

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Subsurface Characterization Using GPR Data by Means of Image Segmentation with YOLO-v8

Duarte Ribeiro Afonso Branco



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Armando Jorge Miranda de Sousa

October 24, 2024

Abstract

Ground Penetrating Radar (GPR) is a geophysical imaging technique used for the characterization of a subsurface's electromagnetic properties, allowing for the detection of buried objects. GPR's major strengths include its versatility across different materials and the high precision, low time-consuming, non-intrusive and non-destructive nature of the scan, making it a cost-effective and environmentally responsible solution. It has applications in many fields of science and engineering, where there is a need to investigate what's beneath a visually opaque surface while preserving the probing environment. In the scope of forest and agricultural applications, GPR can be used to determine soil properties and optimize plantation position, based on information of localization of buried objects.

The characterization of an object's features, such as position, depth and radius, is possible by identifying the distinct hyperbolic signature of objects in GPR scans over a survey line. Traditionally, this process requires human interpretation of the GPR data, which is a difficult operation requiring extensive knowledge of the operating principles of GPR, the propagation of electromagnetic waves and their interactions with a complex medium.

The goal of the dissertation is the development of an automated system to detect and characterize the presence of buried objects through the analysis of GPR data. GPR data processing techniques are used in order to mitigate the effect of noise, cluttering signals and attenuation, thus making it easier to identify the hyperbolic signature. Computer vision techniques are also explored with the goal of further enhancing the interpretation of the data.

State of the art image segmentation deep learning models of the YOLO-v8 family were trained to identify an object's hyperbolic signature. A multi-channel encoding strategy (MCEC) was explored with the aim of maximizing the information inputted into the models. This consisted of training the models with images where different GPR processing and computer vision techniques were stored in each image RGB channel. The analysis of the results of each trained model concluded that more suitable models were obtained when data from complementing techniques were combined. Several MCEC strategies were compared using YOLO-v8 segmentation nano models and, from the best combination, a YOLO-v8 segmentation medium model was trained to further optimize the proposed processing pipeline. Additionally, two approaches were explored and compared when using YOLO to predict the masks. First, the masks were obtained by considering only the results obtained from a single YOLO prediction. Next, a prediction averaging approach based on test-time augmentation was tested. Similar results were obtained in both cases, however, prediction averaging is an interesting approach in reducing the number of false positive and false negative detections. The segmentation masks predicted by the trained YOLO model were fitted to the mathematical model of the GPR hyperbola by using the constrained least-squares method.

A dataset comprising of 294 GPR B-scans was collected and manually annotated, which was enlarged using image augmentation techniques, in order to better train the YOLO model. A second smaller dataset, composed of 30 B-scans, with the object's depth and position ground-truth was collected, in order to verify the precision of the developed system. Both datasets consisted of

B-scans from solid metallic cylinders buried in soil perpendicular to the direction of the scan. The equipment was used the OERAD's Concretto GPR system.

The best MCEC encoding provides the following errors, in average, of the buried object's characteristics: about 3 cm in the lateral position, about 2 cm vertical depth, and about 1 cm in the radius.

Sustainable Development Goals

The 2030 Agenda for Sustainable Development [1], adopted in 2015 by all the United Nations Member States, identifies the actions necessary to address global challenges and achieve a more sustainable future for all. It sets 17 Sustainable Development Goals, composed of a total of 197 Targets, providing a blueprint on the steps to achieve each goal. In figure 1, the Sustainable Development Goals are illustrated.



Figure 1: Sustainable Development Goals. [1]

This dissertation's contributions are related to several of these goals. The advancement of GPR data analysis techniques will impact all of the fields in which GPR is employed, improving the accuracy of object detection and characterization. This would make GPR an even more viable option for the non-destructive inspection of the subsurface. Increasing accuracy would also impact the safety of safety-critical GPR applications, such as infrastructure inspection or utility mapping. The development of automated GPR data analysis techniques increases the potential for its integration as part of automated systems which could have impactful applications, such as autonomous tree plantation. These would contribute to the following Sustainable Development Goals, as listed next.

- Goal 9 - Industry, Innovation and Infrastructure: The development of automated GPR data analysis systems contributes to making this process less reliant on human interpretation of the data. This has the potential of increasing accuracy and safety, reducing the time needed to analyze data, as well as broadening GPR applications, including its implementation as part of automated systems. These factors would make the GPR technology even more appealing in scientific and industrial applications.
- Goal 11 - Sustainable Cities and Communities: The GPR technology has many applications directly related to infrastructure and urban development, namely, utility mapping and structural inspection. The development of automated GPR data analysis systems would in turn

contribute to the utilization of the GPR technology in these fields, by facilitating the analysis of the data, leading to the development of safer and more sustainable cities.

- Goal 15 - Life on Land: The study, conservation and management of ecosystems requires non-destructive probing technologies, which is one of the main advantages of GPR. Geological surveys using GPR systems produce very large quantities of data which can be difficult to interpret. Thus, the development of automated systems for this purpose would be beneficial, by decreasing the time necessary to analyze the data. These could also be later integrated into automated systems for agricultural and forest applications.

Agradecimentos

O desenvolvimento desta dissertação, o conhecimento que adquiri ao longo destes anos e a pessoa que hoje sou deve-se a muitas pessoas, sem as quais nada disto seria possível. A elas gostaria de expressar a minha gratidão.

Aos meus pais, que tudo sempre me deram, o meu mais profundo obrigado! Obrigado por me criarem, por me educarem e pelo vosso compromisso à minha felicidade. Obrigado pela vossa ajuda e apoio e por me tornarem quem hoje sou. Ao meu irmão, parceiro desde sempre, pessoa com quem cresci, obrigado por poder contar sempre contigo.

Aos amigos que tenho a sorte de ter, obrigado por poder partilhar convosco tantos momentos de alegria e companheirismo. Aos de mais longa data, obrigado por estarem sempre presentes, e às amizades que levo comigo deste curso, obrigado por terem partilhado convosco estes anos académicos.

À Faculdade de Engenharia da Universidade do Porto e aos professores que foram parte do meu percurso académico, agradeço todo o rigor e exigência, assim como a transmissão de conhecimento, que contribuíram para o desenvolvimento das minhas competências.

Ao Prof. Doutor Armando Jorge Sousa, agradeço toda a disponibilidade, orientação e aconselhamento durante o desenvolvimento desta dissertação. Ao Mestre Rui Coutinho deixo também o meu cumprimento pela sua colaboração com a equipa de supervisão.

Obrigado a todos!

Duarte Branco

Supervising Team

Rui Manuel Pereira Coutinho, M.Sc. collaborated with the supervising team throughout the development of this dissertation.

Funding and Financing

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within project AgendaTransform, with reference 34.

“Your mind is like a parachute: If it isn’t open, it doesn’t work.”

Buzz Aldrin

Contents

1	Introduction	1
1.1	Context and Motivation	1
1.2	Proposed Solution and Objectives	2
1.3	Contributions	2
1.4	Document Structure	3
2	Background and State of the Art	5
2.1	Fundamentals of GPR Operation	5
2.1.1	Overview	5
2.1.2	Electromagnetism	5
2.1.3	Electromagnetic Wave Properties	7
2.1.4	Hardware	10
2.2	GPR Data Interpretation	16
2.2.1	Scans	16
2.2.2	Hyperbola Models	17
2.2.3	Hyperbola Fitting and Parameter Estimation	21
2.3	GPR Object Detection Methodologies	23
2.3.1	Pre-Processing	24
2.3.2	Traditional Computer Vision	28
2.3.3	AI Based Computer Vision	30
2.4	Summary	39
3	Automatic Object Detection in GPR Data	41
3.1	System Pipeline	41
3.2	GPR Data Gathering	42
3.2.1	Equipment	42
3.2.2	Dataset Creation	43
3.2.3	Data Processing	46
3.2.4	Image Augmentation	49
3.2.5	YOLOv8-seg Applied to Hyperbola Segmentation	52
3.2.6	Hyperbola Prediction Fitting	54
4	Results and Discussion	63
4.1	Analysis of the YOLOv8-seg Nano Models	63
4.2	Evaluation Metrics From Trained YOLOv8-seg Medium Model	65
4.3	Analysis of the Parameters Estimation	67
4.4	Discussion	79

5	Conclusions	81
5.1	Future Work	82
A	Evaluation Metrics of YOLOv8 Nano Models Trained with Multi-channel Encoding Configurations	91
A.1	Multi-Channel Encoding Configuration 1	91
A.2	Multi-Channel Encoding Configuration 2	92
A.3	Multi-Channel Encoding Configuration 3	93
A.4	Multi-Channel Encoding Configuration 4	94
A.5	Multi-Channel Encoding Configuration 5	95
A.6	Multi-Channel Encoding Configuration 6	96
A.7	Multi-Channel Encoding Configuration 7	97
A.8	Multi-Channel Encoding Configuration 8	98
A.9	Multi-Channel Encoding Configuration 9	99
A.10	Multi-Channel Encoding Configuration 10	100
A.11	Multi-Channel Encoding Configuration 11	101
A.12	Multi-Channel Encoding Configuration 12	102

List of Figures

1	Sustainable Development Goals. [1]	iii
2.1	Propagation of EM wave [4]	8
2.2	Relation between frequency and EM wave velocity and attenuation. [4]	9
2.3	Reflection and refraction of an EM wave when hitting the boundary of different mediums. [6]	10
2.4	Block diagram of GPR's main components. [4]	11
2.5	Antenna lobe pattern. Lobe angle Ψ in degrees, signal level in dB. [4]	12
2.6	Propagation shape of EM waves. [8]	12
2.7	Propagation direction of EM signals in GPR applications. [10]	13
2.8	Interactions of closely trimmed pulses. [6]	14
2.9	Vertical and lateral resolution geometrical parameters of objects closely positioned. [6]	15
2.10	Relationship between A-scan, B-scan and C-scan. [12]	17
2.11	Hyperbolic signature of a cylindrical object for common-offset GPR and its relation to the scan's geometrical parameters. [13]	18
2.12	GPR hyperbola model of a cylindrical object, considering zero antenna offset, and the relationship with the scan's geometrical parameters. [16]	19
2.13	Effect of energy radius on power reflectivity. [8]	23
2.14	Time zero offset correction in a GPR A-scan. [2]	25
2.15	<i>Dewow</i> filter correction on a GPR A-scan. [4]	26
2.16	Effect of background removal in a GPR B-scan. [2]	27
2.17	Effect of time varying gain applied to a GPR A-scan. [6]	28
2.18	Structure of a neural network's node. [32]	31
2.19	Organization of nodes into layers in a neural network. [26]	31
2.20	Activation functions. Leaky ReLU with a $\alpha = 0.1$ and ELU with $\alpha = 1$. [34]	33
2.21	Architecture of a convolutional neural network. [38]	35
2.22	Prediction of bounding boxes in YOLO. [42]	36
2.23	YOLOv8 architecture. [43]	37
2.24	Comparison of accuracy vs model size and inference speed of YOLOv8 bounding box version on the COCO dataset. [45]	38
2.25	Architecture of an autoencoder. [48]	39
3.1	System's pipeline.	42
3.2	Experimental setup using plastic box filled with soil. Objects were buried vertically and GPR scans were taken on the biggest side wall.	44
3.3	CVAT annotation tool used for labelling hyperbola bounding boxes, masks and apex position.	45

3.4	Experimental setup used for the verification of the system. Objects were buried horizontally and GPR scans were taken along the soil's surface considering a height offset	46
3.5	GPR processing applied sequentially to a collected B-scan. a) Original, b) Dewow, c) Background removal, d) Band-pass filter, e) AGC	47
3.6	GPR reference pulse. Obtained from a hyperbola of Object 1, where the probing medium was air.	48
3.7	Computer vision techniques applied sequentially to the B-scan shown in figure 3.5.	49
3.8	Images resulting from the augmentation pipeline, applied to the B-scan shown in figure 3.5 after GPR processing and AGC.	51
3.9	Multi-channel encoding of configurations (MCEC), where different B-scan processing techniques are combined, applied to the B-scan shown in figure 3.5.	53
3.10	YOLO hyperbola mask predictions from trained model implemented in the system. For visualization purposes, B-scan is shown after applying GPR-processing techniques and AGC. Multiple hyperbolic reflections exist due to the ringing effect.	55
3.11	Number of YOLO hyperbola mask predictions predicted from 100 augmented images, applied to the B-scan shown in figure 3.10.	56
3.12	Hyperbola skeleton and resulting fitting points, applied to the B-scan shown in figure 3.10.	58
3.13	Constrained least-squares fitting of GPR hyperbola model, applied to the B-scan shown in figure 3.10. Fitting points are shown in red, fitted points are shown in green and the hyperbola apex is shown in blue.	59
3.14	Results from algorithm 2, applied to the B-scan shown in figure 3.10. First-hyperbola is shown in green and echos in blue.	61
4.1	Training and validation results of YOLOv8-seg medium model on multi-channel encoding configuration 6.	65
4.2	Validation metric curves of YOLOv8-seg medium model on multi-channel encoding configuration 6.	66
4.3	YOLO hyperbola mask predictions from trained model implemented in the system of scans from verification dataset. For visualization purposes, B-scans are shown after applying GPR-processing techniques and AGC.	72
4.4	Hyperbola skeleton, applied to the B-scans shown in figure 4.4.	73
4.5	Constrained least-squares fitting of GPR hyperbola model, applied to the B-scans shown in figure 4.4. Fitting points are shown in red, fitted points are shown in green and the hyperbola apex is shown in blue.	74
4.6	Results from algorithm 2, applied to the B-scans shown in figure 4.4. First-hyperbola is shown in green and echos in blue.	75
4.7	Number of YOLO hyperbola mask predictions considering the test-time augmentation approach, applied to scan configurations where false negatives existed, as listed in table 4.6. (1/2)	77
4.8	Number of YOLO hyperbola mask predictions predicted from 100 augmented images, applied to scan configurations where false negatives existed, as listed in table 4.6. (2/2)	78
A.1	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 1.	91
A.2	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 1.	92

A.3	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 2.	92
A.4	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 2.	93
A.5	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 3.	93
A.6	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 3.	94
A.7	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 4.	94
A.8	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 4.	95
A.9	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 5.	95
A.10	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 5.	96
A.11	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 6.	96
A.12	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 6.	97
A.13	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 7.	97
A.14	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 7.	98
A.15	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 8.	98
A.16	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 8.	99
A.17	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 9.	99
A.18	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 9.	100
A.19	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 10.	100
A.20	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 10.	101
A.21	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 11.	101
A.22	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 11.	102
A.23	Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 12.	102
A.24	Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 12.	103

List of Tables

2.1	Typical relative dielectric permittivity, ϵ_r , electrical conductivity, σ , velocity, v , and attenuation, α , observed in common geologic materials. [6]	9
2.2	Mathematical expression of activation functions. [34]	33
3.1	OERAD Concretto technical specifications. [51]	42
3.2	Radius of buried objects used.	44
3.3	Multi-channel encoding configurations (MCEC) of different B-scan processing techniques.	52
3.4	Training hyperparameter used in train the YOLOv8-seg models.	54
4.1	Validation metrics of YOLOv8-seg nano models on multi-channel encoding configurations (MCEC).	64
4.2	Testing metrics of YOLOv8-seg nano models on multi-channel encoding configurations (MCEC).	64
4.3	Validation and testing metrics of YOLOv8-seg medium model on multi-channel encoding configuration 6.	66
4.4	Ground truth values of verification dataset for object's position, x_{GT} , depth, y_{GT} , and radius, r_{GT} .	67
4.5	Estimated parameters for true-positive hyperbola detections. The number of true-positive, TP , false positive, FP and false negative, FN detections is shown and if the first-hyperbola was detected correctly. Medium's EM wave propagation velocity, v and object's position, x , depth, y and radius, r . Calculated considering only first-hyperbola, $1st$, and all hyperbolas from an object, $group$.	68
4.6	Estimated parameters for true-positive hyperbola detections, when test-time augmentation was used. The number of true-positive, TP , false positive, FP and false negative, FN detections is shown and if the first-hyperbola was detected correctly. Medium's EM wave propagation velocity, v and object's position, x , depth, y and radius, r . Calculated considering only first-hyperbola, $1st$, and all hyperbolas from an object, $group$.	69
4.7	Absolute error, Δ , of valid estimated parameters shown in table 4.5.	70
4.8	Absolute error, Δ , of valid estimated parameters shown in table 4.6, when test-time augmentation was used.	71

Abbreviations and Symbols

AI	Artificial Intelligence
AGC	Automatic Gain Control
CIoU	Complete Intersection Over Unit
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CVAT	Computer Vision Annotation Tool
C2f	Cross-Stage Partial Bottleneck With Two Convolutions Faster
DC	Direct Current
DFL	Distribution Focal Loss
FN	False Negative
FP	False Positive
ELU	Exponential Linear Unit
EM	Electromagnetic
GAN	Generative Adversarial Network
GPR	Ground Penetrating Radar
IoU	Intersection Over Unit
IPOPT	Interior Point Optimizer
mAP50	Mean Average Precision at IoU of 0.50
mAP50-95	Mean Average Precision with IoU range from 0.50 to 0.95
MCEC	Multi-Channel Encoding Configuration
NCC	Normalized Cross-Correlation
NN	Neural Networks
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RGB	Red-Green-Blue
SGD	Stochastic Gradient Descent
SPPF	Spacial Pyramid Pooling Faster
SSD	Single-Shot Detectors
TP	True Positive
XML	Extensive Markup Language
YOLO	You Only Look Once
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional

Chapter 1

Introduction

1.1 Context and Motivation

The characterization of a subsurface is an intrinsic problem in many fields of science and engineering, essentially everywhere there is a need to determine what is beneath a visually opaque surface. Naturally, non-destructive and non-intrusive techniques to solve this problem are of great importance where there is a need to preserve the probing environment.

In this context, Ground Penetrating Radar (GPR) has emerged as one of the most useful subsurface probing methods [2]–[5]. The main advantages of this method are the non-destructive and low-time-consuming nature of the scan, allowing for a more efficient, cost-effective and environmentally sustainable survey. In essence, GPR is able to characterize the electromagnetic response of a subsurface, allowing for the detection of buried objects. Object detection and characterization are possible by identifying and analysing the distinct signature of an object in GPR data. GPR has a wide range of applications, spanning fields such as archaeology, geology and civil engineering. Some examples include the detection and mapping of buried pipes, utilities, tree roots and even landmines, infrastructure inspection and geological surveying [5].

Traditionally, GPR data requires human interpretation. However, analysing GPR data is a time-intensive and difficult operation that requires relevant knowledge of the operating principles of GPR, the propagation of electromagnetic waves, and their interactions with the subsurface and buried objects. Consequently, this process is time-consuming and heavily subject to the operator’s empirical interpretation of the GPR data, which may be inaccurate.

In the scope of the introduction of automated robotic systems for agricultural and forest applications, the GPR technology is interesting for several operations, including soil mapping and tree plantation. Essentially, GPR could be used to characterize the soil in order to determine the optimal tree plantation site, by ensuring the absence of buried objects which could interfere with the healthy development of the tree’s root system.

In this context, this dissertation aims to develop an automated system to characterize the presence of buried objects in a subsurface through the analysis of GPR data.

1.2 Proposed Solution and Objectives

The dissertation's objective is the development of a system to characterize the subsurface's features available in the GPR scan, namely, determining the position, depth and radius of buried objects. In order to achieve this several GPR data processing and computer vision techniques were explored, in an approach with a main focus on the utilization of neural networks.

The envisioned system is composed of a data pre-processing step, an object identification step and, finally, a characterization step. In summary, the objective of the first step is processing the data as to make it easier for the hyperbolic signature to be detected. Next, the pre-processed data are inputted to a neural network, with the goal of identifying the characteristic object signature from data where cluttering signals may exist. A multi-channel encoding strategy was considered when training the networks, where complementing processing techniques were stored in each of the image RGB channels. Finally, the detected signature will be analysed with respect to the mathematical model of the GPR hyperbola in order to determine the object's characteristics, including its position, depth and radius, and the propagation velocity of EM waves in the medium.

Summarily, the objectives proposed for this dissertation are:

- Study of the fundamental principles of GPR operation.
- Study of the state of the art GPR data analysis techniques, including traditional GPR processing, computer vision and deep learning methodologies.
- Collection of GPR scan's dataset of a subsurface with buried objects.
- Development of an automated pipeline for the analysis of GPR data, detection and characterization of buried objects.
- Evaluation and characterization of the performance of the developed system.

1.3 Contributions

This dissertation provides the following contributions in the field of GPR data analysis:

- Extensive literature review on:
 - Fundamental principles of GPR operation.
 - Parameter estimation considering GPR hyperbola models.
 - Traditional GPR and computer vision data processing techniques.
 - Neural networks applied to GPR data analysis.
 - Data augmentation techniques.
- Collection and augmentation of GPR datasets.

- Training, testing and analysis of YOLOv8 segmentation models used to detect GPR hyperbolas, considering a multi-channel encoding strategy.
- Implementation and evaluation of an automated pipeline for GPR data analysis.

The dataset gathered and code developed during this dissertation are available in the following page: <https://gitfront.io/r/db01/zKNCvJYD5CfG/DISSERTATION/>

1.4 Document Structure

This document is divided into four more chapters: Background and State of the Art, Automatic Object Detection in GPR Data, Results and Discussion, and Conclusions.

In the next chapter, Chapter 2, the main theoretical aspects and fundamental knowledge relevant to understanding and contextualizing the problem and its possible solutions are explained. It starts by exposing the theoretical concepts associated with electromagnetism and framing them in the context of GPR operation. Following this, the state of the art techniques for GPR data processing and methodologies for object detection are exposed. The chapter concludes with a discussion about the presented literature with the goal of identifying possible solutions for the problem. Next, Chapter 3 thoroughly explains the developed system, exposing in detail the work developed. In Chapter 4, the results obtained are analysed and discussed. Finally, in Chapter 5, the document ends with the conclusions obtained, assessing the fulfilment of the defined objectives, the viability of the proposed system and proposing future improvements.

Chapter 2

Background and State of the Art

2.1 Fundamentals of GPR Operation

This section starts by introducing the theoretical principles necessary to understand GPR systems. Firstly, a brief overview contextualizes the GPR technology, followed by a summary of the most relevant concepts of electromagnetic theory, focused on the basic principles of wave propagation. Next, more practical aspects associated with GPR's working principles are explored.

2.1.1 Overview

The working principles of GPR operation are similar to those of conventional radar but applied to a different probing medium, the subsurface. The most common configuration of GPR systems consists of a transmitter and receiver antenna in a fixed geometry, which is moved over the surface in order to probe it [4]. Essentially, GPR works by transmitting electromagnetic waves into the subsurface, which are then scattered and/or reflected by changes in the medium's electromagnetic properties which are caused by dielectric discontinuities. The reflected waves are picked up by the receiver antenna, allowing for the characterization of the medium and the buried objects' depth and geometry. [6]. Depth is determined by the time interval between the transmission and reception of the electromagnetic (EM) waves and their propagation velocity, which depends on the medium's EM properties. In simple terms, since reflection depends on the object's electrical and magnetic properties, this means that GPR can detect any material provided these properties are sufficiently different from its surrounding materials [3].

2.1.2 Electromagnetism

The physical principles utilized by GPR systems are described by electromagnetic theory. In fact, GPR theory represents a subset of the whole electromagnetic field, focused mainly on the propagation of electromagnetic waves and their interactions with materials with different properties [4], [6], [7]. These concepts are described in the following subsections.

2.1.2.1 Maxwell's Equations

Maxwell's equations mathematically summarize the physics of EM fields and their relationships with each other [4], [6].

$$\bar{\nabla} \times \bar{E} = -\frac{\partial \bar{B}}{\partial t} \quad (2.1)$$

$$\bar{\nabla} \times \bar{H} = \bar{J} + \frac{\partial \bar{D}}{\partial t} \quad (2.2)$$

$$\bar{\nabla} \bullet \bar{D} = q \quad (2.3)$$

$$\bar{\nabla} \bullet \bar{B} = 0 \quad (2.4)$$

- E - electric field strength vector
- B - magnetic flux density vector
- H - magnetic field intensity
- J - electric current density vector
- D - electric displacement vector
- q - electric charge density

Equation 2.1, Faraday's law, describes the relationship between an electric field and the rate of change in a time varying magnetic flux. Equation 2.2, Ampere's law, describes the generation of a magnetic field around a closed loop to the flow of an electric current through the loop. Equation 2.3, Gauss's law for electric fields, states that the total electric flux leaving is proportional to the charge enclosed by the surface. Equation 2.4, Gauss's law for magnetic fields, states that the total magnetic flux leaving a closed surface is equal to zero [6], [7].

2.1.2.2 Constitutive Equations

Constitutive relationships quantify the physical properties of materials by describing the macroscopic behaviour of electrons, atoms and molecules of the material when responding to an EM field [4], [6], [7]. These relationships are expressed by the constitutive equations, expressed in equations 2.5-2.7.

$$\bar{J} = \sigma \bar{E} \quad (2.5)$$

$$\bar{D} = \epsilon \bar{E} \quad (2.6)$$

$$\bar{B} = \mu \bar{H} \quad (2.7)$$

Consequently, it's possible to characterize the material according to its electromagnetic properties, namely, electrical conductivity, σ , dielectric permittivity, ϵ , and magnetic permeability, μ .

Electrical conductivity characterizes the ease of free charge movement, that creates an electric current, when an electric field is present, as described by equation 2.5 [4], [6], [7].

Dielectric permittivity relates the displacement of charge by the material to the presence of an electric field, as described by equation 2.6. The charge displacement is related to the material's capacity to store energy. In the scope of GPR, the relative dielectric permittivity is often used, which simply refers to the dielectric permittivity relative to the permittivity of vacuum, expressed by the following equation [4], [6], [7].

$$\epsilon_r = \frac{\epsilon}{\epsilon_0} \quad (2.8)$$

Magnetic permeability describes how intrinsic atomic and molecular magnetic moments respond to a magnetic field. For simple materials, distorting intrinsic magnetic moments stores energy in the material [4], [6], [7].

Electrical properties, namely electrical conductivity and dielectric permittivity, tend to be the dominant factor controlling GPR applications. Variations in magnetic permeability are usually weak, making less of an impact on GPR applications. GPR operation is favourable in low-electrical-loss materials, where electrical conductivity is high, making the signal able to penetrate increased depths [4], [6].

2.1.3 Electromagnetic Wave Properties

Ground penetrating radar probing works by transmitting EM fields into the subsurface, which propagate as waves in suitable conditions for GPR operation. The wave nature of EM fields becomes apparent when Maxwell's equations are rewritten as equation 2.9 [4].

$$\bar{\nabla} \times \bar{\nabla} \times \bar{E} + \mu \sigma \cdot \frac{\partial \bar{E}}{\partial t} + \mu \epsilon \cdot \frac{\partial^2 \bar{E}}{\partial t^2} = 0 \quad (2.9)$$

Equation 2.9 can be rewritten further in terms of sinusoidal excitation [4].

$$f(\beta, t) = A e^{-i(\frac{\beta}{v} - \omega t)} e^{-\alpha \beta} \quad (2.10)$$

In equation 2.10, A represents peak signal amplitude, while v , α , β and ω represent the velocity, attenuation, travelled distance and angular frequency of an EM wave propagating in a uniform medium [4]. In figure 2.1, the propagation of an EM wave in an attenuating medium can be visualized, in relation with the parameters of equation 2.10.

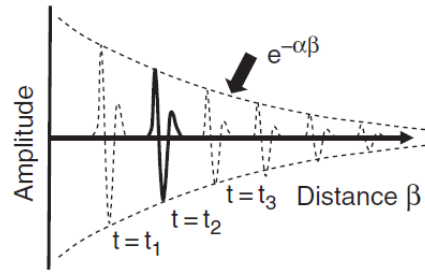


Figure 2.1: Propagation of EM wave [4]

2.1.3.1 Velocity and Attenuation

In the context of GPR applications, velocity and attenuation are some of the most important EM wave properties when analysing GPR data. Velocity describes how fast the EM travels in the medium, making an accurate velocity estimate essential when determining a feature's depth. The attenuation factor relates the attenuation of the EM wave's amplitude with propagation time. Hence, this factor influences the interpretability of the GPR signal at higher depths, since lower amplitude signals are harder to differentiate from noise. These properties are related to the medium's physical properties as described by the following equations [4], [6].

$$v = \frac{1}{\sqrt{\epsilon\mu}} = \frac{c}{\sqrt{\epsilon_r}} \quad (2.11)$$

$$\alpha = \frac{1}{2}\sigma\sqrt{\frac{\mu}{\epsilon}} \quad (2.12)$$

As described by equation 2.12, attenuation is greatly increased in materials with high electrical conductivity. In particular, the water content of an object heavily affects electrical conductivity, which depends on the water's salinity. As water is invariably present in the pore space of natural materials, its presence heavily impacts attenuation [4], [6]. In Table 2.1, the typical velocity and attenuation of common geologic materials are listed.

The velocity and attenuation properties of an EM wave propagating in a simple medium, with fixed electrical conductivity, dielectric permittivity and magnetic permeability, can be expressed with relation to a sinusoidal frequency, as seen in figure 2.2. At low frequencies, both velocity and attenuation are heavily dependent on frequency. The high frequency behaviour is more relevant to GPR applications, where these properties are frequency independent. In these conditions, velocity and attenuation can be described by the following equations [4], [6].

For a simple material, the transition frequency between low and high frequencies is defined by equation 2.13.

$$f_t = \frac{\sigma}{2\pi\epsilon} \quad (2.13)$$

Table 2.1: Typical relative dielectric permittivity, ϵ_r , electrical conductivity, σ , velocity, v , and attenuation, α , observed in common geologic materials. [6]

Material	ϵ_r	σ (mS/m)	v (m/s)	α (dB/m)
Air	1	0	0.30	0
Distilled Water	80	0.01	0.033	0.002
Fresh Water	80	0.5	0.033	0.1
Sea Water	80	300	0.01	103
Dry Sand	3 - 5	0.01	0.15	0.01
Saturated Sand	20 - 30	0.1 - 1.0	0.06	0.03 - 0.3
Limestone	4 - 8	0.5 - 2	0.12	0.4 - 1
Shales	5 - 15	1 - 100	0.09	1 - 100
Silts	5 - 30	1 - 100	0.07	1 - 100
Clays	5 - 40	2 - 1000	0.06	1 - 300
Granite	4 - 6	0.01 - 1	0.13	0.01 - 1
Dry Salt	5 - 6	0.01 - 1	0.13	0.01 - 1
Ice	3 - 4	0.01	0.16	0.01

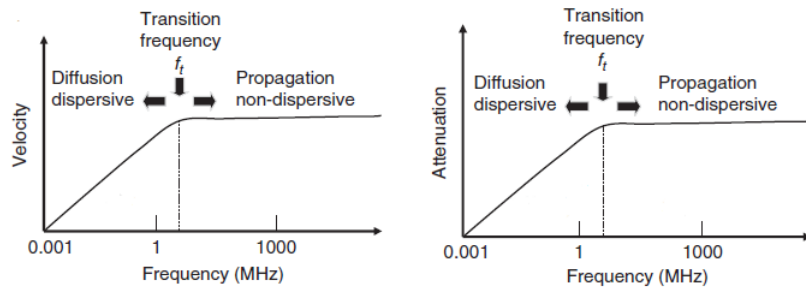


Figure 2.2: Relation between frequency and EM wave velocity and attenuation. [4]

In reality, in more complex materials, both velocity and attenuation are not constant at higher frequencies. This is primarily due to the effect of water content in the material, which absorbs more energy at higher frequencies, and due to scattering losses which also increase at higher frequencies [4], [6].

2.1.3.2 Reflection and Refraction

When a propagating EM wave hits a boundary between two different mediums, namely medium 1 and medium 2, it is partially reflected and partially refracted, as illustrated in figure 2.3. The reflected wave continues travelling across medium 1 with the same incidence angle, while the refracted wave is transmitted into medium 2, changing its direction. [6]

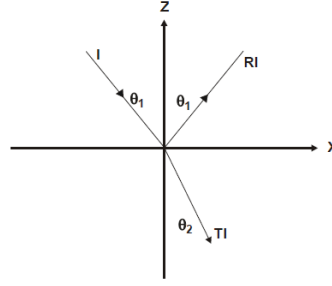


Figure 2.3: Reflection and refraction of an EM wave when hitting the boundary of different mediums. [6]

The direction change of the refracted wave when travelling across mediums with different propagation velocities and low loss is expressed by Snell's law [6].

$$\frac{\sin \theta_1}{v_1} = \frac{\sin \theta_2}{v_2} \quad (2.14)$$

Snell's law requires that the horizontal component of the propagation vector in each material to be equal, in order for the fields to match at the interface between the mediums. When $v_1 > v_2$, there is a range of angles θ_2 where waves can't travel from medium 1 to medium 2. Consequently, the critical angle defines the maximum angle of θ_2 after which refraction is impossible, obtained by setting θ_1 to 90° in equation 2.14 [4], [6].

2.1.4 Hardware

In order to further understand how GPR systems operate it is important to analyse its main components and how they are related. GPR systems are composed of at least one transmitting antenna and one receiving antenna, a transmitter module, a receiver module, a timing unit, a display unit and a storage unit, as seen in figure 2.4. All components are usually set up in a wheeled platform where the antennas are separated by a defined distance, denominated as common-offset configuration [4], [6]. Energy is normally supplied by rechargeable batteries [2]. Additionally, these systems may include a distance measurement device, such as an odometer or encoder, and/or positioning system, such as GPS or total station, allowing to geo-reference a scan [2], [3].

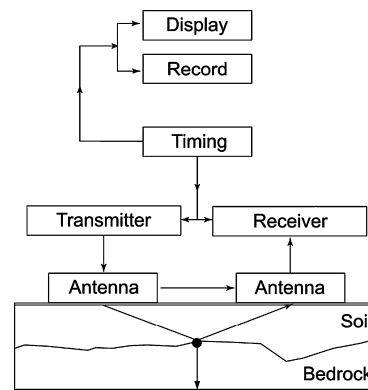


Figure 2.4: Block diagram of GPR's main components. [4]

The timing unit is responsible for the generation of signals which are then passed into the transmitter module and transmitted into the subsurface through the transmitting antenna. The reflected signals are then picked up by the transmitter antenna and module and processed by the timing unit before being stored or displayed [3], [4].

There are essentially two types of GPR systems, namely, pulsed and stepped-frequency. The most common one is pulsed GPR which works by transmitting a time domain pulse and analysing the reflected signal as a function of time. Stepped-frequency systems work by decomposing the EM pulse into its spectral components and transmitting the corresponding sinusoidal signals sequentially. In practical terms, stepped-frequency systems are generally claimed to be better performing, however, they are more complex and higher cost. Regardless, whatever the type of GPR, the signal can be converted to and analysed in either the time or frequency domains, simply by applying the Fourier or inverse Fourier transforms to the signal [2], [4].

