b’ is the in situ collected radargram, ‘e’ is the simulation radargram from GPRMax, ‘c’ & ‘f’ are the respective radargrams after a Kirchhoff migration and ‘d’ & ‘g’ are after a Hilbert transform.

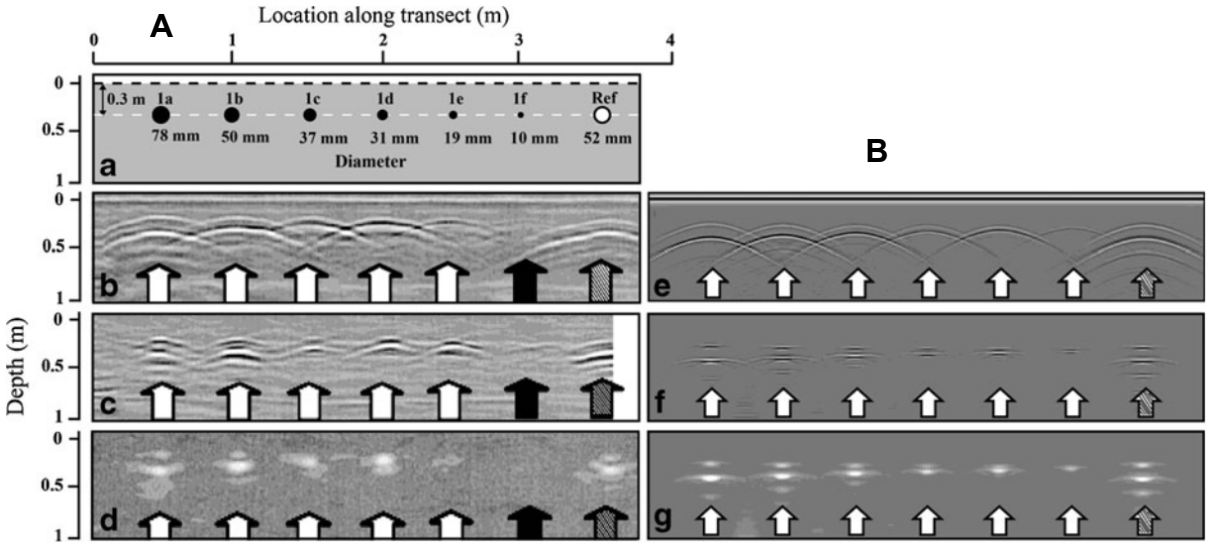


Figure 4-24 Comparison of Collected ('A') and Simulated Data ('B') from Root Study
(Guo *et al.*, 2013)

4.4.2 GPRMax Software

The output from a 2D GPR scan (B-Scan) is a hyperbolic representation of the object. This is because the signal response is in the time domain and will receive a reflection at increasingly smaller time points the closer the antenna approaches the object. The 2D output is a stacking of amplitude vs time signals (A-Scan as shown in Figure 4-25), which represents the antenna response to the returning electromagnetic waves.

The simulation predicts a response which is based on the properties defined for the materials, making classification a lot easier. Later sections of this chapter will show that it is possible to recreate the image that created the signals using only the signals, without any other parameters, effectively solving the inverse problem.

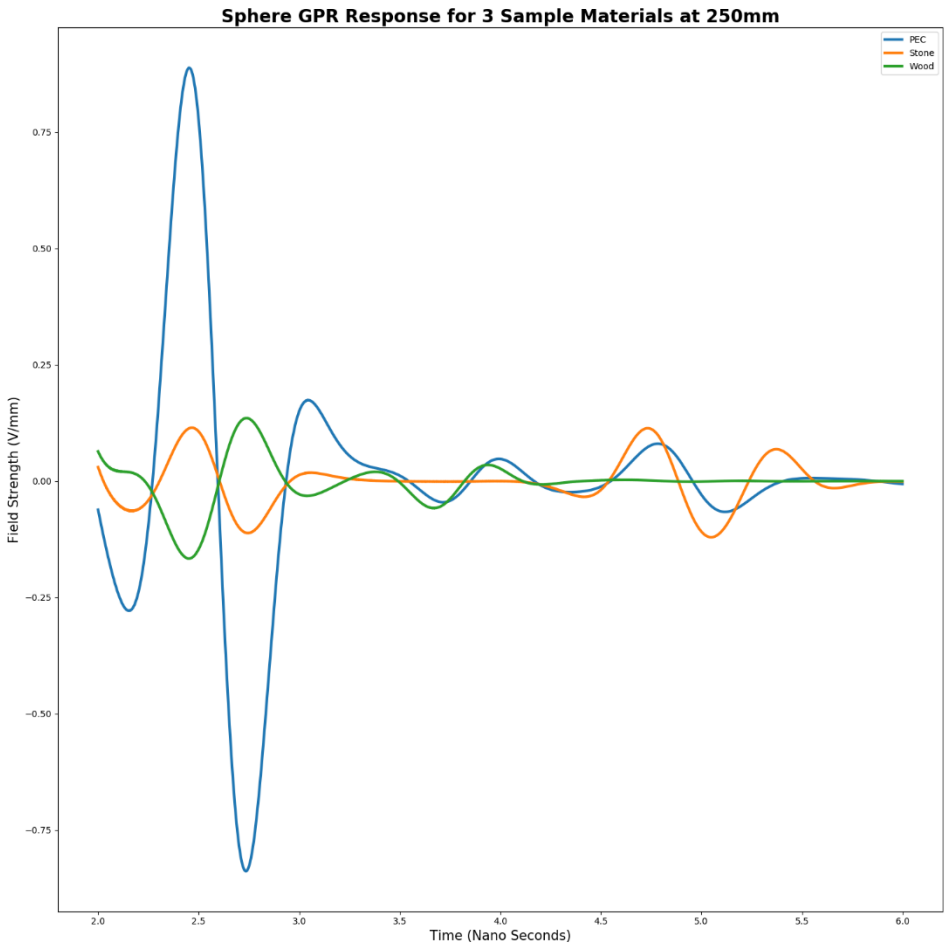


Figure 4-25 Amplitude-Time Trace (A-Scan) for 3 Materials (PEC, Stone & Wood)

To generate a suitable dataset for the training, 15,000 simulations were run using 3 different materials: two materials representing stone (limestone and sand) and one organic (modelled as wet wood) with the properties shown in Table 4-6 below.

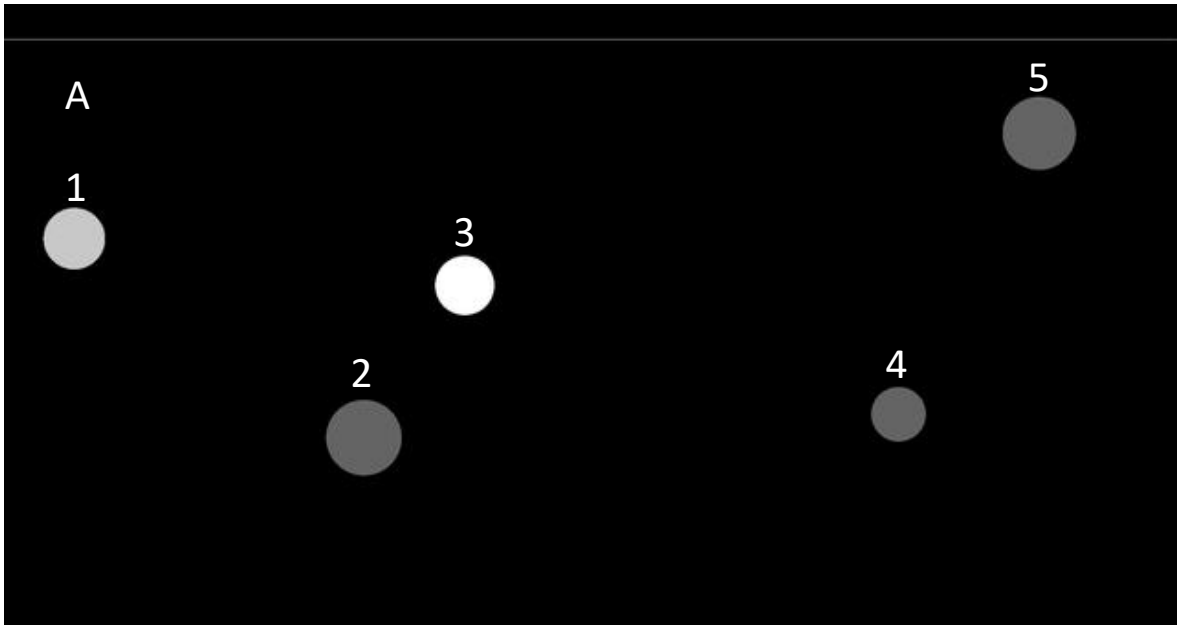
Material	ε_r	CONDUCTIVITY (s/m)
Limestone	10.5	5.5e-2
Sand	21	5.5e-5
Wood	20	5.5e-4

Table 4-6 Material Properties for the 2D GPRMax Simulations

Wet sand was chosen because of the similarity to the wood, and it was felt that having an inorganic material (sand) with similar properties to the target (wood) would be representative of the any real-world application.

A random number of circular objects scattered through each simulation of a perfect medium of homogeneous material with no contaminants. The term “no contaminants” means that the medium is free of any impurities such as water, clay or any other medium which would create noise in the solution. Within the GPRMax software suite there are options to add real-world soil models.

The sizes of objects were random between a diameter of 40mm and 80mm and material was randomly allocated. The results of the simulation are combined into 2 datasets, one containing the GPR B-Scan result, quantity label, material identifier and size label (small, medium, and large), the second contains an additional image representation of the model. These will be discussed in more detail in chapter 0. The B-scan and image are shown in Figure 4-26, where the 3 materials are distinguished by 3 greyscale colours dark grey, light grey and white. Dark grey and light grey are the in organic material and white is organic material.



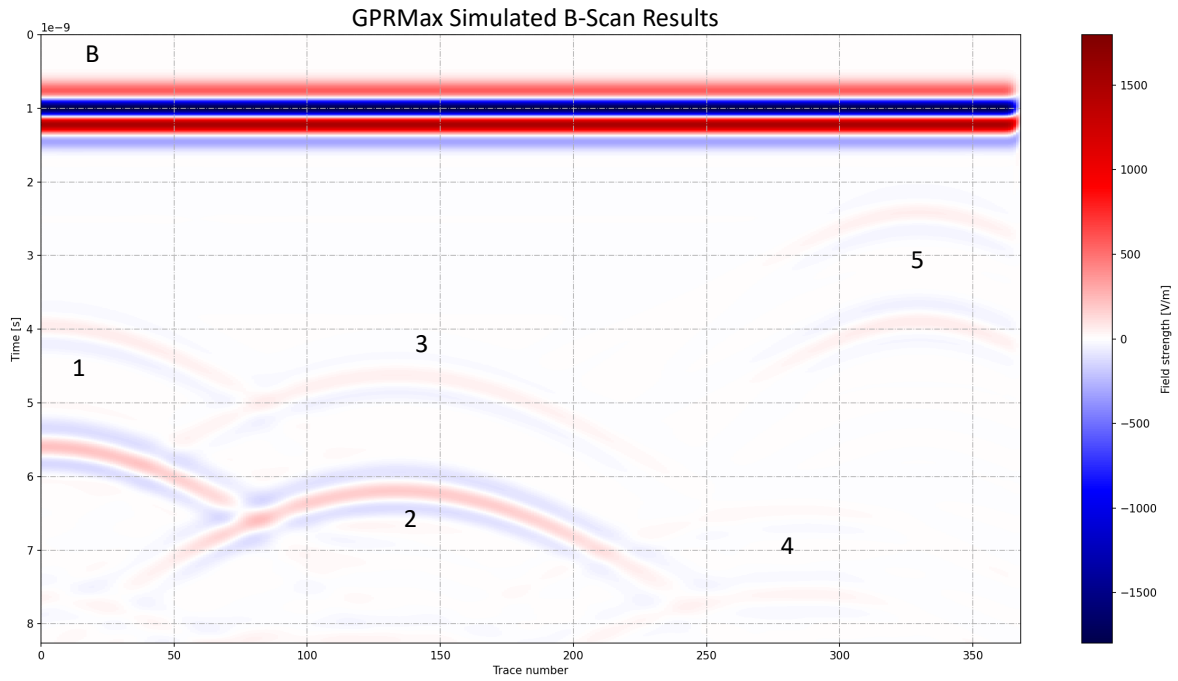


Figure 4-26 GPRMax Physical Model with 3 Materials (Image 'A') and the GPRMax Simulated B-Scan Results ('B') numbers 1-5 in image 'A' match the GPRMax responses 1-5 in 'B'

4.4.3 Assumptions

In the final model, the following assumptions have been made:

1. The depth of the object is of little consequence, and since the properties of the surrounding material affect the time taken to propagate through and dampening (see Figure 4-27) then a simple medium will be used, and any dampening effect can be adjusted with time varying gain.
2. The surrounding medium is homogenous.
3. The air-surface interface will be removed.
4. The material properties are secondary as the important components are the contrast between the surrounding medium and the decay because of moisture.

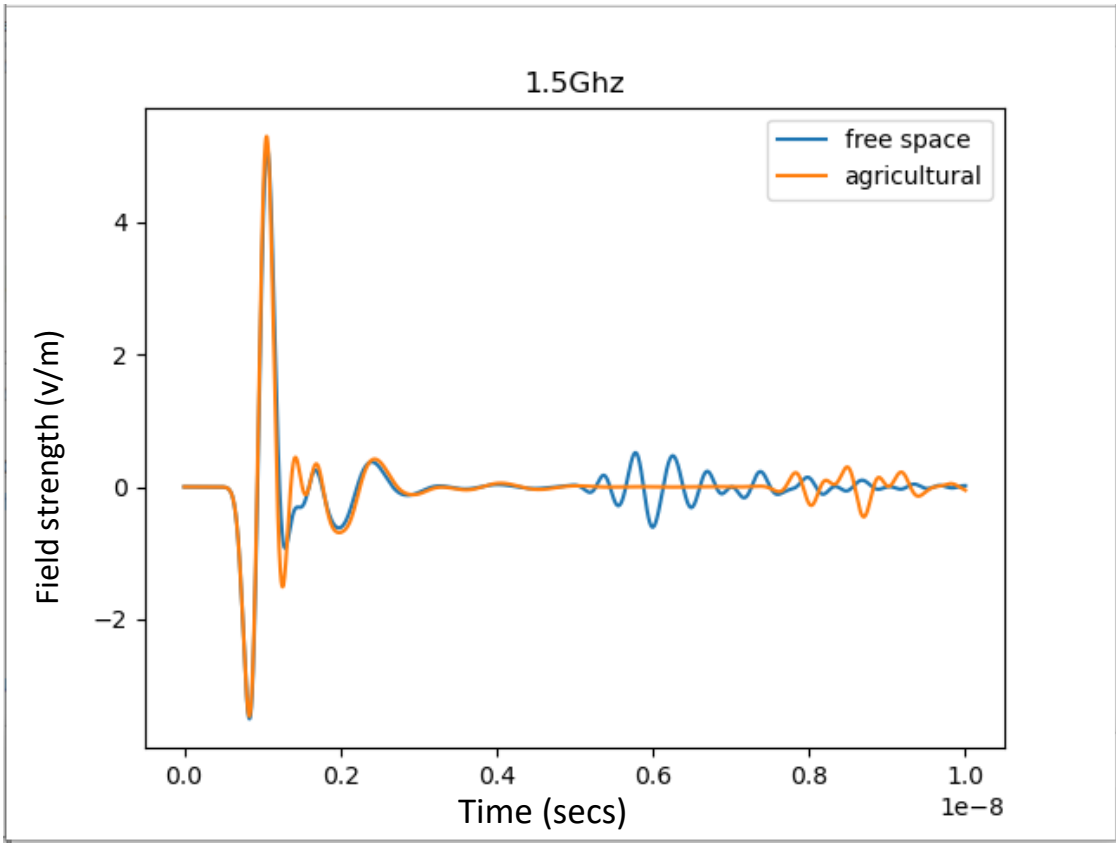


Figure 4-27 Impact of Using Different Background Mediums

4.4.4 3D Tuber Generation

One of the features within GPRMax is that it allows a user to define their own geometry for the modelling. In a separate project, UWE has taken 400 scans of tubers (Liu *et al.*, 2021b) which are surface point clouds, that have been then used to create the dataset used in the deep learning training within this project.

Using Python code written for this PhD project, the 400+ scans of tubers have been placed within a geometry block which is 600x500x500mm at random around the centre of the space. Once placed, a basic clashing routine was used to try and minimise any overlaps between the objects. The tubers were then filled and saved to a file ready for the use in GPRMax. A random quantity of between 0 and 10 was inserted into each model. Random locations and random orientations were used to try and create enough depth in the dataset to define the design space of the real-

world plants, which tend to be more ordered in distribution. In order to maintain a consistent vertical depth, the minimum depth of the upper most surface was set to 100mm below the soil/ground interface. Samples of the models are shown in Figure 4-28 and Figure 4-29 below.

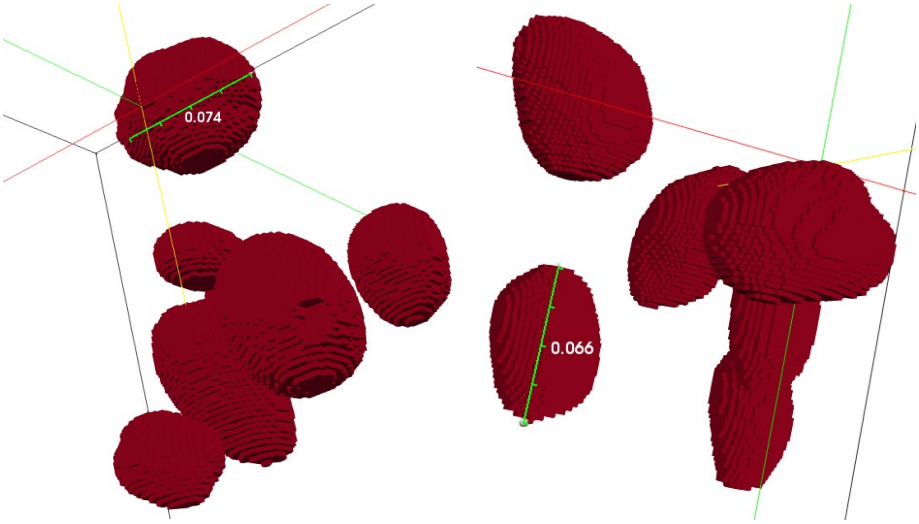


Figure 4-28 Distribution and Selected Sizes of one of the 3D Models

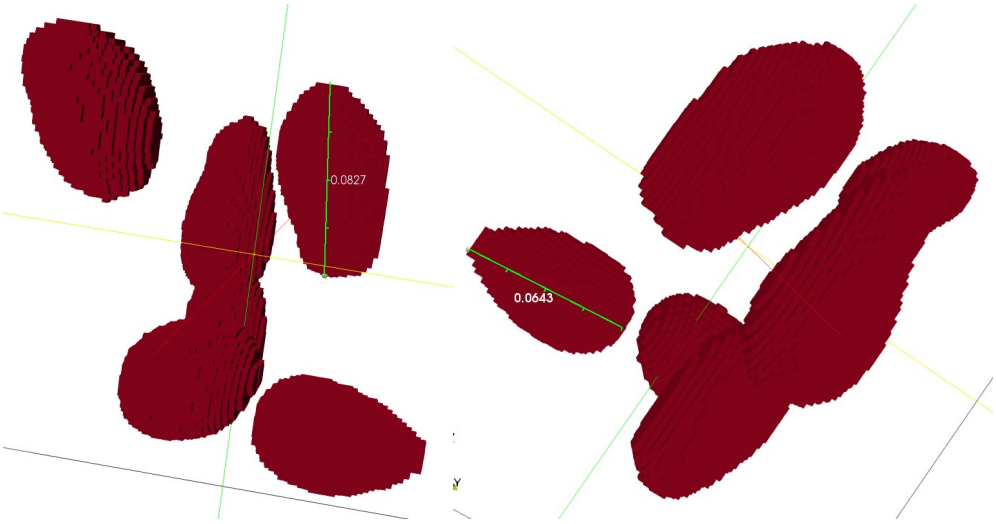


Figure 4-29 Distribution and Selected Sizes of one of the 3D Models

The main problem with using any numerical simulation is accurate input data, and this was no exception. A study was performed on unprocessed potatoes and chips in (PACE *et al.*, 1968), however this paper is unavailable to the public. It is assumed that since a potato is approximately 80% water, then general data for “wet organic” material is suitable. Therefore, since the data is normalised, the most important property is the contrast between the surrounding medium and the objects of interest, and as a result the approach has been to adopt a generic organic value.

The presence of air or water in the medium has a noticeable effect on the GPR signal. The presence of either of these materials will distort the wave velocity due to the EM responses are significantly different to other materials, with water being more likely to distort the transmission velocity, which can result in incorrect depth readings. In the case of peat based soils it is possible to have a transmission velocity as low as 0.035m/ns (Utsi, 2017). In the presence of air, the EM waves will travel faster and thus have the opposite effect on the transmission velocity. The presence of water in a scanning region contributes to attenuation of the signal, which will present itself as a loss of signal in the B-scan. The more similar the surrounding medium in moisture levels to the object of interest results in an object which are

more difficult to distinguish from the surroundings. The surrounding “dry” medium will not attenuate the EM wave significantly so it will be clear when the EM crosses the boundary between “wet” and “dry”.

The contrast is derived from the ratio between the dielectric properties of the medium and the object of interest. This property therefore helps to identify the difference between an object and the surroundings. Using a dielectric property which is based on a high volume of water is reasonable in the models as this will take into account the presence of water, which will have the desired effect on the signal attenuation expected more appropriately than using dielectric properties taken from “dry” materials.

It is expected that this will have a limited impact on the accuracy of the models because the data is normalised to between 0 and 1 in the models so a high response will always be 1 and a low response will be 0.

4.5 Chapter Summary

This chapter summarised the efforts to collect data to be used to validate the model developed as the core part of this project for generating 2D and 3D images of potatoes, along with a discussion regarding the simulation software and then how well the results from the simulation correlate to the tested data points using existing validated antenna models. The GSSI antenna had more success at detecting the potatoes, but due to a significant amount of signal processing resulted in a below expected level of resolution in the output signal. The critical issue in using any GPR system is to be able to collect the signal straight from the antenna, as any processing after this point only reduces the resolution of the signal.

As concluded in chapter 3, the theoretical solution for solving the inverse problem is too complex and too rigid. As seen in this chapter, getting reliable data from the field has also been fraught with difficulties and limitations beyond the scope of this work. Chapter 5 will move on from these issues to describe the selected solution to the Inverse Problem in the context of this work, which is a data-driven model.

5 Data Driven Solution

Now that the inverse problem has been defined, and the methods for collecting data explained, this chapter describes the data-driven solution to resolve the inverse problem. The data driven solution, which was developed to form this thesis, will be detailed in sections 5.1 to 5.1.2, whilst section 5.1.4 to 5.2.6 will walk the reader through the key technologies and concepts employed in developing the solution. The remainder of the chapter discusses the results of the two methods adopted in this thesis. The first set of results, described in section 5.7 is to demonstrate that it is possible to map between the 2D GPR and the 2D image space using an unsupervised learning-based architecture. In principle the approach adopted in this section will create both a forward and reverse map such that:

$$f: 2D\ image \rightarrow 2D\ GPR$$

$$f: 2D\ GPR \rightarrow 2D\ Image \qquad \textbf{5.1}$$

$$(x, y) \in 2D \text{ there is a } (x', y') \in 2D$$

That is, for every location in the 2D GPR space there is a function which relates to the 2D Image space, as shown in Figure 5-1.

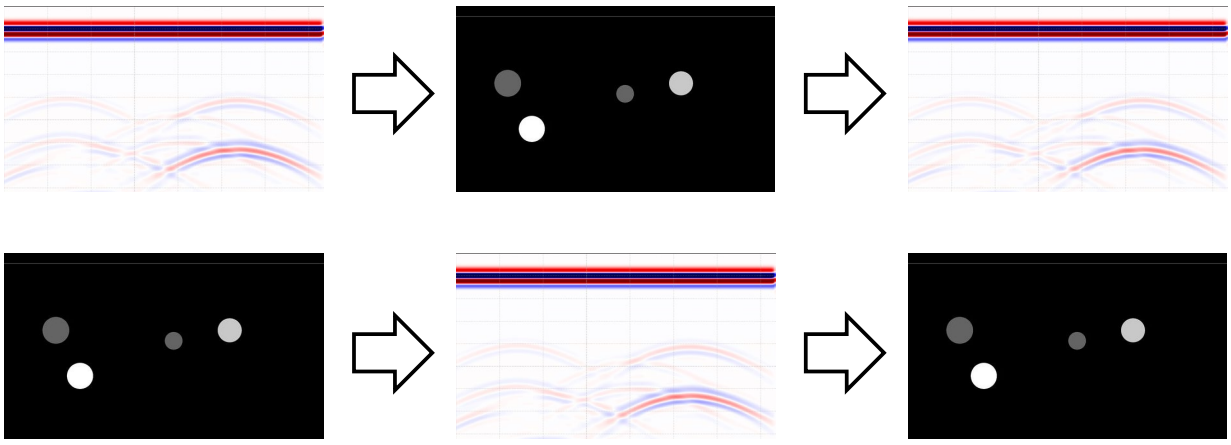


Figure 5-1 Principle of 2D Cycle GANs using Simulated Data

The second set of results are detailed in section 5.4, is to demonstrate that it is possible to map between different dimensional spaces such that:

$$f: 2D\ GPR \rightarrow 3D\ Image$$

$$(x, y) \in 2D \text{ there is a } (x', y') \in 3D \qquad \textbf{5.2}$$

That is, for every location in the 2D space there is a function which relates to the 3D space, as shown in Figure 5-2.

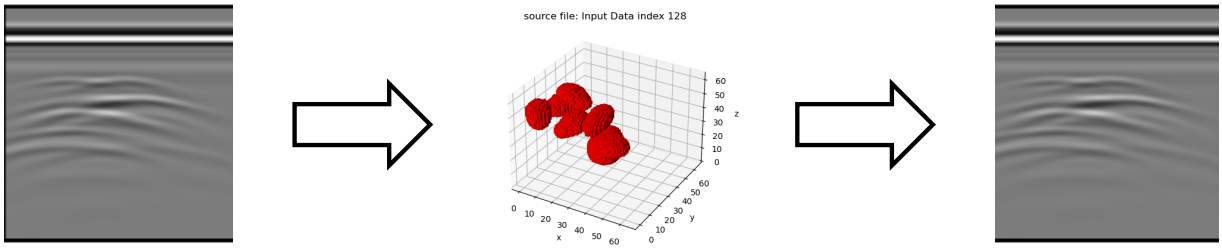


Figure 5-2 Principle of 3D Cycle GANs using Simulated Data

5.1 Design Principals of the Data Driven Solution

This section will discuss in detail the main concepts and features selected for the design of the Data Driven Solution developed to solve the inverse problem in this work. It will explain the computational representation of 3D data, the autoencoder used to process is, the algorithms it builds on, and the reasons that these were selected over other candidates.

5.1.1 Computational Representation of 3D Data

Once GPR data is processed as explained in chapter 4, the next challenge is generating a 3D image from the output. This section describes the challenges of 3D image generation and the approach to resolve them.

The input into the 2D portion of the network is a 2D tabular array. The 3D portion of the network, however, is based on voxels. Voxels are volume pixels, 3D representations of colour, and the cornerstone of 3D computer graphics. The data structure selected in this application is a 3D position vector for each voxel which is

subsequently encoded with a three channel RGB (Red, Green and Blue) value between 0 and 1 (equivalent to 0 and 255). Each voxel grid is related back to the input point cloud. The point cloud is a discrete set of datapoints which have been generated from the real-world potatoes. Currently the python library used for this is open3D (MIT, 2021). There are several limitations to this library, the major one being the inability to save voxel grids directly. The main benefit of this library is that it allows the user to rescale the size of a point cloud and hence the voxel representation. The importance of this is discussed in section 5.4.4.

5.1.2 Design Philosophy

With the objectives set, the next step is to define how to achieve them. In chapter 2, The main focus on applying deep learning has been based on the identification of the object based on the signal as shown by (Kafedziski, Pecov and Tanevski, 2018), (Zhang *et al.*, 2021) and (Hou *et al.*, 2021), attempts have been made to use GANs to enhance a dataset (Puzyrev *et al.*, 2022) and it seemed only logical to use the power of deep learning to solve the inverse problem (which has been justified by another team adopting this technology (Dai *et al.*, 2022). It was demonstrated by (Zhu *et al.*, 2017) that it is possible to map between two spaces using a cycle GAN with a degree of accuracy, however the work performed in this paper maps an image on top of another image, this is in fact only part of the solution. Using the paper on 3D GAN's (Jiajun *et al.*, 2016) and on cycle GANs (Zhu *et al.*, 2017) as inspiration it seemed to be the next logical step forward to use the generative Deep Learning techniques highlighted by the cycle GAN as the cornerstone for the solution adopted address this problem.

The main difference between the application in this thesis and the one in the paper is that here we need to map between two different modalities. In other words, what is required is to create a system which can encode the B-scan and decode it into an image. This can be done using an autoencoder based architecture, as described in more detail sections 5.1.3. The main drawback from using just autoencoder based architecture is that the results tend towards the “fuzzy” image which is a result of the limited amount of information that passes between the encoder and the decoder (the input is reduced to an ‘n’ dimensional vector). Some techniques which can be used to try and minimise this effect such as using Maximum Mean Discrepancy as discussed in section 5.1.4. It is well documented that using a GAN (Goodfellow *et al.*, 2016) to generate an image overcomes this fuzziness as described in section 5.2. It is for this reason that the autoencoder architecture has been supplemented with a combination of using MMD the GAN architecture. This results in a CVAE-GAN and the components are discussed in sections 5.4.3.

5.1.3 Explanation of Conditional Variational Autoencoders

Conditional Variational Autoencoders are a foundation of the solution adopted. Therefore, this section explains how an autoencoder works and how it can be trained such that the objects in the latent dimension are grouped together based on a property. This process is known as conditioning.

Autoencoders are a form of unsupervised deep learning, which learn to encode and then decode an input data signal. There are several types of Autoencoders. The one selected at this point is a Conditional Variational Autoencoder Encoder (CVAE), which is a development of the Variational Autoencoder. An early proposal for a Variational Autoencoder was (Pu *et al.*, 2016).

The architecture of the Convolutional VAE consists of a convolutional encoder layer and an inverse convolutional decoder, which are joined together by a fully connected layer as shown in Figure 5-3 below.

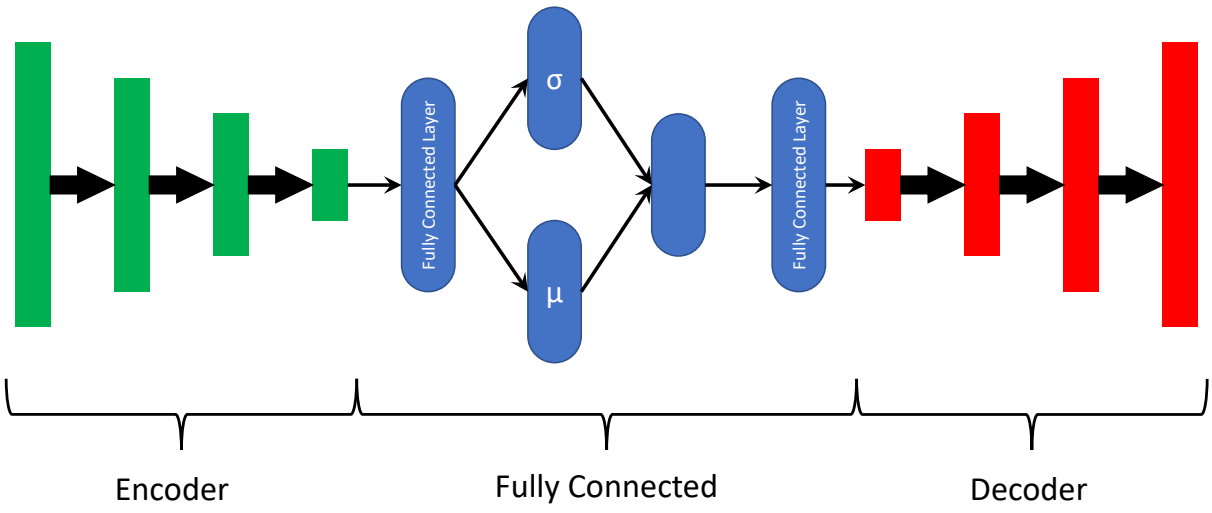


Figure 5-3 VAE Architecture

The main difference between the Variational Autoencoder and a standard Autoencoder is that the latent spaces are continuous, which allows easy random sampling and interpolation. This is achieved by having the output of the encoder consist of two vectors: standard deviation σ, and mean, μ which form a vector of random variables, which is then passed into the decoder. The mean vector centres the encoding zone, and the standard deviation controls the area. The encodings can occur anywhere within this zone, as the decoder learns to not only focus on a single point but that nearby points can all be representative.

