DETECTION AND CLASSIFICATION OF OBJECT PRESENCE AND CHARACTERISTICS IN A WATER CONTAINER USING HIGH FREQUENCY ULTRASOUND

by

MEHUL VISHAL SADH

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2022

To my father Vishal and my mother Sonal

who always pushed me to become my better self.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Manfred Huber for believing in me and constantly motivating me with new ideas, it was genuinely great working under his advisement. I want to thank my committee members Dr. David Levine and Dr. Vamsikrishna Gopikrishna for their interest in my thesis project and for taking time to serve in my thesis committee.

I am thankful for the knowledge that all the teachers provided me during my Bachelors back in India which shaped my intellect and interests. I also want to thank Mr. Prathamesh Tugaonkar for guiding and encouraging me to pursue graduate studies.

In the end, I am extremely grateful to my father, mother and my partner for their encouragement and sacrifice which enabled me to become the person I am today. I am also very grateful to my grandfather who ignited the spark of pursuing Science in me at a very early age which is the cornerstone behind my interests and inspiration. I also want to thank all of my friends who have helped me throughout my education.

May 13, 2022

ABSTRACT

DETECTION AND CLASSIFICATION OF OBJECT PRESENCE AND CHARACTERISTICS IN A WATER CONTAINER USING HIGH FREQUENCY ULTRASOUND

MEHUL VISHAL SADH, MS

The University of Texas at Arlington, 2022

Supervising Professor: Manfred Huber

Detection and characterization of soluble, diffuse, and solid objects and their characteristics in water has important implications in various applications, including water quality assessment and incontinence monitoring for health applications. In particular in the latter task, it is essential to be able to non-intrusively detect the appearance, presence, and consistency of materials in the water without the need for special purpose instruments or a special purpose setting. Rather, it would be important that sensing could be performed int he context of existing toilet systems. To achieve this, this work investigates the potential use of high frequency sonar sensors retrofitted to an existing, water-filled container to detect and characterize events where materials are added to the water, and to classify characteristics of the materials in terms of solubility, granularity, and density of the added materials.

A goal of the sensor choice here it to be non-intrusive, that they can be retrofitted to an existing container, and that they do not require significant cleaning

and maintenance. To address this, this work investigates the use of high frequency ultrasound as a sensing technology. The fact that humans cannot perceive sound waves above 20000 Hz, opens a way to use these waves for a wide variety of problems. Ultrasonic waves have been widely used in various applications such as nondestructive testing of materials, range finding (SONAR), sonography for medical imaging, fluid flow measurement and non-contact sensing. However, for most of these applications specialized devices and test settings have to be designed. Here we aim at using standard transducers attached to existing vessel to obtain the desired information by measuring the changes in absorption and diffraction characteristics as the ultrasonic waves pass through the introduced materials.

Understanding these wave patterns after passing them through materials can here provide a way to understanding unique features of the objects and materials introduced into the water. To allow for the detection and classification of important material properties, frequency features are here extracted using Fourier Transforms and then used to train a model to detect the introduction of objects into the water and to classify the object characteristics while underwater. The goal here is to investigate the extent at which various kinds of objects with different properties can be classified on the basis of the wave patterns that the objects create when submerged underwater, and to build a prototype system that could be retrofitted to existing containers to perform such measurements.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

Ultrasonic waves are used in various applications such as range finding (SONAR), medical imaging (Sonography), flaw detection in metals, fluid flow monitoring, non-contact sensing and non-destructive testing of objects. These applications utilize different properties of ultrasonic waves to solve their respective problems. For example, SONAR looks for reflections of the waves to sense objects underwater. In another example, ultrasonic waves are propagated through metals to find anomalies. The high frequencies propagated behave as predicted when passing through the metals and can help identify the flaws in the metals [1]. The wave patterns made after passing through the materials can thus provide insights about the properties of the respective materials.

Since humans cannot perceive sound waves above 20000 Hz, it enables the use of high frequency ultrasonic waves in health monitoring applications. Currently, there are some systems built to analyze the characteristics of a person's bodily waste by the use of a "Smart Toilet" [2]. The "Smart Toilet" has various sensors mounted to it which can process stool sample images through the use of CNNs. By extracting the features of the bodily waste, it tries to identify for signs of diseases by detecting anomalies and patterns. Although it can be effective, the use of image sensors attached to a toilet can make the user uncomfortable due to the risk of invasion of privacy despite the data being collected in a secure encrypted cloud-based system. Hence, there is a need of non-invasive monitoring that avoids the use of image pro-

cessing in a private setting and that's where the use of ultrasonic signal processing comes into play.

1.2 Purpose

In this Thesis project, we aim to find the extent of using raw high frequency ultrasonic data to non-invasively detect and identify the material being introduced in water. The property of propagation of ultrasonic waves through materials can give us information about the material's characteristics. By exploring the extent of identification for materials of varying solubility and granularity by using ultrasonic waves, it can help us understand the prospective applications it can have on health monitoring systems. The conclusions from this research will give us insights on how ultrasonic wave patterns can help identify different characteristics of materials.

1.3 Objectives

The aim of this thesis is to investigate the extent at which the material can be detected and later identified inside water. The project is focused on building an experimental setup for collecting raw ultrasonic data for different kinds of materials with varying solubility and density. Then this raw ultrasonic data will be further pre-processed to clean it before we try to generate insights from it. The classification task will be solved using neural networks. We are using only the raw ultrasonic data from ultrasonic piezoelectric sensors.

The first part is to identify the stages at when a material is introduced into the water and at which point it is stopped being inserted inside the water. The next part is to then identify the material on the basis of wave patterns created at those

separate segmented events. The data from these separate segmented events for each material will then be used to classify these materials.

An experimental setup will be built to introduce different kinds of materials in a water container and sensors have to be mounted to the container to transmit and record the propagated ultrasonic waves through the materials. All the receivers must record the data in synchronization so that the phase information could also be recorded at the same time which will be beneficial later while classifying.

The major issue with the raw ultrasonic data is that since it is high frequency data, it contains a lot of information and will be unfit to train the neural network because of high dimensionality. We must find a way to reduce the dimensionality while still be able to retain most information about the signal, which will also be important to reduce the computation time.

The recorded data will be transformed to extract significant frequency and phase information of the multiple sensors. The frequency responses at several time slices will be used to analyze the change in events of the signal [1]. After isolating the events, the data from these individual events will be used to train the multiple neural networks separately for these events. Then the results will be evaluated on the basis of the neural networks being able to identify the materials.

1.4 Scope

The project deals with identifying materials only on the basis of raw ultrasonic signals recorded using the high frequency piezoelectric ultrasonic receivers. The final result from the neural networks will only output the classification of the materials. The neural networks cannot determine the position of the material underwater. Net-

[1] the events refer to the different phases in signal (i.e., 1. When material is not yet introduced in water. 2. Material being introduced in water. 3. Material stopped being introduced in water.)

works are trained to classify the events on the basis of 8 classes: Rock, Soap, Metal Wire Mesh, Milk semi-solids, Flour, Sand, Bread Crumbs and Water. The data is collected solely by ourselves in the experimental setup and nothing is outsourced externally except the components required in building the experimental setup. The results of the experiments and classification will investigate the extent to which our fore-mentioned method is successful in identifying the different states and the extent of classifying the materials.

1.5 Outline of document

This document consists of 4 main sections:

Chapter 2: Background This chapter will provide the reader with an introduction to basic concepts and terminologies used throughout the course of this thesis project. It also provides the user with theoretical background and key concepts to understanding the overview of nuances in signal processing and classification techniques in general as well as their evaluation methods.

Chapter 3: Methodology This chapter gives a detailed explanation for the methods and experiments performed for the thesis work. It explains the problems faced in specific scenarios and techniques developed to overcome those.