2.1.4.1 Antenna

An antenna is a device used to transmit and/or receive energy as a radio frequency to/from a transmitting medium. In the case of GPR systems, transmission and reception is achieved by different antennas. The transmission antenna translates an excitation voltage into an EM field. The receiving antenna detects the temporal variation of the EM field and translates it into a recordable electrical signal. Additionally, efficient field generation and detection requires that the size of antennas must be similar to the wavelength of signals, such that the field travel time across the antenna dimension is comparable to the temporal rate of change of the exciting voltage field. Consequently, the physical size restrictions of GPR systems influence antenna design [4], [6].

Antennas can be defined as omni-directional or directional, according to the directions in which they radiate energy. Omni-directional antennas radiate energy in all directions simultaneously. GPR antennas can be characterized as directional antennas. These radiate energy in patterns of lobes that extend outwards from the antenna mainly in one direction, consisting of the main lobe, and also sideways and backwards, in side lobes and back lobes. Lobe patterns can be characterized by gain, direction angle and beam width, as visualized in figure 2.5, [4], [6].

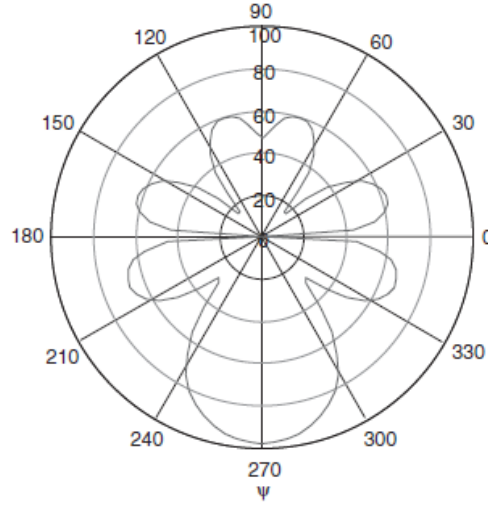


Figure 2.5: Antenna lobe pattern. Lobe angle Ψ in degrees, signal level in dB. [4]

Directional antennas can be characterized by their directivity, that is, the degree of sharpness of the main lobe. An antenna with higher directivity has a narrower main lobe which results in less power needed to cover the same distance when compared to an antenna with lower directivity since power is concentrated. Additionally, antenna directivity is affected by EM characteristics of the subsurface and the elevation of the antenna from the surface [4], [6].

Due to the effect of directivity on the transmitted EM waves, these propagate as ellipses with increasing areas as they propagate deeper, resulting in a beam with an elliptical cone shape. This phenomenon is described by equation 2.15 and figure 2.6, where $E/2$ represents the energy radius and H the propagation distance [8], [9].

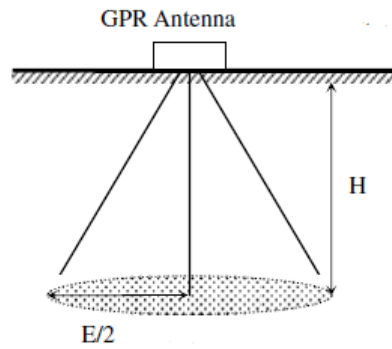


Figure 2.6: Propagation shape of EM waves. [8]

$$\frac{E}{2} = \frac{\lambda}{4} + \frac{H}{\sqrt{\epsilon_r + 1}} \quad (2.15)$$

In the context of GPR applications, the transmitted and received signals can be characterized by their path. A direct signal travels directly from the transmitting antenna to the receiver, due

to the existence of antenna side lobes. The signal transmitted into the subsurface is refracted and reflected at the boundary between different mediums, in accordance with Snell's law. Notably, the first reflection occurs at the interface between the air and the surface. Additionally, interference signals originating from the environment can also be picked up. All of these signals are summarised in figure 2.7.

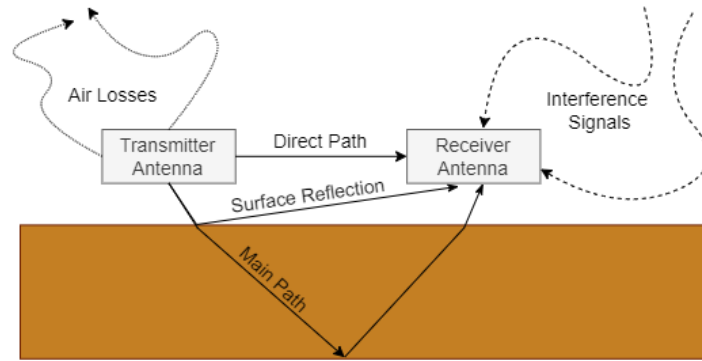


Figure 2.7: Propagation direction of EM signals in GPR applications. [10]

Antenna shielding is used in order to selectively enhance some signals and suppress others, according to their direction. The goal is to eliminate undesired signals, that is, those not coming from reflections from the probing medium. However, in GPR applications shielding can be impractical, since it requires a structure that has an EM response and effective shielding requires a larger antenna size. Altogether, the added complexity, size, weight and cost result in non-shielded configurations being common [4], [6].

Additionally, the rate of energy decay within the antennas is an important parameter in defining the self-clutter effect, known as ringing. Ringing is caused due to signals repeatedly bouncing within an object, usually metallic, or between multiple objects. Notably, ringing may occur inside the GPR's antennas and shielding. Every time the wave hits the object's boundary, according to Snell's law, part of the wave is refracted back to the enclosing material and another part is reflected back into the object. The continually reflected part of the signal resonates within the object until all energy is refracted or dissipated [4], [6], [11].

An object's ability to resonate depends on the signal's wavelength. The resonant frequency depends on the size of the object and the time it resonates depends on the permittivity contrast between the object and the surrounding material [11]. Consequently, instead of the antenna receiving a single impulse, multiple impulse repetitions are observed, which can obscure other signals [4], [6], [11].

In GPR applications, the antenna design is restricted by the characteristics of the propagation medium and portability concerns. Consequently, short electric dipoles or small magnetic dipoles have been preferred. The transmitter and receiver antennas are normally setup as to be separated by a fixed distance, known as common-offset configuration [4], [6], [7].

2.1.4.2 Central Frequency and Bandwidth

The operating frequency of GPR corresponds to the central frequency of its operating band. As previously mentioned, attenuation remains uniform above the transition frequency for simple materials but increases at high frequencies in complex materials. Consequently, the central frequency should be as low as possible while keeping the operating band above the transition frequency, in order to minimize attenuation and thus maximize penetration depth [4], [6].

Bandwidth represents the frequency spectrum of the operating band. In GPR systems, wider bandwidths are desired since these result in a better resolution of the radar. Typically, the goal is to achieve a bandwidth to central frequency ratio, expressed in equation 2.16 of about 1 [4], [6].

$$\frac{B}{f_c} = R \quad (2.16)$$

2.1.4.3 Resolution

Resolution determines how accurately an object's position can be determined and what information can be extracted about its shape. Resolution is related to how closely space in time two signals may be and still be recognized as separate events [6], as illustrated by 2.8.

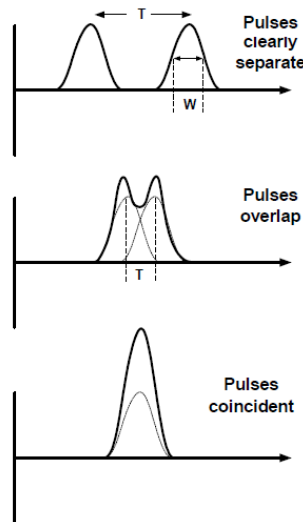


Figure 2.8: Interactions of closely trimmed pulses. [6]

In GPR applications, resolution can be divided into horizontal resolution and vertical resolution, which relate to the minimum lateral and vertical distances between equal objects in order to ensure the detection of both, as seen in figure 2.9. The minimum distance to separately distinguish two identical targets is widely considered to be half their pulse width, where pulse width corresponds to the width of half the peak amplitude [7]. In pulsed GPR, pulse width can be described by the following equations, as related to bandwidth, B , central frequency, f_c , and wavelength, W , by the following equations, keeping in mind the bandwidth to central frequency ratio, R .

$$f_c = \frac{v}{\lambda_c} \quad (2.17)$$

$$W = \frac{1}{B} = \frac{1}{Rf_c} = \frac{\lambda_c}{Rv} \quad (2.18)$$

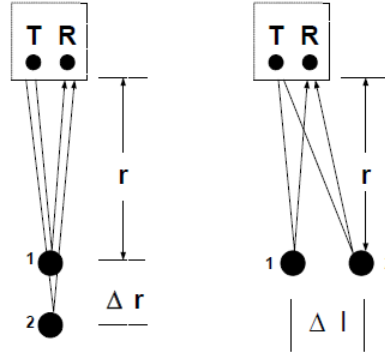


Figure 2.9: Vertical and lateral resolution geometrical parameters of objects closely positioned. [6]

Consequently, it's possible to determine the minimum horizontal and vertical distances by defining the minimum time interval between two reflected signals, according to their paths as illustrated in figure 2.9 [6]. For vertical resolution, this is obtained by subtracting the time of detection for object 1 and object 2, respectively, t_1 and t_2 .

$$t_1 = \frac{2r}{v} \quad (2.19)$$

$$t_2 = \frac{2r + 2\Delta r}{v} \quad (2.20)$$

$$\Delta t = t_2 - t_1 = \frac{2\Delta r}{v} \quad (2.21)$$

Since the time difference must be greater than half the pulse width, the minimum vertical distance can be obtained [7].

$$\frac{W}{2} \leq \Delta t \Leftrightarrow \Delta r \geq \frac{vW}{4} \Leftrightarrow \Delta r \geq \frac{v}{4B} \Leftrightarrow \Delta r \geq \frac{v}{4Rf_c} \Leftrightarrow \Delta r \geq \frac{\lambda_c}{4R} \quad (2.22)$$

Horizontal resolution can be obtained similarly, keeping in mind figure 2.9, noting the different time of detection for object 2 due to the diagonal path of the signal.

$$t_2 = \frac{2\sqrt{r^2 + \Delta l^2}}{v} \quad (2.23)$$

$$\Delta t = t_2 - t_1 = \frac{2(\sqrt{r^2 + \Delta l^2} - r)}{v} \approx \frac{\Delta l^2}{vr} \quad (2.24)$$

Like before, the time difference must be greater than half the pulse width, so the minimum horizontal distance can be obtained [7].

$$\Delta l \geq \sqrt{\frac{vrW}{2}} \Leftrightarrow \Delta l \geq \sqrt{\frac{vr}{2B}} \Leftrightarrow \Delta l \geq \sqrt{\frac{r\lambda_c}{2R}} \quad (2.25)$$

From equations 2.22 and 2.25 it's possible to understand the impact of the antenna's characteristics, namely central frequency and bandwidth, and propagation velocity on correct object detection.

Vertical resolution is essentially independent from depth while it affects horizontal resolution. Both are dependent to the antenna's characteristics and medium propagation velocity which may not be constant. Additionally, the radius of the smallest object that can be detected is defined as at least half the lateral resolution [6], which corresponds to approximately 10% of the wavelength in real conditions [3].

2.2 GPR Data Interpretation

As previously mentioned, GPR systems have been employed in a broad range of applications. In the context of agricultural and forest applications, the GPR technology is particularly interesting for the characterization of the soil's properties and determination of the optimal plantation site. Through the analysis of GPR data, it is possible to detect and characterize the presence of buried objects. Thus, it would be possible to identify objects which would impact a plant's development, as well as map a plant's root system. Additionally, the depth and thickness of different soil layers can be obtained, which is useful information when considering the development of root systems. Other characteristics about the soil can also be estimated when considering its electromagnetic properties. In particular, the soil's water content can be estimated, as related by the medium's relative permittivity and the EM wave velocity [5].

In the following sections, the different types of GPR scans and the mathematical models used for the estimation of the parameters related to the object's and medium's characteristics are explained.

2.2.1 Scans

The data produced during GPR scanning can be visualized in different scans: A-scan, B-scan and C-scan. These are displayed in figure 2.10.

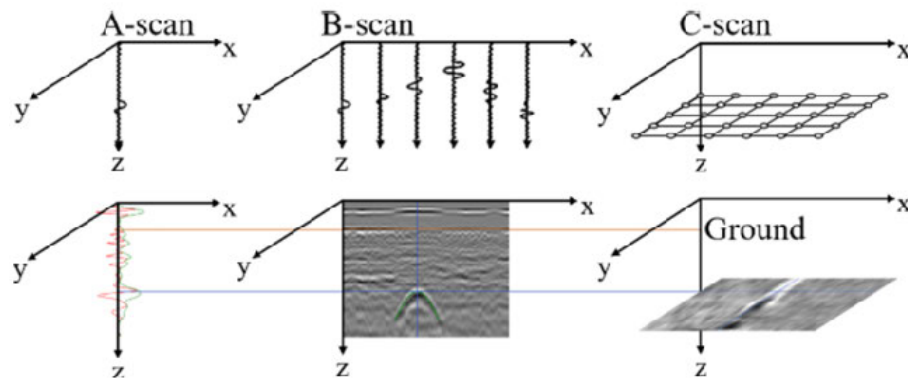


Figure 2.10: Relationship between A-scan, B-scan and C-scan. [12]

An A-scan is the representation of a single reflected EM wave on a time-amplitude graph, plotted along the z -axis. Note that the z -axis can also represent the depth since depth is proportional to the wave's travel time [13], [14].

The aggregation of consecutive A-scans measured along a fixed direction, in this case, the x -axis, results in a B-scan. The resulting B-scan can be interpreted as a two-dimensional (2D) image where the EM waves' amplitude is represented using brightness or colour modulation [13], [14].

Finally, a C-scan is obtained by associating multiple B-scans along a fixed constant depth, corresponding to a value of the z -axis. This creates a 2D image representing a slice of the subsurface along a specific depth. By combining several C-scans along several depths, the subsurface can be fully characterized along its three dimensions, limited only by the GPR's maximum penetration depth. In practice, C-scans are obtained by surveying the site and collecting B-scans along a defined grid [13], [14].

2.2.2 Hyperbola Models

As previously described, the EM waves transmitted by GPR propagate in an elliptical cone shape. This phenomenon results in the reflection, and subsequent recording, of EM waves before and after the GPR is directly above the object of interest. That is, the object will be detected before and after the GPR is directly above it [3], [10], [13].

When the GPR is directly above the object the EM waves will be reflected along the direct path, corresponding to the object's depth. Before and after this, the object will be detected via a longer path, resulting in a longer time interval between the transmission and recording of the EM wave [3], [10], [13].

If the object is cylindrical and perpendicular to the scan's direction, when plotted in a B-scan, this behaviour will result in a hyperbolic shape. The hyperbole's apex indicates the object's position and depth, since it's created by the EM wave reflected when GPR is directly above the object [2], [3], [10], [13].

In essence, this can be modelled mathematically by the geometrical parameters of the scan and the time an EM wave takes to be reflected.

2.2.2.1 Model 1

The creation of the hyperbolic signature in GPR can be modelled by the following model, which considers some simplifications. Namely, it considers a subsurface with a homogeneous medium and it neglects the distance between the antennas and the surface of the medium. Considering figure 2.11, the hyperbola can be modelled for every k th A-scan, by the following equations [13].

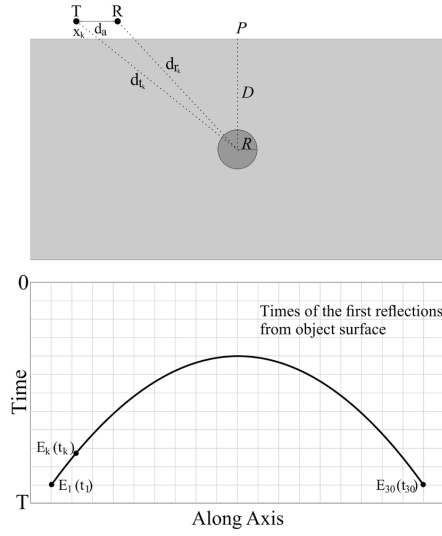


Figure 2.11: Hyperbolic signature of a cylindrical object for common-offset GPR and its relation to the scan's geometrical parameters. [13]

$$d_{t_k} = \sqrt{(x_k - P)^2 + D^2} \quad (2.26)$$

$$d_{r_k} = \sqrt{(x_k - d_a - P)^2 + D^2} \quad (2.27)$$

$$t_k = \frac{\sqrt{\epsilon_r}}{c} ((d_{t_k} + d_{r_k}) - 2R) \quad (2.28)$$

- d_a - offset distance between transmitter and receiver antennas
- d_{t_k} - distance between transmitter antenna and object
- d_{r_k} - distance between receiver antenna and object
- x_k - position of transmitting antenna at beginning of object detection
- P - position of buried object
- D - depth of buried object
- R - radius of buried object

Consequently, the object is detected in each k th A-scan at time t_k , forming a hyperbolic shape when plotted as a function of time and scanned distance. Note that scanned distance is translated by the k number of the A-scans, since these are taken while the GPR is being moved along the probing line.

Furthermore, the hyperbole's aperture is associated with the object's dimensions, as described by the relation between the object's radius and the time interval in equation 2.28. For example, a cylindrical object with a higher radius will create a hyperbole with a bigger aperture than an object with the same shape but a smaller radius. Additionally, an elongated flat object will result in the shape of a stretched hyperbola, more specifically, a continuous line with each end curving downwards [3], [10], [13].

2.2.2.2 Model 2

The hyperbola can also be modelled by the following model, which constitutes a further simplification of Model 1. Namely, the offset between the antennas is not considered. This simplification allows the representation of the hyperbolic shape through the mathematical equation of the hyperbola [15]–[17]. This model is illustrated in figure 2.12. The GPR, moving from left to right, starts detecting the object at position x and is directly above it at x_0 , such that the direct signal travel distance is given by Z and Z_0 , respectively. The hyperbola asymptotes and the hyperbolic shape are shown in dashed lines.

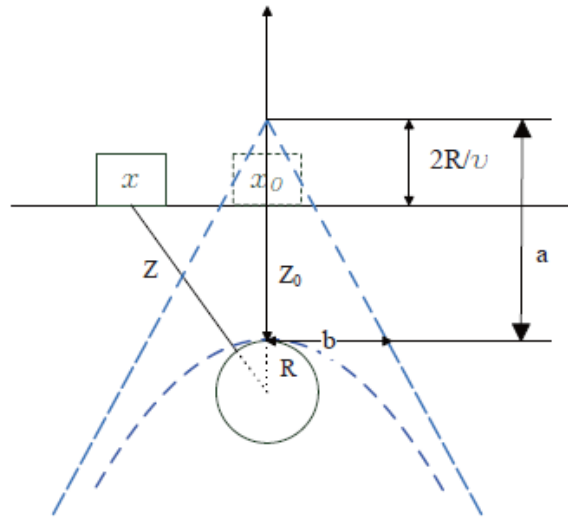


Figure 2.12: GPR hyperbola model of a cylindrical object, considering zero antenna offset, and the relationship with the scan's geometrical parameters. [16]

By disregarding the antenna offset, it's possible to determine the depth of a feature in a determined through equation 2.29, considering the time-stamp of an object in a B-scan and the medium's velocity, noting that t corresponds to the signal's round trip time and v its velocity in the medium.

$$d = v \frac{t}{2} \quad (2.29)$$

Considering figure 2.12, the object's position, given by x_0 , radius, R and the distance travelled by the EM wave to an object, when the object is first detected by the GPR and when the GPR is directly above the object, Z and Z_0 , respectively:

$$(x - x_0)^2 + (Z_0 + R)^2 = (Z + R)^2 \Leftrightarrow (x - x_0)^2 + \left(\frac{vt_0}{2} + R\right)^2 = \left(\frac{vt}{2} + R\right)^2 \quad (2.30)$$

It's possible to represent equation 2.30 as the mathematical hyperbola equation 2.32:

$$\frac{(t + 2R/v)^2}{(t_0 + 2R/v)^2} - \frac{(x - x_0)^2}{(vt_0/2 + R)^2} = 1 \quad (2.31)$$

$$\frac{(y - y_0)^2}{a^2} - \frac{(x - x_0)^2}{b^2} = 1 \quad (2.32)$$

$$a = t_0 + \frac{2R}{v} \Leftrightarrow t_0 = a - \frac{2R}{v} \quad (2.33)$$

$$b = v \frac{t_0}{2} + R \quad (2.34)$$

$$y_0 = \frac{2R}{v} \quad (2.35)$$

By considering equations 2.33 - 2.35 its possible to determine the object's radius and wave velocity:

$$v = \frac{2b}{a} \quad (2.36)$$

$$R = \frac{v(a - t_0)}{2} \Leftrightarrow R = \frac{b(-y_0)}{a} \quad (2.37)$$

Additionally, these parameters have an effect on the shape of the hyperbola. More specifically, the angle between the hyperbola's asymptotes, and hence the openness of the hyperbola, is related only to the velocity [15], [17].

$$\tan \alpha = \frac{b}{a} = \frac{v}{2} \quad (2.38)$$

The hyperbola focus, given by (x_0, y_0) , determines the position of the buried object while also relating its radius to the wave velocity through y_0 . The object's position corresponds to x_0 while the depth is determined by relating t_0 in equation 2.33.

Consequently, it follows that if the hyperbolic shape represented in a B-scan can be fitted with the parameters of the mathematical model of the hyperbola equation, namely a , b , y_0 and x_0 , both the medium's EM wave velocity and object's position, depth and radius can be determined.

This model may be simplified further by considering a point reflector, that is, an object with zero radius [15]. Thus the model equation, equation 2.30 is simplified to:

$$(x - x_0)^2 + Z_0^2 = Z^2 \Leftrightarrow (x - x_0)^2 + \left(\frac{vt_0}{2}\right)^2 = \left(\frac{vt}{2}\right)^2 \quad (2.39)$$

By identifying the hyperbola apex, (x_0, t_0) , and any hyperbola point, (x, t) , it's possible to obtain a direct, yet less accurate, estimate of EM wave velocity without the need for hyperbola fitting, by rewriting equation 2.39:

$$v = 2\sqrt{\frac{(x - x_0)^2}{t^2 - t_0^2}} \quad (2.40)$$

2.2.3 Hyperbola Fitting and Parameter Estimation

The hyperbola parameters in the model described in section 2.2.2.2 can be determined by solving a constrained Least-Squares minimization problem [15]–[19]. The goal consists of minimizing the sum of the squared errors, where the error function, $F(i)$, corresponds to the distance, d_i , between the i th fitting point, p_i , and its closest point in the fitted hyperbola, p_h , such that:

$$F(i) = d_i = |p_i - p_h| \quad (2.41)$$

$$\min \sum_{i=1}^n F(i)^2 \quad (2.42)$$

Considering the mathematical hyperbola equation, equation 2.32, its possible to define the error function as:

$$F(i) = \frac{(y - y_0)^2}{a^2} - \frac{(x - x_0)^2}{b^2} - 1 = 0 \quad (2.43)$$

In order to avoid numerical instabilities when applying minimization algorithms to the error function, due to division by zero, it's necessary to re-write equation 2.32:

$$\begin{aligned} \frac{(y - y_0)^2}{a^2} - \frac{(x - x_0)^2}{b^2} &= 1 \\ b^2(y - y_0)^2 - a^2(x - x_0)^2 &= a^2b^2 \\ b^2(y^2 - 2y_0y + y_0^2) - a^2(x^2 - 2x_0x + x_0^2) &= a^2b^2 \\ b^2y^2 - 2b^2y_0y + b^2y_0^2 - a^2x^2 + 2a^2x_0x - a^2x_0^2 &= a^2b^2 \\ -a^2x^2 + b^2y^2 + 2a^2x_0x - 2b^2y_0y + b^2y_0^2 - a^2x_0^2 - a^2b^2 &= 0 \\ -\frac{a^2}{b^2}x^2 + y^2 + 2\frac{a^2}{b^2}x_0x - 2y_0y + y_0^2 - a^2 - \frac{a^2}{b^2}x_0^2 &= 0 \end{aligned} \quad (2.44)$$

Thus, the problem function can be represented by the general conic equation [18]:

$$F(i) = a_c x^2 + b_c xy + c_c y^2 + d_c x + e_c y + f_c = 0 \quad (2.45)$$

$$\begin{cases} a_c = -\frac{a^2}{b^2} \\ b_c = 0 \\ c_c = 1 \\ d_c = 2\frac{a^2}{b^2}x_0 \\ e_c = -2y_0y \\ f_c = y_0^2 - a^2 - \frac{a^2}{b^2}x_0^2 \end{cases} \quad (2.46)$$

It's possible to determine the parameters of the GPR hyperbola model from the coefficients of the conic equation by solving the system of equations 2.46:

$$\begin{cases} a^2 = \left(\frac{e_c}{2}\right)^2 - f_c + \frac{d_c^2}{4a_c} \\ b^2 = -\frac{(-e_c/2)^2 - f_c - d_c}{a_c} \\ x_0 = \frac{d_c}{-2a_c} \\ y_0 = \frac{-e_c}{2} \end{cases} \quad (2.47)$$

Consequently, through equations 2.29, 2.36 and 2.37, it's possible to determine the object's position, depth and radius and the EM wave velocity in the medium.

Given the physical restrictions on the parameters associated with the scan, it's necessary to constrain the minimization problem. Consecutively, bearing in mind equations 2.36 and 2.37 and the system of equations 2.47, the conic coefficients are constrained such that, where c is the speed of light in vacuum:

$$\begin{cases} R > 0 \\ 0 < v < c \end{cases} \quad (2.48)$$

While the hyperbola fitting process described works well for well-shaped hyperbolas [20], [21], in real GPR data hyperbolas are often miss-shaped. In general, the formation of ill-shaped hyperbolas is due to:

- Presence of cluttering signals
- Interactions with neighbouring reflectors
- Antenna resolution
- Medium heterogeneity
- Intrinsic shape of the object not being a cylinder

- Object angle not being perpendicular to the scan's direction

Consequently, the parameters estimated will be affected by an error, since they will be estimated from noisy points. In particular, the effect of noise is heavily detrimental to an accurate radius estimation. The remaining parameters, namely, EM wave velocity and object depth and position can be sufficiently estimated even considering noise [21].

Considering this, the radius can be estimated in an alternative way considering the GPR beam's shape and energy radius [8], [9], previously described in section 2.1.4.1. Essentially, the object's radius can be estimated by relating the distance where a GPR signal reflected from an object is detected and the energy radius at the object's depth, as illustrated in figure 2.13.

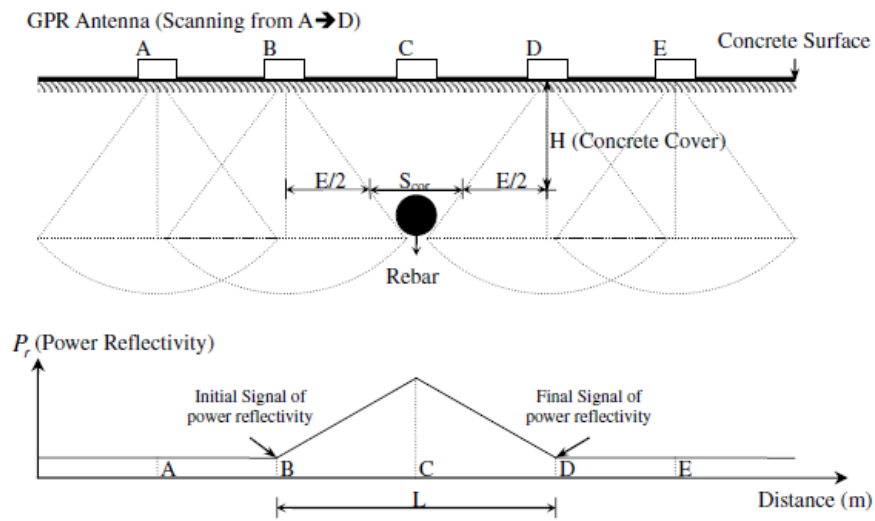


Figure 2.13: Effect of energy radius on power reflectivity. [8]

Before using this method it's first necessary to analyse the hyperbola, in order to determine the object's depth, H , as well as, the horizontal distance, L , where a reflection signal from the object can be determined by identifying the position of the first and last points which belong to a single hyperbola.

Noting equation 2.15 and considering figure 2.13, the object's radius, R , is obtained by:

$$S_{cor} = L - E \quad (2.49)$$

$$R = \frac{S_{cor}}{2\pi} \quad (2.50)$$

2.3 GPR Object Detection Methodologies

In recent years, various systems have been developed for the automatic interpretation of GPR scans with the goal of determining the subsurface's and object's characteristics. In order to automate this process several methodologies are used for the interpretation of B-scans. In general, this

includes a pre-processing step followed by computer vision techniques and/or neural networks (NN) frameworks for hyperbola detection and fitting. In the following sections, the state of the art techniques for object detection through the analysis of GPR data are described.

2.3.1 Pre-Processing

The pre-processing step is used in order to eliminate unwanted effects due to physical phenomena associated with the nature of the scanned environment and the propagation of EM waves. Usually, these corrections are applied directly to the A-scan. In the following section, the most relevant of these techniques are explained.

2.3.1.1 Stacking

Stacking consists of averaging multiple A-scans taken in the same position into a single A-scan, in order to remove random noise [7].

2.3.1.2 Time Zero Correction

In mobile GPR systems, there exists a gap between the antennas and the subsurface, due to the impossibility of keeping the antennas in constant contact with the ground, which would result in damage. Moreover, this gap may change in accordance with the irregularity of the surface. Consequently, there is an interval of time where the signal first travels through air before propagating through the subsurface [3].

If these delays aren't taken into account they will have an adverse effect on the accuracy of depth estimations, due to their effect on travel time, which will not represent the effective time the signal propagates through the subsurface.

The time zero correction solves this problem by setting the effective zero reference of time to the moment the signal comes into contact with the subsurface. This is done by detecting the signal which corresponds to the reflection on the ground surface. If no cluttering signals exist, the ground surface can be identified in an A-scan by the first signal maximum, as seen in figure 2.14. However, due to the effect of cluttering signals, direct signal and ringing, identifying this signal may prove difficult. [2], [3], [7].

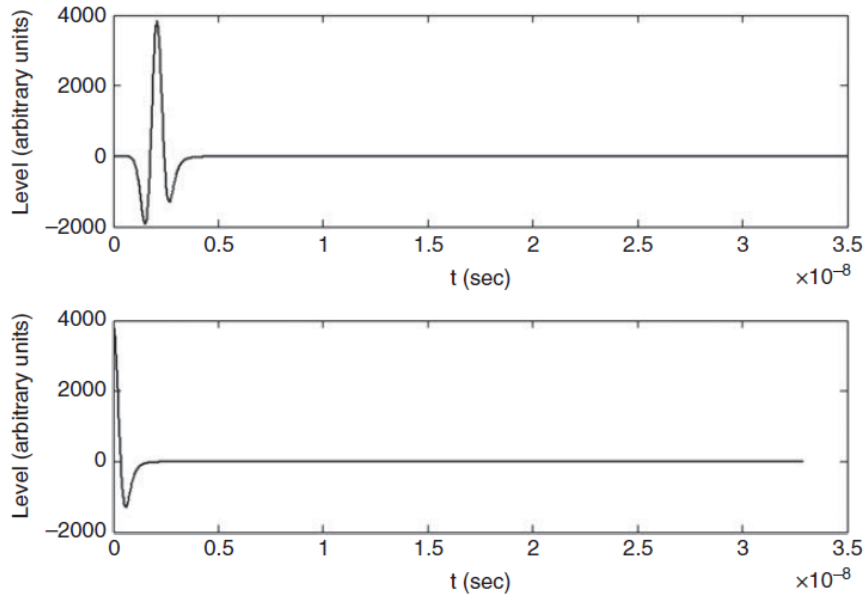


Figure 2.14: Time zero offset correction in a GPR A-scan. [2]

2.3.1.3 Dewow

The proximity between the transmitter and receiver antennas in GPR systems introduces a time-varying delay to the measured data. This is caused by a low-frequency wow effect associated with electrostatic and inductive fields near the transmitter, which causes the signal to have a shifted DC offset. This effect became known as baseline wow [4], [6].

Dewow consists of removing this effect by applying a high-frequency temporal filter to the measured signal, in order to make the baseline of the signal zero, as seen on figure 2.15. This can be achieved using a moving average filter along a defined window, as described by equation 2.51 [4], [6].

$$x_{dewow}[n] = \frac{1}{M} \sum_{k=0}^{M-1} x_{wow}[n-k] \quad (2.51)$$

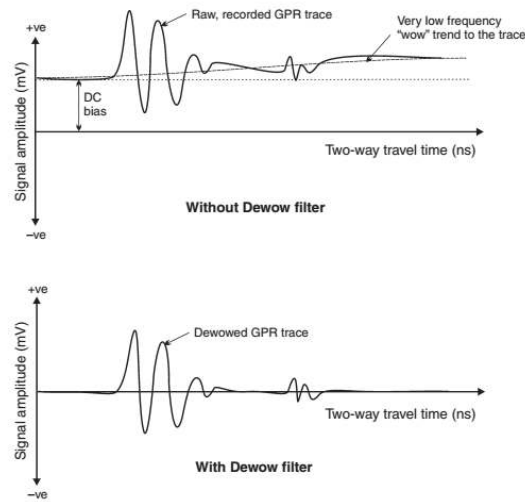


Figure 2.15: *Dewow* filter correction on a GPR A-scan. [4]

2.3.1.4 Frequency Filters

Frequency filters are used in order to filter the frequency spectrum of a signal, removing frequencies in a defined frequency range. These filters include low-pass, high-pass and band-pass, which remove the frequencies lower, higher or outside a specified range, respectively. In the context of GPR, band-pass filters are usually used in order to remove signals with frequencies outside the frequency range of the antennas. This makes it possible to remove both low and high-frequency environment noise from the captured data [3], [5].

2.3.1.5 Background Removal

Background removal is used with the goal of removing cluttering signals which are repeated over a large amount of consecutive A-scans. These signals correspond to reflections of features at constant depths throughout the environment. Namely, the surface reflection or those caused by boundaries between different materials. Additionally, these signals are also caused due to the continuous ringing effect in the antennas or inside the shielding [2], [3].

Since these cluttering signals are repeated over multiple A-scans, they appear as horizontal lines when plotted in a B-scan. Consequently, reflections caused by buried objects are often obscured.

Mean subtraction, illustrated in figure 2.16, is a common background removal technique. It consists in subtracting the average signal amplitude calculated along each time unit, for a defined number of consecutive A-scans.