An improvement in controlling the latent space is achieved by using a conditioning vector before and after the fully connected layer. The fully connected layer is a layer within the neural network in which each input node is connected to each output node. A Conditional VAE (CVAE) uses a one-hot vector, an n-bit binary vector with a 1 in a particular position that signifies the variable to condition. For example, if the numbers were between 0 and 9, a one in position 4 would signify the number was

a 5. The one-hot vectors are appended to the feature matrix before the fully connected layer in the encoder side and appended before the fully connected layer in the decoder side, as shown in Figure 5-4.

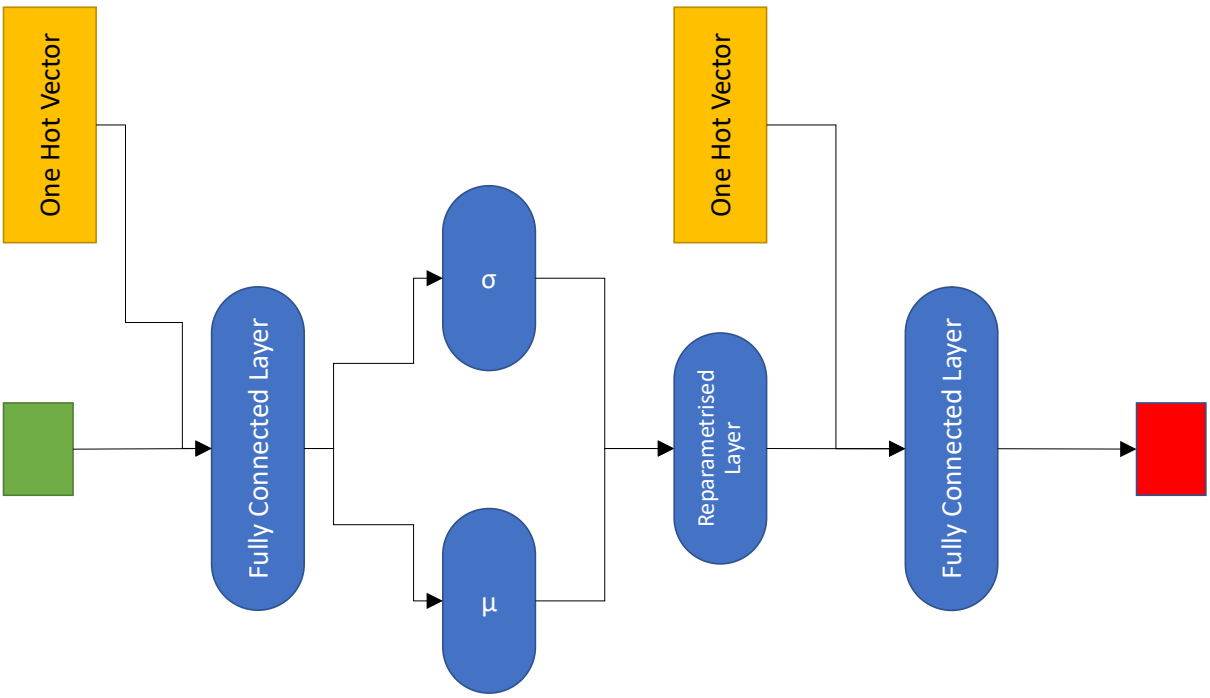


Figure 5-4 Conditioning Variables in the VAE Architecture

Once the latent space has been conditioned then it becomes easier to create a mapping between different latent spaces.

The Autoencoder incorporates the Kullbeck Leiber Divergence (KLD) within the loss function as a method optimising the latent space embedding. The KLD between two Gaussian functions is shown in equation 5.3.

$$\sum_{i=1}^{1} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1 \qquad \textbf{5.3}$$

Where σ and μ are the standard deviation and mean respectively of a gaussian distribution. In a VAE, the KL loss is equivalent to the sum of all the KLD between the component and the standard normal (when σ=1 and $\mu = 0$).

5.1.4 Introduction to Maximum Mean Discrepancy

Maximum Mean Discrepancy (MMD) was proposed in (Gretton *et al.*, 2006) and is based on the assumption that two distributions are the same if and only if the moments are the same. MMD defines the divergence as the difference between the moments of two distributions using the kernel embedding techniques shown in equation 5.4.

$$\begin{aligned} &\mathrm{MMD}\big(\mathrm{p}(\mathrm{z}) \parallel \mathrm{q}(\mathrm{z})\big) \\ &\qquad = \mathrm{Ep}(\mathrm{z}), \mathrm{p}(\mathrm{z}')[\mathrm{k}(\mathrm{z}, \mathrm{z}')] + \mathrm{Eq}(\mathrm{z}), \mathrm{q}(\mathrm{z}')[\mathrm{k}(\mathrm{z}, \mathrm{z}')] \\ &\qquad - 2\mathrm{Ep}(\mathrm{z}), \mathrm{q}(\mathrm{z}')[\mathrm{k}(\mathrm{z}, \mathrm{z}')] \end{aligned} \qquad \textbf{5.4}$$

Where $\mathrm{k}(\mathrm{z}, \mathrm{z}')$ is described as the universal kernel and is often represented using the Gaussian Kernel. The purpose of this kernel is to measure the similarity of the two samples by having a large value if they are similar and low values for dissimilar. MMD works by comparing the average similarity between samples from each distribution and the average similarity between mixed samples from both distributions. If these values are identical then the distributions are the same.

The reason for incorporating MMD in the model is to solve the following two problems:

1. Uninformative latent code – it has been observed (Chen *et al.*, 2017) that it is possible that the AE has failed to learn a meaningful representation of the input.
2. Over Estimation in the Feature Space – VAE's can over-fit data which leads to encoder variance tending to infinity.

5.2 Introduction to Generative Adversarial Networks

As highlighted in the literature review, the Generative Adversarial Network (GAN) was first conceived in 2016 (Goodfellow *et al.*, 2016). GANs fall into the category of unsupervised learning models, similar to the CVAE in the previous section.

The architecture of a GAN consists of two parts, the first is the generator, the second is a discriminator, with the latent space as shown in Figure 5-5. The GAN operates by using an adversarial process to train the generator. The GAN works by training the generator to increase the probability of causing the discriminator to make a mistake.

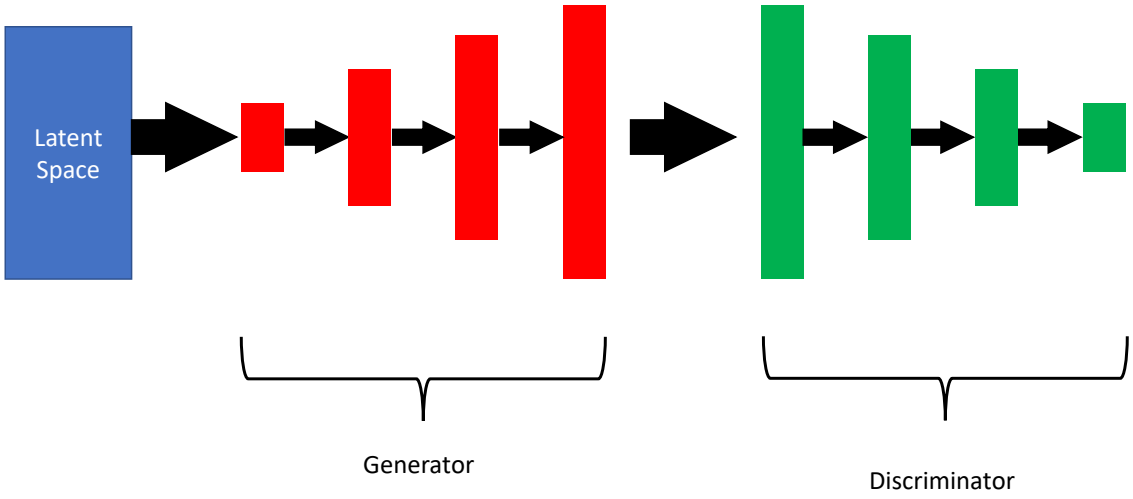


Figure 5-5 GAN General architecture

An example of the training process for a GAN is shown in Figure 5-6. The first step is to take the real data and train the discriminator so that the output is a “1” for true. The second step is to train the discriminator on all fake data so that the output is a “0” for false. Once the discriminator has been trained to identify the real and fake data, the third step is to train the whole network. This is done by using fake data in the generator but setting the label to a “1”.

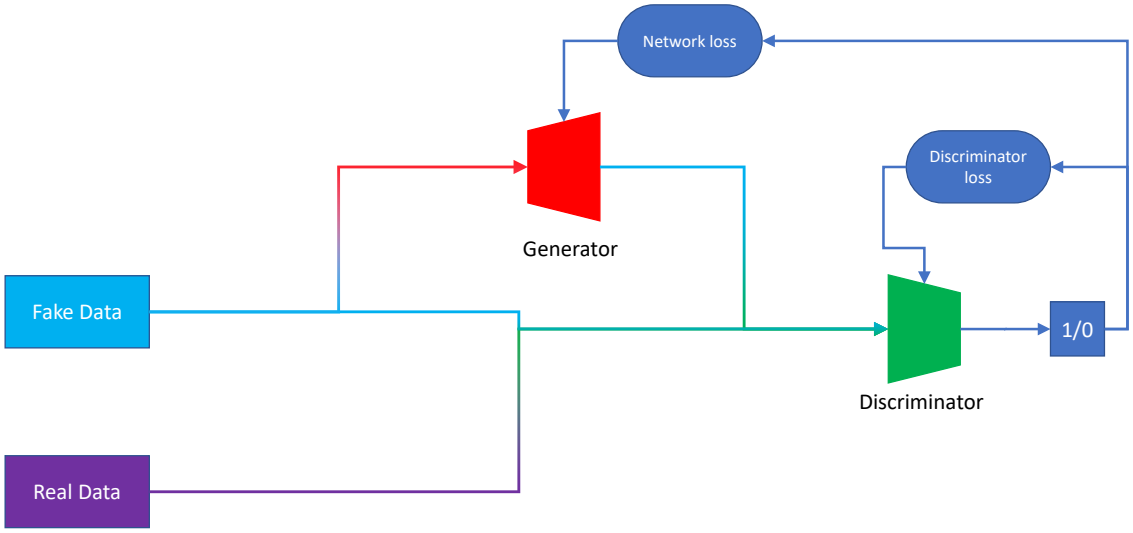


Figure 5-6 GAN Training Process

Early GANs had several disadvantages. The main disadvantage is that the discriminator must be synchronised with the generator during training. This means that the generator cannot be overtrained without updating the discriminator as this can result in the generator collapsing the latent space. The main advantage over other adversarial networks is that back propagation is sufficient to obtain the gradients, compared with Markov chains required with other adversarial networks, which results in computational improvements. The main advantage of adversarial networks over unsupervised networks that rely on the use of Markov Chains is that the network generates clear outputs, whereas the outputs from a Markov chain are blurrier.

5.2.1 Understanding the Details of a GAN

This section provides more details of GANs with the aim to make it clearer to the reader how and why GANs were used to develop the Data Driven solution. As has been stated, there are two models within the GAN: the discriminator and the generator, and hence two different loss functions. The key to understanding how GANs work is to understand how these loss functions behave.

5.2.2 The Discriminator Loss Function

As stated in the introduction of this section, the sole purpose of the discriminator is to identify the output from the generator as either real ('1') or fake ('0') and hence the loss function can be described as shown in 5.5 below.

$$L_D = \mathbb{E}(D(x), 1) + \mathbb{E}\big(D\big(G(z)\big), 0\big) \quad \textbf{5.5}$$

Where:

- D(x) is the discriminator evaluation of real 'x'
- D((G(z)) is the discriminator evaluation of fake 'x'
- $\mathbb{E}$ is the error

The error function is a way of measuring the distance or difference between the two distributions, similar to the Kullback-Leibler Divergence criteria discussed earlier.

The most common loss function employed by the discriminator is Binary Cross Entropy (BCE), which is described in Equation 5.6.

$$H(p, q) = \mathbb{E}_{x \sim p(x)}[-logq(x)] \quad \textbf{5.6}$$

In terms of a classification application, it can be expressed as:

$$H(p, q) = -\sum_{x \in \chi} p(x) logq(x) \quad \textbf{5.7}$$

Since there are only two labels in a binary classification, problem 5.7 can be written as:

$$H(y, \hat{y}) = -\sum ylog(\hat{y}) + (1 - y)\log\,(1 - \hat{y}) \quad \textbf{5.8}$$

Equation 5.8 is the error function mentioned in 5.5.

BCE measures how different two distributions are in the context of binary classification, as well determining if the input is true or false. This allows the loss

functions to be derived by inserting into 5.5, which gives the loss function required to train the discriminator in equation 5.9.

$$L_D = -\sum_{x\in\chi, z\in\zeta} log\left(D(x)\right) + \log\left(1 - D\big(G(z)\big)\right) \quad \textbf{5.9}$$

5.2.3 The Generator Loss Function

The next loss function to consider is for the generator. As previously stated, the primary aim of the generator is to create believable fake data, such that the discriminator misidentifies the input and wins the adversarial game (as discussed in the introduction of this section). The loss function employed is shown in 5.10 below.

$$L_G = \mathbb{E}\big(D\big(G(z)\big), 1\big) \quad \textbf{5.10}$$

Where D(G(z)) is the discriminator's evaluation of fake data. It should be noted that the aim of the generator is to try and minimise the loss between real data '1' and the fake data which is labelled as '1'. Using the BCE function approach outlined in section 5.2.2 and applying to 5.10 gives the following loss function for the generator in equation 5.11:

$$L_G = -\sum_{z\in\zeta} \log\, D\big(G(z)\big) \quad \textbf{5.11}$$

Which results in a small loss function when the output is close to 1 (log (1) = 0), or in other words, the better the generator becomes at faking the results, the lower the loss is.

In some papers, the loss function for the discriminator has slight variations. The only difference is whether the aim is to minimise, as shown in 5.9, or by removing the negative sign at the beginning of the loss function, the aim is to maximise the loss function (Goodfellow *et al.*, 2016). The important point to note is that in (Goodfellow

et al., 2016), the game is framed as a min-max one, where the discriminator is aiming to maximise the reward, while the generator is aiming to do the opposite. This approach leads to a one-line equation which demonstrates the nature of the adversarial competition. In the real world though, it is better to define separate loss functions for the generator and discriminator.

Now that the loss functions have been described the next stage is to explain how the GAN is trained.

5.2.4 Training

The training of the GAN is done in stages. The first stage is to train the discriminator with real and fake labels while fixing the generator, then the generator is trained while fixing the discriminator. The quantities of interest in the discriminator are how well the generator (G) is working and the discriminator (D) in equation 5.12.

$$V(G,D) = \mathbb{E}_{x\sim Pdata}\, log\left(D(x)\right) + \mathbb{E}_{z\sim P_g} \log\left(1 - D(G(z))\right) \quad \textbf{5.12}$$

The aim of the discriminator is to maximise the integral of this function, and as a result the optimal value discriminator is achieved when equations 5.13 and 5.14 are:

$$\frac{pdata(x)}{D(x)} - \frac{p_g(x)}{1 - D(x)} = 0 \quad \textbf{5.13}$$

$$D(x) = \frac{pdata(x)}{pdata(x) + p_g(x)} \quad \textbf{5.14}$$

It follows that, when a sample (x) is genuine, then $pdata(x)$ is 1 and $p_g(x)$ is 0, and in the case of a generated data point the discriminator should be 0.

The next stage is to look at training the generator, this is done by fixing the discriminator, and then combining 5.12 and 5.14 to give equation 5.15:

$$V(G,D) = \mathbb{E}_{x\sim Pdata}\, log\left(\frac{pdata(x)}{pdata(x)+p_g(x)}\right) + \mathbb{E}_{x\sim P_g}\log\left(\frac{p_g(x)}{pdata(x)+p_g(x)}\right) \quad \textbf{5.15}$$

Rewriting this using log identities gives equation 5.16:

$$V(G,D) = \mathbb{E}_{x\sim Pdata}\, log\,(pdata(x)) - \log\left(pdata(x)+p_g(x)\right) + \mathbb{E}_{x\sim P_g}\log\left(p_g(x)\right) - \log\left(pdata(x)+p_g(x)\right) \quad \textbf{5.16}$$

It is possible to write 5.16 in the form of the Kullback-Leibler divergence by dividing $\log\left(pdata(x)+p_g(x)\right)$ by 2 and taking the log(4) out,

$$V(G,D) = -\log(4) + \mathbb{E}_{x\sim Pdata}\, log\,(pdata(x)) - \log\left(\left(pdata(x)+p_g(x)\right)/2\right) + \mathbb{E}_{x\sim P_g}\log\left(p_g(x)\right) - \log\left(\left(pdata(x)+p_g(x)\right)/2\right) \quad \textbf{5.17}$$

Equation 5.17 can be rewritten as equation 5.18.

$$V(G,D) = -\log(4) + DKL\left(Pdata\|\frac{pdata+p_g}{2}\right) + DKL\left(p_g\|\frac{p_g+p_g}{2}\right) \quad \textbf{5.18}$$

Which means that the goal of the generator is to minimise the function V(G, D) so that the difference between the real and fake data has to be as small as possible, or in other words P_g should be as close to P_{data} as possible.

5.2.5 Mode Collapse in GANs

GANs pose a number of challenges. One of the significant issues with them is the concept of mode collapse. Mode collapse is where the generator is able to produce a single output, or a small set of outputs, for example if trained to produce numbers

on the MNIST dataset the generator only generates the number ‘8’ as shown in Figure 5-7. The reason for this is often found during training, where the generator finds a particular set of data that is able to fool the discriminator and, since in the GAN algorithm there is no incentive to change from this space, the generator will optimise around that given space. This over-training results in random output.

Figure 5-7 An Example of Mode Collapse in a GAN (Shen, 2020)

5.2.6 Latent Space

A key parameter of the neural networks used in this research is the latent space. The latent space is the mathematical region to which the learned parameters are mapped. This space is often multidimensional, however an example of a 2D latent space is shown in Figure 5-8, where each colour blob represents a collection of common features, for example the orange blobs are the number 4, the green blobs are the number 3, blue are the number 2, and red blobs are the number 1. Inevitably there will be some blobs that encroach on each other’s learned space which accounts for misrepresentations.

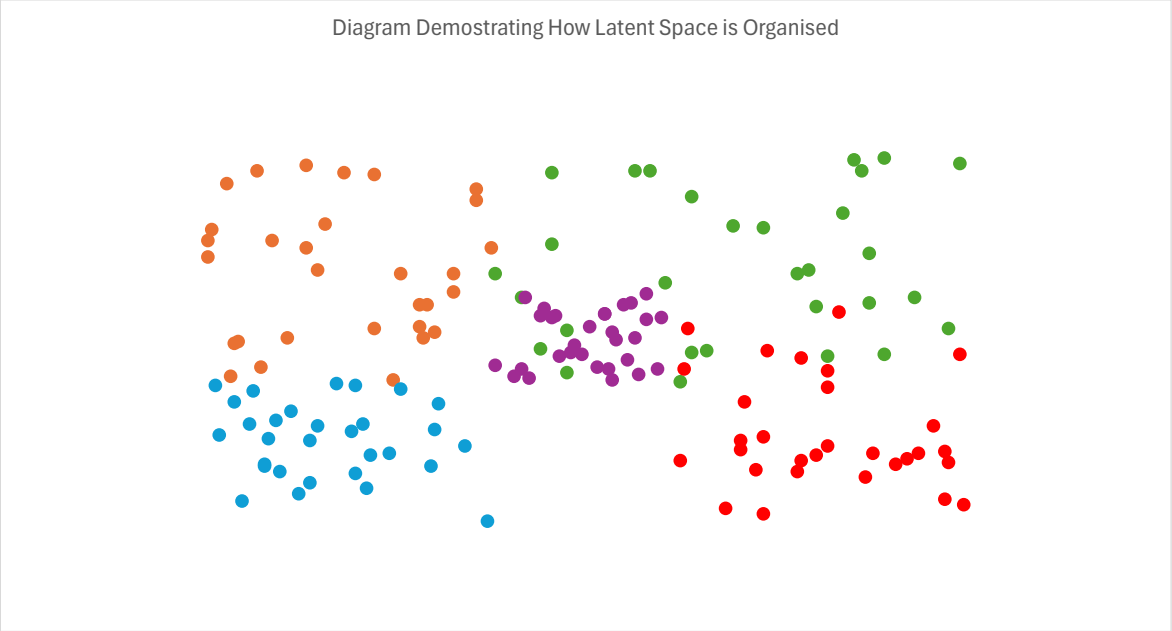


Figure 5-8 Diagram of the Learned Latent Space from Unsupervised Model

Now that the core features of the model design and underlying concepts are explained, the next sections will discuss the outputs of the computational model.

5.3 Conversion of GPR signal to images

As mentioned in Chapter 3 the aim of this thesis is to find the real-world objects that create GPR response curves. This can be done by either solving a system of linear equations as suggested in section 3.1 or by using a data driven model. The approach described in this section is to adopt the data driven model by using the GPR output and physical models, convolve them into two different latent spaces and create a mapping between the two zones. These latent spaces have high dimensionality, 100 in this case, and have been conditioned on the material types in the model. The idealised form of this model is shown in Figure 5-9.

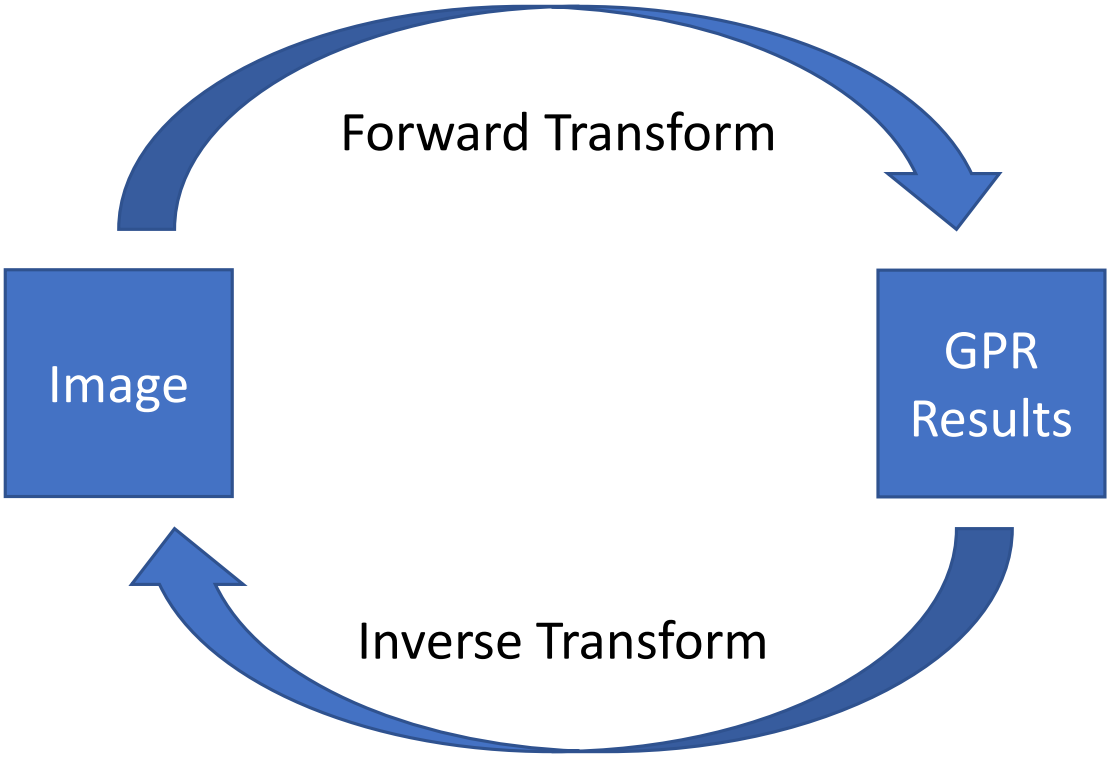


Figure 5-9 The Idealised Proposed Forward and Reversed Mapping System

Developing the Data Driven solution required multiple experiments and iterations of the computational model. During the iterative process it was often found that the output from the models would be random noise and not show any detail of required images which sometimes failed to converge onto a solution, or the models would exceed the GPU memory limits. The first version which showed convergence was Version 1, and then after further refinement on the training protocols and layer details Version 5 showed an improved output, both of these versions will be described in detail in this section.

5.3.1 2D to 2D Cycle GANs

The cycle GAN as proposed by (Zhu *et al.*, 2017) and described in section 2.1.3 was conceived as an image-to-image translation system. The mapping is under constrained and as a result a second mapping is performed for the inverse process. This results in the requirement for two GANs to perform the mapping between two inputs, the first will map from GPR -> Simulation Image -> GPR and the second will map from Simulation Image -> GPR -> Simulation Image. As shown in Figure 5-1, this results in the creation of mappings between two latent spaces of the two GANs.

In this application, the cycle GAN uses two CVAEs which has the advantage of being able to organise or condition the latent spaces based on a user defined criterion. In this case, the latent spaces will be organised by the material in the model. The conditions are shown in Table 5-1. The final column shows the decimal value of the conditional variable, which can be translated into a “one-hot vector”, such that the position in the vector relates to the condition. For example, if the input consisted of only organic objects, the condition variable would be 5, which would then translate to a vector [0,0,0,0,1,0,0] and the Inorganic 1 and Inorganic 2 is 3, which would be [0,0,0,1,0,0,0,0].

Physical Model	Active = 1 Inactive =0			Numeric
No material	0	0	0	0
Inorganic 1	0	0	1	1
Inorganic 2	0	1	0	2
Inorganic 1 & Inorganic 2	0	1	1	3
Organic	1	0	0	4
Organic & Inorganic 1	1	0	1	5
Organic & Inorganic 2	1	1	0	6
Organic, Inorganic 1& Inorganic 2	1	1	1	7

Table 5-1 Conditional Variables

5.3.1.1 Version 1 Architecture and Training

The first part of the model is to incorporate two CVAEs sequentially with a discriminator on the output of the first CVAE, which also feeds the input into the second CVAE. These latent spaces have been defined by CVAEs mentioned in section 5.2.6. The architecture adopted is shown in Figure 5-10. The Kernels are shown in Figure 5-11 and Figure 5-12. The decoder side of the model uses the same parameters as the encoder except to invert the convolutions. The fully connected layers are 32,768 to 512 with a latent dimension of 100.

The first version of the approach only used two CVAEs in each GAN, and the discriminator runs off the output from the first cycle GAN. The model was trained for 250 epochs over 3 days on a windows 10 PC using a 1080Ti Graphics card with 11GB memory. An epoch in machine learning terms is used to describe a single pass through the entire training dataset and the corresponding update of the weights using back propagation. The data set was 15,000 simulated scans. Version 1 of the model was conditioned on the number of objects and not the type of objects and not their types. At this point there were no distinctions made about object type.

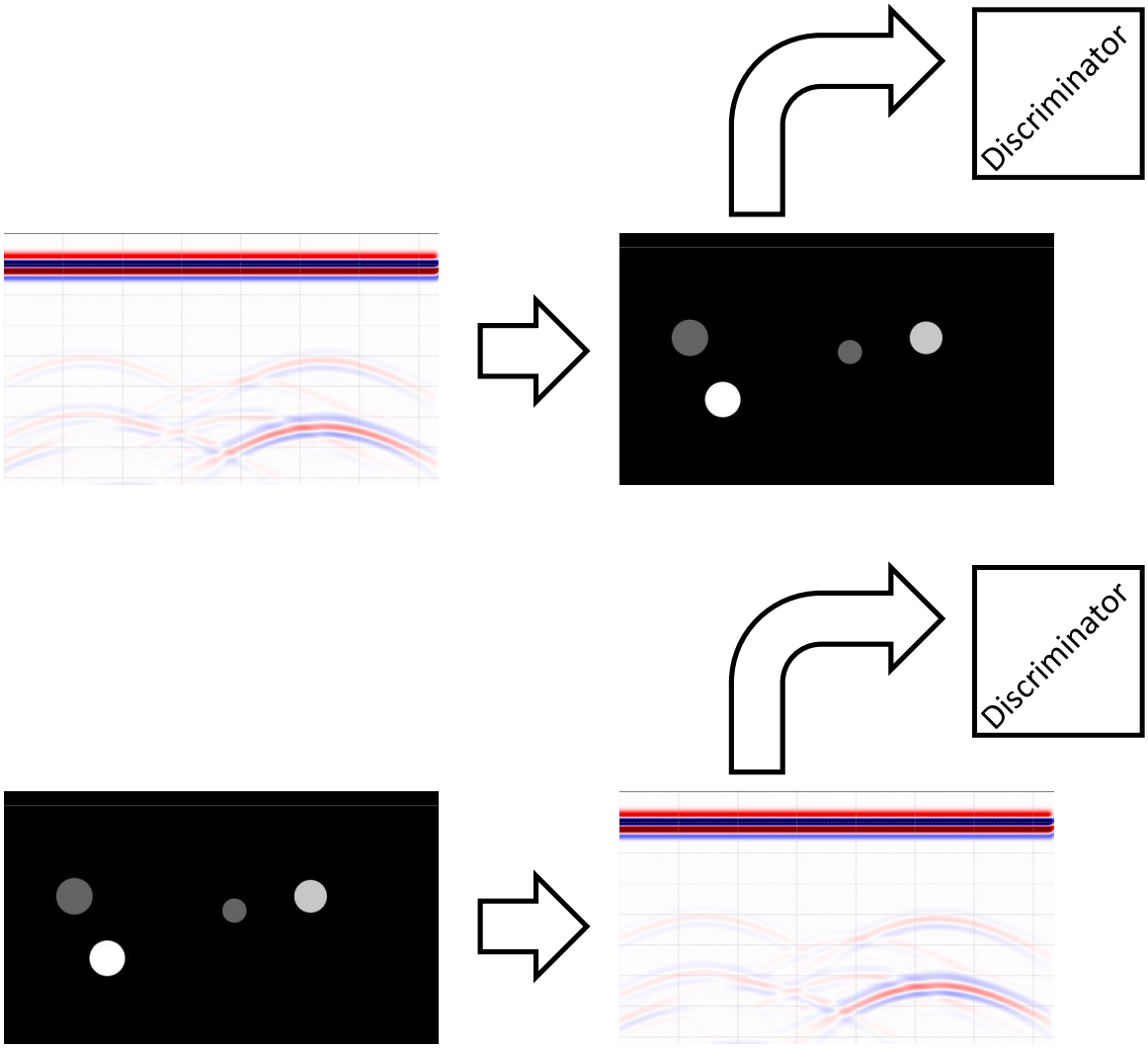


Figure 5-10 Version 1 Architecture Approach

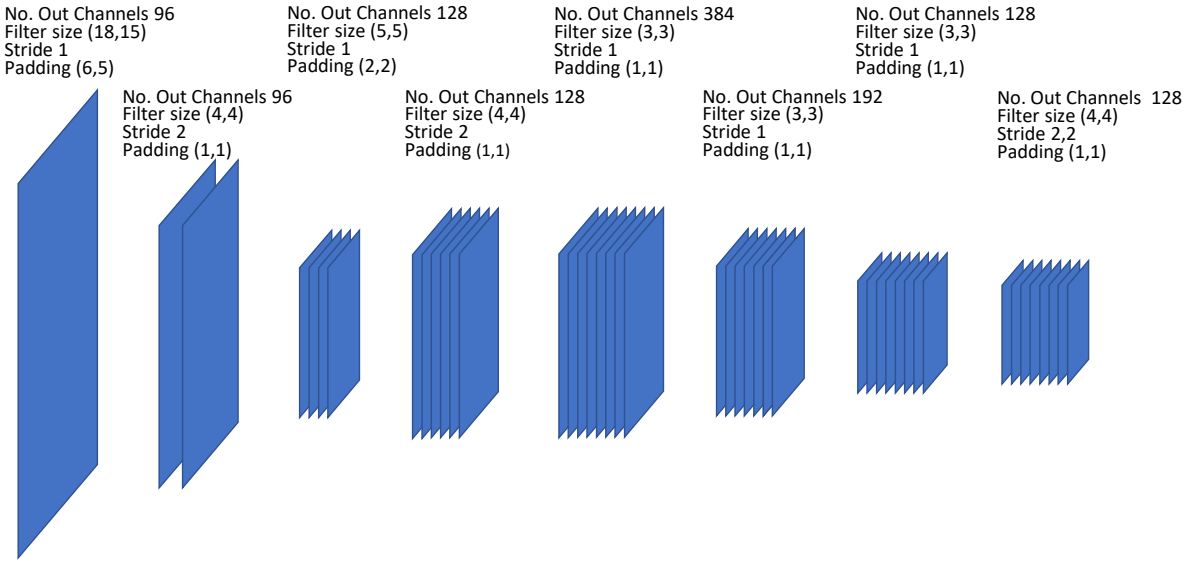


Figure 5-11 Version 1 Convolution Kernels for the GPR CVAE

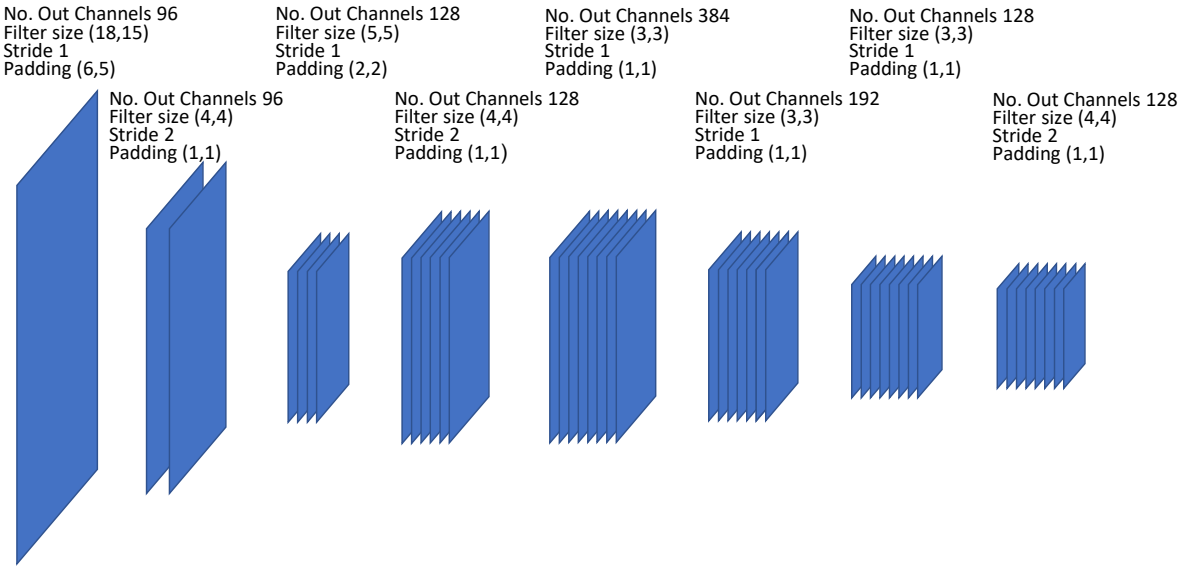


Figure 5-12 Version 1 Convolution Kernels for the Image CVAE

5.3.1.2 Version 5 Training and Architecture

Following from Version 1, several iterations of the model were attempted, each one aiming to improve the quality of the output by increasing the complexity of the model. This activity converged in the 6th iteration of this version which adopts a more traditional Cycle GAN Approach, as shown in Figure 5-13. The model uses the two CVAEs with the Convolution Kernels shown in Figure 5-14 for the GPR and Figure 5-15 for the Image processing. The Decoder modules use the same parameters as the encoder to perform the transpose convolutions. The fully connected layers are 4096 to 1024 with a latent dimension of 100. The training method for this version is shown in Figure 5-16.

