Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Institute for Computer Science VII
Robotics and Telematics

Bachelor's thesis

Radar-based Material Classification using Signal Intensity and Doppler Signatures

Jannik Hohmann

August 2025

First reviewer:	Prof. Dr. Andreas Nüchter
Advisor:	M. Sc. Dong Wang

Abstract

This thesis investigates radar-based material classification using the IWRL6432BOOST development board from Texas Instruments, a low-power Frequency-Modulated Continuous Wave (FMCW) millimeter-wave (Millimeter Wave (mmWave)) radar sensor. The target domain comprises mobile robot platforms operating in visually degraded or privacy-sensitive environments. A lightweight signal processing pipeline converts raw Analog-Digital Converter (ADC) data into range and signal-intensity representations, which are segmented into data frames. From these, compact feature descriptors are derived, capturing intensity statistics over significant range bins. A multilayer perceptron (Multilayer Perceptron (MLP)) is trained on these features to enable real-time inference on radar data.

Evaluation is conducted in a controlled indoor environment with a nadir-looking sensor configuration. Five common surface materials (iron, aluminum, plexiglass, wood, limestone) are classified using an 80/20 train-test split. Performance is assessed through accuracy, macro-precision, macro-recall, macro-F1 score, and confusion matrices. Model decomposition studies demonstrate that integrating Doppler and intensity features improves robustness across measurement sessions and enables effective discrimination between materials with similar reflectivity but distinct microvibration patterns. Short temporal frames provide an optimal trade-off between responsiveness and accuracy.

The developed system achieves high data efficiency and real-time capability without reliance on camera or Light Detection and Ranging (LiDAR) sensors, while meeting the computational and power constraints of compact robotic platforms. Limitations regarding cross-domain generalization are discussed, and potential extensions—such as lightweight convolutional neural networks (Convolutional Neural Networks (CNNs)) for range-Doppler tensor processing and multi-sensor fusion for comprehensive 3D scene understanding—are outlined.

Zusammenfassung

Diese Arbeit untersucht die radarbasierte Materialklassifikation unter ausschließlicher Verwendung des IWRL6432BOOST-Entwicklungsboards von Texas Instruments, einem stromsparenden FMCW-Millimeterwellen-mmWave-Radarsensor. Zielsysteme sind mobile Roboterplattformen in visuell beeinträchtigten oder datenschutzsensiblen Umgebungen.

Eine schlanke Signalverarbeitungspipeline transformiert rohe ADC-Daten, die verarbeitet werden, um Entfernungs- und Signalintensitätsdarstellungen zu erzeugen, welche anschließend in einzelne Datenrahmen unterteilt werden. Darauf aufbauend werden kompakte Merkmalsdeskriptoren abgeleitet, die Intensitätsstatistiken über signifikante Entfernungs-Bins erfassen. Ein kompaktes mehrschichtiges Perzeptron (MLP) wird auf diesen Merkmalen trainiert, um eine Echtzeitinferenz auf den Radardaten durchzuführen.

Die Evaluierung erfolgt in einer kontrollierten Innenraumumgebung unter Verwendung einer nach unten gerichteten Sensorkonfiguration. Fünf gängige Oberflächenmaterialien (Eisen, Aluminium, Plexiglas, Holz, Kalkstein) werden mit einer 80/20-Training-Test-Aufteilung getestet. Die Leistung wird anhand von Genauigkeit, Makro-Präzision, Makro-Recall, Makro-F1-Score und Konfusionsmatrizen bewertet.

Modellzerlegungsstudien zeigen, dass die Integration von Doppler- und Intensitätsmerkmalen die Systemrobustheit über verschiedene Messsitzungen hinweg verbessert und eine effektive Unterscheidung zwischen Materialien ermöglicht, die ähnliche Reflexionseigenschaften, aber unterschiedliche Mikrovibrationsmuster aufweisen. Kurzzeitige Datenrahmen bieten hierbei eine optimale Antwortlatenz.

Das entwickelte System bietet eine reduzierte, aber funktionale Pipeline, hohe Dateneffizienz, arbeitet ohne Kamera- oder LiDAR-Sensoren und erfüllt die Rechen- und Energieanforderungen kompakter Roboterplattformen. Die Limitationen hinsichtlich der domänenübergreifenden Generalisierung werden diskutiert, und potenzielle Erweiterungen - einschließlich leichtgewichtiger CNNs für die Entfernungs-Doppler-Tensorverarbeitung sowie Multi-Sensor-Fusionsansätze für ein umfassenderes 3D-Szenenverständnis werden aufgezeigt.

Contents

List of Figures

List of Tables

Acronyms

ADAS Advanced Driver-Assistance Systems

ADC Analog-Digital Converter

BPF Bandpass Filter

CNN Convolutional Neural Network

FMCW Frequency-Modulated Continuous Wave

FR4 Glass-reinforced epoxy laminate

IF Intermediate Frequency

LiDAR Light Detection and Ranging

LNA Low Noise Amplifier

MLP Multilayer Perceptron

mmWave Millimeter Wave

MSps Mega Samples per second

RCS Radar Cross Section

ReLU Rectified Linear Unit

RF Radio Frequency

RFCMOS Radio-Frequency CMOS

Rx Receiver

SDK Software Development Kit

TI Texas Instruments

Tx Transmitter

USB Universal Serial Bus

Chapter 1

Introduction

Radar, short for *Radio Detection and Ranging*, is a sensing technology that originated in the early 20th century and was initially developed for military purposes. The first practical radar systems appeared in the 1930s and 1940s, with significant advancements during World War II, where radar was used to detect and track enemy aircraft and ships [10]. These early applications demonstrated the potential of radar as a reliable detection technology, even under challenging environmental conditions.

Since then, radar technology has evolved substantially and has become an integral part of modern sensing systems across numerous domains, including astronomy, meteorology, robotics, automotive systems, medical diagnostics, and two- or three-dimensional mapping [20]. One of radar's defining strengths is its capability to operate independently of ambient light and in adverse weather conditions, such as fog, rain, or snow [1]. This robustness has driven its continued adoption and adaptation to an increasingly wide range of applications.

In recent years, mmWave radar sensors have gained particular importance. Operating in the 30–300 GHz frequency range, these sensors combine the established advantages of radar with the benefits of higher spatial resolution and compact hardware design. Their short wavelengths allow them to capture fine structural details, penetrate certain materials, and deliver accurate distance and velocity estimates. Unlike cameras or LiDAR, mmWave radar remains largely unaffected by poor lighting or adverse weather conditions, which makes it especially useful for robust perception in safety-critical environments [17]. Additionally, mmWave radar, unlike LiDAR, can operate in privacy-sensitive applications without compromising user confidentiality.

Beyond object detection and tracking, mmWave radar offers unique potential for material classification. Differences in reflectivity, dielectric properties, and Doppler micro-signatures enable the discrimination of surfaces that appear visually similar but differ in physical composition. This capability is valuable in scenarios where visual sensing is limited or not desirable due to privacy concerns. Typical use cases include mobile robotics, where surface type recognition can guide navigation and manipulation strategies. industrial automation and quality control, where non-destructive material inspection is required; automotive guidance systems, where material classification is crucial for safe and efficient operation; and recycling processes, where robust sorting of different materials is essential [28].

The focus of this thesis is the development of a lightweight radar-based material classification system using the Texas Instruments IWRL6432BOOST mmWave development board. The system is designed to operate on embedded hardware with strict computational and energy constraints, while still achieving reliable classification of five common surface materials. To this end, a compact machine learning pipeline based on feature extraction and a MLP is implemented, evaluated, and analyzed under different testing conditions.

The remainder of this thesis is organized as follows. Chapter 2 introduces the theoretical background of radar sensing, mmWave technology, and the fundamentals of machine learning for classification tasks. Chapter 3 describes the design of the radar-based material classification system, including hardware setup, data acquisition, and preprocessing methods. Chapter 4 presents the machine learning pipeline, model architecture, and training procedure. Chapter 5 provides a detailed evaluation of the system, analyzing its performance under different test scenarios and discussing strengths and weaknesses. Finally, Chapter 6 concludes the thesis with a summary of findings, an outlook on possible improvements, and potential applications of the system.

Chapter 2

Technical Background

2.1 mmWave Radar Sensors: An Overview

2.1.1 Basics of Radar Technology

The foundations of radar technology were laid by Heinrich Hertz in the late 19th century, when he demonstrated that radio waves could be reflected by metallic objects. This insight inspired the idea of using electromagnetic waves for detecting and locating targets. In 1904, Christian Hülsmeyer introduced the *Telemobiloscope*, the first practical radar device, marking the beginning of applied radar sensing [5]. Radar systems underwent significant advancements in the 1930s and 1940s, particularly during World War II, and subsequently found wide use in civil applications such as aviation, meteorology, and navigation.

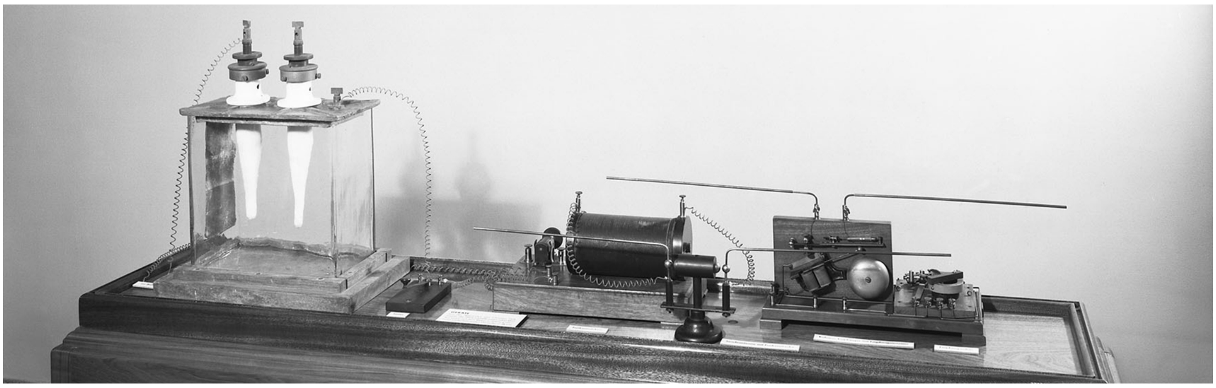


Figure 2.1: Christian Hülsmeyer's ”Telemobiloskop”. The device was demonstrated in 1904 as a system for detecting metallic objects using electromagnetic waves.
Source: [3]

The fundamental range measurement in radar is based on the following equation:

$$d = \frac{c \cdot t}{2} \tag{2.1}$$

where d is the distance between the radar and the target, c is the speed of light, and t is the round-trip time of the transmitted wave [27].

Modern radar systems follow the same principles but are capable of extracting additional information from the received signal. In addition to distance, the echo also carries information about *signal intensity* and *frequency shift*. The signal intensity depends on the distance, the target's material properties, and its Radar Cross Section (RCS). The frequency shift arises due to the Doppler effect when a target moves relative to the radar, allowing estimation of its radial velocity:

$$v = \frac{f_d \cdot c}{2 \cdot f_0} \tag{2.2}$$

where v is the relative velocity, f_d is the Doppler frequency shift, and f_0 is the transmitted frequency of the radar signal [26]. Note that this relationship applies only to the radial component of motion.

Another important relation is the radar equation, which expresses the received signal power as a function of system parameters and target properties:

$$P_r = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot d^4} \tag{2.3}$$

Here, P_r is the received power, P_t is the transmitted power, G_t and G_r are the antenna gains, λ is the wavelength, σ is the RCS of the target, and d is the distance to the target. As shown in Equation 2.3, the received power decreases proportionally to $1/d^4$, which severely limits the operational range of radar systems and necessitates the use of high-gain antennas and sufficient transmit power.

These physical principles form the foundation of all radar systems, from early mechanical prototypes to modern high-resolution mmWave sensors, which will be discussed in the following section.

2.1.2 Millimeter Wave Radar Sensors

Millimeter-wave (mmWave) radar sensors are a specialized type of radar technology that operate with short wavelengths of electromagnetic radiation, typically in the range of 30 to 300 GHz, corresponding to wavelengths between 10 mm and 1 mm [17]. These small wavelengths allow for compact hardware design with relatively small antennas, making them suitable for modern systems where space is limited. Furthermore, short wavelengths ensure strong reflections from most objects and prevent the radar signal from simply passing through them. Another advantage is the high measurement precision, which enables the detection of displacements smaller than 1 mm under controlled conditions [15].

A typical mmWave radar system consists of a transmitter, a receiver, and an antenna or radiating element. Often, an antenna array is used, which provides angular resolution and enables signal processing techniques such as beam steering. This improves the signal-to-noise ratio and overall system performance in terms of range and resolution [7, 17, 25]. The transmitter and receiver (Transmitter (Tx) and Receiver (Rx) in Figure 2.2) are typically integrated into a single chip. They determine the operating frequency and transmit power, which directly influence how well the sensor processes reflected signals. Their design is also crucial for compensating for limitations such as in-phase and quadrature imbalance [14, 17].

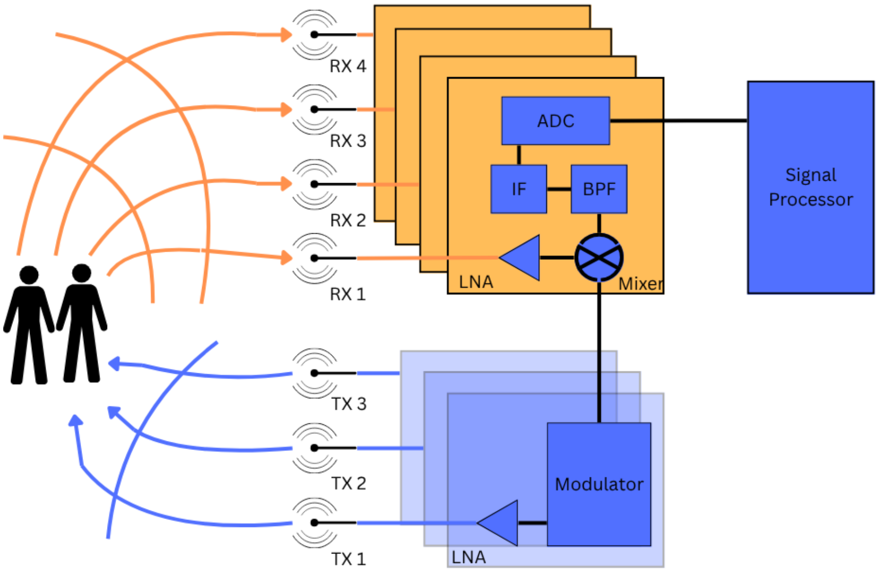


Figure 2.2: Schematic representing the basic design of antenna components in a mmWave radar device. *Source:* [17]

Based on the hardware architecture in Figure 2.2, the basic detection process of an mmWave radar can be described as follows. The transmitter emits a radar signal, which is radiated by the antenna and propagates through the environment. After reflection from an object, the echo signal is collected by the receiver antenna, amplified by a Low Noise Amplifier (LNA), and mixed with a local oscillator to generate an Intermediate Frequency (IF) signal. The IF signal is then passed through a Bandpass Filter (BPF) and digitized by an ADC. Subsequently, pulse compression is applied to enhance range resolution.

The processed signal is further analyzed to extract distance, relative velocity, and angle information. From these parameters, the radar system generates detection points. If detection points persist over time, they are aggregated into tracks, which are updated using filtering techniques such as Kalman filters to provide a stable and accurate representation of the target state [14, 15, 17].

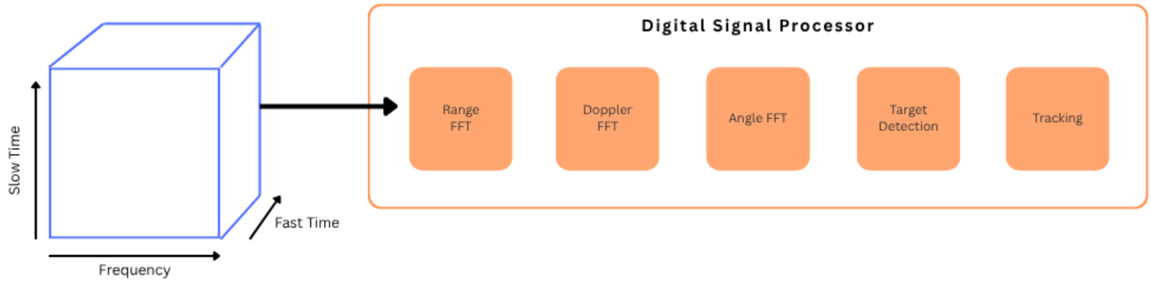


Figure 2.3: High-level schematic of the signal processing chain in mmWave radar devices. Objects are detected and tracked step by step from left to right.
Source: [17]

MmWave radar sensors can be categorized based on their operating mode. The most common approach is the FMCW mode, which is standard in most modern devices. Other modes include pulse radar and continuous-wave radar, which offer fewer capabilities and are therefore less common.

FMCW radar continuously transmits a signal that is modulated in frequency over time (a chirp signal). In this mode, the frequency shift of the reflected signal compared to the transmitted signal (the beat frequency) is used to determine the relative velocity of the object, while distance is measured analogously to traditional radar systems [4].

Key characteristics of FMCW radar include continuous data acquisition, high precision, and simultaneous measurement of distance and velocity [21]. These properties explain its widespread adoption in modern radar systems.

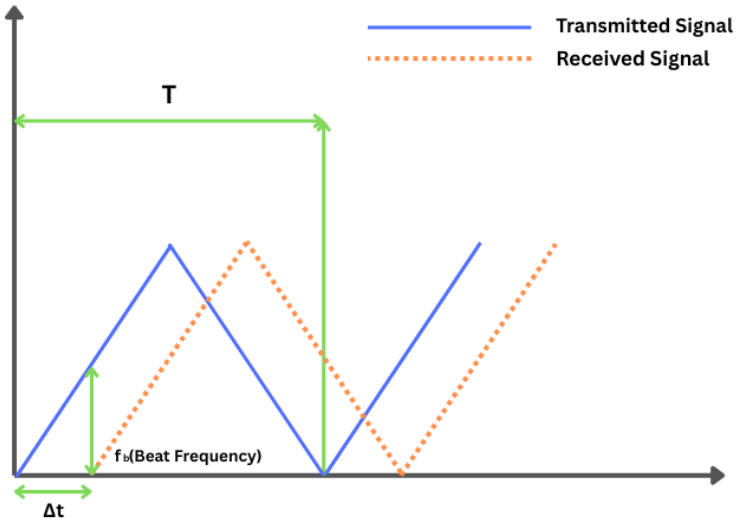


Figure 2.4: Visualization of the frequency modulation over time used in FMCW radar. The modulation typically follows a sawtooth or triangular pattern.
Source: [4]

2.2 Texas Instruments (TI) mmWave Plattform

This work is based on the TI mmWave platform, a development platform for mmWave radar sensors. It encompasses a wide range of devices and dedicated software tools that support the development of new applications and the integration of mmWave radar sensors into existing systems.