Chapter 4: Results This summarizes the results obtained after performing the experiments in the methodology chapter. The results are a comparison between performance of different clustering algorithms. It also provides the results from both major tasks and gives an insight on how specific issues were dealt with.

Chapter 5: Conclusions This provides a brief view of the entire thesis work and gives insights into possible future work.

CHAPTER 2

BACKGROUND

2.1 Overview

This chapter gives an introduction to all the methods, concepts and components used in this project. It gives a basic explanation of the working of ultrasonic sensors and the various pre-processing and post-processing concepts used throughout in this thesis.

2.2 Ultrasonic equipment

Piezoelectricity is the electric charge that accumulates in certain materials under mechanical stress. An Ultrasonic sensor is made of piezoelectric material and hence it can convert mechanical energy into electrical energy which enables it to sense ultrasonic sound waves. There is also a converse effect of inverse piezoelectricity which enables the ultrasonic sensor to convert electrical energy into mechanical energy which enables it to produce very fast vibrations and cause it to emit ultrasonic waves. This behaviour of the ultrasonic sensors is generally used to detect objects in space by sensing the reflected ultrasonic wave from the objects after emitting those waves. Since we know the speed of sound, this application is used to sense objects and also tell the distance from the sensor to the object when an echo is received from the object. This distance d to the object from the sensor is calculated by the equation

$$d = \frac{v(t_2 - t_1)}{2} \tag{2.1}$$

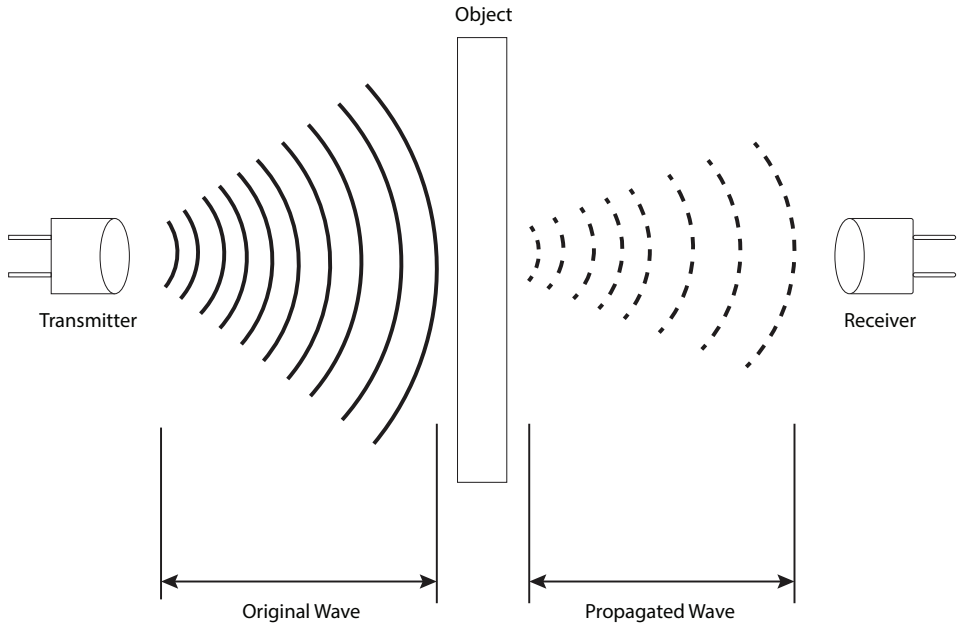


Figure 2.1. Ultrasonic transmitter and receiver setup for propagating waves.

where $(t_2 - t_1)$ denotes the time taken starting from when the sensor transmitted the wave until the wave's echo reaches back the sensor, v denotes the velocity of sound in the respective medium.

Hence, an ultrasonic transducer system consists of a transmitter and a receiver. The transmitter and receiver can also be housed in the same package which can emit and receive ultrasonic waves. Fig. 2.1 illustrates the setup which is used to record the characteristics of the propagated wave. We will also record the reflected wave pattern from the object to another sensor so that we get more characteristics of the object from the waves as depicted in Fig. 2.2.

2.3 Signal characteristics and pre-processing

The signals which are recorded from the sensors can be very noisy and hence we need to do some pre-processing on the signal before performing our detection and classification tasks. It will play a very important role in helping us extract vital

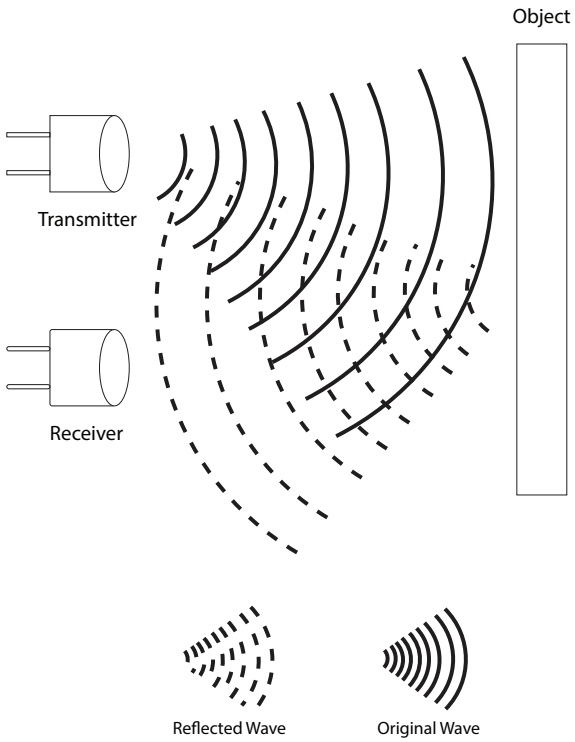


Figure 2.2. Ultrasonic transmitter and receiver setup for reflected waves.

information from the signal. Since we are dealing with the signal in digital format, this project will utilize all digital signal processing concepts opposed to analog processing.

The recorded signal can be represented in the frequency domain as well as in the time domain. In the time domain, we can see the signal's amplitude varying with time and it is the most commonly used. In the Frequency domain, we describe the signal's frequency varying with time. Both domains represent different characteristics of the signal. The amplitude can represent the power, voltage, pressure, etc. of the signal.

2.3.1 Analog to digital conversion

Almost all the signals that are found in nature are in the analog form, which represents continuously time-varying property. These analog signals have to be con-

verted to digital representations to enable further processing in digital systems. The analog signal has to be sampled at a desired sampling frequency to represent it digitally. This sampling frequency should be high enough so we don't lose the resolution of the analog signal.

This conversion is performed using an analog to digital converter (ADC). It is an electronic integrated circuit which is used to convert analog signals in the form of voltage to digital form in binary representation. The ADC converts the voltage received at the source to its appropriate digital representation with the desired resolution. The result we get from the ADC is a series of conversions from analog to digital values with the specified sampling frequency. It can be seen in the Fig. 2.3 that the discrete values are obtained when converting the analog to digital signal on linear time steps.

While converting from Analog to Digital, it is very important to choose the sampling rate and sampling resolution according to the signal frequency to obtain the most accurate representation of the signal without losing much information. If the sampling frequency is not high enough, a problem of aliasing can occur which can cause us to lose information about the signal as illustrated in Fig. 2.4. It can be seen that a lot of data points between the periods are missed and we get inaccurate information about the signal. The Nyquist theorem specifies that a sinuisoidal function in time or distance can be regenerated with no loss of information as long as it is sampled at a frequency greater than or equal to twice per cycle. The relation is explained by the equation

$$f_{SAMPLE} > 2 \times k \times f_{SIGNAL(MAX)} \tag{2.2}$$

where f_{SAMPLE} is the sampling frequency, $k > 1$ is the over-sampling factor and $f_{SIGNAL(MAX)}$ is the highest frequency signal that we are interested in the original signal.

