Road Condition Sensing Using Deep Learning and Wireless Signals

by

Soroush Ameli

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public

Statement of contributions

This project is joint work with another graduate student, Alex Chen. The algorithms and architectures presented in this thesis were developed during brainstorming sessions between me and Alex. I was responsible for capturing and pre-processing data, building and training models, and evaluating the performance of models. I enjoyed working with Alex and appreciate all the help.

Abstract

Similar to human car drivers, future driverless cars need to sense the condition of road surfaces so that they can adjust their speed and distance from other cars. This awareness necessitates the need for a sensing mechanism that enables cars to sense the surface type (gravel versus asphalt) and condition (dry versus wet) of a road. Unfortunately, existing road sensing approaches have major limitations. Vision-based approaches do not work in bad weather conditions and darkness. Mechanical-based approaches are either expensive or do not have enough resolution and robustness.

In this thesis, we introduce VIVA, which uses mmWave to enable robust and practical road sensing. Our key insight is that mmWave radar devices enable high resolution ranging, which can be used to scan the roughness of a road surface. Moreover, mmWave radar devices use high-frequency signals, which are significantly reflected by water, and hence can be used to sense the moisture level of a road. However, due to the high sensitivity of mmWave radar devices, other factors such as car vibration also impact their signals, resulting in noisy measurements. To extract information about road surfaces from noisy signals, we have developed a cross-modal supervised model that uses mmWave measurements to sense road surfaces. Our prototype of VIVA costs less than $300 and achieves more than 98% accuracy in classifying road types (gravel versus asphalt) and 99% accuracy in classifying road conditions (wet versus dry), even in bad weather and darkness.

Acknowledgements

I want to thank my advisor, Professor Omid Abari, for his invaluable guidance. Omid has been supportive and has allowed me to pursue various projects without objection. He has also provided insightful discussions about the research. Indeed, It is as a result of his excellent guidance and support that this research has been possible. I am also very grateful to Professor Ehsan Hashemi for his scientific advice and knowledge and many insightful discussions and suggestions during this work. I am also grateful to Professor Ali José Mashtizadeh and Dr. Ali Abedi for accepting to review this thesis as the members of my committee. My special thanks are extended to my friends and colleagues at Intelligent Connectivity (ICON) group for making my research an enjoyable experience. I wish to thank my parents who have always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve. Finally, a big thanks for the love and confidence of my siblings, in-laws, and extended family, who have always encouraged me.

Dedication

This thesis is dedicated to my wife, Zahra, who has been supporting and encouraging me during difficult moments. I am truly thankful for having you in my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous driving is an active area of research in both academia and industry, with much investment in large-scale projects by major car manufacturers and technology companies. Although driverless cars are already entering production, ensuring their safety and reliability remains a challenge. One part of this challenge is the cars' ability to sense road conditions. For example, imagine a rainy day. A human will notice that the roads are wet and drive carefully; a driverless car needs to do the same. Likewise, imagine a road whose surface changes from asphalt to gravel. The driverless car needs to scan the road, detect the change in the road's surface condition, and make important driving decisions and control strategies based on that.

Although there has been significant progress in building systems that enable cars to robustly detect surrounding objects [2, 12, 10, 25], there has not been much research in developing systems that can sense road conditions robustly under different weather conditions. Ideally, we would like to have a system that enables cars to detect a dry road from a wet one, and an asphalt road from a gravel one. This capability is important since, similar to human-driven cars, driverless cars need to adjust their speed and driving styles depending on changes in road conditions. Past work has tried to solve this problem using vision-based systems such as Camera, LiDAR, and Infrared. However, these systems suffer

Figure 1.1: VIVA enables cars to sense the road condition using mmWave signals. VIVA's sensor can be installed to the front (as shown here) or the rear side of the car.

from "weather blindness" and fail to function properly to provide robust perception in snow, rain, fog, darkness, or even direct sunlight since they all rely on light to sense the road. Another approach to estimate road conditions is to measure the wheel torque and tire force and use them to estimate the road friction coefficient. Unfortunately, the existing tire force and moment measurement systems are either very expensive (cost a few tens of thousands of dollars), or they do not have enough resolution and accuracy to estimate road conditions [23, 26, 5]. Ideally, we would like to have a sensor that is low-cost, has high-accuracy, and works robustly in different weather conditions.

mmWave technology has multiple properties that make it ideal for this application. First, due to a huge unlicensed spectrum available to mmWave technology, this technology can enable high-resolution ranging capability (i.e., sub-centimeter resolution). Therefore, it has the potential to measure the roughness of a road. Second, due to its high frequency, mmWave signals reflect differently from a dry surface than from a wet surface. Third, mmWave technology works in darkness. Moreover, because its wavelength is larger than raindrops and snow, it is not impacted by rain, snow, or fog. Finally, mmWave radar chips cost less than $50, which makes it very cost-effective for automobile applications.

Although mmWave technology has the ability to capture changes in road conditions and types, realizing the goal of road sensing using mmWave still requires addressing multiple

challenges. First, the captured reflected signal is a function of three parameters: wetness of a road, the distance of the sensor from a road, and the height of a car. Second, mmWave radar has a limited range resolution, and, consequently, a small vibration can move the reflected signal outside of a radar's bin. With that being said, a car's vibration adds a significant source of noise to the reflected signal. These vibrations are small. However, they result in a significant change in the measured reflected power. Hence, the effect of vibration on the signal can be as large as the effect of changes in road roughness. To overcome these challenges, we propose VIVA, which uses mmWave signals to sense road conditions. Our key insight is to use a learning model that separates informative signal features from noise sources (vibration, height, etc.) in order to robustly classify the road condition and its type.

We have built a prototype of VIVA using off-the-shelf components and devices. Our sensor costs less than $300 and can be easily installed under a car as shown in Figure 1.1. This research makes the following contributions:

- We introduce VIVA, an AI-powered mmWave system that enables cars to detect road conditions and road types.
- We develop a cross-modal supervised learning model that uses noisy signals reflected from road surfaces to extract informative features and classify road conditions.
- We have implemented and evaluated our system in real-world experiments. Our results (from more than 80,000 measurements) show that VIVA achieves more than 98% accuracy in detecting road conditions and types.

The rest of this thesis is structured as follows. Chapter 2 presents an overview of the related work in the area of mmWave sensing and road surface condition sensing. Chapter 3 presents background on FMCW, mmWave signals, and multi-layer feed-forward neural networks. In Chapter 4, the design of the mmWave model and the training network are elaborated. Implementation details, such as details about hardware and software, are

presented in Chapter 5. In Chapter 6, we first introduce our evaluation steps and the performance metrics for road surface sensing. Then, we show the results of classification for dry asphalt, wet asphalt, dry gravel, and wet gravel. Finally, in Chapter 7, we conclude the thesis by summarizing the key advantages of VIVA and suggesting some of the potential development opportunities.

Chapter 2

Related Work

Related work can be categorized into two main topics: mmWave sensing and road surface detection.

2.1 mmWave Sensing

Using mmWave to sense the environment has received significant attention in recent years. For example, [33] introduces a system that uses mmWave signals and a deep neural network to sense the environment and reconstruct a high-quality audio signal. The work presented in [17] introduces a gas sensor that uses a 60 GHz mmWave radar. This sensor enables real-time gas monitoring in different environments. Additionally, mmWave technology has been used in gesture recognition and tracking systems. For example, Soli is a low power mmWave sensor that senses hand gestures [14]. The system presented in [32] detects the motion of objects and tracks them using a 60 GHz mmWave signal. Furthermore, mmWave technology has been used to detect and identify multiple people in an environment. For example, the authors of [8] used the characteristics of mmWave signals to fingerprint an environment. In their work, measurements from an environment with and without people are collected. Then, these measurements are fed to an LSTM based classification model

that identifies the people. The system presented in [7] uses multiple mmWave transmitters and receivers to provide a real-time high resolution image of the body for security screening. In addition, mmWave sensors are used in health and agriculture. For example, SleepSense [15] is an in-home sleep monitoring system that collects respiration data during sleep. The system introduced in [35] uses mmWave signals to monitor vital signs, including breathing rate, heart rate, and sleep. In [28], blood glucose levels are measured using the penetration property of 60 GHz signals. In contrast to these systems, VIVA uses mmWave to enable robust road surface monitoring for driverless cars.