The TI platform consists of several hardware components, such as the xWRLx432 family of mmWave radar sensors and the AWR2x44 radar sensors. TI devices use FMCW sensing with a specialized Radio-Frequency CMOS (RFCMOS) single-chip design, which offers flexibility and programmability in both the Radio Frequency (RF) front-end and the back-end of development. TI's mmWave sensors can store up to 512 chirps with four profiles per frame, enabling flexible, real-time configuration for various application needs. This allows the sensors to maximize useful data extraction and adapt chirps and processing to achieve higher range, velocity resolution, or to optimize for specific algorithms as required [14].

For their radar devices, TI provides comprehensive software, drivers, and tools in the form of the mmWave Software Development Kit (SDK) and mmWave Studio. The SDK includes several example applications to test device functionality, along with a set of libraries and workflows to support the development of new applications. Furthermore, it offers tools to visualize data collected by the radar device in real time, such as the mmWave Demo Visualizer and the headless parser. TI also provides the Uniflash tool to flash firmware to the radar device and update its software. The main focus of TI's platform is to support the development of industrial and automotive applications [14].

2.3 Fundamentals of Machine Learning for Classification

Machine learning enables computers to learn patterns from data and make predictions or decisions without explicit case-by-case programming. In classification, the goal is to assign an input (for example, a radar signature) to one of several predefined categories [16].

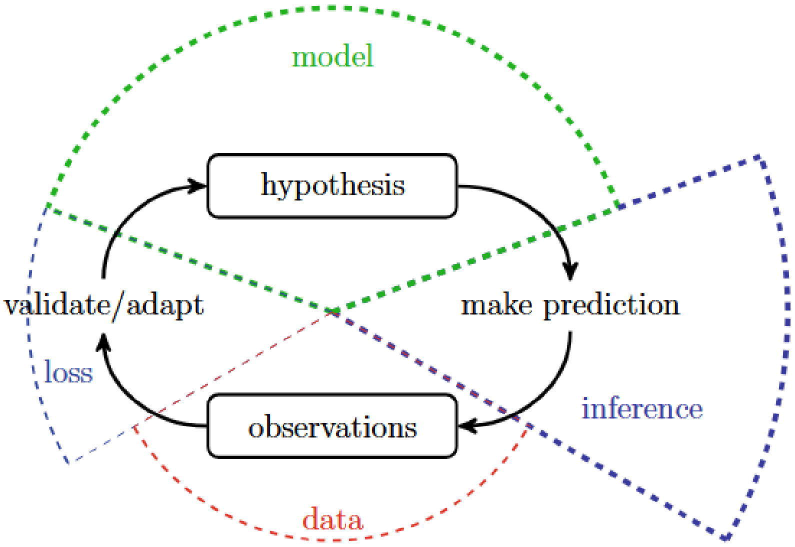


Figure 2.5: Overview of how machine learning operates. Data are mapped through a model to class scores, which are converted into probabilities and compared to the ground truth via the a loss that guides learning.
Source: [16]

In supervised learning, a model f_θ is trained on labeled pairs $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $\mathbf{x} \in \mathbb{R}^d$ denotes a feature vector and $y \in \{1, \dots, C\}$ the class. The model outputs one score (logit) z_k per class k. These scores are converted to probabilities with the softmax

$$p_k = \frac{\exp(z_k)}{\sum_{j=1}^{C} \exp(z_j)}, \qquad \sum_{k=1}^{C} p_k = 1, \tag{2.4}$$

which normalizes the scores so that p_k represents the estimated probability of the input belonging to class k [22].

Learning is guided by a loss function that quantifies the discrepancy between predicted probabilities and true labels. For multi-class classification with one-hot encoded labels, the most common choice is categorical cross-entropy. It combines the softmax output with a logarithmic penalty on the probability assigned to the correct class:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(z_{y^{(i)}}^{(i)})}{\sum_{j=1}^{C}\exp(z_j^{(i)})}\right), \tag{2.5}$$

where N is the number of training samples, $z_j^{(i)}$ is the logit of class j for sample i, and $y^{(i)}$ is the correct class index. Equation 2.5 penalizes the model when the assigned probability for the true class is low, thereby encouraging the parameters θ to shift probability mass toward the correct class during training [22].

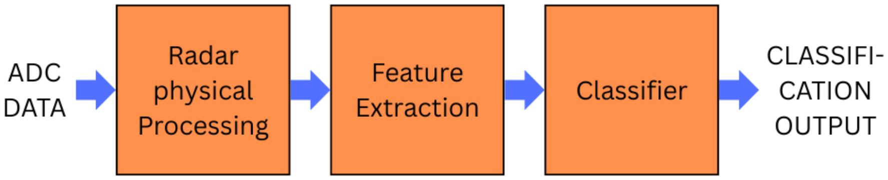


Figure 2.6: High level view of a radar based classification pipeline from signal to features to class prediction.
Source: [28]

To assess a classifier, it is not enough to train it. Also, standard metrics that describe how well it performs are needed. The most basic metric is the overall accuracy

$$\text{Accuracy} = \frac{\text{number of correctly classified samples}}{\text{total number of samples}}, \tag{2.6}$$

which reports the fraction of correct predictions. Accuracy is easy to interpret but can be misleading when classes have very different frequencies.
Therefore, per-class metrics are widely used. For a given class i, let TP_i denote true positives,

FP_i false positives, and FN_i false negatives. Precision, recall, and F1 score are then

$$\mathrm{Precision}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}, \tag{2.7}$$

$$\tag{2.8}$$

$$\mathrm{Recall}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{2.9}$$

$$\tag{2.10}$$

$$\mathrm{F1}_i = \frac{2 \cdot \mathrm{Precision}_i \cdot \mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i}. \tag{2.11}$$

Equation (2.7) reflects how many predicted positives are correct. Equation (2.9) reflects how many true instances are identified. Equation (2.11) balances both in a single number [9].

Since results should represent all classes equally, macro averages are often reported by taking the unweighted mean across classes, for example, $\mathrm{F1}_{\mathrm{macro}} = \frac{1}{C}\sum_{i=1}^{C} \mathrm{F1}_i$. In addition, confusion matrices visualize correct and incorrect assignments per class and reveal systematic error patterns [9].

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Figure 2.7: Schematic representation of a binary confusion matrix with the four basic outcomes: true positives, false positives, true negatives, and false negatives.

An important aspect of any machine learning workflow is the choice of hyperparameters, which control how the training process is carried out. The *learning rate* determines the step size with which the optimizer updates the model parameters; overly large values can cause divergence, while overly small values lead to very slow convergence. The *batch size* specifies how many samples are processed before the model parameters are updated, influencing both the stability of training and the required computational resources. The *number of epochs* defines how many times the model iterates over the entire training dataset, balancing between underfitting (too few epochs) and overfitting (too many epochs). To further guide generalization, *regularization* techniques are employed, for example, *weight decay*, which penalizes large parameter values to prevent the model from becoming overly complex. Finally, mechanisms such as *early stopping* and *learning rate schedulers* adapt the training dynamically: early stopping halts training once validation performance stagnates, while schedulers gradually reduce the learning rate when improvements slow down. Together, these hyperparameters significantly affect the final accuracy, stability, and generalization ability of a classification model [23].

Hyperparameter	**Function**	**Effect on Training**
Learning Rate	Controls the step size of parameter updates	High values speed up learning but risk instability; low values improve stability but slow convergence
Batch Size	Number of samples per update	Larger batches yield smoother gradients but require more memory; smaller batches increase stochasticity
Epochs	Number of full passes over the dataset	Too few cause underfitting; too many can lead to overfitting
Weight Decay (L2 regularization)	Penalizes large parameter values	Prevents overfitting and encourages simpler models
Early Stopping	Stops training when validation performance stagnates	Avoids overfitting and reduces unnecessary computation
Learning Rate Scheduler	Adjusts the learning rate during training	Reduces the step size when progress slows, improving convergence

Table 2.1: Overview of commonly used hyperparameters in classification tasks [23]

Chapter 3

System Design and Implementation

3.1 Selection and Commissioning of the Sensor

In this work, the TI IWRL6432BOOST development board is used as the mmWave radar sensor. It is a low-power, 60 GHz radar application development platform based on Glass-reinforced epoxy laminate (FR4), which supports the TI mmWave SDK. The board provides all the necessary features for this work while remaining a cost-effective and compact solution.

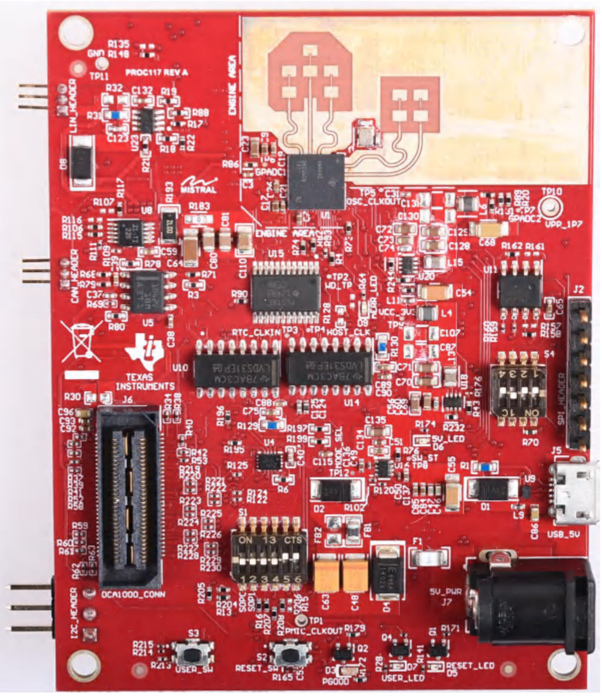


Figure 3.1: TI IWRL6432BOOST development board top view.
Source: [12]

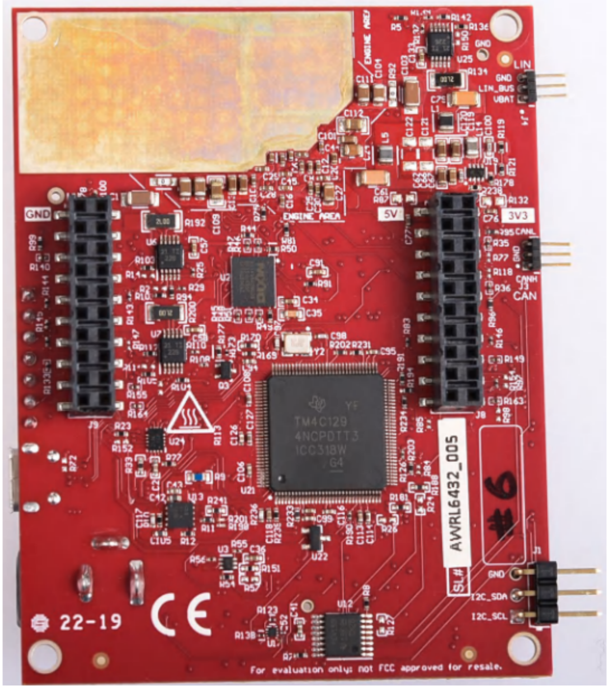


Figure 3.2: TI IWRL6432BOOST development board bottom view.
Source: [12]

The IWRL6432BOOST uses an antenna array designed as seen in Figure 3.3.
The transmitter and receiver antennas are arranged to form a virtual array with six transmitter-receiver pairs. This configuration enables an azimuth angular resolution of 29° and a coarse elevation angular resolution of 68°. The receiver antennas are placed at $\lambda/2$ spacing from each other in both the azimuth and elevation planes.

The transmitter antennas are spaced at λ in the azimuthal plane and at $\lambda/2$ in the elevation plane. The IWRL6432BOOST is connected and powered via a Universal Serial Bus (USB) connection. To flash firmware onto the device, it must be set to flashing mode. For this, the device needs to be connected to a computer via USB, and the switches must be set as follows:

S1.1	**S1.2**	**S1.3**	**S1.4**	**S1.5**	**S1.6**	**S4.1**	**S4.2**	**S4.3**	**S4.4**
OFF	OFF	OFF	OFF	OFF	ON	ON	OFF	ON	–

Table 3.1: Switch configuration for flashing mode. The switches S1.1 to S1.6 are located on the top side of the board in the lower center, while the switches S4.1 to S4.4 are located on the top side of the board in the center-right half.

If the device is reset in this configuration, it will enter flashing mode, and the Uniflash tool can

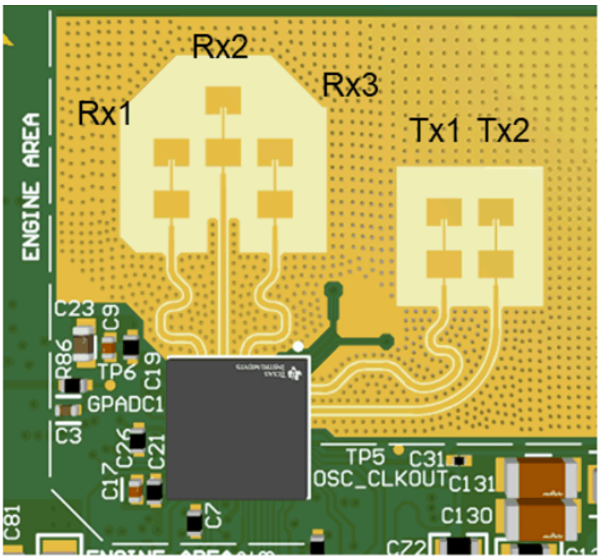


Figure 3.3: Antenna array configuration of the TI IWRL6432BOOST development board with 3 receivers (Rx) and two transmitters(Tx). They are FR4-based on the Printed Circuit Board
Source: [12]

be used to flash the firmware onto the device.

By switching **S1.1** to ON, the device can be set to functional mode and will start to operate the firmware. Once in functional mode, the device sends data to the connected computer, which can be accessed using the Parser or Visualizer tools provided with the mmWave SDK.

After commissioning, the functionality of the device was validated by running the Surface Classification demo application, which is part of the mmWave SDK. No additional calibration was necessary.

The preloaded demo project, Surface Classification, was also used as a starting point. Modifications to the firmware were made to enable custom data logging, labeling, training, and output.

3.2 Testing Environment

The testing environment for this work consisted of the TI IWRL6432BOOST development board, a mounting clamp to hold the device in a fixed position, and a computer to provide power and communicate with the device.
The computer was connected to the device via USB and was used to control the device and collect the data. The surface samples were placed on a 16.5 cm high elevation on a table under

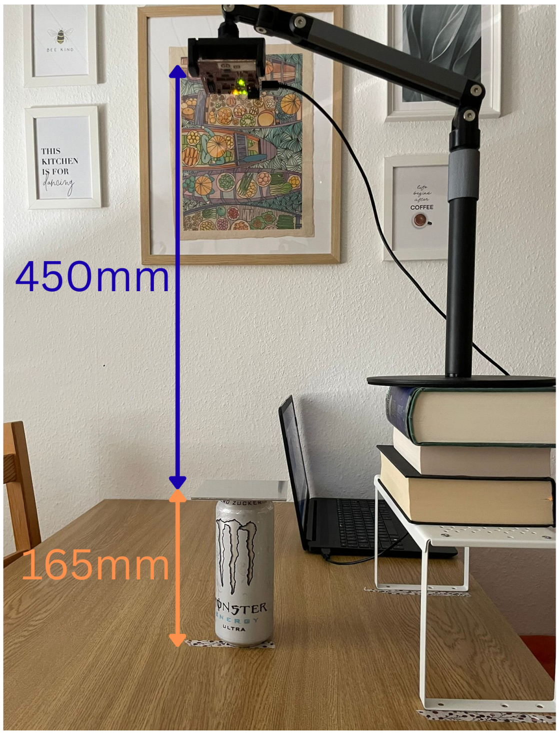


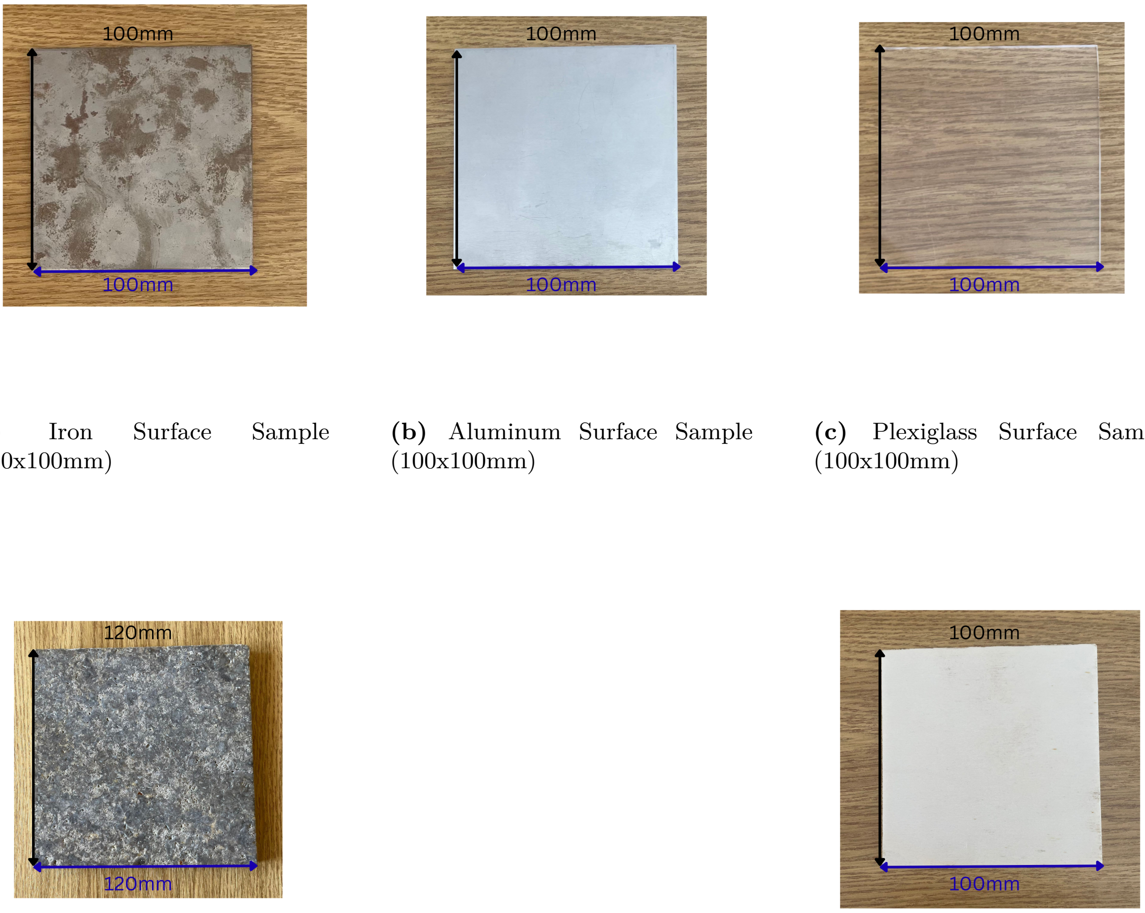
Figure 3.4: Default setup for data collection with the TI IWRL6432BOOST development board, the mounting clamp and the elevation with height descriptions.

indoor conditions. The device itself was mounted 45 cm above the surface samples and pointed downward at a 90-degree angle toward them. This position was chosen to ensure strong signal reflection and to minimize the influence of the surrounding environment on the collected data. The elevation is used to focus the data on the specific surface samples and to ignore other environmental features, such as the surface of the table. The mounting clamp and elevation were always placed in the same position to ensure reproducibility. This setup will be referred to as the **default setup** (see 3.4) in the following sections.