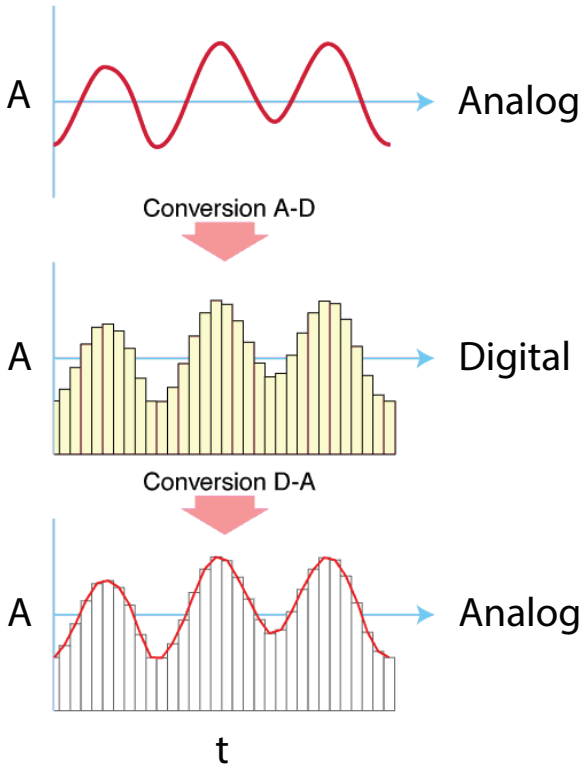


Figure 2.3. Conversions of Analog-to-Digital and Digital-to-Analog visualized from Wikipedia [8].

Further, it is also vital to filter the noise in the signal before processing it for further tasks. Analog filtering should be performed to get a cleaner signal at the source. It is easy to implement with simple analog components. At the last, the resolution of the ADC conversion also plays and important role as more resolution provides us more granularity on the signal with more discrete steps in the digital counterpart. With a higher resolution of quantization of ADC, we can extract more information

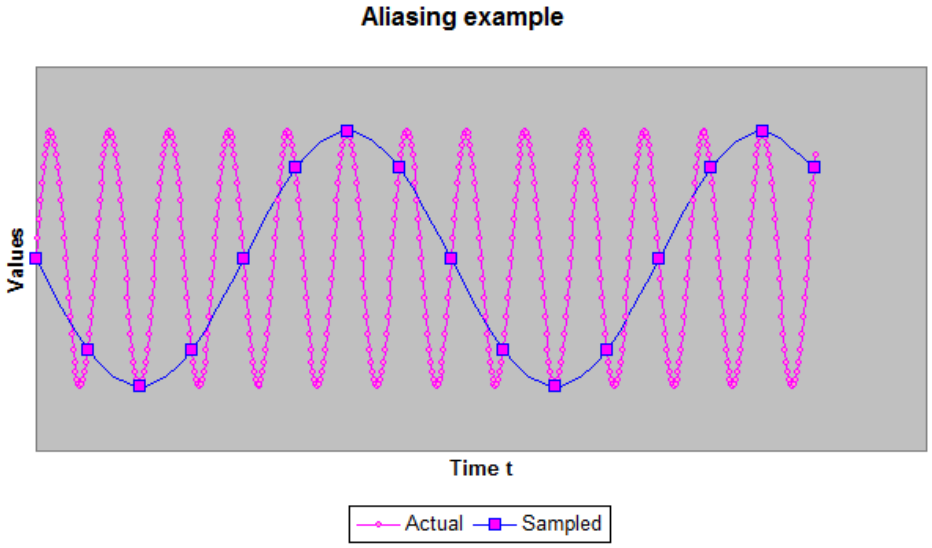


Figure 2.4. Effect of aliasing on actual signal illustrated from Sampling, Aliasing, and Analog Anti-Alias Filtering [9].

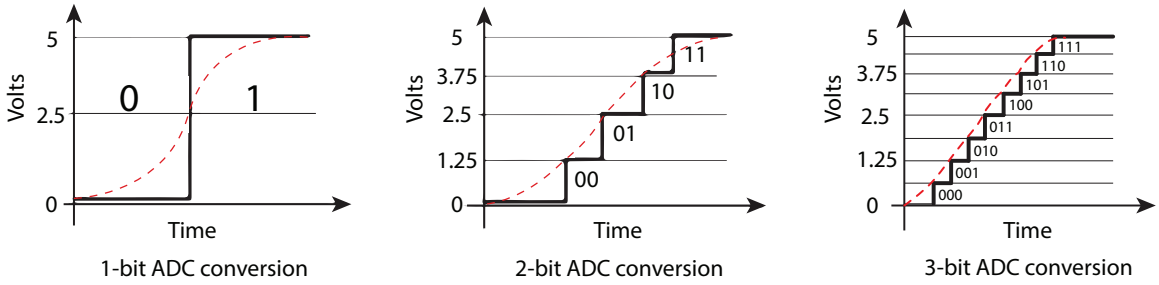


Figure 2.5. Different resolutions of ADC conversion.

from the signal. As resolution increases, we get a signal which will correspond to the original signal more accurately. Increasing resolution steps is illustrated in Fig. 2.5

2.3.2 Processing in Frequency Domain

The transformation of a time domain signal to the frequency domain can be estimated by an operation known as the Fourier transform. Depending upon the nature of the time signal, whether it is discrete or continuous, periodic or aperiodic, we can perform different types of Fourier Transforms. There are four types of Fourier

Transforms as denoted in the Table 2.1. Detailed information can also be found in [10].

Table 2.1. Types of Fourier Transforms

Transform	**Nature of time domain signal**	**Nature of frequency spectrum**
Fourier Transform (FT)	continuous, non-periodic	non-periodic, continuous
Discrete Time Fourier Transform (DTFT)	discrete, non-periodic	periodic, continuous
Fourier Series (FS)	continuous, periodic	non-periodic, discrete
Discrete Fourier Transform (DTFT)	discrete, periodic	periodic, discrete

In the above mentioned Fourier Transforms, the result that we get contains a real part and a complex part. The real part of the Fourier transform inputs real numbers and outputs two sets of realfrequency domain responses, where one set contains co-efficients over sine function and the other set contains co-efficients over cosine function. The complex part of the Fourier response contain positive and negative frequencies in a combined array. The complex part can accept both real and complex signals. The Fig. 2.6 illustrates the real and complex parts of the DFT.

2.3.2.1 Real DFT

In the scenario of a N-point DFT, it will take N samples of real valued time domain signal and will give two responses of length $N/2+1$ each for cosine and sine respectively.

$$X_{Re}[k] = \frac{2}{N}\sum_{n=0}^{N-1} x[n] cos\left(\frac{2\pi kn}{N}\right) \tag{2.3}$$