2.2 Road Surface Condition Detection

Existing approaches for detecting and classifying road surface conditions fall into three major categories: vision-based (Lidar, Camera, etc), Ultrasonic-based Sensors, and vehicle dynamic-based (mechanical) methods. Unfortunately, these approaches are either very expensive or unreliable. For example, the authors of [4] proposed a system that detects roads based on backscattered ultrasonic signals. They showed that they could classify five kinds of roads. However, this system achieves only 80% accuracy. In addition, their system is prone to errors due to environmental conditions, vehicle movement, and air current fluctuations. The authors of [27] proposed a vision-based system that attains only 68% in detecting road conditions. Furthermore, it does not work robustly in darkness and bad weather conditions since it uses a camera. The system introduced in [1] uses Lidar to classify roads. However, the limitation of [1] is that Lidar is an expensive sensor. Moreover, its performance would be affected on rainy days or in fog. [24] uses IR to detect different types of roads. Unfortunately, this system cannot detect whether a road is wet or dry. Also, it has not been tested on practical road surfaces. In contrast to these works, VIVA works in any weather and darkness and achieves an accuracy of more than 98% in detecting road conditions and road types.

Finally, the mechanical-based road surface classification methods can be divided into

two groups: direct tire force/moment measurement systems, and parameter identification methods (PIMs). The first method is very expensive (more than fifty thousand US dollars). Also, it requires complex devices that could not be used in most cars [6, 19, 5]. In the second approach, PIMs use inertial measurement unit (IMU), wheel speed, and vehicle dynamics. This makes PIMs more cost effective. However, they do not have enough resolution and accuracy to enable reliable road surface sensing [20, 31, 18, 3, 29]. For example, in [9], the authors developed a system that uses IMU data to measure car vibrations. Then, these vibrations are used to detect patches or potholes in the ground. Nevertheless, this system cannot detect road conditions. In contrast to these works, VIVA costs less than $300. Furthermore, VIVA achieves more than 98% accuracy in detecting road conditions and road types. Moreover, VIVA is able to maintain its performance in low-excitation and low-speed scenarios, where most mechanical approaches fail.

Chapter 3

Background

VIVA builds on three main technologies: FMCW, mmWave and Artificial Neural Networks. In this section, we provide background on these technologies.

3.1 FMCW Radio

Frequency Modulated Continuous Wave (FMCW) is a technique used in radar devices to detect the distance or speed of an object. In this technique, the radar transmits a sine wave (also called a chirp) whose frequency increases linearly with time, as shown in Figure 3.1. The transmitted signal reflects from an object, and receiver antennas receive the reflected signal after some delay. Then, a mixer mixes both the transmitted and received signals and filters the high-frequency components to produce an intermediate frequency (IF) signal. This process can be shown using Equation 3.1,

$$\begin{aligned} f_t &= \sin(\omega_t t + \phi_t) \\ f_r &= \sin(\omega_r t + \phi_r) \\ f_{IF} &= \sin((\omega_t - \omega_r)t + (\phi_t - \phi_r)), \end{aligned} \tag{3.1}$$

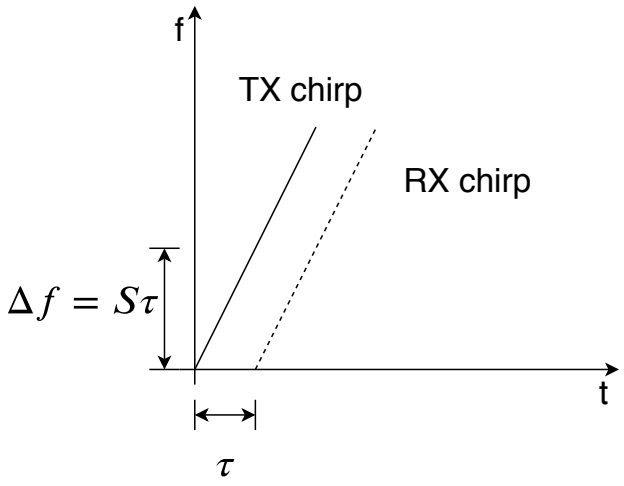


Figure 3.1: FMCW radar transmits a sine wave (chirp) whose frequency increases linearly with time. The received signal is a delayed version of the transmitted one.

where f_t and f_r are transmitted and received signals respectively, and f_{IF} is the signal at the output of the mixer.

The IF's frequency is the frequency difference between the transmitted and received signals (Δf) at any point in time. This frequency difference is measured by calculating the Fast Fourier Transform (FFT) of the IF signal. Since the received signal is just a delayed version of the transmitted signal, the time of flight of the transmitted signal (τ) is calculated as follows:

$$\tau = \frac{\Delta f}{S}, \tag{3.2}$$

where S is the slope of the chirp, and it is known. The distance between the radar and the object, i.e. d, is calculated using the measured time of flight as follows:

$$d = \frac{1}{2} \times C \times \tau, \tag{3.3}$$

where C is the speed of light. Note that the range resolution of the FMCW is a function of the total bandwidth that the transmitted signal sweeps. The resolution, R, can be

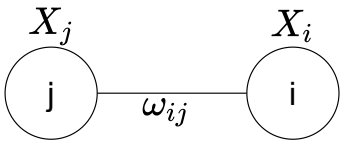


Figure 3.2: The connection between two neurons i and j

formulated as follows:

$$R = \frac{C}{2B}, \tag{3.4}$$

where C is the speed of light and B is the total bandwidth of the frequency sweep. Thus, larger bandwidths enable higher resolutions in ranging.

3.2 mmWave Signal

mmWave signals are RF signals that operate at frequencies ranging from 30GHz to 300GHz. At this frequency range, a huge unlicensed spectrum is available. For example, there is a 4GHz bandwidth available at 77GHz. As mentioned earlier, this huge bandwidth enables high-resolution (sub-centimeter) ranging using FMCW. Furthermore, mmWave signals have a small wavelength. Therefore, antennas at these frequencies are tiny (a few millimeters). Hence, an array of antennas can be packed into a small area to form an antenna array. This type of array enables FMCW radios to focus their signal to a very narrow beam which can be steered electronically to scan in different directions. The beam allows them to achieve long-range sensing. Moreover, mmWave signals can penetrate through certain materials like plastic and operate in a wide range of weather conditions such as rain, fog, dust, or snow. Finally, due to the high frequency of mmWave signals, their reflection coefficient is very sensitive to minor changes in the surface of a reflector and its moisture level. VIVA exploits these properties of mmWave signals to detect road conditions and types.

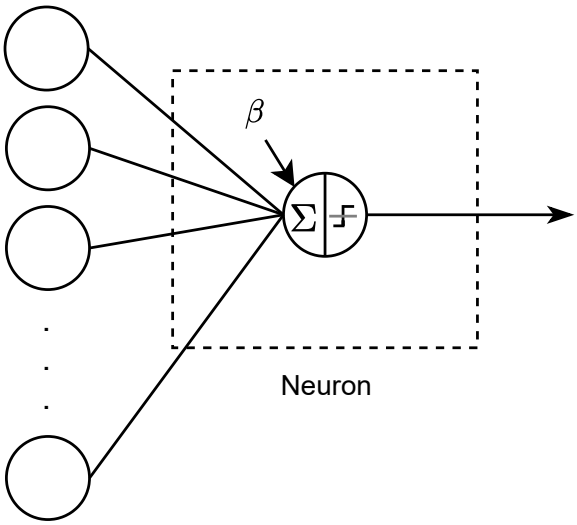


Figure 3.3: A single neuron's structure

3.3 Multi-Layer Feed-Forward Neural Network

A neural network is a combination of small processing units, called neurons. Each neuron receives data, processes it, and communicates the result with other neurons. The importance of a connection between two neurons is measured by a real number, called a weight. The Multi-Layer Feed-Forward (MLF) neural network is a classic structure widely used in many applications. An MLF neural network consists of three types of layers: one input layer, one or several hidden layers, and one output layer.