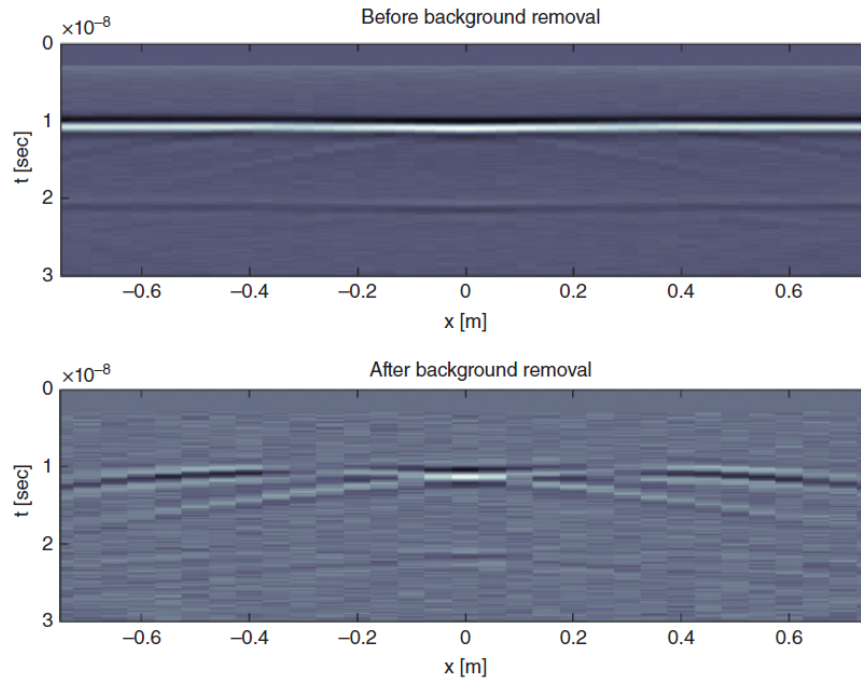


Figure 2.16: Effect of background removal in a GPR B-scan. [2]

Recently, other approaches based on subspace methods and deep-learning have been proposed. Subspace-based methods include singular value decomposition, principal component analysis and independent component analysis. In essence, these aim to project the original B-scan data onto clutter and target subspace, removing the clutter component [22]–[24]. Deep-learning methods will be explained in section 2.3.3.7.

2.3.1.6 Time Gain

As previously mentioned, EM waves suffer attenuation as they propagate through the subsurface, resulting in signals with decreasing amplitudes as they travel deeper. In order to mitigate this effect, gaining can be applied to the recorded A-scan. It consists of multiplying the signal by a gain factor. Since the more a signal travels the more it suffers attenuation, the gain factor can be calculated as a function of travel time, usually through a linear or an exponential function, as seen of figure 2.17 [4]–[6].

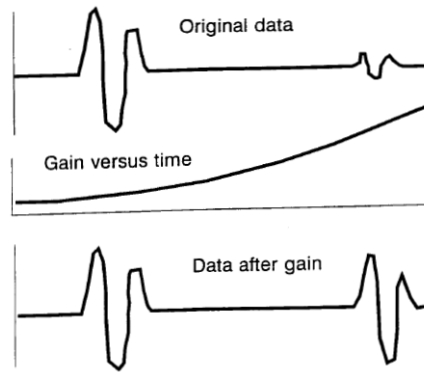


Figure 2.17: Effect of time varying gain applied to a GPR A-scan. [6]

However, determining the gain function can be difficult due to the non-linearity of attenuation in a medium [4], [6]. Additionally, gaining should only be applied to an A-scan after removing the DC offset and in low noise conditions, otherwise, noise will be increased [5].

2.3.1.7 Automatic Gain Control

Automatic gain control (AGC) equalizes the energy spectrum along a defined time window, making it have uniform power. The average signal in each window is computed and the data point at the center of the window is gained, or attenuated, such that the energy in each window is normalized. This type of gain is suitable for improving very attenuated signals, however, noise is also heavily amplified [6].

2.3.1.8 Migration

Migration algorithms aim to remove the effect of antenna directivity in B-scan, in order to reconstruct the geometrically correct radar reflectivity distribution of the subsurface. There are several types of migration algorithms but, in essence, these work by reversing the effect that results in the hyperbolic shape by taking into account the shape and velocity of the propagating waves. However, these algorithms require that these variables be accurately estimated and behave uniformly throughout the medium, which often does not happen in real conditions [2], [4]–[6].

2.3.2 Traditional Computer Vision

As previously mentioned, a B-scan can be interpreted as a grey-scale image, where brightness represents the reflected EM wave's amplitude. Following this, computer vision techniques can be applied to highlight and detect the features of interest. In the scope of GPR object detection, these are often used for hyperbola detection. The following subsections summarize the most relevant techniques used for feature detection in gray-scale images.

2.3.2.1 Thresholding

Thresholding is a step used to transform a gray-scale image into a binary image, by defining an appropriate threshold brightness value that separates the pixels of interest from the others. The resulting binary image has the pixels of interest set to the maximum brightness value and the others to zero. In the case of global thresholding, the threshold value is determined for the whole image. For local thresholding, the threshold value is calculated along the image by considering a region of pixels [25].

Furthermore, after thresholding, the binary image may be processed further by using techniques such as morphological operations, distance transforms and connected component analysis [26].

2.3.2.2 Edge Detectors

Edge detection algorithms are used to detect the edge of a feature in gray-scale images, making it possible to segment the feature of interest from the background. Mathematically, edges are defined by large variations in pixel brightness values. These can be characterized by the gradient of pixel brightness of the image. Before computing the gradient, a low-pass filter or other filters are usually applied by the algorithm with the goal of mitigating the effect of noise on the gradient. By analysing the gradient in neighboring pixels it's possible to determine which pixels belong to an edge. The gradient is computed by determining the first derivative by applying a convolution operation to the image, using kernels. The direction in which the derivative is taken determines the orientation of which edges are detected. If the derivative is taken across the x direction only vertical edges will be detected, and vice-versa. The size of the kernels is related to the number of points along which the gradient is calculated. There exist several edge detection kernels, of which Sobel stands out, due to its smoothing properties and edge localization [25]–[29]. The Sobel kernels are defined in equations 2.52 and 2.53.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.52)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.53)$$

2.3.2.3 Fitting

Fitting is a process that aims to represent the underlying structure of groups of related points by models that emphasize their characteristics. Generally, fitting involves determining which mathematical function best relates to the points by minimizing the error between the function and the fitting points [27].

Naturally, these techniques are very useful in computer vision since they allow to determine the existence of features of interest with a unique structure, such as lines, curves, circles or, in this case, hyperbolas.

2.3.2.4 Template Matching

Template matching is used with the goal of identifying the coherence/similarities between two data segments. Mathematically, this is done by calculating the cross-correlation between the data, I_0 and I_1 , as described by the convolution operation in equation 2.54. The maximum value of the correlation function effectively represents the similarity between the data [26], [27].

$$E_{CC}(u) = \sum_i I_0(x_i) I_1(x_i + u) \quad (2.54)$$

In order to mitigate the effect of signal amplitude when computing the convolution operation, normalized cross-correlation (NCC) is often used, as described by equations 2.55 - 2.57. The output of the normalized cross-correlation is in the range $[-1, 1]$, making analysing the similarity between signals, even with different amplitudes, easier. [27]

$$E_{NCC}(u) = \frac{\sum_i [I_0(x_i) - \bar{I}_0] [I_1(x_i + u) - \bar{I}_1]}{\sqrt{\sum_i [I_0(x_i) - \bar{I}_0]^2} \sqrt{\sum_i [I_1(x_i + u) - \bar{I}_1]^2}} \quad (2.55)$$

$$\bar{I}_0 = \frac{1}{N} \sum_i I_0(x_i) \quad (2.56)$$

$$\bar{I}_1 = \frac{1}{N} \sum_i I_1(x_i + u) \quad (2.57)$$

These operations may be employed to analyse the similarity between both 1D signals or 2D images. In the context of GPR data, correlation analysis has been used to obtain similarity maps of B-scans. For every data point in a B-scan, these maps represent the maximum value of the correlation operation, and thus similarity, between the data point and a reference signal [30], [31]. This may be done using a 1D reference signal, using a known GPR pulse, or with a 2D matrix, representing a known hyperbola shape.

2.3.3 AI Based Computer Vision

In recent years, neural networks have started to replace computer vision algorithms in many fields, including in GPR object detection. In essence, these work by training a model to infer conclusions about the inputted data, using a training dataset. The trained model is then tested against a different testing dataset in order to validate the capacity of the model to correctly infer conclusions about new data.

In the following subsections, the basic concepts associated with neural networks and convolutional neural networks (CNN) are exposed, followed by a description of the types of networks used for the analysis of GPR data.

2.3.3.1 Overview on Neural Networks

Neural networks are computational models inspired by the structure of the human brain, as they mimic the way that biological neurons signal each other [32]. They are formed by nodes that encode a linear regression model composed of input data, weights, bias, activation function and an output [26], [27], [32], [33]. The structure of a node is illustrated in figure 2.18.

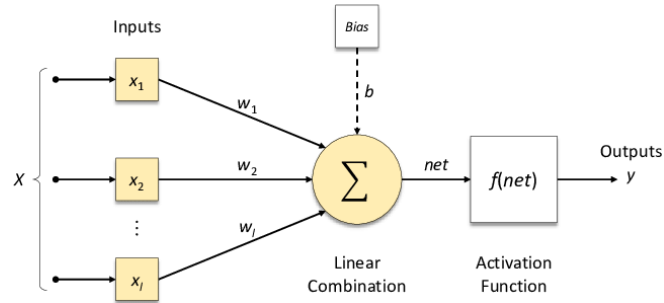


Figure 2.18: Structure of a neural network's node. [32]

The weights can assume positive or negative values reflecting the importance of the input [32]. The net value of the node is calculated by adding the weighted inputs and the bias. The net value is then inserted in the activation function which determines the output. Keeping in mind figure 2.18, the mathematical node model is defined by the following expression [26], [27], [32], [33].

$$y_{pred} = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (2.58)$$

The activation function determines the output value of the node by inserting it into a defined function, namely, ReLU, Sigmoid, Tanh, etc. These functions are used in order to introduce non-linearity into the network, allowing it to better understand complex relationships [26], [27].

A neural network is formed by sequentially connected nodes which are organized in layers, as seen in figure 2.19. If all the inputs of a layer are connected to the outputs of the previous layer, the layer is called fully connected [26], [27], [32], [33].

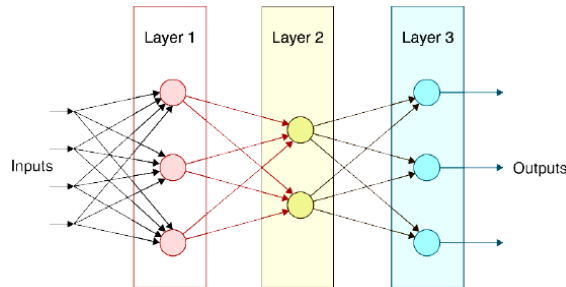


Figure 2.19: Organization of nodes into layers in a neural network. [26]

In order to train a neural network, training data is inputted and the output of the network is analysed. In essence, this process consists of adjusting the weights and bias of nodes with the goal

of minimizing a loss function by following a gradient descent approach. Several techniques may be used during this process to reduce over-fitting, that is, in order to increase the network's ability to generalize about the underlying patterns of the data. The trained network is then tested against a different testing dataset in order to validate the network's capacity to correctly infer conclusions about new unseen data [26], [27], [32], [33].

2.3.3.2 Datasets, Training and Evaluation

Neural networks are trained with the goal of understanding the underlying concepts of the trained classes, thus having the ability to generalize well to previously unseen data. Models with poor generalization ability are considered to be overfitted to the training data. Consequently, the goal of the training process is to obtain a model which performs accurately on the training dataset, while minimizing overfitting. To achieve this several techniques are employed [34]–[36].

- Datasets are usually organized into training, test, and sometimes also validation subsets. There exist several ways of organizing the data into these datasets, through K-Fold cross-validation techniques. In general, the goal is to split the data into different representative subsets, with the same distribution of classes. The model is trained using the training dataset and its performance is evaluated through the test and validation datasets [34], [36].
- Transfer Learning consists of training the network on a large pre-existing dataset and then using those weights as the initial weights in a new classification task. Normally, only the weights from convolutional layers are kept, rather than the complete network, which would include the fully connected layers. This technique is very effective since standard image datasets share low-level spatial characteristics [35], [36].
- Dropout is a regularization technique used to avoid overfitting. It consists in randomly setting neuron weights to zero based on a defined probability. This effectively forces the network to learn more robust features rather than relying on the predictive capacity of some neuron weights which may be overfitted to the training data [34], [35].
- Batch normalization is a regularization technique used to normalize the activations in a layer. This is done by subtracting the mean of the batch from each activation and dividing it by the batch's standard deviation. By ensuring that the inputs to each layer have a consistent distribution, it provides a regularizing effect that can reduce overfitting [35].

Training a neural network is done by using an optimizer to reach an optimal solution for a given loss function. During this process, the networks' parameters, namely weights and bias, are adjusted in order to achieve a minimum error for the training samples [34].

During training random samples of the training dataset are inputted into the the model, forming batches, and the output value is analyzed with respect to the expected result. The error is related by a loss function. Next, the network parameters are adjusted and the process is repeated until the entire training dataset has been read multiple times, corresponding to the number of epochs [34].

The activation functions are used when calculating a neuron's output in order to introduce non-linearity in the model and limit the vanishing or exploding gradients problems. A list of commonly used activation functions is available in table 2.2 and illustrated in figure 2.20 [34].

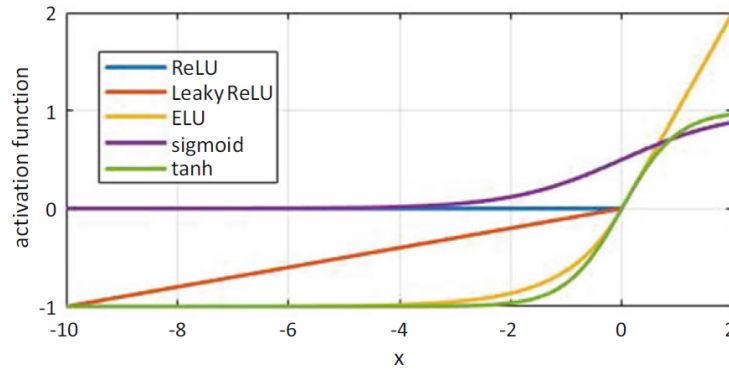


Figure 2.20: Activation functions. Leaky ReLU with a $\alpha = 0.1$ and ELU with $\alpha = 1$. [34]

Table 2.2: Mathematical expression of activation functions. [34]

Function	Formulation	Derivative	Output range
ReLU	$\begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases}$	$\begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}$	$[0, \infty)$
Leaky ReLU	$\begin{cases} \alpha x, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases}$	$\begin{cases} \alpha, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}$	$(-\infty, \infty)$
ELU	$\begin{cases} \alpha x(e^x - 1), & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases}$	$\begin{cases} \alpha x(e^x - 1), & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}$	$[-\alpha, \infty)$
Sigmoid	$\frac{1}{1+e^{-x}}$	$\frac{e^{-x}}{1+e^{-x}}$	$(0, 1)$
Tanh	$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$	$1 - \tanh^2(x)$	$(-1, 1)$
Soft-max	$\text{softmax}(v_i) = \frac{e^{v_i}}{\sum_{j=1}^N e^{v_j}}$	-	-

The backpropagation algorithm is used to relate the gradients of the error function with the network parameters. These gradients are used in order to adjust the network's parameters in a direction which minimizes the error. Neuron weights are adjusted recursively by calculating the error gradients for each layer, from the final layer to the input layer [34]. There exist several optimizers, but in general, they follow a gradient descent based approach. The networks's parameters are adjusted in the opposite direction of the error function's gradient, thus minimizing error. The learning rate is a parameter multiplied by the gradient during this process. Essentially, it is used to determine the step size taken to reach the local minimum [34].

The performance of a neural network may be evaluated using the ratios between the following metrics [34].

- True Positive (TP): the prediction correctly matches its ground-truth label.
- False Positive (FP): the prediction gives a positive match but no match should exist.

- False Negative (FN): the prediction gives a negative match but a match should exist.

Precision, P , described by equation 2.59, evaluates the correctness of the model's prediction with respect to the number of false positives. Recall, R , described by equation 2.60, measures the model's correctness with respect to the number of missed detections. F1-score, $F1$, described in equation 2.61, relates both precision and recall, thus being useful to evaluate the network with respect to both false positives and false negatives [34].

$$P = \frac{TP}{TP + FP} \quad (2.59)$$

$$R = \frac{TP}{TP + FN} \quad (2.60)$$

$$F1 = 2 \frac{P \cdot R}{P + R} \quad (2.61)$$

Intersection-over-Union (IoU) relates the ratio between the intersection area of the ground-truth and predicted bounding boxes or segmentation masks, as described by equation 2.62 [34].

$$IoU = \frac{Area(B_{ground-truth} \cap B_{predicted})}{Area(B_{ground-truth} \cup B_{predicted})} \quad (2.62)$$

Finally, several loss functions are used to describe the errors between predictions and ground-truth labels for each epoch during training. Other metrics like mAP50 and mAP50-95 describe the models' mean average precision for predictions at an IoU of 0.50 and from 0.50 to 0.95, respectively. Thus, it's possible to relate the model's accuracy for easier detections, using a lower IoU , and for more general cases, using broader IoU values [37].

2.3.3.3 Convolutional Neural Networks

Convolutional neural networks are a type of network architecture which have proven very useful in vision applications [27]. In principle, these work by reconstructing the image as a collection of local features which can then be combined in different ways in order to represent a more discriminative and semantically meaningful feature [26]. A CNN is composed of a feature extraction step followed by a classification step, as illustrated by figure 2.21.

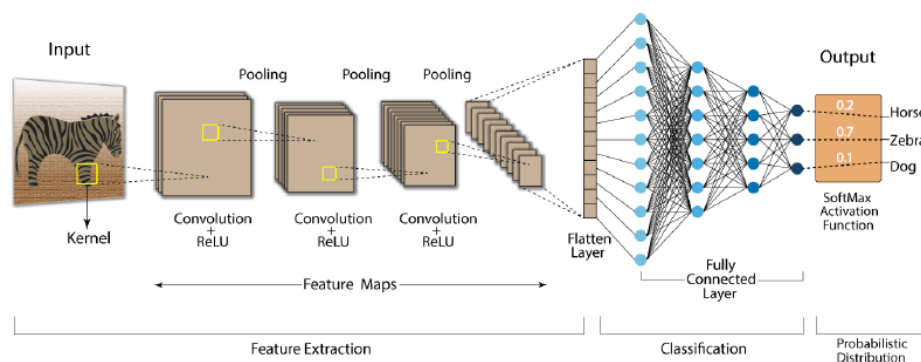


Figure 2.21: Architecture of a convolutional neural network. [38]

Features are extracted into feature maps through a series of convolutional and pooling layers. Each convolutional layer applies a set of filters to the input image in order to create a feature map that represents a specific aspect of the input. Following this, the pooling layer is used to down-sample the feature map into a smaller size, while keeping the most important information. This process is repeated sequentially, resulting in feature maps that better represent complex features [38].

After feature extraction, the resulting feature maps are passed through a fully-connected neural network responsible for classifying the input image, by predicting a class label or probability score for each class [38].

2.3.3.4 CNNs Applied to GPR Object Detection

Neural networks are becoming the standard approach in GPR object detection [39]. In general, the proposed solutions detect hyperbolas by utilizing a pre-trained model and adapting it to improve its performance. This process usually includes retraining the model with new data, more relevant in the context of the experiment, and/or altering parts of the model's architecture to obtain a model that performs better. An in-depth review on AI based approaches for GPR data analysis can be found in [40].

There are several different types of CNN models which have been implemented in systems proposed for GPR object detection. These models can be divided into two categories: region selection and regression [39].

The region selection category detects features by proposing plausible candidate regions. These are then inputted into the convolutional layers which determine the feature maps. Consequently, the fully connected layer determines if that region is of interest or not, and classify the region. Some examples of these networks are R-CNN, Fast-RCNN, Faster-R-CNN and Mask-R-CNN. These provide highly accurate results, however, at the expense of detection speed [39], [41], [42].

In comparison, regression-based methods have an advantage in detection speed, as they don't need region proposal generation. Some examples of these are YOLO, SSD and CenterNet [39].

2.3.3.5 YOLO

The You Only Look Once (YOLO) family of networks is one of the more advanced one-stage detection models [39]. As indicated by its name, YOLO networks only need to look at the input image once, as a whole. YOLO works by dividing the input image into a grid of cells and predicting the object centered in each cell. In each cell, several bounding boxes are predicted, characterized by their confidence score, and the corresponding class probabilities are calculated. An object is detected in the bounding box according to the probability of it containing an object class [39], [41]–[44]. This process is illustrated in figure 2.22.

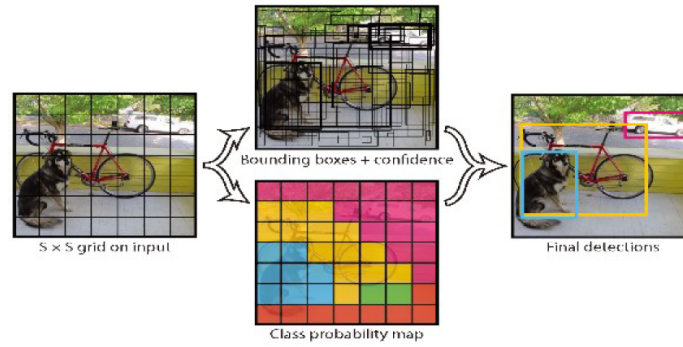


Figure 2.22: Prediction of bounding boxes in YOLO. [42]

YOLO is an open-source project with continuous development spanning many years. Currently, there exist 10 official versions and several other adaptations. An in-depth review of several YOLO versions can be found in [43]. At the time of the development of this dissertation, YOLOv8, released in January 2023 by Ultralytics [45], was the state-of-the-art version.

YOLOv8 is composed of three main parts: Backbone, Neck and Head [43], [44]. These are illustrated in figure 2.23. The backbone is used to extract relevant features from the input image. It is a modified version of the CSPDarknet53 architecture, consisting of 53 convolutional layers, organized in a sequential manner. A C2f module uses bottleneck and concatenation operations in order to combine high-level features with contextual information, to enhance accuracy. The SPPF module, and the following convolution layers in the Neck, process feature at various scales, using scaling operations. Finally, the Head uses convolutional and linear layers to map the high-dimensional features as predictions of bounding boxes and class probabilities. The activation function for bounding boxes and class probabilities are Sigmoid and Soft-max, respectively. Additionally, YOLOv8 uses the CIOU and DFL loss function for bounding-box and classification losses, leading to improved performance, particularly when dealing with small objects [43], [44].

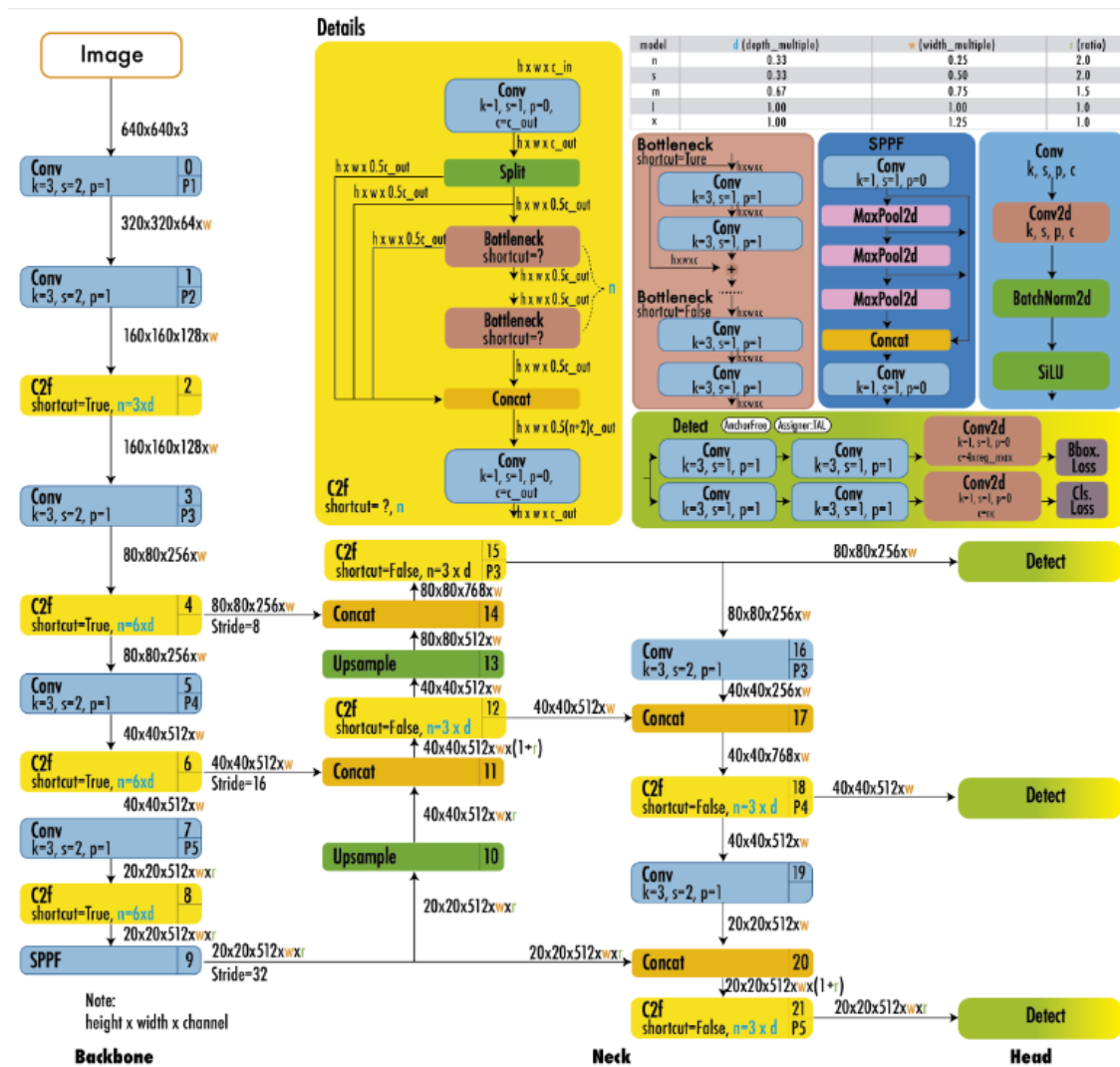


Figure 2.23: YOLOv8 architecture. [43]

YOLOv8-seg is a semantic segmentation version of the YOLOv8 model. Instead of predicting bounding boxes encompassing an object, this model produces pixel segmentation masks delimiting the exact boundaries of the object. This model's overall architecture is slightly changed from the YOLOv8 model. The Backbone is now a CSPDarknet53 feature extractor, followed by a C2f module. Next, this module is followed by two segmentation heads, used to predict the semantic segmentation masks. The Head consists of five detection modules and a prediction layer, used to classify masks [43], [44].

YOLOv8 introduces various variants with different variable model sizes and capabilities, catering to different scenarios, where there may exist limitations in model size or a need for greater precision or speed [43], [44]. In figure 2.24, the relation between accuracy, model parameter size and inference time for the bounding box versions are shown.

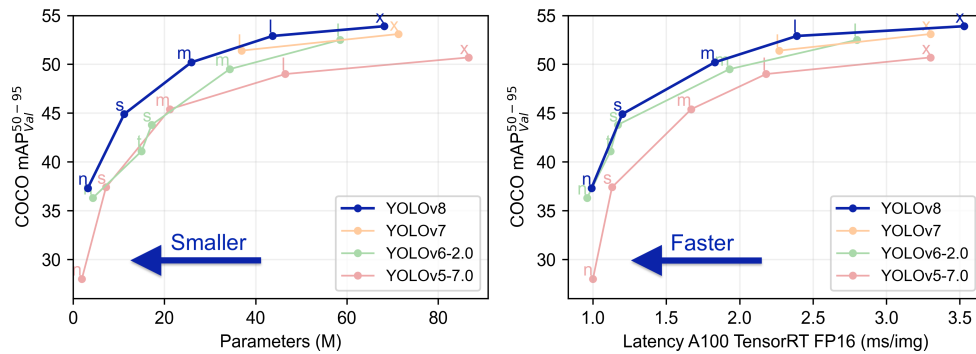


Figure 2.24: Comparison of accuracy vs model size and inference speed of YOLOv8 bounding box version on the COCO dataset. [45]

2.3.3.6 Image Augmentation

In order to train neural networks large amounts of data are required, such that diverse data samples exist to avoid overfitting. However, producing large datasets is a time-intensive activity. To mitigate this problem data augmentation techniques are used to generate realistic synthetic data from the original dataset. These techniques use the available images in order to generate new ones. Traditionally, this is done by applying geometrical transformations, colour-space transformations, filtering operations and noise injection [35].

Generative adversarial networks (GAN) are a type of CNN architecture used to generate synthetic images. Consequently, these may be used for the purpose of data generation, including the generation of GPR scans. Essentially, GANs are composed of two different networks, the generator, which generates images and the discriminator, which must determine if an image is real or fake. The goal of the generator is to generate images that fool the discriminator into accepting them as real. Consequently, since both networks are trying to be as accurate as possible, the generated images will be very similar to the real images [26], [46].

2.3.3.7 Neural Networks Applied to GPR Denoising

In GPR B-scans, the visibility of the hyperbolic signatures of subsurface objects is often obscured by cluttering signals. These are mainly caused due to the complexity of the embedded targets, coupling between the transmitter and receiver antennas, reflection from the ground surface and scattering response from non-uniform terrain [47], [48].

While more traditional denoising techniques exist, mainly based on filtering and transformations as described in section 2.3, autoencoders have been used to tackle this issue. An autoencoder is a type of CNN architecture that learns how to reproduce input data by transforming it through a hidden space. It is typically a three-layer neural network, as shown in figure 2.25 [47].

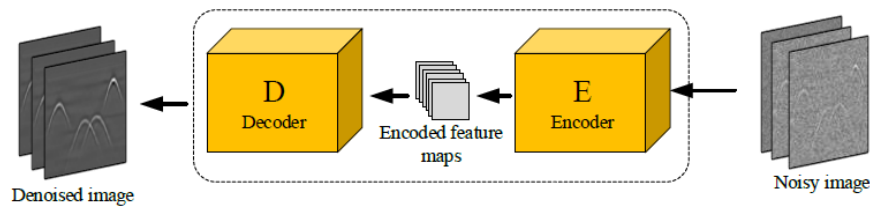


Figure 2.25: Architecture of an autoencoder. [48]

The encoder compresses the input data into feature maps while the decoder does the inverse operation, reconstructing the image from the feature maps [47]–[49]. In general, the autoencoder is able to determine which features belong to the background and which are of interest, thus being able to reconstruct a denoised image [47].

Other similar networks have also been used with the same objective. GANs can be used to generate a noise-free image from a noisy scan, using the generator. Then the discriminator can be used to evaluate the generated noise-free scan and determine if noise removal was successful, considering the network was trained with the noisy and noise-free data pairs [50]. In general, these methods focus on utilizing neural networks to analyze the B-scan image and predict a clutter-free image [24].

However, the training process of these networks requires that noisy and noise-free data pairs of the same scan exist. Since in practice, it's not possible to obtain noise-free scans, obtaining these data pairs heavily relies on image augmentation techniques to insert noise into a noise-free scan, obtained through simulation. Since the goal of these networks is to understand which components are noisy, the generated images need to have model realistic noisy images.

2.4 Summary

In this chapter, the main theoretical aspects and fundamental knowledge associated with the operational principles of GPR and data analysis methodologies were thoroughly exposed. Summarily, an object may be identified in a B-scan through the detection of its hyperbolic signature, which is related to its position, depth and radius of the object and the EM wave propagation velocity in the medium. These are modelled considering the geometric parameters of the scan and the time elapsed between transmission and detection. Hyperbola fitting strategies can be used to determine these parameters.

The investigation into the state of the art of GPR data analysis showed that, essentially, the problem of object detection and characterization may be solved through a combination of the techniques described in section 2.3. Traditional GPR data processing and computer vision techniques are essential in order to mitigate the effect of noise, cluttering signals and attenuation, having the goal of highlighting the hyperbola reflections. Neural networks are of particular interest to solve the problem of hyperbola detection given the complex nature of the data, which makes it harder to employ only classical vision techniques.

Chapter 3

Automatic Object Detection in GPR Data

In this chapter, all of the work developed to implement a system capable of automatically analyzing a GPR B-scan is described. For further reference, the scan's parameters consist of the object's depth, position and radius, and the propagation velocity in the medium. Firstly, the system's architecture is shown, providing a better high-level understanding of the system's components. Next, the experimental procedures and equipment used to collect the data are described. Following this, the methodologies used to process GPR data and training the neural network models are explained. The chapter concludes with a description of the implementation of the fitting procedure used to relate the mask segmentation predictions with the hyperbola model and, consequently, the scan's parameters.

3.1 System Pipeline

The system's pipeline is characterized by three main sequential components: Image processing, Hyperbola detection and Hyperbola fitting. Image processing is done with the goal of making it easier to identify hyperbolic patterns. It consists of both the traditional GPR processing and computer vision techniques, described in sections 2.3.2 and 2.3.3. Hyperbola detection is achieved using a YOLOv8-seg neural network. The networks were trained using several multi-channel encoding approaches, with the goal of maximizing the information inputted into the model. Different processing techniques were applied to GPR B-scans and stored in each RGB channel of the image, such that complementing processing techniques could be passed to the network. Hyperbola fitting was done after hyperbola segmentation with the goal of estimating the object's parameters, using the hyperbola model described in section 2.2.2.2 and the constrained least-squares fitting strategy described in section 2.2.3. Additionally, this step also includes a hyperbola mask processing component, used to determine the hyperbola fitting points. The system's pipeline is illustrated in the diagram of figure 3.1.

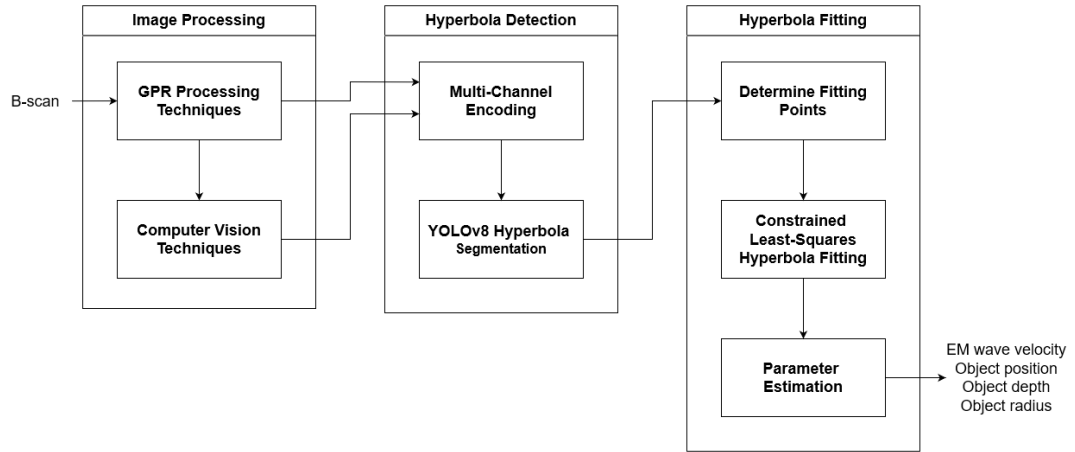


Figure 3.1: System's pipeline.