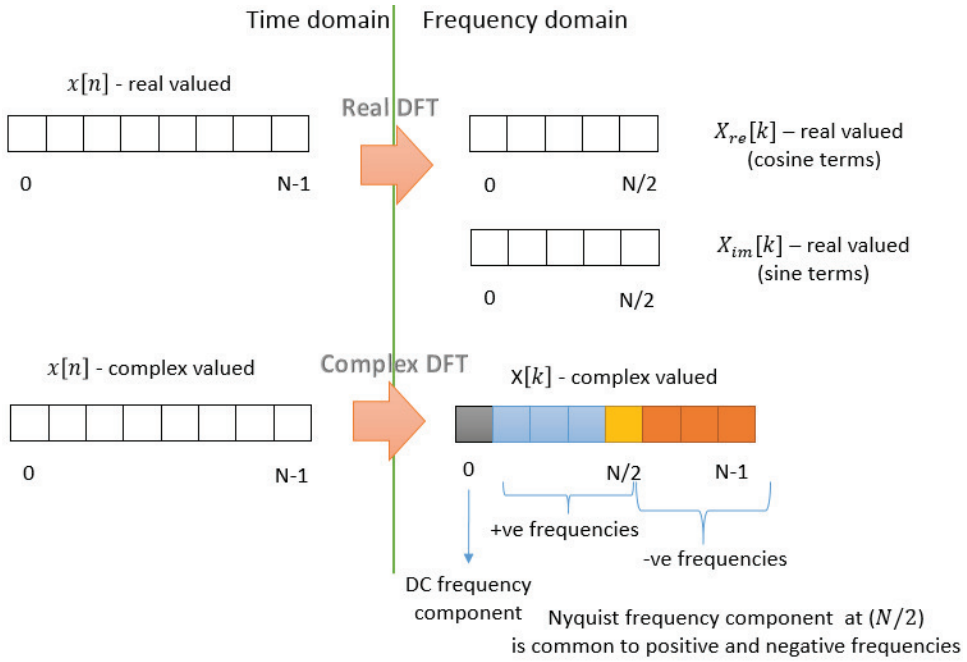


Figure 2.6. Real and Complex DFT illustrated from [10].

$$X_{im}[k] = -\frac{2}{N}\sum_{n=0}^{N-1} x[n] sin\left(\frac{2\pi kn}{N}\right) \tag{2.4}$$

where the time domain n goes from $0 \rightarrow N$, the frequency domain k goes from $0 \rightarrow N/2$

Hence, the real valued time domain signal $x[n]$ is represented from the real DFT sets as

$$x[n] = \sum_{k=0}^{N/2} X_{Re}[K] cos\left(\frac{2\pi kn}{N}\right) - X_{im}[K] sin\left(\frac{2\pi kn}{N}\right) \tag{2.5}$$

2.3.2.2 Complex DFT

For the case of N-point *complex* DFT, it takes N complex valued samples in time domain $x[n]$ and returns an array $X[k]$ of length N.

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}} \tag{2.6}$$

where $X[0]$ represents DC frequency part, the following $N/2$ terms are positive frequency components and the term at position $X[N/2]$ is the Nyquist frequency (equal to half of sampling frequency), the final remaining $N/2 - 1$ terms are negative frequency components (Here, negative components represent that the phasors are rotating in opposite direction and can be omitted). The corresponding equation for obtaining time domain signal can be denoted as,

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{\frac{j2\pi kn}{N}} \tag{2.7}$$

2.3.2.3 Application

The output from an ADC is discrete therefore it is necessary to use the discrete time Fourier transform (DTFT). Since it will be very difficult to determine the accurate DTFT of a discrete signal $x(n)$, it is more straightforward to analyze a finite number of elements and then present the results to compare with the DTFT. This is discrete Fourier transform (DFT). The DFT is defined as,

$$X(k) = DFT\left[x(k)\right] = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \tag{2.8}$$

where N is the number of samples. Also, $X(k)$ is N-periodic, i.e. $X(k+N) = X(k)$. In applications, an efficient way of calculating the DFT is to use the Fast Fourier Transform (FFT) which is a algorithmic scheme which requires less operations and can be utilized from python's scipy library. The DTFT is given by equation,

$$X(\omega) = DTFT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n \Delta t} \tag{2.9}$$

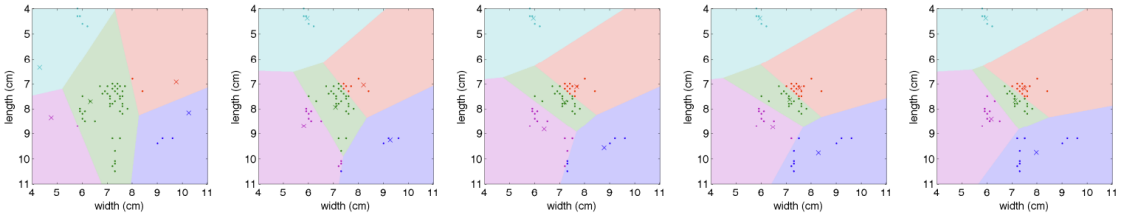
where ω is of unit radians per second.

2.4 Classification Techniques

Various techniques for classification in different scenarios are implemented during the course of this thesis. In this thesis, we will compare the clustering accuracies of various different clustering algorithms to segment events in our application. These techniques are further discussed in this section.

2.4.1 K-Means Clustering

K-Means clustering is widely used in various applications used to group a set of similar points and build K defined clusters of respective similar points. Clusters in K-Means are formed on the basis of initialized centroids and then the centroids are iteratively updated along with the clusters until the centroids don't change and we have stable clusters. The clusters are formed on the basis of distance measures between the cluster centroid and all the points. The points are further assigned to the clusters which have the smallest distance with the respective cluster center. Most commonly used distance measure in K-Means clustering is Euclidean distance. We have N points data in X, hence we can denote the n^{th} of these data points as x_n, so our dataset will be the set $\{x_n\}_{n=1}^{N}$. K-Means will assign each of these data points to a cluster. Hence, we want to find the best possible assignment for a point to a cluster and thus we have a binary **responsibility vector** r_n for all individual points. This vector will be zero for all clusters except to the cluster it is assigned to, where it will be 1. This is also known as **one-hot encoding** where all values are zero except for one place in a binary vector. The algorithm for K-Means clustering is summarized in Algorithm 1. The iterations of K-Means is visualized in Figure 2.7.



(a) Initialization (b) Iteration 1 (c) Iteration 2 (d) Iteration 3 (e) Iteration 4

Figure 2.7. Four iterations of K-Means applied to the lengths and widths of fruit (oranges and lemons), as measured in centimeters. Here, K = 5 and means are initialized randomly, illustrated from [11].

Algorithm 1 K-Means Clustering

Input: Data vectors $\{x_n\}_{n=1}^{N}$, number of clusters K
for $n \leftarrow 1$ ... N **do** ▷ Initialize all of the responsibilities.
    $r_n \leftarrow [0, 0, ..., 0]$ ▷ Zero out the responsibilities.
    $k' \leftarrow$ RandomInteger(1, K) ▷ Make one of them randomly one to initialize.
    $r_{nk'} = 1$
end for
repeat
    for $k \leftarrow 1$ ... K **do** ▷ Loop over the clusters.
        $N_k \leftarrow \sum_{n=1}^{N} r_{nk}$ ▷ Compute the number assigned to cluster k.
        $\mu_k \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n$ ▷ Compute the mean of the kth cluster
    end for
    for $n \leftarrow 1$ ... N **do** ▷ Loop over the data.
        $r_n \leftarrow [0, 0, ..., 0]$ ▷ Zero out the responsibilities.
        $k' \leftarrow argmin_k \|x_n - \mu_k\|^2$ ▷ Find the closest mean.
        $r_{nk'} = 1$
    end for
until none of r_n change
Return assignments $\{r_n\}_{n=1}^{N}$ for each column and cluster means $\{\mu_k\}_{k=1}^{K}$

K-Means requires a priori information on the number of clusters to start picking random K centroids and assigning clusters to points. The initialization of the cluster centers plays a major role in efficient well-separated clustering and thus must be initialized based on some factors. Further details are discussed in [11].