Each neuron in a layer is connected to all the neurons in the next layer. Figure 3.2 shows how the neuron j in a layer is connected to the neuron i in the next layer. A neuron has three inputs: the outputs of the neurons in the previous layer, the weight coefficients, and a threshold (or bias). A neuron combines these inputs to output a value. The inner structure of a neuron is shown in Figure 3.3. The output of a neuron can be formulated as follows:

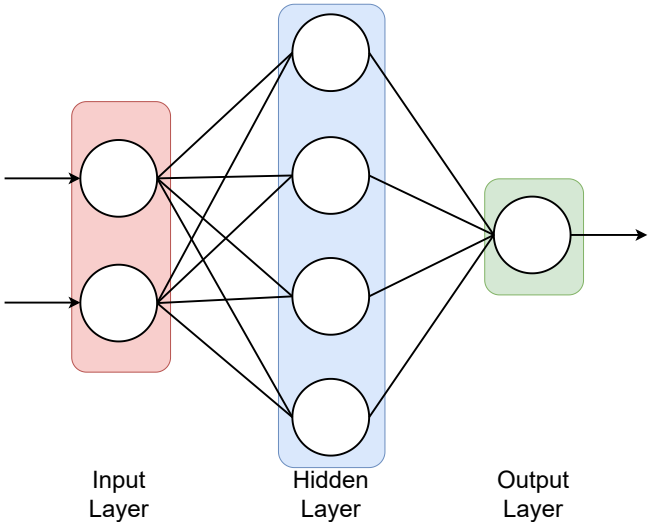


Figure 3.4: A feed-forward neural network with one input layer, one hidden layer, and one output layer

$$X_i = f(\sum_{j=1}^{J} \omega_{ji} X_j + \beta), \tag{3.5}$$

Here, f and β represent the activation function and bias, respectively. A bias is a neuron with a constant value that shifts the activation function to the left or right. In Figure 3.3, a sign function is considered for the activation function. However, this could be replaced with any function, such as a Sigmoid, rectified linear unit (ReLu [22]), or other functions.

The numbers of neurons in input and output layers are decided based on the dimension of inputs and the number of labels. However, the number of hidden layers and the number of neurons used in each hidden layer depend on the application. Figure 3.4 represents a simple MLF neural network with one hidden layer. This architecture is also known as a fully connected neural network since each neuron is connected to all the neurons in the next layer.

The outputs of the last layer are scores for each output class. These non-normalized

Algorithm 1 Training Procedure for MLF Neural Network

Input: Data, Labels, Epoch, MLF Neural Network

```
X ← Data
Y ← Labels
ω ← random Initialization
epoch ← Epoch
i ← 1
while i ≤ epoch do
    for all x ∈ D, y ∈ Y do
        Forward pass data to compute scores for each class
        Use Softmax to compute P(ŷ|ω, w)
        Backpropagate the error through the network
        Update parameters
    end for
    i ← i + 1
end while
```

scores must be converted to a probability distribution over predicted output classes. Softmax is a function that takes a vector of K real numbers as input. Then, it normalizes the vector into a probability distribution consisting of K probabilities. That is, prior to applying Softmax, some vector components could be negative or greater than one, and might not sum to 1. However, after applying Softmax, each component will be between zero and one, and the components will add up to 1. Hence, they can be interpreted as probabilities. In addition, the larger input components will correspond to larger probabilities. The Softmax functionality is as follows:

$$\sigma(z)_i = \frac{e^{\frac{z_i}{T}}}{\sum_{j=1}^{K} e^{\frac{z_j}{T}}} \tag{3.6}$$

Training a neural network follows a simple procedure. First, all the weights are initial-

Algorithm 2 Deployment Procedure for MLF Neural Network

Input: Data, Label, MLF Neural Network

Output: Predicted Label

$x \leftarrow Data$

$y \leftarrow Label$

Forward pass data to compute scores for each class

Use Softmax to compute $P(\hat{y}|x, w)$

$\hat{y} = arg \max_{\hat{y}} P$

ized to a random number between zero and one. Afterward, each example in a dataset is passed to the input layer of the network. Then the scores for all output classes will be calculated and given to the Softmax function to convert them to probabilities. The class with the highest probability is considered as the predicted class. By comparing the predicted and actual classes, an error is calculated. This error is back-propagated through the entire network to update the parameters of the MLF neural network. This procedure should be repeated more than once to ensure parameters are converged to an optimal value. The training procedure for an MLF is shown in Algorithm 1.

Deploying a neural network to make a prediction is quite straightforward. First, an example is fed to the first level, thereby generating the scores in the output layer. Using the Softmax function, the probability of each class is calculated, and the predicted class is the one with the largest probability. The deployment procedure for an MLF neural network is shown in Algorithm 2.

Chapter 4

VIVA

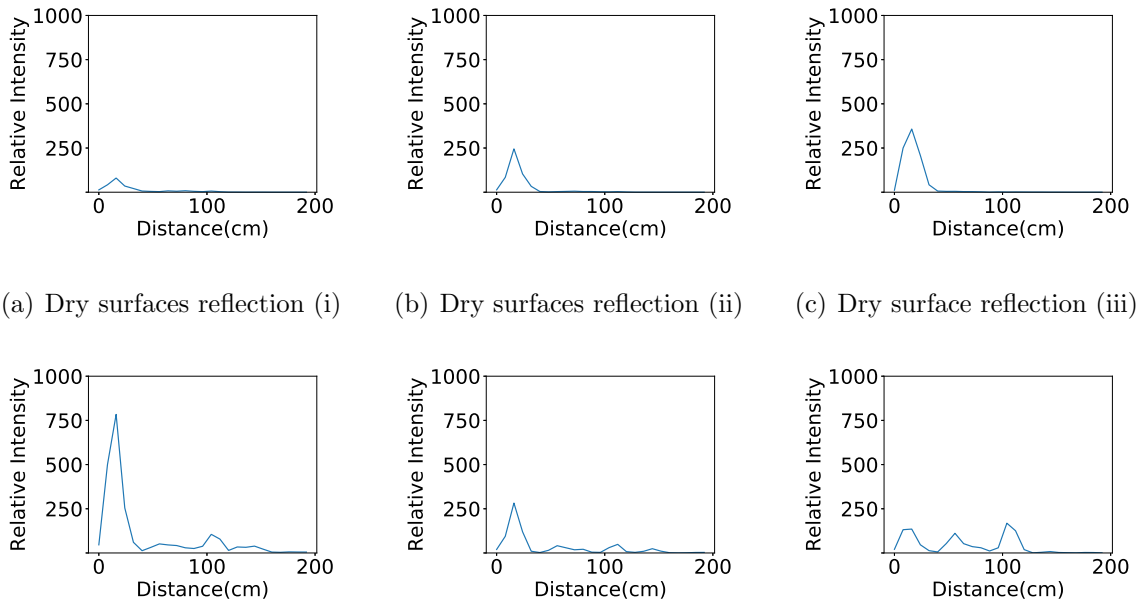
VIVA is an AI-powered mmWave system that enables driverless cars to sense road conditions (dry, wet, asphalt, gravel). VIVA has three main components: a mmWave radio, which transmits and receives FMCW signals, a low-cost camera, and a processor, which runs VIVA's algorithm and software. VIVA can be easily attached to a car, as shown in Figure 1.1. It relies on the key insight that the type and the condition of a road impact the reflection coefficient of roads' surfaces. Therefore, by using mmWave radar, we can measure the signal reflected from the road and use it to detect the road type and condition.

However, noise from various sources, such as car vibration, impact the measurement of reflected signals. Therefore, detecting the road condition using reflected signals is very challenging. To solve this problem, VIVA uses mmWave signal measurements and feeds them into a multi-layer feed-forward neural network. Our model takes advantage of cross-modal supervision to improve the accuracy of the system. The next few sections present the components that contribute to the design of VIVA. In this chapter, we first investigate the feasibility of using mmWave for road condition and type detection. We, then discuss the challenges of robust road sensing using mmWave. Finally, we explain how VIVA overcomes those challenges to ensure robust road sensing.