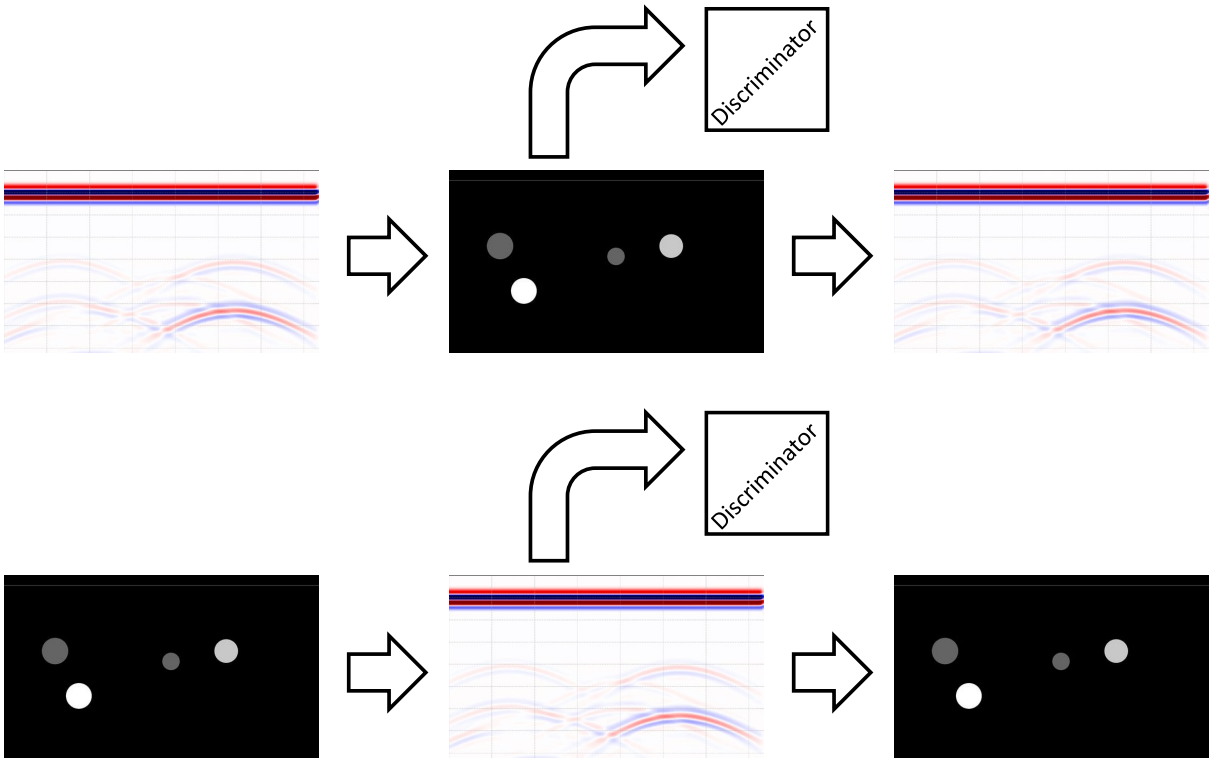


Figure 5-13 Version 5 Cycle GAN Mapping System used

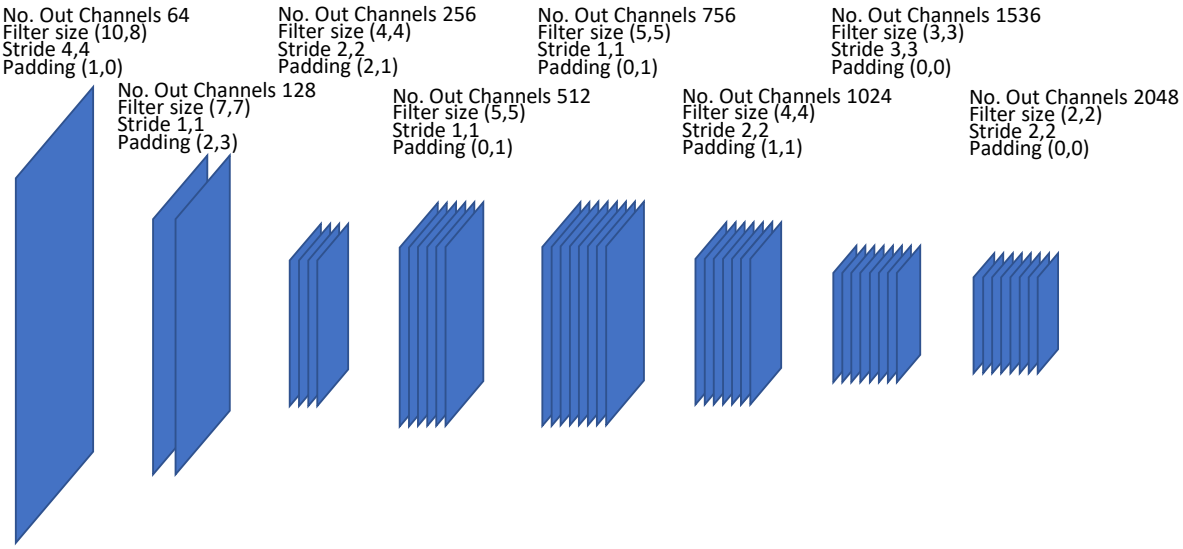


Figure 5-14 Version 5 Convolution Kernels for the GPR CVAE

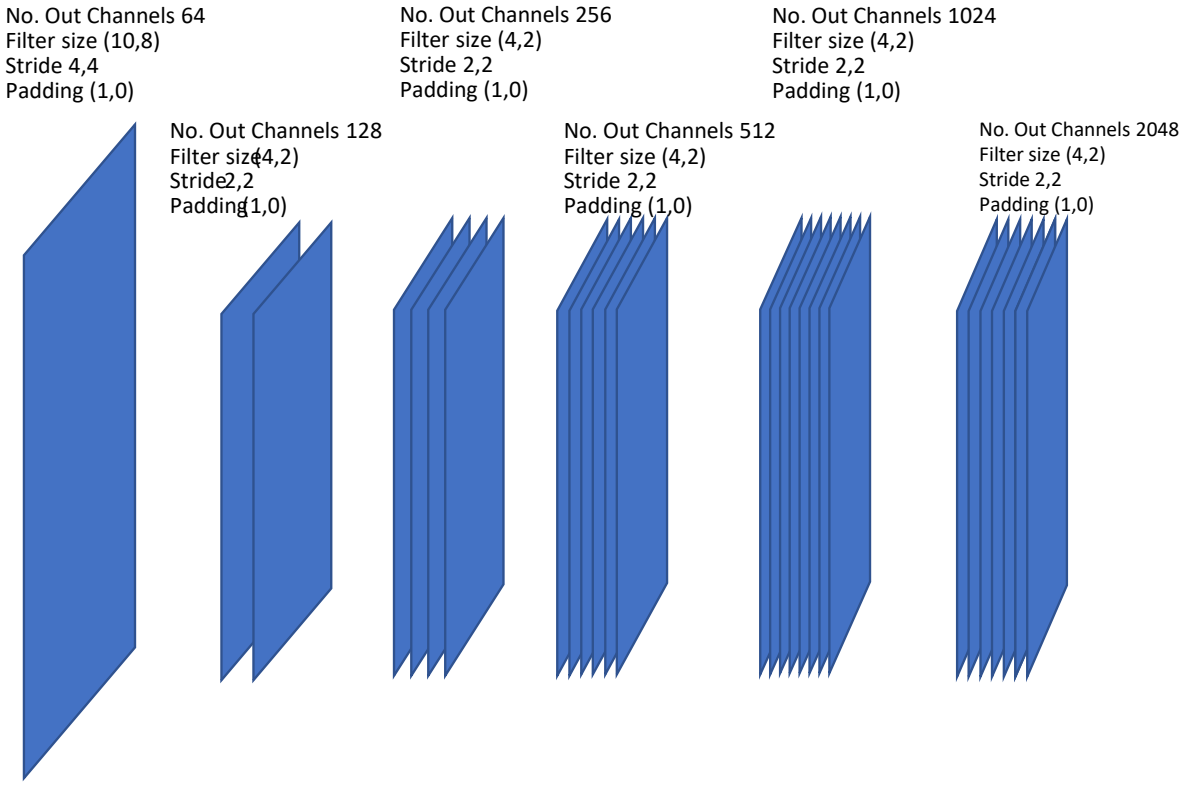


Figure 5-15 Version 5 Convolution Kernels for the Image CVAE

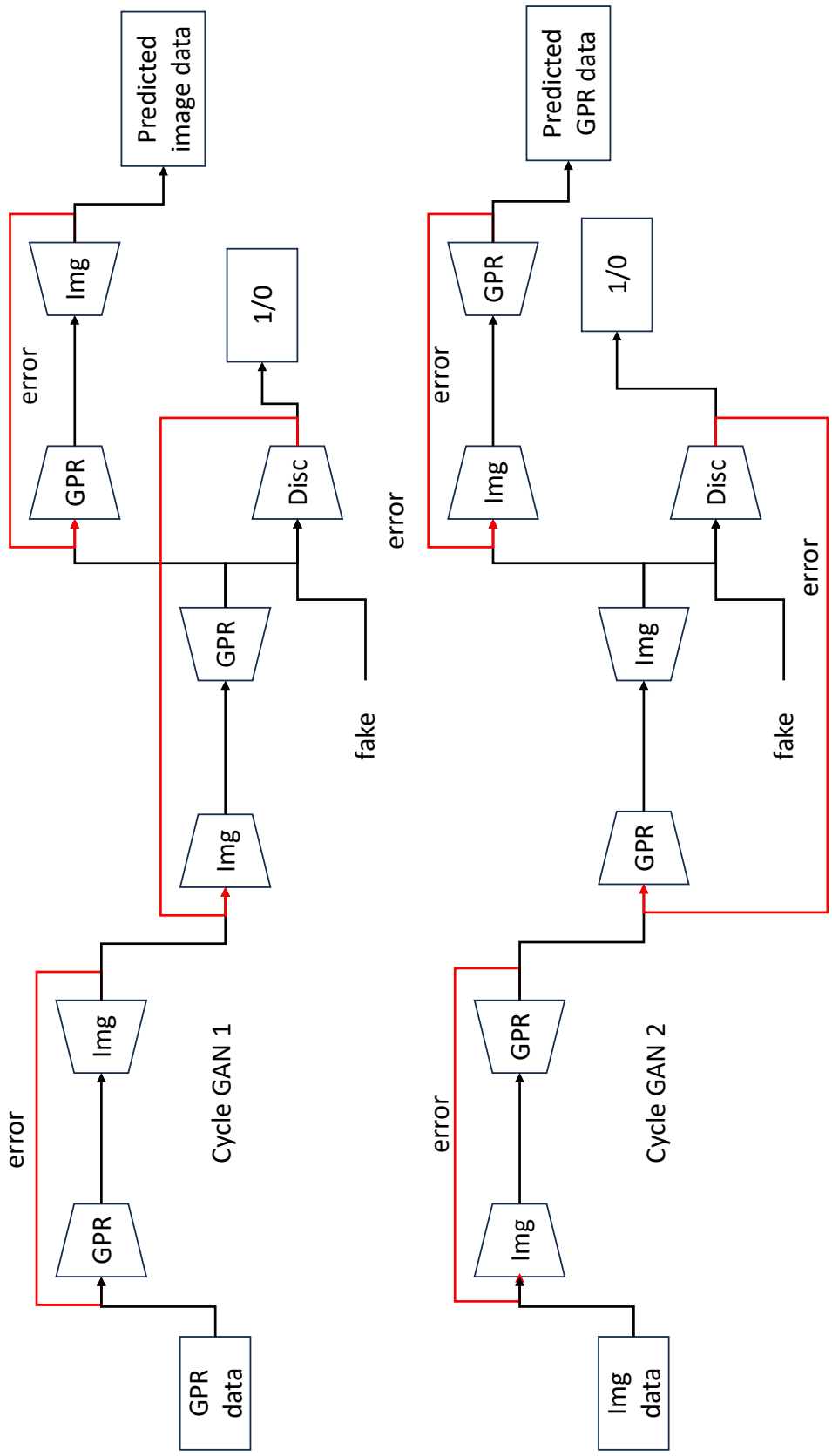


Figure 5-16 Training Philosophy for the 2D-2D cycle GAN's Showing Both GAN1 (GPR to Image) and GAN2 (Image to GPR)

5.3.2 2D Results

The following section discusses the results from the first successful version of the software written for this project and the latest version. The total size was 15,998 images and simulated B-Scan results, and the batch size 10. The Version 1 models were run for 250 epochs, and the Version 6 models were run for 300 epochs.

5.3.2.1 Version 1

The results at the end of training are shown in Figure 5-17. The first image shows the GPR simulation input into the network, the second image shows the “mapped” output and the final image shows the actual simulation model. Overall, when visually reviewing the results, it seems that the number and position of the objects demonstrates a good correlation, however the level of noise in the background needs to be addressed since it prevents an accurate approximation of the corelation to be made. An in-depth review into the images will be performed on the testing images shown in Figure 5-18

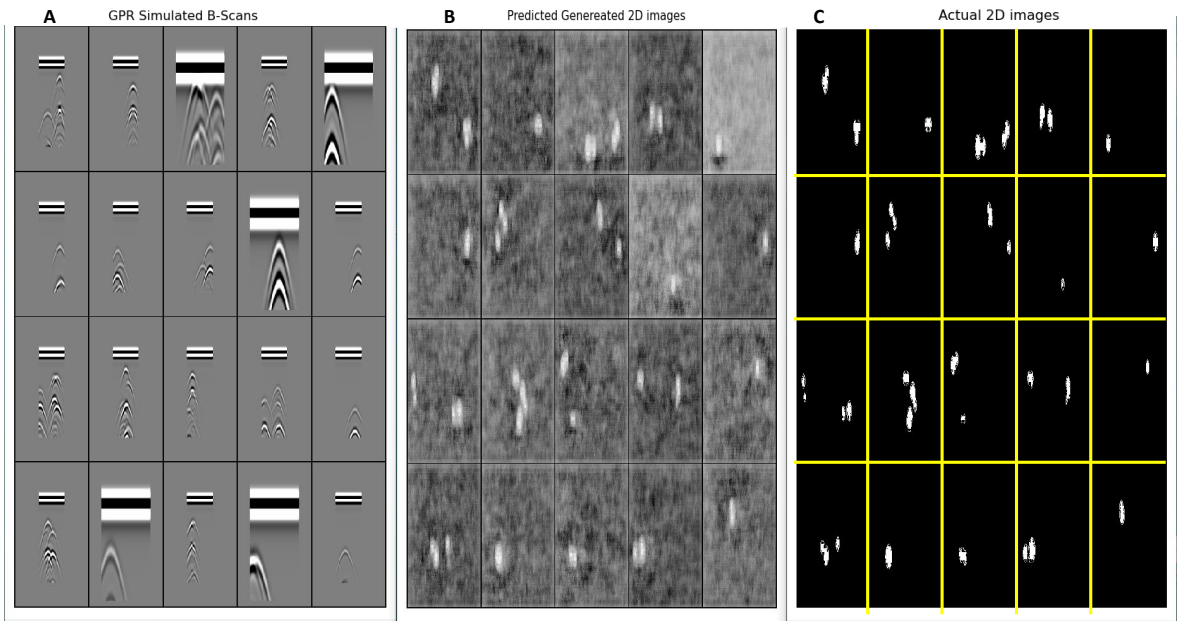


Figure 5-17 Training Data after 250 Epochs (Image ‘A’ is the Input, Image ‘B’ is the Mapped Output and Image ‘C’ is Simulation Output)

While Figure 5-17 shows there is a good correlation between the prediction and the data used during training, and that the model is learning latent space effectively, Figure 5-18 shows that the network performs well when previously unseen data are used, that is data that are separate to the training set. To allow the visual comparisons to be made, the matching values have been highlighted with yellow ellipses. Although a few locations are not mapped accurately, this shows that the model behaves as expected.

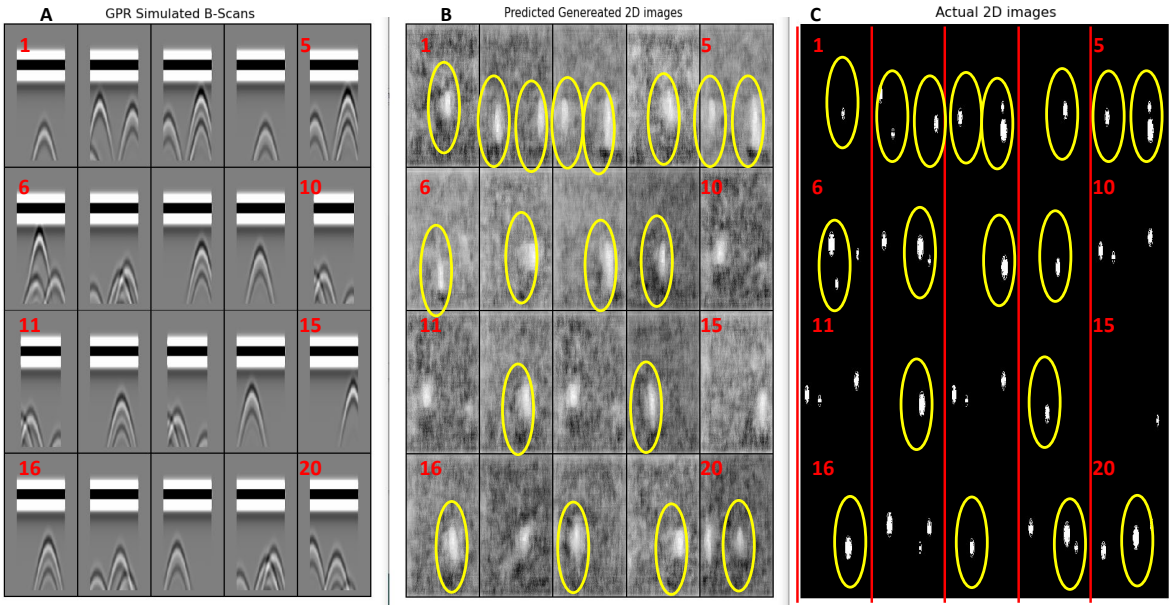


Figure 5-18 Testing Data after 250 Epochs (Image 'A' is the Input, Image 'B' is the Mapped Output and Image 'C' is Simulation Output)

Taking images 1 and 15 in the sequence shown in Figure 5-18 and enlarging in Figure 5-19 shows the following observations can be made:

1) The position and clustering are consistent between the generated image 1 and the actual image 1.
2) In the generated image there a significant amount of background noise which is detrimental to the image usefulness.

3) In image 15 the generated image is too obscured by the noise level to prove any useful comparisons; however, the image does not look consistent with the actual image in shape, position looks reasonable though.

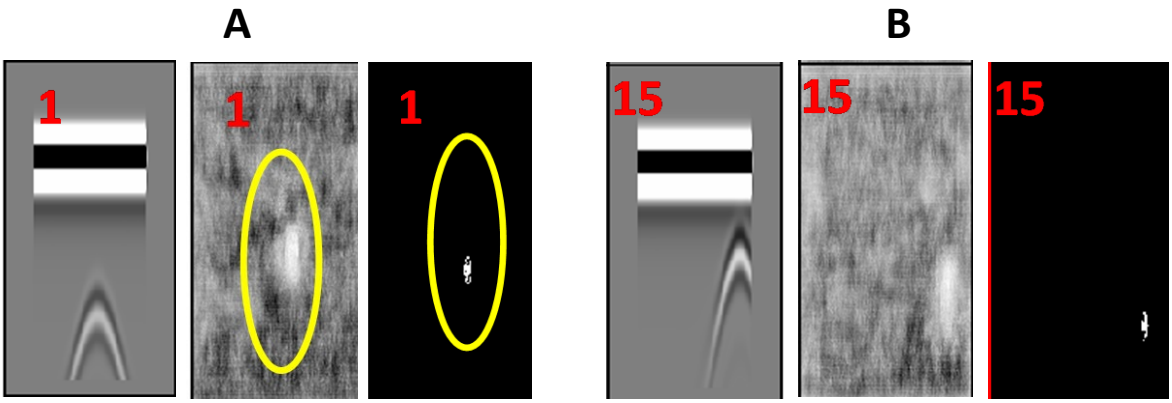


Figure 5-19 Enlarged image showing in image 'A' the details of testing data for a good match and in Image 'B' poor match

5.3.2.2 Version 5

Based on the results above, it was obvious that significant improvements had to be made in order to demonstrate that the mapping is possible. Several iterations were performed until the results shown in Figure 5-20 and Figure 5-21 were created. There are some changes to results, the first being that the GPR plots have been normalised in a slightly different manner to help the training process. The data is normalised as a method of improving both the performance and stability during the training process. In this update the data normalised to between 0 and 1 and A-Scan field strength values limited to 4000.

In order to simplify the GPR image, the grey background used in the previous dataset was switched to be white, this has the benefit of reducing the number variables which the model has to optimise. In addition, this allowed the data to be much more easily normalised to between 0 and 1.

The improvements have resulted in a reduction of training time from 250 epochs to just 100 epochs. The most striking difference between the version 1 and version 5 results is the background noise was eliminated due to the optimised parameters in version 6's results.

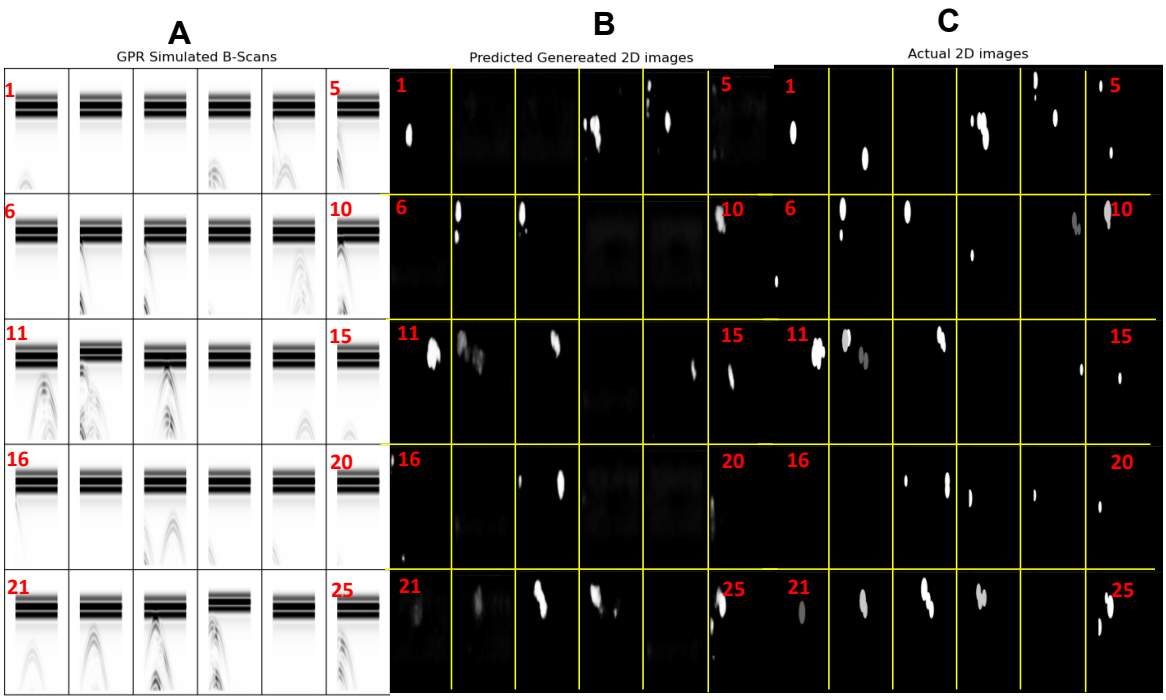


Figure 5-20 Training results after 100 Epochs (Image 'A' is the GPRMax Input, 'B' Shows the Mapped Output and 'C' is the Target Output)

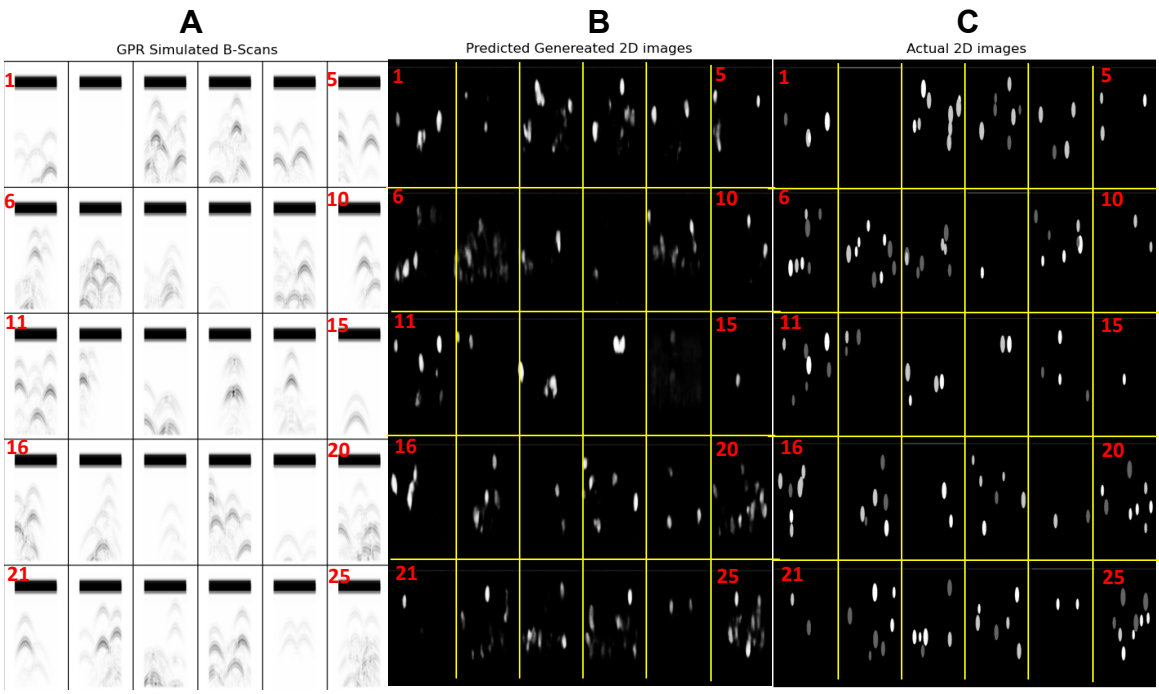


Figure 5-21 Testing results after 100 Epochs (Image 'A' is the GPRMax Input, 'B' Shows the Mapped Output and 'C' is the Target Output)

The results above are assessed using 3 metrics, these are:

1. Pixel Values
2. Mean Squared Error
3. Intersection over Union

In the following sections the term ground truth. In this context ground truth is the reality the model knows and covers both the input and testing data. The generated data is then compared to this data as a method of comparing the accuracy of the output. In the context of this project, the term is applied to the actual images which the machine learning generated images should match.

Using the actual mapped pixel values and comparing the values to the ground truth data gives an indication of the accuracy achieved by the mapping, since there are three different colours in the grey scale (two grey values and one white).

Mean Squared Error (MSE) is used in statistics to measure the average squared difference between estimated and real values. MSE is often used to measure the quality of the estimator, since it is always positive and tends to zero as the accuracy increases the equation is shown in 5.19.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad \textbf{5.19}$$

When the pixel values in Figure 5-21 are compared with the predicted and target outputs it is found that 42% of the actual values are within ±15% of the target for the training data, while 25% are within the target for the testing data. The black background has been ignored in this calculation. While the value is low, this is not as bad as it seems because the value calculated is based on the pixel value and not the location of the pixels, which is a very harsh metric. The mean square errors

between the actual and generated images are shown in Table 5-2. The ideal target for the MSE value is 0, however this is rarely achieved and as a result the metric should be viewed as a comparator instead of in absolute terms. This means that the smaller the number the better corelation between the ground truth and predicted results, while larger numbers indicate poor corelation. In Table 5-2 the training data has a standard deviation of 673.2 and mean of 717, while the testing data has a standard deviation of 548.4 and a mean of 965. Comparing these values suggesting that the testing generated images are closer together, however the this is around a lower accuracy (this is shown Figure 5-22 below).

Training Data

Image Number	MSE
0	932
1	942
2	14
3	2357
4	1048
5	469
6	241
7	1170
8	942
9	242
10	160
11	720
12	2383
13	1076
14	1099
15	14
16	243
17	238
18	14
19	14
20	1083
21	383
22	201
23	243
24	158
25	781
26	1921
27	866
28	14
29	1535

Testing Data

Image Number	MSE
0	1078
1	145
2	2572
3	1025
4	922
5	650
6	1249
7	1625
8	766
9	307
10	1205
11	306
12	1475
13	267
14	1254
15	804
16	632
17	226
18	1457
19	1213
20	858
21	1035
22	280
23	1367
24	241
25	1351
26	1384
27	1320
28	507
29	1433

Table 5-2 Showing the Mean Squared Error for Training Data and Testing Data

Figure 5-22 Comparing the Normal Distributions of the MSE for Training and Testing Data

Intersection over Union (IoU), this metric measures the accuracy of an object detector and is often used as a method to evaluate the performance of R-CNNs. Using two bounding boxes, one for the ground truth and the second for the prediction results, this allows the intersected areas to be compared. For example, in the left image in Figure 5-23, the actual bounding box is in green, while the predicted box is in red and the simple ratio of the two areas is calculated using equation 5.20:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \qquad \textbf{5.20}$$

Where the Area of Overlap (middle image of Figure 5-23) is the area of overlap of the two boxes, while the Area of Union is the Total area denoted by the perimeter of the two boxes right hand image of Figure 5-23.