2.4.2 Agglomerative Hierarchical Clustering

Agglomerative clustering is a type of Hierarchical clustering where clusters are generated in a bottom-up approach. Hence, each point is initially considered as a cluster of its own known as **leaf**. Then the points are merged into bigger cluster **nodes** in an iterative approach until all points are combined into a single cluster. The visual representation of these clusters is denoted by a **Dendogram**. Generally, the clusters are formed on the basis of similarity measures such as Euclidean distance or Manhattan distance. The main attributes considered while performing Agglomerative clustering is distance measure and linkage. The distance measure is same as the similarity measure discussed earlier. The linkage function will describe how the clustering is performed. Linkage function takes the distance measure of the points and then performs clustering on the basis of the function used. There are 4 major types of linkage functions:

- **Maximum or complete linkage** : The maximum value of pairwise distance is taken between points in a one cluster and another cluster to form a bigger cluster. It mostly produces compact clusters.
- **Minimum or single linkage** : The minimum value of pairwise distance is taken between points in a one cluster and another cluster to form a bigger cluster. It mostly produces loose clusters.
- **Centroid linkage** : The distance between centroids of two clusters is taken as the distance between two clusters. The centroid is the mean vector of the points.

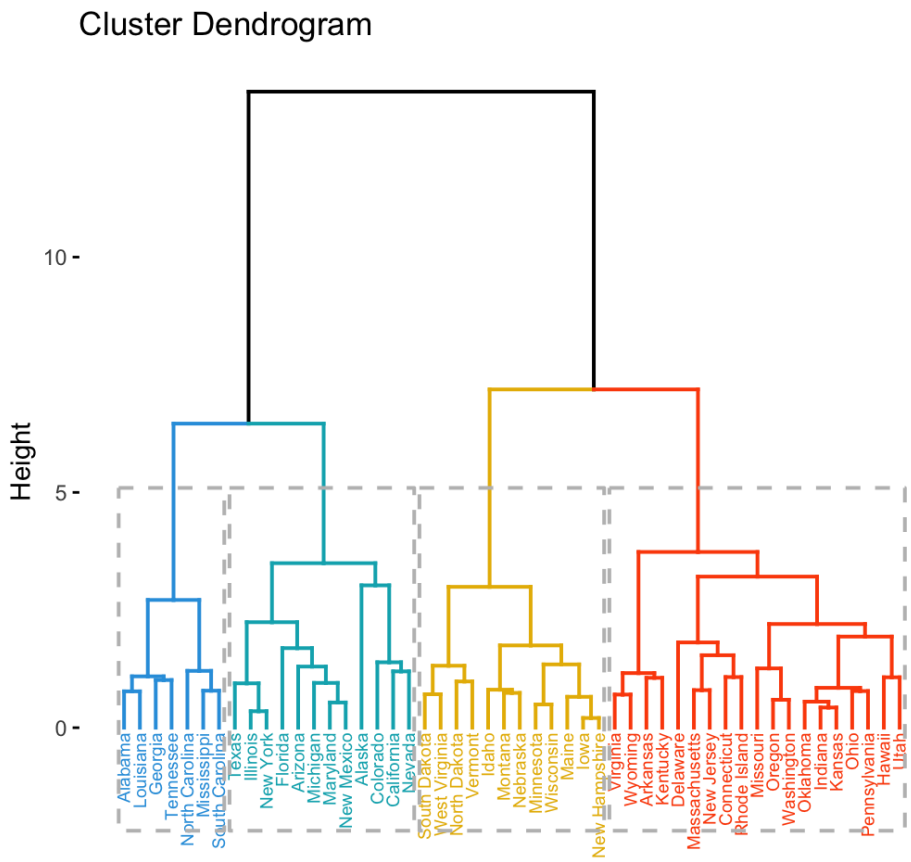


Figure 2.8. Dendogram being cut to obtain 4 clusters, illustrated from [12].

- **Ward's method** : This method minimizes the within-cluster variance. Clusters with minimum internal variance are merged at each iteration.

The main issue with hierarchical clustering is that it doesn't tell us how many clusters are obtained, hence the dendogram should be cut at the point according to the number of needed clusters. The results of the agglomerative clustering is represented graphically in the form of dendograms. It is illustrated in the Fig. 2.8

2.4.3 Neural Networks

Many Machine Learning algorithms are built upon Neural Networks. The name of Neural Networks is derived from the neurons in the human brain, because in a

sense, the neurons in an Artificial Neural Network fire outputs based on some activation function. They are used in various applications ranging from weather forecasting, natural language processing to computer vision and being able to understand images. Deep Neural Networks are a type of Neural Network which has a larger number of consecutive layers which can either be fully connected (i.e., each neuron is connected to all others between consecutive layers) or partially connected such as in convolutional networks or in multi-branch architectures. There can be various hidden layers inside a Neural Network and these layers get input sequentially in the order that they are arranged. The first layer to get an input directly from sample is the input layer. The layers take input from their previous layer, then compute a new value based on some activation function and pass the output to the next layer. Each neuron in the Neural Network has a weight and bias, based on these two factors the respective neuron will have weightage in regard to the whole classification result. A fully-connected Neural Network is illustrated in Fig. 2.9. The parts involved in training a Neural Network are explained further.

2.4.3.1 Activation functions for Neurons

The activation function enables a neuron to calculate a value and behave particularly for a respective input and then it can pass the output to the next layer's neurons. The activation function can be linear and can also be non-linear. The most widely used activation function is the sigmoid function $\sigma(x)$ which is non-linear and is given as,

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.10}$$

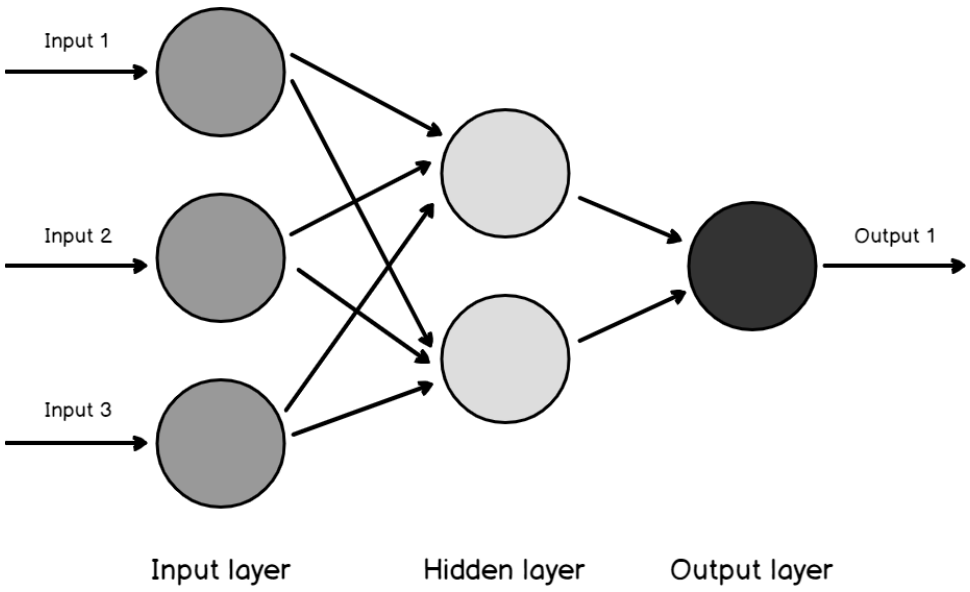


Figure 2.9. A simple fully connected Neural Network, illustrated from [13].

Another activation function which gives good results for classification problems is the softmax function, which is generally used in the last layer of the Neural Network for a multi-class classification problem,

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.11}$$

In problems where we want to generate a probabilistic output of the probabilities of the object belonging to multiple classes, we use the softmax function for the last layer. We get a vector presenting the probabilities of those classes. Whereas Sigmoid is used for binary label classification or for one-vs-rest classification in multi-class problems when the input might contain multiple classes and probabilities might not make sense in that case.