4.1 Challenges in using mmWave for road sensing

mmWave signals reflect differently over different road types and conditions. The differences in the reflected signals are due to multiple reasons: (i) the reflection coefficient of mmWave signals is very high for water; hence, the amplitude of the reflected signal from a wet road is much higher than the one from a dry road; (ii) the surface of a wet road is flatter and has less scatter effect compared to a dry road; thus, the signals reflected from a wet surface experiences less multipath effect compared to ones from a dry road; (iii) for a similar reason, the signal reflected from an asphalt road experiences less multipath effect than ones from a gravel road do.

Next, we run experiments to observe the impact of road type and road conditions on mmWave signals. In these experiments, we install an FMCW mmWave radar device under a car. The radar transmits a signal and measures the amplitude of the signal reflected from a road to the sensor. We first drive on a dry road, then repeat this experiment on a wet one. For both experiments, we make sure the speed of the car and other parameters are the same. Figure 4.1 shows the results of these experiments, with three consecutive output frames from the mmWave radar. The x-axis is the distance of the reflector from the sensor, and the y-axis is the amount of reflection at that specific distance. The peak shows the power reflected from the surface of the road since the road surface is the main source of reflection. Contrary to our expectations, the signal reflected from the wet road does not always have a higher amplitude than the one from the dry road. Although the average amount of signal intensity reflected by the wet surfaces is higher, we cannot simply rely on the main peak in the spectrum to detect the road condition. Therefore, we need to take more factors into account when distinguishing a wet surface from a dry one.



(a) Dry surfaces reflection (i) (b) Dry surfaces reflection (ii) (c) Dry surface reflection (iii)

(d) Wet surfaces reflection (i) (e) Wet surfaces reflection (ii) (f) Wet surfaces reflection (iii)

Figure 4.1: The relative power intensity of mmWave signal reflected by dry and wet surfaces for three different frames. Although a wet surface is a better reflector (and scatters less) than a dry surface, the signal peak in the reflection of a wet surface is not always greater than the one from a dry surface.

4.2 Feature Extraction

Although a wet road reflects more signals than a dry road, the amplitude of the reflected signal at the output of the mmWave radar is sometimes lower for a wet road compared to a dry road. Our hypothesis is that car vibrations and the limited range resolution of the mmWave radar cause this outcome. Specifically, when the car vibrates, the distance of the sensor from the road slightly changes, as a consequence slightly shifting the peak's location in the output of the radar. Since the radar has a limited resolution, the peak will fall between two radar bins, resulting in a *spectrum leakage* problem and a smaller peak amplitude. In this situation, the power of the reflected signal is not changed in reality. However, it might vary in the output of the radar since it moves outside of a radar bin. Figure 4.2 demonstrates this problem. In Figure 4.2(top), the signal falls on a radar bin, whereas in Figure 4.2(bottom), the signal experiences a slight shift, and hence, falls between two bins. In this case, the amplitude of the peak has decreased, and the signal has generated leakage in the other bins. Therefore, to detect the road condition from the mmWave signal, we need to consider both the amplitude of the main peak and other bins caused by the radar's range leakage as the input features. Furthermore, when the roughness of road changes, the signal experiences different multipath. Therefore, the other bins of the radar will have a higher amplitude. Thus, we need to develop a method that considers both the amplitude of the peak and the amplitude of the other bins in the output of the radar in order to classify the road type (gravel and asphalt) and condition (wet and dry).

4.3 Two-label Classification

We first develop a model to distinguish a dry road from a wet road. Due to the nature of our dataset and the fact that the number of our features is not very large, we design a multi-layer feed-forward (MLF) neural network, shown in Figure 4.3, for distinguishing a dry road from a wet road. This model consists of eight layers: the input layer, three

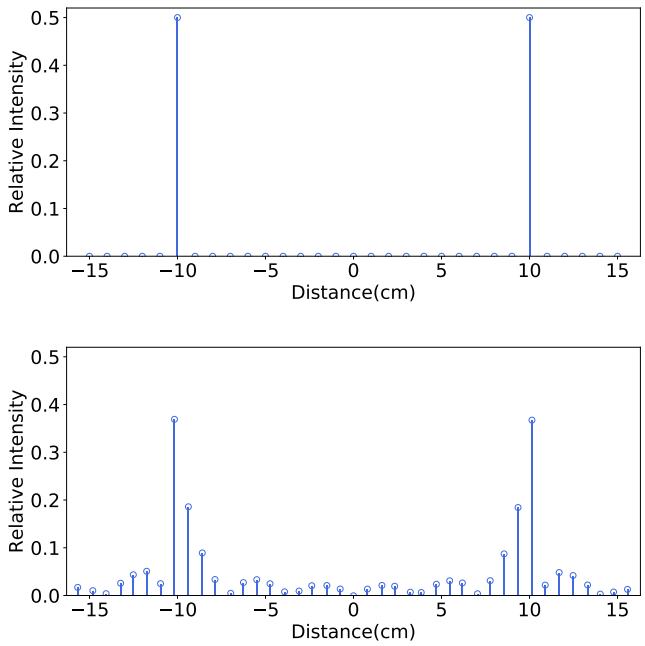


Figure 4.2: The output of a radar for an object located 10 cm and 10.1 cm away from the sensor with a range resolution of 1 cm. The figure shows that when the signal peak is outside of a radar bin, it causes spectrum leakage and, therefore, the amplitude of the main peak decreases.

dense layers each followed by a dropout layer, and the output layer. The dropout layers will prevent our model from overfitting during the training process. However, they are not used during testing and deployment. Our data is the whole radar's range, and it consists of 256 bins. Therefore, the input layer has 256 units. We designed the model such that the size of the first hidden layer is larger than the input layer's size. By doing this, we are mapping input features into a higher dimension where all the subtle changes and the correlations between the features can be captured. This level of detail allows the model to extract useful information, like the leakages on other bins from the main peak, from all of the features. Then, we gradually reduce the size of hidden layers, so that we focus on important information, such as the largest peak of the signal. We choose ReLu as our

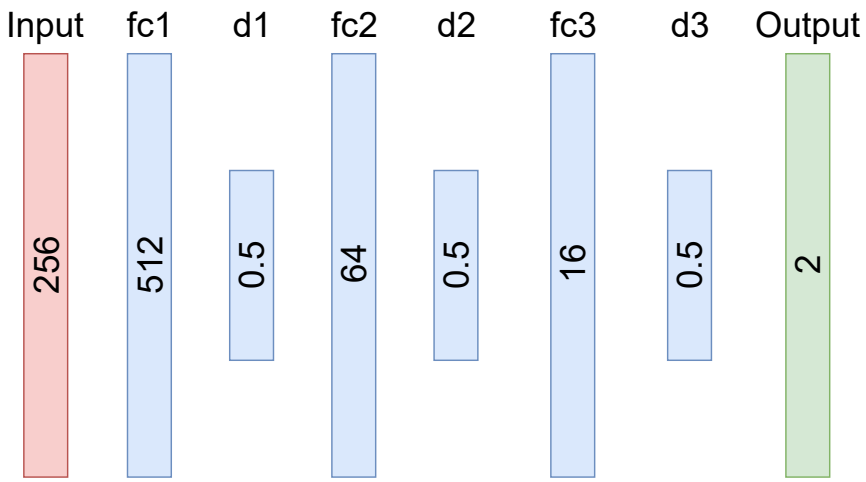


Figure 4.3: The architecture used in VIVA for detecting dry versus wet roads

activation function in all the dense layers. The output of this function is equal to its input, x, if x is positive and 0 otherwise. This function enables the neural network to avoid the vanishing gradient problem. Since we are doing a binary classification task, the last layer must have two units. Each unit represents a number that shows a score for each predicted class. We use Softmax as the activation function for the output layer. This function normalizes the output vector of the output layer to a probability distribution. Since the ranges of features in the dataset differ, and the features have a large magnitude, the data cannot be directly fed to the model. Therefore, we have to pre-process our mmWave dataset. For the dry versus wet classification, we treated each example in the dataset as a vector and divided it by its norm to make it a unit vector. This normalization scales down the magnitude (which was over 10^{12}) of all the features while preserving the shape of the spectrum, resulting in faster computation.