The system was developed using the Python programming language, due to its compatibility with the several established libraries used in computer vision and machine learning, including PyTorch and OpenCV.

3.2 GPR Data Gathering

3.2.1 Equipment

The Ground penetrating radar used to collect the data used in this dissertation was the OERAD Concretto [51]. Its technical specifications are listed in table 3.1. The surveys were all done using the 7.5 ns operating time window. Additionally, this GPR has an encoder in its rear-left wheel which makes it possible to obtain equidistant A-scans, given that the pulse transmission is synced with the movement of the encoder wheel.

Table 3.1: OERAD Concretto technical specifications. [51]

Parameter	Values
Frequency Bandwidth	200 MHz - 4000 MHz
Central Frequency	2000 MHz
Operating Time Windows	7.5 ns / 15 ns
Horizontal Resolution	0.001 m
Vertical Resolution	0.001 m
Pulse Length	600 ps
Pulse Energy	30 V
Pulse Rise Time	200 ps
Peak Tx Power	3 W
Samples per trace	585

3.2.1.1 Image Scaling

The B-scans obtained from a survey consist of images where the vertical and horizontal axis represent time and horizontal distance, respectively. However, the pixel units obtained from the B-scan image correspond only to the number of samples and traces taken, respectively. Consequently, in order to obtain meaningful physical units from the radiograms, the acquisition settings of the survey have to be taken into account. Thus, it's possible to obtain the ratios which allow for the conversion into physical units. Considering that the operating time window was set to 7.5 ns for all scans and that the number of traces per meter was determined to be 135, calculated over an average of 10 scans:

$$\epsilon_{\text{samples to seconds}} = \frac{7.5 \cdot 10^{-9}}{585} = 1.28205 \cdot 10^{-11} \quad (3.1)$$

$$\epsilon_{\text{traces to meters}} = \frac{1}{135} \quad (3.2)$$

Considering a unit analysis, it's also possible to determine the constant which relates velocity to real units:

$$\frac{\text{traces}}{\text{samples}} \rightarrow \frac{\text{meters}}{\text{seconds}} : \epsilon_{\text{traces/samples to meters/seconds}} = \frac{\epsilon_{\text{traces to meters}}}{\epsilon_{\text{samples to seconds}}} \quad (3.3)$$

The real units may be obtained by multiplying the pixel units in B-scans with these constants, as shown in the following equations.

$$t_{\text{seconds}} = t_{\text{samples}} \cdot \epsilon_{\text{samples to seconds}} \quad (3.4)$$

$$x_{\text{meters}} = x_{\text{traces}} \cdot \epsilon_{\text{traces to meters}} \quad (3.5)$$

$$v_{\text{meters/seconds}} = v_{\text{traces/samples}} \cdot \epsilon_{\text{traces/samples to meters/seconds}} \quad (3.6)$$

3.2.2 Dataset Creation

The surveys were gathered considering two different setups, described in the following subsections. For both, the probing medium consisted of soil where solid metallic cylindrical objects were buried perpendicular to the scan's direction. Considering the effect of attenuation, objects weren't buried at depths greater than 20 cm. Metallic objects were used since their reflections could be easily identified, however, due to the ringing effect, described in section 2.1.4.1, multiple hyperbola reflections were observed. The radius of these objects is shown in table 3.2.

Table 3.2: Radius of buried objects used.

Object Name	Radius (cm)
Object 1	1.4
Object 2	0.5
Object 3	1.5

3.2.2.1 Neural Network Training Dataset

As previously stated, in order to train neural networks large amounts of data are required. Consequently, an experimental setup was created which allowed for a convenient way to re-arrange the buried objects, thus decreasing the time needed to obtain a scan. This setup consisted of a plastic box filled with soil where the objects were inserted vertically, as illustrated by figure 3.2. Since objects weren't actually buried it was much faster to re-arrange their position. The GPR scans were taken on the largest side wall of the box, such that the distance between the object and the wall corresponds to the depth measured by the scan. However, due to the weight of the soil, the box's plastic walls started to deform, which made using ground-truth measurements of depth and position unreasonable.



Figure 3.2: Experimental setup using plastic box filled with soil. Objects were buried vertically and GPR scans were taken on the biggest side wall.

The scans were obtained in a sequential manner. First, a scan of the entire length of the box was taken with all three objects buried, where Object 2 was between Object 1 and Object 3. Next, Object 2 was removed and again the entire box's length was scanned. Finally, Object 1 and Object 3 were scanned individually. This process was repeated in both directions of the survey line.

Using this experimental procedure, 294 B-scan images were obtained. In accordance with the dataset organization principles, previously explained in section 2.3.3.2, the data was split into a training, testing and validation subsets, following an 80%, 10%, 10% splits, respectively.

The data was annotated using the CVAT annotation tool [52]. For each hyperbola observed in the B-scan, the bounding box, segmentation mask and hyperbola apex were labelled at the transition between positive and negative signal amplitude values, as illustrated in figure 3.3. Due to the approach taken, only the segmentation masks were necessary but regardless the dataset

was labeled considering other possible approaches. Labelling was done after applying the GPR processing techniques, such that all of the hyperbolas could be identified. The annotations were exported in CVAT's XML format and tools were developed for the conversion to/from COCO annotation format, compatible with YOLO. This was necessary since despite CVAT allowing to export annotations in YOLO compatible format, at the time when this dissertation took place, it only supported the conversion of bounding boxes and not segmentation masks. Additionally, the CVAT XML format was used to save the annotations obtained from augmented images, also making it necessary to implement local annotation format conversion.

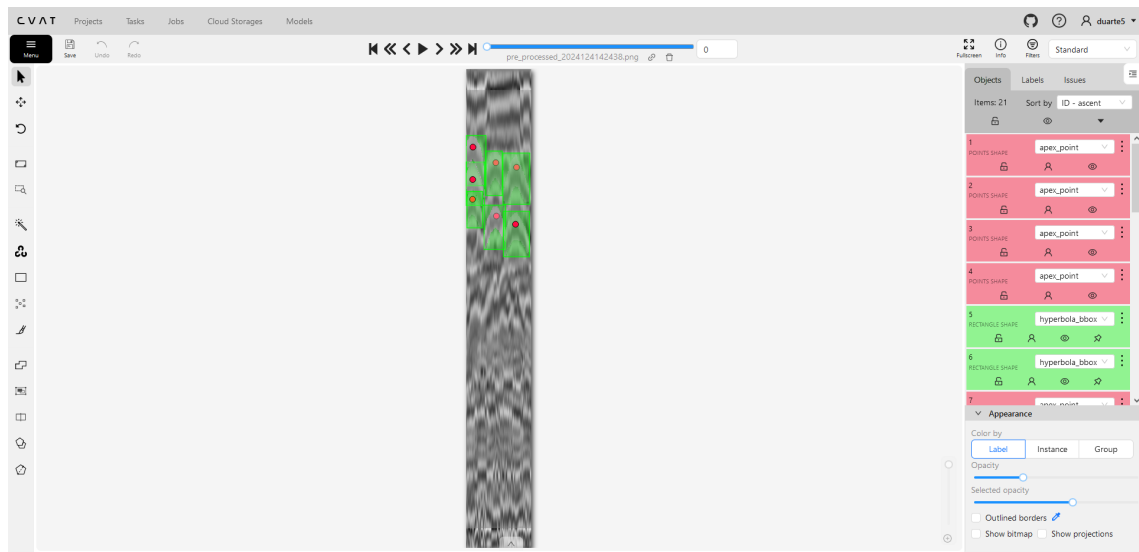


Figure 3.3: CVAT annotation tool used for labelling hyperbola bounding boxes, masks and apex position.

3.2.2.2 System Verification Dataset

A second smaller dataset was gathered in order to evaluate the accuracy of the parameters estimated by the complete system. The experimental setup used consisted of a rectangular space filled with soil, walled with concrete blocks with a height of 20 cm. The objects were buried horizontally, parallel to the surface of the medium. The GPR scan was taken from above in order to not be affected by any layer apart from the soil's surface. The GPR's wheels were set on top of two planks, given the need for a smooth surface, such that the GPR was elevated from the surface by 3 cm. Consequently, this would later have to be corrected by taking into account the time-zero offset, by subtracting the corresponding time. In order to avoid reflections caused by the concrete blocks, the scan was started and terminated with a 20cm distance to the end walls, and the survey line also had a distance greater than 20cm to the side walls. The experimental setup is shown in figure 3.4.



Figure 3.4: Experimental setup used for the verification of the system. Objects were buried horizontally and GPR scans were taken along the soil's surface considering a height offset

The scans recorded only had one buried object at a time. The object would be buried and its depth and position registered. Then five scans would be taken for each direction. This process was repeated for all three objects, resulting in a dataset with 30 B-scans.

3.2.3 Data Processing

Different data processing techniques were applied to the raw B-scans images, including both GPR processing and computer vision techniques. These were applied with the goal of removing the effect of noise and cluttering signals and highlighting the hyperbolas, allowing an easier identification.

3.2.3.1 GPR Processing Techniques

A sequential pipeline composed of the following GPR processing techniques was used: dewow, mean background removal and band-pass filter. This pipeline was applied to every B-scan.

Dewow was applied in order to remove the low-frequency wow effect. This filter was applied using a window of 50 samples, which corresponds to the 600 ps pulse length of the GPR system used.

Next, background removal via the mean subtraction approach was applied. This method had a very noticeable impact on the interpretability of the B-scans since it removed the cluttering signals caused by the ringing effect of the antennas and air-surface reflection. Principal component analysis methods weren't used due to the difficulty in determining the components of interest, which would lead to removing too much information. Neural network approaches weren't used due to the reliance on simulated data in modelling the noisy and noise free scans used in training.

Finally, a band-pass filter was used in order to remove noise from frequencies outside the frequency spectrum of the GPR's antennas. Hence, the filter used was a 20th-order Butterworth filter with minimum and maximum cutoff frequencies set to 200 MHz and 4000 MHz.

Additionally, automatic gain control was applied to the scan, after the previous processing, in order to mitigate attenuation. Like in the dewow filter, a window of 50 samples was used since this corresponded to the GPR pulse length. However, this gain heavily increased noise, particularly at higher depths, where the signal had a very low amplitude. Consequently, this technique was used as part of the multi-channel encoding strategy, usually combined with the other techniques described. This technique was later integrated considering the multi-channel encoding strategies.

These techniques are illustrated in figure 3.5.

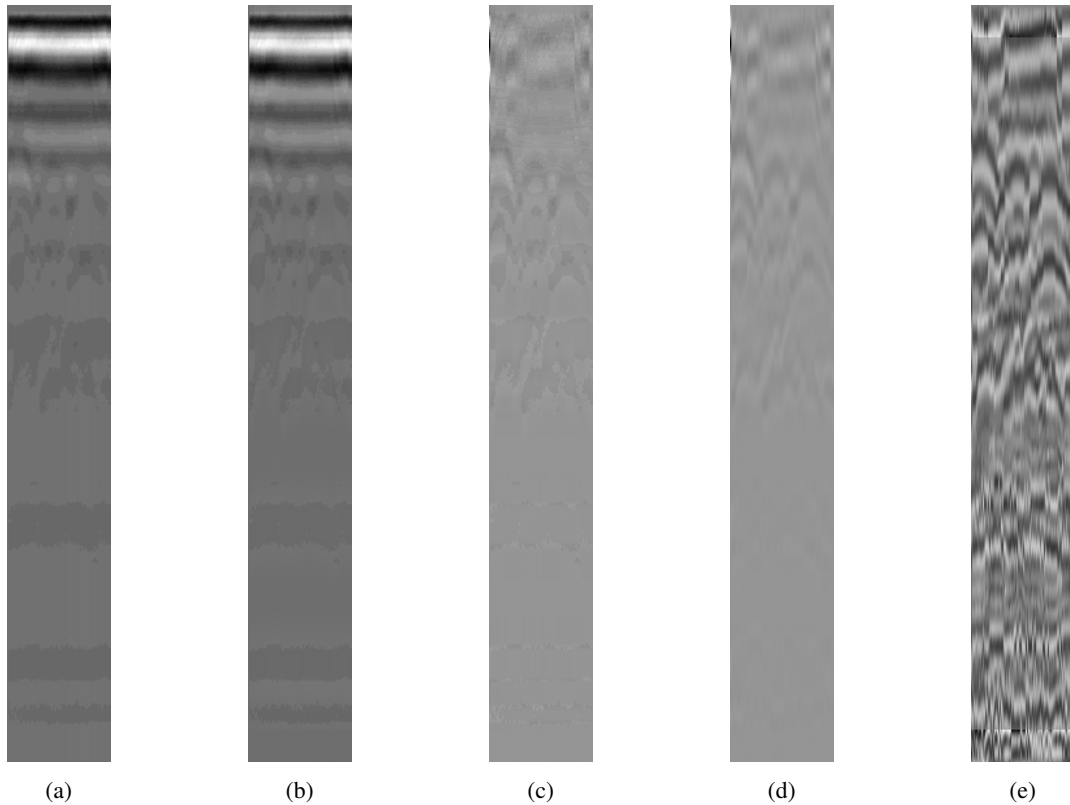


Figure 3.5: GPR processing applied sequentially to a collected B-scan. a) Original, b) Dewow, c) Background removal, d) Band-pass filter, e) AGC

3.2.3.2 Computer Vision Techniques

Computer vision techniques were applied after the GPR processing techniques with the goal of highlighting the hyperbolas. These techniques were later employed in the multi-channel encoding strategy.

A template matching approach was used to determine the similarity between the received signals and a reference GPR signal, using the normalized cross-correlation after applying the GPR processing pipeline to the data. The reference GPR signal was obtained by identifying the GPR pulse in a scan taken of Object 1, after applying the GPR processing pipeline. The probing medium was air, such that the obtained signal wasn't affected by noise or cluttering signals. Thus, this

signal would correspond to the transmitted pulse. The reference signal used is illustrated in figure 3.6.

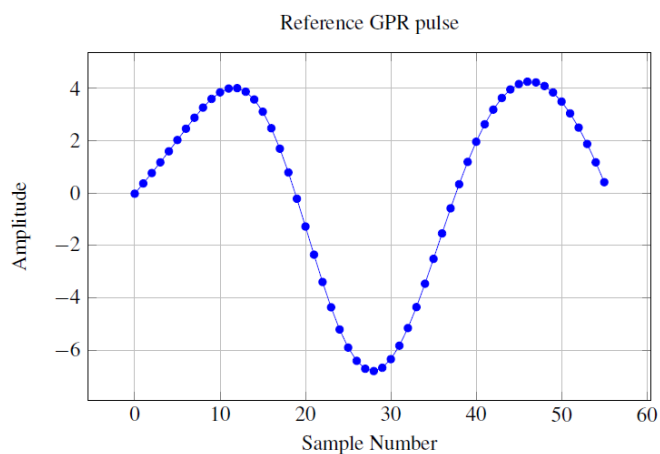


Figure 3.6: GPR reference pulse. Obtained from a hyperbola of Object 1, where the probing medium was air.

A Sobel edge detector filter was also used in order to identify edges in the B-scan image, after applying the GPR processing pipeline, including AGC. This was done by applying two Sobel filters and adding their results, one with a horizontal kernel and one with a vertical kernel. This allowed for the detection of both horizontal and vertical edges, necessary since the hyperbola's branches are angled diagonally, making it possible to highlight the edges of the hyperbola. These techniques are illustrated in figure 3.7.

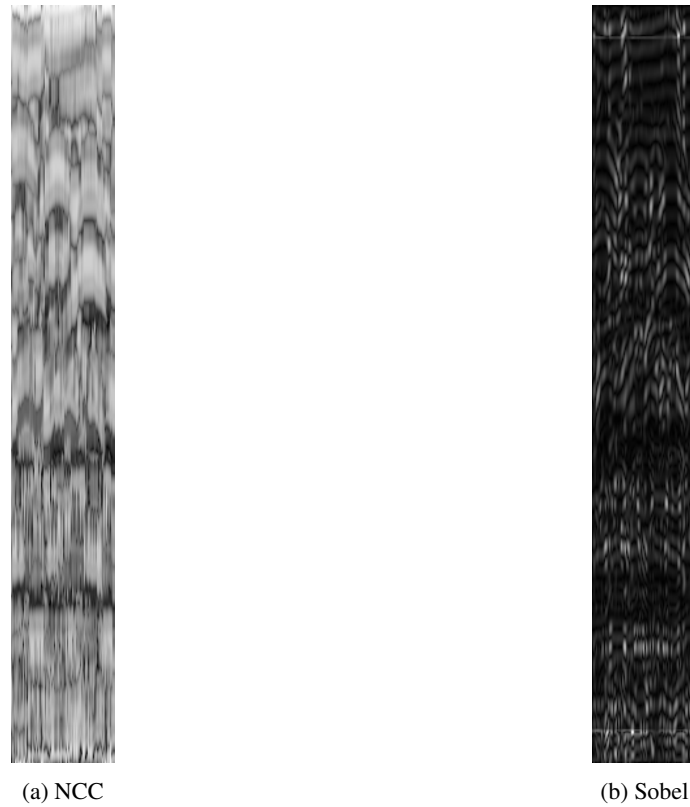


Figure 3.7: Computer vision techniques applied sequentially to the B-scan shown in figure 3.5.

3.2.4 Image Augmentation

Due to the need to train neural networks, it was necessary to use image augmentation techniques to increase the network training dataset. Traditional augmentation techniques were preferred, as opposed to GANs. This was due to the fact that GANs also need to be trained, so a large dataset would already be required, and the generated images aren't annotated. Consequently, techniques based on geometrical transformations and contrast transformations were preferred, since it's possible to recalculate the annotations of the original image.

The techniques were chosen with the goal of modeling B-scans. Geometrical transformations were used to simulate objects and scans with different sizes. Contrast transformations were used to model objects with different relative permittivities, which would result in reflections with different amplitudes. These techniques are described in the following paragraphs.

- Contrast Limited Adaptive Histogram Equalization (CLAHE) is a noise-robust contrast transformation. The image is divided into a grid of tiles and contrast is adjusted locally inside each tile. In order to prevent the amplification of noise, contrast is limited. This process is repeated for every tile, which are then combined and their edges smoothed [35].
- Flipping consists of mirroring the image relative to an axis. Given the context of GPR B-scans, only horizontal flipping was used [35].

- Resizing was used to change the dimensions of the image. Height and width were changed randomly, limited by set bounds [35].
- Rotation was applied in order to change the angle of the scan image. The rotation angle was bounded, such that the image wasn't over-rotated [35].

In order to generate a greater number of images, several combinations of these techniques were applied to the original image and between the augmented images. After applying these transformations 8 images were generated from an original scan. Augmentation was done on the images after applying the GPR processing techniques. This was done such that the results from the networks trained with the various multi-channel encoding techniques were obtained from images with the same augmentations and could later be compared since the other techniques would always be applied after the GPR processing pipeline. The images generated from the augmentation can be observed in figure 3.8, resulting from the augmentation of a B-scan image after applying GPR processing techniques and AGC.

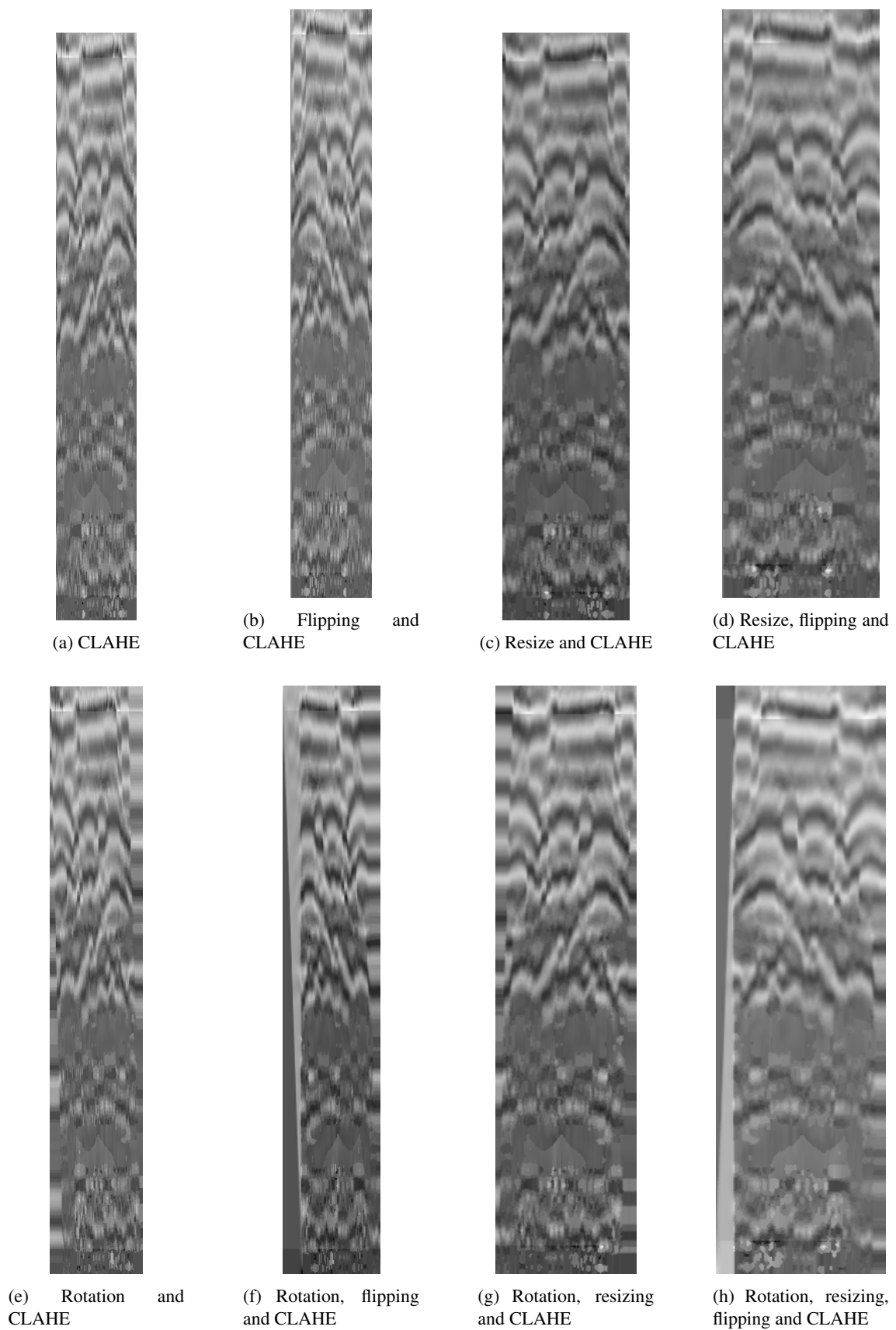


Figure 3.8: Images resulting from the augmentation pipeline, applied to the B-scan shown in figure 3.5 after GPR processing and AGC.

3.2.5 YOLOv8-seg Applied to Hyperbola Segmentation

As in many computer vision tasks, neural networks have become widespread in GPR hyperbola detection, due to the complex nature of the scan image, where identifying the hyperbolas is made difficult by the effect of noise, cluttering signals and overlapping hyperbolas, even after processing the B-scans. Consequently, YOLOv8 segmentation models were explored for the task of detecting and segmenting the hyperbolic pattern. These models were trained following the strategies described next.

3.2.5.1 Multi-Channel Encoding Strategy

Multi-channel encoding consists of storing different information in each of the three image RGB colour channels. Thus, it's possible to take full advantage of the YOLO networks, which support inputs with three channels, by providing it with multiple B-scans images, where different processing techniques were applied, and combined in an RGB image.

In order to determine the validity of this approach, multiple combinations of processed B-scans combined into RGB images were explored when training the networks. These were then analysed to determine which combination provided the best results. These combinations were chosen with the goal of providing complementary information to the network. Some configurations don't include multiple processing techniques, such that the results from these may be compared to those where this is done. In these cases, the image is grey-scale since all channels have the same values. The chosen combinations of B-scan processing techniques into RGB images used are listed in table 3.3, as multi-channel encoding configurations (MCEC), and can be visualized in figure 3.9.

Table 3.3: Multi-channel encoding configurations (MCEC) of different B-scan processing techniques.

MCEC	Blue Channel	Green Channel	Red Channel
1	GPR Processing	GPR Processing	GPR Processing
2	GPR Processing	AGC	-
3	GPR Processing	NCC	-
4	GPR Processing	Sobel	-
5	GPR Processing	AGC	NCC
6	GPR Processing	AGC	Sobel
7	AGC	AGC	AGC
8	AGC	NCC	-
9	AGC	Sobel	-
10	AGC	NCC	Sobel
11	NCC	NCC	NCC
12	Sobel	Sobel	Sobel

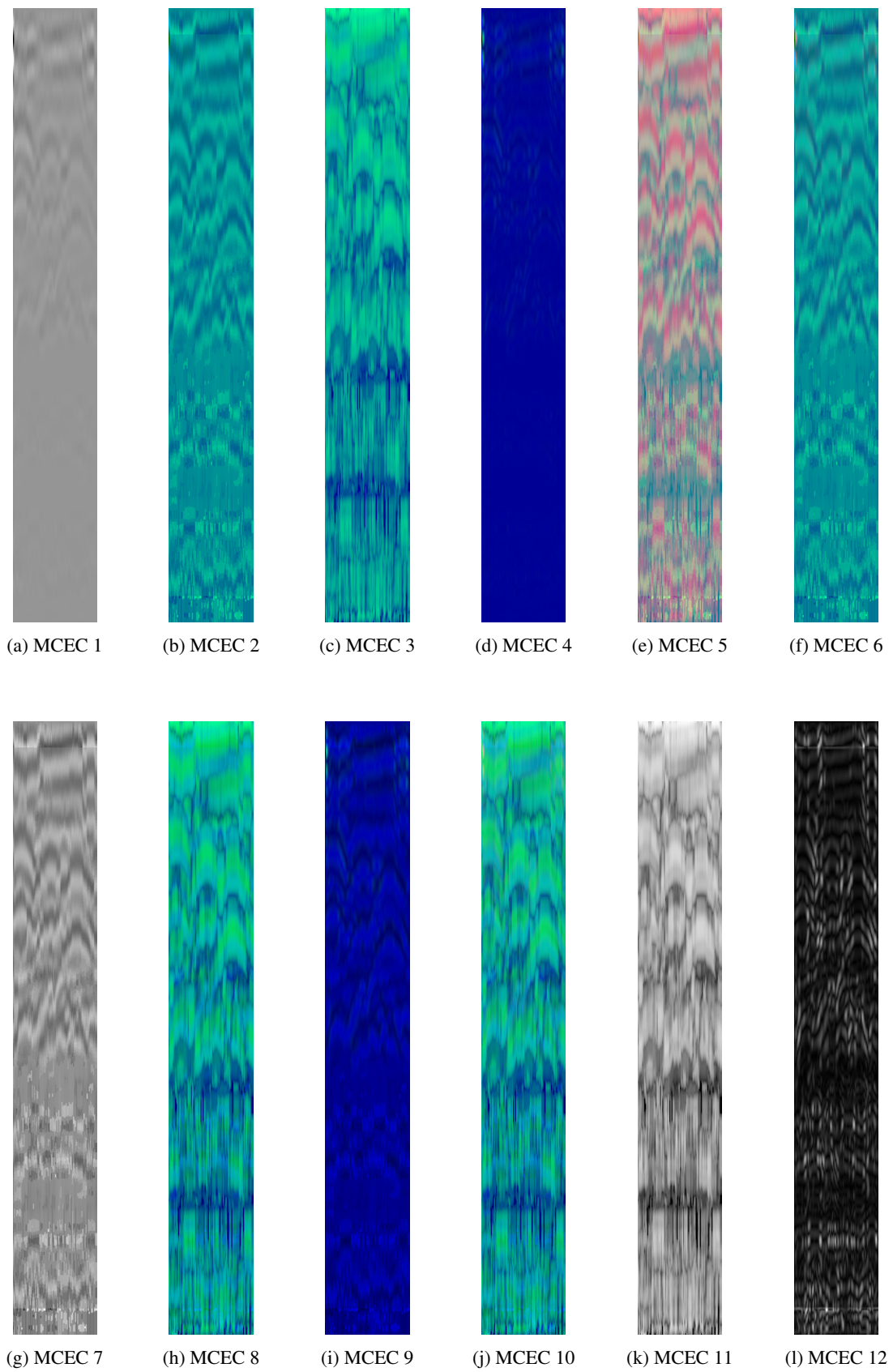


Figure 3.9: Multi-channel encoding of configurations (MCEC), where different B-scan processing techniques are combined, applied to the B-scan shown in figure 3.5.

3.2.5.2 Training Procedure

YOLOv8-seg nano models were trained for each of the combinations of processed B-scans following the multi-channel encoding strategy described previously. Then, after analysing the results the YOLOv8-seg medium model was trained in order to determine if better results could be obtained with a more complex model. These models were trained for 40 and 20 epochs, respectively, using the provided pre-trained YOLO models in the COCO dataset. The chosen number of epochs was obtained after experimentation, such that the results obtained after training indicated that no further improvement was possible, as further discussed in sections 4.1 and 4.2. For all, the hyperparameters used were the defaults, except for those listed in table 3.4. These were adjusted such that the training provided satisfactory results.

Table 3.4: Training hyperparameter used in train the YOLOv8-seg models.

Hyperparameter	Value
Learning rate	0.0001
Image size	640
Batch size	24
Dropout	0.2
Optimizer	SGD
Early stopping	Off

The models were trained and their performance was evaluated using the dataset splits described previously. The model's output consisted of hyperbolas mask predictions associated with a confidence score. The mask prediction consisted of lists of points belonging to a hyperbola. Additionally, the training could be evaluated via the metrics described in section 2.3.3.2, namely, precision, recall, f1-score and the several losses which characterize errors.

3.2.6 Hyperbola Prediction Fitting

In order to determine the parameters of from a B-scan, namely, the object's position, depth and radius, and the medium's wave propagation velocity, the predicted hyperbola masks were fitted with the GPR hyperbola model via a contained least-squares fitting algorithm, as described in sections 2.2.2.2 and 2.2.3.

After training, the model which provided the best results was chosen and used when implementing the system. The whole pipeline, now including hyperbola fitting, was verified using the verification dataset. Since it was considered beneficial to obtain false positive detections as opposed to missing an object, the confidence threshold was set to 0.15. The IoU threshold was set to 0.5 to avoid overlapping detections. For each B-scan image, the YOLO network predicted segmentation masks for each hyperbola it could identify. The masks predicted by a trained YOLO model can be observed in figure 3.10.

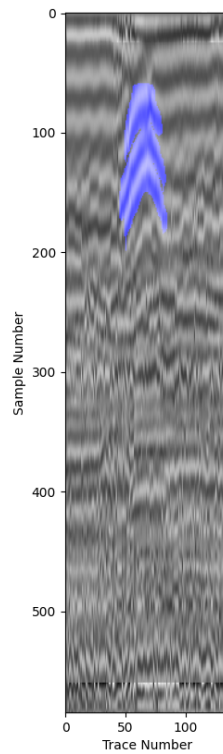


Figure 3.10: YOLO hyperbola mask predictions from trained model implemented in the system. For visualization purposes, B-scan is shown after applying GPR-processing techniques and AGC. Multiple hyperbolic reflections exist due to the ringing effect.

Two approaches were explored when using YOLO to predict the masks. First, the masks were obtained by considering only the results obtained from a single YOLO prediction. Later, a prediction averaging approach, based on test-time augmentation, was experimented. Test time augmentation is a technique used to improve the performance of neural networks during the inference phase [53]. In the context of image processing, this process involves providing the model with multiple images augmented from the image being analyzed, thus obtaining multiple predictions. The prediction from this image is obtained by considering all of the multiple predictions. Consequently, this technique improves the model's robustness, making it more resilient to variations in input data.

This technique was implemented in the system by obtaining 100 predictions from images augmented from the input image, such that the final prediction was obtained from a broad range of predictions. These augmentations consisted of resizing and horizontal flipping. The masks obtained were re-converted to the original image coordinates by reversing the geometrical augmentation applied. IoU was used to associate masks from multiple predictions. Masks were matched by considering the biggest IoU between a mask and all other masks of previous detections. A minimum IoU of 0.2 was defined to avoid matching between masks with too little overlap. If a mask matched with another, these represented the same hyperbola, thus the number of times it was detected was incremented. Additionally, the mask of a hyperbola was considered to be the mask with a bigger area, since it provided a more complete segmentation of the hyperbola. Finally, the

final prediction of hyperbola masks was obtained, considering a minimum of 50 detections. This process is illustrated in figure 3.11.

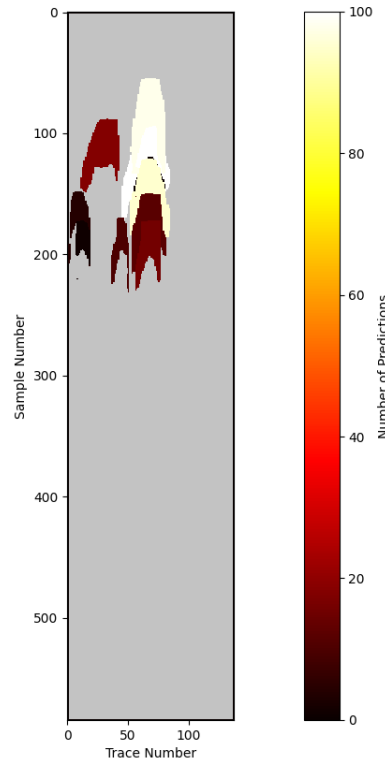


Figure 3.11: Number of YOLO hyperbola mask predictions predicted from 100 augmented images, applied to the B-scan shown in figure 3.10.

As previously stated, due to the ringing effect, multiple hyperbolas per object would be present in a B-scan, all of which would be identified by the YOLO model. Additionally, the YOLO mask predictions could have unexpected shapes, where branches from other hyperbolas would also be present. Consequently, the following algorithms were developed.

3.2.6.1 Hyperbola Mask Processing

An algorithm was developed in order to determine the fitting points from a mask prediction. The algorithm started by skeletonizing the hyperbola mask, through which the essential shape of the mask was obtained. The skeletonization transform produces a skeleton, which consists of pixels that are of equal distance to two or more boundaries of a component [26].