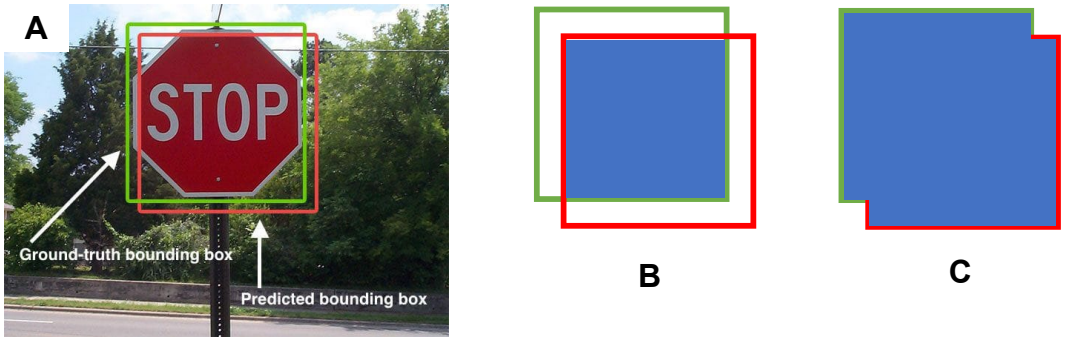


Figure 5-23 Example Showing the Two Bounding Boxes in Image 'A', Image 'B' is the Area of Intersection and Image 'C' Shows the Area of Union (Anon, 2023)

image	object 1	object 2	object 3	object 4
0	0.9	1.0		
1	0.0	1.0		
2	0.0			
3	0.5	0.9	0.0	
4	0.0	0.0	0.0	0.0
5	0.3	0.0	0.0	
6	0.3	1.0		
7	0.0	0.0	0.0	
8	0.0	0.0		
9	0.3	0.0		
10	0.0	0.0		
11	0.5	1.0		
12	0.9	1.0		
13	0.0	0.2	1.0	
14	0.0	0.0		
15	0.0			
16	0.0	0.0		
17	0.2	1.0		
18	0.0			
19	0.0			
20	0.0	0.0	0.0	
21	0.0	0.0		
22	0.0	0.0		
23	0.2	1.0		
24	0.3	0.0		
25	0.0	0.0		
26	0.8	1.0		
27	0.0	0.0		
28	0.0			
29	0.0	0.0	0.0	

Table 5-3 The IoU details for selected images for the Training dataset

image	object 1	object 2	object 3	object 4	object 5	object 6	object 7	object 8	object 9	object 10
0	0.4	0.6	0.6	1.0						
1	0.0									
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
4	0.1	0.0	0.7	0.8	1.0					
5	0.0	0.0	0.0	0.0						
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.3	1.0			
9	0.4	1.0								
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
11	0.0	0.0	0.0							
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
13	0.0	0.0	0.6	0.0						
14	0.0	0.3	0.0	0.0						
15	0.3	0.0								
16	0.0	0.0	0.0	0.0	0.0					
17	0.7	1.0								
18	0.0	0.0	0.0	0.0						
19	0.0	0.0	0.0	0.1	0.0	0.0				
20	0.6	0.9	1.0							
21	0.4	0.0	0.0	0.0	0.0	0.0	0.0			
22	0.0	0.0	0.0							
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
24	0.5	0.6	1.0							
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
26	0.0	0.0	0.0	0.1	0.0	0.0	0.0			
27	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
28	0.0	0.0	1.0							
29	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0		

Table 5-4 The IoU details for the Testing dataset

However, this metric is not without issues, in (Redmon and Farhadi, 2018) the author discusses the IoU metric and questions the applicability of this metric. In images 7 and 8 from Figure 5-20 and Figure 5-21, which is enlarged in Figure 5-24 and Figure 5-25 below the IoU metric shows that there is minimal correlation between the two images with only 2 images correlating (Table 5-5). It can be seen that the images are actually representative after thresholding to perform the IoU, the shape, and clustering is all reasonable and the positional accuracy looks correct. Overall, it could be considered that despite the metrics there is an adequate correlation between the target and predicted values.

	Training Data			Testing Data							
image	object 1	object 2	object 3	Image	object 1	object 2	object 3	object 4	object 5	object 6	object 7
7	0.0	0.0	0.0	8	0.0	0.0	0.0	0.0	0.0	0.3	1.0

Table 5-5 Image 7 & 8 IoU Issues

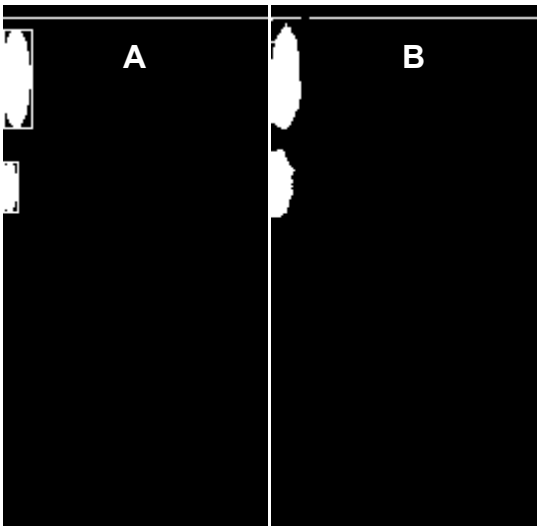


Figure 5-24 7 IoU issues for Image 7 Training Data Actual ('A') and Predicted ('B')

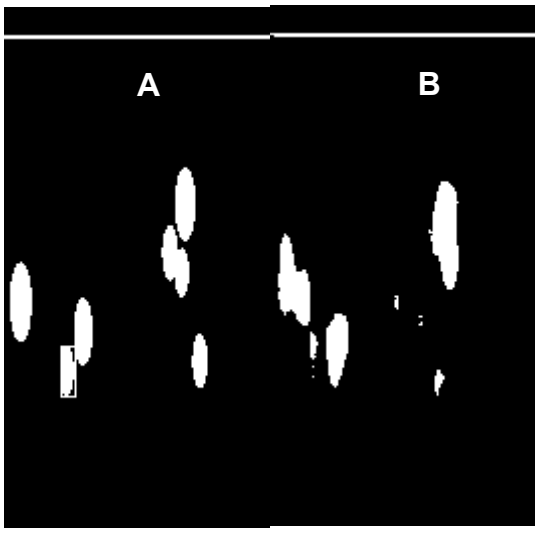


Figure 5-25 7 IoU issues for image 8 Testing Data Actual ('A') and Predicted ('B')

5.3.3 Multi-Dimensional Conditioning on the CVAE.

The intention of multi-dimensional conditioning is to attempt to condition the latent space based on two variables, such as material and size. The reason for looking into this is to improve the segmentation and thus gain better correlations between the two latent spaces. Whilst some attempts have been made, the results are far from conclusive, though are looking promising.

5.4 2D to 3D Image Generation

The previous sections described transitioning between 2D images in different spaces which was done to prove that the algorithm works. This section will focus on implementing a 2D to 3D transition. This was done by building on Version 1 and Version 5, with the addition of a new Version 6 of the software, all will be described below.

The primary aim of this project is to perform a 3D reconstruction of GPR images, for deep learning this leads to a fundamental problem of size and capacity of hardware. A 2D array of size [64, 64] is 4096 cells, where as a 3D array of size [64,64,64] is 262,144 cells. Mathematically, this means that a 2D tensor array expands at the rate of n^3 in size when converted into a 3D tensor array, which is a memory usage problem in the neural network when mapping between the 2D and 3D domains. An example is that a 2D table of size containing the coordinates of 196,133 pixels takes up 2.24Mbytes, however converting this into a 3D array, where a 1 represents an active voxel and 0 is empty takes up 84Mbytes.

Several options were assessed in order to try and minimise this issue and will be discussed in the following sub-sections:

- Use bigger Voxels
- Down sample the point cloud
- Resample the Voxel space

5.4.1.1 Increase the Size of Voxels

Figure 5-24 uses stock images of a scene to demonstrate the options available when processing Voxels. The first is to use bigger voxels, which encompass more points but reduces resolution. The image on the right is 0.095 Mbytes compared to 84Mbytes on the left.

Figure 5-26 Comparison Between a Small Voxel Image and Large Voxel Sample Image

5.4.1.2 Down Sample the Point Cloud

The second alternative is to down sample the point cloud, this has the impact of reducing every other voxel from the image thus having the same impact on file size as changing the voxel size, without detrimentally reducing the resolution as shown in Figure 5-27.

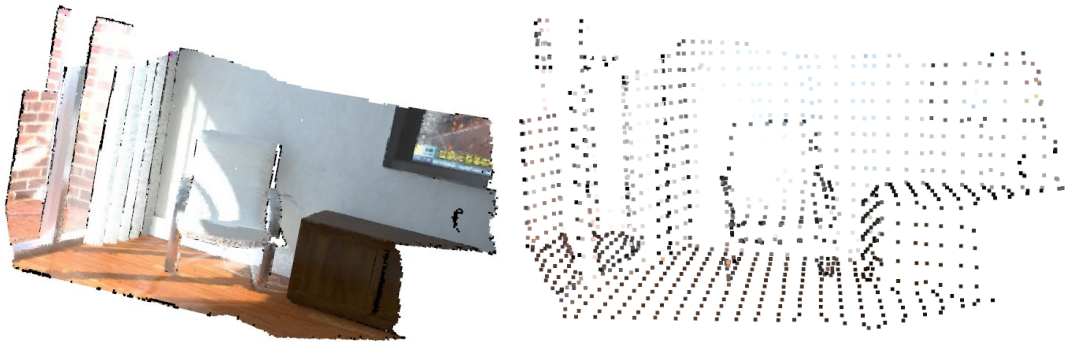


Figure 5-27 Effect of Down Sampling on A Voxel Image

5.4.1.3 Resize the Voxel Array

The final option is to treat the voxel array in the same approach as an image and effectively rescale the image to a much smaller grid, this is shown in Figure 5-28 below.

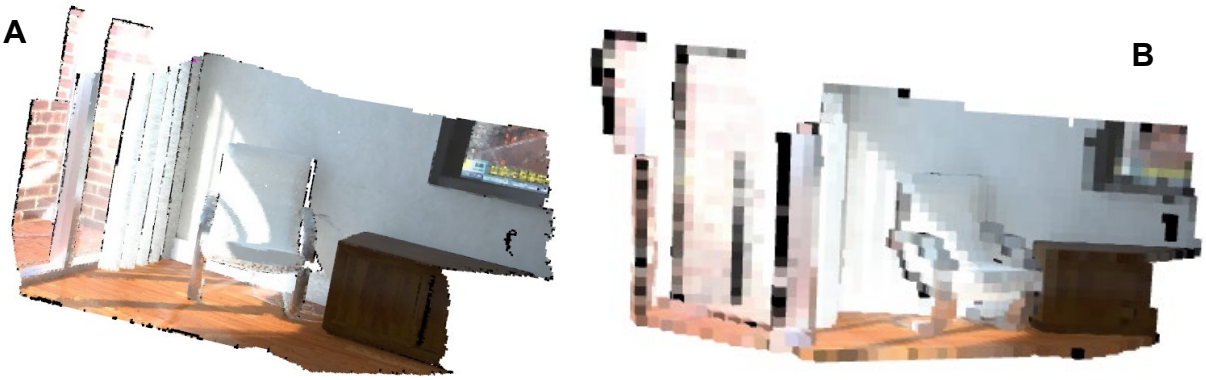


Figure 5-28 Image 'A' Shows the Original image and Image 'B' shows the effect of Resizing the Voxel Array to 1/64th of the Original

5.4.1.4 O-CNN

The O-CNN (Wang *et al.*, 2017) approach (see section 6.2) solely focuses on the efficiencies within the convolutional kernels by combining groups of similar regions into a super region, thus reducing the number of computations required, for this reason it is expected give the biggest impact on reducing computation resources. This technique would require significant effort to migrate to the transpose convolutional required for this project, and it was felt that while the memory efficiency was attractive it was not the best use of time on the project.

The options discussed in this section all show potential. Of the three identified options, the down sampling and reduced resolution were not the best application since this would reduce the resolution of the output. Instead, the approach taken forward was to resize the voxel array in a similar manner to a 2D pixel array as this is assumed to give the best compromise of quality, memory, and effort.

5.4.2 2D-3D Mapping

In this section, the intention is to show that by building on the knowledge gained from the 2D-to-2D mapping, it is possible to map between the spaces populated by the GPR B-scan results (2D) and the space populated by the point cloud representation of the tubers (3D). In other words, the intention of this activity is to define:

$$f: 2D \rightarrow 3D$$
$$f: 3D \rightarrow 2D \qquad \textbf{5.21}$$
$$(x, y) \in 2D \text{ there is a } (x', y') \in 3D$$

Or in other words, for every location in the 2D space there is a function which relates to the 3D space.

5.4.3 Version 6

The aim of this section is to use the 2D-to-2D mapping as a way of creating an optimised 2D-to-3D mapping architecture. If hardware resources were infinite, or this solution was developed commercially, the same architecture developed in the previous chapter would be read directly across to this section, with the only change made being converting the 2D decoder into a 3D decoder. This, however, is not possible due to the issues discussed in section 5.4.2, where the main problem is one of memory resources.

As a result, a new more simplified architecture had to be found and tested in order to be used as a springboard for the 2D-to-3D mapping. To this end, the Version 6 – series was created and tested. This is a significantly less complicated and more streamlined approach with respect to the architecture, which uses a single CVAE and GAN to perform the mapping between the two spaces. However, the training process is slightly more complicated. There is only a forward pass and no cycle consistency checks performed. The result of this was to optimise the 2D dataset over 90 epochs and the results in the 2D space were really promising – see Figure 5-29 where the top set of images are the actual images, and the bottom set of images are the predicted values.

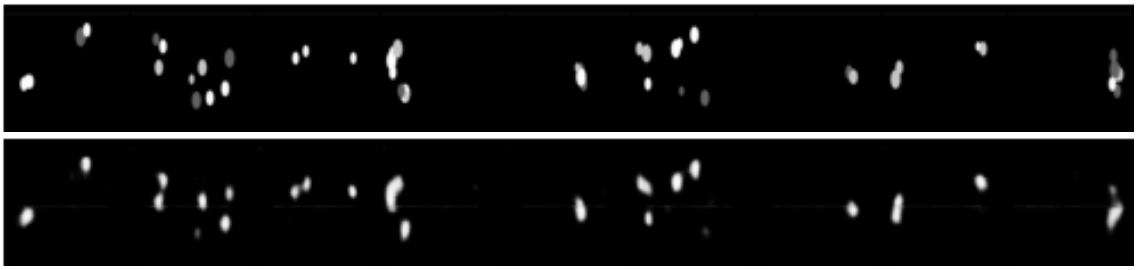


Figure 5-29 Version 5 2D Improvements to the 2D Mapping between the B-Scan and 2D Image (the top images show the target, and the bottom images show the predicted)

5.4.4 Architecture of the Computational Model

Several variations of the same architecture have been attempted, with varying levels of success. Some did not create a mapping at all and just generated noise, and some had too many parameters for the graphic card GPU memory. Generally, all the architecture is based on Figure 5-30, as stated in section 5.1.2 the core is a CVAE-GAN and the output is mapped 3D data, which allows the mapping between the 2D and 3D spaces to be achieved with the CVAE and the improvement in clarity being performed by the GAN.

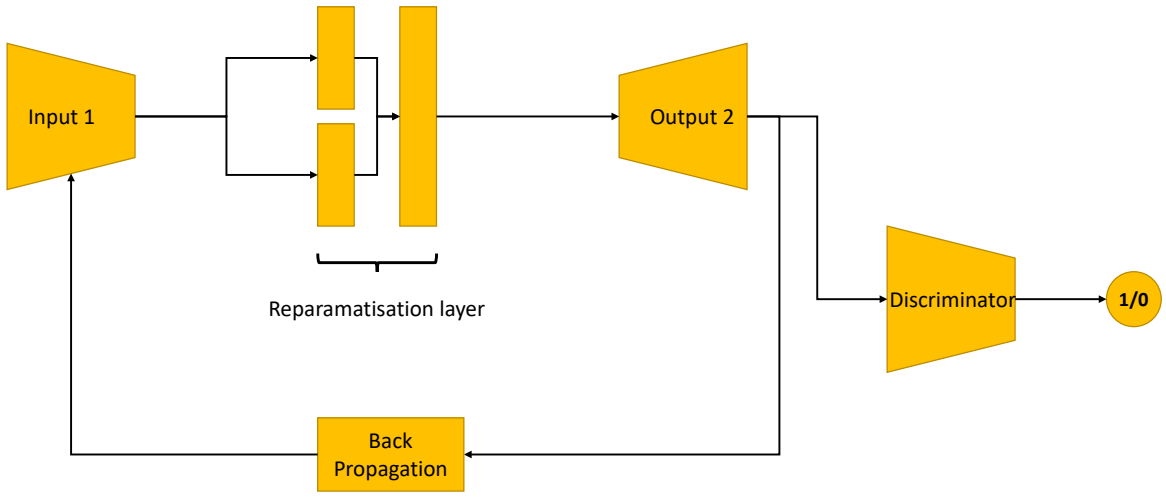


Figure 5-30 Version 6 Revised Architecture

During training, the model would often collapse, with values in the convolutional areas going to NaN. The reason for this was never fully conclusive, however all indications point to the gradients going to zero. A large amount of effort was put into trying to overcome this and several modifications were made to the training algorithm resulted in the convergence of the model without the gradients collapsing to zero, however there was an impact on the time taken to train the model. In all the variations, the deep learning side did not change much, the biggest change was to add an extra layer after increasing the 2D image size from 64 to 128 pixels square. However, changes were made to the way in which the training was performed.

The input 3D image size and hence the generated 3D image size is limited to a three-channel tensor of 64 voxels in each of the three dimensions. The size of the output is constrained by the memory available in the Graphical Processor Unit (GPU). Increasing the image size increases the memory requirements by a cubed relationship. The [64,64,64] size was found to be suitable at this stage for the network. When the final postprocessing happens, it is possible to scale these images back up to the “real-world” size to get a true representation of the tubers. In order to increase the resolution of the voxel array, care has to be taken in order to

not introduce numerical errors by interpolating the between the existing volumes and this can be done using affine transformations or a simple B-spline type approach which would use the existing voxel positions as the control points. Issues do exist at with end-effects, and this will need to be explored in future work.

5.4.5 Results

5.4.5.1 Version 1

Using a very limited dataset of only 8,000 images and 3D models the results shown in Figure 5-32 were produced. While the images were promising, the model was unstable, and it took a lot of care to get the model to converge as discussed in section 5.4.4. The model was considerably more unstable when the larger dataset (16,000) was used. It was paramount to try and improve the stability of the model without impeding the quality of the mapping. After a significant amount of trial and error, the source of the instability has been identified as the retention of the graph in the back propagation.

In terms of machine learning, backpropagation is the method used to calculate the gradients of the parameters in the neural network. This is done by traversing the network starting at the output and finishing at the inputs. The graph is computed during the forward pass through the network and during the reverse pass the gradients are calculated. Ordinarily the stored graph is reset after each iteration, however using the retain_graph=true statement keeps the graph history after each iteration. Retaining the graph during the back propagation of the model forces the model to retain necessary information such as gradients as well as intermediate values that are required to calculate the 3D mapping. This leads to a constantly increasing size array and eventually instability. As a result, this parameter was removed by recoding this section and resulted in increased stability of the model.

The results for the increased stability model are shown in Figure 5-31 and Figure 5-32 below. In the following images, the colours do not hold any significant and are used as an aid to identify individual potato representations.

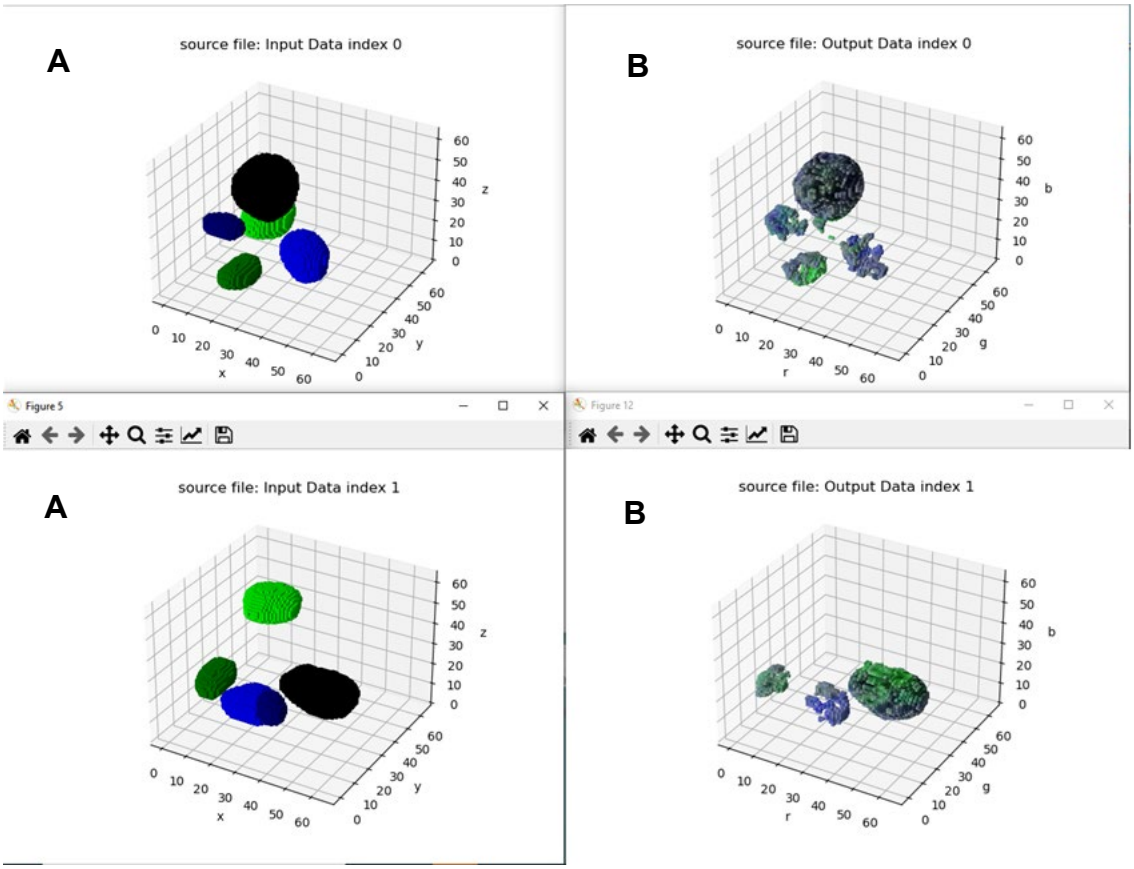


Figure 5-31 Comparison of Version 1 Results for the 3D Output ('B') vs the 3D Target Image ('A').

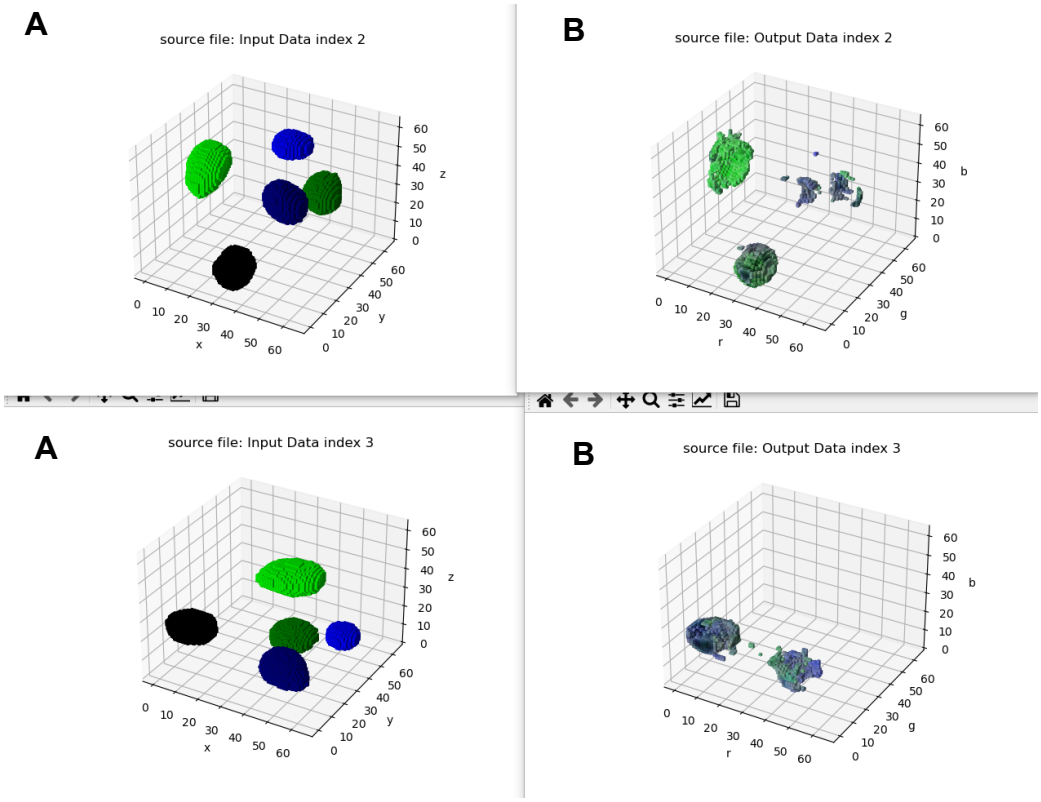


Figure 5-32 Comparison of Version 1 Results for the 3D Output ('B') vs the 3D Target Image ('A').

In Figure 5-32, the index value in the title is based on the number generated, i.e. index 0 is the based on the first item in the dataset, and index 3 is the fourth item in the dataset (this convention is used in all proceeding images). It can be seen that the model is very adept at recreating the deeper objects, but consistently struggles with the shallow objects. One reason for this is that the near surface signal is "swamped" by the air-soil interface as discussed in chapter 3.

5.4.5.2 Version 5

Taking the optimised 2D mapping tool described in section 5.4.3 above and translating it into the 2D-3D mapping space leads to a specific problem created the version 5 of the 2D-3D mapping. The problem came when the architecture principles were read across to the 3D models as shown in Figure 5-33 and Figure 5-34 and the model was run over 200 epochs. It can be seen that the 3D results are starting

to converge, however they are not as good as version 1 results after only 150 epochs.

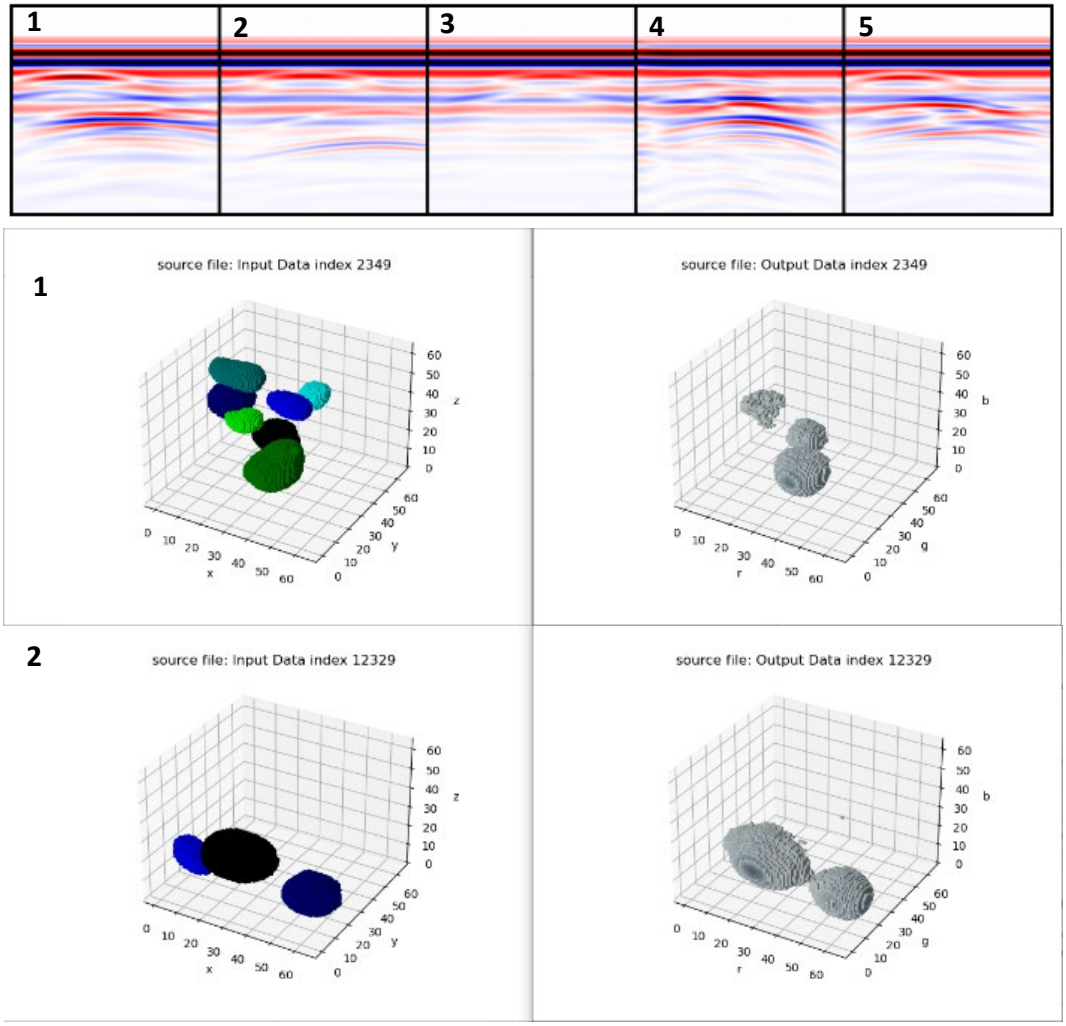


Figure 5-33 Version 5 Mapping Between the B-Scan and 3D Image - Training Data Images 1 & 2

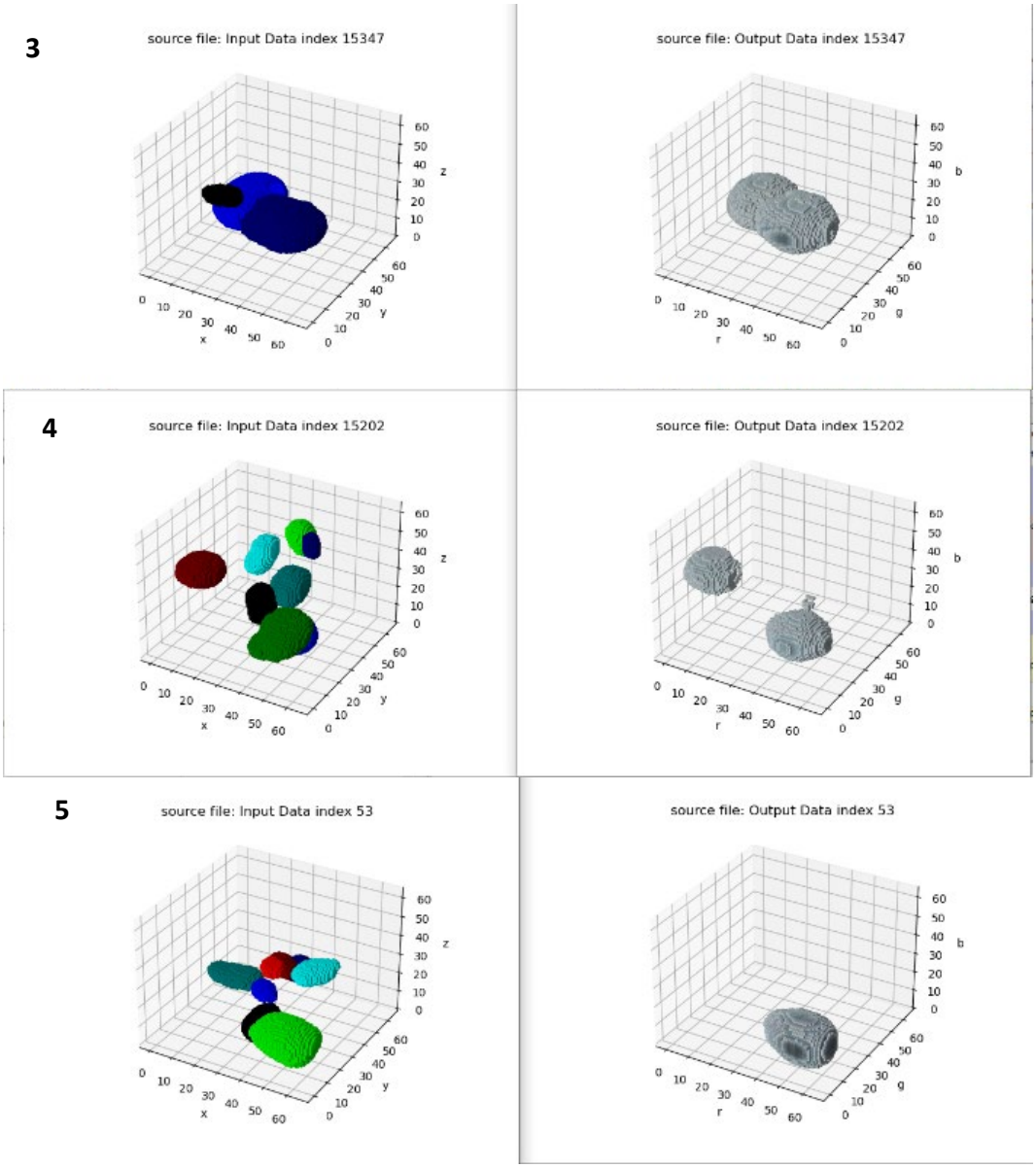


Figure 5-34 Version 5 Mapping Between the B-Scan and 3D Image - Training Data Images 3 to 5

The discrepancy between 2D and 3D results for the same architecture may be due to the additional dimensions - the 2D output is a single channel 2D array, while the 3D output is a 3-channel 3D array, which, as discussed in section 5.4.4 leads to an additional level of complexity. Therefore, it might be a case of leaving the model to run on for additional epochs and wait for further results.