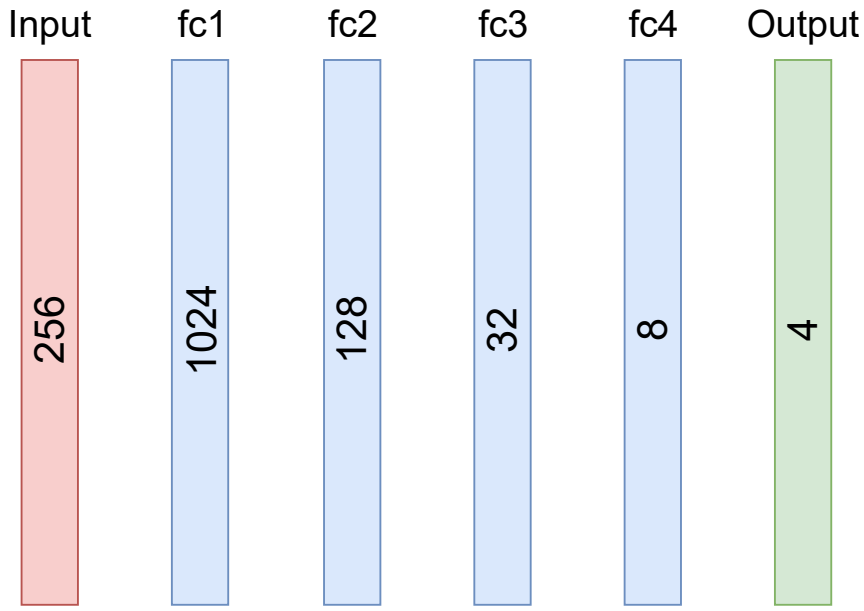


Figure 4.4: The architecture used in VIVA for detecting dry versus wet, and gravel versus asphalt roads

4.4 Four-label Classification

Next, we design a model that not only distinguishes dry roads from wet ones but also asphalt from gravel. To do so, we design a more sophisticated multi-layer feed-forward neural network. As shown in Figure 4.4, it has six layers: the input layer, four dense layers and the output layer. We do not have any dropout layers for this model since the model does not overfit. We choose the same activation functions as the two-label model for the dense layers and the output layer. Note that the size of the first hidden layer in the four-label classification is larger than the first hidden layer in two-label classification. This difference arises since the difference between a gravel road and an asphalt road is much more tenuous than that between a dry road and a wet road. Thus, mapping features to a higher dimension gives the model more detailed information about the relationship between features.

For data pre-processing, the normalization, as in the two-label classification, does not work since the amplitude of the peaks in the reflections of gravel and asphalt might not be significantly different. Thus, giving higher weights to stronger features is not sufficient, and we need to focus on the changes in each feature. We use MinMax scaler on the features, transforming them such that the maximum and minimum values of the feature are mapped to one and zero respectively. This transformation allows our neural network to observe changes in all the features with equal emphasis. In other words, the model takes small changes in the signal into account when making predictions.

4.5 Cross-modal Supervision

So far, we have designed a model that can classify the road condition using mmWave radar measurements. Although our empirical results show that the model performs very well once it is trained, it has one limitation; changing the orientation or position of the sensor can affect the measurements, and hence the mmWave model requires to be re-trained. Unfortunately, re-training of a neural network is costly due to two main reasons: (i) training procedure requires expensive computational resources; (ii) providing labelled data requires a tremendous effort and manpower.

To solve this problem, we design a teacher-student network that uses both image data and mmWave measurements. Note, although the camera does not work well during the night and foggy weather, it does not require re-training if its position changes. Hence the image model can be trained once and used as a teacher to train the mmWave model if the position of sensor changes. Our teacher-student network is illustrated in Figure 4.5. The top pipeline in the figure illustrates the teacher model, which provides cross-modal supervision; the bottom pipeline shows the student model, which performs mmWave-based road surface sensing. In the following, we explain the model in detail.

Neural networks usually use a Softmax, illustrated in Equation 3.6, in the output layer to convert the computed logit, i.e. z_i, for each class into a probability, i.e. q_i. In Equation

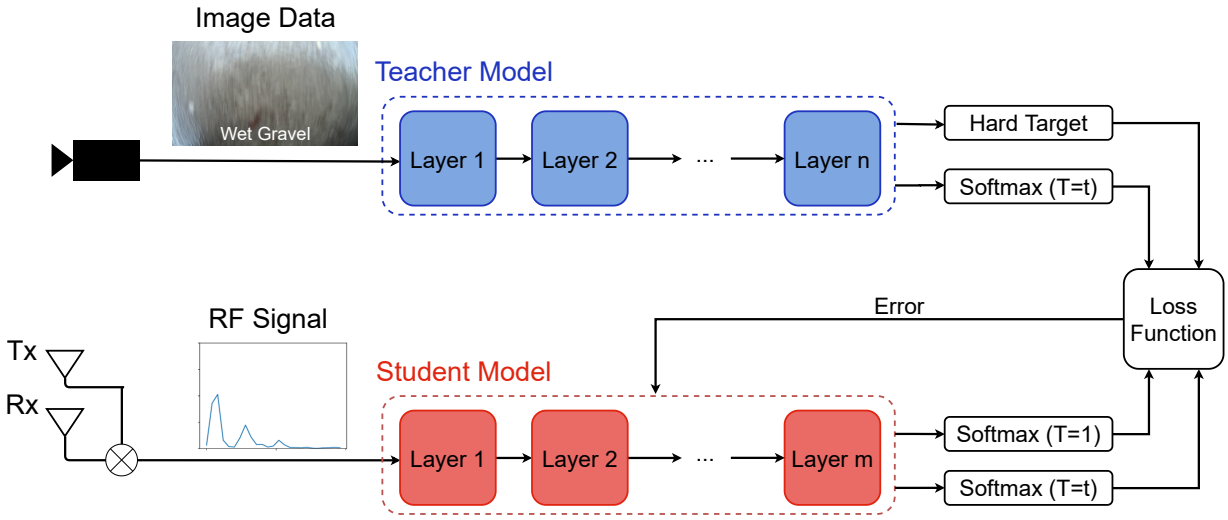


Figure 4.5: Training overview of the VIVA system. VIVA combines the signal measurements captured by a mmWave FMCW radar with the images captured by a camera to train a robust mmWave-based road type and condition sensing system.

3.6, T represents a temperature, and it is usually set to one. For high temperatures ($T \to \infty$), all classes have nearly the same probabilities. On the other hand, at the lower the temperatures ($T \to 0$), the probability of the most probable class tends to 1.

Now, consider a pair of image and RF signals (I, R), where R and I denote the mmWave measurement and the synchronized frame, respectively. The teacher model takes frame I as input and generates two sets of targets: a hard and a soft target. A hard target is the predicted class for an example, whereas a soft target represents the likelihood of each class for a given example at temperature T. Soft targets provide much more information per training case than hard targets, while both provide cross-modal supervision for training the mmWave model. Hence, the mmWave model learns to predict road surface condition and type using mmWave signals. We use the Resnet50 network, proposed in [11], as the teacher model. The training objective function is to minimize the difference between the predictions of mmWave and the image network. Equation 4.1 shows the loss function where x is the input, W is the student model's parameters, y is the hard labels generated

by the teacher model, H is the cross-entropy loss function, σ is the Softmax function parameterized by the temperature T, and α and β are coefficients. Additionally, z_s and z_t are the logits of the student and teacher respectively. The first term of Equation 4.1 is computed using the hard targets of the image model and soft targets of the mmWave model at $T = 1$, while the second term is the cross-entropy of soft targets at $T = \tau$.

$$\mathcal{L}(x; W) = \alpha \times \mathcal{H}(y, \sigma(z_s; T = 1)) + \beta \times \mathcal{H}(\sigma(z_t; T = \tau), \sigma(z_s; T = \tau)) \tag{4.1}$$

After training the model using Algorithm 3, the mmWave model is deployed using Algorithm 4 for road surface sensing.