Due to the fact that YOLO masks could have unexpected shapes, some other considerations would have to be taken into account. First, the biggest connected component would have to be selected, since masks could have multiple disconnected components it was considered that the biggest one would represent the hyperbola. Masks could also have holes, that is, points not belonging to the mask would exist surrounded by points which belonged to the mask. Since the hyperbolic signature is a continuous shape, these holes were set to belong to the mask. Finally,

hyperbola masks could have more than two branches. This would occur when overlapping hyperbolas or signals resembling hyperbolas existed. Here the predicted mask could follow all of the branches resembling the downwards sloping branch. Additionally, the skeletonized points also presented a smaller upward branch near the hyperbola apex, since at this place the mask would usually have larger height differences between the top and lower part of the mask. Consequently, the algorithm would determine the widest connected branch from the skeletonized points. This was done by determining the intersection points where multiple branches connected and then determining the widest continuously connected branches, which would correspond to the correct hyperbola. This process was repeated recursively until all possible branch connections were determined. Taking into account the hyperbolas' shape only horizontal connections between branches were considered valid, that is, between branches where no overlap in the x coordinates existed. After determining the skeleton which effectively describes the hyperbola, the fitting points are obtained by considering the mask's mean point in each hyperbola's branch. This algorithm is described in algorithm 1 and its results are visualized in figure 3.12.

Algorithm 1: Determine Fitting Points

Input: Hyperbola mask (*hyperbola_mask*)

Output: Hyperbola fitting points (*fitting_points*)

```

biggest_component ← determine_biggest_connected_component(hyperbola_mask)
biggest_component ← fill_holes(biggest_component)
skeleton ← skeletonize(biggest_component)
branch_connection_points ← determine_branch_points(skeleton)
branches ← determine_branches(skeleton, branch_connection_points)
connected_branches ← []

// Determine all possible connections between branches
for each branch1 ∈ branches do
    connected_branches ← connected_branches ∪ branch1
    // Look for connections between already determined connected branches and
    // remaining branches
    for each branch2 ∈ connected_branches do
        for each branch3 ∈ branches do
            if not exists_x_overlap(branch2, branch3) then
                if connected_by_branch_connection_point(branch2, branch3, branch_connection_points)
                then
                    connected_branch ← branch2 ∪ branch3
                    if not branch_already_exists(connected_branches, connected_branch) then
                        connected_branches ← connected_branches ∪ connected_branch

widest_branch ← determine_widest_branch(connected_branches)
fitting_points ← determine_fitting_points(widest_branch)
return fitting_points

```

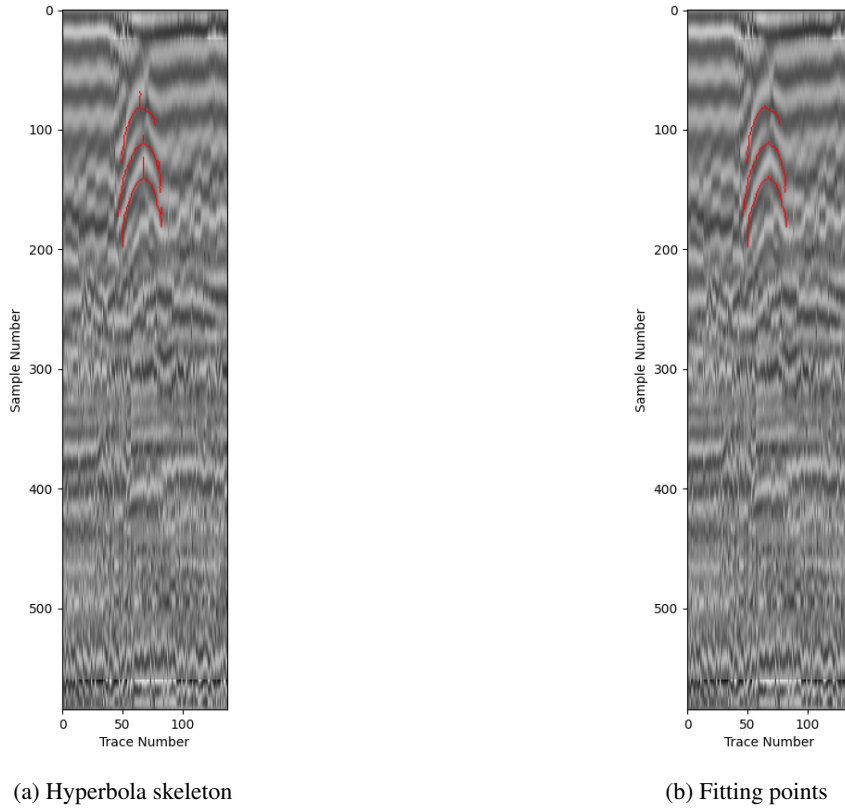


Figure 3.12: Hyperbola skeleton and resulting fitting points, applied to the B-scan shown in figure 3.10.

3.2.6.2 Hyperbola Fitting

The fitting points obtained from algorithm 1 were then fitted by the hyperbola model following a constrained least-squares minimization, described in sections 2.2.2.2 and 2.2.3. This was implemented using the Pyomo software package [54] and the IPOPT solver [55]. Pyomo is an open-source Python project used as a framework for modelling optimization problems, supporting integration with a variety of different optimizers. IPOPT (Interior Point Optimizer) is a software package designed to solve large-scale nonlinear and/or non-convex optimization problems. It is designed to find the local solutions of mathematical optimization problems subject to constraints, making it suitable for this fitting problem. In short, IPOPT works by minimizing an error function by iteratively adjusting the problem's parameters in the direction where the gradient of the error function decreases. Constraints are included in the error function as a penalty parameter, increasing the error if they are not satisfied. A thorough mathematical description of this solver can be found in [55].

The optimization problem is described by equations 2.46, 2.46 and 2.47 given restrictions from equation 2.48. In fact, given the horizontal resolution of 0.001m of the GPR system used, the minimum value for the radius was set to half this value, since no objects with a diameter smaller than the resolution could be detected. In figure 3.13 the results from this fitting process can be observed.

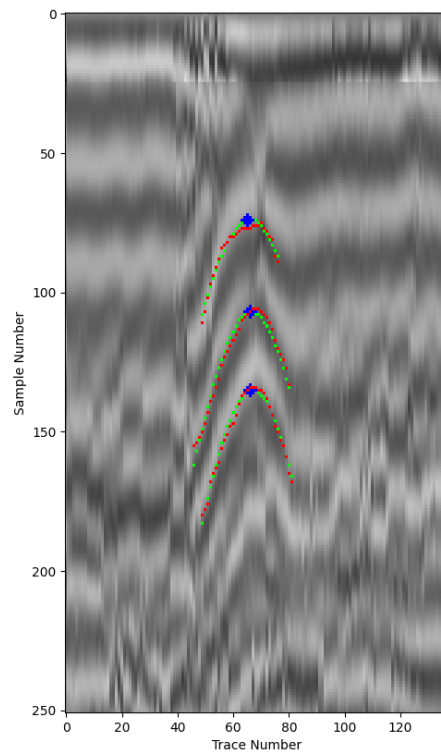


Figure 3.13: Constrained least-squares fitting of GPR hyperbola model, applied to the B-scan shown in figure 3.10. Fitting points are shown in red, fitted points are shown in green and the hyperbola apex is shown in blue.

3.2.6.3 Parameter Estimation

After hyperbola fitting the scan's parameters, namely object position, depth and radius and medium wave velocity, were calculated from the conic coefficients obtained. First, the conic parameters were related with the hyperbola model parameters using equations in 2.47. Next, the fitting was rejected if the position of the hyperbola apex, x_0 , was outside the fitting points range of x coordinates, given a margin of 5 pixels. This operation would effectively filter out fitting results from masks where only half of the hyperbola shape was identified, which was considered insufficient for a proper fit.

Given the rigging effect which caused multiple hyperbolic reflections per object, an algorithm was developed to identify the first hyperbolic reflection, which effectively relates to the object's position and the following hyperbola repetitions, also referred to as echos. These hyperbola groups were associated by analysing the position of the x coordinate of the hyperbola apex points and the hyperbolas' width. If these were similar between hyperbolas, given a margin, the hyperbolas were set to the same hyperbola group. After sorting all the hyperbolas into groups, the first-hyperbola was identified by having the smallest apex depth, given by the signal's travel time, t_0 . This process is described in algorithm 2 and illustrated in figure 3.14.

Algorithm 2: Determine First Hyperbola and Echos

Input: Fitted hyperbolas points (*fitted_hyperbolas_points*)
Output: Hyperbola groups, sorted by hyperbola apex depth (*hyperbola_groups*)

$x_margin \leftarrow 5$

$hyperbola_groups \leftarrow []$

// Determine hyperbolas belonging to a hyperbola group

for each *hyperbola1* **in** *fitted_hyperbolas_points* **do**

if *is_not_valid_hyperbola*(*hyperbola1*) **then**

$\hookrightarrow$ continue

$apex_x1 \leftarrow get_apex_x(hyperbola1)$

$hyperbola_width1 \leftarrow determine_hyperbola_width(hyperbola1)$

 // Check against all other fitted hyperbolas

for each *hyperbola2* **in** *fitted_hyperbolas_points* **do**

if *is_not_valid_hyperbola*(*hyperbola2*) **then**

$\hookrightarrow$ continue

$apex_x2 \leftarrow get_apex_x(hyperbola2)$

$hyperbola_width2 \leftarrow determine_hyperbola_width(hyperbola2)$

$apex_x_distance \leftarrow |apex_x1 - apex_x2|$

$hyperbola_width_difference \leftarrow |hyperbola_width1 - hyperbola_width2|$

 // Hyperbola echo found

if *hyperbola1* $\neq$ *hyperbola2* **then**

if $apex_x_distance < x_margin$ & $hyperbola_width_difference < 4 \cdot x_margin$ **then**

 // Update hyperbola groups lists. Append fitted points to the correct hyperbola group list. Create a new hyperbola group by adding both hyperbolas if no group was found, or append only the second hyperbola if a group containing the first hyperbola already exists.

$hyperbola_groups \leftarrow insert_in_hyperbola_group(hyperbola1, hyperbola2)$

 // Check hyperbola groups, if none was found where the first hyperbola exists, no echos exist, add a list with only first-hyperbola.

$hyperbola_groups \leftarrow insert_in_hyperbola_group(hyperbola1)$

// Sort hyperbola groups per apex height, such that the first hyperbola is in the first position.

$hyperbola_groups \leftarrow sort_hyperbola_groups(hyperbola_groups)$

return *hyperbola_groups*

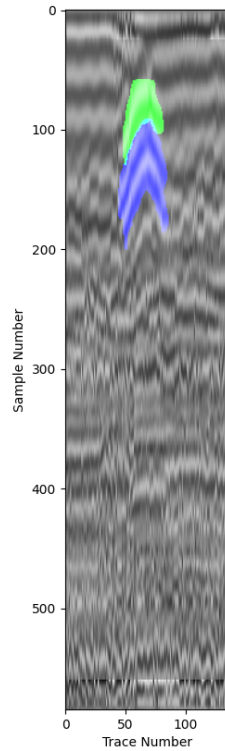


Figure 3.14: Results from algorithm 2, applied to the B-scan shown in figure 3.10. First-hyperbola is shown in green and echos in blue.

Finally, the scan's parameters were determined. Wave velocity and object depth were calculated from the hyperbola parameters using equations 2.29 and 2.36. Time-zero offset was corrected by subtracting the time needed to travel the known distance where the signal propagated in the air. The object's position directly corresponded to the apex's position, x_0 . These parameters were then converted into real units following the image scaling conversions described in section 3.2.1.1.

Due to the fact that the radius determined from hyperbola fitting is heavily affected by miss-shaped hyperbolas, the radius was calculated using the energy radius method, as described in section 2.2.3. The energy radius was calculated considering the calculated depth, using equation 2.15. Then, considering the horizontal width of the hyperbola, the radius was obtained using equations 2.49 and 2.50. The hyperbola width was obtained from the width of the segmentation mask or from double the distance from the x position of the determined apex to the furthest x coordinate of a point belonging to the fitted points. The biggest of these distances would be considered the hyperbola's width. This was done with the goal of mitigating the effect of asymmetric segmentation masks, where the full length of both of the hyperbola branches wouldn't be detected, such that the mask's width wouldn't effectively correspond to the actual width of the hyperbola.

All of these parameters were calculated from the first hyperbolic reflection and also from a mean of the parameters considering the first hyperbola and its echos, and the results compared. This was done with the goal of trying to take advantage of the multiple hyperbola reflections and obtaining a more accurate result, by mitigating the effect of a miss-shaped hyperbola in the

fitting process. When calculating the parameters, the time-zero offset was taken into account by subtracting the known distance where the signal travelled in the air.

Chapter 4

Results and Discussion

In this chapter, the results from the work presented in the previous section are exposed and discussed. First, the models trained with the different multi-channel encoding strategies are analyzed. Next, the accuracy of the parameters estimated from the whole scan analysis pipeline is evaluated. The chapter concludes with a discussion of the obtained results and a critical analysis of the developed system.

The YOLO framework automatically provides several graphs useful for evaluating the success of the training. Notably, the evolution per epoch of loss functions, precision, recall, mAP50 and mAP50-95 are plotted for the training and validation datasets. Additionally, the evaluation metrics are related in several graphs, for the training, validation and testing datasets. The metrics from the validations and testing datasets are also shown. Since a segmentation model can effectively also identify an object's bounding box, by considering the height and width of the segmentation mask, these metrics are provided for both of the bounding boxes and masks, respectively noted as B and M in the metrics tables.

4.1 Analysis of the YOLOv8-seg Nano Models

In this section, the evaluation metrics, described in section 2.3.3.2, from the YOLOv8-seg nano and medium models trained with the various multi-channel encoding strategies are shown.

All of the metrics resulting from the training, validation and testing of the YOLOv8-seg nano models are shown in appendix A. A model's training is considered to be successful when both the training and validation losses decrease with the number of epochs, while the evaluation metrics increase. Training can be considered complete when all of these stabilize, indicating that no further improvement is possible. When the loss functions of the training and validation datasets present the same behaviour, that is, decreasing and later stabilizing, it's an indication of no overfitting existing. By analyzing the evolution of the metrics per epoch, it's possible to conclude that the training converges for all configurations. The plots which relate the evaluation metrics with confidence can be used to characterize the model's ability to give a correct prediction with respect to confidence. In particular, the confidence threshold should be picked where the F1-score is higher

since this metric relates the effective precision of the model with respect to false positives and missed detections. The precision-recall curve, which is computed for an IoU of 0.50, also relates the model’s precision with respect to false positives. The area under the curve corresponds to the mAP50. These graphs are obtained from the validation dataset.

A concise summary of the evaluation metrics for the validation and testing datasets is shown in tables 4.2 and 4.2. By analysing the evaluation metrics of the segmentation masks, in particular the precision, recall, mAP50 and mAP50-95, it’s possible to compare the different models. For the sake of conciseness, each multi-channel encoding strategy is referred to by its configuration, as listed in table 3.3.

Table 4.1: Validation metrics of YOLOv8-seg nano models on multi-channel encoding configurations (MCEC).

MCEC	Precision (B)	Recall (B)	mAP50 (B)	mAP50-95 (B)	Precision (M)	Recall (M)	mAP50 (M)	mAP50-95 (M)
1	0.8431	0.7407	0.8421	0.4383	0.8379	0.7265	0.8171	0.3092
2	0.8625	0.7581	0.8609	0.4554	0.8453	0.7429	0.8289	0.3115
3	0.8543	0.7105	0.8103	0.3986	0.8580	0.6912	0.7855	0.2840
4	0.8554	0.7588	0.8529	0.4551	0.8494	0.7461	0.8338	0.3134
5	0.8588	0.7292	0.8429	0.4415	0.8624	0.7092	0.8171	0.3148
6	0.8592	0.7712	0.8661	0.4682	0.8513	0.7537	0.8369	0.3189
7	0.8401	0.7622	0.8584	0.4534	0.8647	0.7223	0.8317	0.3170
8	0.8347	0.7112	0.8201	0.4183	0.8375	0.6988	0.7958	0.2972
9	0.8793	0.7582	0.8677	0.4624	0.8636	0.7354	0.8288	0.3175
10	0.8508	0.7229	0.8294	0.4319	0.8456	0.7057	0.8009	0.3036
11	0.7768	0.6299	0.7405	0.3438	0.7539	0.5789	0.6670	0.2090
12	0.6534	0.6237	0.6851	0.3147	0.5701	0.5479	0.5502	0.1524

Table 4.2: Testing metrics of YOLOv8-seg nano models on multi-channel encoding configurations (MCEC).

MCEC	Precision (B)	Recall (B)	mAP50 (B)	mAP50-95 (B)	Precision (M)	Recall (M)	mAP50 (M)	mAP50-95 (M)
1	0.8729	0.7105	0.8451	0.4721	0.8940	0.7063	0.8100	0.3140
2	0.9279	0.6700	0.8396	0.4831	0.9219	0.6650	0.8135	0.3022
3	0.7709	0.6010	0.7256	0.3616	0.7726	0.5714	0.6715	0.2236
4	0.8350	0.7094	0.8394	0.4484	0.8172	0.6798	0.7828	0.2854
5	0.8582	0.6857	0.8414	0.4377	0.8369	0.6650	0.7856	0.2863
6	0.8934	0.7428	0.8684	0.4775	0.9136	0.7340	0.8382	0.3080
7	0.9279	0.6700	0.8396	0.4831	0.9219	0.6650	0.8135	0.3022
8	0.7906	0.6798	0.8091	0.4181	0.7757	0.6643	0.7561	0.2744
9	0.8713	0.7143	0.8622	0.4642	0.8546	0.6897	0.8214	0.3088
10	0.8442	0.6941	0.8132	0.4372	0.8283	0.6700	0.7623	0.2814
11	0.8031	0.6158	0.7384	0.3539	0.7806	0.5783	0.6721	0.2006
12	0.6397	0.6821	0.6661	0.3198	0.5793	0.5468	0.5296	0.1491

Notably, configuration 6 stands out due to its good performance both in the validation and testing data. This configuration achieves the highest results in many of the metrics and closely matches other configurations in the remaining metrics. Notably, its robustness to false positives, given by the recall, stands out, particularly in the testing dataset. The precision value, mAP50 and mAP50-95 are also some of the highest. For these reasons, multi-channel encoding configuration 6 was considered the best.

4.2 Evaluation Metrics From Trained YOLOv8-seg Medium Model

As previously mentioned, after analysing the results from the various multi-channel encoding strategies applied to the YOLOv8-seg nano models, the configuration which was considered to produce the best results, namely configuration 6, was used to train a YOLOv8-seg medium model. A medium model was chosen since it was considered that it provided a better performance relative to model size and speed when compared to the small, large and extra-large models, as referenced in figure 2.24. Due to the increase in training time, this model was trained only for 20 epochs, since this was sufficient to conclude training. Next, the evaluation metrics of these models are presented.

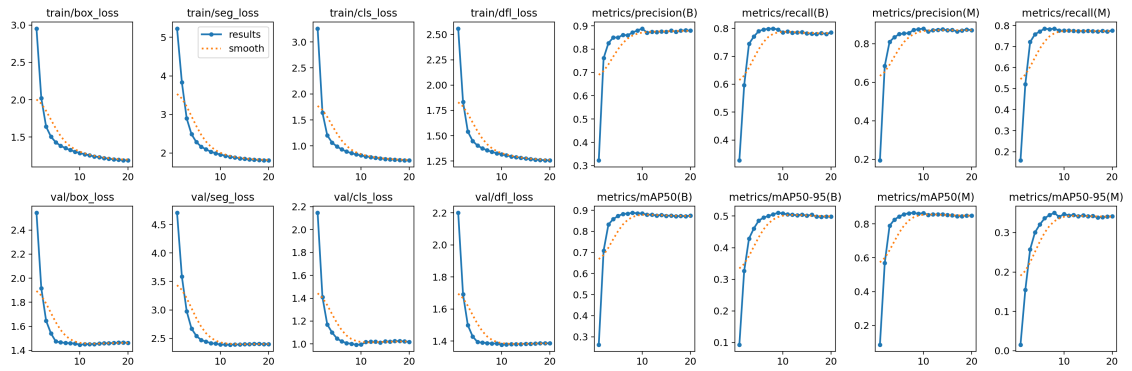


Figure 4.1: Training and validation results of YOLOv8-seg medium model on multi-channel encoding configuration 6.

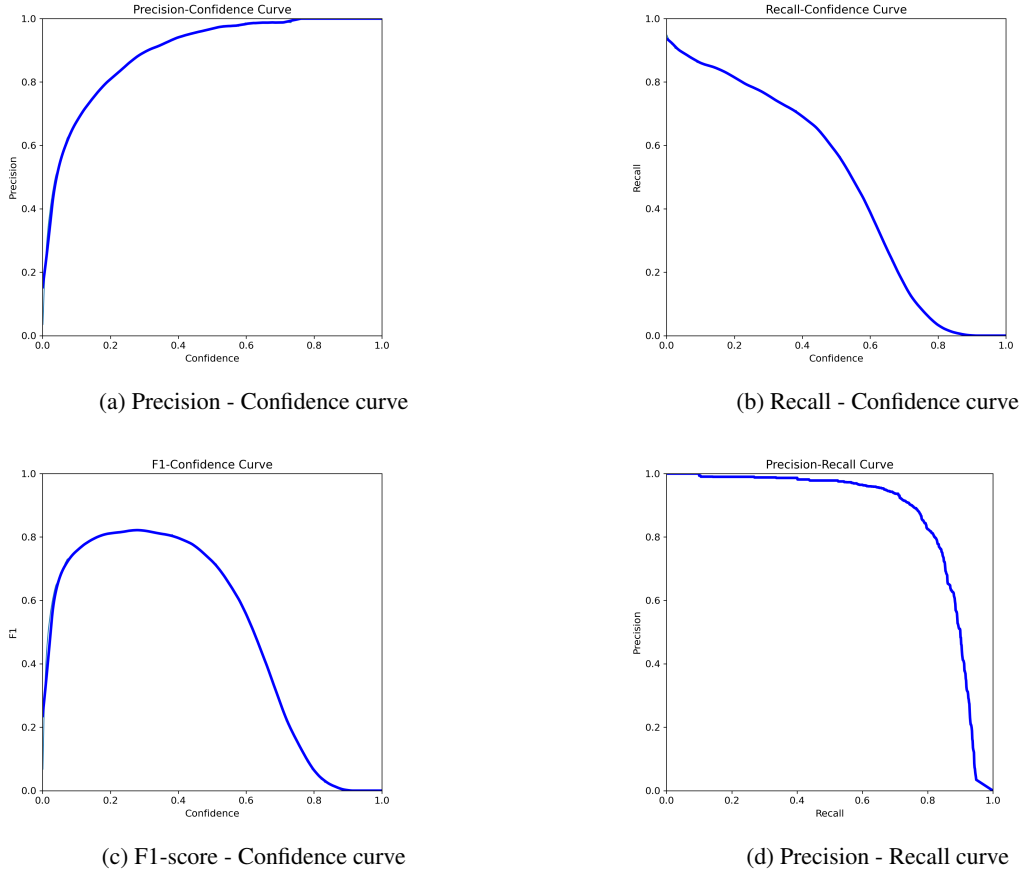


Figure 4.2: Validation metric curves of YOLOv8-seg medium model on multi-channel encoding configuration 6.

Table 4.3: Validation and testing metrics of YOLOv8-seg medium model on multi-channel encoding configuration 6.

Metric	Validation	Testing
Precision (B)	0.8903	0.8678
Recall (B)	0.7822	0.7980
mAP50 (B)	0.8853	0.8880
mAP50-95 (B)	0.5011	0.5138
Precision (M)	0.8801	0.8624
Recall (M)	0.7733	0.7931
mAP50 (M)	0.8617	0.8770
mAP50-95 (M)	0.3461	0.3339
Fitness	0.9372	0.9394

Like before, training was successful, with the value per epoch of the loss functions decreasing and the evaluation metrics increasing in both the training and testing datasets, as seen in figure 4.1. The values of the loss functions, for both the training and validation datasets, stabilize at the end of training, hence training was completed without overfitting.

Considering the increase in the recall, mAP50, mAP50-95 and fitness for both the validation and testing datasets, shown in table 4.3, when compared to the YOLOv8-seg nano model trained with the same multi-channel encoding configuration, this model was used in the final system. Considering the F1-score-confidence curve, a confidence threshold of 0.15, was used to accept a prediction. This rather low value was chosen since it was considered more detrimental to have missed detections rather than false positives.

4.3 Analysis of the Parameters Estimation

In order to evaluate the success of the whole system in estimating the scan's parameters, the system verification dataset was used. Due to the fact that the system essentially depends on correct hyperbola segmentation masks predicted by the YOLO network and the goal here is to evaluate the efficacy of the parameter estimation via the constrained least-squares fitting of the GPR hyperbola model, the verification dataset consisted of 6 scan configurations, each repeated 5 times, such that the system could be evaluated considering multiple runs. The ground-truth values are shown in table 4.4. The results of parameter estimation were obtained considering only the first-hyperbola and the hyperbola groups. They are shown in tables 4.5 and table 4.6, when predictions were obtained from a single prediction and multiple predictions from test-time augmentation, respectively. In figures 4.4-4.6 the predicted YOLO masks, skeletonization, hyperbola fitting and hyperbola groups are shown by one scan of each scan configuration, not considering test-time augmentation. The total time elapsed necessary to execute the system over the verification dataset was 101.45 and 219.39 seconds, without and with test time augmentation, such that the average time elapsed per scan was 3.38 and 7.31 seconds, respectively.

Table 4.4: Ground truth values of verification dataset for object's position, x_{GT} , depth, y_{GT} , and radius, r_{GT} .

Scan Config.	x_{GT} (cm)	y_{GT} (cm)	r_{GT} (cm)
1.1 - 1.5	51.0	7.6	1.4
2.1 - 2.5	48.0	7.6	1.4
3.1 - 3.5	50.0	7.4	0.5
4.1 - 4.5	49.2	7.4	0.5
5.1 - 5.5	49.8	7.0	1.5
6.1 - 6.5	49.2	7.0	1.5

Table 4.5: Estimated parameters for true-positive hyperbola detections. The number of true-positive, TP , false positive, FP and false negative, FN detections is shown and if the first-hyperbola was detected correctly. Medium's EM wave propagation velocity, v and object's position, x , depth, y and radius, r . Calculated considering only first-hyperbola, $1st$, and all hyperbolas from an object, $group$.

Scan Config.	TP	FP	FN	Detected 1st-hyperbola	v_{1st} (m/s)	v_{group} (m/s)	x_{1st} (cm)	x_{group} (cm)	y_{1st} (cm)	y_{group} (cm)	r_{1st} (cm)	r_{group} (cm)
1.1	3	0	0	Yes	247939319.307	203264759.402	48.8	49.2	9.3	7.6	0.6	1.4
1.2	3	1	0	Yes	245049288.061	201943684.354	49.0	49.5	8.5	7.0	0.86	2.1
1.3	3	0	0	Yes	204619975.091	194445334.989	47.4	49.1	7.6	7.2	1.0	2.1
1.4	3	0	0	Yes	237283004.012	205410911.989	49.2	49.7	8.4	7.2	0.6	2.2
1.5	3	0	0	Yes	223786707.401	196536532.165	47.7	48.4	7.9	7.0	1.3	2.3
2.1	3	1	0	Yes	211757434.248	193298173.048	47.4	47.6	9.3	8.5	1.1	1.5
2.2	3	0	1	No	213483508.192	172794304.371	48.1	48.8	12.6	10.2	-0.1	0.8
2.3	2	1	1	No	206539899.637	186225027.458	48.0	48.0	11.9	10.7	0.6	1.2
2.4	3	1	0	Yes	224400963.490	197701702.993	49.1	48.5	7.8	6.9	-0.5	1.2
2.5	3	1	0	Yes	30000005.776	222768987.613	49.9	49.5	10.9	8.1	-1.1	0.9
3.1	3	0	0	Yes	211757434.248	193298173.048	47.4	47.6	9.3	8.5	1.1	1.5
3.2	3	1	0	Yes	248731607.052	222777603.244	48.9	49.3	11.2	10.0	0.0	1.1
3.3	3	1	0	Yes	259407170.707	231413838.049	48.2	48.8	11.3	12.7	1.3	1.8
3.4	3	0	0	Yes	243979900.710	219205335.046	46.1	46.4	10.9	9.8	0.5	1.4
3.5	3	0	0	Yes	238578172.727	204221621.383	46.1	46.0	10.9	9.3	1.0	1.8
4.1	3	0	0	Yes	208575500.968	181314509.419	53.5	54.3	10.8	7.6	1.5	1.4
4.2	4	3	1	No	162322957.642	169425538.696	53.5	53.4	11.1	11.6	0.3	1.1
4.3	3	1	1	No	183375600.779	185028830.454	53.7	54.7	12.6	12.7	0.3	1.1
4.4	2	1	1	No	207985151.285	179778151.663	54.4	56.1	13.6	13.5	-0.2	0.5
4.5	3	2	1	No	169135583.023	165183533.222	57.4	57.2	12.2	11.9	0.8	0.6
5.1	3	0	0	Yes	236566368.410	204982740.405	46.6	46.6	8.7	8.9	1.5	1.7
5.2	3	0	0	Yes	226257113.404	214150643.499	50.3	50.4	9.5	9.0	1.5	1.8
5.3	3	0	0	Yes	244397620.814	217650012.142	46.3	46.5	10.6	9.5	1.2	1.8
5.4	3	0	0	Yes	241266027.447	215111584.201	45.1	45.7	10.5	9.4	0.7	1.7
5.5	3	0	0	Yes	240743588.274	214638835.140	46.6	46.3	10.4	9.3	1.4	2.1
6.1	3	1	0	Yes	210073761.716	180782665.501	51.5	52.3	8.8	7.6	1.8	1.6
6.2	3	1	0	Yes	216909385.146	187741122.844	50.4	51.4	8.9	7.8	1.8	1.9
6.3	3	3	0	Yes	230236462.433	194972490.793	53.1	54.8	9.7	8.2	1.6	2.1
6.4	3	1	0	Yes	230964719.972	205795090.487	51.5	52.6	9.8	8.7	1.3	1.9
6.5	3	2	0	Yes	228942541.755	198871140.484	51.7	52.5	9.6	8.3	1.7	1.8

Table 4.6: Estimated parameters for true-positive hyperbola detections, when test-time augmentation was used. The number of true-positive, TP , false positive, FP and false negative, FN detections is shown and if the first-hyperbola was detected correctly. Medium's EM wave propagation velocity, v and object's position, x , depth, y and radius, r . Calculated considering only first-hyperbola, $1st$, and all hyperbolas from an object, $group$.

Scan Config.	TP	FP	FN	Detected 1st-hyperbola	v_{1st} (m/s)	v_{group} (m/s)	x_{1st} (cm)	x_{group} (cm)	y_{1st} (cm)	y_{group} (cm)	r_{1st} (cm)	r_{group} (cm)
1.1	3	0	0	Yes	177043665.808	183297247.838	50.4	50.3	6.8	7.0	1.2	2.1
1.2	3	1	0	Yes	209887990.394	198309321.908	49.3	50.2	7.6	7.2	1.3	2.3
1.3	3	0	0	Yes	167156758.715	182060814.735	48.8	49.6	7.0	7.7	1.3	1.9
1.4	3	0	0	Yes	183641641.689	197516982.872	47.8	49.4	6.8	7.3	1.5	2.6
1.5	3	0	0	Yes	241639133.538	216479393.274	48.2	48.6	8.4	7.5	1.2	2.4
2.1	2	0	1	No	183405690.708	179176726.380	50.1	50.6	10.4	10.1	0.9	1.1
2.2	3	0	1	No	172446535.998	168607227.951	49.9	49.8	10.6	10.4	0.5	1.2
2.3	3	0	0	Yes	300000005.777	237170269.013	47.1	48.1	10.8	8.6	-0.6	1.2
2.4	3	1	0	Yes	270957152.954	198528714.113	49.3	48.9	9.6	7.0	-0.3	1.5
2.5	2	0	1	No	222904572.567	184121342.334	48.7	49.5	12.7	10.4	0.2	1.4
3.1	3	0	0	Yes	238921050.906	225509249.364	46.5	47.4	10.7	10.1	1.1	2.1
3.2	3	1	0	Yes	276658664.801	230098866.477	52.7	51.1	11.2	9.3	0.9	1.8
3.3	3	0	0	Yes	300000005.778	246289932.718	48.1	48.8	13.5	11.1	0.3	1.7
3.4	3	0	0	Yes	300000005.778	239130419.179	47.5	47.1	13.1	10.4	-0.3	1.1
3.5	3	0	0	Yes	231995339.209	213264128.129	46.2	45.8	10.6	9.7	1.4	2.1
4.1	3	0	0	Yes	300000005.778	216959640.647	54.5	54.7	12.3	8.9	-0.3	1.7
4.2	3	1	0	Yes	233402503.440	195365494.626	53.3	53.2	10.1	8.5	0.9	1.7
4.3	2	1	1	No	197497560.922	176897688.184	54.1	54.6	13.7	12.3	1.2	1.5
4.4	2	2	1	No	205758973.594	172792586.505	55.9	56.6	14.6	12.2	1.0	1.0
4.5	2	1	1	No	184013184.552	167848550.390	56.7	56.9	12.7	11.5	0.3	0.8
5.1	3	0	0	Yes	271444373.191	226431183.568	46.4	47.2	11.7	9.8	0.7	1.8
5.2	3	0	0	Yes	249067346.325	225337387.726	50.4	50.8	10.5	9.5	1.6	1.9
5.3	3	0	0	Yes	245996317.168	233142192.226	46.6	47.8	10.3	9.8	1.4	1.8
5.4	3	0	0	Yes	230581498.050	212343004.070	45.4	45.9	9.4	8.7	1.1	2.3
5.5	3	0	0	Yes	216858759.269	216233396.302	45.9	46.7	8.5	8.5	2.1	2.7
6.1	3	1	0	No	300000005.778	238339912.451	56.8	54.5	4.0	3.2	3.0	3.3
6.2	3	2	0	Yes	227846346.588	202004283.690	51.4	52.0	9.7	8.6	1.4	2.0
6.3	3	2	0	Yes	234645967.154	209472150.141	54.4	55.3	9.7	8.7	1.3	1.9
6.4	3	2	0	Yes	300000005.778	235660296.928	53.7	53.5	12.2	9.6	0.2	1.7
6.5	3	1	0	Yes	269041003.475	206123074.665	54.6	54.0	11.3	8.7	1.3	1.9

The absolute error for the estimated parameters is shown next, in tables 4.7 and 4.8. In order to better represent the estimation error, some considerations were taken, such that the estimated parameters are valid and the error properly indicated the accuracy of estimation. For the position error, all of the scans were considered when calculating, since at least one true-positive hyperbola was detected in all scans. For the depth error, only scans where the first-hyperbola was correctly labelled were considered. For the radius, the previous consideration was also used as well as the radius being greater than zero, since otherwise it would indicate that the hyperbola's width wasn't properly determined.