During testing, both the distance and the angle between the sensor and the surface samples were varied to assess their impact on the performance of the system.

3.3 Data Collection

The materials used for the surface samples were iron, aluminum, plexiglass, wood, and limestone. All materials were prepared as square samples (10 x 10 cm) with a uniform thickness of 2.5 mm, except for the limestone sample, which had a thickness of 9 mm.



(a) Iron Surface Sample (100x100mm)

(b) Aluminum Surface Sample (100x100mm)

(c) Plexiglass Surface Sample (100x100mm)

(d) Limestone Surface Sample (120x120mm)

(e) Wood Surface Sample (100x100mm)

Figure 3.5: Surface-Samples used for data collection and testing with individual size specifications.

Data collection was performed using the terminal emulator Tera Term, which enables serial communication and logging. The five surface samples were placed on the elevation platform and centered using a ruler and table markings.

The device was set to functional mode while running the Surface Classification demo application. For each surface sample, 510 seconds of data were gathered in the default setup (see 3.4). Additional data was collected using different setups that varied the distance between the sensor and the surface. For each height variation—consisting of distances of 34 cm, 40 cm, 46 cm, and 48 cm—an additional 60 seconds of data were recorded. This was done across multiple recording sessions, and the live data transmitted by the device was logged in binary format. Data logging was conducted under consistent environmental conditions, such as temperature and humidity.

The data was collected with an ADC sampling rate of 12.5 Mega Samples per second (MSps) and a frequency shift per FMCW chirp of 45 MHz per microsecond, completing 10 frames per second. This resulted in approximately 7500 frames per surface sample. The decision to use multiple short recordings of data, rather than one long recording, was made to improve variability and reduce the influence of environmental factors or sensor drift over time.

3.4 Preprocessing of the Data and Feature Engineering

The data collected by the device is prepared for further processing to enable feature engineering and training of the machine learning model.

3.4.1 Data Preprocessing

The raw data collected by the device is in binary format and must be converted into a format suitable for the machine learning model. After collection, the data was labeled with the corresponding material name and converted to CSV format using a custom Python script based on the mmWave SDK's L6432 bin-to-CSV script from the machine learning workflow. When executed, the program expects a sequence of range bins to extract from the binary file, as shown in 3.1.

```
python L6432_bin_to_CSV_custom.py dataset-binary/ 1 20
```

Listing 3.1: Example of using a Python script for binary-to-CSV data format conversion

The script extracts the surface type of the collected data from the filename and adds it as the first column of the CSV to form a header formatted as seen in 3.2.

Surface Type	Range Bin 1	Range Bin 2	Range Bin ...	Range Bin 20

Table 3.2: Example of the CSV file format used for feature engineering and model training. The first column contains the surface type, followed by the range bin values.

The script filters incomplete or corrupted frames and removes them from the dataset to ensure data validity. Furthermore, a label file is generated that contains the surface type of the dataset and the duration of the recording.

The resulting CSV file is then used for further processing and feature engineering. For this application, range bins 6 to 17 were selected for feature extraction, as they contain the most relevant information for distinguishing between the different surface types in the given setup. The processed data is shown as follows:

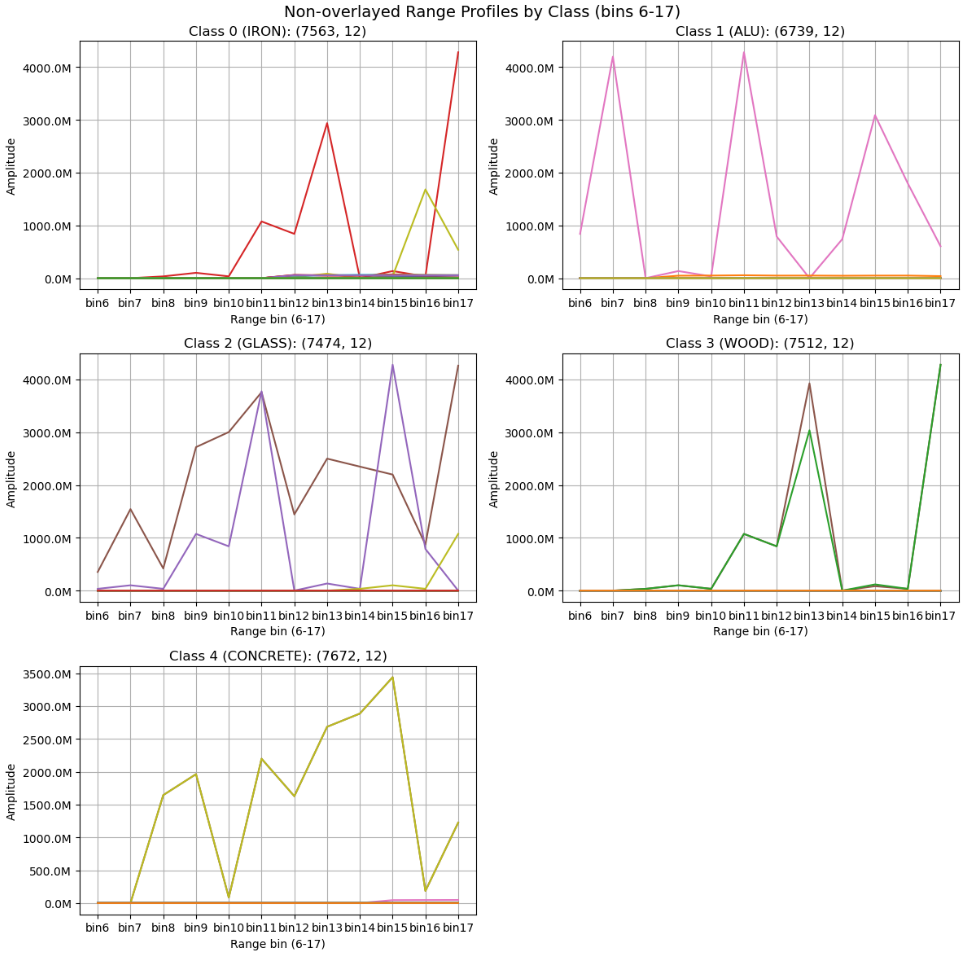


Figure 3.6: Preprocessed data visualization showing the selected range bins for feature extraction.

3.4.2 Feature Engineering

Feature engineering was conducted using a custom Jupyter notebook, adapted from the mmWave SDK's PyTorch model training workflow. The pipeline processes raw CSV files containing radar measurements and produces normalized feature vectors for training and evaluation.

The workflow consists of the following steps:

1. **Data Loading and Aggregation:** All CSV files are read and combined into a single dataframe. The class labels (surface types) are extracted and used to segregate the data by material.

2. **Windowing:** Each class's data is segmented into window frames. For this study, a window size of 1 is applied, which has no effect on signal-to-noise ratio but ensures compatibility with the SDK's workflow. Each window corresponds to a feature vector of length 12.

3. **Tensor Conversion:** Features are stored as 32-bit floating-point tensors, while class labels are stored as long-type tensors to ensure compatibility with the PyTorch cross-entropy loss function. A single test window thus has the shape $(1, 12)$.

4. **Dataset Splitting:** The dataset is divided into 80% training and 20% testing samples using stratified sampling, preserving class distribution across both sets. DataLoaders batch the data into mini-batches of size 2048 and shuffle the training data after each epoch to reduce overfitting.

5. **Normalization:** Finally, Min-Max scaling is applied to all feature tensors, ensuring that each feature lies within a comparable numerical range.

This procedure ensures consistent feature representation, balanced training and testing splits, and compatibility with the embedded inference workflow.

Chapter 4

Classification Model Development

4.1 Choice of Deep Learning Model

The classification of material types from radar data requires a model capable of capturing relationships in high-dimensional feature spaces while remaining computationally efficient. The architecture is based on the machine learning workflow provided in the mmWave SDK and adapted to the requirements of this application.

The model is a fully connected Feedforward Neural Network, or MLP, chosen for its simplicity, efficiency, and ability to approximate non-linear decision boundaries. MLPs are well-established for classification tasks and have been successfully applied in prior mmWave surface classification studies [8].

The architecture consists of the following components:

- **Input Layer:** A feature vector of preprocessed radar features is normalized using a batch normalization layer, improving training stability and convergence [13].

- **Hidden Layer:** A single fully connected layer with 8 neurons, followed by batch normalization and a Rectified Linear Unit (ReLU) activation function. ReLU is chosen for its computational efficiency and ability to mitigate vanishing gradients [19].

- **Output Layer:** A linear layer maps the hidden representation to the five material classes (iron, aluminum, plexiglass, wood, limestone). The outputs are logits, with the softmax operation applied implicitly by the cross-entropy loss during training.

The model intentionally avoids deeper architectures. Additional layers would increase memory requirements and risk overfitting, while offering limited performance gain for this dataset. Thus, the chosen MLP architecture strikes a balance between lightweight efficiency and sufficient expressive power for radar-based material classification.

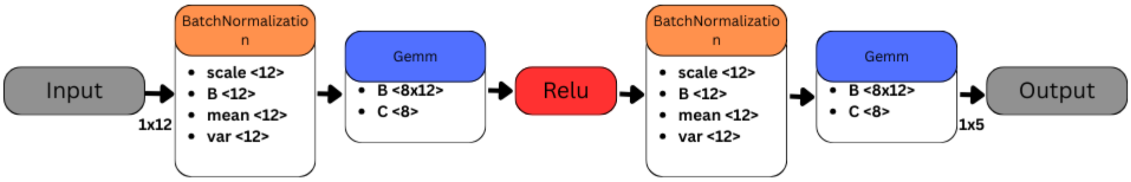


Figure 4.1: Visual representation of the material classification model architecture. It contains the input layer, two Batch-normalization layers, a hidden layer with 8 neurons, a ReLU activation function, and the output layer with 5 neurons.

4.2 Training and Hyperparameter Tuning

4.2.1 Choice of Hyperparameters

The choice of hyperparameters is crucial for the performance of the model and can significantly impact the training process and the final results. In the following section, all used Hyperparameters are introduced and their usage is explained.

Learning Rate: Initialized at 10^{-4}, providing a balance between stability and convergence speed.

Optimizer: Adam optimizer with weight decay 10^{-4} and $\epsilon = 10^{-8}$, standard settings that ensure stable convergence.

Learning-Rate Scheduler: A *ReduceLROnPlateau* scheduler reduces the learning rate by a factor of 0.7 if the validation loss does not improve for 10 consecutive epochs.

Batch Size: Set to 2048, which balances training speed, memory usage, and gradient stability given the dataset size.

Epochs: The model is trained for up to 500 epochs; convergence and stabilization of both training and validation metrics occur within this range.

Loss Function: Cross-Entropy Loss, the standard choice for multi-class classification tasks.

Window Size: A window size of 1 is used, treating each feature vector as an independent sample, which improves both speed and accuracy.

Test Split: 20% of the dataset is reserved for testing, ensuring representative evaluation while maintaining sufficient training data.

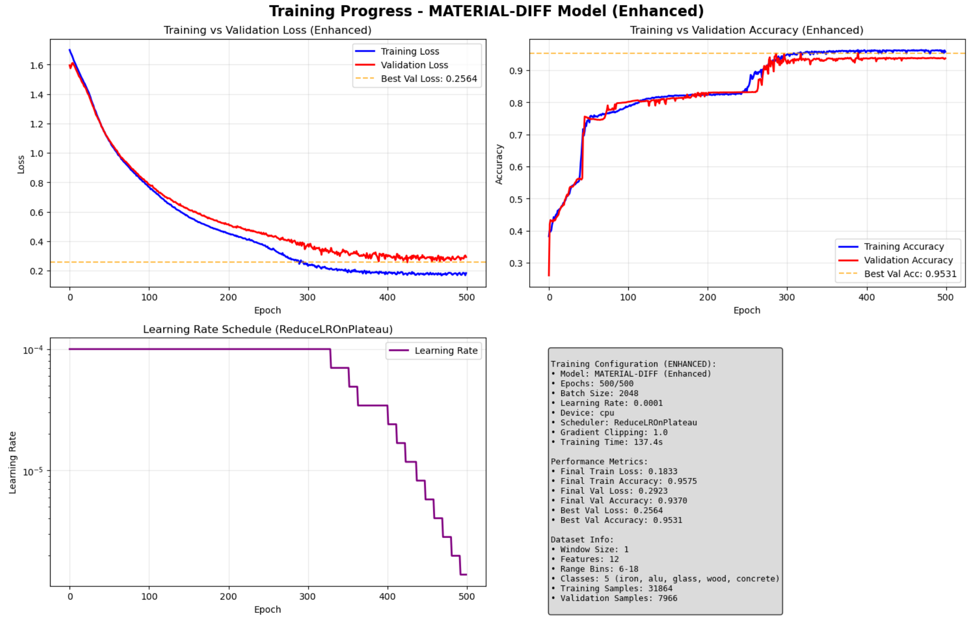
Hidden Layer Size: A hidden layer with 8 neurons provides compact yet sufficient capacity to capture patterns without overfitting.

Batch Normalization: Applied after both the input and hidden linear layers to improve stability and convergence.

4.2.2 Training Process

The training process begins with model initialization, using an input size of 12 (the number of features in the feature vector) and an output size of 5 (the number of classes). The model is trained for up to 500 epochs. In each epoch, batches of training data are processed to generate predictions, which are compared with the true labels using the loss function. Gradients are computed and the Adam optimizer updates the weights via backpropagation.

After each epoch, the model is evaluated on the validation dataset using the same forward pass procedure, and the corresponding validation loss and accuracy are logged. A learning-rate scheduler (ReduceLROnPlateau) adjusts the learning rate when validation performance stagnates, while early stopping halts training if the validation loss fails to improve for 50 consecutive epochs. Training is thus terminated either after 500 epochs or earlier if the early-stopping criterion is met, and the best-performing model is saved. Figure 4.2 provides an overview of the training workflow and results.



```
Training Configuration (ENHANCED):
• Model: MATERIAL-DIFF (Enhanced)
• Epochs: 500/500
• Batch Size: 2048
• Learning Rate: 0.0001
• Device: cpu
• Scheduler: ReduceLROnPlateau
• Gradient Clipping: 1.0
• Training Time: 137.4s

Performance Metrics:
• Final Train Loss: 0.1833
• Final Train Accuracy: 0.9575
• Final Val Loss: 0.2923
• Final Val Accuracy: 0.9370
• Best Val Loss: 0.2564
• Best Val Accuracy: 0.9531

Dataset Info:
• Window Size: 1
• Features: 12
• Range Bins: 6-18
• Classes: 5 (iron, alu, glass, wood, concrete)
• Training Samples: 31864
• Validation Samples: 7966
```

Figure 4.2: Overview of the model training process. **Top-left—Training vs. Validation Loss:** Training loss decreases steadily from 1.6 to 0.18 by epoch 500, while validation loss follows a similar trend with fluctuations, decreasing from 1.6 to 0.29. **Top-right—Training vs. Validation Accuracy:** Accuracy rises quickly at the start, improves sharply around epoch 50 (0.56 to 0.75), then plateaus after 200 epochs before rising again after 250 to stabilize at 0.957; the validation curve shows a comparable trend with stronger fluctuations, plateauing near 0.93. **Bottom-left—Learning-rate schedule (ReduceLROnPlateau):** The learning rate begins at 10^{-4}, is reduced multiple times after epoch 330, and reaches 10^{-6} by epoch 500. **Bottom-right—Infographic:** A summary of the training workflow, results, and key hyperparameters.

The final model achieved a training accuracy of 95.75% and a validation accuracy of 93.7%, with

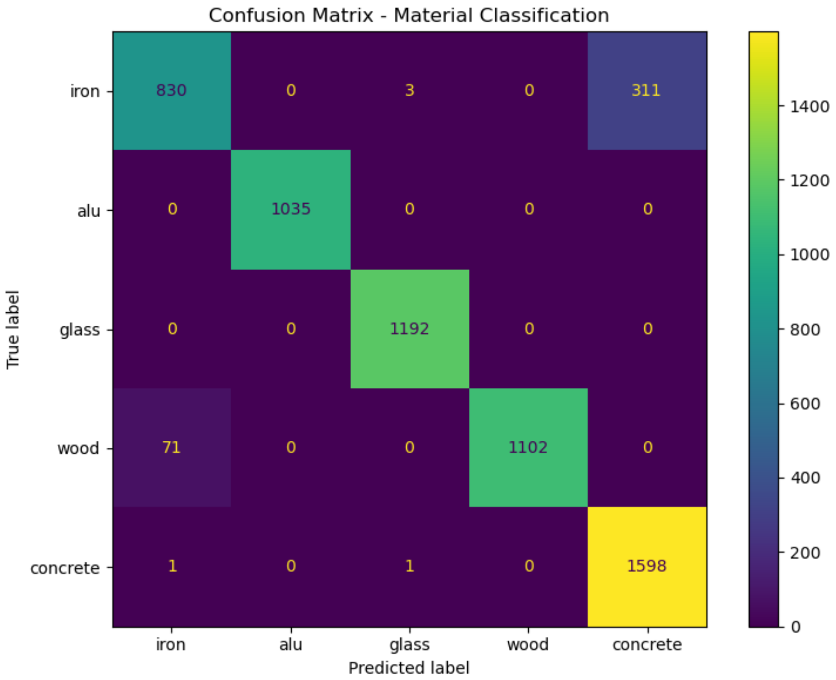


Figure 4.3: Confusion matrix of the material classification model during training. The diagonal elements represent correctly classified samples, while the off-diagonal elements indicate misclassifications. The most frequent error is the confusion of iron with concrete (limestone), occurring 311 times. Additional misclassifications include wood being classified as iron 71 times, iron being misclassified as plexiglass 3 times, and concrete being incorrectly classified once as iron and once as plexiglass.

corresponding training and validation losses of 18.3% and 29.2%, respectively.