Algorithm 3 Training Procedure for Teacher-Student Network

Inputs: Image Data, mmWave Data, Epoch, Image model

Output: mmWave model(ω)

$X_t \leftarrow Image\ Data$

$X_s \leftarrow mmWave\ Data$

$\alpha \leftarrow 1$

$\beta \leftarrow 0.07$

$T_1 \leftarrow 1$

$T_2 \leftarrow 5$

$N \leftarrow Number\ of\ Examples$

$\omega \leftarrow random\ Initialization$

for all $i \in [1, Epoch]$ **do**

  for all $j \in [1, N]$ **do**

    Forward pass $X_t(j)$ to image model to compute hard target(y) and logits(z_t)

    Use Softmax to compute soft targets at T_1 and T_2

    Forward pass $X_s(j)$ to mmWave model to compute logits(z_s)

    Use Softmax to compute soft targets at T_1 and T_2

    Calculate the loss based on equation 4.1

    Backpropagate the error and update ω

  end for

end for

Algorithm 4 Deployment Procedure for mmWave model

Inputs: mmWave Data, mmWave model

Output: $\hat{y}$

$D \leftarrow mmWave\ Data$

$M \leftarrow mmWave\ \ model$

$N \leftarrow\ \ Number\ of\ Examples$

if $n = 2$ **then**

  for $x \in D$ **do**

    $Normalize(x)$

  end for

else

  $MinMaxScale(D)$

end if

for all $j \in [1, N]$ **do**

  $P_j \leftarrow \sigma(M(D),\ T = 1)$

  $\hat{y_j} = arg\max P_j$

end for

Chapter 5

VIVA Implementation

5.1 Hardware

VIVA consists of a mmWave radar device and a low-cost camera, as shown in Figure 5.1. For the radar, we use the AWR1642 device from Texas Instruments (TI). The sensor has a UART port that allows one to store data on a laptop. For the camera, we use the ELP 2.8-12mm Varifocal Lens 2.0 megapixel, whose resolution is 1920 x 1080 pixels. The camera is connected to the same laptop through a USB cable and is installed right beside the sensor so that the image data and the sensor data can represent the same time frame. We use a ThinkPad T580 with an Intel CORE i7(8th Gen) to collect the data. Both the camera and the sensor operate at the frame rate of 30 fps, and we assemble them in a box as shown in Figure 5.1. For image classification, we use a GPU (TITAN RTX with 24GB memory) to speed up the training process. The GPU is installed on a desktop with an AMD® Fx(TM)-6300 six-core processor and 500GB SSD. All the images are stored on the SSD so that they can be accessed faster and then send to the GPU.

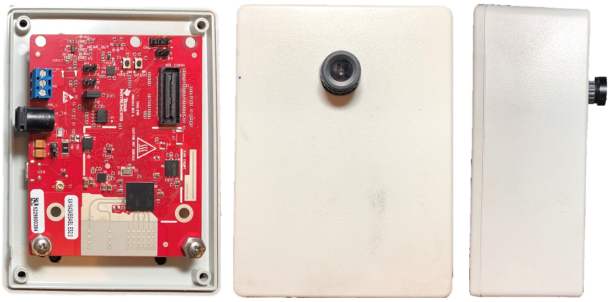


Figure 5.1: The prototype of VIVA's sensor (inside and outside). The sensor includes a low-cost camera and a mmWave radar module placed in a waterproof box. The sensor is small ($11 \times 8 \times 4$ cm) and can be easily installed on a car.

5.2 Software

We use Windows 10 for data collection and Ubuntu 16.04 for the data programming tasks. We collect the data using the version 2.0.0 of mmwave demo visualizer, developed by mmwavebeta group [21]. We set the frame rate to 30 and use 4 RX and 2 TX as the antenna configuration of the sensor. We run the software on Windows 10 and collect the binary data exported by the visualizer. This binary file contains the FFT results for each frame. We reconstruct the data structure from documentation provided by TI in the mmwave-SDK and decode the binary data back into the C++ structure so we can parse them. Finally, we store all the data as a CSV file. To prepare the dataset, we use Python 3.6, Pandas, Scikit-Learn, and Imblearn to transform data into an appropriate form. Finally, we use Google Colaboratory to train the mmWave models and a local GPU for training the image classifier. The framework we use to train our models is Tensorflow 2.0, and we specifically use Keras to build and train the models. For image data, we store the video data and then extract frames from the videos using OpenCV 4.2. We then use ResNet50 as the image classification model.

Chapter 6

Evaluation

In this chapter, we evaluate the performance of VIVA in sensing road condition and type. To do so, we place our device in a waterproof box and attach it to a car, as shown in Figure 6.1. We installed VIVA in the rear side of the car since it is easier for cabling. However, a car manufacturer can easily install it in front of the car. Our device is powered through a power cable with 5V working voltage and has a USB port connected to the laptop.

6.1 Data Collection

We collected around 50000 samples from several asphalt and gravel roads, for speeds ranging from 10km/h to 60km/h, and across different weather conditions including rainy, cloudy and sunny days. For each example, we captured the whole frequency spectrum of the received signal, which consists of 256 features. Since the goal is to first distinguish between dry and wet roads and then detect asphalt versus gravel, we labeled our dataset in two ways. First, we used zero and one as the labels, respectively representing a dry and wet road. In the other scenario, we used four consecutive numbers, from 0 to 3, representing respectively dry asphalt, wet asphalt, dry gravel, and wet gravel. The number of examples

(a) (b)

Figure 6.1: VIVA's sensor attached to a car.

in each class for 2 labels and 4 labels are shown in Table 6.1 and 6.2 respectively. Furthermore, we used the camera to collect the same amount of data with the same distribution for the image classification task.

6.2 Data Preparation

As illustrated in Tables 6.1 and 6.2, the populations of classes are highly imbalanced. Feeding this data into the model directly causes problems. For example, if we train the model using the imbalanced data, the model's accuracy might be high, even though most of the examples in class wet are mispredicted. There are multiple ways to solve this issue, including collecting more examples or using re-sampling techniques. Since collecting, processing, and labeling more data is an expensive process, we use over-sampling techniques. Specifically, we randomly over-sample examples from classes with less population to make the size of classes equal.

Next, we need to feed the data to the models. For measuring how well the models perform, we need a set of data that is not seen by the models. To do so, we split our dataset into two subsets: a training set and a test set. The training set consists of eighty

Classes	Percentage
Dry	72.6%
Wet	27.4%

Table 6.1: Distribution of instances in dry road and wet road classes

Classes	Percentage
Dry Asphalt	56.8%
Wet Asphalt	21.5%
Dry Gravel	15.8%
Wet Gravel	5.9%

Table 6.2: Distribution of instances in dry asphalt road, wet asphalt road, dry gravel road, and wet gravel road classes

percent of the original dataset, each example having been randomly drawn from the original dataset. Then, we use the training set to train the models. We use the remaining twenty percent of the original dataset as the test set.

6.3 Training Procedure

We adopted ResNet50 as the base model for building our image classifier (teacher model). We use the pre-trained weights on the ImageNet dataset as the initialization for the model. Also, at the beginning of each epoch, We shuffle the training examples, set the batch size to 64, and use Adagrad as the optimization function. Then, we train the model for 100 epochs.

To train the mmWave model for detecting dry road versus wet road, similar to the teacher model, we set the batch size to 64 and shuffle the data at the beginning of each epoch. However, we train the model for 100 epochs and use Adam as the optimizer. Since

	Precision	Recall	F1-score
Dry	100%	99.99%	100%
Wet	99.94%	100%	99.97%

Table 6.3: Results of classification in detecting wet versus dry road.

we are predicting two situations using this model, we use binary cross-entropy as the loss function. Training the mmWave model for detecting dry road versus wet road and gravel road versus asphalt road includes the same batch size and shuffling procedure. In addition, we choose $T = 5$, $\alpha = 1$, and $\beta = 0.07$ as the hyperparameters for the teacher-student network. Finally, we use Adam as the optimizer, and we train the model for 125 epochs.