Alternatively, the 2D model utilises the Maximum Mean Discrepancy (MMD) function- as described in section 5.1.4 and MSE loss as opposed to the Kullback-Leibler Divergence (KLD) and MSE loss as a way of trying to overcome a fundamental problem with the CVAE based architecture, where a limited amount of information passed between the encoder and decoder (Zhao, Song and Ermon, 2018). This was attempted within the 2D-3D translation; however, the solution was unstable and as a result never converged, this caused the 2D-3D uses only the KLD and MSE loss functions.

Two changes to the above architecture were made in an attempt to make the model perform better:

1) The inclusion of dropout layers in an effort to try and reduce over fitting on the 2D encoder side of the model.
2) The addition of a median filter as discussed in section 4.3.4, with the aim of trying to improve deep learning architecture's ability to utilise real world data.

The results for this model are shown in Figure 5-35 and Figure 5-36 The testing data is shown in Figure 5-37 and Figure 5-38.

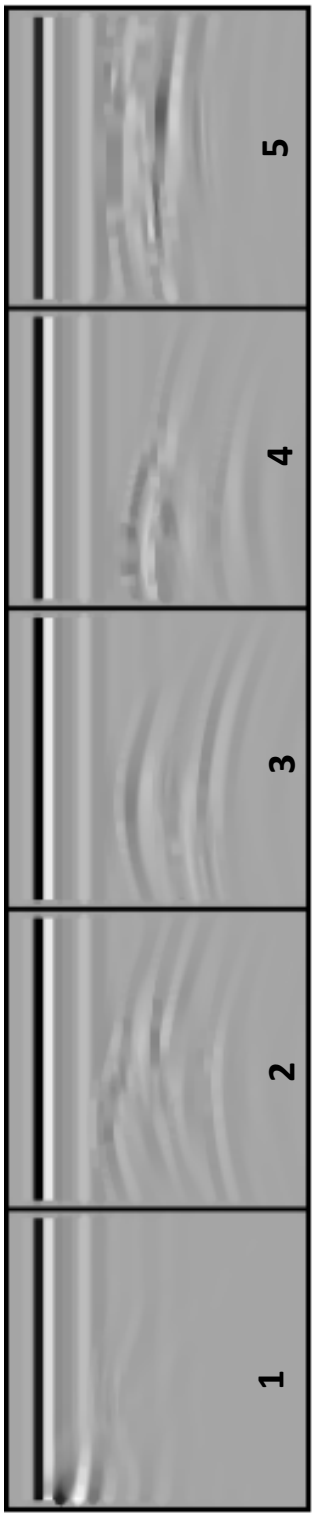


Figure 5-35 Filtered GPR B-Scan Input Data

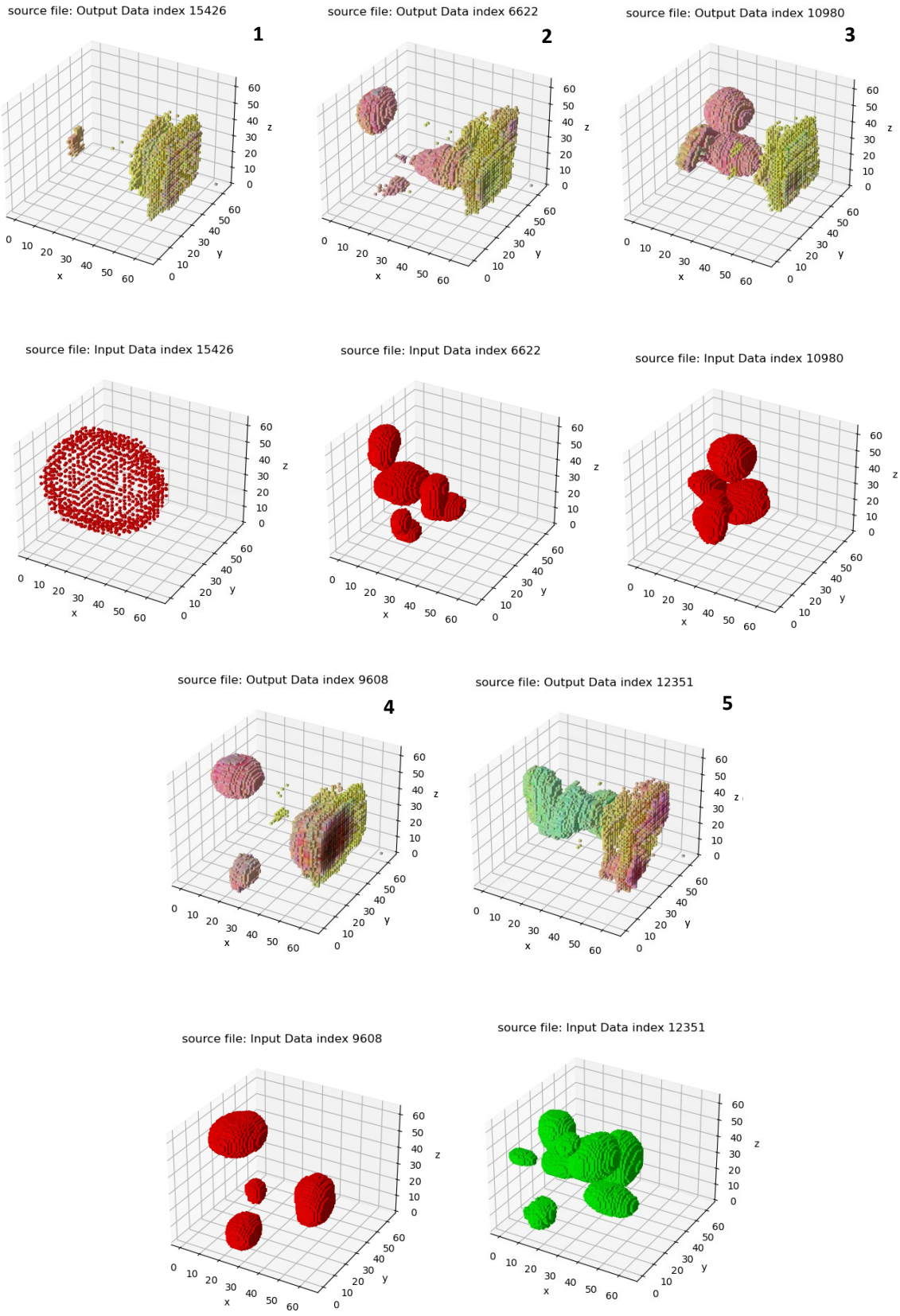


Figure 5-36 Mapped 3D Training Images (predicted top and actual bottom

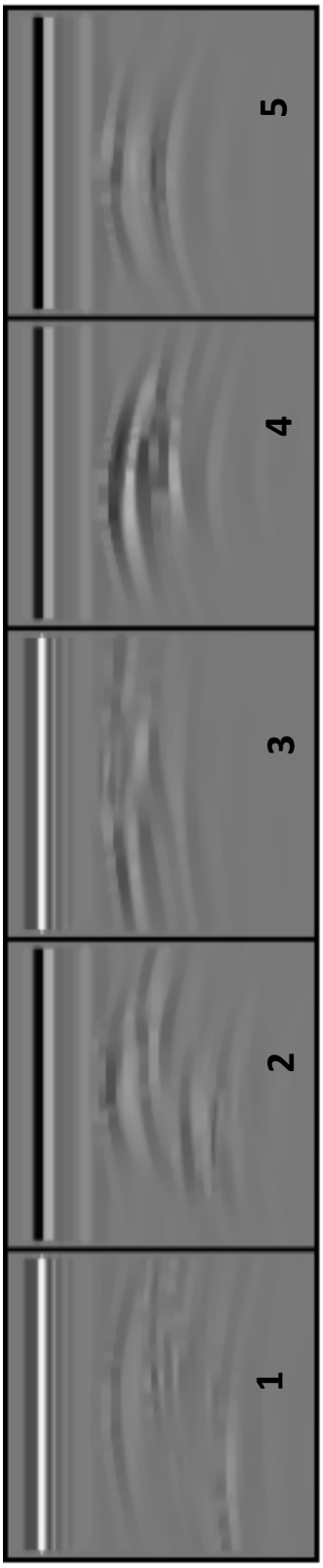


Figure 5-37 Testing Filtered GPR B-Scan Input Data

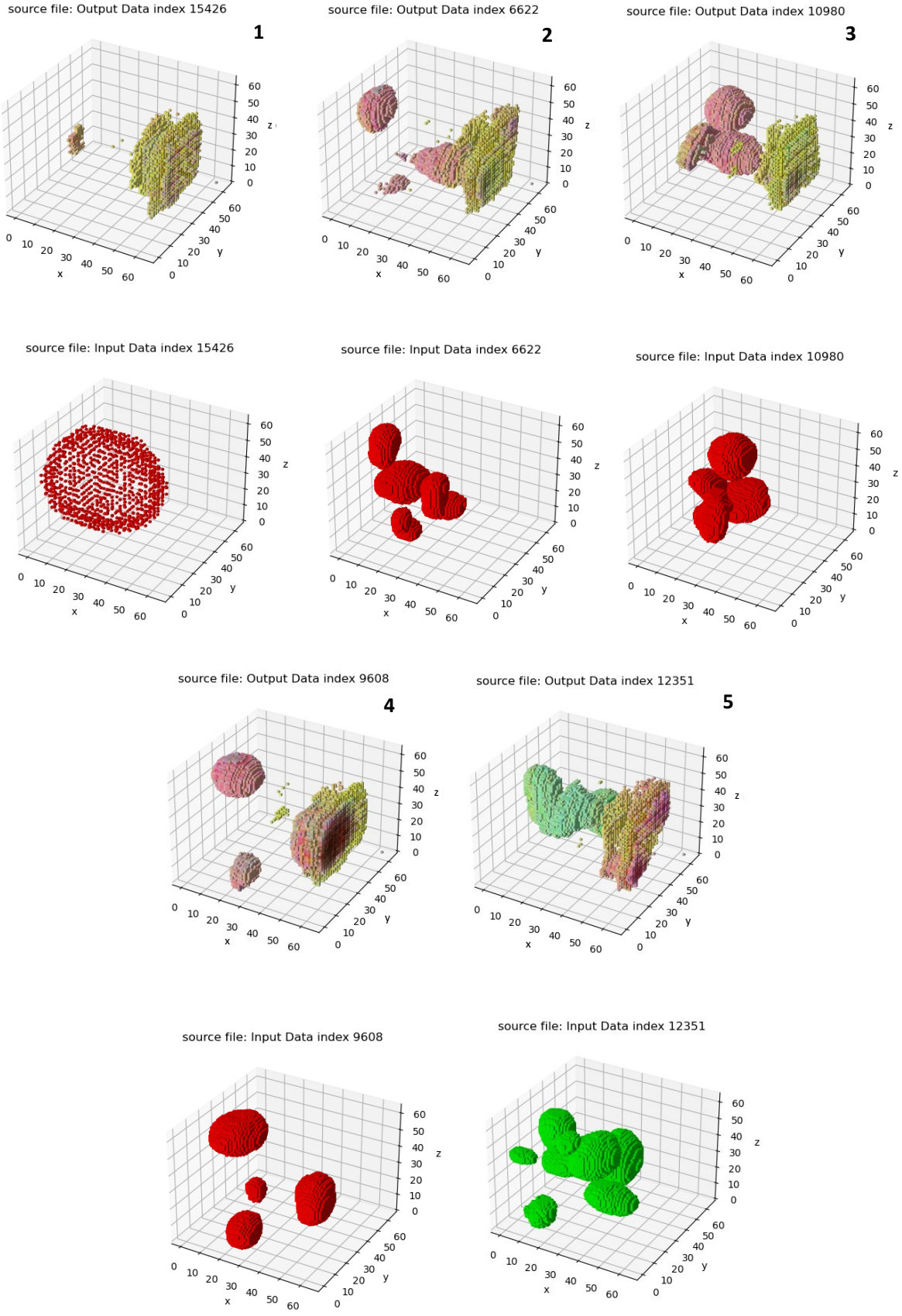


Figure 5-38 Mapped 3D Testing Images (predicted top and actual bottom)

In Figure 5-38, there is a large “blob” on the extreme edge of the output. Since this does not exist in the input data, it is assumed that this is the result of a lack of conversion in the model, and not as a result of the preprocessing of the input data which creates “edge effects” when the convolutional kernel tries to smear values close to the boundary. It is unlikely to be down to issues within the model, since the models which use the “clean” GPRMax data does not exhibit this behaviour.

5.5 Testing the Model with Real data

A special test zone was set up in a field, which used two ridges. These ridges were approximately 20m long with groups of potatoes placed at regular intervals consistent with a field. The scanning equipment ran parallel along the axis of the ridge as shown in Figure 5-39 below. The potatoes were clustered either individually or in groups which represented different yield ratings (high, medium and low)

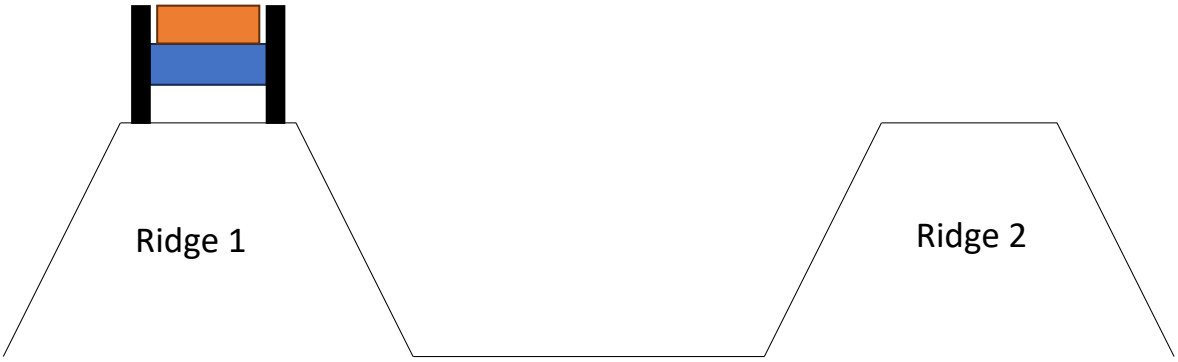


Figure 5-39 Showing how the Real-World Data was Collected (GSSI Antenna Direction of Travel is into the Page).

In section 4.2.2, attempts were made to find the best way of pre-processing the raw data from the field to minimise noise and enhance the signal. It was found that using the SVD along with a Time Varied Gain gave the most consistent results. In the last days before the harvest was performed, several scans of crops were made along with photos taken as shown in Figure 5-40 to Figure 5-43.

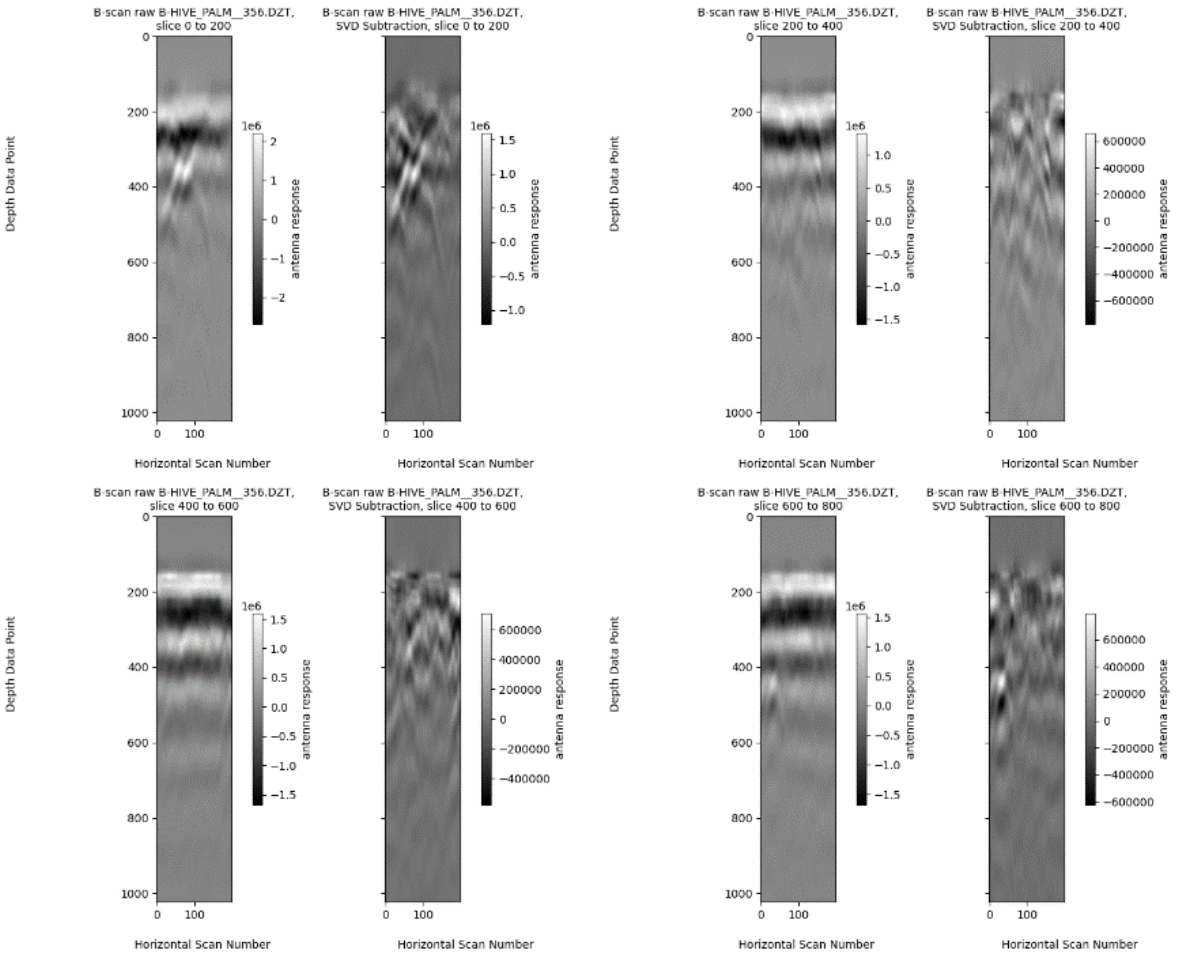


Figure 5-40 Showing the Raw Data and Pre-Processed Data Scan Sections 0 to 800. (B-Hive and Lewis, 2021)

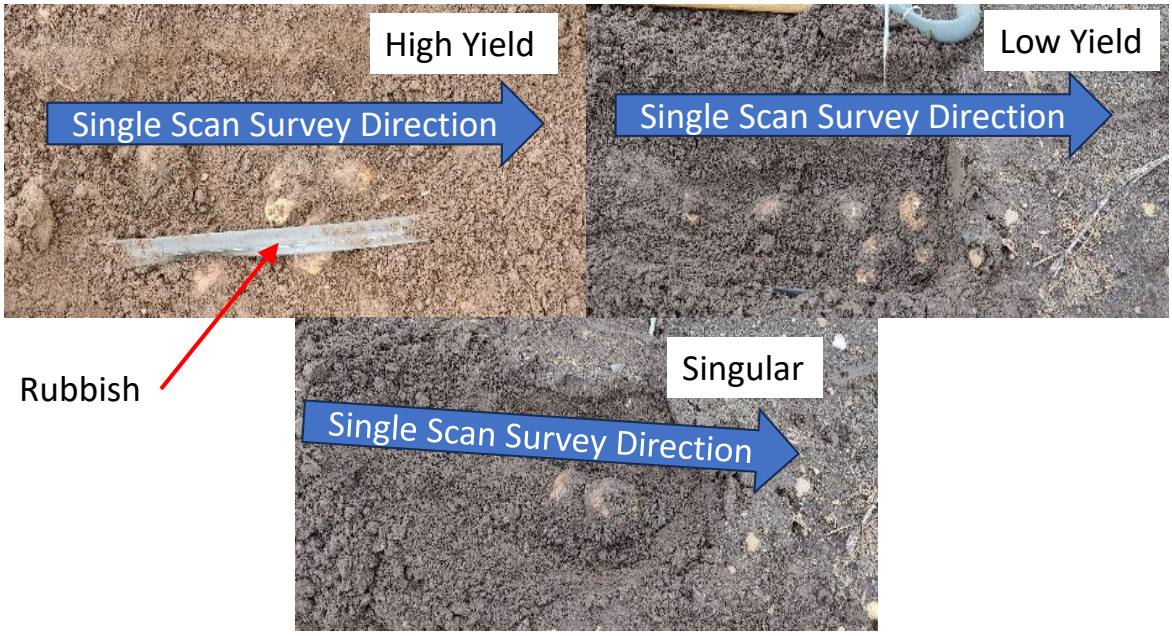


Figure 5-41 Actual Tubers in the Ground Scan 0-200 (B-Hive and Lewis, 2021)

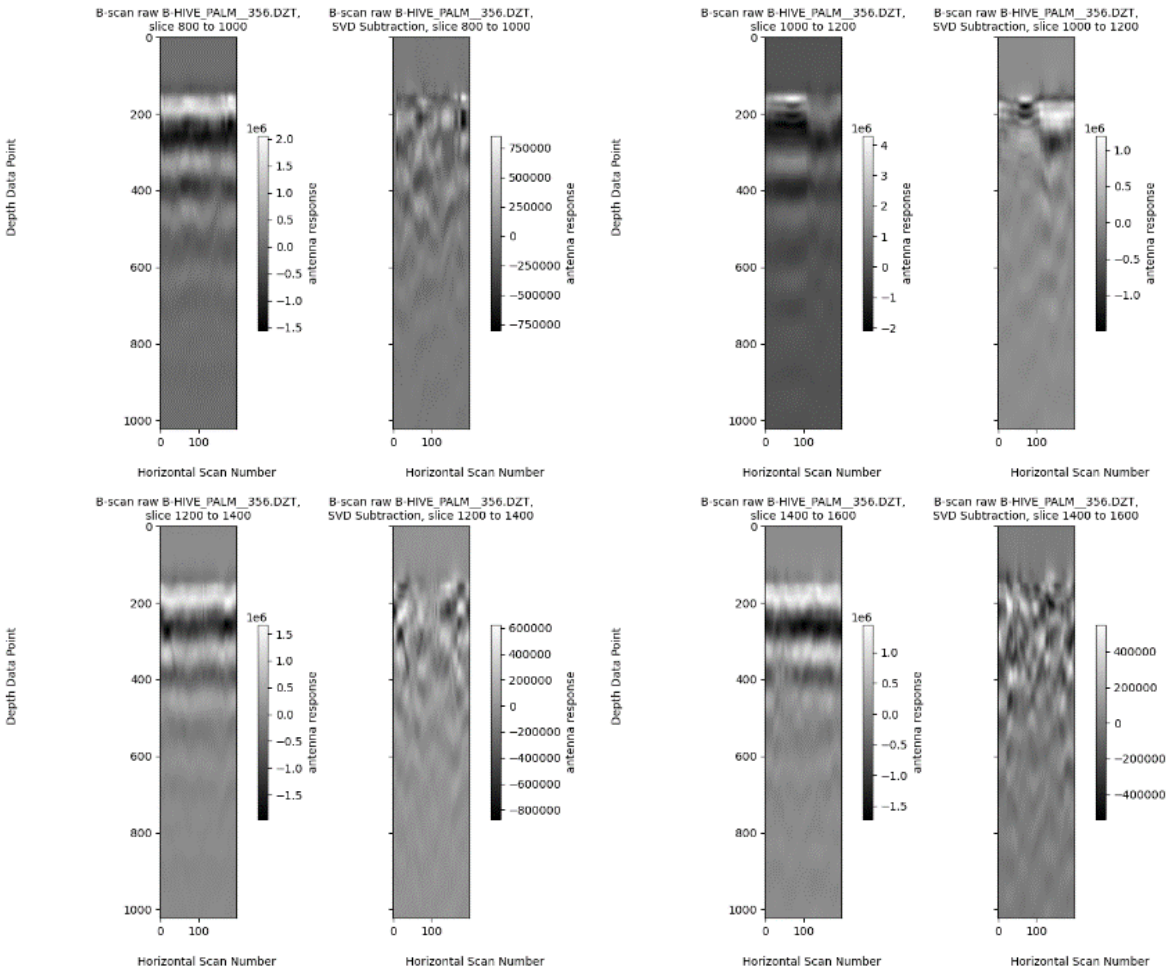


Figure 5-42 Showing the Raw Data and Pre-Processed Data Scan Sections 800 to 1600. (B-Hive and Lewis, 2021)

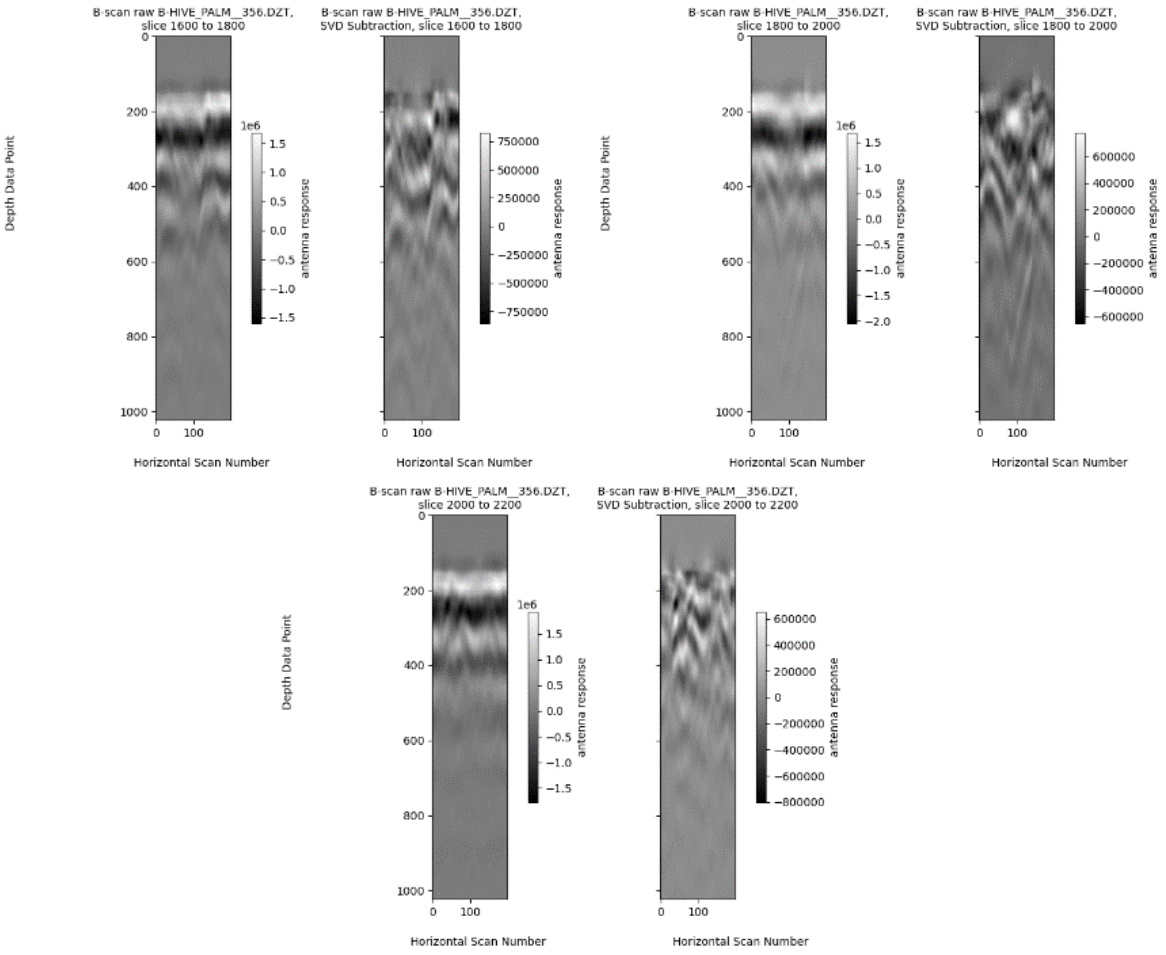


Figure 5-43 Showing the Raw Data and Pre-Processed Data Scan Sections 1600 to 2200. (B-Hive and Lewis, 2021)

Within the initial GPR data, it was found that there were duplicated columns randomly through each of the 200 data sets. This data has been removed and then a linear interpolation has been performed between the columns either side of the missing data. In Table 5-6, a sample of the raw data from the GSSI hardware is shown, each column is an A-scan, and each row is a measurement point, it can be seen that columns 20 and 22 are duplicates of 19 and 20 respectively. The reason for this duplication is unknown.

The revised data is shown below in Table 5-7. The revised data is a linear interpolation between the columns either side, so column 20 is effectively the mean

of columns 19 and 21. This has the effect of smoothing the data out between the individual A-Scans.

There is a lot of noise on the individual A-Scan which becomes more obvious once a time varying gain has been applied to counter the damping effects as discussed in section 3. This was removed using a Savitzky-Golay filter as this was found to be the best option for cleaning up the signal. The before and after samples are shown in Figure 5-44 and Figure 5-45 below. In the following images the term “Depth Increment” is used to identify the position in the time series, for example Depth Increment=0 is the first value in the series.

		Horizontal Increment					
Depth Increment		18	19	20	21	22	23
	0	-391.3	3013.65	3013.65	8034.32	8034.32	4607.95
	1	576.988	3589.64	3589.64	8143.94	8143.94	5040.39
	2	1473.91	4136.69	4136.69	8234.28	8234.28	5440.85
	3	2303.23	4656.42	4656.42	8307.32	8307.32	5811.63
	4	3068.68	5150.48	5150.48	8365.02	8365.02	6155
	5	3774.01	5620.49	5620.49	8409.36	8409.36	6473.25
	6	4422.96	6068.1	6068.1	8442.31	8442.31	6768.66
	7	5019.3	6494.94	6494.94	8465.82	8465.82	7043.52
	8	5566.75	6902.65	6902.65	8481.89	8481.89	7300.12
	9	6069.07	7292.86	7292.86	8492.47	8492.47	7540.73
	10	6530.01	7667.21	7667.21	8499.53	8499.53	7767.65
	11	6953.3	8027.33	8027.33	8505.05	8505.05	7983.15
	12	7342.7	8374.87	8374.87	8510.99	8510.99	8189.53
	13	7701.96	8711.46	8711.46	8519.33	8519.33	8389.06
	14	8034.81	9038.73	9038.73	8532.03	8532.03	8584.03
	15	8345.01	9358.32	9358.32	8551.07	8551.07	8776.72

Table 5-6 Sample of the Raw Data from the GSSI Hardware showing duplicate columns.