2.4.3.2 Training the Network

A Neural Network is trained on the basis of some parameters of the neurons where the network adjusts those parameters to learn the classification task and rep-

resent the model that we want. Forward Propagation and Back Propagation enables a Neural Network to adjust its weights and bias for a neuron which are learnable parameters and help to determine how much changing these parameters affects the end result. The input to a neuron is denoted by x_i, which is multiplied with the neuron's weight w_i and further added with the neuron's bias b_i. This result is then passed through an activation function, ex. $\sigma(x)$, that sends the output to the next layer's neuron. The output from the neuron is given as $y = \sigma(w^T + b)$. This y becomes the input to the next layer's neuron and so on. This is called Forward Propagation and it produces the classification result at the end of the layer. A loss function is monitored to see how well the network performed in classification. Most commonly used loss is mean squared error (MSE). It is given as,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i' - y_i)^2 \tag{2.12}$$

where y_i' is the ground truth (actual value), y_i is the predicted value and N is the total number of samples being used for training. MSE performs well until the values from features are nearly normally distributed. If they are not normally distributed, the loss function will produce higher loss values due to squaring them. Many losses are tailored for specific issues and in such cases where we are dealing with randomly distributed data, the categorical cross entropy loss performs good since y' is a one hot encoded vector with same length as number of classes, and assigns 1 for the correct class and 0 for others. It is given by the equation,

$$CCE = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} \times log(y_{ij}')) \tag{2.13}$$

Now, for updating the trainable parameters in the network, Backpropagation is performed which represents computing the gradients of the loss function w.r.t. the parameters. The weights will be changed along with biases to minimize the loss in the next iteration.

To optimize the loss function, we need to update weights and biases but we don't know how much to update them as loss does not give us any information about updating the parameters. Hence, optimization algorithms come into play and they can be used to determine the updation of the parameters. Most common optimization algorithms use gradient descent which takes a step in the direction of the gradient and tries to optimize the loss function by updating weights and biases. Adam (adaptive moment estimation optimizer) is an algorithm which uses some parts of Adagrad and RMSprop. Adam uses the momentum from Adagrad which updates the parameters to preserve the gradient trajectory. From RMSprop, Adam follows the approach of moving in smaller steps in steep gradient descents and longer steps in shallow descents to maintain the balance between overstepping and slow stepping.

2.5 Evaluation Metrics

Evaluation metrics are discussed for our fore-mentioned classification techniques.

2.5.1 Silhouette Coefficient

The general goal for an optimal clustering algorithm is to minimize the intra-cluster distances and maximize the inter-cluster distances. It is basically the principle of keeping similar objects together and dissimilar objects far apart. To measure this property for clustering, Silhouette coefficient is used.

The Silhouette Coefficient can tell us the measure of how properly a point is assigned to its cluster. It is given by the equation,

$$S(i) = \frac{b_i - a_i}{max\left\{a_i, b_i\right\}} \tag{2.14}$$

where a is the mean distance between the sample and all the other points in the cluster, b is the mean distance between the sample and all the other points in the next nearest cluster.

The Silhouette Coefficient ranges between -1 to 1, where -1 denotes that the point is incorrectly clustered and 1 denotes that the point is correctly clustered and well-assigned. This score is also highest when the clusters are properly separated from each other and have high density.

2.5.2 Accuracy of classification

It is the most common metric used to evaluate the performance of classification by a model. The calculation of accuracy is done by taking the ratio of correct predictions with total number of predictions.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Predictions} \tag{2.15}$$

The major issue with accuracy is that it cannot give us the granularity of which classes were actually predicted wrong as opposed to other classes, hence we won't know the poor performance of the model with respect to specific classes and it could become difficult in improving the performance for better classification of those classes.

2.5.3 Confusion Matrix

A confusion matrix can provide us more detailed information with respect to classification metrics for each individual class. It is a matrix with true class labels as rows and predicted class labels as columns. For example, when the class is correctly predicted, it is a true positive prediction. When a class is wrongly predicted, it is a false positive. This can be explained in a matrix as shown in Table 2.2,

Table 2.2. Confusion matrix with 2 classes

Total Predictions = 20	Predicted: A	Predicted B
Actual: A	5	3
Actual: B	2	10

where True positive = 5 (class 1 is predicted and it is actual class 1), False Negative = 3 (class 2 is predicted but actual is class 1), False Positive = 2 (class 1 is predicted but actual class is class 2) and True Negative = 10 (class 2 is predicted and actual class is class 2).

With the use of confusion matrix, we can also derive the performance of the model by other metrics such as F1 Score given by the equation,

$$F1 = 2 \cdot \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{2.16}$$

Further, Precision and Recall is given by the equations,

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2.17}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{2.18}$$

Precision can be said to be the ratio of the instances which are relevant. Whereas, Recall is the ratio of the relevant elements that were obtained.

2.6 Related Work

In medical field, there are a lot of applications which deal with monitoring anomalies in human excreta to detect early signs of diseases. Most of those deal with the chemical analysis of stool samples to identify its characteristics. Doing so is necessary, but it can require maintenance of the systems involved in testing the

stool samples and there are very few options which can be retrofitted over existing toilet systems to collect samples. One such example is "Smart Toilet" which is a new disease-detecting technology [2]. It is a smart toilet, but not the automated kind which lifts its lid for use; this toilet involves an array of sensors for detecting characteristics of stool sample involved. The system first identifies the user by their "anal print" where the image of anus of the user is taken and then user is identified to start monitoring their samples. Further, the stool samples are captured on video by the sensors and try to extract characteristics based on image processing. While the system claims to store all data in a secure cloud-based system, it can be very risky in terms of invasion of privacy if the data gets leaked since all digital systems are still vulnerable to attacks. This data is very intimate and user can feel uncomfortable using the toilet when image scanners are involved in such a private setting. This thesis project aims to be as non-invasive as possible without compromising user's privacy, so they feel comfortable using the attachment. The project will work as an attachment that can be retrofitted to existing toilet systems and since this will be fitted in the external part of the toilet, requirement of maintenance will be very low.

Another example of a smart toilet technology is to monitor the stool sample after it has been flushed down from the toilet as denoted in [5]. Stool monitoring can be important in long term care facilities where patients might not remember the color and characteristics of their stool and how often they had a bowel movement, which can create a problem because gastroenterologists rely on patient's self-reported information to determine the cause of their health issues relating to gastrointestinal tract. So, to solve this, they introduce a smart toilet technology which can monitor and scan stool samples when a person flushes the toilet and then images are taken of the stool passing through the pipes. The researchers gathered up to 3,328 unique stool images and then the gastroenterologists annotated these images according to

Bristol Stool scale (scale for stool classification). Further, these images were used to train a convolutional neural network to classify the stool form. This research was completely based on image processing, hence the limitations of gathering images and maintenance of the equipment will have to be taken in consideration for this system and can be restrictive. Moreover, placing cameras in pipes creates a maintenance problem as lenses and other optical elements have to stay clean.

Smart toilets could also be potentially used for passive monitoring of COVID-19 as discussed in [6]. Since primary manifestations of COVID-19 involve gastrointestinal symptoms, a number of studies have shown that remnants of SARS-CoV-2 can be found in stool samples even after 5 weeks of COVID-19 RNA tests becoming negative. This can possibly provide a way for monitoring COVID-19 spread with user's consent. A smart toilet system has been proposed which can involve automated stool sample collection and try to detect traces of COVID-19 through testing of stool samples after user's consent. Since it requires the actual collection of stool sample, a lot of maintenance will be involved on a daily basis and can become a hassle, alternative ways to building a low maintenance system must be explored.

Further, signal processing has been done in applications involving ultrasonic data for object recognition in robotics field where identifying objects in a real world scenario might help in better path planning [3]. The ultrasonic sensor data is processed to generate insights on what the object might be and its physical properties. But since our ultrasonic sensors deal with high frequency data of 300kHz, the sample rate must be around twice that of the highest frequency that we want to extract according to Nyquist Theorem. So, that would mean we will generate up to 600,000 data points per second for each sensor which will cause a huge problem due to high dimensionality for Machine learning algorithms. Hence, we must also explore dimension reduction techniques as in [4] for dealing with big data.