6.4 Performance Metrics

To evaluate the performance of VIVA, we use multiple metrics: *Accuracy*, *Recall*, *Precision* and F_1 *score*. Accuracy is defined as the total number of correct hits over all the examples, and can be calculated using Equation 6.1

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

where TP, TN, FP and FN stand for true positive, true negative, false positive, and false negative, respectively. However, since in this work the classes are not balanced, accuracy is not enough to judge the performance of our system. Specifically, we need other metrics to measure the amount of relevance in classified examples. To solve this issue, we also look at recall and precision as complementary metrics. Recall, also known as sensitivity, is defined as the fraction of the total amount of relevant instances that were retrieved, and can be calculated as follow:

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

Precision is about finding how many of the returned hits were correct and belonged to

	Precision	Recall	F1-score
Dry Asphalt	100%	100%	100%
Wet Asphalt	100%	99%	100%
Dry Gravel	99%	100%	99%
Wet Gravel	99%	100%	99%

Table 6.4: Results of image classification, on four kinds of roads using the ResNet50 model.

	Precision	Recall	F1-score
Dry Asphalt	98.90%	98.84%	98.87%
Wet Asphalt	98.58%	98.34%	98.46%
Dry Gravel	96.26%	96.10%	96.18%
Wet Gravel	94.35%	95.53%	94.95%

Table 6.5: Results of classification, on four kinds of roads, using a cross-modal supervised model that relies only on mmWave measurements.

our desired class. Precision can be calculated using the following equation:

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

Ideally, we would like to consider both precision and recall. Therefore, we also use F_1 *score* as an additional metric to evaluate our system. $F_1 score$ is defined as the harmonic mean of precision and recall, and can be calculated as follows

$$F_1 score = \frac{Precision \cdot Recall}{Precision + Recall} \tag{6.4}$$

6.5 Classification Results

We evaluated all the models on a test dataset. We first evaluated the image model, which provides supervision in training other models. Table 6.4 shows the precision, recall, and F1-

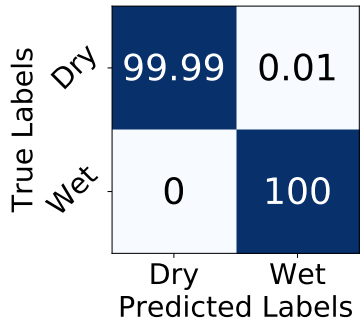


Figure 6.2: Results of classifying dry versus wet road using mmWave signals.

score for this classification task, while the overall accuracy of the model is almost 100%. Figure 6.3 shows the confusion matrix where rows present the true labels and columns present the predicted labels. These results show that the image model can supervise the training of the mmWave model and decrease the labeling costs by providing a perfect set of training targets.

We then evaluated the model that predicts a wet road versus a dry road. Table 6.3 shows the performance results of this classification. The results show that the model classifies the examples with 99.82% accuracy. In addition, recall for the wet class is almost 100% and no wet example is labeled as dry. Figure 6.2 shows the confusion matrix of the two label classification.

Once we had confirmed that our model can distinguish a dry road from a wet road with 100% accuracy, we examined whether it was also capable of detecting a gravel road from an asphalt road. Table 6.5 shows the results for this classification; VIVA is able to distinguish all four road conditions with 98% accuracy. Note that the recall for all of them is more than 95%. As shown in Equation 6.2, this means that we have very few false negatives. Figure 6.4 shows the confusion matrix of the four label classification.

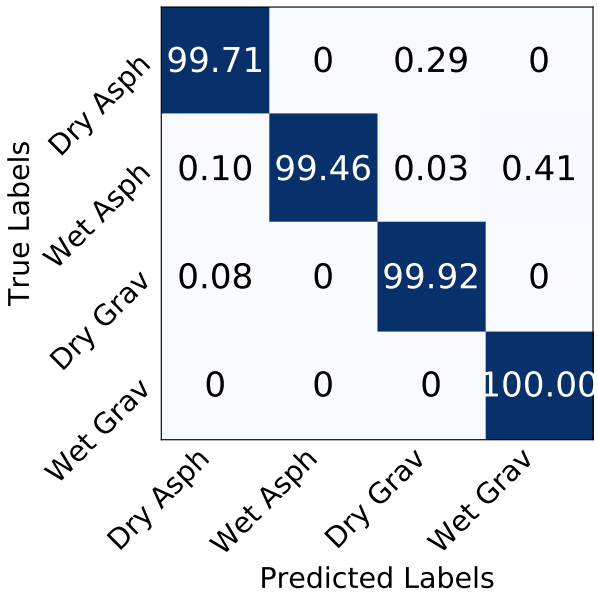


Figure 6.3: Results of classifying dry versus wet, and asphalt versus gravel for the image model (teacher).

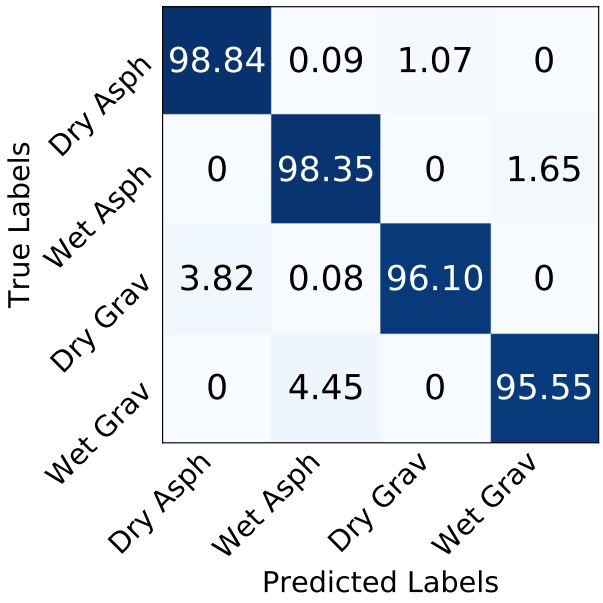


Figure 6.4: Results of classifying dry versus wet, and asphalt versus gravel using mmWave signals.

Chapter 7

Conclusion and Discussion

This thesis has introduced VIVA, a low-cost and accurate road sensing system that uses mmWave technology. In particular, VIVA overcomes the reliability and cost limitations of existing road sensing systems (such as vision-based and mechanical sensing approaches) to enable robust road sensing for driverless cars. VIVA achieves this robustness by taking advantage of the high-sensitivity of mmWave signals. However, the main challenge is that since mmWave signals are very sensitive, not only do changes in road surface impact mmWave signals but also other factors such as car vibration affect the measurements. To solve this problem and extract information from noisy measurements, we have designed a cross-modal supervised model to sense a road surface using a mmWave radar device. We have developed a prototype of VIVA and evaluated its performance in real-world experiments. Our results show that VIVA achieves more than 98% accuracy in detecting the road types and road conditions, even in scenarios in which existing vision-based systems fail (such as darkness and adverse weather conditions). Moreover, VIVA's sensor prototype costs less than $300[1], which is significantly cheaper than the cost of mechanical-based systems that achieve reasonable accuracy and robustness. Another advantage of VIVA compared to mechanical-based systems is that it can classify road surfaces not only in nor-

[1]We believe this cost will be significantly reduced in mass production.

mal driving scenarios but also in stationary or very low-speed scenarios. These scenarios are very challenging for the existing vehicle dynamic-based classification methods since they cannot capture enough measurements. Finally, although in this thesis we focus on road condition sensing, we believe VIVA can also make a significant contribution to the reliability and performance of vehicle stabilization, path planning, and guidance control systems. Below, we highlight some of the VIVA's limitations and potential development opportunities:

- *Unseen Classes:* VIVA can only detect four scenarios. However, there are more diverse situations to consider, including snowy or icy surfaces. One direction for future research is to improve the system by doing more measurements and building a more complex system.