Table 4.7: Absolute error, Δ , of valid estimated parameters shown in table 4.5.

Scan config.	Δx_{1st} (cm)	Δx_{group} (cm)	Δy_{1st} (cm)	Δy_{group} (cm)	Δr_{1st} (cm)	Δr_{group} (cm)
1.1	2.2	1.8	1.7	0.0	0.8	0.0
1.2	2.0	1.5	0.9	0.6	0.5	0.7
1.3	3.6	1.9	0.0	0.4	0.4	0.7
1.4	1.8	1.3	0.8	0.4	0.8	0.8
1.5	3.3	2.6	0.3	0.6	0.1	0.9
2.1	0.6	0.4	1.7	0.9	0.3	0.1
2.2	0.1	0.8	-	-	-	-
2.3	0.0	0.0	-	-	-	-
2.4	1.1	0.5	0.2	0.7	-	0.2
2.5	1.9	1.5	3.3	0.5	-	0.5
3.1	2.6	2.4	1.9	1.1	0.6	1.0
3.2	1.1	0.7	3.8	2.6	0.5	0.6
3.3	1.8	1.2	3.9	5.3	0.8	1.3
3.4	3.9	3.6	3.5	2.4	0.0	0.9
3.5	3.9	4.0	3.5	1.9	0.5	1.3
4.1	4.3	5.1	3.4	0.2	1.0	0.9
4.2	4.3	4.2	-	-	-	-
4.3	4.5	5.5	-	-	-	-
4.4	5.2	6.9	-	-	-	-
4.5	8.2	8.0	-	-	-	-
5.1	3.2	3.2	1.7	1.9	0.0	0.2
5.2	0.5	0.6	2.5	2.0	0.0	0.3
5.3	3.5	3.3	3.6	2.5	0.3	0.3
5.4	4.7	4.1	3.5	2.4	0.8	0.2
5.5	3.2	3.5	3.4	2.3	0.1	0.6
6.1	2.3	3.1	1.8	0.6	0.3	0.1
6.2	1.2	2.2	1.9	0.8	0.3	0.4
6.3	3.9	5.6	2.7	1.2	0.1	0.6
6.4	2.3	3.4	2.8	1.7	0.2	0.4
6.5	2.5	3.3	2.6	1.3	0.2	0.3
Average	2.7	2.9	2.3	1.4	0.4	0.6

Table 4.8: Absolute error, Δ , of valid estimated parameters shown in table 4.6, when test-time augmentation was used.

Scan config.	Δx_{1st} (cm)	Δx_{group} (cm)	Δy_{1st} (cm)	Δy_{group} (cm)	Δr_{1st} (cm)	Δr_{group} (cm)
1.1	0.6	0.7	0.8	0.6	0.2	0.7
1.2	1.7	0.8	0.0	0.4	0.1	0.9
1.3	2.2	1.4	0.6	0.1	0.1	0.5
1.4	3.2	1.6	0.8	0.3	0.1	1.2
1.5	2.8	2.4	0.8	0.1	0.2	1.0
2.1	2.1	2.6	-	-	-	-
2.2	1.9	1.8	-	-	-	-
2.3	0.9	0.1	3.2	1.0	-	0.2
2.4	1.3	0.9	2.0	0.6	-	0.1
2.5	0.7	1.5	-	-	-	-
3.1	3.5	2.6	3.3	2.7	0.6	1.6
3.2	2.7	1.1	3.8	1.9	0.4	1.3
3.3	1.9	1.2	6.1	3.7	0.2	1.2
3.4	2.5	2.9	5.7	3.0	0.8	0.6
3.5	3.8	4.2	3.2	2.3	0.9	1.6
4.1	5.3	5.5	4.9	1.5	0.8	1.2
4.2	4.1	4.0	2.7	1.1	-	1.2
4.3	4.9	5.4	-	-	-	-
4.4	6.7	7.4	-	-	-	-
4.5	7.5	7.7	-	-	-	-
5.1	3.4	2.6	4.7	2.8	0.8	0.3
5.2	0.6	1.0	3.5	2.5	0.1	0.4
5.3	3.2	2.0	3.3	2.8	0.1	0.3
5.4	4.4	3.9	2.4	1.7	0.4	0.8
5.5	3.9	3.1	1.5	1.5	0.6	1.2
6.1	7.6	5.3	-	-	-	-
6.2	2.2	2.8	2.7	1.6	0.1	0.5
6.3	5.2	6.1	2.7	1.7	0.2	0.4
6.4	4.5	4.3	5.2	2.6	1.3	0.2
6.5	5.4	4.8	4.3	1.7	0.2	0.4
Average	3.4	3.1	3.0	1.7	0.4	0.8

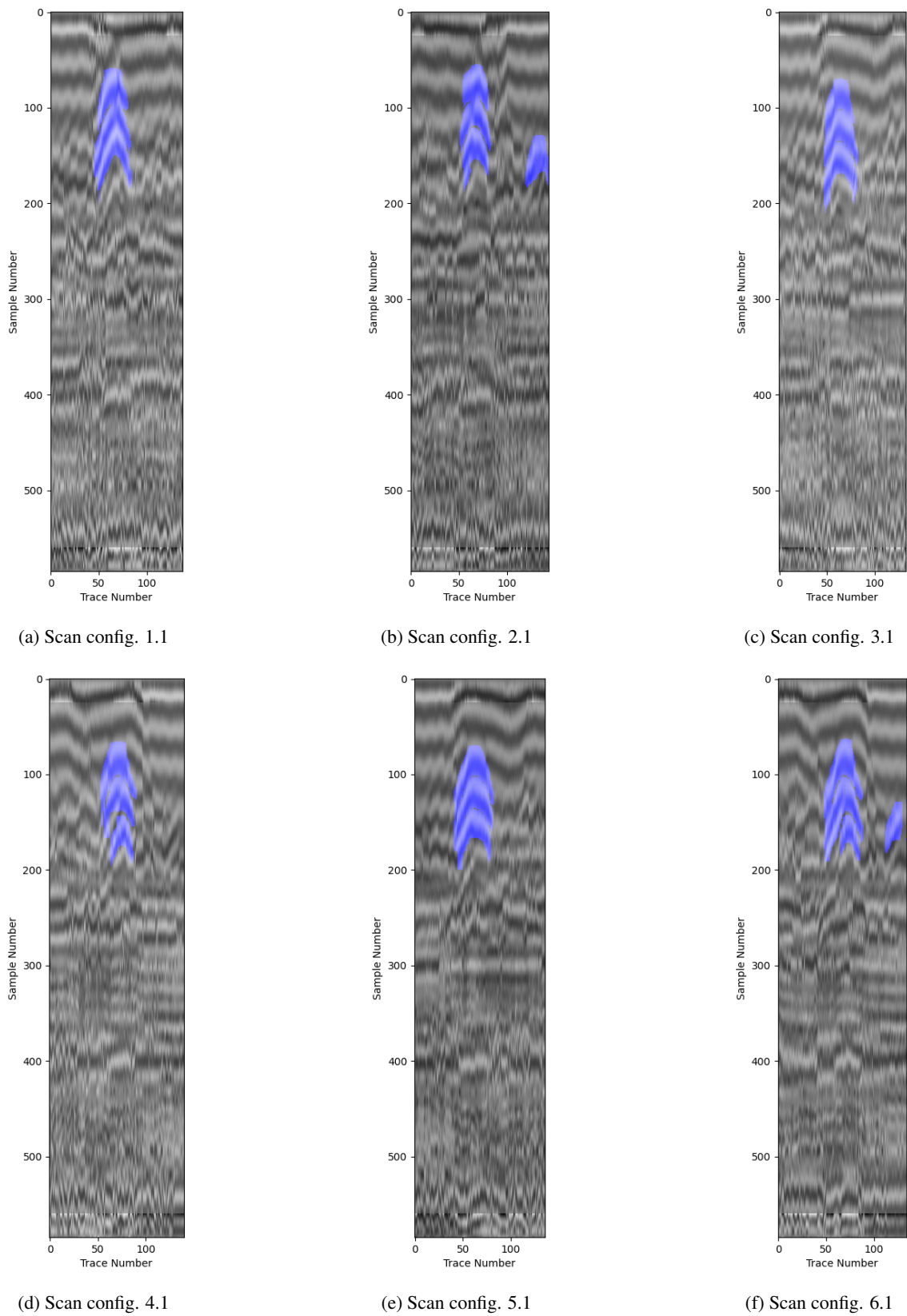
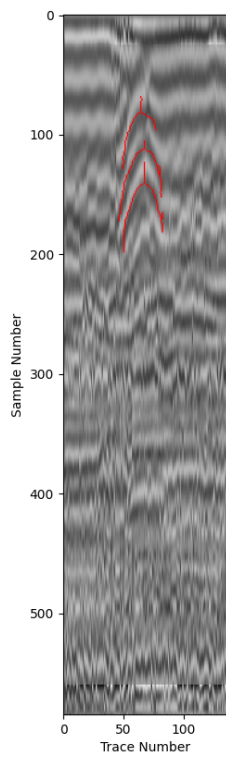
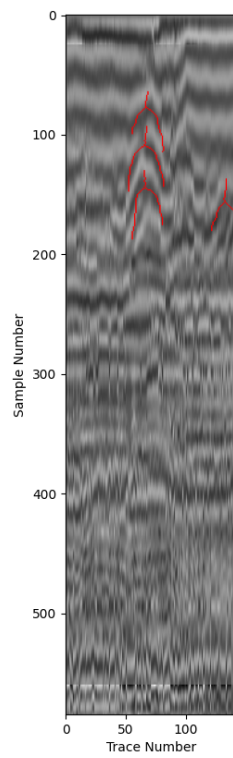


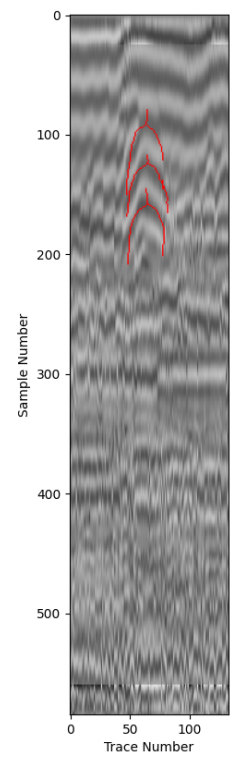
Figure 4.3: YOLO hyperbola mask predictions from trained model implemented in the system of scans from verification dataset. For visualization purposes, B-scans are shown after applying GPR-processing techniques and AGC.



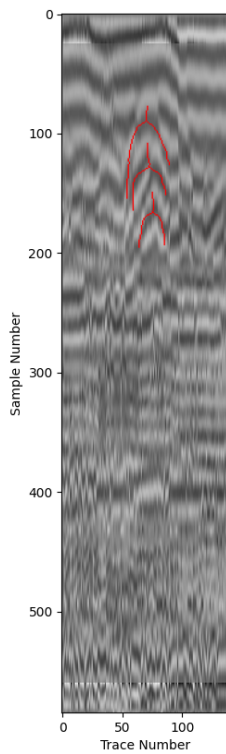
(a) Scan config. 1.1



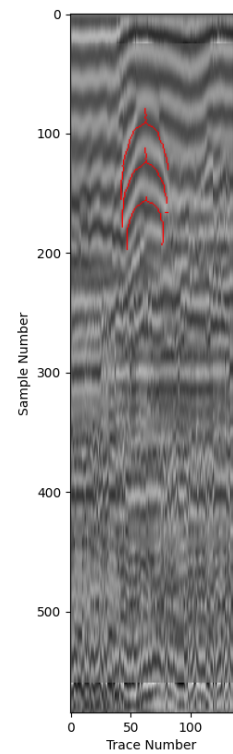
(b) Scan config. 2.1



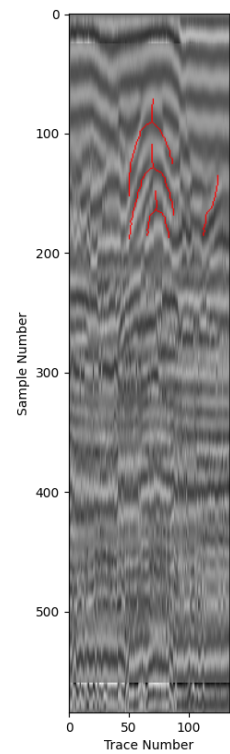
(c) Scan config. 3.1



(d) Scan config. 4.1

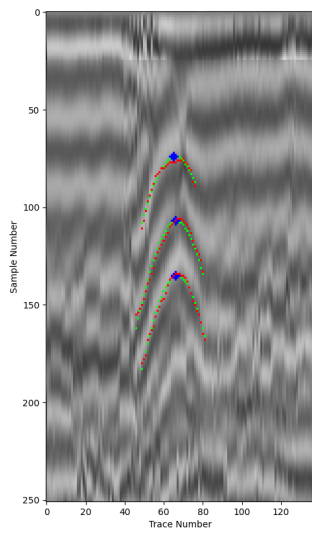


(e) Scan config. 5.1

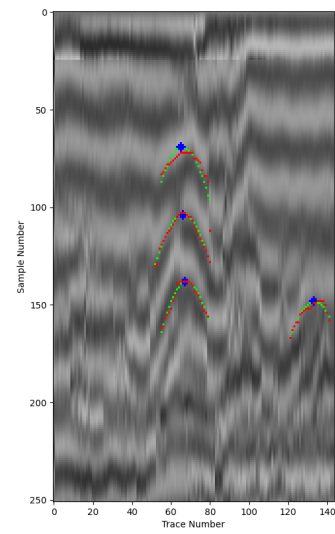


(f) Scan config. 6.1

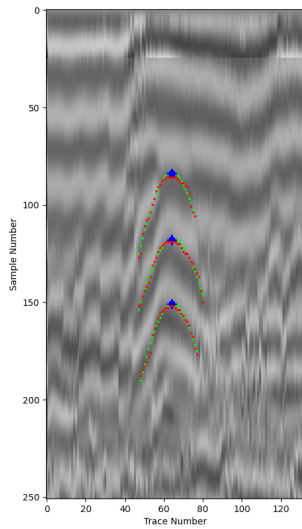
Figure 4.4: Hyperbola skeleton, applied to the B-scans shown in figure 4.4.



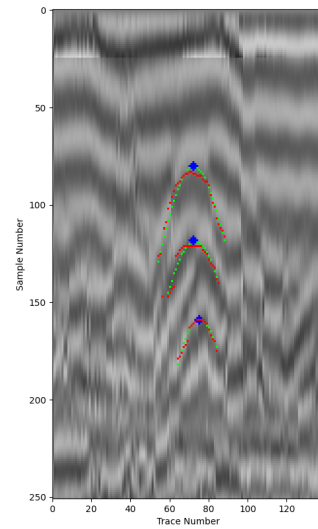
(a) Scan config. 1.1



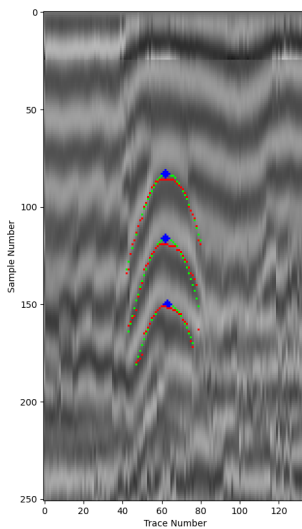
(b) Scan config. 2.1



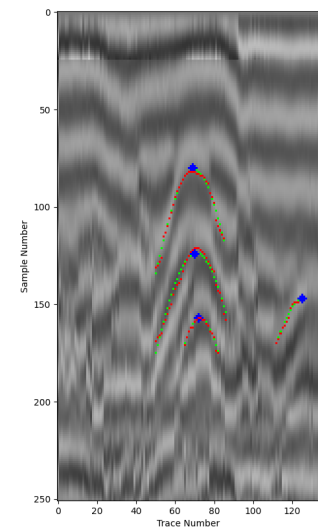
(c) Scan config. 3.1



(d) Scan config. 4.1

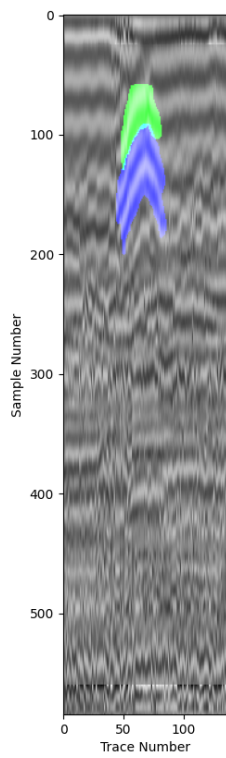


(e) Scan config. 5.1

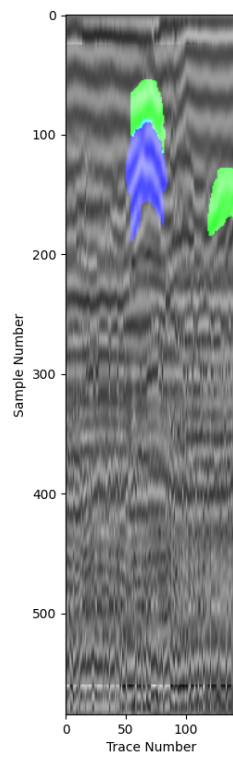


(f) Scan config. 6.1

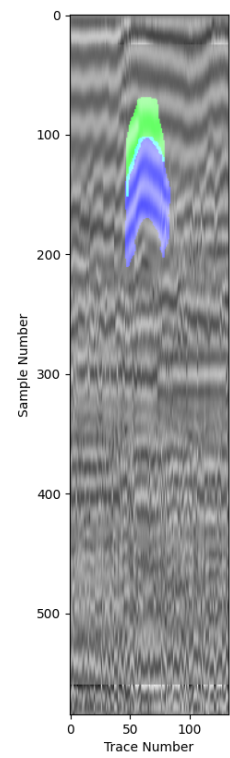
Figure 4.5: Constrained least-squares fitting of GPR hyperbola model, applied to the B-scans shown in figure 4.4. Fitting points are shown in red, fitted points are shown in green and the hyperbola apex is shown in blue.



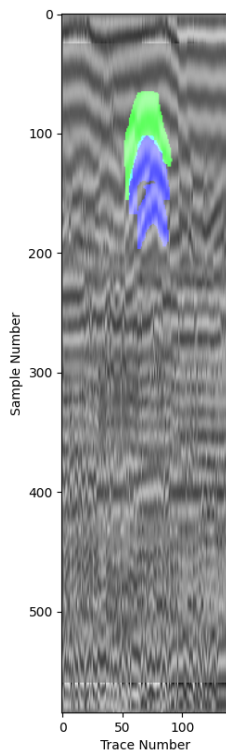
(a) Scan config. 1.1



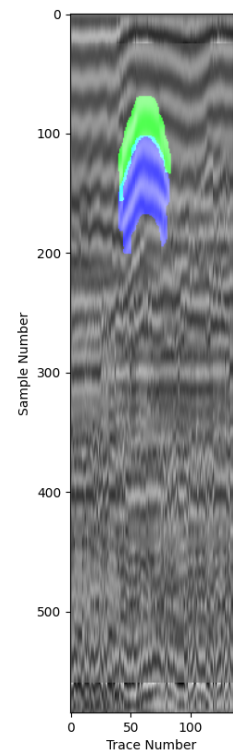
(b) Scan config. 2.1



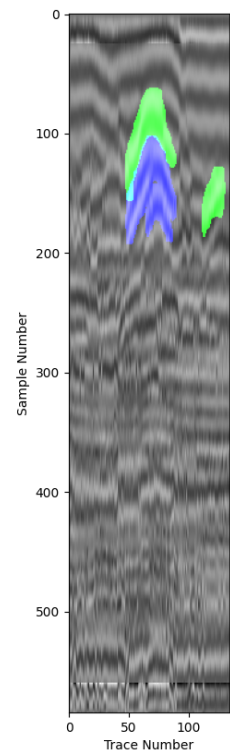
(c) Scan config. 3.1



(d) Scan config. 4.1



(e) Scan config. 5.1



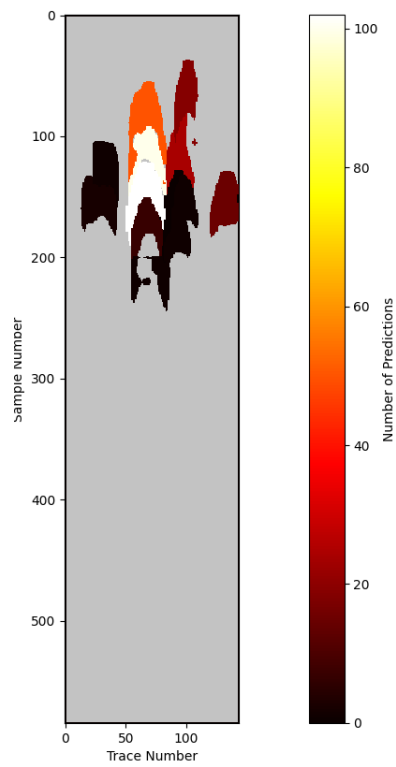
(f) Scan config. 6.1

Figure 4.6: Results from algorithm 2, applied to the B-scans shown in figure 4.4. First-hyperbola is shown in green and echos in blue.

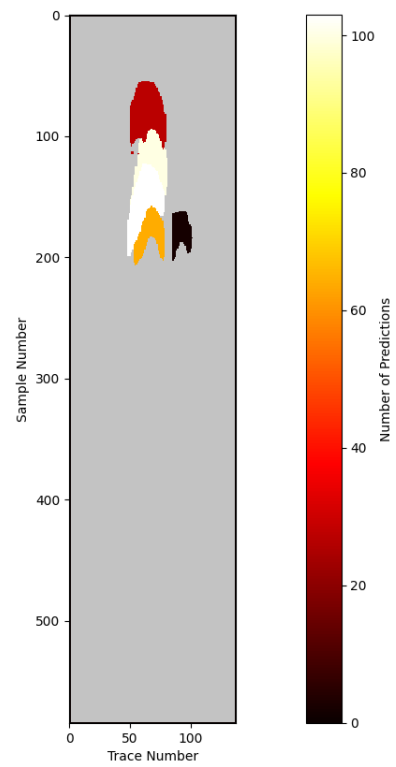
The success of the entire system essentially depends on the good performance of the YOLO model. When this is not the case, the accuracy of the estimated parameters will be affected, as characterized in the following points.

- In the case of false positive detections, an object will be identified where none exists and the parameters estimated will not be of interest.
- In the case of false negatives, a hyperbola will be missed. Due to the ringing effect, however, the repeated hyperbolas may still be identified, leading to an accurate position estimate, but inaccurate depth and radius estimate. When the first hyperbola isn't detected but echoes are, the object's depth will be exaggerated. The object's position will be accurate since the apex of repeated hyperbolas are aligned with the x position of the apex of the first-hyperbola. Additionally, the radius estimation will also be affected. Considering equation 2.15, since the calculated energy radius will increase due to the exaggerated depth, while the hyperbola width will remain equal to the missed first-hyperbola, since all hyperbola repetitions are equal. This will effectively invalidate the radius estimation.
- If the hyperbola mask doesn't effectively label the shape and inclination of the hyperbola branches, the estimation of propagation velocity will be affected. This in turn will result in inaccuracy in the estimated depth and radius.
- Additionally, at least a complete hyperbola branch is required for parameter estimation, as previously described in section 3.2.6.3. If the mask doesn't capture at least half of the hyperbola, that is one of the branches, the fitting may be rejected. If the fitting isn't rejected, but at least one full branch isn't captured, the estimation of radius will be affected when considering the hyperbola width.
- Multiple factors affect the estimation of radius, according to equation 2.15. Error in estimation velocity or hyperbola detection will affect the calculation of the energy radius, due to the effect on estimated depth. Additionally, an error in the value of the hyperbola's width will also affect the radius calculation.

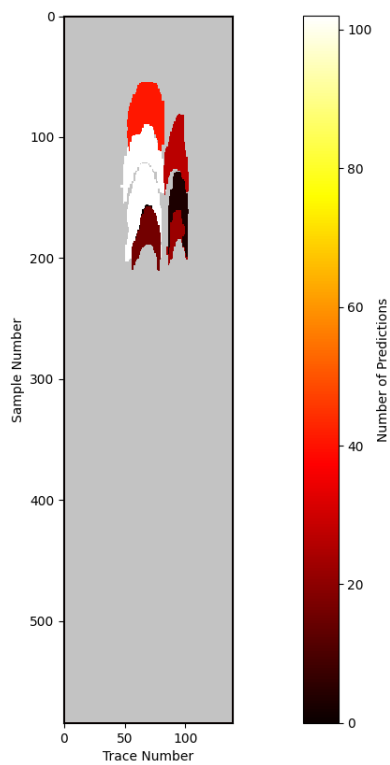
In order to try to mitigate the effect of poor YOLO predictions a prediction-averaging methodology based on test-time augmentation was explored. However, the results obtained were similar to when this technique was not used, as evidenced by the similar estimation error, shown in tables 4.7 and 4.8. Nevertheless, this technique has the potential to provide more accurate predictions concerning false positives and false negatives. The results from the predictions obtained using test-time augmentation are shown in figures 4.7 and 4.8, for the scan configurations where false positives existed as listed in table 4.6. As observed in these figures, false positive and false negative detections have a smaller number of detections when compared to true-positive detections. Consequently, this technique could be used to improve detection results, as it is able to identify false negative sections, however, it can be difficult to distinguish these from false positive detections.



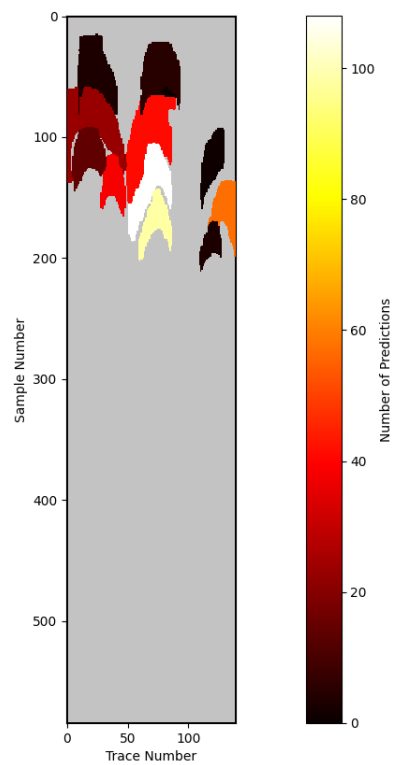
(a) Scan config. 2.1



(b) Scan config. 2.2



(c) Scan config. 2.5



(d) Scan config. 4.2

Figure 4.7: Number of YOLO hyperbola mask predictions considering the test-time augmentation approach, applied to scan configurations where false negatives existed, as listed in table 4.6. (1/2)

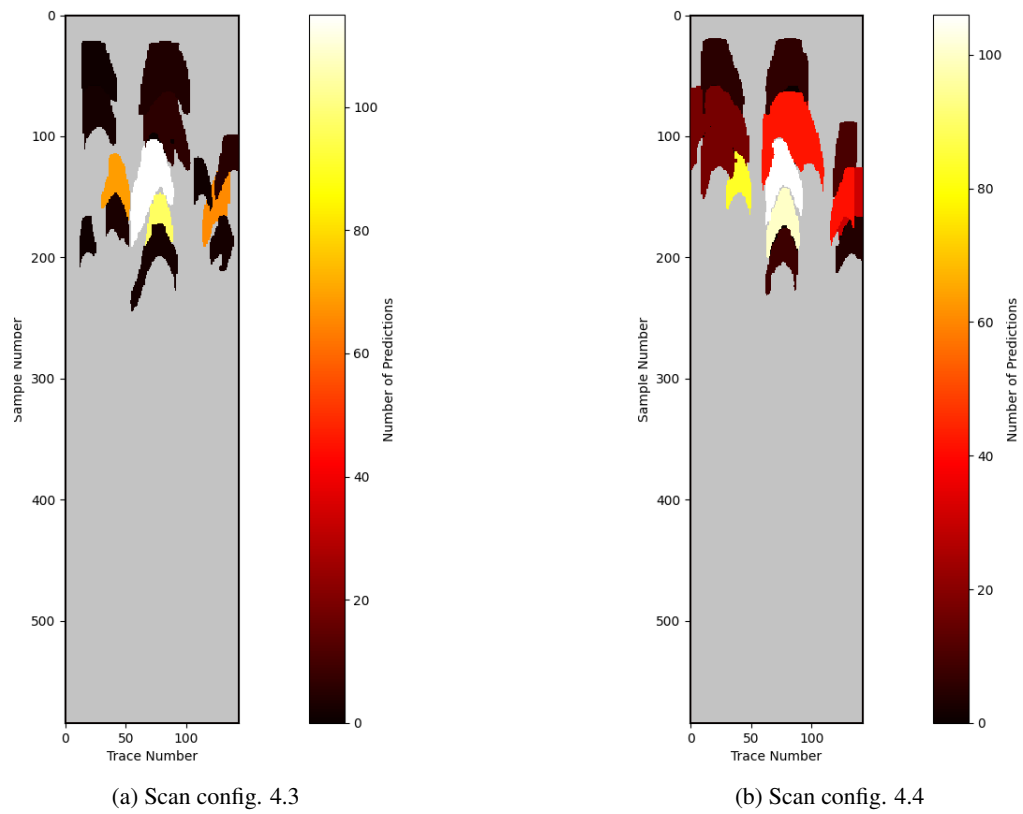


Figure 4.8: Number of YOLO hyperbola mask predictions predicted from 100 augmented images, applied to scan configurations where false negatives existed, as listed in table 4.6. (2/2)

4.4 Discussion

In the previous sections, the results obtained for both the training process of the YOLO models and the hyperbola fitting strategy were shown. The evaluation metrics from the trained models considering several multi-channel encoding configurations were compared. To achieve this, multiple YOLOv8-seg nano models were trained and the resulting best configuration was used to train a YOLOv8-seg medium model. The results for the nano models are summarized in tables 4.1 and 4.2, and shown in full detail in appendix A. By analyzing these results, the multi-channel configuration 6, composed of GPR processing, AGC and Sobel processing techniques, showed the best performance. More specifically, this model produced the best recall metric, making it the most robust to false negatives. Additionally, the other metrics concerning precision were also very high, when compared to the other models. Consequently, this configuration was then trained in a medium model. This model showed an improved performance when compared with the previous nano model when comparing the performance metrics, such that it was used when implementing the complete system.

The estimated scan's parameters were obtained using the verification dataset and compared with the ground-truth values for true-positive detections. These results are listed in tables 4.4, 4.5, 4.6, 4.7 and 4.8. As previously mentioned, these parameters were estimated considering only the first-hyperbola, and the hyperbola groups. In both cases, for the position and depth values, the results obtained showed relatively small errors, corroborating the applicability of the proposed fitting strategy for true-positive detections in real data. However, given the relatively small size of the objects' radii and the fact that this parameter is calculated considering other estimated parameters, which may be affected by the error, it can be concluded that radius estimation can be inaccurate for objects with small radii. A test-time augmentation approach was explored when obtaining the prediction masks from the YOLO model, however, the results from parameter estimation were similar to when this was not used. Despite this, this technique has the potential to reduce false positive and false negative detections, by improving the thresholding approach of the number of predictions to define a mask as true positive.

The whole fitting process is shown in figures 4.4-4.6. An inspection of the fitting results in figure 3.13 shows that the fitted points and hyperbola apex match the shape of the observed hyperbola. This confirms a good estimation of the fitting points, thus validating the fitting strategy. Finally, the algorithm used in order to identify hyperbola groups and the first-hyperbola worked as expected, as shown in figure 4.6.

In general, the parameters calculated considering the first hyperbola and its echos were more accurate for the estimation of the object's depth. This can be attributed to the fact that, by essentially considering multiple hyperbola repetitions, the effect of an ill-shaped hyperbola mask on the estimation of velocity is decreased. Thus the values of velocity, which are related to the slope of the hyperbola's asymptotes, are more accurate. For the other parameters, the accuracy is similar, when considering only the first-hyperbola and its echos. If the average of x position of the apexes and the width of all hyperbolas are less accurate, the parameters estimated will be affected.

Perhaps unexpectedly, greater errors occurred in the estimation of position. This can be attributed to inaccuracies in the intrinsic calibration of the encoder in the GPR system. The conversion constant was obtained from 10 one-meter-long samples and it was observed that the number of traces varied significantly. Additionally, position errors could also be caused due to inaccuracies in the experimental setup, namely, deviations in the scanning path and starting position.

The entire process, however, somewhat depends on the success of the trained YOLO model to correctly segment the hyperbolas. The consequences of this not occurring were also characterized. In short, false positive detections result in an object being identified where none exists and vice-versa in false negative detections. Here, it may still be possible to identify the object if the hyperbola echos are identified. This would result in an accurate estimation of position but exaggerated depth. The shape of the segmentation mask needs to correctly match the shape of the hyperbola in order to obtain an accurate value of velocity and hyperbola width. All of these parameters impact the estimation of radius, the most relevant being hyperbola width.

Chapter 5

Conclusions

This dissertation had the goal of developing a noise robust automated pipeline for GPR data analysis, which would allow the detection and characterization of buried objects. In order to achieve the developed system integrated GPR and computer vision processing techniques, for data processing, state of the art YOLOv8 segmentation models, for hyperbola detection, and a constrained least-squares GPR hyperbola model, for parameter estimation.

A dataset comprised of 294 GPR B-scans was collected and annotated, in order to train the neural networks. This dataset was split into training, validation and testing datasets, following an 80%, 10% and 10% splits, respectively. Traditional augmentation techniques were used to further increase the size of the training and validation dataset, due to the need to increase its size when training the neural networks, generating 8 images from an original image. The testing dataset remained un-augmented, such that it contained only real B-scans. A second smaller dataset was collected, using ground-truth information to evaluate the accuracy of the estimated parameters.

The GPR processing steps consisted of dewow, mean background removal and band-pass filtering were used, considering the characteristics of the GPR system. Additionally, AGC gain was used to reverse the effect of signal attenuation. Computer vision techniques were also explored, namely, template matching and Sobel edge detection filter. All of these techniques effectively have different benefits when considering the task of highlighting the hyperbola, in a scan affected by noise and cluttering signals. A multi-channel encoding strategy was used in order to combine several of these techniques into a single image, by storing in each RGB channel the B-scan after different processing were applied.