To further evaluate the training outcome, a confusion matrix is generated to visualize the model's classification performance on the test data.

4.3 Validation under Controlled Conditions

The model is evaluated on multiple datasets to provide a comprehensive assessment of its performance. This includes testing on data that was also used during training, to verify whether the trained model performs consistently with the results shown in 4.2. In addition, the model is tested on unseen data recorded under similar conditions, to evaluate its ability to generalize and operate reliably in real-world scenarios. Finally, datasets with controlled variations in sensor angle and height are employed to assess the robustness and adaptability of the model.

For each test, the dataset is preprocessed and scaled to match the training data format before being fed into the model, which then predicts the material class for each sample. Performance is evaluated using standard metrics such as accuracy, inference time, and confidence scores. In the known-data scenario, 50 random samples per class are used to generate a comprehensive evaluation report.

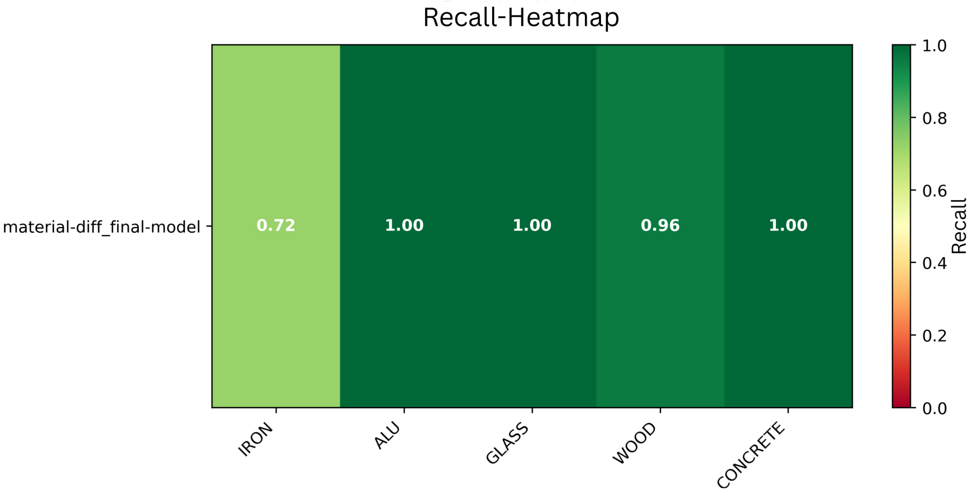
The unseen-data and variation datasets are evaluated in the same manner. The angle-variation dataset was recorded with sensor orientations of 80° and 100° relative to the surface samples, while the height-variation dataset was recorded at distances of 40 cm and 50 cm. In all evaluation heatmaps, the trained model is referred to as `material-diff_final-model`.

4.3.1 Known Data Validation

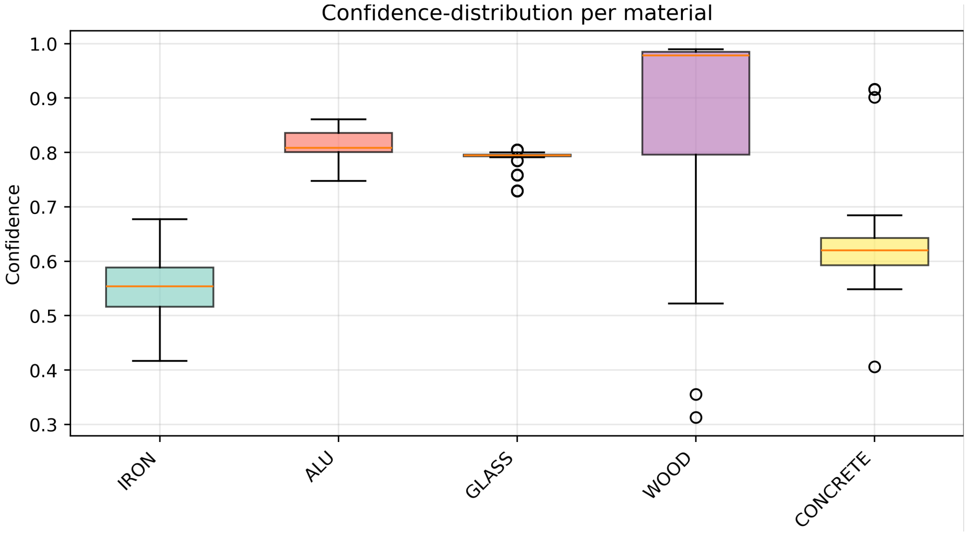
The following graphics show the conducted tests on the known data:

4.3.2 Similar Unseen Data Validation

The following graphics show the conducted tests on the unseen data that was recorded in similar conditions to the known data.

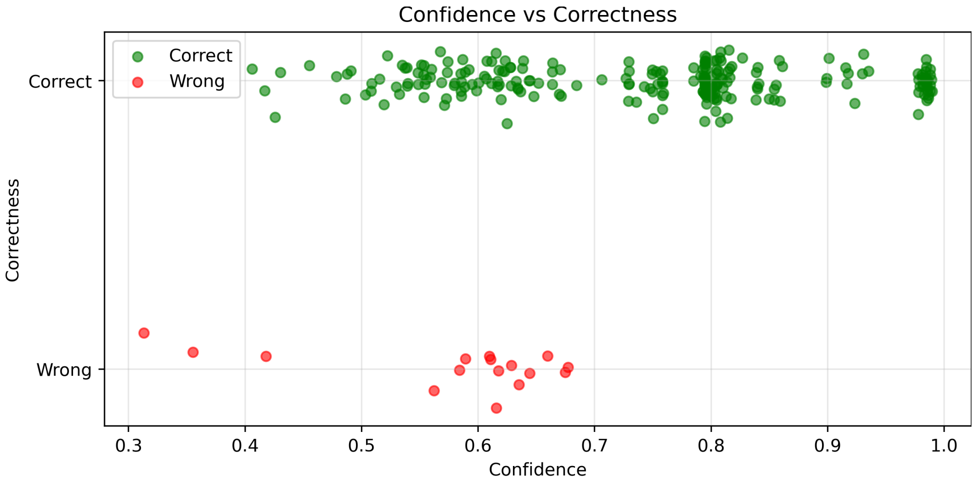


(a) Recall-heatmap of the model for each surface type in absolute values. The colorbar indicates the Recall. The model successfully classified iron 72% of the time, aluminum 100%, plexiglass 100%, wood 96%, and concrete 100%.

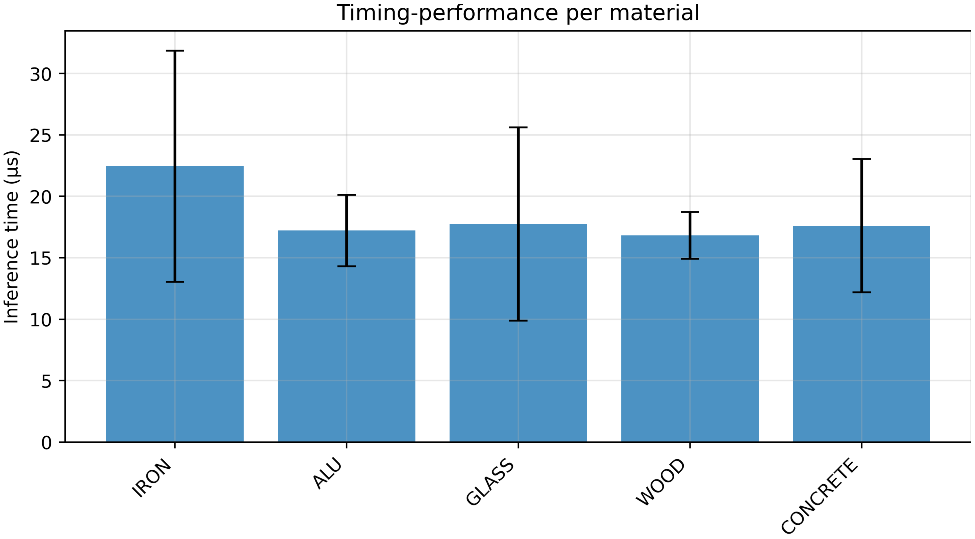


(b) Confidence-distribution diagram of the model for each surface type in absolute values. The orange bar inside each box indicates the median confidence value, while the box shows the interquartile range (IQR), representing the middle 50% of values. Whiskers extend to the minimum and maximum non-outlier values, and circles denote outliers. Confidence values were mainly between 52%-59% for iron, 80%-83.5% for aluminum, around 79% for plexiglass, 80%-98% for wood, and 59%-64% for concrete. Iron showed a wider dispersion from 42%-68%, aluminum from 75%-86%, and plexiglass exhibited limited dispersion with isolated values between 72%-81%. Wood confidence ranged broadly from 52%-99%, with outliers at 31% and 37%. Concrete remained mostly between 56%-69%, with isolated higher values at 90% and 91% and one lower value at 41%. The medians were: Iron 55%, Aluminum 81%, Plexiglass 79%, Wood 98%, and Concrete 63%.

Figure 4.4: Recall-Heatmap and Confidence-Distribution Diagram for Known Data.



(a) Confidence vs. correctness diagram of the model on the known data test set. When the model correctly classified a material, confidence values ranged from 40% to 99%, with notable concentrations around 75%, 80%, and 98%. Additional clusters were observed in the 50%-68% range. For misclassifications, confidence values were mainly between 56% and 68%, with no cases above 68% and three outliers below 56% at 32%, 36%, and 42%.



(b) Timing diagram of the model on the known data test set. The height of the blue bars indicates the mean inference time per material class (in microseconds), while the whiskers represent the standard deviation. Mean inference times are: iron 22.5 μs, aluminum 17.0 μs, plexiglass 17.5 μs, wood 17.0 μs, and concrete 17.5 μs. Standard deviations are: iron 9.5 μs, aluminum 3.0 μs, plexiglass 7.5 μs, wood 2.0 μs, and concrete 5.5 μs.

Figure 4.5: Confidence vs correctness and Timing Diagram for Known Data.

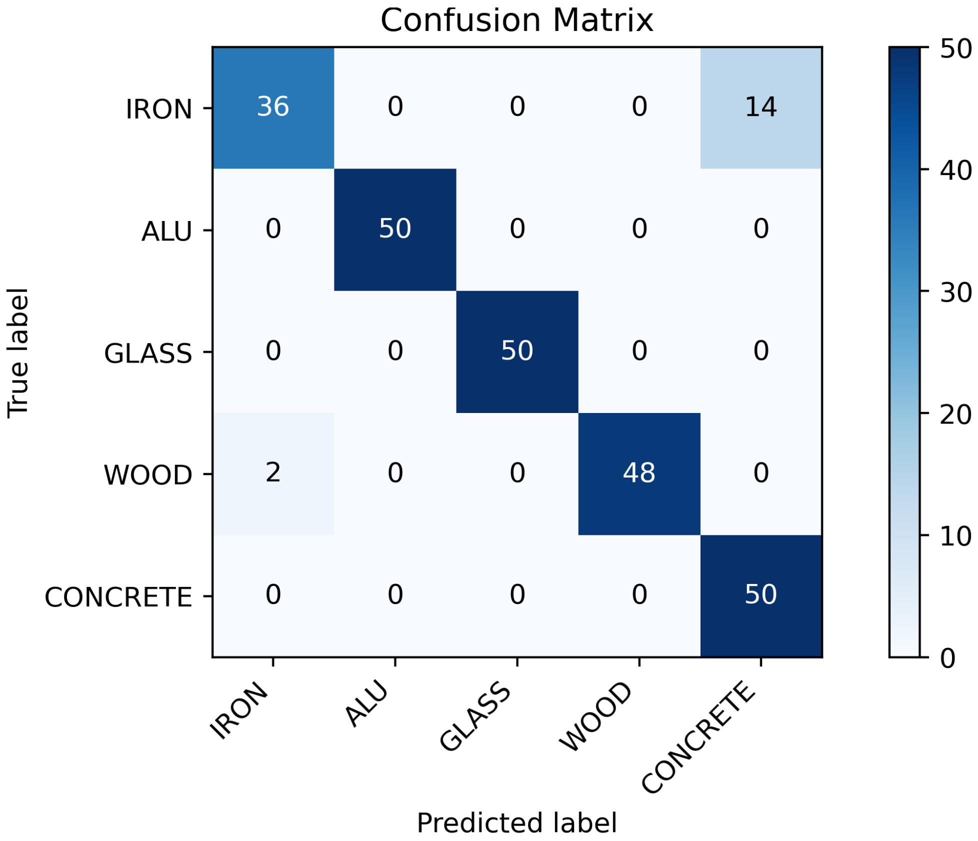
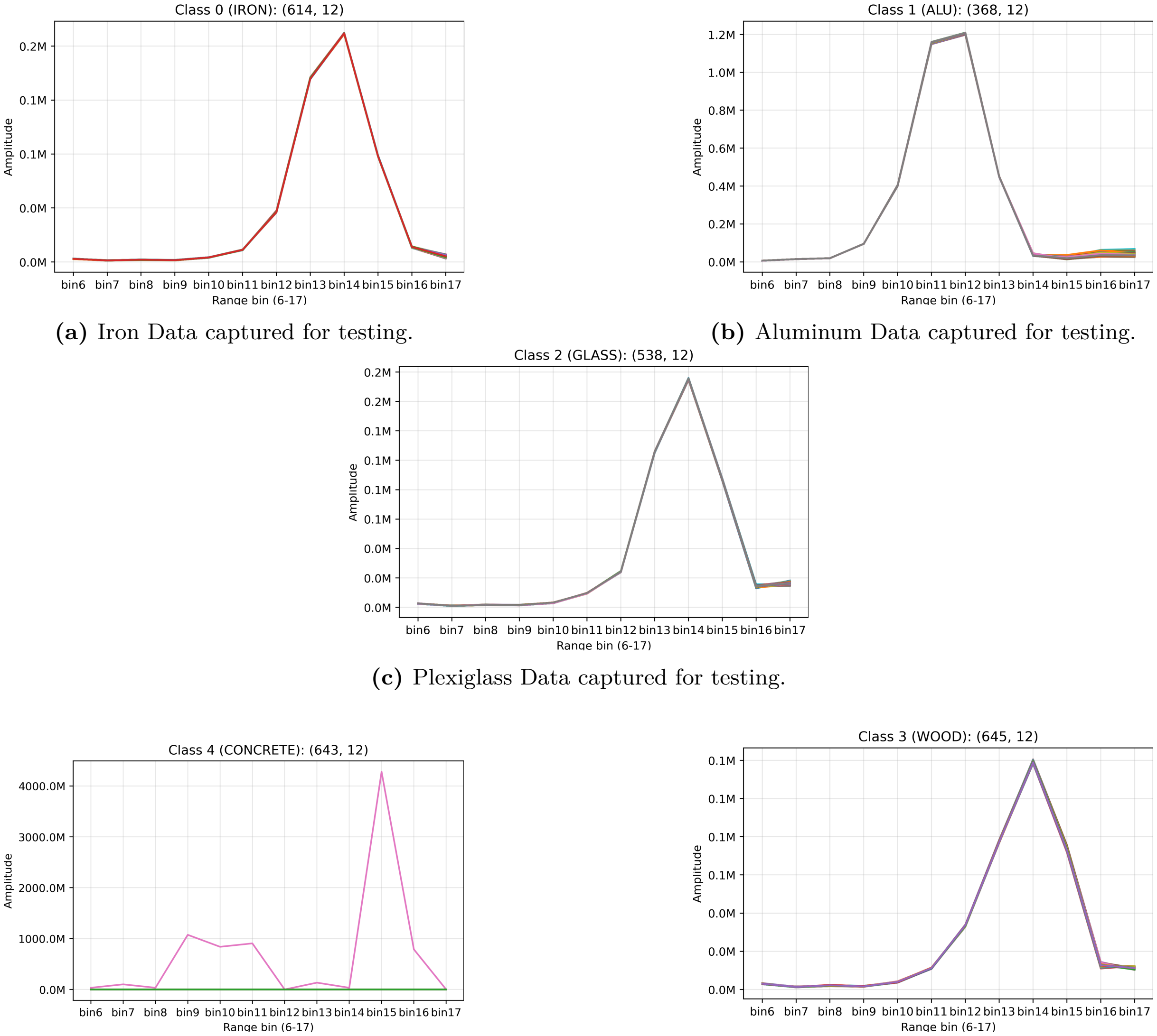


Figure 4.6: Confusion matrix of the model on the known data test set. Rows represent true class labels and columns predicted labels. The diagonal elements indicate correct classifications, while off-diagonal elements represent misclassifications. The model classified aluminum, plexiglass, and concrete with 100% accuracy (50/50), but misclassified 14 instances of iron (all predicted as concrete) and 2 instances of wood (predicted as iron).



(a) Iron Data captured for testing.

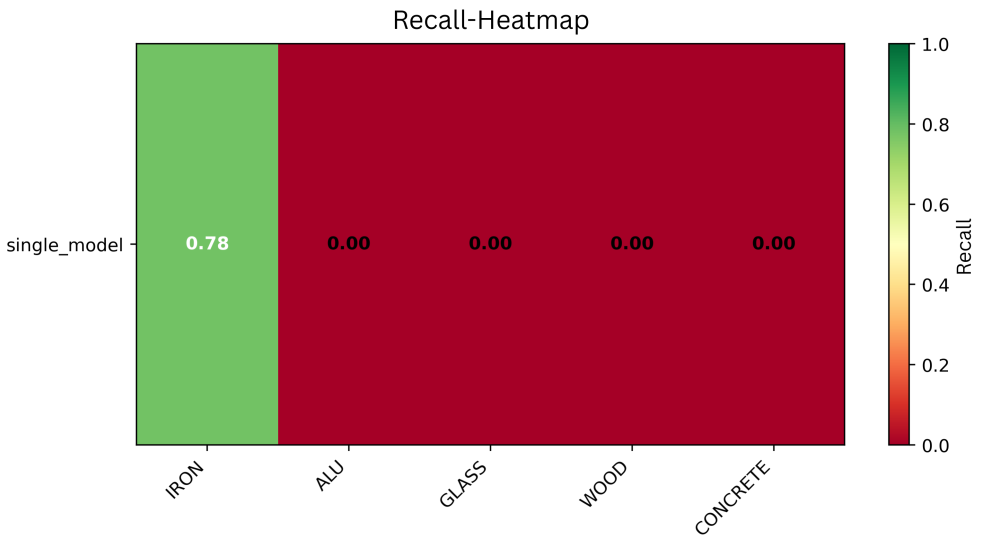
(b) Aluminum Data captured for testing.