		Horizontal Increment					
Depth Increment		b	19	20	21	22	23
	0	-391.3	3013.6	5524	8034.3	6321.1	4608
	1	576.99	3589.6	5866.8	8143.9	6592.2	5040.4
	2	1473.9	4136.7	6185.5	8234.3	6837.6	5440.9
	3	2303.2	4656.4	6481.9	8307.3	7059.5	5811.6
	4	3068.7	5150.5	6757.8	8365	7260	6155
	5	3774	5620.5	7014.9	8409.4	7441.3	6473.2
	6	4423	6068.1	7255.2	8442.3	7605.5	6768.7
	7	5019.3	6494.9	7480.4	8465.8	7754.7	7043.5
	8	5566.8	6902.6	7692.3	8481.9	7891	7300.1
	9	6069.1	7292.9	7892.7	8492.5	8016.6	7540.7
	10	6530	7667.2	8083.4	8499.5	8133.6	7767.6
	11	6953.3	8027.3	8266.2	8505	8244.1	7983.2
	12	7342.7	8374.9	8442.9	8511	8350.3	8189.5
	13	7702	8711.5	8615.4	8519.3	8454.2	8389.1
	14	8034.8	9038.7	8785.4	8532	8558	8584
	15	8345	9358.3	8954.7	8551.1	8663.9	8776.7

Table 5-7 Sample of the Interpolated Data from the GSSI Hardware

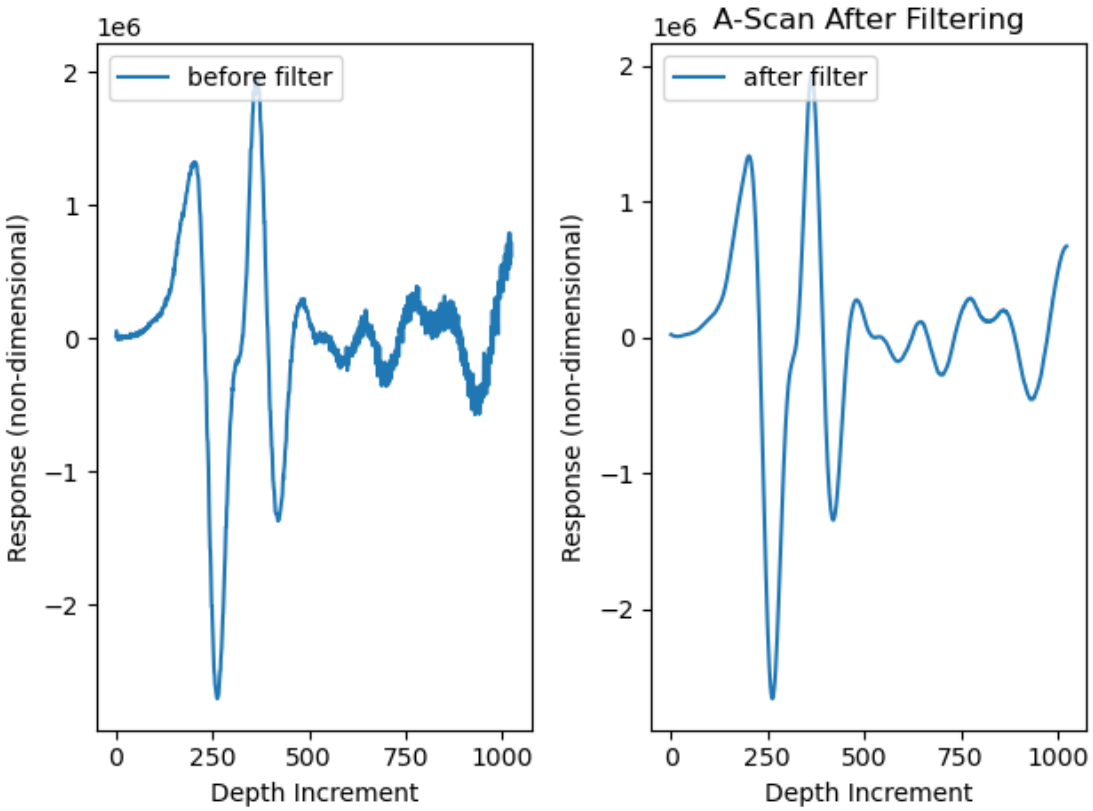


Figure 5-44 Sample of an A-scan Before and After Filtering

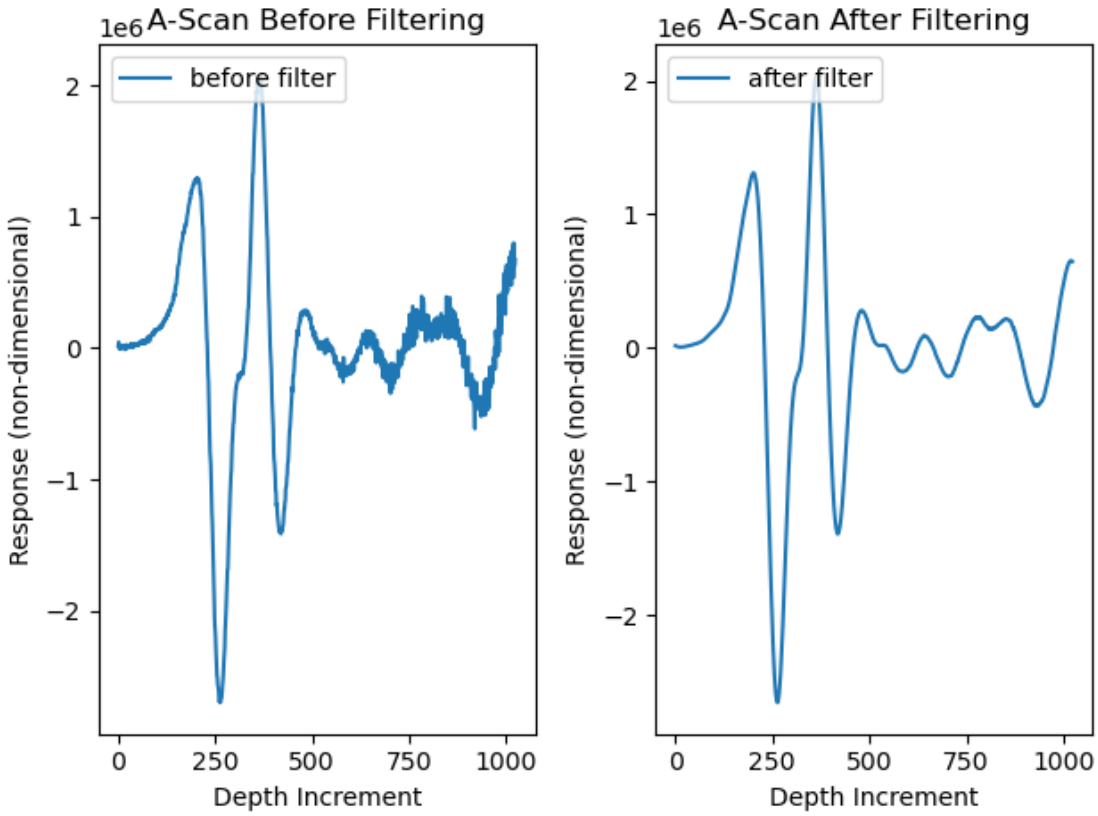


Figure 5-45 Sample of an A-scan Before and After Filtering

The B-scans after pre-processing are shown in Figure 5-46, and were used in the trained model version 1.5, 1.5b and version 5. In the following images the index reference refers to the number of the B-scan in the corresponding image. The results for version 1.5 are shown in Figure 5-47 and Figure 5-48. Using the SVD process described in section 6, has an input B-scan shown in Figure 5-49, with the mapped results shown in Figure 5-50 and Figure 5-51. The results for 1.5b are shown in Figure 5-52 and Figure 5-53. The results for version 5 are shown in Figure 5-54 and Figure 5-55. In the following figures the 'X', 'Y' and 'Z' axis are dimensionless and represent the position in the 64x64x64 array which is used to create the image.

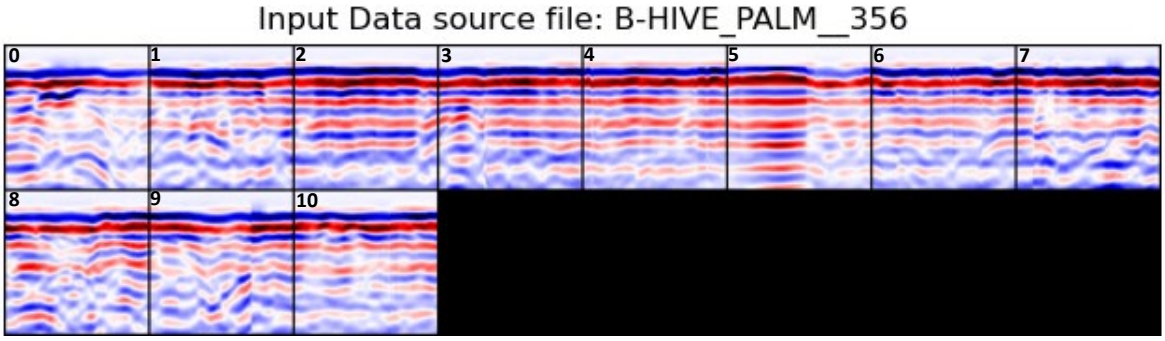


Figure 5-46 GSSI B-scans (B-Hive and Lewis, 2021)