In order to determine which multi-channel configuration produced the best detection results, multiple YOLOv8-seg nano models were trained and their metrics were compared. This analysis concluded that the best multi-channel encoding configuration consisted of the GPR processing steps, AGC and Sobel filter. Next, a YOLOv8-seg medium model was trained with this configuration achieving improved results, more specifically for the validation and testing datasets, respectively, a precision of 0.8801 and 0.8624, recall of 0.7733 and 0.7931, mAP50 of 0.8617 and 0.7931, mAP50-95 of 0.3461 and 0.3339. This model is the one which was integrated into the full system to infer the segmentation masks of the hyperbolas. Two approaches were explored and

compared when using YOLO to predict the masks. First, the masks were obtained by considering only the results obtained from a single YOLO prediction. Later, a prediction averaging approach, based on test-time augmentation. In both cases the results were similar. However, prediction averaging could be used to reduce the number of false negative detections if the number of predictions for these hyperbolas is sufficiently distinguishable from false positive detections.

The verification dataset was used to evaluate the whole system, in particular, including the parameter estimation step. This step included the extraction of fitting points from the segmentation masks provided by the trained model, the constrained least squares fitting of the GPR hyperbola model and the calculation of the scan's parameters, which included the object's position, depth and radius and the EM wave propagation velocity in the medium. The parameters were estimated from two approaches, considering only the first hyperbola, and considering the first-hyperbola and its echos. The parameters estimated from the true-positive hyperbola detections, considering both approaches, were compared to their ground-truth. It was concluded that the approach which considered the echos resulted in better depth estimations, since the effect of ill-shaped hyperbolas on velocity estimation was reduced. For the position and radius parameters, similar results were obtained in both approaches. The average absolute errors calculated for the object's position, depth and radius were 2.7 cm and 2.9 cm, 2.3 cm and 1.4 cm, and 0.4 cm and 0.6 cm, respectively, considering the parameter estimation approaches with and without taking into account the hyperbola echos, both without prediction averaging. Using prediction averaging, these results corresponded to, 3.4 cm and 3.1 cm, 3.0 cm and 1.7 cm, 0.4 cm and 0.8 cm, respectively. These results indicate that parameter estimation was successful for position and depth. The accuracy in depth also indicates an accurate velocity estimation, which in turn indicates that the shape of the masks correctly models the slopes of the hyperbola branches. However, when considering the relatively small radius of the objects, the radius estimate is relatively inaccurate. Additionally, in cases where the segmentation masks aren't correctly detected or misrepresent the shape of the hyperbola, parameter estimation is adversely affected, resulting in errors in parameter estimation.

In conclusion, a system capable of analyzing GPR data in order to detect and characterise buried objects was successfully implemented. YOLOv8 models were trained following several multi-channel encoding configurations, which combined GPR and computer vision image processing. Configurations where complementing techniques were combined showed better results. Parameter estimation was achieved by following a constrained least squares fitting strategy of the mathematical GPR hyperbola model, proving accurate when the hyperbolas were correctly segmented.

5.1 Future Work

As previously explained, the developed system can be improved. The following points identify the future work which could lead to improved performance.

In particular, the performance of hyperbola segmentation could be increased. Firstly, the collection of a larger dataset with a broader range of buried objects would be beneficial in an approach

based on neural networks for hyperbola detection. Additionally, exploring and comparing other types of neural networks, as well as modifying existing networks or developing a specialized network would also be of interest. Network hyperparameter tuning should also be explored in order to determine the best possible parameters used for the training process. Improving image augmentation by considering other techniques, such as GANs, would also help improve the training by generating more realistic synthetic data. Finally, further exploring more multi-channel encoding configurations with other processing techniques would be interesting, such that the most relevant information would be passed to the networks. These improvements would allow for the training of a more accurate model, which would increase the accuracy of the whole system, by providing more accurate segmentation masks.

Developing a system to geo-reference the collected GPR data would also be relevant. This would effectively mitigate the error of position estimation considering our GPR's encoder and allow for the 3D mapping of the subsurface. The analysis of 3D GPR data could also lead to other object detection approaches.

Lastly, the integration of the automated hyperbola detection pipeline in a program with a user interface would allow for better visualization and verification of the results. This could be done in such a way that the predicted segmentation masks could be removed, adjusted or manually drawn by an expert, effectively eliminating the problem of ill-shaped masks, false positives or missed detections. This kind of strategy would thus reduce the time needed for data analysis while also producing verified results.

References

- [1] “United nations sustainable development goals”. (2015), [Online]. Available: <https://sdgs.un.org/goals> (visited on 07/31/2024).
- [2] R. Persico, *Introduction to Ground Penetrating Radar: Inverse Scattering and Data Processing*. Hoboken, New Jersey: John Wiley and Sons, Inc., 2014.
- [3] E. C. Utsi, *Ground Penetrating Radar: Theory and Practice*. Oxford, United Kingdom: Butterworth-Heinemann, 2017.
- [4] H. M. Jol, *Ground Penetrating Radar Theory and Applications*. Amsterdam, Netherlands: Elsevier Science, 2009.
- [5] D. J. Daniels, *Ground Penetrating Radar, 2nd Edition*. London, United Kingdom: Institution of Electrical Engineers, 2004.
- [6] A. P. Annan, *Ground Penetrating Radar principles, Procedures and Applications*. Mississauga, Canada: Sensors and Software, 2004.
- [7] C. Warren, “Numerical modelling of high-frequency ground-penetrating radar antennas”, Ph.D. dissertation, University of Edinburgh, 2009. [Online]. Available: <https://era.ed.ac.uk/handle/1842/4074> (visited on 12/26/2023).
- [8] C. W. Chang, C. H. Lin, and H. S. Lien, “Measurement radius of reinforcing steel bar in concrete using digital image GPR”, *Construction and Building Materials*, vol. 23, no. 2, pp. 1057–1063, Feb. 2009. DOI: [10.1016/j.conbuildmat.2008.05.018](https://doi.org/10.1016/j.conbuildmat.2008.05.018). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950061808001463?via%3Dihub> (visited on 07/31/2024).
- [9] K. A. Lakshmi and A. Rahamath, “Estimation of rebar diameter using ground penetrating radar”, *International Journal of Advances in Scientific Research and Engineering*, vol. 3, no. 1, pp. 370–375, Aug. 2017. DOI: [10.31695/IJASRE](https://doi.org/10.31695/IJASRE). [Online]. Available: <https://ijasre.net/index.php/ijasre/article/view/933> (visited on 07/31/2024).
- [10] M. M. Rocha, “Object detection in GPR data using classical vision and YOLOv5”, M.S. thesis, Faculdade de Engenharia da Universidade do Porto, 2022.

- [11] J. Daniels, “Ground penetrating radar fundamentals”, *Prepared as an appendix to a Report to the US EPA, Region V*, pp. 1–21, Nov. 2000. DOI: 10.4133/1.2921864. [Online]. Available: https://www.researchgate.net/publication/237508286_Ground_Penetrating_Radar_Fundamentals (visited on 07/31/2024).
- [12] D. Merkle, C. Frey, and A. Reiterer, “Fusion of ground penetrating radar and laser scanning for infrastructure mapping”, *Journal of Applied Geodesy*, vol. 15, no. 1, pp. 31–45, Jan. 2020. DOI: 10.1515/jag-2020-0004. [Online]. Available: <https://www.degruyter.com/document/doi/10.1515/jag-2020-0004/html> (visited on 07/31/2024).
- [13] R. Yurt, H. Torpi, A. Kizilay, S. Koziel, A. Pietrenko-Dabrowska, and P. Mahouti, “Buried object characterization by data-driven surrogates and regression-enabled hyperbolic signature extraction”, *Scientific Reports*, vol. 13, no. 1, p. 5717, Apr. 2023. DOI: 10.1038/s41598-023-32925-6. [Online]. Available: <https://www.nature.com/articles/s41598-023-32925-6> (visited on 07/31/2024).
- [14] U. Özkaya, Ş. Öztürk, and L. Seyfi, “Region analysis of buried object based on fully automated image processing”, *International Conference on Advances in Science and Arts*, Mar. 2017. [Online]. Available: <https://www.researchgate.net/publication/343548181> (visited on 07/31/2024).
- [15] A. Ristic, D. Petrovacki, and M. Govedarica, “A new method to simultaneously estimate the radius of a cylindrical object and the wave propagation velocity from GPR data”, *Computers Geosciences*, vol. 35, no. 8, pp. 1620–1630, Aug. 2009. DOI: 10.1016/j.cageo.2009.01.003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098300409000661> (visited on 07/31/2024).
- [16] H. Chen and A. Cohn, “Probabilistic robust hyperbola mixture model for interpreting ground penetrating radar data”, *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8, Jul. 2010. DOI: 10.1109/IJCNN.2010.5596298. [Online]. Available: https://www.researchgate.net/publication/221534578_Probabilistic_robust_hyperbola_mixture_model_for_interpreting_ground_penetrating_radar_data (visited on 07/31/2024).
- [17] S. Shihab and W. Al-Nuaimy, “Radius estimation for cylindrical objects detected by ground penetrating radar”, *Subsurface Sensing Technologies and Applications*, vol. 6, no. 2, pp. 151–166, Apr. 2005. DOI: 10.1007/s11220-005-0004-1. [Online]. Available: <https://link.springer.com/article/10.1007/s11220-005-0004-1> (visited on 07/31/2024).
- [18] Q. Dai, Y. Li, T. Jiang, and Q. Sun, “A new b-scan interpretation model in complex environments”, *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, Oct. 2022. DOI: 10.1109/TIM.2022.3214294. (visited on 07/31/2024).

- [19] X. Zhou, H. Chen, and J. Li, “An automatic GPR b-scan image interpreting model”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3398–3412, Jun. 2018. DOI: 10.1109/TGRS.2018.2799586. [Online]. Available: <https://ieeexplore.ieee.org/document/8301009> (visited on 07/24/2024).
- [20] L. Mertens, R. Persico, L. Matera, and S. Lambot, “Automated detection of reflection hyperbolas in complex GPR images with no a priori knowledge on the medium”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 580–596, Aug. 2015. DOI: 10.1109/TGRS.2015.2462727. [Online]. Available: <https://ieeexplore.ieee.org/document/7230274> (visited on 07/31/2024).
- [21] I. Giannakis, F. Zhou, C. Warren, and A. Giannopoulos, “On the limitations of hyperbola fitting for estimating the radius of cylindrical targets in nondestructive testing and utility detection”, *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Aug. 2022. DOI: 10.1109/LGRS.2022.3195947. [Online]. Available: <https://ieeexplore.ieee.org/document/9847240> (visited on 07/31/2024).
- [22] D. Kumlu and I. Erer, “Gpr clutter reduction by robust orthonormal subspace learning”, *IEEE Access*, vol. 8, pp. 74 145–74 156, Apr. 2020. DOI: 10.1109/ACCESS.2020.2988333. [Online]. Available: <https://ieeexplore.ieee.org/document/9069272> (visited on 07/31/2024).
- [23] G. Chen, L. Fu, K. Chen, C. D. Boateng, and S. Ge, “Adaptive ground clutter reduction in ground-penetrating radar data based on principal component analysis”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3271–3282, Jun. 2019. DOI: 10.1109/TGRS.2018.2882912. [Online]. Available: <https://ieeexplore.ieee.org/document/8604012> (visited on 07/31/2024).
- [24] G. Yang, D. Yuan, T. Xu, and B. Li, “An adaptive clutter-immune method for pipeline detection with GPR”, *IEEE Sensors Journal*, vol. 23, no. 19, pp. 22 984–22 997, Aug. 2023. DOI: 10.1109/JSEN.2023.3305681. [Online]. Available: <https://ieeexplore.ieee.org/document/10225652> (visited on 07/31/2024).
- [25] J. Kuruvilla, D. Sukumaran, A. Sankar, and S. P. Joy, “A review on image processing and image segmentation”, *International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pp. 198–203, Mar. 2016. DOI: 10.1109/sapience.2016.7684170. [Online]. Available: <https://ieeexplore.ieee.org/document/7684170> (visited on 07/31/2024).
- [26] R. Szeliski, *Computer vision: Algorithms and applications*. Cham, Switzerland: Springer, 2022.
- [27] D. Forsyth, J. Ponce, S. Mukherjee, and A. K. Bhattacharjee, *Computer vision: A modern approach*. Uttar Pradesh, India: Pearson India Education Services Pvt. Ltd., 2015.

- [28] S. Savant, “A review on edge detection techniques for image segmentation”, *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5898–5900, Aug. 2014. [Online]. Available: https://www.researchgate.net/publication/291975278_A_Review_on_Edge_Detection_Techniques_for_Image_Segmentation (visited on 07/31/2024).
- [29] H. Spontón and J. Cardelino, “A review of classic edge detectors”, *Image Processing On Line*, vol. 5, pp. 90–123, Jun. 2015. DOI: 10.5201/ipol.2015.35. [Online]. Available: https://www.researchgate.net/publication/281392911_AReview_of_Classic_Edge_Detectors (visited on 07/31/2024).
- [30] A. Di Donato, M. Farina, A. Morini, *et al.*, “Using correlation maps in a wide-band microwave GPR”, *Progress In Electromagnetics Research B*, vol. 30, pp. 371–387, Jan. 2011. DOI: 10.2528/PIERB11050304. [Online]. Available: https://www.researchgate.net/publication/271722537_Using_Correlation_Maps_in_a_Wide-Band_Microwave_GPR (visited on 07/31/2024).
- [31] K. Feng, Y. Zhao, J. Wu, and S. Ge, “Cross-correlation attribute analysis of GPR data for tunnel engineering”, pp. 435–440, Dec. 2014. DOI: 10.1109/ICGPR.2014.6970461. [Online]. Available: <https://ieeexplore.ieee.org/document/6970461> (visited on 07/31/2024).
- [32] A. R. Parmezan, V. M. Souza, and G. E. Batista, “Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model”, *Information Sciences*, vol. 484, pp. 302–337, 2019. DOI: 10.1016/j.ins.2019.01.076. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025519300945> (visited on 07/31/2024).
- [33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, United Kingdom: MIT Press, 2016.
- [34] L. Rokach, O. Maimon, and E. Shmueli, *Machine Learning for Data Science Handbook Data Mining and Knowledge Discovery Handbook*. Springer, 2023.
- [35] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning”, *Journal of Big Data*, vol. 6, pp. 1–48, Jan. 2019. DOI: 10.1186/s40537-019-0197-0. [Online]. Available: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0#citeas> (visited on 07/31/2024).
- [36] E. Zvornicanin and G. Piwowarek. “How to use k-fold cross-validation in a neural network?” (Mar. 2024), [Online]. Available: <https://www.baeldung.com/cs/k-fold-cross-validation> (visited on 07/31/2024).
- [37] Ultralytics. “Performance metrics deep dive”. (2024), [Online]. Available: <https://docs.ultralytics.com/guides/yolo-performance-metrics/#how-to-calculate-metrics-for-yolov8-model> (visited on 07/31/2024).

- [38] N. Shahriar. “What is Convolutional Neural Network - CNN (Deep Learning)”. (Feb. 2023), [Online]. Available: <https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5> (visited on 07/31/2024).
- [39] X. Zheng, S. Fang, H. Chen, L. Peng, and Z. Ye, “Internal detection of ground-penetrating radar images using yolox-s with modified backbone”, *Electronics*, vol. 12, no. 16, p. 3520, Aug. 2023. DOI: 10.3390/electronics12163520. [Online]. Available: <https://doi.org/10.3390/electronics12163520> (visited on 07/31/2024).
- [40] X. Bai, Y. Yang, S. Wei, *et al.*, “A comprehensive review of conventional and deep learning approaches for ground-penetrating radar detection of raw data”, *Applied Sciences*, vol. 13, no. 13, p. 7992, Jul. 2023. DOI: 10.3390/app13137992. [Online]. Available: <https://www.mdpi.com/2076-3417/13/13/7992> (visited on 07/31/2024).
- [41] R. Gandhi. “R-cnn, fast r-cnn, faster r-cnn, yolo - object detection algorithms”. (Jul. 2018), [Online]. Available: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> (visited on 07/31/2024).
- [42] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. DOI: <https://doi.org/10.48550/arXiv.1807.05511>. [Online]. Available: <https://arxiv.org/abs/1807.05511> (visited on 07/31/2024).
- [43] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas”, *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023. DOI: 10.3390/make5040083. [Online]. Available: <https://doi.org/10.3390/make5040083> (visited on 07/31/2024).
- [44] M. Sohan, T. Ram, and V. Ch, “A review on yolov8 and its advancements”, *Data Intelligence and Cognitive Informatics*, pp. 529–545, Jan. 2024. DOI: 10.1007/978-981-99-7962-2_39. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-99-7962-2_39#citeas (visited on 07/31/2024).
- [45] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics yolov8*, version 8.0.0, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [46] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning”, *Journal of Big Data*, vol. 6, no. 1, 2019. DOI: 10.1186/s40537-019-0197-0. [Online]. Available: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0> (visited on 07/31/2024).
- [47] Z.-K. Ni, S. Ye, C. Shi, C. Li, and G. Fang, “Clutter suppression in GPR B-scan images using robust autoencoder”, *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Oct. 2022. DOI: 10.1109/lgrs.2020.3026007. [Online]. Available: <https://ieeexplore.ieee.org/document/9211492> (visited on 07/31/2024).

- [48] J. Luo, W. Lei, F. Hou, *et al.*, “GPR B-scan image denoising via multi-scale convolutional autoencoder with data augmentation”, *Electronics*, vol. 10, no. 11, p. 1269, 2021. DOI: 10.3390/electronics10111269. [Online]. Available: <https://www.mdpi.com/2079-9292/10/11/1269> (visited on 07/31/2024).
- [49] P. Bestagini, F. Lombardi, M. Lualdi, F. Picetti, and S. Tubaro, “Landmine detection using autoencoders on multipolarization GPR volumetric data”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 182–195, May 2020. DOI: 10.1109/tgrs.2020.2984951. [Online]. Available: <https://ieeexplore.ieee.org/document/9103952> (visited on 07/31/2024).
- [50] Z.-K. Ni, C. Shi, J. Pan, Z. Zheng, S. Ye, and G. Fang, “Declutter-GAN: GPR b-scan data clutter removal using conditional generative adversarial nets”, *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Mar. 2022. DOI: 10.1109/LGRS.2022.3159788. [Online]. Available: <https://ieeexplore.ieee.org/document/9736999> (visited on 07/31/2024).
- [51] OERAD, *Oerad concreto technical specifications*, 2024. [Online]. Available: <https://www.oerad.eu/concreto> (visited on 07/31/2024).
- [52] CVAT.ai Corporation, *Computer vision annotation tool (cvat)*, Jul. 2024. DOI: 10.5281/zenodo.12771595. [Online]. Available: <https://github.com/cvat-ai/cvat> (visited on 07/31/2024).
- [53] A. Tareq, M. I. Faisal, M. S. Islam, *et al.*, “Visual diagnostics of dental caries through deep learning of non-standardised photographs using a hybrid yolo ensemble and transfer learning model”, *International Journal of Environmental Research and Public Health*, vol. 20, no. 7, p. 5351, Mar. 2023. DOI: 10.3390/ijerph20075351. [Online]. Available: https://www.researchgate.net/publication/369717297_Visual_Diagnostics_of_Dental_Caries_through_Deep_Learning_of_Non-Standardised_Photosgraphs_Using_a_Hybrid_YOLO_Ensemble_and_Transfer_Learning_Model (visited on 07/31/2024).
- [54] W. E. Hart, J.-P. Watson, and D. L. Woodruff, “Pyomo: Modeling and solving mathematical programs in python”, *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, Aug. 2011. DOI: 10.1007/s12532-011-0026-8. [Online]. Available: <https://link.springer.com/article/10.1007/s12532-011-0026-8> (visited on 07/31/2024).
- [55] A. Wächter and L. T. Biegler, “On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming”, *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Apr. 2006. DOI: 10.1007/s10107-004-0559-y. [Online]. Available: <https://link.springer.com/article/10.1007/s10107-004-0559-y> (visited on 07/31/2024).

Appendix A

Evaluation Metrics of YOLOv8 Nano Models Trained with Multi-channel Encoding Configurations

In this section, the evaluation metrics obtained from the various multi-channel encoding configurations used to train the YOLOv8-seg nano models are listed. These consist of the graphs used to interpret the training performance and validation evaluation metrics. The evaluation metrics computed when applied to the validation and testing datasets were shown in tables 4.1 and 4.2. Each multi-channel encoding configuration is listed in table 3.3.

A.1 Multi-Channel Encoding Configuration 1

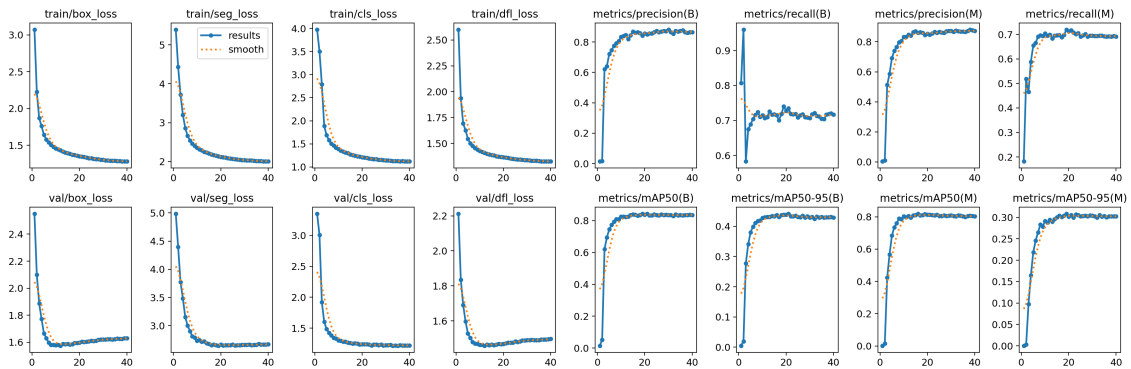
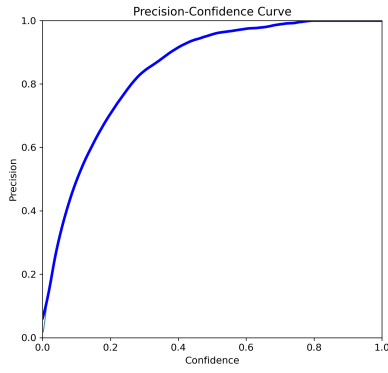
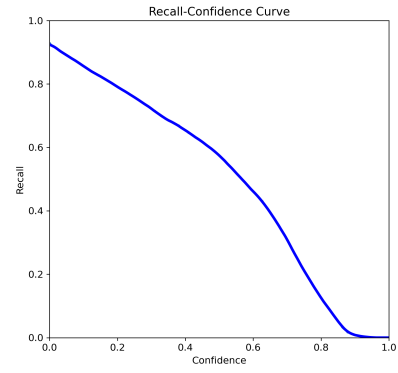


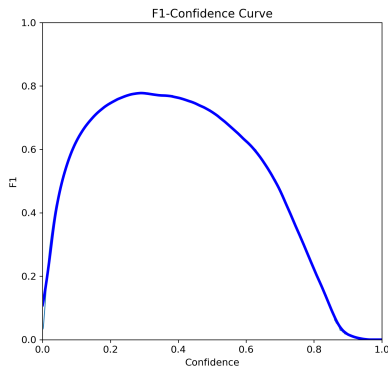
Figure A.1: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 1.



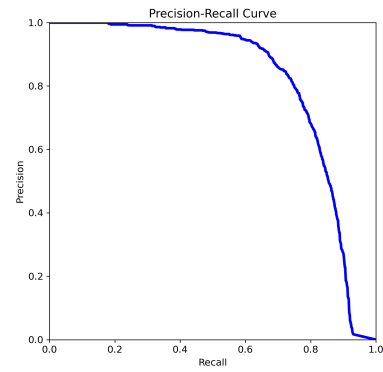
(a) Precision - Confidence curve



(b) Recall - Confidence curve



(c) F1-score - Confidence curve



(d) Precision - Recall curve

Figure A.2: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 1.

A.2 Multi-Channel Encoding Configuration 2

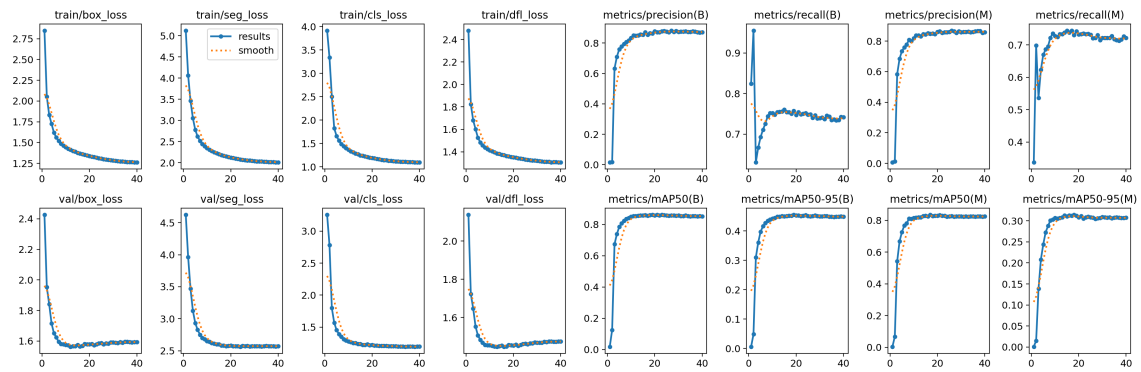


Figure A.3: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 2.

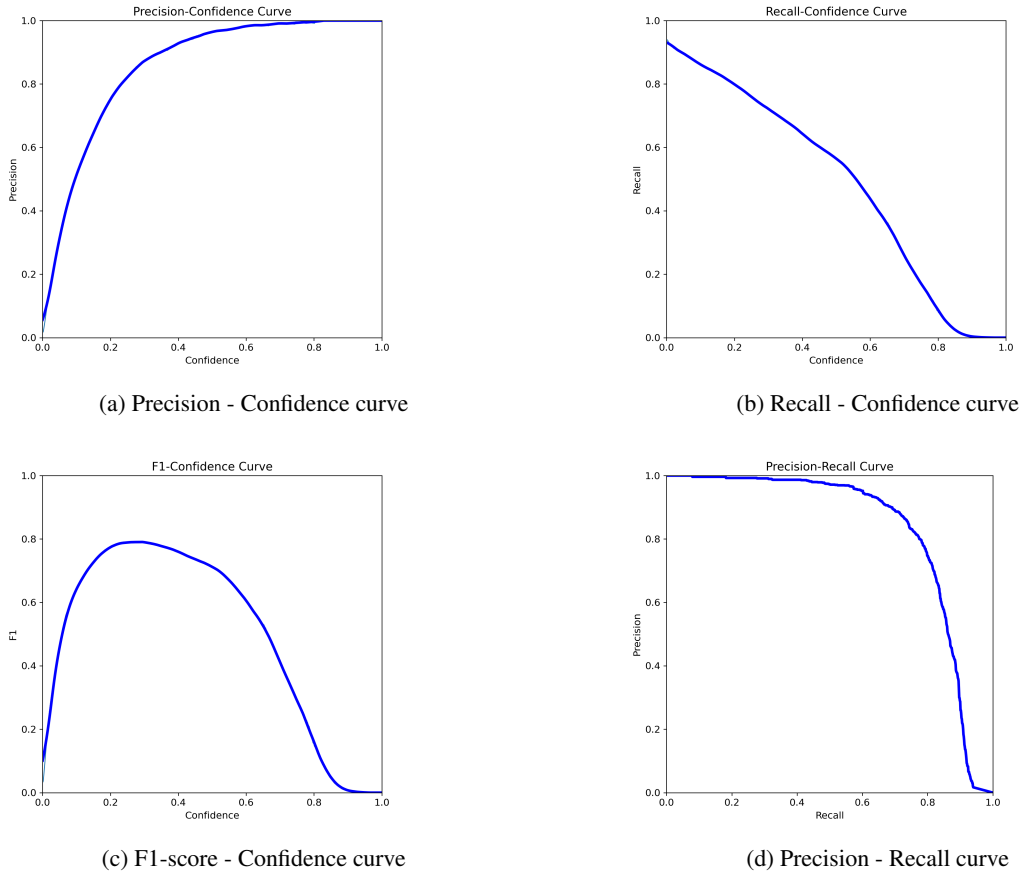


Figure A.4: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 2.

A.3 Multi-Channel Encoding Configuration 3

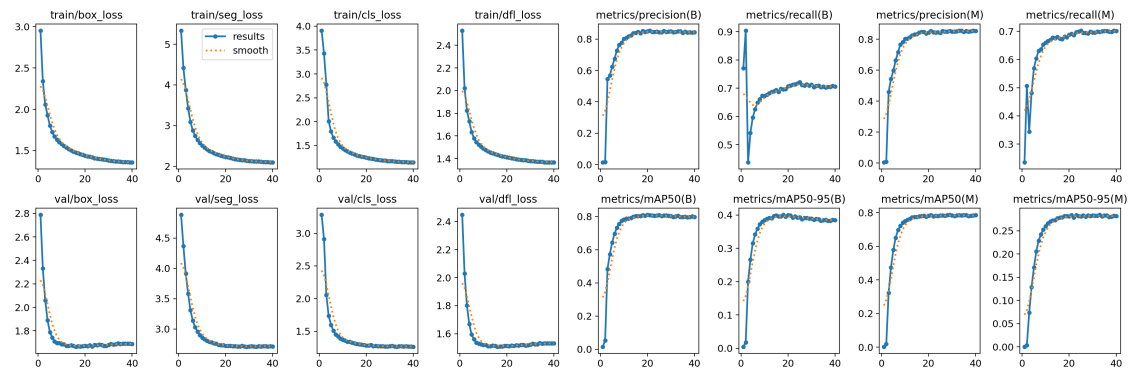


Figure A.5: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 3.

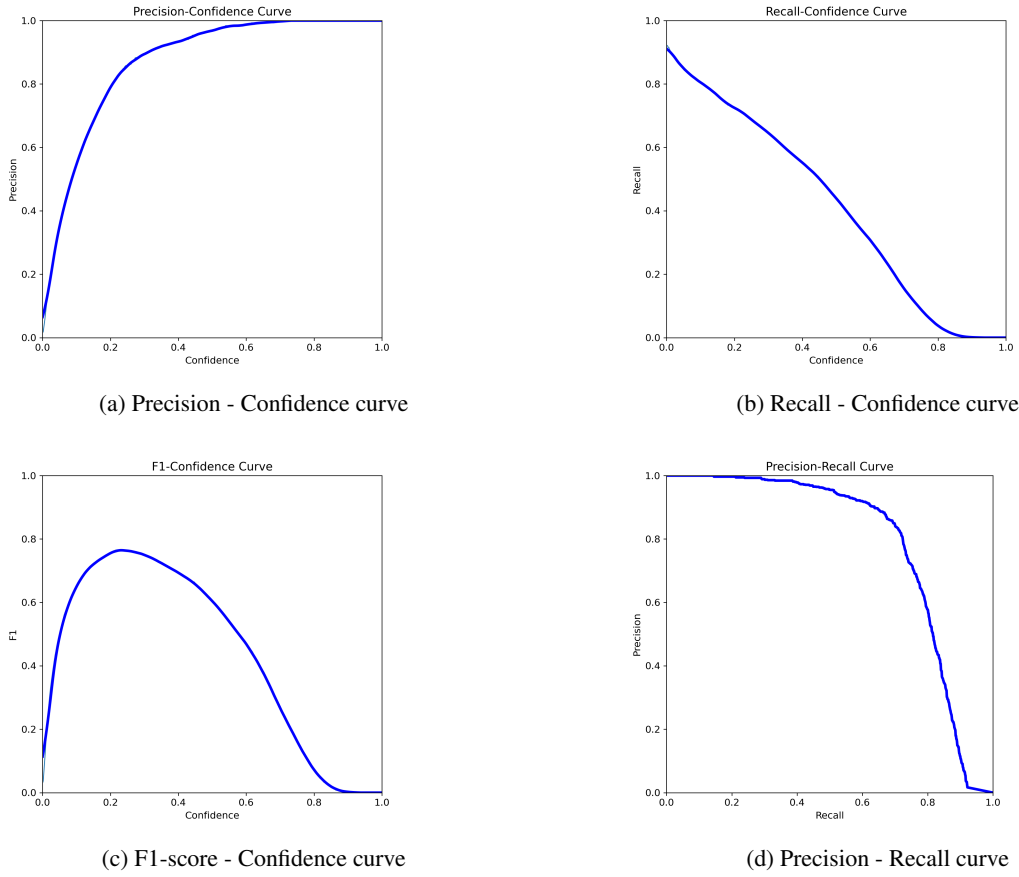


Figure A.6: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 3.

A.4 Multi-Channel Encoding Configuration 4

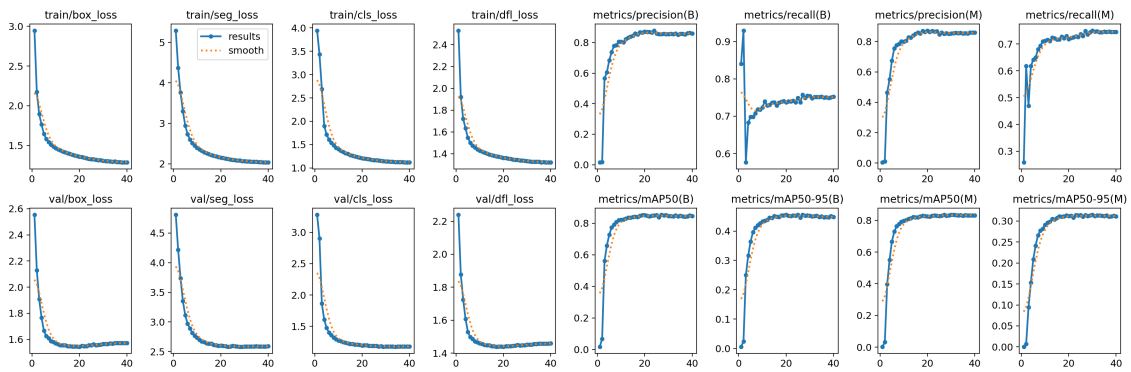


Figure A.7: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 4.