(c) Plexiglass Data captured for testing.

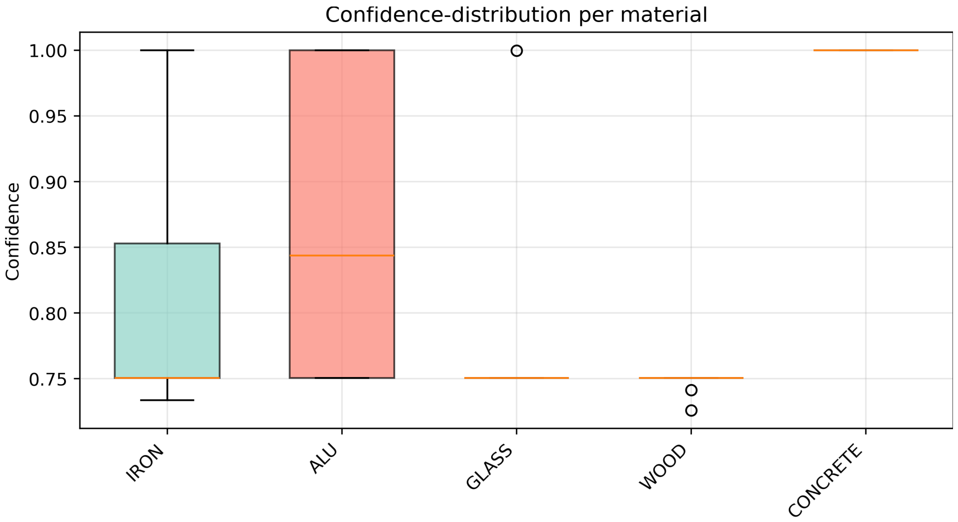
(d) Concrete Data captured for testing.

(e) Wood Data captured for testing.

Figure 4.7: Testing Data captured under similar conditions to the known data.

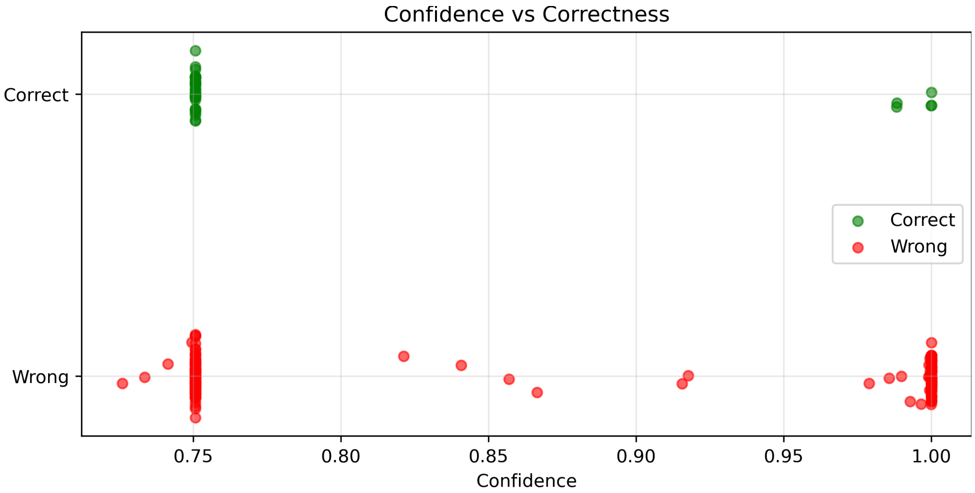


(a) Recall-heatmap of the model under similar but unseen data. The colorbar indicates the absolute Recall values per surface type. The model correctly classified iron in 78% of cases, while all other materials were misclassified (0% Recall).

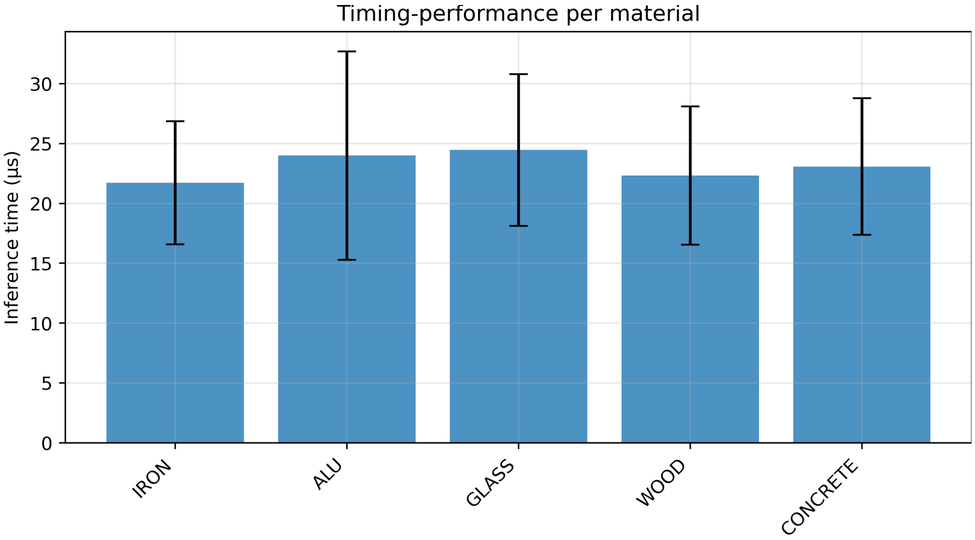


(b) Confidence-distribution diagram of the model under similar but unseen data. The orange line inside each box indicates the median confidence per class, while the box represents the interquartile range (IQR), covering 50% of the confidence values. The whiskers show the full range of values, and circles denote outliers. The confidence was mainly between 75% and 85.5% for iron, between 75% and 100% for aluminum, concentrated at 75% for plexiglass and wood, and fixed at 100% for concrete. The absolute dispersions are: iron (73-100%), aluminum (75-100%), plexiglass (single outlier at 100%), and wood (two outliers at 74% and 75%). Concrete shows no dispersion. The medians are: iron 75%, aluminum 84%, plexiglass 75%, wood 75%, and concrete 100%.

Figure 4.8: Recall-Heatmap and Confidence-Distribution Diagram for Similar Data.



(a) Confidence vs correctness diagram of the model under similar but unseen data. Correct classifications are mostly concentrated at 75%, with additional clusters at 98% and 100%. Misclassifications are mainly concentrated at 75% and 100%, with a few outliers around 73%, 85%, and 92%.



(b) Timing diagram of the model under similar but unseen data. The blue bars indicate the mean inference time per material class (in μs), and the whiskers represent the standard deviation. The mean inference times were: iron 22 μs, aluminum 24 μs, plexiglass 24.5 μs, wood 22.5 μs, and concrete 23 μs. The standard deviations were 5 μs for iron, 8.5 μs for aluminum, 7 μs for plexiglass, 6 μs for wood, and 6 μs for concrete.

Figure 4.9: Confidence vs correctness and Timing Diagram for Similar Data.

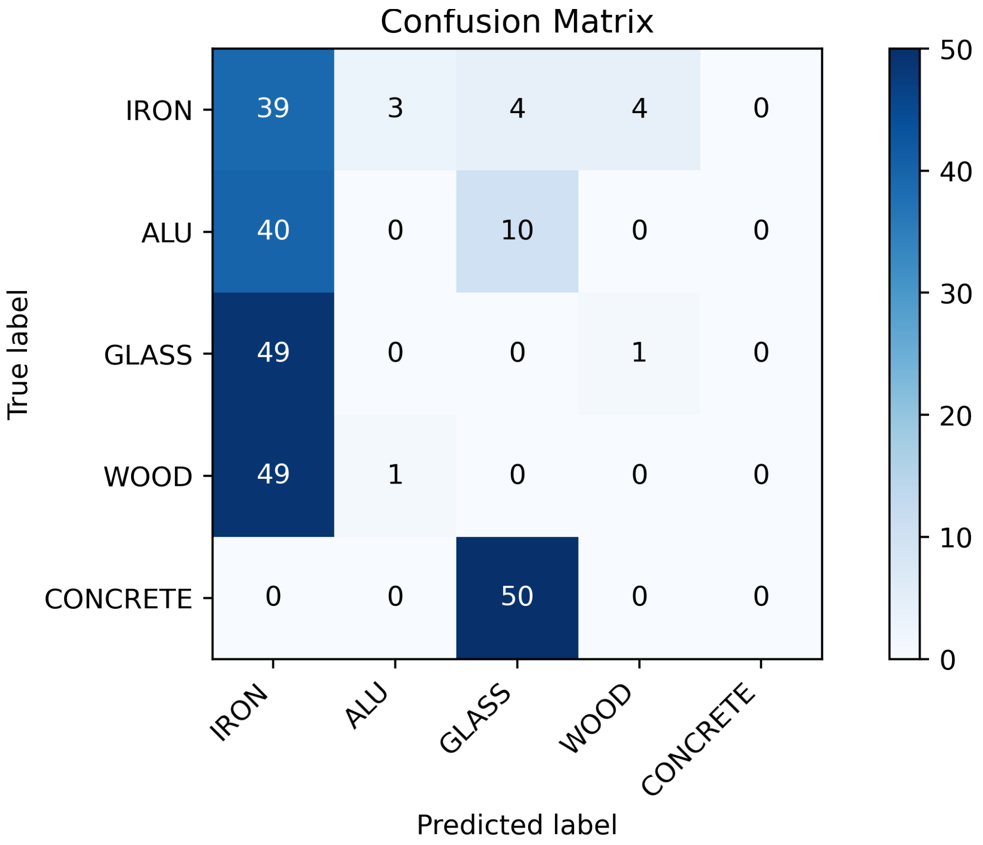
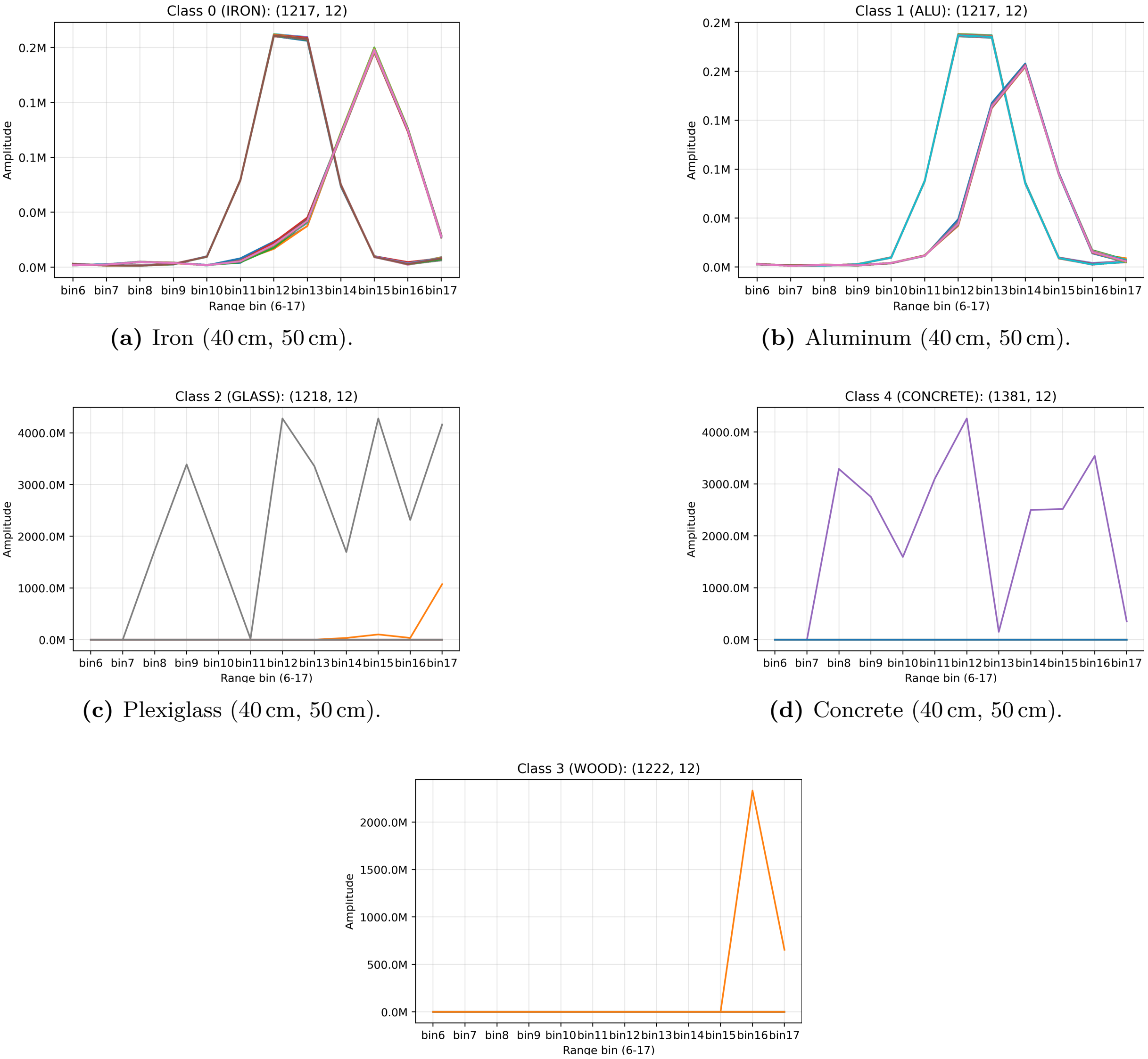


Figure 4.10: Confusion matrix of the model under height variation. Each row corresponds to the true class labels, and each column to the predicted labels. Diagonal elements indicate correctly classified instances, while off-diagonal entries represent misclassifications. The model successfully classified 39 samples of iron, but misclassified aluminum, plexiglass, wood, and concrete in all 50 cases. Specifically, aluminum was misclassified 40 times as iron and 10 times as plexiglass; plexiglass and wood were both misclassified 49 times as iron (with one additional error each as wood and aluminum, respectively). Concrete was misclassified 50 times as plexiglass. Iron errors were distributed as 3 misclassifications into aluminum, 4 into plexiglass, and 4 into wood.

4.3.3 Height Variation Robustness Test

The following graphics show the conducted tests on the unseen data, which was recorded with height variations to the training data.



(a) Iron (40 cm, 50 cm).

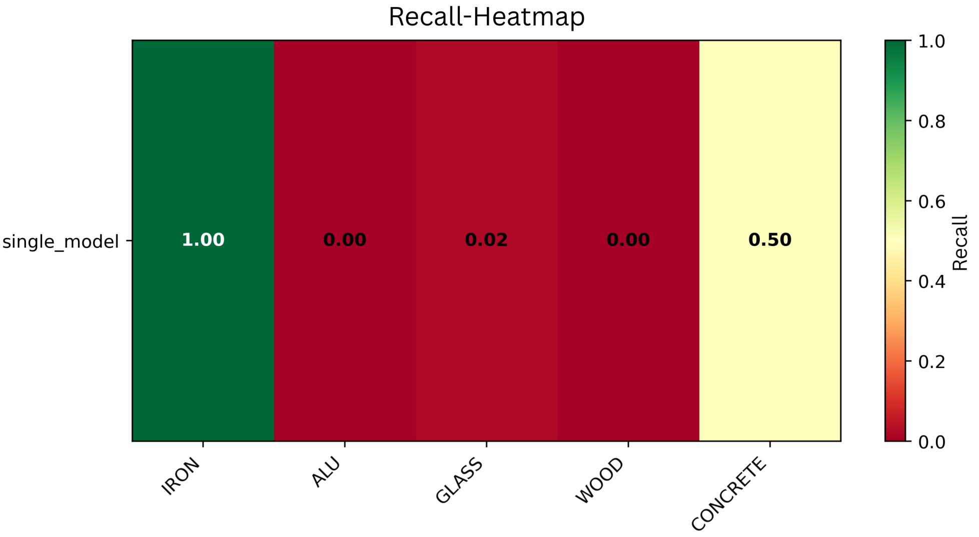
(b) Aluminum (40 cm, 50 cm).

(c) Plexiglass (40 cm, 50 cm).

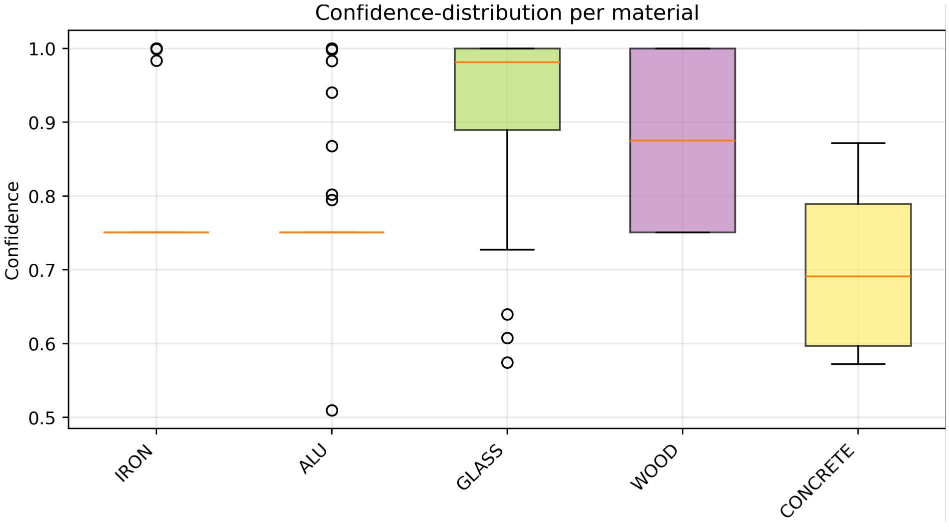
(d) Concrete (40 cm, 50 cm).

(e) Wood (40 cm, 50 cm).

Figure 4.11: Testing data captured with height variations (40 cm and 50 cm distances) compared to the training data.