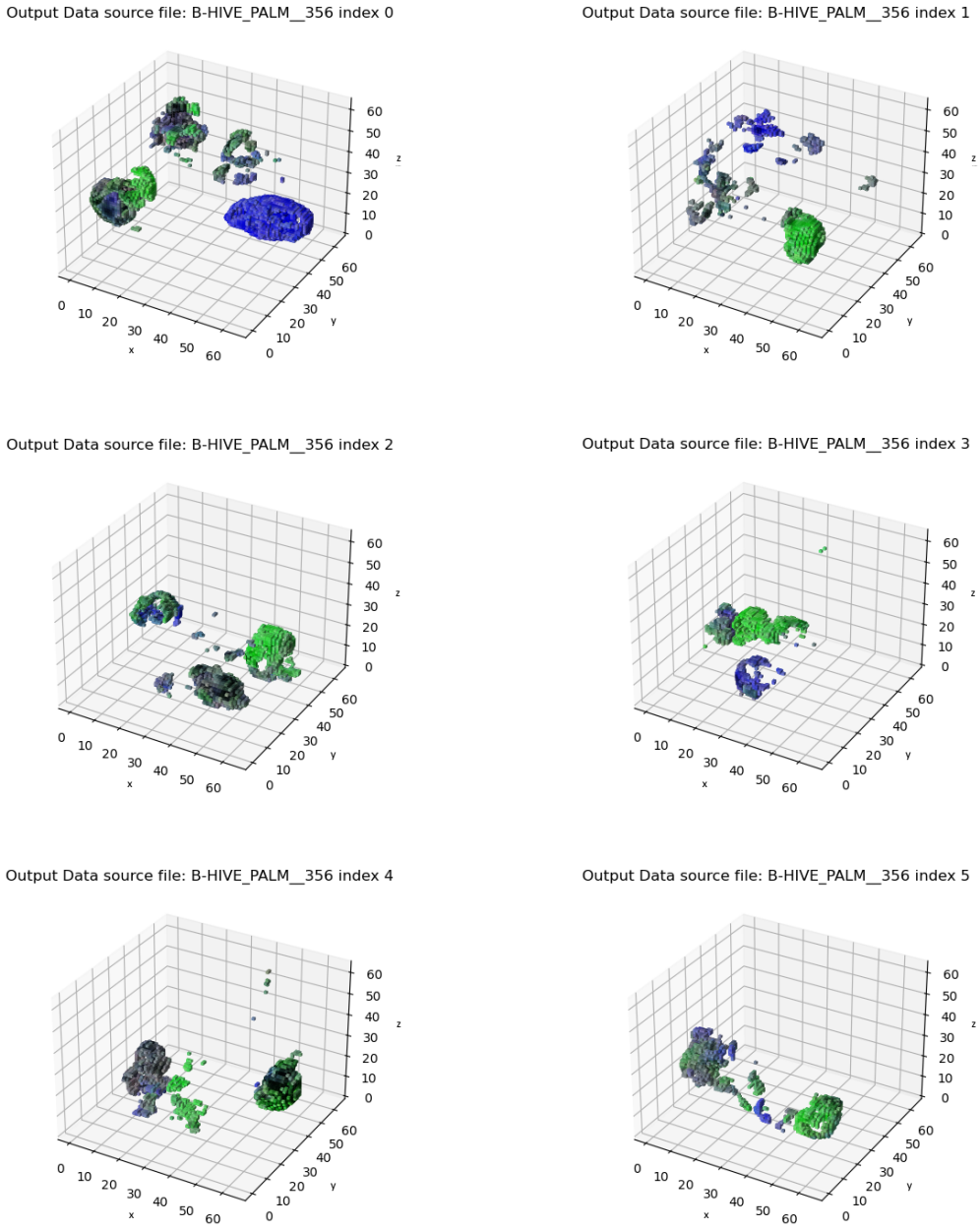


Figure 5-47 Version 1.5 Real World 3D Predictions from B-scans 0 to 5

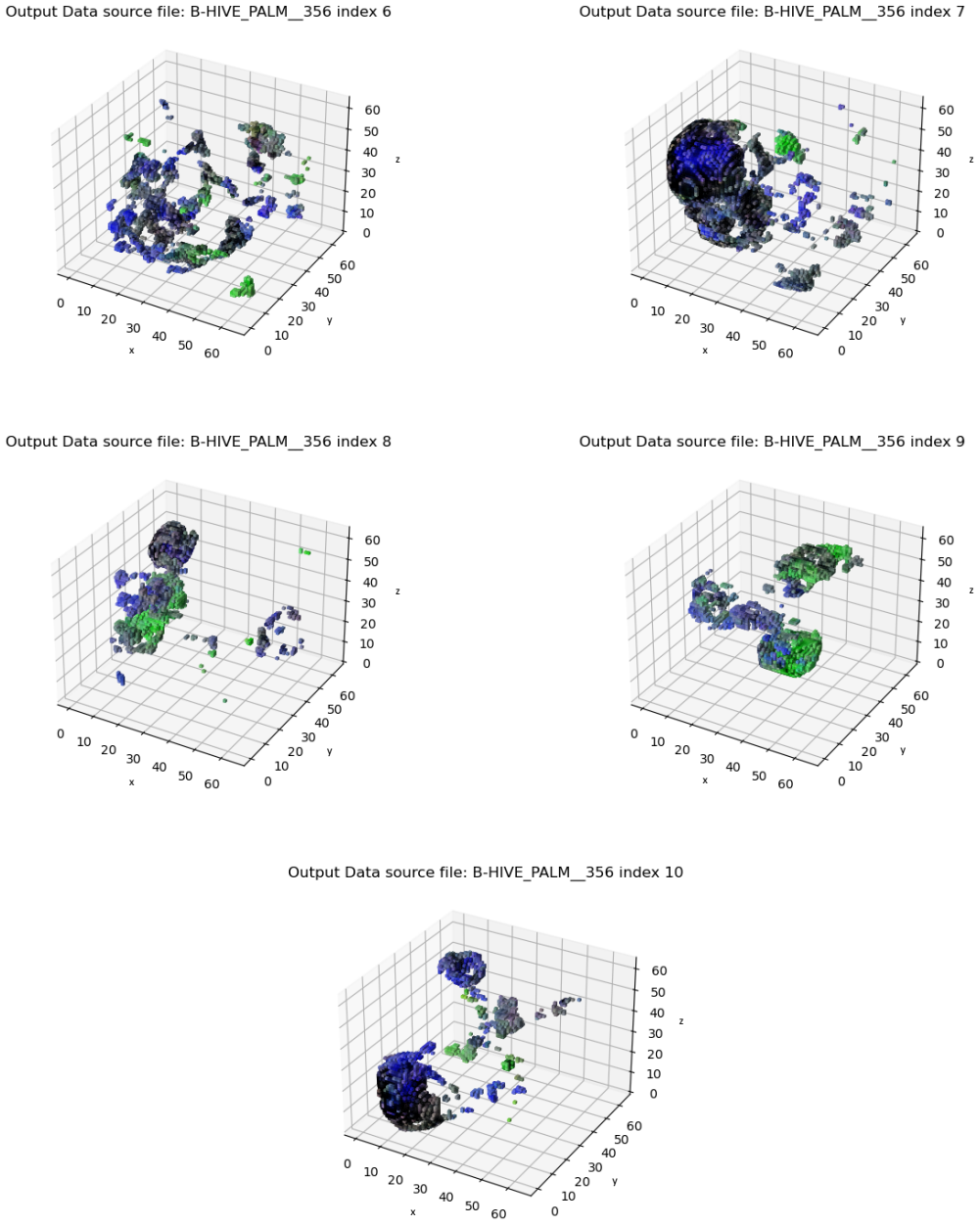


Figure 5-48 Version 1.5 Real World 3D Predictions from B-scans 6 to 10

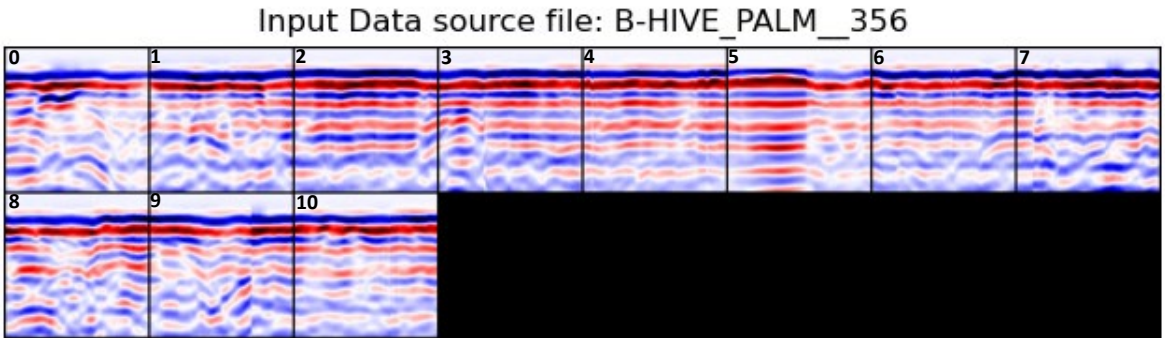


Figure 5-49 GSSI B-scans after SVD Background Removal

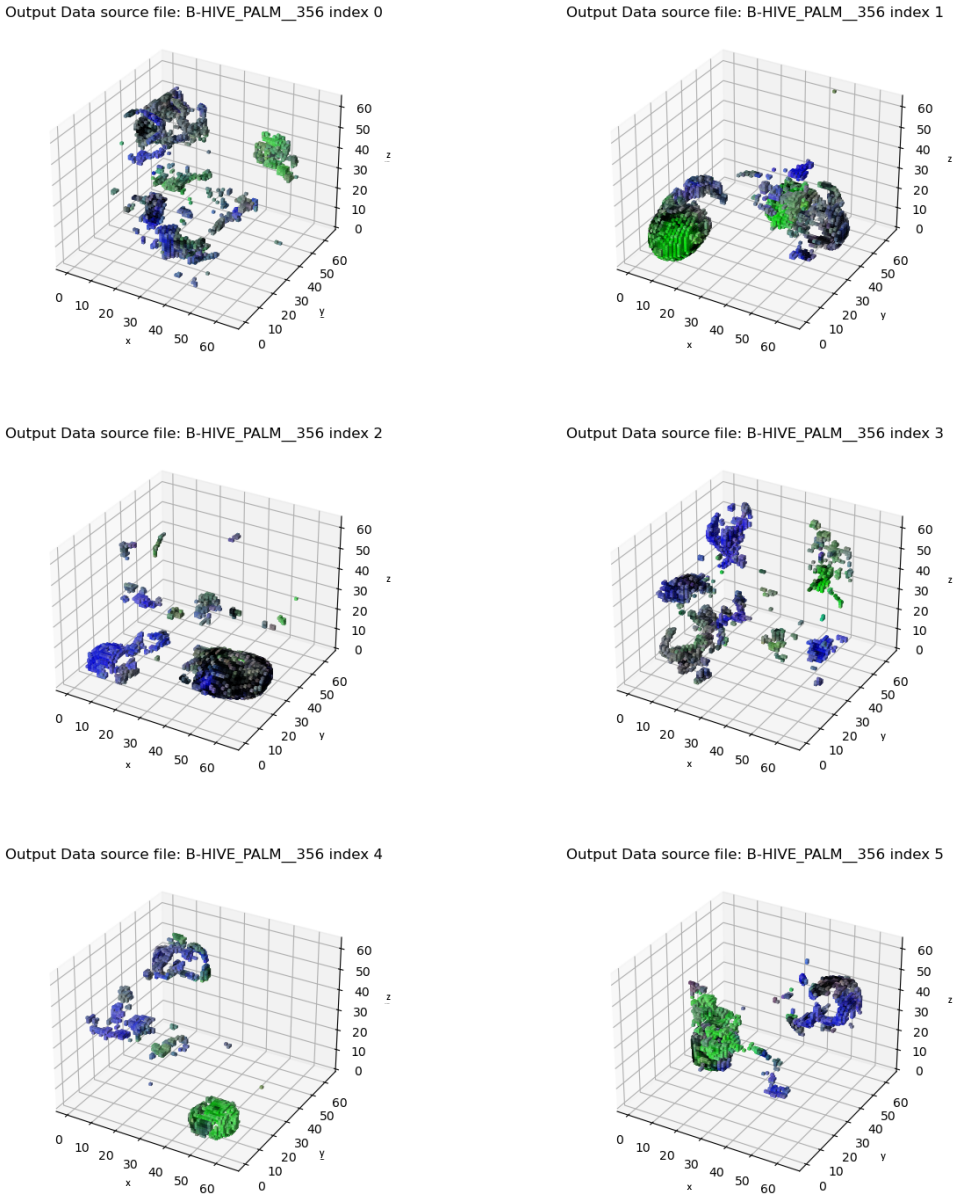


Figure 5-50 Version 1.5 with Real World SVD 3D Predictions from B-scans 0 to 5

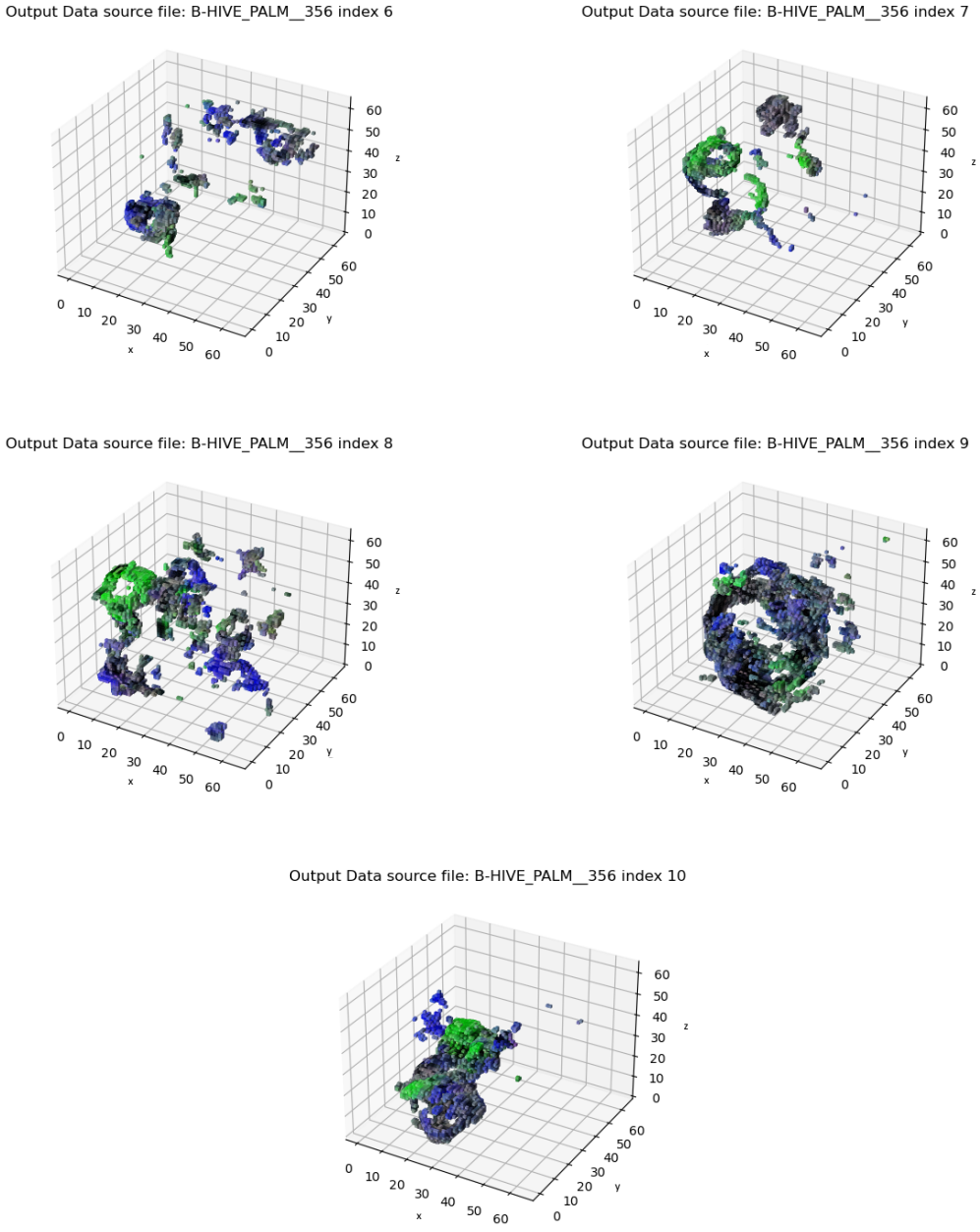


Figure 5-51 Version 1.5 with SVD Real World 3D Predictions from B-scans 6 to 10

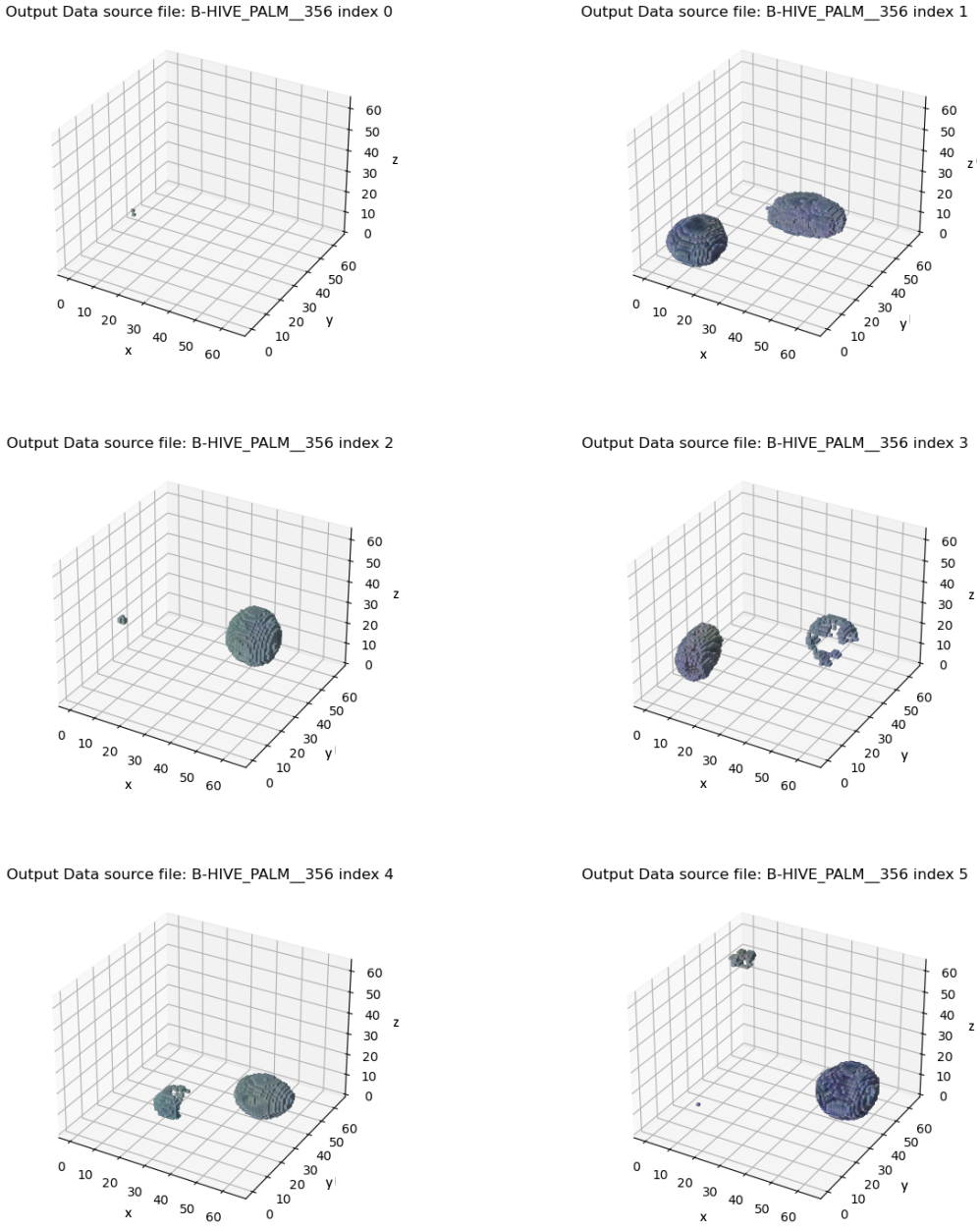


Figure 5-52 Version 1.5d 3D Real World Predictions from B-scans 0 to 5

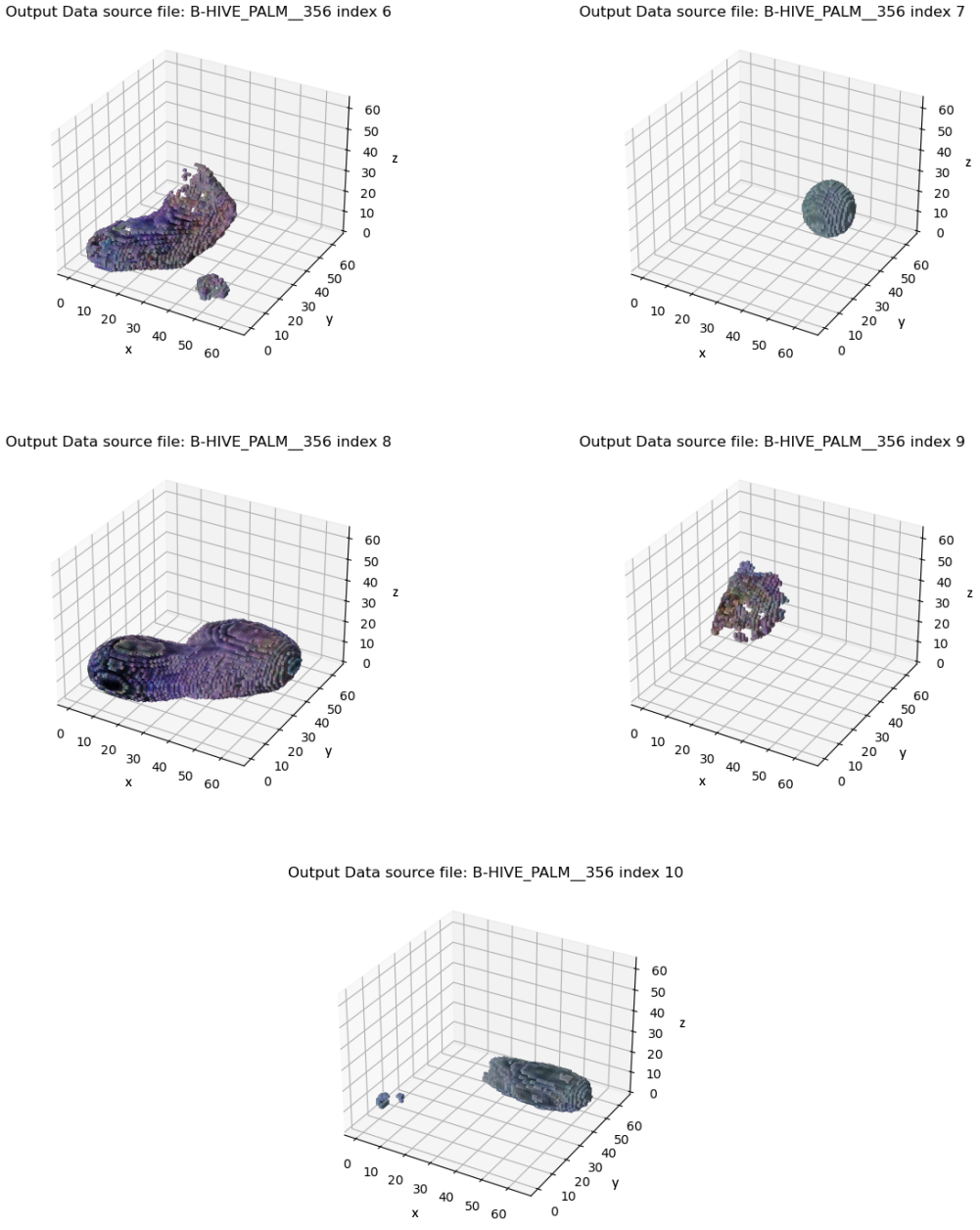


Figure 5-53 Version 1.5d Real World 3D Predictions from B-scans 6 to 10

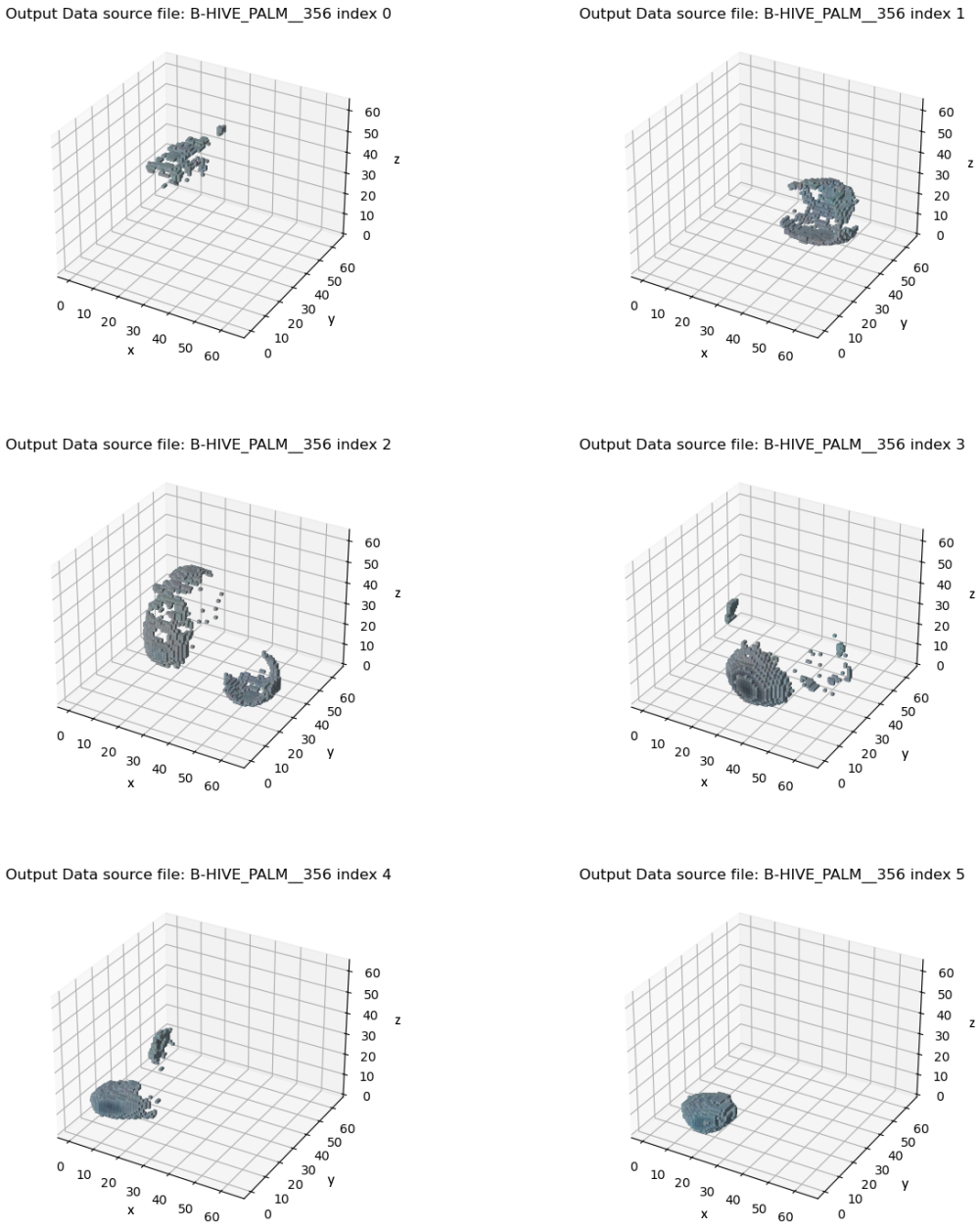


Figure 5-54 Version 5.1 Real World 3D Predictions from B-scans 0 to 5

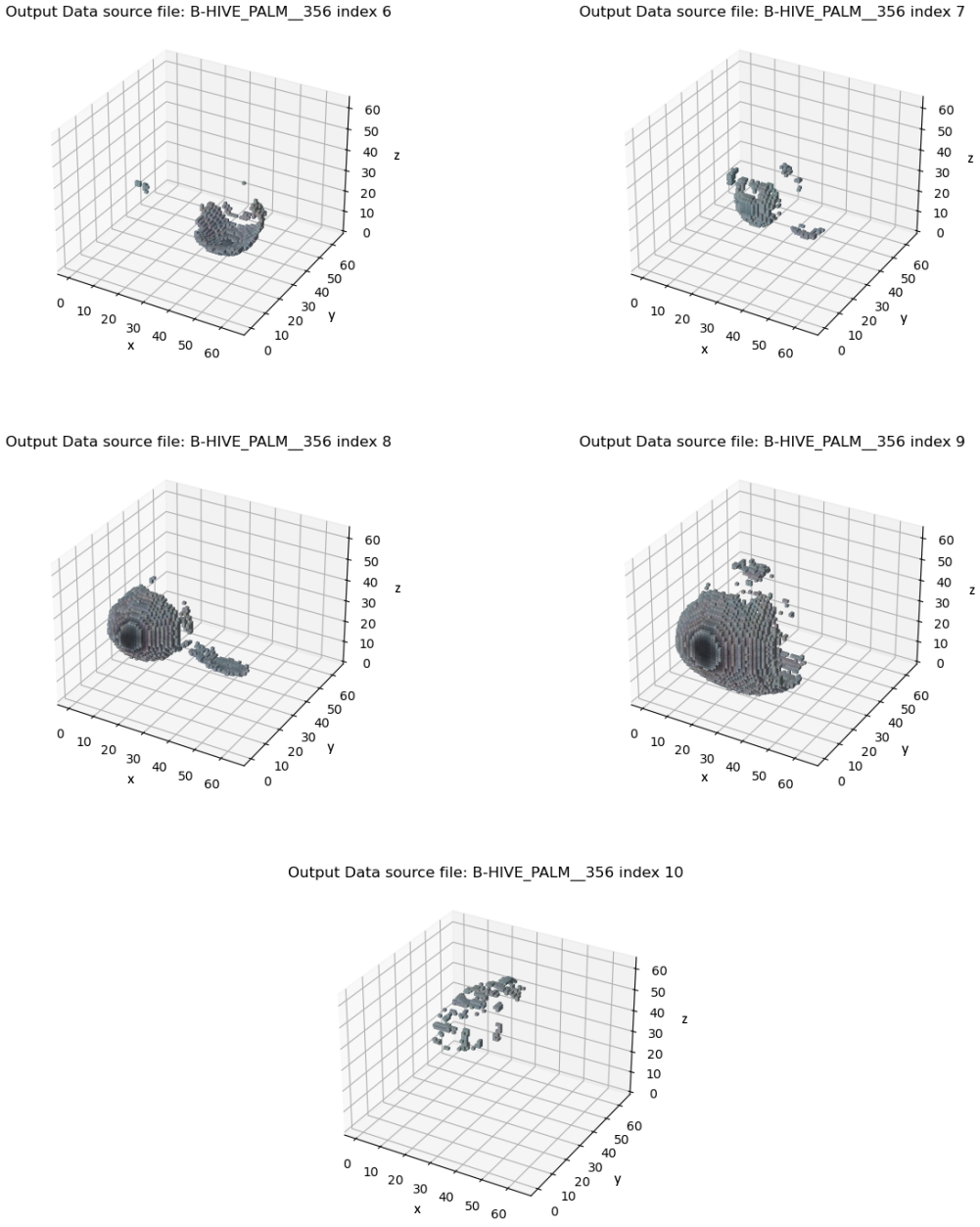


Figure 5-55 Version 5.1 Real World 3D Predictions from B-scans 6 to 10

This section has shown that Version 6 can produce good results on GPR data produced in an actual field, and not just synthetic training data. More work can be done to improve the results further, as will be discussed in section 5.10.

5.6 Discussion

The approach adopted in this thesis differs from existing methods of employing Deep learning to this field rely on identifying the nature of the response as described in Chapter 2 until the application of a UNET to solve the inverse problem was proposed by (Dai *et al.*, 2022). The approach used by this team is to use a two-stage method.

The 2D-to-2D mapping has given some promising results. Version 5 created the clearest and most accurate results based on the IoU and the MSE values, however it will be very difficult to use these results for estimating yield, since there is no knowledge of the third dimension and hence the volume will be difficult to approximate. It does show the benefit of the cyclic consistency in creating readable and usable results.

When comparing the results back to the alternative solution used by (Dai *et al.*, 2022), it can be seen that they have experienced similar problems, the approach works well when a single shape/response is used. However, once the scan includes multiple objects close together, the accuracy of the system starts to decline. It is also clear that, while the approach has created images which are geometric representations of the input data, the approach has struggled to create distinguishable representations based on the permittivity, which the UNET approach has succeeded in doing. The significant difference between the inputs for these two approaches is that use of RGB based inputs by the UNET method, as opposed to grey scale used in this thesis which was selected due to computational limitations. In hindsight it would have been better to utilise an RGB format for the images as the contrast between colours would be significantly higher than with Greyscale (light grey is very close to white and dark grey is very close to black). In Figure 5-29, there

are signs that the model is differentiating between the Dark Grey object and the other two colours, however there is little evidence that the model is differentiating between the light grey and white, which further suggests that there is an issue with contrast in the model. In Table 5-8 below it can be seen that as the objects become closer together, so the MSE deteriorates.

Scenario Type	MSE (↓)		
	FWI	SMRF-UNet	DMRF-UNet
One Object	1.5693	0.2225	**0.1804**
Two Separated Objects	7.0984	0.8116	**0.5544**
Two Interfaced Objects	3.8661	0.9687	**0.8579**
Ave.	4.7119	0.6676	**0.5298**

Table 5-8 Results for MSE for a UNet based Inversion Model (Dai *et al.*, 2022)

The results displayed in the 2D-3D mapping section 5.4.5 are also promising, however it is clear that the most promising version is version 1. Version 1 is a hybrid between the CVAE and GAN, while versions 5 & 6 use a second optimiser to train the generator of the CVAE as part of the GAN separately in addition to the optimiser for the CVAE. While using the SVD approach has removed a lot of the background clutter, the results look less encouraging than without SVD. Nevertheless, the results, though encouraging, are not inconclusive, the reason for this is thought to be due to the lack of resolution in the GPR system compared to the simulations. The main issue with the approach adopted in this thesis is that the training time is too long, and the results are not conclusive enough With this in mind, the main focus of development has to be in making the process more efficient, using examples such as quantisation or octree convolutions both of which are discussed in more detail within chapter 6.

In the 2D inversion model paper (Dai *et al.*, 2022), it has been identified in the future work section that the next stage is to identify a suitable architecture to perform 3D reconstructions. A suitable architecture has been identified in this work and with some additional research it should be possible to improve the results contained within this thesis.

It is expected that running the model for longer iterations should improve the convergence, however there is a danger that this could lead to overfitting the data (which means that it is less able to create approximations outside of the fixed points and is often considered detrimental to a machine learning model. That said it would be worth experimenting with and if overfitting starts to be a problem, then using a second GAN to generate additional data could be a way to reduce this effect.

Using a B-spline to interpolate between the existing points has been demonstrated to be both a faster conversion in under-determined and if over determined lead to a more accurate result (Fomel, 2000), from the b-splines it will also be possible to create surfaces using techniques such as B-spline surfaces. B-Splines are defined using a knot sequence (a vector) and a set of control points. The main drawback is that each vertex is given their own basis function which leads to a more complex description, however that said they are more flexible than other splines and it is possible to change a b-spline by altering either its knot-sequence or the control points.

It is clear that by adding filters to the input that this actually degrades the mapping. This is due to reducing the variation between the scans that the models require. If the resolution is going to be reduced, then additional scans need to be taken to allow for the recovery of resolution.

Further improvements to the learning algorithm could be made by performing additional pre-processing on the GPR data by removing the air-ground boundary spike. Doing this not as straight forward as taking just the air-ground signal from all and could involve a deconvolution technique as discussed in later in this section, or even using blind source separation (BSS). BSS is an established method of separating out individual signals from a mixed signal and has been used in fields such as the cocktail party problem (Cherry, 1953), image processing and EEG analysis.

From the results so far, it is clear that some additional processing of the input data is required to try and improve the final data. The real-world data will involve several layers of processing, including an Analogue to Digital Convertor (ADC), up sampling and filtering. There is also the need to increase the level of background noise in the simulations, which can be done by using the heterogeneous soil models built into GPRMax, using the same 3 Sphere model as Figure 4-3, but with a chaotic distribution of water, clay and sand through the model as proposed in (Peplinski, Fawwaz and Myron, 1995) is shown in Figure 5-56 below.

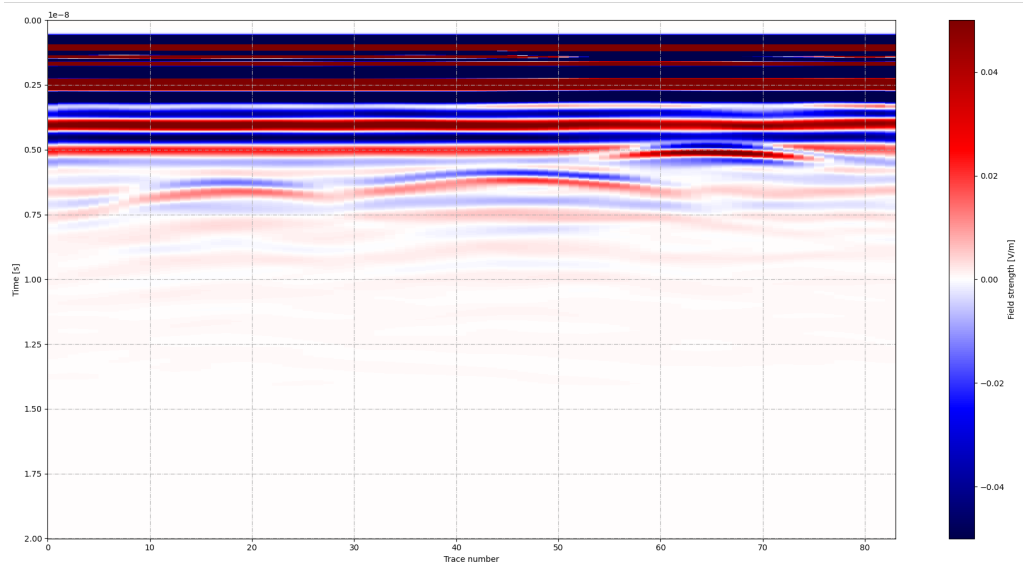


Figure 5-56 GPRMax Simulation of 3 Spheres within a Chaotic Soil Medium

Using a very basic ADC signed 12-bit convertor, which converts the analogue signal into a digital signal ranging from -2048 to 2048 (the 12 bit is for the sign) on each A-scan is shown in Figure 5-57. The analogue signal is plotted against the right-hand axis and the digital sampling points are the stick graphs plotted on the left-hand side. In Figure 5-58, the resulting digital signal can be seen.

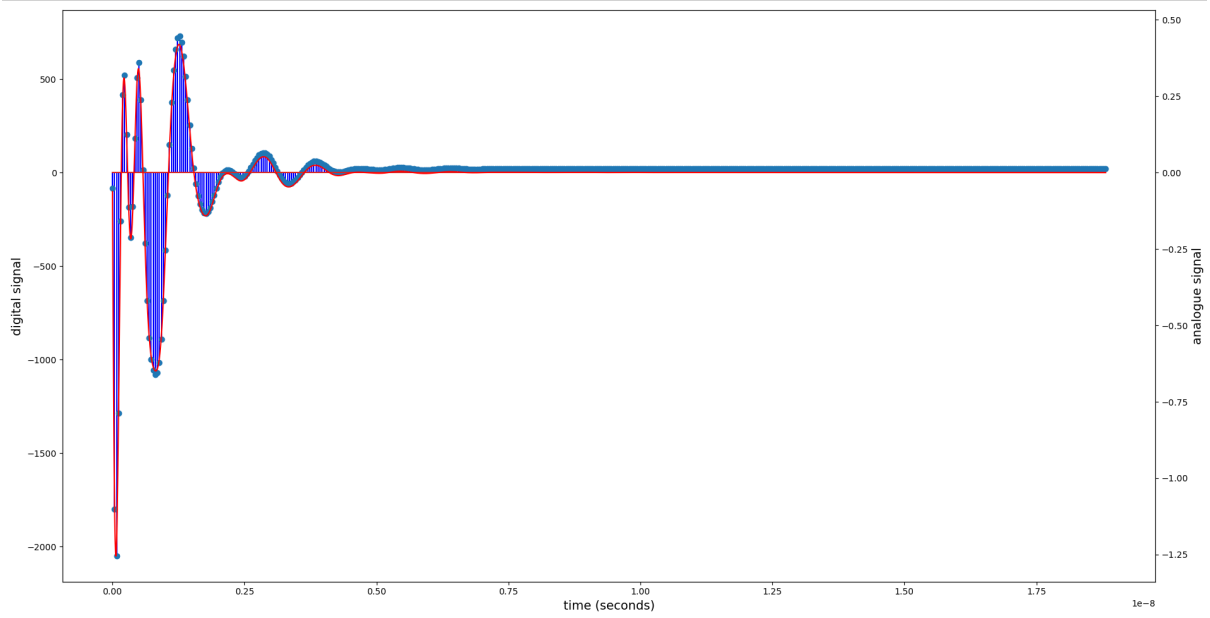


Figure 5-57 Analogue Signal and Digital Discretisation Points

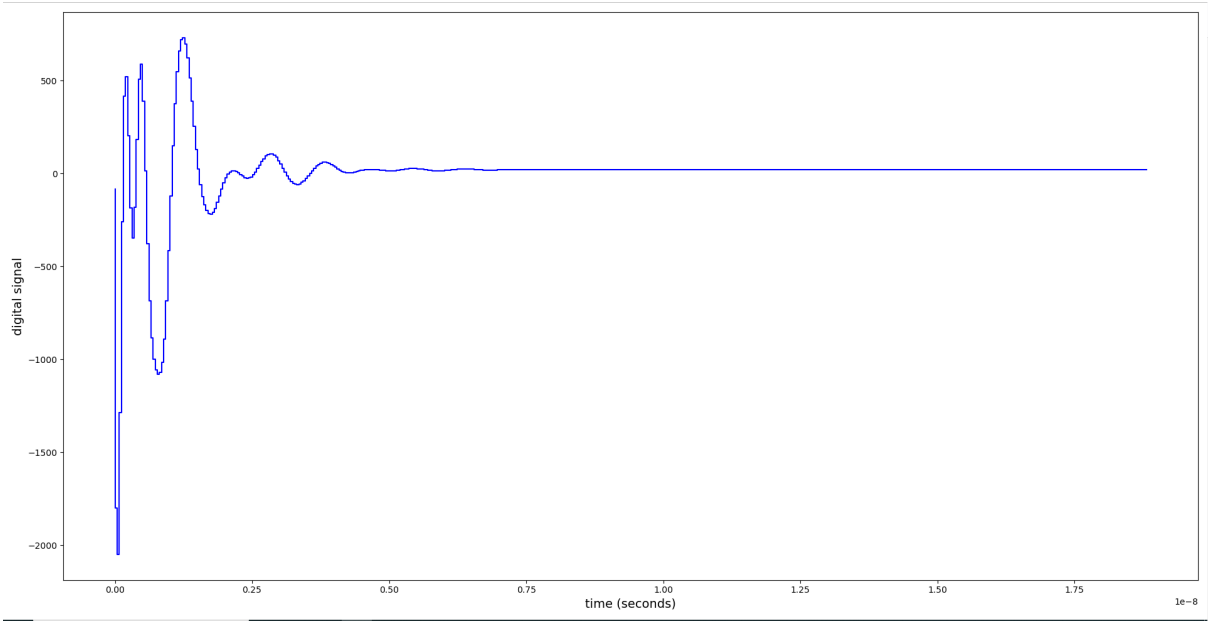


Figure 5-58 Digitised Signal of Figure 5-57

A sample of the stored data by the GSSI equipment is show in Table 5-6, which is not a raw digital signal, but it has had some signal processing performed by the GPR system as discussed in chapter 4. It can be deduced that there must be an up-sampling step or a digital to analogue conversion step which would further reduce the data size and then an interpolation step to up sample the data. A simplistic demonstration is shown in Figure 5-59 with the resulting data in Table 5-9 and the revised B-Scan image is shown in Figure 5-60.

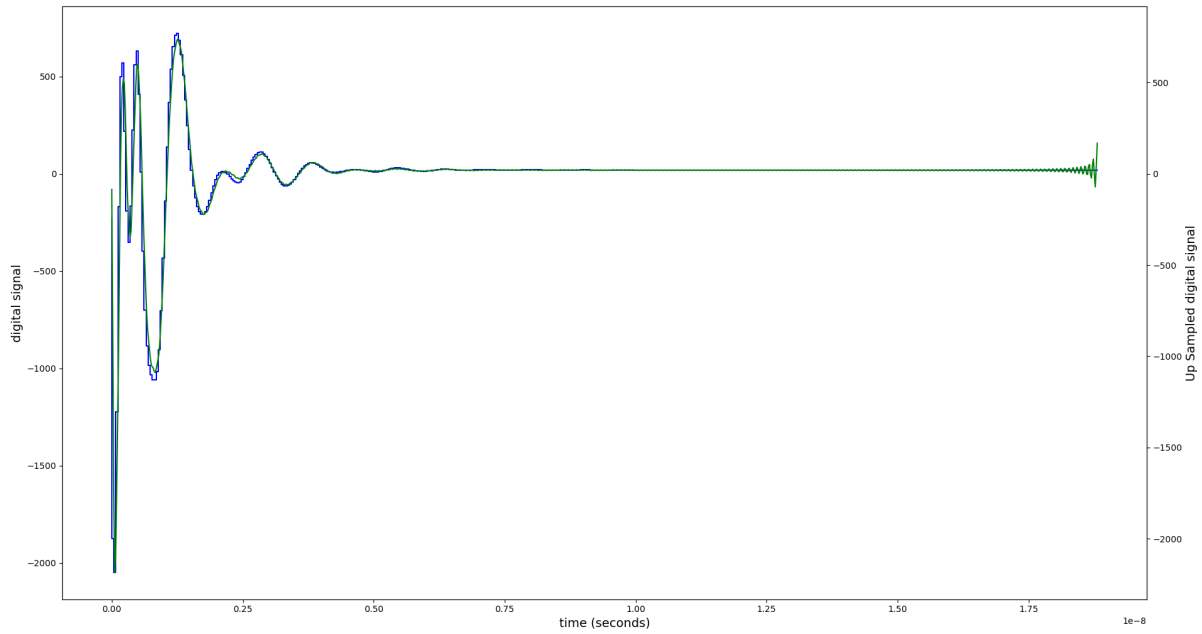


Figure 5-59 Before and After Up Sampling (Green and Blue)

	0	1	2	3	4	5	6	7	8	9	10
Before	-85		-1798		-2048		-1285		-260		416
After	-85.00	-853.17	-1731.24	-2186.30	-2102.77	-1762.26	-1393.89	-973.96	-457.41	30.66	337.78

Table 5-9 Showing a Sample of Data Before and After Sampling

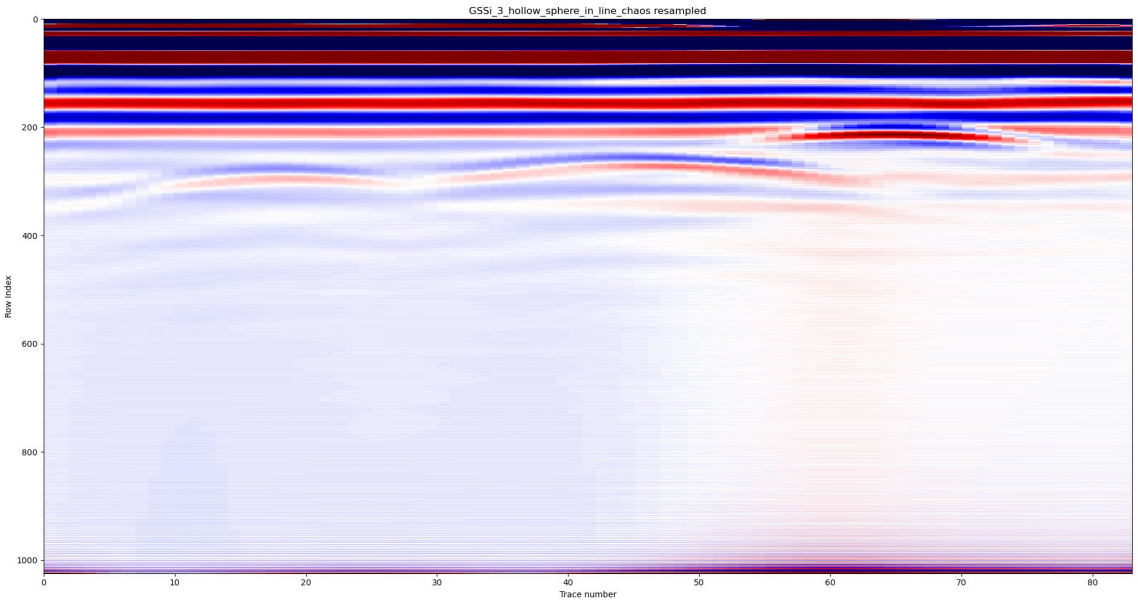


Figure 5-60 B-Scan of 3 Spheres in a Chaotic Medium after Resampling

The inclusion of a median filter in the pre-processing part of the network actually reduces the overall effectiveness of the mapping between the spaces.

While the above process does not fully recreate the GSSI data, it does show that it is possible to go some way to replicating the GSSI process. However, the better approach would be to use the signal generator and the antenna from the GSSI and develop bespoke signal processing hardware which would allow the simulation suite to fully represent the data and improve the corelation between the model and the real-world data.

The technology presented in this thesis shows that it is possible to map between a GPR B-scan into the 3D graphical domain. This opens exciting opportunities in the field of unexploded ordnance detection, archaeological and defect detection in civil engineering as well as agriculture. The main improvement over the current systems is that it will reduce the number of required scans performed to get an overall view of the subsurface. This leads to the main commercial benefit, namely the reduction in costs by both reducing hardware costs and the time taken to scan a location. Using current technology requires several B-scans to be taken over the site of interest, and then using software methods similar to that described in chapter 3 it is possible to predict an image. In reducing the number of traverses across a field which farmers prefer to avoid as it causes soil compression and reduction of drainage, an array of antennas would be required. It can therefore be seen that if the is a single traverse with a single antenna, this would be both a significant saving in cost and time.

This reduction in costs and increase in functionality, should allow the use of GPR to become more prevalent for viewing the subsurface world.

The principal limitation of the approach adopted is that the scanning resolution of the system does not match the resolution of the resolution of the simulation. One

method of improving the resolution is to apply a deconvolution technique (BENEDETTO *et al.*, 2017), which works on the assumption that the GPR signal is a convolution between a transmitted wavelet $w(t)$ and target reflectivity $e(t)$ as shown in 5.22. This approach requires isolating the term e(t) by accounting for the reflected signal and retrieval of the source wavelet.

$$x(t) = w(t) * e(t) \quad \textbf{5.22}$$

The main limitation is that, by default deconvolution is an ill-posed problem and therefore creates an approximation and not an exact solution, and a secondary issue is that since in GPR the wavelet is non-stationary and mixed has led to some issues in the effectiveness of this approach.

Existing approaches to the inverse problem require bespoke models for specific use-cases, if the boundary conditions change, then the model needs to be updated. Using the approach proposed in this thesis allows for more flexibility in the boundary conditions since, once trained it is relatively simple via transfer learning (the process of retraining models to new situations) is simple and relatively low resource requirement. That said, if the initial training data is sufficiently representative of conditions in the real-world, then retraining would be rarely required.

Using existing approaches requires multiple scans to be able to interpolate the 3rd dimension accurately, however with this approach the principal benefit is that, whilst additional scans would improve the accuracy of the model it is not a necessity. Since the approach generates 3D solids, it is possible for the farmers to perform harvesting, based on accurate information and determining the state of the crop based on a small section of the field and experience. In addition to this if the farmer

has an accurate image of the field, they will be able to harvest areas that are ready to harvest and leave other areas to mature.

It can also be concluded that computationally this does not seem to be the most efficient approach. The resources deployed in solving this problem were insufficient. Since the start of this project, transformer architecture (Vaswani *et al.*, 2017) developed to become a dominant force in the generative field with well-known applications such as Chat-GPT (OpenAI, 2022) and in the field of image-to-image translation (Zheng *et al.*, 2022) it has been shown to perform better than CycleGAN discussed in section 2.1.3. The results for this are shown in Table 5-10 below.

Method	Cityscapes				Cat → Dog	H → Z
	mAP↑	pixAcc↑	clsAcc↑	FID↓	FID↓	FID↓
CycleGAN [59]	20.4	57.2	25.4	76.3	85.9	77.2
UNIT [29]	16.9	58.4	22.5	91.4	104.4	98.0
DRIT++ [27]	17.0	60.3	22.2	96.2	123.4	88.5
Distance [2]	8.4	47.2	12.6	75.9	155.3	67.2
GcGAN [9]	21.2	65.5	26.6	57.4	96.6	86.7
CUT [34]	24.7	68.8	30.7	56.4	76.2	45.5
LSeSim [58]	–	73.2	–	49.7	–	38.0
I2ISA* [20]	20.0	69.5	25.8	151.8	103.6	248.2
CycleGAN* [59]	22.1	67.8	28.7	65.1	87.6	77.2
CUT* [34]	29.0	83.7	35.8	47.8	74.4	36.4
LSeSim* [58]	28.9	75.7	37.4	55.9	72.8	38.9
ITTR (CUT)	**32.5**	**86.0**	**39.7**	**45.1**	**68.6**	**33.6**
ITTR (LSeSim)	28.9	78.8	36.7	53.3	68.7	36.5

Table 5-10 Comparison between ITTR and state-of-arts on three datasets. Evaluation metric with sign ↓ indicates that lower is better, while ↑ indicates higher is better (Zheng *et al.*, 2022)

It became clear that the main limitation for this work was a hardware one, this could have been addressed by using the power of the cloud to train the model.

5.7 Chapter Summary

In this chapter, the design principles have described in detail in sections 5.1 to 5.1.2. The solution adopted utilise combinations of CVAEs & GANs both in cycle GANs and CVAE-GAN configurations. The theory behind the various building blocks for the architecture used in the implementation has been described in sections 5.1.3 to

5.2. In sections 5.3 and 5.4 it has been shown that it is possible to use deep learning to achieve the following aims:

a) Map a 2D B-scan to a corresponding 2D image (section 5.3.2).
b) Map a 2D B-scan to a corresponding 3D image (section 5.4.5).

The theoretical solutions have then been tested on real world data as described in section 5.5. in section 5.6, the results are discussed along with suggestions to improve the outputs which form the backbone of chapter 6.

6 Further Work

During the research towards this thesis and the development of the computational solution, several items have been identified that could benefit from further work in order to make the solution commercially viable. These could not be pursued within this work due to cost and time constraints.

The following items were identified for further work:

- Change the 2D-to-2D architecture to a colour based one to improve the results in section 5.3.2.
- Investigate the best approach for resampling the voxel grid to a real-world value by using affine transforms or b-splines as mentioned in section 5.4.
- Further enhancements, including the use of additional lateral B-Scans to improve the 3D mapping.
- The use of the unprocessed antenna signal from the field to better match the simulation data.
- Improvements are required to reconstructing the GPR scan resolution using deconvolutional techniques as discussed in section 5.6.
- Quantization of the final model to allow deployment into the field, explained in section 6.1.
- Improvements in computational efficiency, for example 3D kernels, discussed in section 6.2.
- Using additional scans (more than 1) per row.
- Using better computer hardware to allow for bigger models required to use a traditional cycle GAN type architecture.

6.1 Quantization

Quantization is the process used to minimise trained deep learning models, so they use minimal resources to perform. Quantization techniques could be applied to the model to reduce its size and make it executable on edge devices such as tablets or smart phones. The benefit would be ease of use by farmers in the field.

During quantization, the floating-point numbers used in the weights and bias of the model are converted into low bit width numbers often in base 2 integer increments. The most common values for this are 8- and 4-bit integer numbers. This has an impact on the accuracy of the model, but using modern techniques it can be minimised, as shown in Table 6-1 below. Looking on the diagonal of the table shows an error rate of 8.3% for 4-bit quantisation, 6.95% for 8 bit and 6.98 for floating point errors.

Activation Bit-width	Weight Bit-width			
	4	8	16	Float
4	8.30	7.50	7.40	7.44
8	7.58	6.95	6.95	6.78
16	7.58	6.82	6.92	6.83
Float	7.62	6.94	6.96	6.98

Table 6-1 CIFAR-10 Classification Error Rate with Different Bit-Width Combinations
(Lin, Talathi and Annapureddy, 2016)

Quantization was not investigated during the thesis, which focused on developing a viable solution to the inverse problem rather than its implementation in the field.

6.2 3D Convolution Kernels

The incorporation of a 3D convolutional kernel based on the octree data structure was suggested in section 5.4.1.4.

There are a few options for improving the allocation of computer hardware resources to optimise the solution further. One option is to change the data structure used in

the 3D convolution Kernels. This can be achieved by using an octree instead of a 3D array. An octree is a derivative of the binary tree, but instead of two child nodes there are eight child nodes for each root. This data structure is mainly used in 3D graphical applications and is a highly efficient way of storing data as only areas where fidelity is required are divided as shown in Figure 6-1.