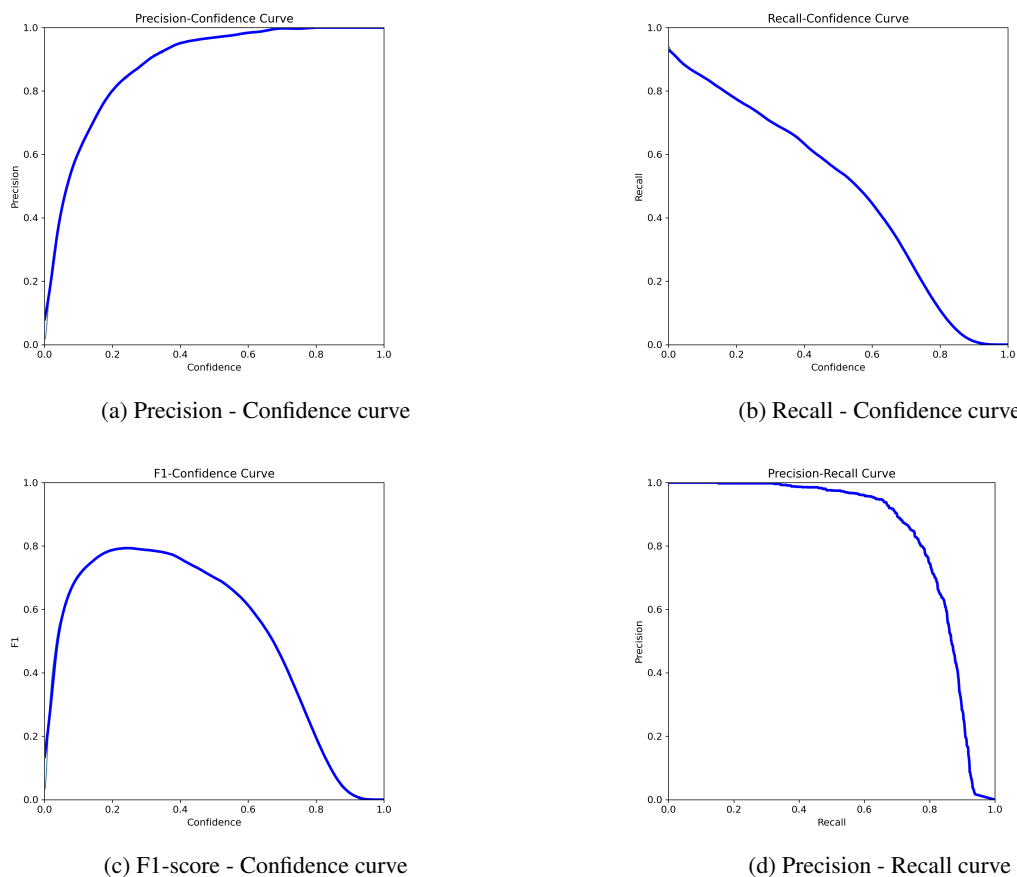


Figure A.8: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 4.

A.5 Multi-Channel Encoding Configuration 5

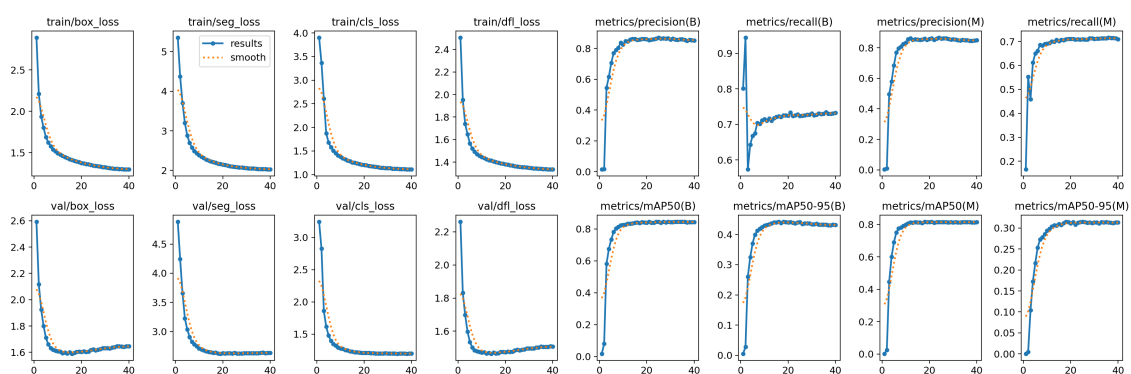
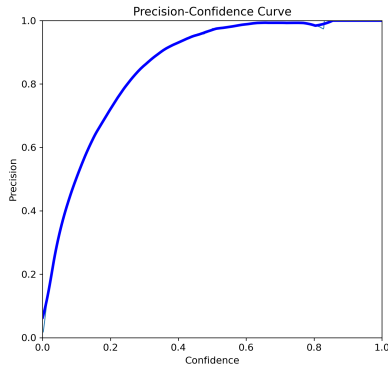
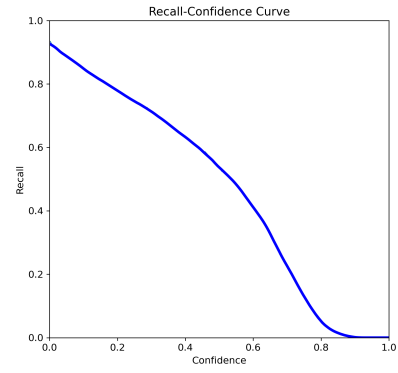


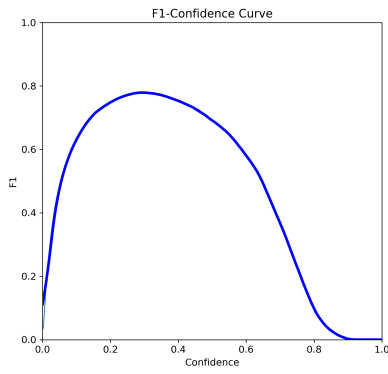
Figure A.9: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 5.



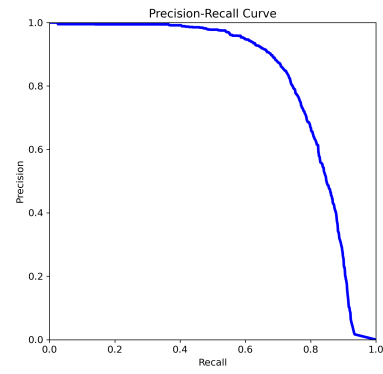
(a) Precision - Confidence curve



(b) Recall - Confidence curve



(c) F1-score - Confidence curve



(d) Precision - Recall curve

Figure A.10: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 5.

A.6 Multi-Channel Encoding Configuration 6

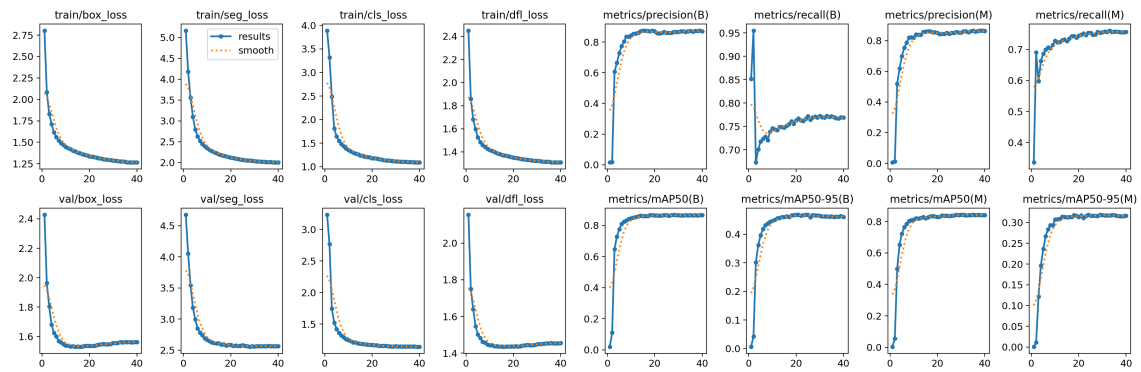


Figure A.11: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 6.

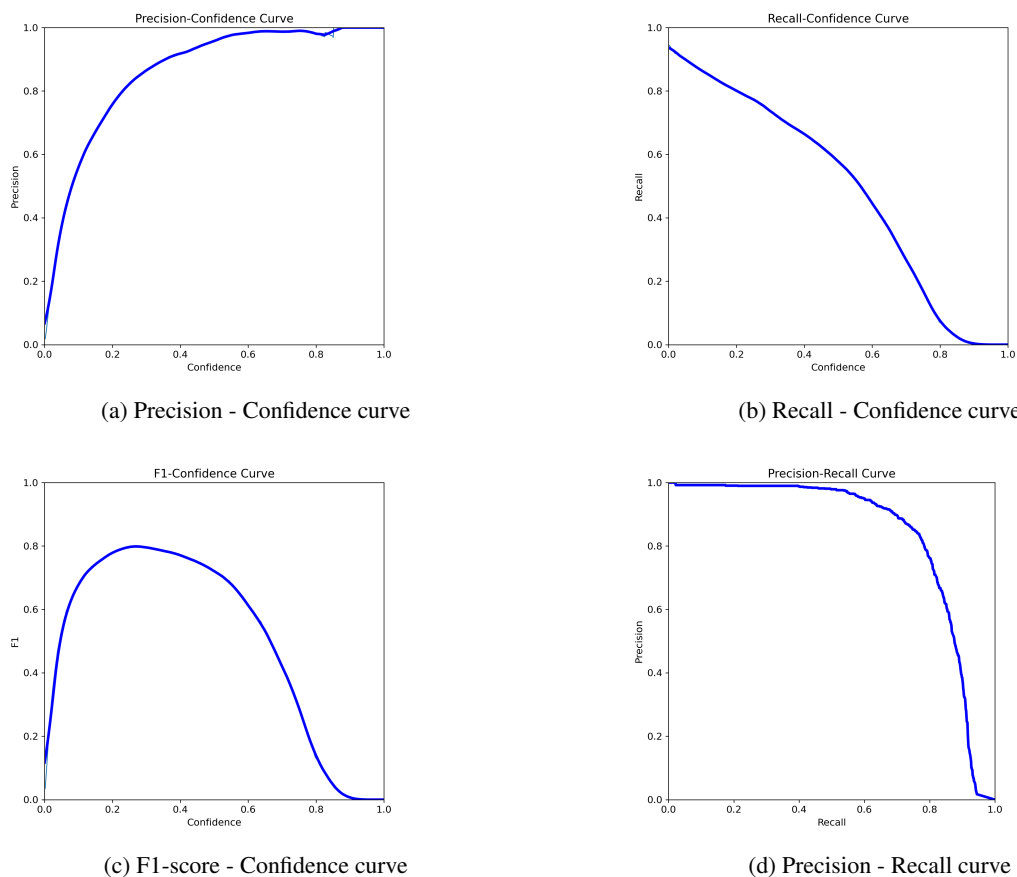


Figure A.12: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 6.

A.7 Multi-Channel Encoding Configuration 7

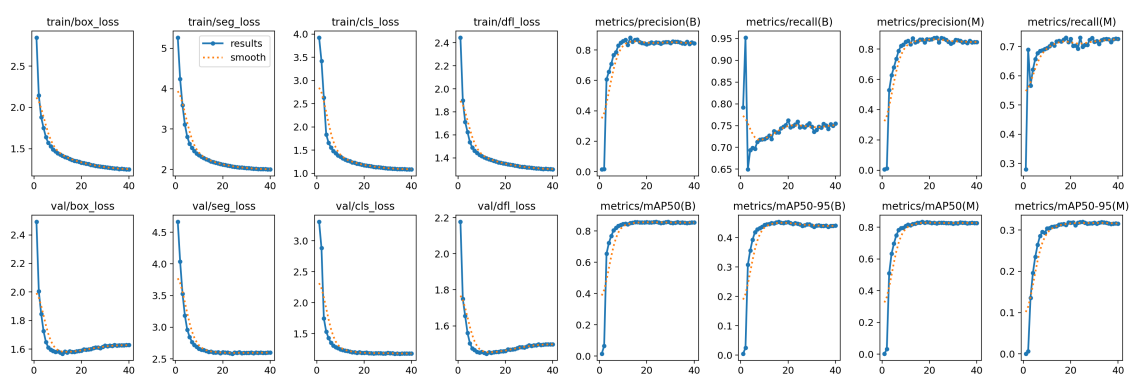


Figure A.13: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 7.

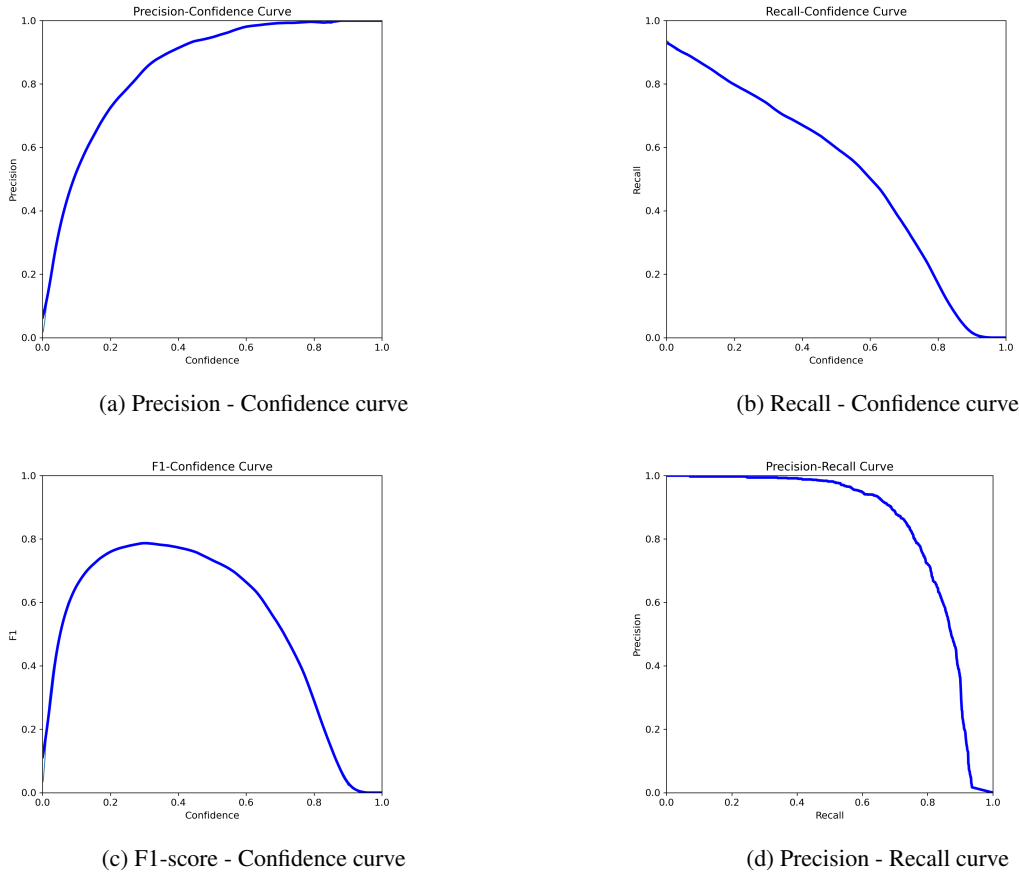


Figure A.14: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 7.

A.8 Multi-Channel Encoding Configuration 8

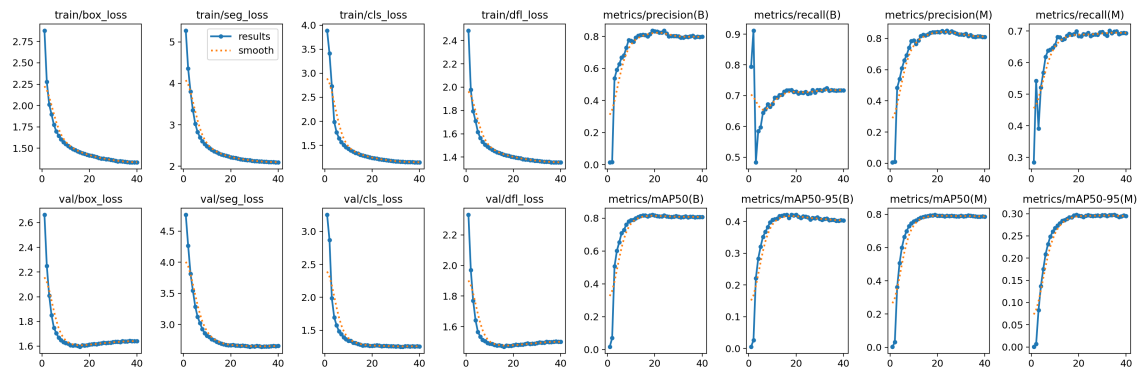


Figure A.15: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 8.

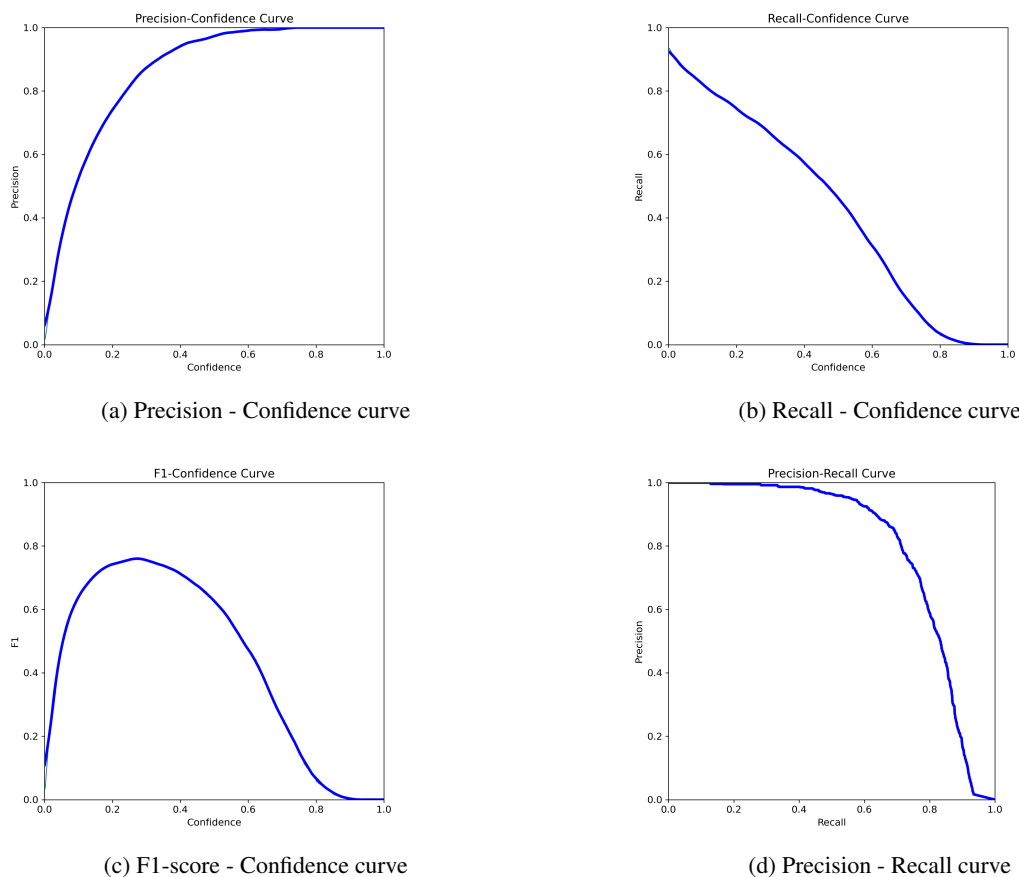


Figure A.16: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 8.

A.9 Multi-Channel Encoding Configuration 9

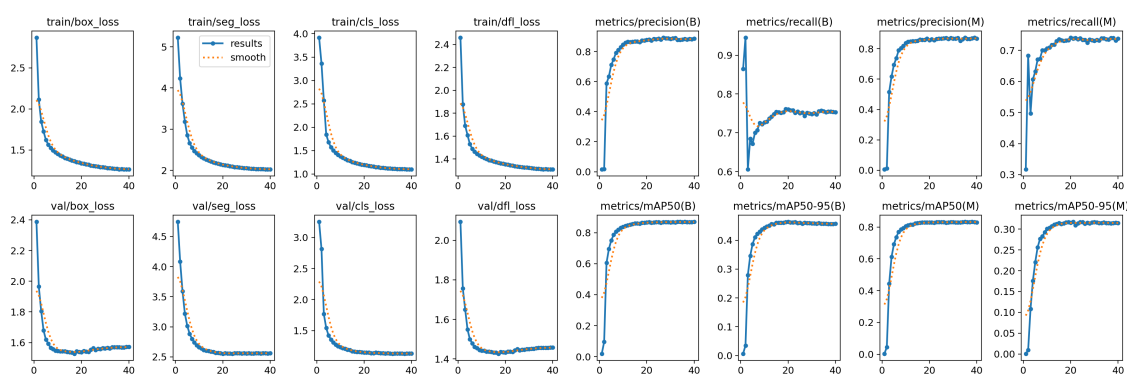
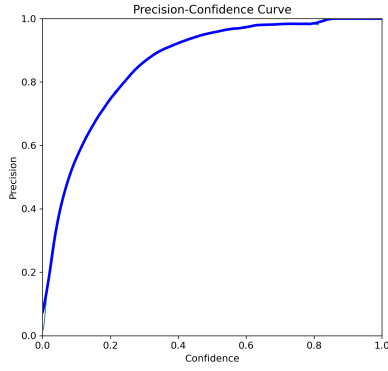
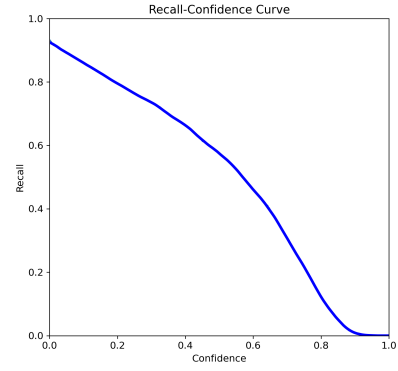


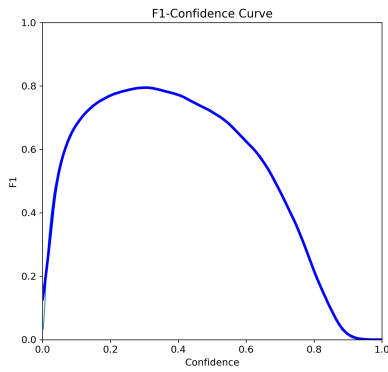
Figure A.17: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 9.



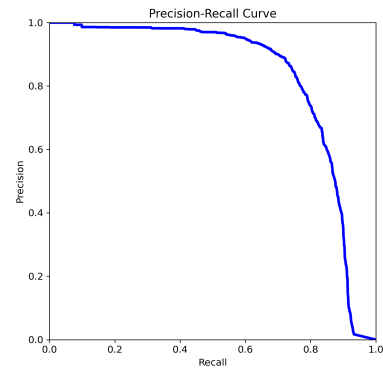
(a) Precision - Confidence curve



(b) Recall - Confidence curve



(c) F1-score - Confidence curve



(d) Precision - Recall curve

Figure A.18: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 9.

A.10 Multi-Channel Encoding Configuration 10

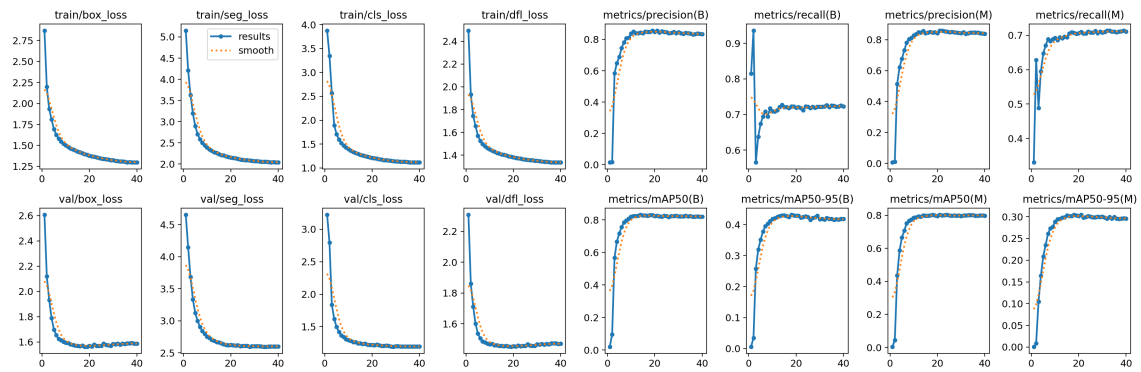


Figure A.19: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 10.

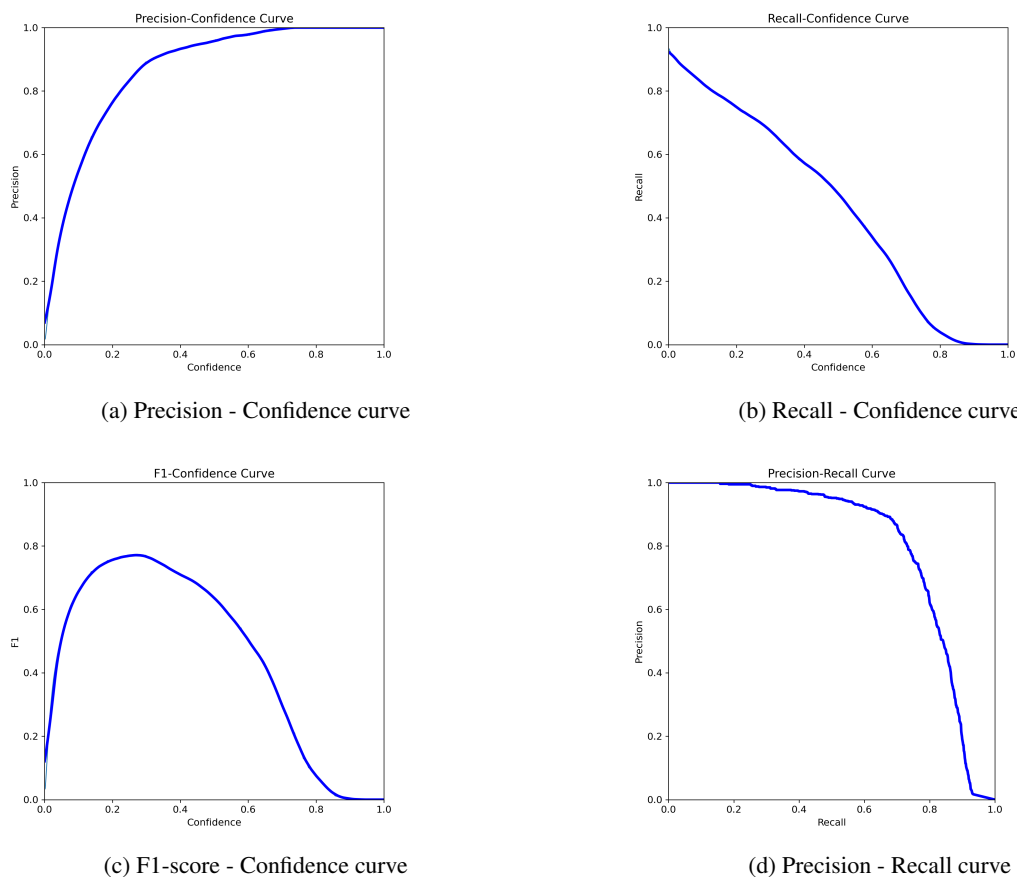


Figure A.20: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 10.

A.11 Multi-Channel Encoding Configuration 11

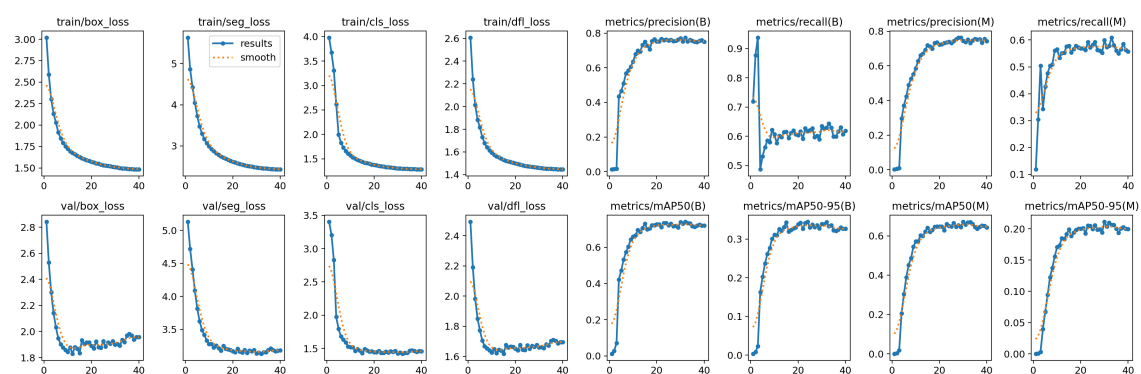


Figure A.21: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 11.

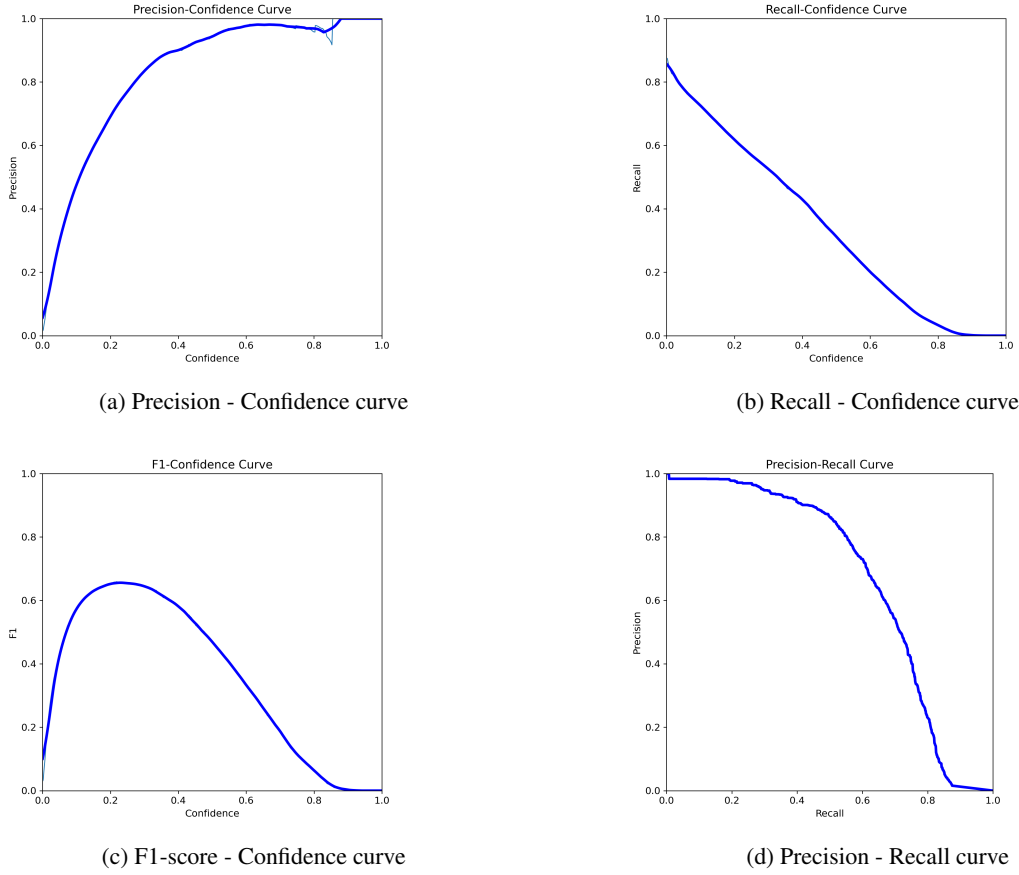


Figure A.22: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 11.

A.12 Multi-Channel Encoding Configuration 12

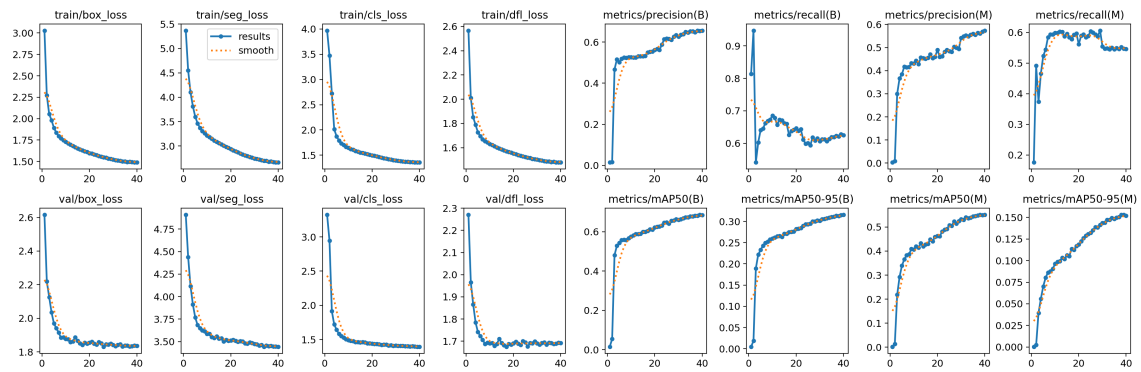
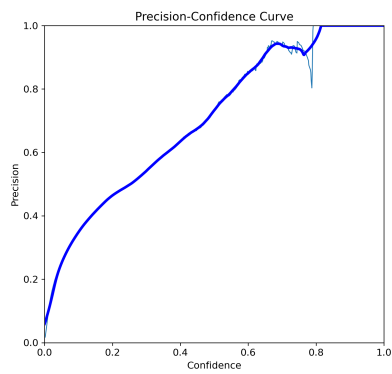
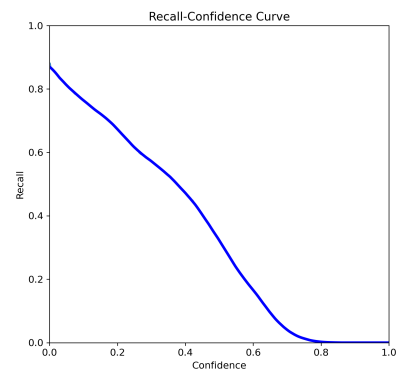


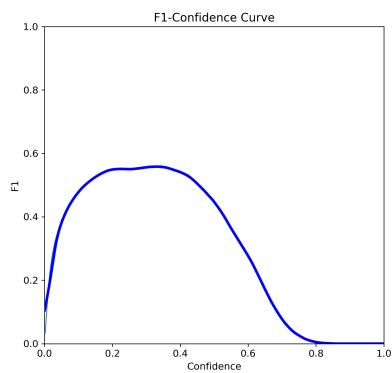
Figure A.23: Training and validation results of YOLOv8-seg nano model on multi-channel encoding configuration 12.



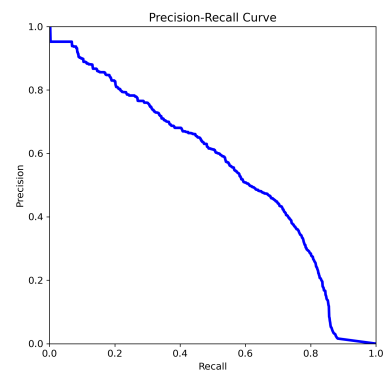
(a) Precision - Confidence curve



(b) Recall - Confidence curve



(c) F1-score - Confidence curve



(d) Precision - Recall curve

Figure A.24: Validation metric curves of YOLOv8-seg nano model on multi-channel encoding configuration 12.