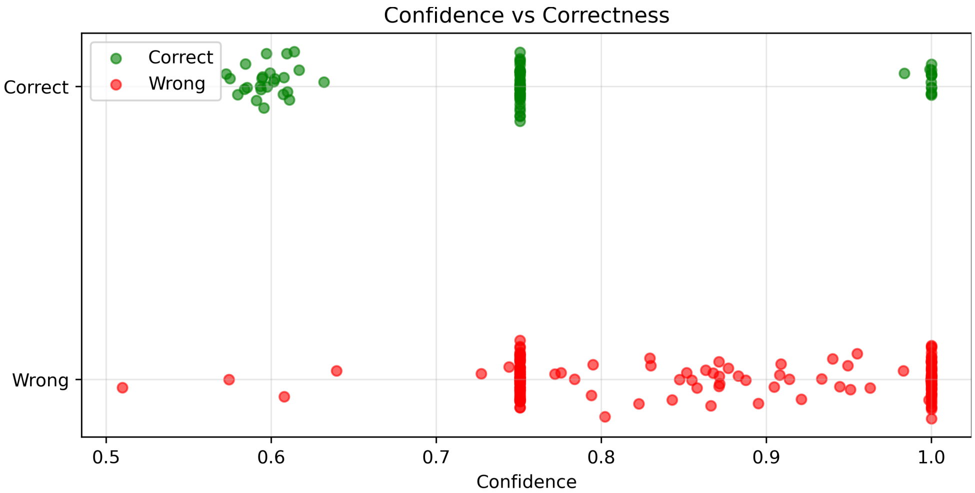


(a) Recall heatmap of the model for each surface type under height variation. The colorbar indicates absolute Recall values. Results show that iron was classified correctly in 100% of cases, aluminum and wood in 0%, plexiglass in 2%, and concrete in 50%. These results highlight the model's strong robustness to distance changes for iron, partial robustness for concrete, and severe performance degradation for aluminum, plexiglass, and wood.

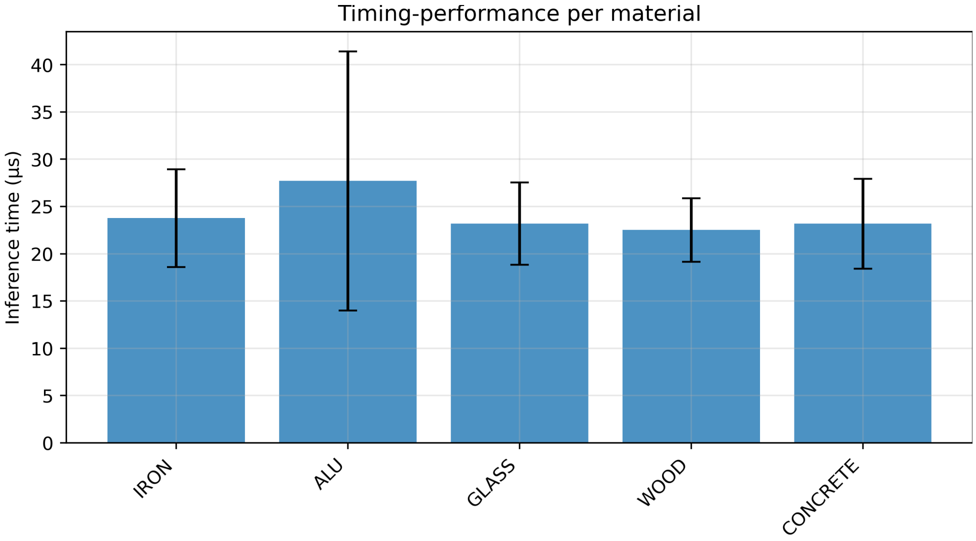


(b) Confidence distribution of the model for each surface type under height variation. The orange line in each box indicates the median confidence, while the box represents the interquartile range (IQR, middle 50% of values). Whiskers denote the full observed range, and circles mark outliers. The results show that iron and aluminum are concentrated at 75% confidence, with iron having two outliers at 98% and 100% and aluminum showing scattered outliers up to 100%. Plexiglass exhibits high median confidence at 98% but a wide dispersion (73-100%), including outliers as low as 58-65%. Wood confidence spans 75-100% with a median of 88%, while concrete has the lowest median (69%) and the widest spread (58-88%), indicating higher uncertainty. Overall, the model tends to maintain high confidence even when predictions are incorrect, highlighting systematic overconfidence in this scenario.

Figure 4.12: Recall-Heatmap and Confidence-Distribution Diagram for Height Variation data.



(a) Confidence versus correctness for the height variation scenario. Correct classifications cluster mainly at 75% and 100% confidence, with a few lower-confidence cases around 55-64%. Misclassifications show the same pattern, also concentrating at 75% and 100%, with scattered outliers near 50-65%. This overlap indicates systematic overconfidence: the model assigns high confidence to both correct and incorrect predictions, limiting its ability to signal uncertainty under distance variations.



(b) The Timing diagram shows the time taken for each classification by the model in microseconds. The height of the blue bars indicates the mean inference-time for each material class. The whiskers represent the standard deviation. The mean inference-time for each class is as follows: iron 24μs, aluminum 27.5μs, plexiglass 23μs, wood 22.5μs, and concrete 23μs. The iron inference time shows a standard deviation of 5μs , aluminum 13μs , plexiglass 5μs, wood 4μs, and concrete 5μs.

Figure 4.13: Confidence vs correctness and Timing Diagram for Height Variation data.

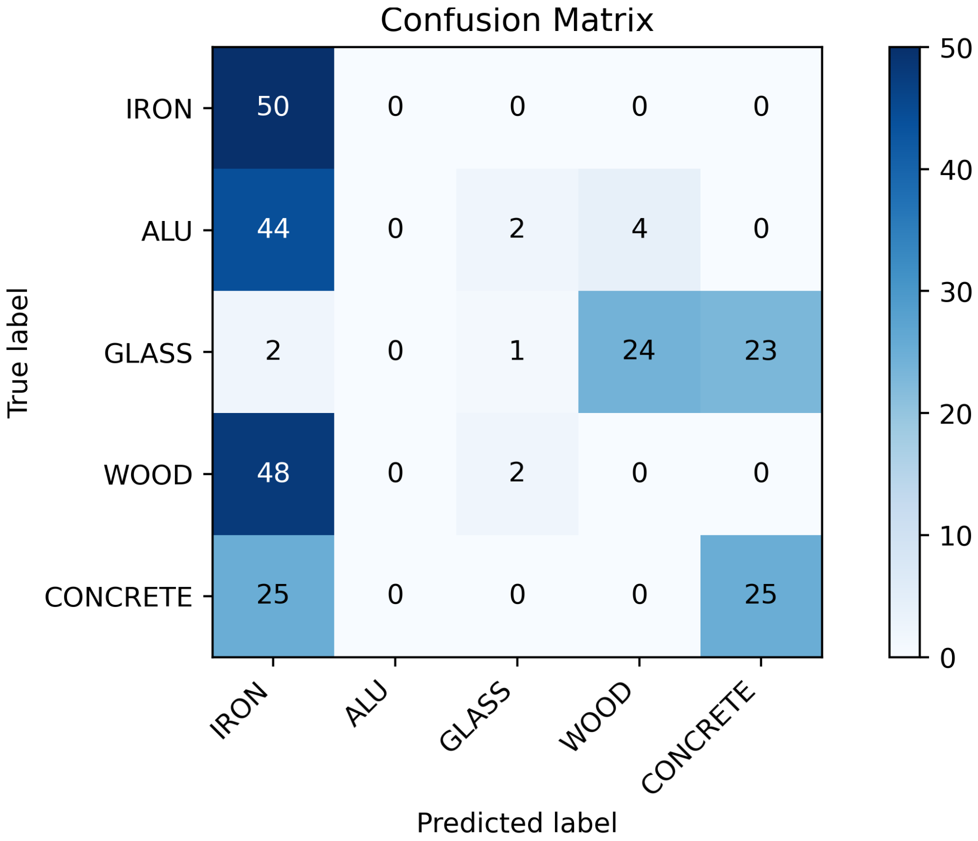
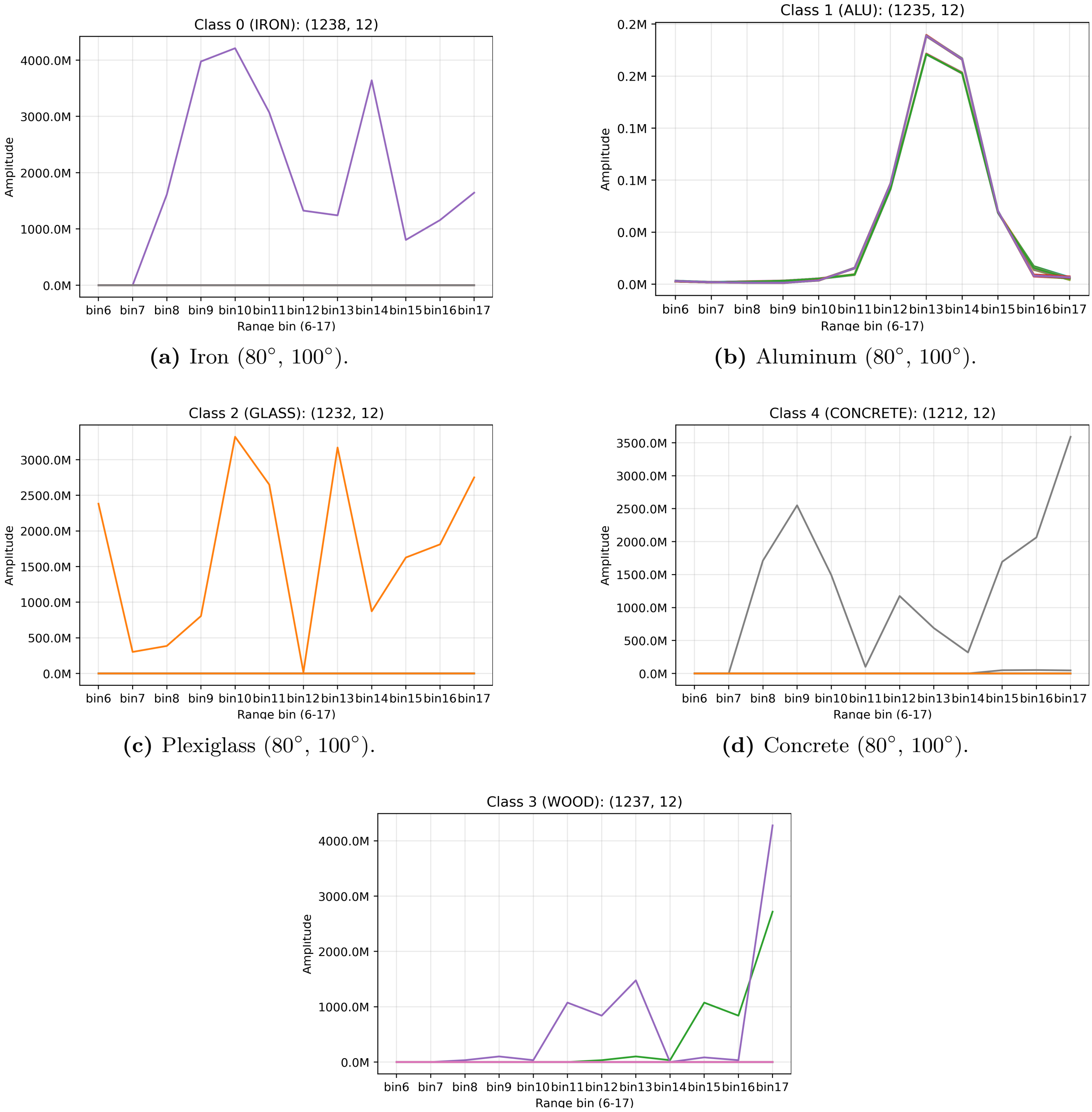


Figure 4.14: Confusion matrix for the height variation scenario. The model failed to correctly classify aluminum and wood (0% recall) and showed severe confusion for plexiglass (98% misclassified) and concrete (50% misclassified, mostly as iron). Iron was the only material classified reliably. These results indicate that changes in sensor-to-surface distance severely reduce discriminability, with most errors collapsing into iron predictions.

4.3.4 Angle Variation Robustness Test

The following graphics show the conducted tests on the unseen data, which was recorded with angle variations to the training data.



(a) Iron (80°, 100°).

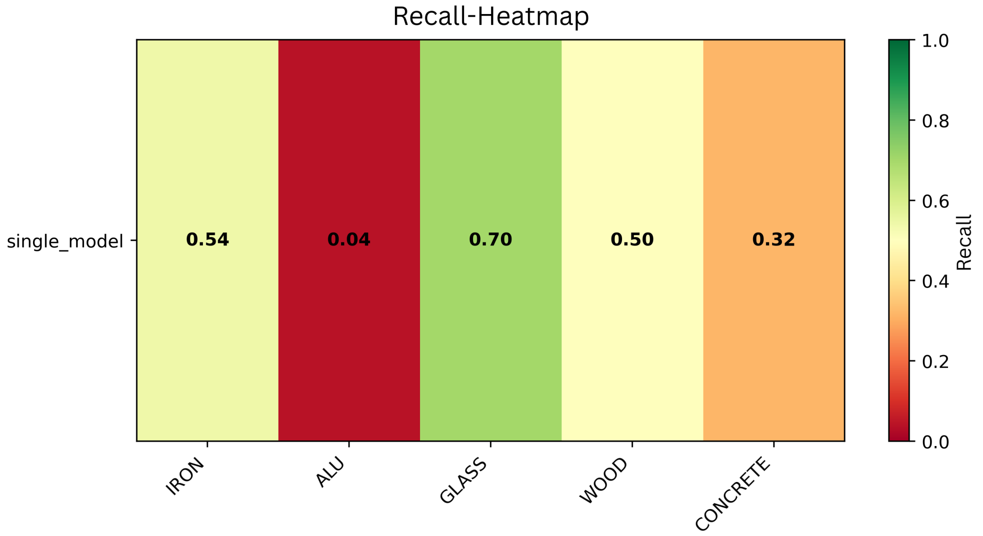
(b) Aluminum (80°, 100°).

(c) Plexiglass (80°, 100°).

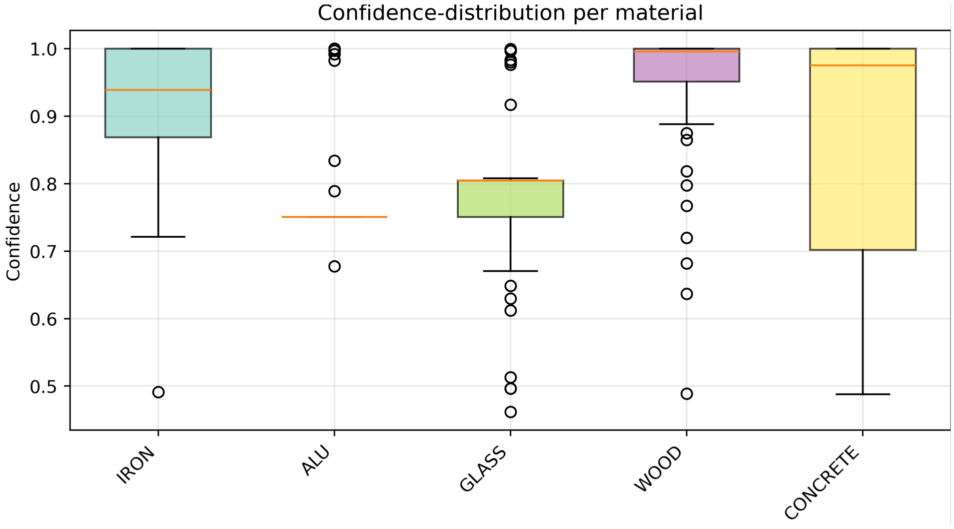
(d) Concrete (80°, 100°).

(e) Wood (80°, 100°).

Figure 4.15: Testing data captured with angle variations (80° and 100°) compared to the training data.

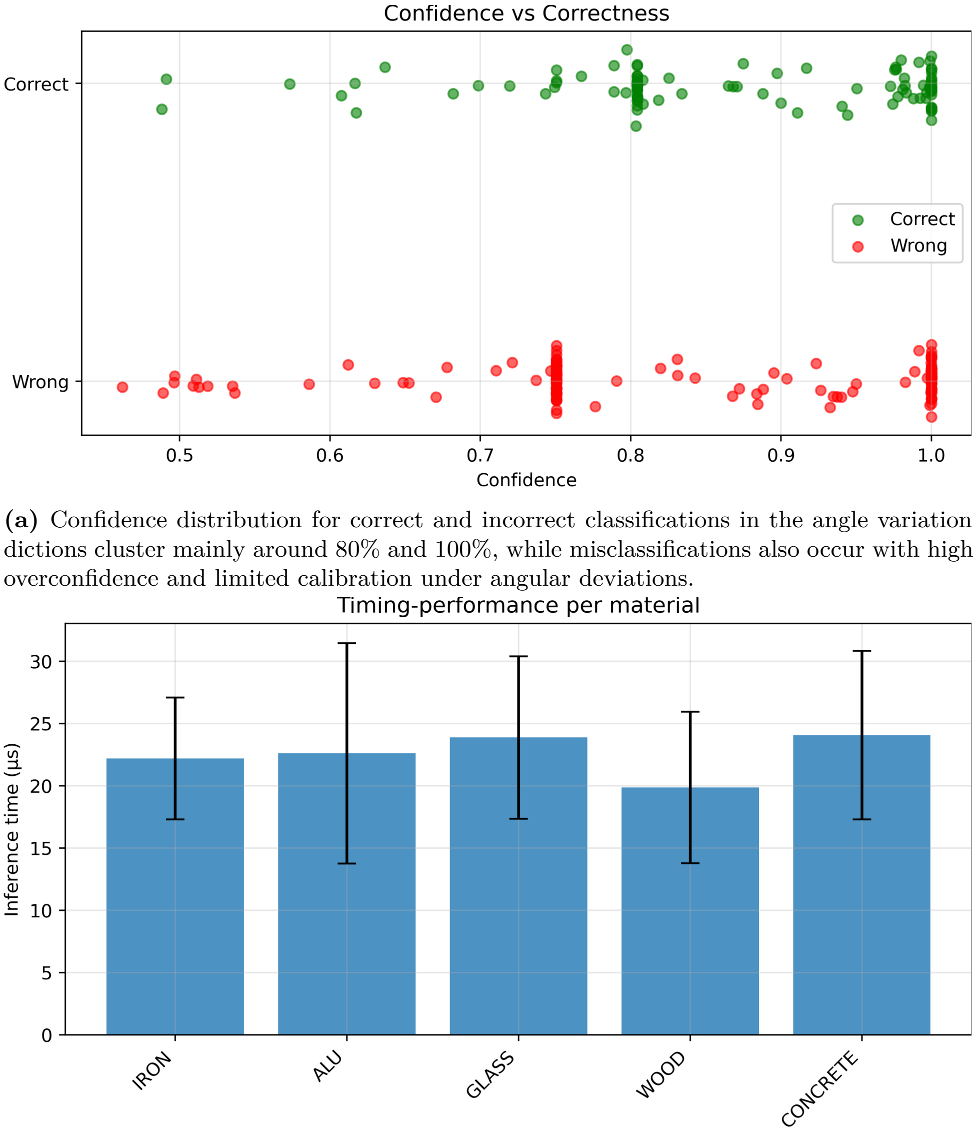


(a) Recall heatmap for each material class under angle variation. The model retained moderate recall for plexiglass (70%), iron (54%), and wood (50%), while aluminum (4%) and concrete (32%) dropped sharply. This indicates that angular changes particularly affect reflective materials such as aluminum and concrete.