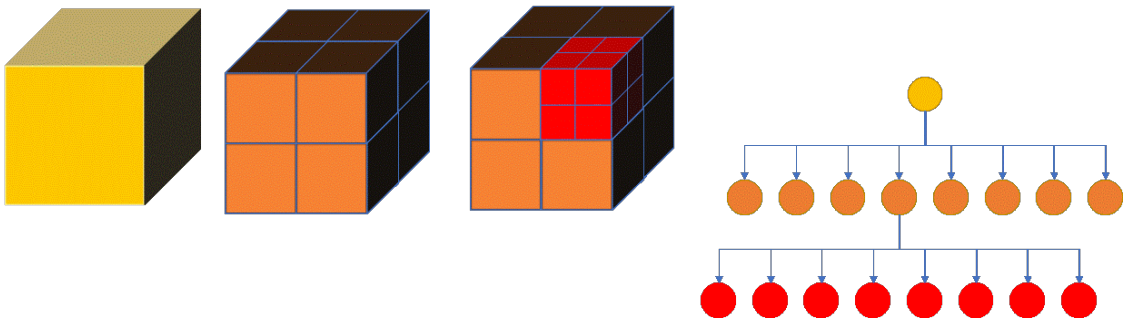


Figure 6-1 3 Level Octree Data Structure

Each node can be made to represent a region of space, where the point stored in each node represents the centre of the subdivision, while the root node defines the whole space. This is a very efficient way of storing data and an example of their use is demonstrated in (Wang *et al.*, 2017) which shows that it is possible to build an octree-based CNN with a comparison of memory requirements shown in Figure 6-2 below. This data structure has potential to make 3D convolutions a realistic proposition at high resolutions.

Method	16^3	32^3	64^3	128^3	256^3
O-CNN	0.32GB	0.58GB	1.1GB	2.7GB	6.4GB
full voxel+binary	0.23GB	0.71GB	3.7GB	Out of memory	Out of memory
full voxel+normal	0.27GB	1.20GB	4.3GB	Out of memory	Out of memory

Figure 6-2 Memory Requirements Comparison for a batch size of 32 (Wang *et al.*, 2017)

The convolutional operations work by identify the neighbouring octants at the same level within the tree, and the convolutional operator is written as 6.1.

$$\Phi_C(O) = \sum_n \sum_i \sum_j \sum_k W_{ijk}^{(n)} . T^{(n)}(O_{ijk}) \quad \textbf{6.1}$$

O_{ijk} is the neighbouring octant of O, $T^{(n)}$ represents the nth channel feature vector associated with O_{ijk} with $W_{ijk}^{(n)}$ being the weighting for the convolution operation. In this form, it is possible to convert the convolution operation into a matrix product. For a given kernel size "K", the operator requires K^3-1 neighbouring cells. When "K" is large, this requires a large amount of I/O processing, which leads to a large memory footprint. To counter this a hash table is used. The approach adopted in this paper results in stride levels which are of size equal to 2^n level. Introducing octrees to the architecture will results in the memory requirements being $O(n^2)$ compared to $O(n^3)$ of a conventional full voxel solution. This approach has not been fully developed as part of this thesis but could be a basis for future enhancements to the technology presented here.

6.3 Summary

In this chapter, future work has been identified to take the project forward with the aim of improving the results from the methods deployed in this thesis. These approaches are focused in two main regions, the first looks at the performance of the both the hardware and the deep learning approaches, the seconds approaches attempt to improve efficiency within the deep learning by looking at methods to reduce the computational requirements deployed to solve the mathematical problems.

7 Conclusion

This thesis has investigated a data driven approach to solving the inverse problem using deep learning. Chapter 2 presented a review of current literature on the use of deep learning and GPR systems, along with current analytical solutions. Chapter 3 delved into an explanation of an empirical solution currently used to solve the inverse problem, and the reasons it was found lacking in the case of tubers in soil. Chapter 4 discussed the GPR scanning hardware selected for use in this thesis, its limitations, the experiments conducted, the method used for collecting real data, and the simulation tools utilised. Chapter 5 explained the design principles for the data driven solutions that was tested with both simulated and real-world data. In this chapter the results have been compared back to state-of-the-art academic papers. The final chapter builds on the findings from chapter 5 and suggests future directions for this work to address the issues highlighted.

The thesis has achieved the objectives defined at the outset to:

7.1. **Identify and size individual tubers using existing geophysics techniques**. Working with B-hive there has been some success in detecting and identifying potatoes, though the resolution of the data leaves a lot to be desired. The signal strength is representative of the bulk biomass however the individual tuber size is unobtainable with the current system.

7.2. **Investigate whether deep learning can significantly improve the accuracy of quantification of tubers whilst below ground**. This research has shown that it is possible to correlate the simulation results against the GPR signals, and that it is possible to create a 2D image from a single B-Scan. It is also possible to create a mapping using the B-scan to generate 3D images, however

this is reliant on improving the scanning hardware and gaining access to signal straight from the antenna before any processing apart from filtering and digitising.

A comprehensive literature review was conduct and has shown that although some foundation work was done in this field that was built upon to create this solution, no published work has achieved this solution thus far. It has been identified in (Dai *et al.*, 2022) that a suitable architecture to produce the 3D mapping results has yet to be found, a gap that this thesis goes someway to closing.

Along the way, the thesis explained in detail what an inverse problem is, the mathematics required to solve it, and why a data driven solution is more applicable in this case.

The work proved it is possible to utilise machine learning to solve the inverse problem of ground penetrating radar and its application to potato farming. This was accomplished by developing a novel algorithm, based on published papers in similar fields. The algorithm was developed to generate 3D scans from 2D B-scans, which has proven to produce positive results for predicting the size and quantity of tubers in the growing stages. While developing the computational solution, the theoretical model has been studied and explained, and a 2D-to-2D transformation model was developed as an intermediary approach for mapping the 2D space. The proof of concept can be used in agriculture but would have further implementations in other fields, such as civil engineering, airport security, geological surveying, and humanitarian mine disposal systems.

Although the novel algorithm can be improved on further with additional computational hardware and development of some novel concepts mentioned in this

thesis, the thesis has proven the concept and detailed possible next steps. If these recommendations are implemented, there is no reason deep learning cannot be used to create improved 3D images from B-scan data further. Additional improvements to the pre-processing of the input data will also have a beneficial effect on the results for the real-world data.

The algorithm developed and tested as part of this thesis, with the improvements identified in chapter 6 will further enhance the results achieved thus far and facilitate the transformation of the research project into a commercially viable solution.

8 Acknowledgements

I would like to thank the following people for their help in this undertaking.

- B-Hive for sponsoring this work, access to equipment and agricultural knowledge.
- Laura and Paul Collett for proof reading.
- Dr Mark Hansen and Prof. Lyndon Smith for supervision, wisdom, encouragement, and support.

9 References

A.P., A. (2005) Ground Penetrating Radar, in near surface Geophysics. *Investigations in Geophysics*. 13 pp. 357–438.

Annan, A.P. (2009) Electromagnatic Principles of Ground Penetrating Radar. *Ground Penetrating Radar: Theory and Applications*. 1 (1), pp. 3–41.

Anon (2023) *Intersection over Union (IoU) for object detection*. Available from: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ [Accessed 22 May 2022].

B-Hive and Lewis, M. (2021) *No Title*.

BENEDETTO, A., TOSTI, F., CIAMPOLI, L.B. and D'AMICO, F. (2017) *An overview of ground-penetrating radar signal processing techniques for road inspections*. Available from: https://core.ac.uk/download/pdf/46597923.pdf.

Born, M. (1926) Quantenmechanik der Stossvorgänge. *Zeitschrift für Physik*. pp. 803–827.

Cassidy, N. (2009) *Ground Penetrating Radar: Theory and Applications*. In: Harry Jol (ed.). First edit. (no place) Elsevier. pp. 41–72.

Chen, X., Kingma, D.P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I. and Abbeel, P. (2017) Variational lossy autoencoder. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. pp. 1–17.

Cherry, E.C. (1953) Coktail Party Effect Cherry 1953.pdf *The Journal of the*

Acoustical Society of America 25 (5) p.pp. 975–979.

Chew, W.C. and Wang, Y.M. (1990) Reconstruction of two-dimensional permittivity distribution using the distorted Born iterative method. *IEEE Trans. Med. Imag*. 9 pp. 218–225.

Dai, Q., Lee, Y.H., Member, S., Sun, H., Ow, G., Lokman, M., Yusof, M., Yucel, A.C. and Member, S. (2022) *DMRF-UNet: A Two-Stage Deep Learning Scheme for GPR Data Inversion under Heterogeneous Soil Conditions*.

Delgado, A., Novo, A. and Hays, D.B. (2019) Data acquisition methodologies utilizing ground penetrating radar for cassava (Manihot esculenta crantz) root architecture. *Geosciences (Switzerland)*. 9 (4), . doi:10.3390/geosciences9040171.

Fomel, S. (2000) *Inverse B-spline interpolation*. 108 pp. 1–29.

Giannakis, I., Giannopoulos, A. and Warren, C. (2019) Realistic FDTD GPR Antenna Models Optimized Using a Novel Linear/Nonlinear Full-Waveform Inversion. *IEEE Transactions on Geoscience and Remote Sensing*. 57 (3), pp. 1768–1778. doi:10.1109/TGRS.2018.2869027.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Bing, X., Warde-Farley, D., Ozair†, S., Courville, A., Bengio, Y. and Wang, Y.Y. (2016) Generative Adversarial Nets. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* [online]. 08-12-Sept pp. 715–719. Available from: http://www.github.com/goodfeli/adversarialdoi:10.21437/Interspeech.2016-402.

Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B. and Smola, A.J. (2006) A Kernel Method for the Two-Sample-Problem. *NIPS 2006: Proceedings of the 19th International Conference on Neural Information Processing Systems*. 1 pp. 513–520. doi:10.7551/mitpress/7503.003.0069.

GSSI (2022) *GSSI*. Available from: https://www.geophysical.com/products/structurescan-pro [Accessed 22 May 2022].

Guo, L., Lin, H., Fan, B., Cui, X. and Chen, J. (2013) Forward simulation of root's ground penetrating radar signal: Simulator development and validation. *Plant and Soil*. 372 (1–2), pp. 487–505. doi:10.1007/s11104-013-1751-8.

Al Hagrey, S.A. (2007) Geophysical imaging of root-zone, trunk, and moisture heterogeneity. *Journal of Experimental Botany*. 58 (4), pp. 839–854. doi:10.1093/jxb/erl237.

Hou, F., Lei, W., Li, S. and Xi, J. (2021) Deep Learning-Based Subsurface Target Detection from GPR Scans. *IEEE Sensors Journal*. 21 (6), pp. 8161–8171. doi:10.1109/JSEN.2021.3050262.

Irving, J. and Knight, R. (2006) Numerical modeling of ground-penetrating radar in 2-D using MATLAB. *Computers and Geosciences*. doi:10.1016/j.cageo.2005.11.006.

J. Ježová , L. Mertens, S.L. (2016) Ground-penetrating radar for observing tree trunks and other cylindrical objects. *Construction and Building Materials*. 123 pp. 214–225.

Jiajun, W., Zhang, C., Xue, T., Freeman, W.T. and Tenenbaum, J.B. (2016)

Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. (Nips), .

Jin, K.H., McCann, M.T., Froustey, E. and Unser, M. (2017) Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Transactions on Image Processing*. doi:10.1109/TIP.2017.2713099.

Kafedziski, V., Pecov, S. and Tanevski, D. (2018) Detection and Classification of Land Mines from Ground Penetrating Radar Data Using Faster R-CNN. In: *2018 26th Telecommunications Forum (TELFOR)* [online]. November 2018 (no place) IEEE. pp. 1–4. Available from: https://ieeexplore.ieee.org/document/8612117/doi:10.1109/TELFOR.2018.8612117 [Accessed 2 April 2019].

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. and Aila, T. (2020) Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 8107–8116. doi:10.1109/CVPR42600.2020.00813.

Ketkar, N. (2017) Introduction to Deep Learning. In: *Deep Learning with Python*. doi:10.1007/978-1-4842-2766-4_1.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2 pp. 1097–1105.

Lin, D.D., Talathi, S.S. and Annapureddy, V.S. (2016) Fixed point quantization of deep convolutional networks. *33rd International Conference on Machine Learning, ICML 2016*. 6 pp. 4166–4175.

Lippman, B.A. and Schwinger J (1950) *Variational Principles for Scattering Processes*. pp. 469–480.

Liu, B., Ren, Y., Liu, H., Xu, H., Wang, Z., Cohn, A.G. and Jiang, P. (2021a) GPRInvNet: Deep Learning-Based Ground-Penetrating Radar Data Inversion for Tunnel Linings. *IEEE Transactions on Geoscience and Remote Sensing*. 59 (10), pp. 8305–8325. doi:10.1109/TGRS.2020.3046454.

Liu, J., Xu, X., Liu, Y., Rao, Z., Smith, M.L., Jin, L. and Li, B. (2021b) Quantitative potato tuber phenotyping by 3D imaging. *Biosystems Engineering*. 210 pp. 48–59. doi:10.1016/j.biosystemseng.2021.08.001.

Liu, X., Dong, X., Xue, Q., Leskovar, D.I., Jifon, J., Butnor, J.R. and Marek, T. (2018) Ground penetrating radar (GPR) detects fine roots of agricultural crops in the field. *Plant and Soil*. doi:10.1007/s11104-017-3531-3.

Millington, T.M., Cassidy, N.J., Nuzzo, L., Crocco, L., Soldovieri, F. and Pringle, J.K. (2011) Interpreting complex, three-dimensional, near-surface GPR surveys: An integrated modelling and inversion approach. *Near Surface Geophysics*. 9 (3), pp. 297–304. doi:10.3997/1873-0604.2010010.

MIT (2021) *Open3D*. Available from: http://www.open3d.org/.

OpenAI (2022) *Introducing ChatGPT*. Available from: https://openai.com/blog/chatgpt [Accessed 1 December 2023].

PACE, W.E., WESTPHAL, W.B., GOLDBLITH, S.A. and VAN DYKE, D. (1968) Dielectric Properties of Potatoes and Potato Chips. *Journal of Food Science*. 33 (1), pp. 37–42.

Peplinski, N.R., Fawwaz, U.T. and Myron, D.C. (1995) *IEEE Xplore Full-Text PDF:*

Available from: https://ieeexplore-ieee-org.ezproxy.uwe.ac.uk/stamp/stamp.jsp?tp=&arnumber=387598 [Accessed 17 November 2021].

Persico, R. (2014) *Introduction to Ground Penetrating Radar: Inverse Scattering and Data Processing*. 1st edition. (no place) Wiley.

Persico, R., Ludeno, G. and Lambot, S. (2017) Two-Dimensional Linear Inversion of GPR Data with a Shifting Zoom along the Observation Line. *Remote Sensing*. pp. 1–15. doi:10.3390/rs9100980.

Picetti, F. (2023) How Deep Learning Can Help Solving Geophysical Inverse Problems. In: Carlo G. Riva (ed.). *SpringerBriefs in Applied Sciences and Technology* 1st edition. Milan: Springer. pp. 141–152. doi:10.1007/978-3-031-15374-7_10.

Puzyrev, V., Salles, T., Surma, G. and Elders, C. (2022) Geophysical model generation with generative adversarial networks. *Geoscience Letters* [online]. 9 (1), . Available from: https://doi.org/10.1186/s40562-022-00241-ydoi:10.1186/s40562-022-00241-y.

Redmon, J. and Farhadi, A. (2018) *YOLOv3: An Incremental Improvement*. Available from: http://arxiv.org/abs/1804.02767.

Reynolds, J.M. (2011) *An introduction to applied and environmental geophysics*. (no place) Wiley-Blackwell.

Sonoda, J. and Kimoto, T. (2019) Object identification form GPR images by deep learning. In: *Asia-Pacific Microwave Conference Proceedings, APMC* [online]. November 2019 (no place) IEEE. pp. 1298–1300. Available from:

https://ieeexplore.ieee.org/document/8617556/doi:10.23919/APMC.2018.8617556 [Accessed 2 April 2019].

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015) Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 07-12-June pp. 1–9. doi:10.1109/CVPR.2015.7298594.

Szymczyk, P. and Szymczyk, M. (2015a) Classification of geological structure using ground penetrating radar and Laplace transform artificial neural networks. *Neurocomputing*. 148 pp. 354–362. doi:10.1016/j.neucom.2014.06.025.

Szymczyk, P. and Szymczyk, M. (2015b) Supervised learning Laplace transform artificial neural networks and using it for automatic classification of geological structure. *Neurocomputing*. 154 pp. 70–76. doi:10.1016/j.neucom.2014.12.018.

Travassos, X.L., Avila, S.L. and Ida, N. (2018) Artificial Neural Networks and Machine Learning techniques applied to Ground Penetrating Radar: A review. *Applied Computing and Informatics* [online]. Available from: https://www.sciencedirect.com/science/article/pii/S2210832718302266doi:10.1016/J.ACI.2018.10.001 [Accessed 2 April 2019].

Ulaby, F. and Ravaioli, U. (2015) *Fundamentals of Applied Electromagnetics*. 7th edition.

Utsi, E.C. (2017) *Ground Penetrating Radar Theory and Practice*. 1st edition. (no place) Elesvier.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) Attention is all you need. *Advances in Neural Information Processing Systems*. 2017-Decem (Nips), pp. 5999–6009.

Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y. and Tong, X. (2017) O-CNN: Octree-based convolutional neural networks for 3D shape analysis. In: *ACM Transactions on Graphics*. 2017 (no place) Association for Computing Machinery. doi:10.1145/3072959.3073608.

Wang, Z., Simoncelli, E.P. and Bovik, A.C. (2003) MULTI-SCALE STRUCTURAL SIMILARITY FOR IMAGE QUALITY ASSESSMENT. *Proceedings of the 37th IEEE Asilomar Conference on Signals, Systems and Computers*.

Warren, C. and Giannopoulos, A. (2011) Creating finite-difference time-domain models of commercial ground-penetrating radar antennas using Taguchi's optimization method. *Geophysics*. 76 (2), pp. 37–47. doi:10.1190/1.3548506.

Warren, C. and Giannopoulos, A. (2019) *gprMax*. Available from: http://www.gprmax.com/.

Xu, Z., Valdes, C. and Clarke, J. (2018) Existing and Potential Statistical and Computational Approaches for the Analysis of 3D CT Images of Plant Roots. *Agronomy*. doi:10.3390/agronomy8050071.

Yang, F. and Ma, J. (2019) Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*. 84 pp. 583–599.

Yu, C., Yuan, M. and Liu, Q.H. (2009) Reconstruction of 3D objects from multi-frequency experimental data with a fast DBIM-BCGS method. *Inverse Problems*. doi:10.1088/0266-5611/25/2/024007.

Yu, S. and Ma, J. (2021) Deep Learning for Geophysics: Current and Future Trends. *Reviews of Geophysics*. 59 (3), pp. 1–36. doi:10.1029/2021RG000742.

Zhang, X., Xue, F., Wang, Z., Wen, J., Guan, C., Wang, F., Han, L. and Ying, N. (2021) A novel method of hyperbola recognition in ground penetrating radar (Gpr) b-scan image for tree roots detection. *Forests*. 12 (8), . doi:10.3390/f12081019.

Zheng, W., Li, Q., Zhang, G., Wan, P. and Wang, Z. (2022) *ITTR: Unpaired Image-to-Image Translation with Transformers*. Available from: http://arxiv.org/abs/2203.16015.

Zhu, J., Park, T., Efros, A.A., Ai, B. and Berkeley, U.C. (2017) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *ICCV*.

Zong, Z., Chen, C., Mi, X., Sun, W., Song, Y., Li, J., Dong, Z., Huang, R. and Yang, B. (2019) A deep learning approach for urban underground objects detection from vehicle-borne ground penetrating radar data in real-time. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 42 (2/W16), pp. 293–299. doi:10.5194/isprs-archives-XLII-2-W16-293-2019.

Appendix A **Classification** using 2D Pretrained CNNs

The aim of this appendix is to classify the quantity of objects based on the GPR responses, later in this section the work classifies on two features; size and material with the view of removing false positives from the solution and gaining an accurate prediction of both the biomass and the best time to intervene with chemicals to boost yield. Based on the published work in section 2.1.2, it was thought that CNNs could be used to identify organic material in the 2D B-scan. However, the work in this chapter was not progressed further because the direction chosen was to use to generate images directly from B-scan as discussed in chapter 0.

The CNNs chosen are existing networks, available for download from MATLAB as part of the deep learning toolbox. Each of the networks were trained for 60 epochs with the training accuracies and number of layers summarised in Table Appendix A-1. Once trained, 16 images are passed through the network to assess the performance. The results are shown in Figure Appendix A-1, Figure Appendix A-2 & Figure Appendix A-3. Each figure shows the 16 images selected at random with the predicted number of items and the probability that the network has associated to the identification. The predicted results are then compared against the actual results in the truth tables shown in Table Appendix A-2 for transfer learning on quantity.

Architecture	Layers	Training Accuracy
Alex Net (Krizhevsky, Sutskever and Hinton, 2012)	25	80.25%
GoogleNet (Szegedy *et al.*, 2015)	144	73.46%
RESnet101 (Ketkar, 2017)	347	64.81%

Table Appendix A-1 Transfer Learning Accuracy

image	AlexNet	GoogleNet	ResNet 101
1	F	T	T
2	T	T	T
3	F	T	T
4	T	F	F
5	F	T	F
6	T	T	T
7	F	T	F
8	T	T	T
9	T	T	T
10	T	T	T
11	F	T	T
12	T	T	T
13	T	F	T
14	F	T	T
15	T	F	F
16	T	T	F
Accuracy	**62.50%**	**81.25%**	**68.75%**

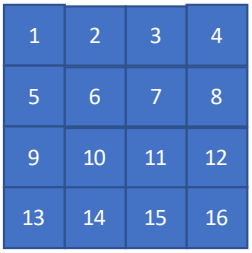


Table Appendix A-2 Transfer Learning Truth Table for Quantity

The following sections will detail the results achieved with publicly available CNNs.

Appendix A.1 Alex Net

During training, the accuracy for Alexnet becomes stable and alternates around 80% after epoch 30. Despite having the highest level of convergence accuracy, the truth table (Table Appendix A-2) shows that using 16 random samples shown in Figure Appendix A-1 Alexnet performed the worst of the 3 selected models. Further work is required to understand if this down the to the random selection, the amount of training data or due to the architecture of the model.

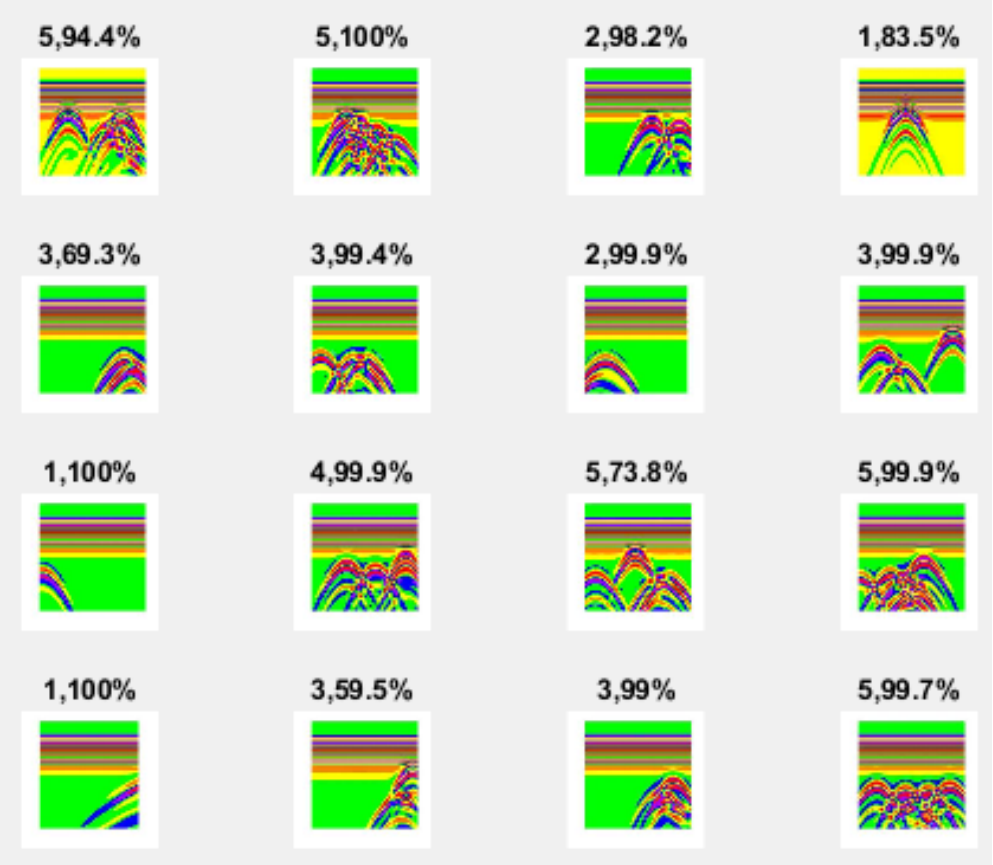


Figure Appendix A-1 Alexnet Training Results for Quantity & Predicted Confidence

Appendix A.2 GoogleNet & RESnet101

During training, the accuracy for both GoogleNet and RESnet101 shows a similar behaviour to Alexnet (with the exception of 70% & 64% accuracy respectively) however the validation loss is increasing on both networks. This phenomenon is often caused by overfitting. The results for both of these models are shown in Figure Appendix A-2 & Figure Appendix A-3 for quantity.

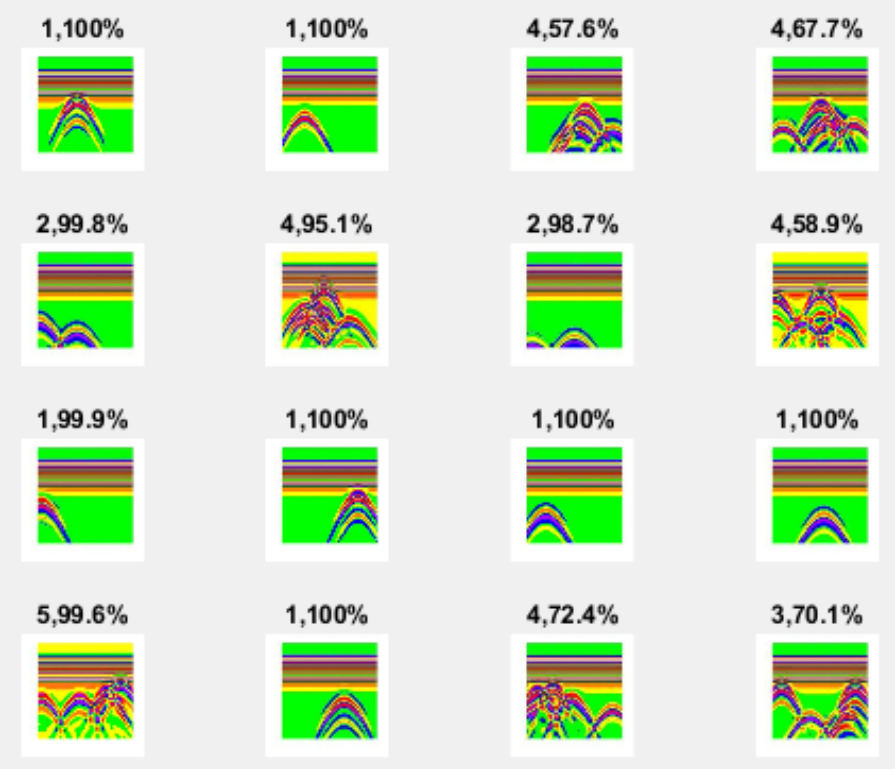


Figure Appendix A-2 GoogleNet Training Results for Quantity & Predicted Confidence

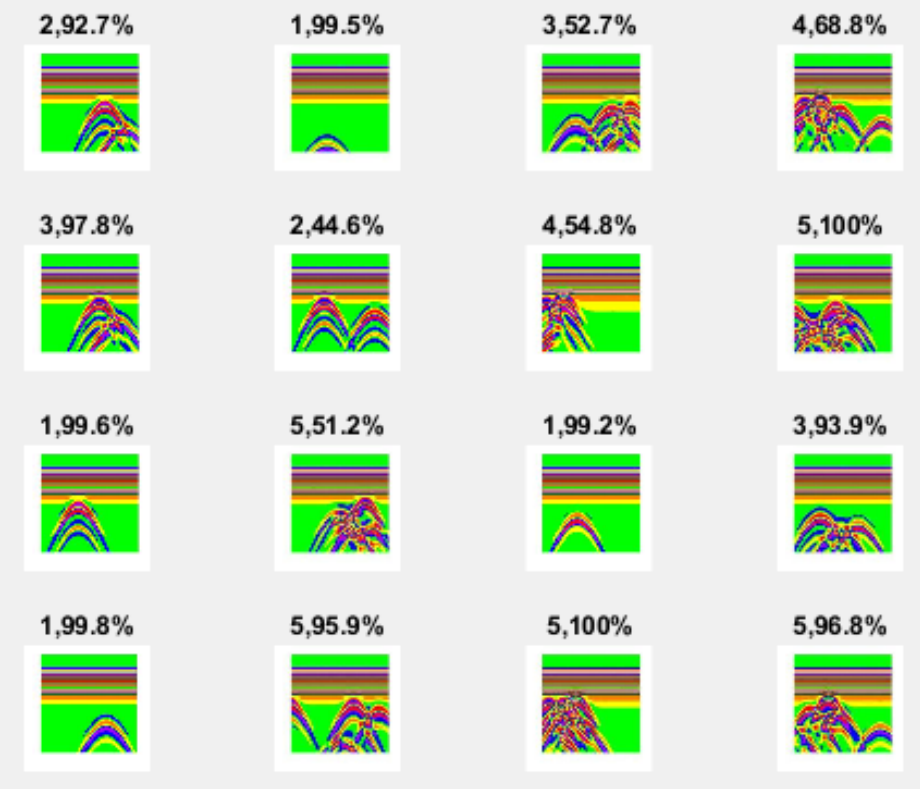


Figure Appendix A-3 RESnet101 Training Results for Quantity & Predicted Confidence

Appendix A.3 Inception net

It can be clearly seen from the previous section that the model which has the best reliability for the classification on quantity of the GPR B-Scan result is inception net. Because the classification is going to be on images and not the GPR signal, inception net has been chosen as the way forward as it is expected to be both the most reliable and provide flexibility should the problem increase in complexity.

Currently the convolutional filters are the same as per the baseline (Szegedy *et al.*, 2015) model with the exception of the interim and output layers which are controlled by the use of the cross-entropy loss function. The cross-entropy loss function allows multiple classification problems to be solved.

When creating the network, it was found that the loss based solely on the output vs target term the loss function did not seem to have enough “power” to propagate

through the whole network, as a result interim loss functions are calculated in inception modules 4 & 7. These values are then weighted by 30% for inception module 4 and 25% for inception module 7.

The first version of the classifier only classified on XXX or YYY, the second version started to look at multiple classification by classifying on size (Small, Medium and Large) and the third version (results presented in Appendix A.4) classifies using 10 “buckets” for size and 3 materials.

Appendix A.4 Results

The condensed output is a 10x3 array, where the rows are buckets of sizes, and the columns represent the material shown in Table Appendix A-3. The testing size is 200 unseen images.

Object Radius (m)	Organic	Inorganic 1	Inorganic 2
0.020 – 0.022	0	0	0
0.022 - 0.024	0	0	0
0.024 - 0.026	0	0	1
0.026 - 0.028	0	0	0
0.028 - 0.030	0	0	1
0.030 - 0.032	1	1	0
0.032 - 0.034	0	0	0
0.034 - 0.036	0	0	0
0.036 - 0.038	0	0	0
0.038 - 0.040	0	0	0

Table Appendix A-3 Output from the Inception Net

An example of the results from testing are in Table Appendix A-4. Incorrect classifications are highlighted in red. The total number of entries misclassified from the 200-sample size is 247 out of 6000 entries, this gives an error of 4.1%, or 95.9% of entries were classified correctly.

	Actual			Predicted		
Object Radius (mm)	**Organic**	**Inorganic 1**	**Inorganic 2**	**Organic**	**Inorganic 1**	**Inorganic 2**
0.020 – 0.022	1	1	0	1	1	0
0.022 - 0.024	0	0	0	0	0	0
0.024 - 0.026	0	2	0	0	**1**	0
0.026 - 0.028	0	1	1	0	1	**0**
0.028 - 0.030	0	0	0	0	0	0
0.030 - 0.032	0	1	0	0	1	0
0.032 - 0.034	0	0	0	0	0	0
0.034 - 0.036	0	0	0	0	0	0
0.036 - 0.038	0	0	0	0	0	**1**
0.038 - 0.040	0	0	1	0	0	**0**

Table Appendix A-4 Sample Output Classification