In a comprehensive research of [7], a study was done to determine material classification using the reflections of ultrasonic signals. The paper involved study of echo signal amplitude and the measurement of time of flight to classify five objects: aluminium, concrete, wood, glass and steel. The decay in the echo of the ultrasonic signal was computed to extract characteristics of the object it echoed from. Then the reflection co-efficient is modeled to determine the total intensity reduction of a signal. Echoes at varying distances were recorded for all the materials and then, from the data, the relation between distance and magnitude of the reflected signal is used to predict the kind of material. This research uses the reflection of ultrasound to classify materials. Although the cited research deals with material classification for the purpose of map building in robotics, this can also be extended to different applications where detection and classification might be involved. We will try to implement a hybrid of this approach where we look for the propagated signal along with reflected signal to extract characteristics in the frequency domain of ultrasound signal.

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter explains the experimental setup, the hardware, the data collection methodology and the data pre-processing steps, and the various tests performed for classification tasks during the course of testing and experimentation.

3.2 Equipments

Ultrasonic waves are generated using high frequency ultrasonic sensors with the help of a Teensy microcontroller which is used to generate high frequency signals. Data is recorded using the same Teensy microcontroller coupled with the receiving ultrasonic sensor. The transmitting and receiving circuit are separate, but are connected to the same microcontroller.

3.2.1 Teensy microcontroller

The Teensy microcontroller version 4.0 is used for the course of this project as a wave function generator as well as a High speed Data Acquisition device. The transmitting ultrasonic sensor is connected to a voltage level shifter which is increasing the voltage of the signals being sent by the Teensy Microcontroller as it is only capable of sending 3.3V logic signals. The ultrasonic receiving sensors are both connected to the Teensy Microcontroller and then it converts the analog to digital signal and sends it via USB to a laptop where data is stored.

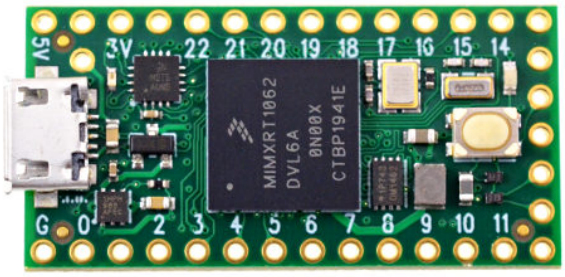


Figure 3.1. Teensy Microcontroller.

3.2.2 Ultrasonic sensors

The ultrasonic sensors used in this project is muRata MA300D1-1 which is a ultrasonic transducer (transmitter and receiver housed into the same package) with a center frequency of 300kHz and a peak input voltage of 50V. For further details, this datasheet can be referred [14]. 1 transmitter and 2 receivers are used in this project. The sensor is illustrated in Fig. 3.2

Figure 3.2. Ultrasonic sensor.

3.2.3 Analog Signal Generator

Analog signals are generated from the Teensy Microcontroller and fed to the ultrasonic transmitter. The signal is of 300 kHz with a 50% Duty cycle. A PWM pin on the Teensy is used to transmit analog signal to the ultrasonic transmitter. The signal is then up-converted to 12 V using a level shifter to allow the generation of strong enough sonar signal.

3.2.4 Operational Amplifier

The Operational Amplifier is used to amplify weak voltages obtained from the ultrasonic receivers to be able to properly read them on the Microcontroller's end. Without amplification, the signal will be too weak to process. In this project, a series of inverting amplifier circuits are built using the OP-AMP TL084IN from Texas Instruments. The details of the OP-AMP can be found in the datasheet [15]. The amplifiers are connected with analog filters to eliminate sensor drift and reduce low frequency noise in the signal. The Operational amplifier circuit is illustrated in Fig. 3.3

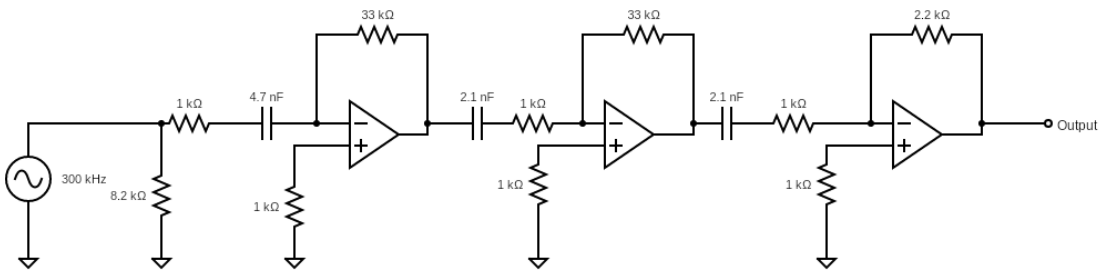


Figure 3.3. Operational Amplifier circuit.

As the used sonar sensor operates on a center voltage of $6V$, the final amplified signal is down-shifted using a voltage divider to be within the range of the Teensy microcontroller and the ADC input is protected using a Zener diode.

3.3 Experimentation Methodology

We are using a ceramic container setup where the ultrasonic transmitter and receivers are mounted to the ceramic container with hot glue so as to provide a vacuum seal between the surfaces. Ceramic is a good conducting medium for sound and hence

it should propagate the waves easily. Water is filled into this ceramic container and to maintain consistency we are also using a filter mount so as to introduce the materials into the water container from the same height and same area. The materials will be introduced into the water up to a certain level and then will be stopped being introduced and let the material settle in water. During this whole process, all the ultrasonic wave patterns will be recorded from the moment before the material gets introduced into the water until the material settles or stabilizes underwater. We have two receivers mounted on the container's surface and one transmitter mounted on the opposite side of the container. The receivers are mounted at 90 degrees angle towards each other so as to get the wave patterns from different point of views. The transmitter is mounted at 0 degrees angle from one of the receivers. A top view of the container is shown in Fig. 3.5

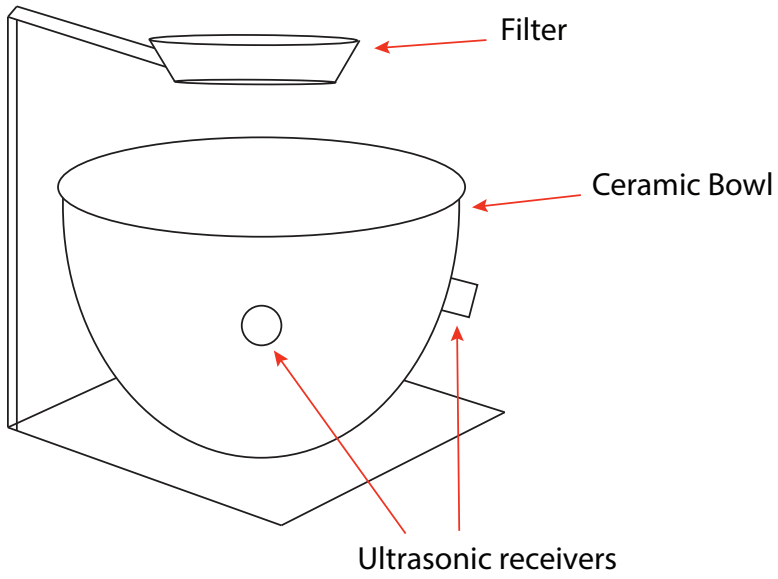


Figure 3.4. Setup for introducing materials into water.