(b) Confidence distribution of the model for each material class under angle variation. Most predictions show high confidence, with wood and concrete clustering above 95%, while aluminum remains centered at 75%. Iron and plexiglass display wider dispersion and several low-confidence outliers, indicating that some classes are less stable under angular changes.

Figure 4.16: Recall-Heatmap and Confidence-Distribution Diagram for Angle Variation data.



(a) Confidence distribution for correct and incorrect classifications in the angle variation dataset. Correct predictions cluster mainly around 80% and 100%, while misclassifications also occur with high confidence, indicating overconfidence and limited calibration under angular deviations.

(b) Inference-time distribution for the angle variation dataset. Bars show mean inference times per material class, with whiskers indicating standard deviations. Inference times are consistent across classes (≈20–24 μs), demonstrating stable computational efficiency despite angular variations.

Figure 4.17: Confidence vs correctness and Timing Diagram for Angle Variation data.

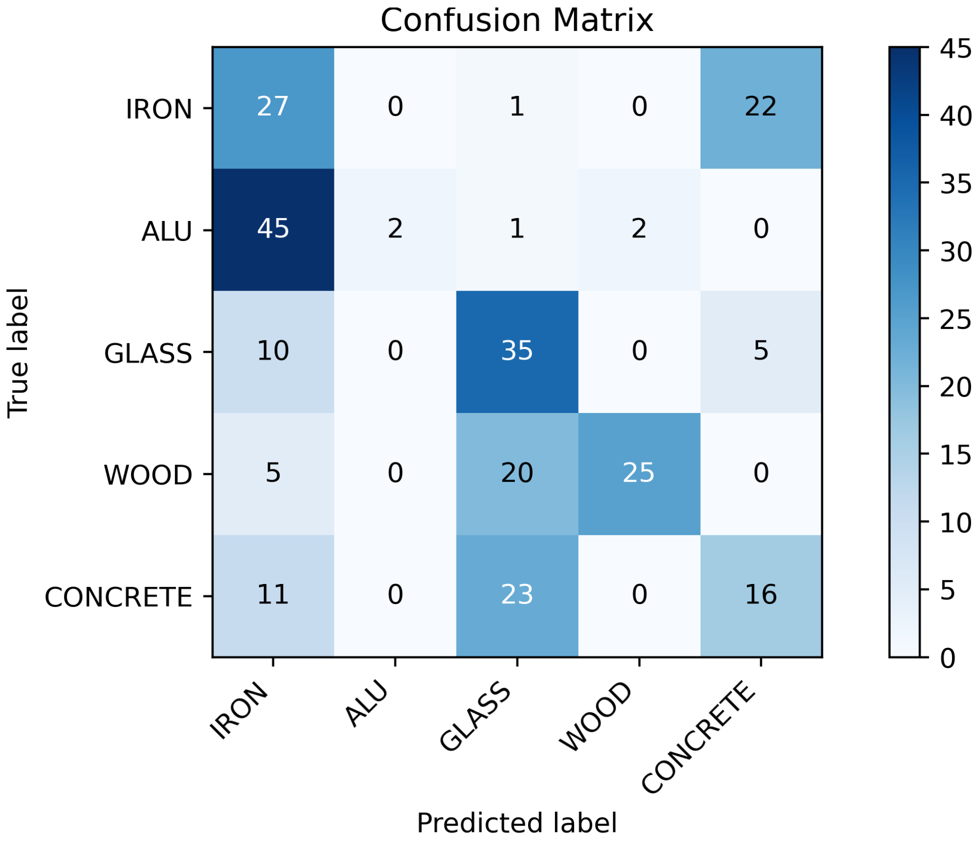


Figure 4.18: Confusion matrix of the model performance on the angle variation test dataset. Rows correspond to true class labels and columns to predicted class labels. Diagonal elements indicate correct classifications, while off-diagonal entries represent misclassifications. The results reveal frequent confusions between aluminum and iron, as well as between concrete and plexiglass, highlighting the model's reduced robustness under angular deviations.

Chapter 5

Performance Evaluation

5.1 Performance Metrics and Analysis

The purpose of this section is to consolidate and interpret the results obtained from the different testing scenarios described in Section 3.2. The analysis aims to identify performance trends and key differences across the four evaluation settings: *Known Data*, *Similar Unseen Data*, *Height Variation*, and *Angle Variation*. The discussion focuses on four central performance metrics—Recall, Precision, Confidence, and Inference Time— and examines how each of them is affected under different testing conditions.

5.1.1 Recall Analysis

Recall is a key metric for evaluating a classification model, as it measures the proportion of correctly identified positive samples relative to the total number of actual instances of a given class 2.9. High Recall indicates that the model is effective at detecting and recognizing material types without overlooking relevant cases.

Test Data Type	**Iron**	**Aluminum**	**Plexiglass**	**Wood**	**Concrete**
Known Data	0.72	1.00	1.00	0.96	1.00
Similar Unseen Data	0.78	0.00	0.00	0.00	0.00
Height Variation	1.00	0.00	0.02	0.00	0.50
Angle Variation	0.54	0.04	0.70	0.50	0.32

Table 5.1: Overview of model Recall across test data types and material classes

The table 5.1 shows that the model achieved high Recall in the Known Data scenario, with only iron and wood being misclassified in some cases. Iron achieved the lowest Recall at 0.72 (72%), which is consistent with the training confusion matrix 4.3, where iron was also the most challenging class to identify (314 misclassifications out of 1144 samples, Recall of 0.726). Wood slightly outperformed expectations with a Recall of 0.96 compared to 0.94 during training. Overall, the model behaved as expected under known conditions, confirming the effectiveness of the training process and the model's functionality.

In the Similar Unseen Data scenario, performance dropped sharply for all materials except iron, which retained a Recall of 78%. All other classes were completely misclassified, indicating poor generalization and overfitting to radar signatures seen during training. This shows that the learned features were strongly tied to the training environment and not robust against even small distributional shifts.

For the Height Variation tests, iron again dominated performance with a perfect Recall of 100%, while aluminum and wood dropped to 0%. Concrete achieved 0.50 (50%), and plexiglass slightly improved compared to the Similar Unseen Data scenario with a Recall of 0.02 (2%). These results indicate selective robustness: iron, with its strong radar cross section (RCS), remains relatively unaffected by distance changes, while concrete benefits from its density and thickness, which provide a distinct RCS. In contrast, plexiglass and wood exhibit weaker or more ambiguous RCS signatures, making them harder to classify under changing distances. The degradation in general is explained by radar equation 2.3, as varying distances alter signal intensity and range-bin energy distributions, reducing feature separability. Since the model was trained only on range-bin energy data, it failed to generalize across these variations.

The Angle Variation scenario resulted in a more uniform decrease across classes. Plexiglass, wood, and iron saw moderate declines of 20-40% compared to known data, while aluminum dropped to just 4% Recall and concrete decreased by 68%. This pattern reflects the impact of sensor orientation: tilting from the nadir position shifts the specular reflection point, causing reduced power in the primary range bin and redistribution of energy into side lobes. For strongly specular materials such as aluminum and iron, this effect is severe, as reflections can be redirected away from the receiver, producing feature patterns very different from training conditions. More diffuse scatterers like plexiglass and wood were less affected, as their broader scattering preserved aspects of the learned features.

Interestingly, iron still performed better than aluminum, suggesting more robust feature representations and better generalization due to richer training variation. Concrete's poor performance is attributable to its heterogeneous surface, which scatters energy more unpredictably, reducing signal intensity in key range bins under angular deviations.

Taken together, the Recall results across all scenarios confirm the model's reliability under controlled, training-like conditions and its strong vulnerability to distributional shifts. In the Known Data scenario, aluminum, plexiglass, and concrete achieved perfect Recall, with only iron and wood showing minor confusion. In contrast, the Similar Unseen Data scenario saw a near-complete collapse of performance, with only iron retaining accuracy. The Height Variation tests showed selective robustness (iron 100%, concrete 50%), but almost complete failure for other classes. The Angle Variation scenario produced more balanced but still degraded results, with plexiglass, wood, and iron retaining 50-70% Recall, while concrete and aluminum dropped sharply.

Overall, the Recall patterns show that the model is reliable only when the acquisition geometry matches training conditions. Robustness is highest for iron due to its consistently strong RCS, whereas aluminum and wood are most sensitive to geometric and environmental variations.

5.1.2 Confidence Analysis

Examining a model's confidence behaviour provides valuable insights into the reliability and adaptability of its predictions. High confidence in correct classifications indicates that the model is making well-supported decisions, whereas high confidence in incorrect classifications reveals potential overfitting or poor calibration. Analysing confidence distributions across different testing scenarios allows the identification of conditions under which the model becomes overconfident, loses discriminative ability, or fails to adjust its certainty when confronted with unfamiliar inputs. Such observations are crucial for assessing the model's robustness, reliability, and overall functionality.

Test Scenario	**Iron**	**Aluminum**	**Plexiglass**	**Wood**	**Concrete**
Known Data	55	81	79	98	63
Similar Unseen Data	75	84	75	75	100
Height Variation	75	75	98	88	69
Angle Variation	94	75	81	99	97

Table 5.2: Median confidence values (%) per material class across all test scenarios

The confidence plots and Table 5.2 provide additional insight into the model's decision-making process and the reliability of its predictions across different testing scenarios.

For known data, the confidence distributions are generally high for most classes, with aluminum, plexiglass, and wood showing median confidences above 79%. Aluminum and plexiglass in particular exhibit relatively narrow interquartile ranges as indicated in Figure 4.4b. Iron displays a broader confidence spread and overall lower confidence, ranging from 42% to 68%, reflecting occasional misclassifications with concrete. Concrete itself shows a moderate median confidence around 63% but includes several high-confidence outliers, indicating that the model can sometimes predict this class with strong certainty.

The plot in Figure 4.5a further suggests a correlation between confidence and correctness, with higher-confidence predictions generally aligning with correct classifications (often at 98%). False classifications tend to occur at lower confidence levels, particularly around 60% and not exceeding 68%, showing that the model does not typically make highly confident misclassifications. This suggests a degree of stability with known data.

In the Similar Unseen Data scenario, confidence values remain high even when predictions are incorrect. This is particularly evident for concrete but also for aluminum, plexiglass, and wood, which were exclusively misclassified with confidences around 75-100%. Such behavior indicates overconfidence in non-representative inputs, suggesting that the model relies heavily on features specific to the training distribution and fails to adjust its certainty when encountering novel patterns. Another noteworthy observation is that plexiglass, wood, and concrete show little to no dispersion in confidence values, indicating that the model is unable to adapt to unseen data and is not making nuanced classifications.

Iron shows higher confidence values around 75-85% with a few outliers at 100%, which is an improvement compared to its performance on known data and may suggest a more robust feature representation for iron.

The plot in Figure 4.9a illustrates the overall confidence trend in unseen data, showing that although overall confidence is high, it is predominantly clustered at two levels, 75% and 100%, regardless of prediction correctness. This further supports the conclusion that the model is overconfident and struggles to make adaptive classifications.

For the Height Variation tests, the confidence levels for misclassified samples are again relatively high, often in the 75-100% range, despite the drop in recall. Iron continues to be predicted with consistent confidence, showing little dispersion or variance in Figure 4.13a, reflecting robustness to height changes. Aluminum shares this pattern despite being misclassified in every instance. Plexiglass and wood also display elevated confidence in incorrect predictions, reinforcing the overconfidence trend observed in the similar-data scenario. Notably, concrete, which achieved the second-highest recall, shows the lowest median confidence (69%) and a wide dispersion, indicating that while the model captures some of its features, it remains uncertain with altered data.

The confidence-correctness chart in Figure 4.13a, similar to the unseen data scenario, shows two highly frequent confidence levels at 75% and 100%. A few correct classifications occur around 60%, exclusively for concrete predictions. Nearly all other correct predictions are located at either 75% or 100% confidence, which is also where most false predictions occur. This suggests systematic behavior, likely caused by the model exploiting specific training features and consistently converging on the same predictions.

The angle variation tests (Figure b) exhibit a wider range of confidence values and a greater number of outliers across all classes. Iron, plexiglass, wood, and concrete retain high medians above 80%, while aluminum shows a concentrated cluster at 75%, as observed in other scenarios, but with several extreme outliers approaching 100%. Again, many incorrect predictions also have confidence levels above 75%, highlighting that the model does not generalize well under angular deviations.

In general, confidence is higher in this scenario than in others, with the majority of predictions clustering above 90%. The high confidence values, combined with improved precision compared to other unseen data scenarios, suggest that the model is better able to generalize to angular variations than to height variations.

The confidence-correctness diagram in Figure 4.17a shows a more dispersed pattern compared to the height variation tests, with correct predictions spread across a wider confidence range. Although peaks remain at 75% and 100% for false predictions, the presence of both correct and false classifications at lower confidence levels indicates that the model can make more nuanced distinctions between materials under angular variations. The peak for correct classifications at 75% has shifted toward 80%, while the 100% peak is more spread out than in other scenarios. This suggests that the model becomes more versatile in its predictions, handling angular changes better than height variations or other unseen data.

Overall, the confidence analysis reveals systematic patterns:

- The model tends to exhibit high confidence levels for correct predictions, particularly in known data scenarios.
- Wrong predictions are often made with high confidence, showing that the model does not adapt well to unknown or altered scenarios.
- The model's ability to generalize varies across types of variation, with angular changes handled more effectively than height variations.

This behavior highlights the importance of a dynamic and diverse training dataset and strategy to improve the relationship between recall and confidence, especially in scenarios with unseen or altered data.

5.1.3 Inference-Time Analysis

Evaluating inference times is essential for assessing a model's suitability for real-time or resource-constrained applications. Consistently low and predictable inference times indicate that the model can deliver rapid predictions without exceeding computational limits, which is particularly important for embedded systems and robotics. Analyzing inference times across different scenarios also helps identify performance bottlenecks, account for variations in computational load, and ensure that the model meets the latency requirements of its intended deployment environment.

Test Scenario	**Iron**	**Aluminum**	**Plexiglass**	**Wood**	**Concrete**
Known Data	22.5	17.0	17.5	17.0	17.5
Similar Unseen Data	22.0	24.0	24.5	22.5	23.0
Height Variation	24.0	27.5	23.0	22.5	23.0
Angle Variation	22.0	22.5	24.0	20.0	24.0

Table 5.3: Mean inference time (μs) per material class across all test scenarios

The table 5.3 presents the mean inference times for each material. The model maintains relatively consistent inference times across different test scenarios, with only minor variations. Inference times for known data are slightly lower than for the other scenarios, which is expected since the model is more familiar with the training data and can therefore make predictions with less computational overhead. The other scenarios differ only marginally, with the height variation scenario showing the highest mean inference time of 27.5 μs for aluminum. Inference times for iron are more stable compared to the other materials, consistent with the overall more stable accuracy and confidence of this material observed in 5.1.2 and 5.1. Wood exhibits slightly lower inference times than the other materials, which may be attributed to its simpler RCS profile. No clear relation between recall or confidence and inference time is apparent, suggesting that these factors do not significantly influence the model's inference speed.

Overall, the model demonstrates efficient and predictable inference performance in the range of 17.0 to 27.5 μs across different materials and scenarios, making it technically suitable for real-time and resource-constrained applications.

5.1.4 Precision Analysis

Precision 2.7 measures the proportion of correctly classified samples among all predictions made for a given class. It reflects the model's ability to avoid false positives and is particularly relevant when misclassifying a sample into a certain class could lead to critical errors. High precision indicates that predictions for a class are trustworthy, while low precision suggests that the model frequently assigns the label incorrectly.

Scenario	**Iron**	**Aluminum**	**Plexiglass**	**Wood**	**Concrete**
Known Data	94.7	100.0	100.0	100.0	78.1
Similar Unseen	22.0	0.0	0.0	0.0	0.0
Height Variation	29.6	0.0	88.9	0.0	100.0
Angle Variation	27.6	100.0	43.8	92.6	37.2

Table 5.4: Precision (%) per material class across all test scenarios

In the controlled Known Data scenario, precision was high for all classes, with aluminum, plexiglass, and wood achieving perfect scores of 100%. Iron also showed high precision at 94.7%, with only a small number of false positives, while concrete reached 78.1%, affected by misclassifications from iron. These results confirm that the model is highly accurate when operating on familiar input distributions and can produce reliable class predictions under training-like conditions.

The Similar Unseen Data scenario revealed a strong bias toward predicting iron, which achieved a precision of 22.0% despite dominating the predictions. All other classes recorded 0% precision, meaning that no samples predicted as these classes were correct. This behavior indicates that the model likely collapses to a single-class output when confronted with data slightly outside its training distribution, severely compromising its discriminative ability.

For the Height Variation tests, precision dropped sharply for most classes compared to the known data. Aluminum and wood recorded 0%, while iron achieved 29.6%, indicating frequent false positives from other classes being predicted as iron. In contrast, plexiglass maintained a high precision of 88.9%, and concrete achieved 100%, suggesting that their radar signatures remained distinctive despite the change in sensor-to-surface distance. The uneven impact across classes highlights that vertical displacement affects materials differently in terms of signal separability.

In the Angle Variation scenario, aluminum and wood retained high precision scores of 100% and 92.6%, respectively, while plexiglass (43.8%) and concrete (37.2%) were moderately affected. Iron, however, dropped to 27.6%, suffering from a large number of false positives originating from other materials. This pattern suggests that certain materials preserve unique features even under angular changes, whereas others lose distinctiveness, increasing the likelihood of misclassification.

Across all test scenarios, precision results show that the model is reliable under training-like conditions with known data, but struggles significantly when facing environmental shifts. In the Known Data scenario, all classes except concrete achieved near-perfect precision, confirming the separation in the feature space for familiar inputs, as indicated by the training graphs 4.2 and 4.3. However, Similar Unseen Data caused a near-complete collapse of class discrimination, with only iron achieving a low but non-zero precision. The Height Variation tests revealed uneven robustness, with plexiglass and concrete maintaining high precision while other classes dropped to near zero. In the Angle Variation scenario, aluminum and wood retained high precision, whereas iron, plexiglass, and concrete experienced substantial drops due to increased false positives. Overall, these results indicate that while the model can be highly precise in ideal conditions or with known data, its false positive rate rises sharply, especially for iron under geometric or data distribution changes.