- *Surface Roughness level:* Roads with the same kind of pavement might have a different level of roughness. For instance, the roughness of gravel sites are not the same, and the driving behavior must be adjusted based on the roughness. Otherwise, the car might experience serious damages. In future work, we plan to estimate the roughness level of surfaces based on mmWave measurements. This enables us to find a lower and upper bound for the roughness of different surfaces and build a more robust road sensing system.

- *Surface Moisture level:* Although VIVA can distinguish a dry surface from wet one, a reliable driving system requires a more comprehensive understanding of the road condition. Thus, in future work, we will scan a surface to estimate the moisture level of the surface with higher resolution.

- *Real-time Processing:* Our results show that VIVA is capable of real-time processing on a Raspberry Pi. However, due to the limitations of the developed software for the TI mmWave board, we were not able to capture mmWave measurements in real-time.

References

[1] Masahiko Aki, Teerapat Rojanaarpa, Kimihiko Nakano, Yoshihiro Suda, Naohito Takasuka, Toshiki Isogai, and Takeo Kawai. Road surface recognition using laser radar for automatic platooning. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2800–2810, 2016.

[2] Alireza Asvadi, Cristiano Premebida, Paulo Peixoto, and Urbano Nunes. 3d lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems*, 83:299–311, 2016.

[3] Karl Berntorp and Stefano Di Cairano. Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering. *IEEE Transactions on Control Systems Technology*, 27(3):1100–1114, 2018.

[4] Aleksandr Bystrov, Edward Hoare, Thuy-Yung Tran, Nigel Clarke, Marina Gashinova, and Mikhail Cherniakov. Road surface classification using automotive ultrasonic sensor. *Procedia Engineering*, 168:19–22, 2016.

[5] Long Chen, Mingyuan Bian, Yugong Luo, and Keqiang Li. Real-time identification of the tyre–road friction coefficient using an unscented kalman filter and mean-square-error-weighted fusion. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 230(6):788–802, 2016.

[6] Mooryong Choi, Jiwon J Oh, and Seibum B Choi. Linearized recursive least squares methods for real-time identification of tire–road friction coefficient. *IEEE Transactions on Vehicular Technology*, 62(7):2906–2918, 2013.

[7] Borja Gonzalez-Valdes, Yuri Álvarez, Yolanda Rodriguez-Vaqueiro, Ana Arboleya-Arboleya, Antonio García-Pino, Carey M Rappaport, Fernando Las-Heras, and Jose A Martinez-Lorenzo. Millimeter wave imaging architecture for on-the-move whole body imaging. *IEEE Transactions on Antennas and Propagation*, 64(6):2328–2338, 2016.

[8] Tianbo Gu, Zheng Fang, Zhicheng Yang, Pengfei Hu, and Prasant Mohapatra. mmsense: Multi-person detection and identification via mmwave sensing. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, pages 45–50. ACM, 2019.

[9] Lounell B Gueta and Akiko Sato. Classifying road surface conditions using vibration signals. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 039–043. IEEE, 2017.

[10] Alberto Y Hata and Denis F Wolf. Feature detection for vehicle localization in urban environments using a multilayer lidar. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):420–429, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[13] Yong Li, Weili Ding, XuGuang Zhang, and Zhaojie Ju. Road detection algorithm for autonomous navigation systems based on dark channel prior and vanishing point in complex road scenes. *Robotics and Autonomous Systems*, 85:1–11, 2016.

[14] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)*, 35(4):142, 2016.

[15] Feng Lin, Yan Zhuang, Chen Song, Aosen Wang, Yiran Li, Changzhan Gu, Changzhi Li, and Wenyao Xu. Sleepsense: A noncontact and cost-effective sleep monitoring system. *IEEE transactions on biomedical circuits and systems*, 11(1):189–202, 2016.

[16] Kuan Liu, Yanen Li, Ning Xu, and Prem Natarajan. Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730*, 2018.

[17] Shuo Liu, George Shaker, Safieddin Safavi-Naeini, and J Michael Chong. Low-cost gas sensors utilizing mm-wave radars. In *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, pages 1853–1854. IEEE, 2017.

[18] Y-H Liu, T Li, Y-Y Yang, X-W Ji, and J Wu. Estimation of tire-road friction coefficient based on combined apf-iekf and iteration algorithm. *Mechanical Systems and Signal Processing*, 88:25–35, 2017.

[19] Anil Kunnappillil Madhusudhanan, Matteo Corno, Mustafa Ali Arat, and Edward Holweg. Load sensing bearing based road-tyre friction estimation considering combined tyre slip. *Mechatronics*, 39:136–146, 2016.

[20] GA Magallan, Cristian H. De Angelo, and Guillermo O. García. Maximization of the traction forces in a 2WD electric vehicle. *Vehicular Technology, IEEE Transactions on*, 60(2):369–380, 2011.

[21] mmwavebeta(Group). mmwave Demo Visualizer. dev.ti.com/gallery/view/mmwavebeta/mmWave_Demo_Visualizer_Record/ver/2.0.0/.

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[23] Kanghyun Nam, Sehoon Oh, Hiroshi Fujimoto, and Yoichi Hori. Estimation of sideslip and roll angles of electric vehicles using lateral tire force sensors through rls and kalman filter approaches. *IEEE Transactions on Industrial Electronics*, 60(3):988–1000, 2012.

[24] Susmit Nanda, Sourav Manna, Arup Kumar Sadhu, Amit Konar, and Diptendu Bhattacharya. Real-time surface material identification using infrared sensor to control speed of an arduino based car like mobile robot. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pages 1–6. IEEE, 2015.

[25] Scott Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Eng, Daniela Rus, and Marcelo Ang. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1):6, 2017.

[26] Zhiquan Qi, Saied Taheri, Baofeng Wang, and Hongxiao Yu. Estimation of the tyre–road maximum friction coefficient and slip slope based on a novel tyre model. *Vehicle System Dynamics*, 53(4):506–525, 2015.

[27] Yiming Qian, Emilio J Almazan, and James H Elder. Evaluating features and classifiers for road weather condition analysis. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 4403–4407. IEEE, 2016.

[28] Shimul Saha, Helena Cano-Garcia, Ioannis Sotiriou, Oliver Lipscombe, Ioannis Gouzouasis, Maria Koutsoupidou, George Palikaras, Richard Mackenzie, Thomas Reeve, Panagiotis Kosmas, et al. A glucose sensing system based on transmission measurements at millimetre waves using micro strip patch antennas. *Scientific reports*, 7(1):6855, 2017.

[29] Liang Shao, Chi Jin, Cornelia Lex, and Arno Eichberger. Robust road friction estimation during vehicle steering. *Vehicle system dynamics*, 57(4):493–519, 2019.

[30] Inwook Shim, Jongwon Choi, Seunghak Shin, Tae-Hyun Oh, Unghui Lee, Byungtae Ahn, Dong-Geol Choi, David Hyunchul Shim, and In-So Kweon. An autonomous driving system for unknown environments using a unified map. *IEEE transactions on intelligent transportation systems*, 16(4):1999–2013, 2015.

[31] Rongrong Wang, Chuan Hu, Zejiang Wang, Fengjun Yan, and Nan Chen. Integrated optimal dynamics control of 4wd4ws electric ground vehicle with tire-road frictional coefficient estimation. *Mechanical Systems and Signal Processing*, 60:727–741, 2015.

[32] Teng Wei and Xinyu Zhang. mtrack: High-precision passive tracking using millimeter wave radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 117–129. ACM, 2015.

[33] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 14–26. ACM, 2019.

[34] Zhicheng Yang, Parth H Pathak, Mo Sha, Tingting Zhu, Junai Gan, Pengfei Hu, and Prasant Mohapatra. On the feasibility of estimating soluble sugar content using millimeter-wave. In *IoTDI*, pages 13–24, 2019.

[35] Zhicheng Yang, Parth H Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. Vital sign and sleep monitoring using millimeter wave. *ACM Transactions on Sensor Networks (TOSN)*, 13(2):14, 2017.