5.2 Overall Results

This section reports macro-averaged Precision, Recall, F1-score, Confidence, and Inference Time across the five material classes for each testing scenario. Macro values are computed as the unweighted mean of the per-class scores for the five material classes (iron, aluminum, plexiglass, wood, concrete). The total number of samples per class in each scenario was 50, resulting in 250 predictions per scenario. Macro-F1 is computed as

$$F1_{\text{macro}} = \frac{2 \cdot P_{\text{macro}} \cdot R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}}.$$

The F1-score 2.11 is the harmonic mean of Precision and Recall. It combines both metrics into a single value, providing a balanced measure of a model's accuracy that accounts for both false positives and false negatives.

Scenario	**Prec. (%)**	**Rec. (%)**	**F1 (%)**	**Conf. (%)**	**Inf. Time (μs)**
Known Data	94.6	94.0	94.3	75.2	18.3
Similar Unseen Data	4.4	30.0	7.7	81.8	23.2
Height Variation	43.7	30.8	36.1	81.0	24.0
Angle Variation	60.2	43.0	50.2	89.2	22.5

Table 5.5: Macro-averaged metrics across all test scenarios

The evaluation shows that the highest macro-F1 score was achieved in the Known Data scenario, with values above 94% for both macro-precision and macro-recall. Performance dropped notably in the Height Variation and Angle Variation tests, with macro-F1 scores of 36.1% and 50.2%, respectively. Their macro-precision and macro-recall values decreased to 43.7% and 30.8% or 60.2% and 43.0% in a similar manner. The Similar Unseen Data scenario resulted in the lowest macro-F1 of 7.7%, caused by a sharp decline in macro-precision despite a less significant drop in macro-recall values. Macro-confidence values remained comparatively high across all scenarios, ranging from 75.2% to 89.2%, while macro-inference times were consistently low between 18.3μs and 24.0μs.

5.3 Error and Failure Case Analysis

In this section, the error behaviour of the model is analyzed. The focus is on understanding the types of errors made by the model, their frequency, and potential causes. This analysis is crucial for identifying areas of improvement and guiding future research efforts. First, the analysis of the errors is conducted for each scenario separately, examining the specific challenges and misclassifications encountered in each case. In the end, a summary of the overall error behaviour is provided, highlighting the key findings and patterns across the different scenarios.

5.3.1 Known Data Error Analysis

With high precision and recall, the model showed robust and reliable performance for known data.

True Label	Predicted Label	% of class Predictions
IRON	CONCRETE	28.0
WOOD	IRON	4.0

Table 5.6: Misclassification rates (%) per pair — Known Data

This table shows the misclassifications made by the model in the Known Data scenario. The most significant misclassification occurred between iron and concrete, with 28.0% of iron data classified as concrete. Additionally, wood was misclassified as iron in 4.0% of the cases. These misclassifications highlight some of the challenges the model faces even in familiar contexts. It is evident that the model struggles to distinguish between iron and concrete when presented with iron data, as seen in 4.3 and 4.6. This may be due to the comparatively high RCS of the concrete sample relative to the other materials and their naturally occurring features. The concrete sample's RCS is higher than expected because of its larger dimensions (120×120 mm compared to 100×100 mm for the other surface samples).

Wood and iron, however, have very different RCS profiles, and their surface samples are of equal size. Combined with the high confidence of wood predictions in the Known Data scenario, this misclassification is unexpected and warrants further investigation. A likely explanation is poor or indistinguishable training data, where wood shares overlapping features with iron due to environmental variations during data collection.

Overall, the model performs well in the Known Data scenario, with the exception of the significant confusion between iron and concrete, which hinders overall performance. This behavior is consistent with the training results and was therefore expected.

5.3.2 Similar Unseen Data Error Analysis

In the Similar Unseen Data scenario, the model faced greater challenges, resulting in a substantial drop in performance metrics.

True Label	**Predicted Label**	**% of class Predictions**
IRON	ALU	6.0
IRON	GLASS	8.0
IRON	WOOD	8.0
ALU	IRON	80.0
ALU	GLASS	20.0
GLASS	IRON	98.0
GLASS	WOOD	2.0
WOOD	IRON	98.0
WOOD	ALU	2.0
CONCRETE	GLASS	100.0

Table 5.7: Misclassification rates (%) per pair — Similar Unseen Data

The confusion matrix 4.10 and Table 5.7 reveal a complex pattern of misclassifications, with certain materials being consistently misidentified. Notably, iron accounts for the majority of misclassifications, particularly with plexiglass and wood, but also with aluminum. These three materials do not all share comparable RCS profiles, which points to shortcomings in the model's generalization capabilities. Additionally, concrete is exclusively misclassified as glass, with a rate of 100%. This suggests that the model struggles to differentiate between these two materials despite their distinct RCS profiles. Together with the other misclassifications—such as iron being misclassified 22% of the time, the highest false-classification rate—this indicates overfitting to the training data.

Despite these misclassifications, the model remains highly confident in its predictions, as evidenced by the consistently high confidence scores reported in Table 5.2. This overconfidence likely stems from the model's reliance on narrow feature patterns present in the training data, which do not generalize well to unseen examples and further underscore the issue of overfitting.

Overall, the model proved vulnerable in the similar-but-unseen data scenario, with errors arising from the limited diversity of the training data. The error types and frequencies show no clear relation to the training results, reinforcing the conclusion of overfitting and a lack of adaptability.

5.3.3 Height Variation Error Analysis

In the tests with height variations, the model exhibited similar challenges in accurately classifying materials.

The confusion matrix 4.14 and Table 5.8 highlight the difficulties faced by the model when dealing with height variations. Iron is again responsible for the majority of misclassifications, particularly with aluminum and wood, but also with concrete. While the challenges with concrete and aluminum may be related to comparable RCS profiles of these surface samples, wood should in principle be distinguishable from iron due to its very different RCS characteristics. Considering that some datasets with height variations were also used for training, this outcome further indicates overfitting issues and a lack of generalization, as already observed in the previous experiments. However, the variation in misclassifications increased slightly, suggesting that the model was able to generalize marginally better than in the unseen data scenario, albeit without meaningful improvements in precision or recall.

True label	**Pred label**	**% of class Predictions**
ALU	IRON	88.0
ALU	GLASS	4.0
ALU	WOOD	8.0
GLASS	IRON	4.0
GLASS	WOOD	48.0
GLASS	CONCRETE	46.0
WOOD	IRON	96.0
WOOD	GLASS	4.0
CONCRETE	IRON	50.0

Table 5.8: Misclassification rates (%) per pair — Height Variation

Confidence values again remained high across materials, reflecting the same issues discussed in the previous section.

Overall, the model proved unreliable in this scenario as well, primarily due to overfitting and limited generalization. Nevertheless, the increased variability of misclassifications and predictions under height changes reveals some potential for improved robustness, as the model may align better with the trained data under certain conditions.

5.3.4 Angular Variation Error Analysis

In the tests with angular variations, the model demonstrated a more nuanced error pattern compared to previous scenarios.

True label	Pred label	% of class Predictions
IRON	GLASS	2.0
IRON	CONCRETE	44.0
ALU	IRON	90.0
ALU	GLASS	2.0
ALU	WOOD	4.0
GLASS	IRON	20.0
GLASS	CONCRETE	10.0
WOOD	IRON	10.0
WOOD	GLASS	40.0
CONCRETE	IRON	22.0
CONCRETE	GLASS	46.0

Table 5.9: Misclassification rates (%) per pair — Angle Variation

The confusion matrix 4.18 and Table 5.9 display a broad variation in correct and incorrect predictions. A moderate decrease in misclassification rates can be observed overall, indicating again that the model generalizes better when unknown data is presented in a more varied manner. Iron remains a frequent source of misclassifications, though not to the extent seen in the two previous scenarios. This is accompanied by an increase in misclassifications involving glass and concrete. The confusion between wood and glass can be explained by their similar RCS profiles, which are not easily distinguishable, especially when irradiated from angles not encountered during training. Another expected source of confusion arises from similarities in RCS profiles between iron and aluminum as well as iron and concrete, which may account for the misclassifications of those materials. Nevertheless, aluminum is confused more than twice as often as wood is confused with glass—an unexpected outcome given the comparable differences in their RCS profiles. This discrepancy, which also cannot be explained by the model's training results, indicates that generalization issues and possible overfitting remain. Another indicator of this hypothesis is the consistently high confidence values across all materials, with some outliers at 100% confidence, suggesting that the model is overly confident in its predictions despite evident misclassifications.

Overall, the model makes fewer misclassifications and produces more expected errors, which can be better explained by RCS profile characteristics in this scenario. This demonstrates an improvement in generalization and robustness compared to previous scenarios with unseen data. However, the model's overconfidence in its predictions persists, indicating ongoing overfitting.

5.3.5 Summary

The error analysis across all scenarios reveals several key patterns and challenges faced by the model:

- The model struggles to differentiate between iron and concrete, even during training and with known data, likely due to their similar RCS profiles and differing sample proportions.

- Overfitting is a significant issue, as indicated by consistently high confidence values even in misclassified instances.
- Misclassifications are only partially linked to specific material properties, such as similarities in RCS profiles.
- Certain materials, such as iron, dominate predictions in the unseen data and height variation scenarios, indicating a potential bias in the model or a lack of generalization.
- The model's performance improves slightly under angular variations compared to other unseen data scenarios, suggesting some robustness to changes in acquisition geometry and a more balanced prediction pattern when more versatile data is present.

Chapter 6

Conclusion

This thesis investigated the feasibility and performance of a radar-based material classification system using the TI IWRL6432BOOST mmWave development board. The main objective was to evaluate whether range-bin intensity features, processed by a lightweight MLP, can reliably distinguish between five common surface materials (iron, aluminum, plexiglass, wood, and concrete) under real-time constraints on embedded platforms. The study encompassed the complete development pipeline—from sensor selection and commissioning, through controlled data collection and preprocessing, to model training, hyperparameter tuning, and evaluation across multiple testing scenarios. Performance was assessed using accuracy, macro-precision, macro-recall, macro-F1, inference-time measurements, and confusion-matrix analysis, with additional emphasis on generalization to unseen data and variations in acquisition geometry.

6.1 System Evaluation

The radar-based material classification system achieved its design objectives during training, delivering high accuracy and efficiency when operating with training data. In the Known Data scenario, macro-precision, macro-recall, and macro-F1 all exceeded 94%, with inference times consistently low (between 17-23 μs), confirming potential suitability for real-time embedded use. The integration of intensity features from selected range bins, processed by an extremely lightweight MLP architecture, enabled robust classification of five distinct surface materials with minimal computational overhead.

However, the system demonstrated severely limited generalization capability when exposed to distributional shifts. In the Similar Unseen Data scenario, performance degraded drastically (macro-F1 $\approx$ 7.7%), accompanied by a strong prediction bias toward iron and persistently high confidence in misclassifications, indicating overfitting to training-specific signal patterns. Height variation tests revealed selective robustness—iron and concrete maintained higher recall—yet most other materials experienced severe drops in both recall and precision. Angle variations were handled comparatively better, achieving a macro-F1 of $\approx$ 50%, suggesting partial resilience to geometric changes, likely due to more diffuse scattering patterns in certain materials.

Across all scenarios, inference times remained low, predictable, and stable, confirming computational suitability for deployment on resource-constrained robotic platforms. Nevertheless, the consistently high confidence in incorrect predictions, especially under domain shifts, represents a reliability risk and currently renders the system unsuited for practical use. This overconfidence, combined with the sharp performance drop outside the trained acquisition geometry, underscores the need for broader and more diverse training data, enhanced feature extraction methods, and the potential use of larger and more flexible models to improve robustness.

6.2 Real-World Applications

The radar-based material classification system developed in this thesis has potential for a broad range of application domains that require contactless and lighting-independent sensing [6]. In autonomous robotics, such systems can be applied to surface type recognition for adapting locomotion parameters or selecting appropriate manipulation strategies in unstructured environments, particularly when camera or LiDAR sensors are hindered by poor lighting, dust, fog, or privacy constraints [20]. A similar concept for detecting road surfaces can be integrated into an Advanced Driver-Assistance Systems (ADAS), for example as an advanced traction control feature, thereby enhancing vehicle perception and environmental awareness in automotive applications [24]. In industrial automation and quality control, this approach can provide rapid, non-destructive inspection of raw materials and finished products, enabling the identification of surface types or the detection of material inconsistencies on production lines [18]. In recycling and waste sorting, radar-based classification can facilitate the automated separation of metals, glass, plastics, and composites, even when objects are partially obscured or contaminated [2]. Furthermore, service and household robots could use such sensing to distinguish between fragile and durable objects, allowing safer handling and more context-aware interaction in domestic settings. It should be noted that the current system is not yet fully optimized for real-world deployment and requires further refinement and testing to ensure reliability and robustness across diverse environments.

6.3 Future Work

The developed radar-based material classification system demonstrated strong performance during training. However, its limited generalization capability in scenarios with distributional shifts underscores the need for improvement. The most critical issues include overfitting to training-specific signal patterns, high confidence in incorrect predictions under domain shifts, and a significant performance drop when acquisition geometry or environmental conditions differ from the training setup.

Several steps can address these weaknesses. Increasing the variability and quantity of the training data is the most important measure. This includes collecting data under different angles, distances, and environmental conditions. Incorporating additional features, such as Doppler micro-motion patterns or phase-based descriptors, could also provide more geometry-independent discriminative information. More advanced model architectures, such as CNNs for processing range-Doppler tensors or hybrid MLP-attention models, may improve generalization while keeping inference times low enough for embedded deployment. Data augmentation techniques, such as simulated range-Doppler distortions or noise injection, can further enhance robustness without excessive data collection.

Future work could integrate the system into multi-sensor fusion frameworks, combining radar with LiDAR or hyperspectral imaging to mitigate the limitations of a single sensing modality. Additionally, more surface types could be included in the training data and classification process to expand the functionality of the system. On the algorithmic side, confidence calibration and out-of-distribution detection could mitigate the risks of overconfident misclassifications in safety-critical applications. Domain adaptation techniques, such as transfer learning from larger pre-trained radar datasets, may also accelerate adaptation to new environments or material sets [11].

By addressing these points through broader data coverage, enhanced feature engineering, and more flexible model designs, the system can progress from a proof-of-concept for laboratory conditions to a robust solution for real-world applications.

In summary, this thesis presents empirical findings and interpretations of a lightweight radar-based material classification system's performance under varying conditions. While the developed approach meets its objectives in controlled scenarios, its limitations in generalization highlight clear directions for future improvement and provide a foundation for advancing radar-based material classification towards robust real-world applications.

Bibliography

[1] Daniel Adolfsson, Martin Magnusson, Anas Alhashimi, Achim J. Lilienthal, and Henrik Andreasson. Lidar-level localization with radar? the cfear approach to accurate, fast, and robust large-scale radar odometry in diverse environments. *IEEE Transactions on Robotics*, 39(2):1476–1495, 2023.

[2] Tommy Albing and Rikard Nelander. Material classification of recyclable containers using 60 ghz radar. *arXiv preprint*, 2023. Accessed: 2025-08-15.

[3] anonymous. Das telemobiloskop. `https://www.daidalos.blog/wissenschaft/naturwissenschaft/artikel/telemobiloskop/`, n.d. Accessed: 2025-06-24.

[4] anonymous. Fmcw radar. `https://www.geeksforgeeks.org/fmcwr-radar/`, n.d. Accessed: 2025-06-24.

[5] Blackvalue GmbH. Radar histrory. `https://www.blackvalue.de/en/radarbasics/radar-history.html`, n.d. Accessed: 2025-06-08.

[6] Snehal Buche. Understanding mmwave radar, its principle & applications. `https://www.design-reuse.com/article/61510-understanding-mmwave-radar-its-principle-applications`, 2024. Accessed: 2025-08-15.

[7] Aarushi Dhami, Naitik N Parekh, and Yash Vasavada. Digital beamforming for antenna arrays. In *2019 IEEE Indian Conference on Antennas and Propogation (InCAP)*, pages 1–5, 2019.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[9] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020.

[10] Massimo Guarnieri. The early history of radar [historical]. *IEEE Industrial Electronics Magazine*, 4(3):36–42, 2010.

[11] M. Henne, J. Ganslöser, A. Schwaiger, and G. Weiss. Machine learning methods for enhanced reliable perception of autonomous systems. Technical report, Fraunhofer IKS, 2021. Whitepaper.

[12] Texas Instruments. Iwrl6432boost/awrl6432boost evm: Fr4-based low power 60 ghz mmwave sensor evm user guide. https://www.ti.com/lit/ug/swru596/swru596.pdf?ts=1751894377837, 2022. Accessed: 2025-06-29.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[14] Cesar Iovescu and Sandeep Rao. The fundamentals of millimeter wave radar sensors. Technical report, Texas Instruments, 2021.

[15] Pradipta Bandyopadhyay Jitendra Gupta. Getting started with mmwave sensors, March 2025.

[16] Alexander Jung. *Machine Learning: The Basics.* Springer Singapore, Singapore, 2022.

[17] Tenner Lee. MmWave Sensors. https://www.mouser.de/applications/mmwave-sensors/, 2023. Accessed: 2025-06-08.

[18] Hironaru Murakami, Taiga Fukuda, Hiroshi Otera, Hiroyuki Kamo, and Akito Miyoshi. Development of a high-sensitivity millimeter-wave radar imaging system for non-destructive testing. *Sensors*, 24(15), 2024.

[19] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814. Omnipress, 2010.

[20] Sujeet Milind Patole, Murat Torlak, Dan Wang, and Murtaza Ali. Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, 34(2):22–35, 2017.

[21] Sandeep Rao. Introduction to mmwave sensing: Fmcw radars. Texas Instruments Application Note, March 2017.

[22] Deval Shah. Cross entropy loss: Intro, applications, code. https://www.v7labs.com/blog/cross-entropy-loss-guide, 2023. Accessed: 2025-08-17.

[23] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018.

[24] A. Soumya, C. Krishna Mohan, and L. R. Cenkeramaddi. Recent advances in mmwave-radar-based sensing, its applications, and machine learning techniques: A review. *Sensors*, 23(21):8901, November 2023.

[25] Simon White. Beamforming Basics. https://rfengineer.net/mimo/beamforming/, 2025.

[26] Christian Wolf. Doppler-effekt. https://www.radartutorial.eu/11.coherent/co06.de.html, n.d. Accessed: 2025-06-08.

[27] Christian Wolf. Entfernungsmessung mit radar. https://www.radartutorial.eu/01.basics/Entfernungsmessung%20mit%20Radar.de.html, n.d. Accessed: 2025-06-08.

[28] Muhammet Yanik, Akshay Kumar Chandrasekaran, and Sandeep Rao. Machine learning on the edge with the mmwave radar device iwrl6432. Technical report, Texas Instruments, May 2023.

Proclamation

Hereby I confirm that I wrote this thesis independently and that I have not made use of any other resources or means than those indicated.

Würzburg, August 2025