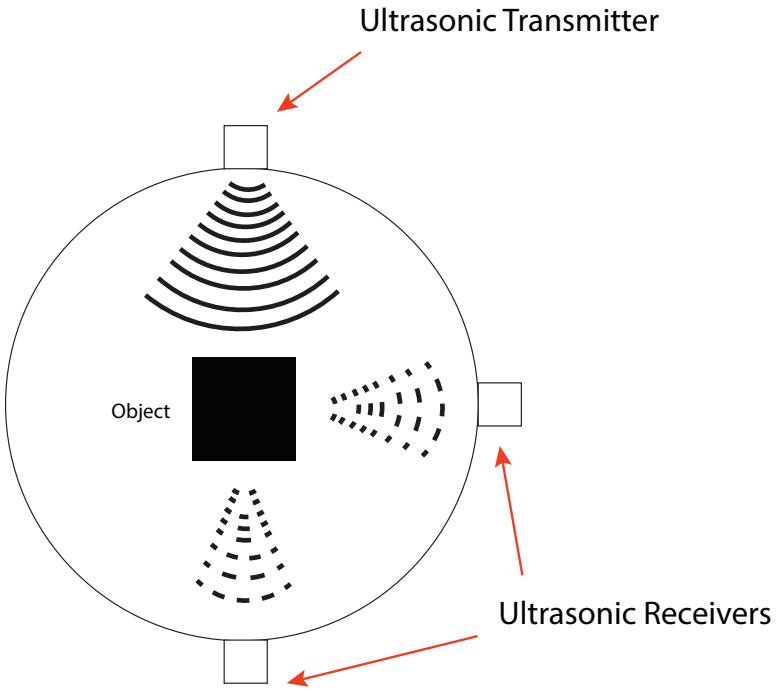


Figure 3.5. Top View of the Ceramic Container.

The goal of this setup is to be able to estimate back-scatter and diffuse reflections caused by the introduced material which should change the proportion of signal received by the two receivers, thus changing the signal amplitude, as well as introduce phase differences between the received signals in different frequency ranges.

All the ultrasonic sensors are connected to the Teensy Microcontroller which is generating the wave for the transmitter and reading signals from both receivers after analog to digital converting them synchronously. This data is being constantly sent to a laptop machine over USB connection. In the laptop, this data is being recorded at a rate of 2MBps[1] from the Teensy via USB data transfer.

[1]Megabytes per second

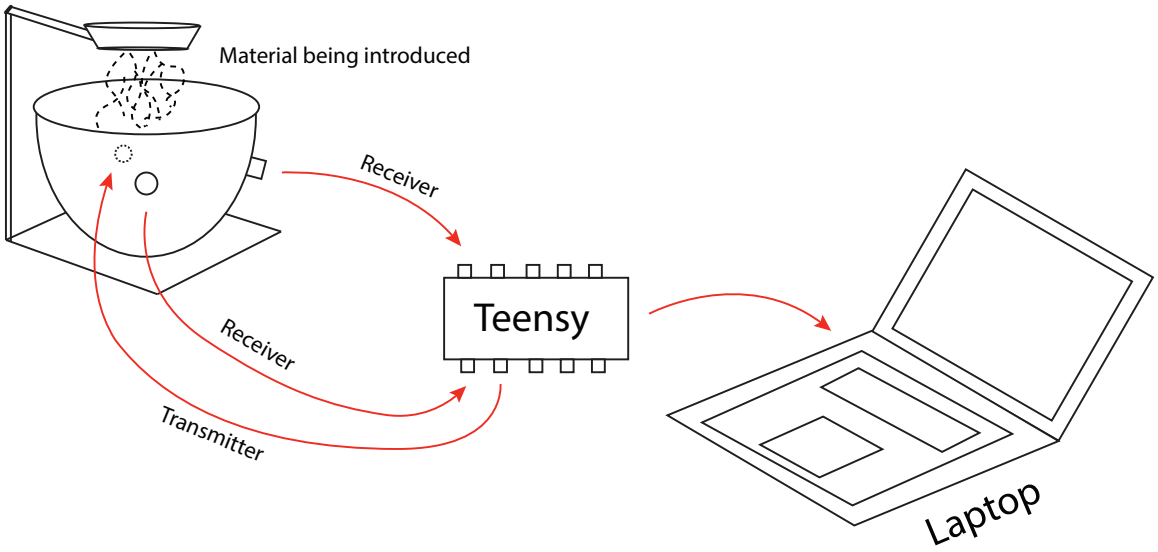


Figure 3.6. High level working of the system.

8 materials were used to record the data which are of varying properties in terms of solubility, granularity and density. These materials chosen were: Rock, Soap, Metal Wire Mesh, Milk semi-solids, Flour, Sand, Bread Crumbs and Water.

3.4 Data Acquisition

Since we are dealing with very high frequency data (300kHz), the Microcontroller has to be very fast in sampling the analog signal and sending it over to storage. Hence we are using the Teensy Microcontroller which can sample the Analog signal at a speed upto 20MHz. The ADC resolution of the ADC of Teensy can go upto 12-bit. A big advantage in Teensy is that it has two ADCs on-board, which enables us to record the signal from both the receivers simultaneously all using a single device. But reading the signal fast enough isn't the only problem, we also have to send it fast and efficiently to our main laptop so as to save the signal data on storage. Getting the signal data over USB serial line is a little tricky since some libraries have a bottleneck when trying to read raw data from USB Serial and have less maximum speed. To

do so efficiently, we will have to send the data in packets of single byte information. There is a method in Teensy library called,

```
Serial.write()
```

which sends a single byte chunk to the serial USB line or a custom size buffer with elements of single bytes each. So, since our single ADC reading from Teensy is 12-bit, we can wrap it with the other ADC reading from the other sensor together. We can do so by byte-shifting 16-bits in a 32 bit chunk and saving one ADC value in the lower 16 bits and the other 12-bit value in the upper 16 bits. Then we will send this 32-bit (4 Byte) chunk to our serial USB line for our laptop to read. On the laptop's end, we unpack both the ADC values by shifting the bits and read the values using a Python program.

3.5 Data pre-processing

After we unpack the individual ADC values, we need to save them separately into a file for further processing. We save these values into a **csv** file. Now, while we were collecting data from our sensors, some values got corrupted (values greater than 600000 while the maximum possible value given by an ADC is 4095) when recording the data at such a fast speed from Teensy to laptop. Hence, to deal with this, we will replace these values with the mean of our signal. Now we have our complete raw signal without corrupted values.

As discussed in Section 2.3.2, we convert our raw signal from time domain to frequency domain using Fast Fourier Transform as shown in Equation 2.8 to generate insights from the data. We implement a sliding window approach to slide through the raw signal for a specific window size(ex. 100 milliseconds) with a slide size (ex. 20 milliseconds) over the whole signal and compute the FFTs of those respective

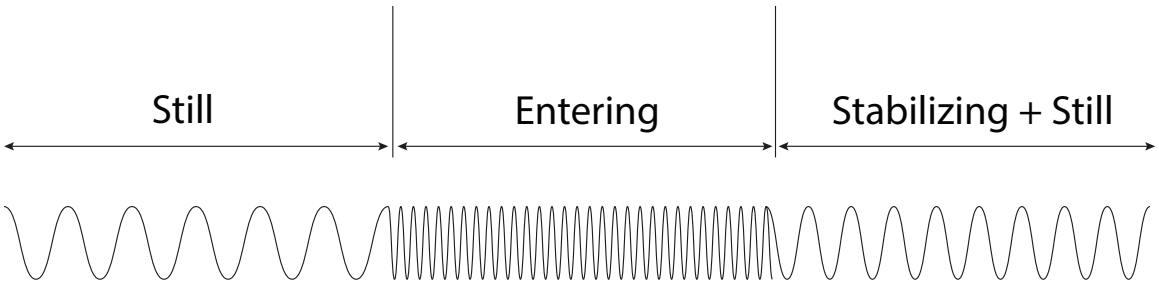


Figure 3.7. States of a raw signal over time.

windows as we slide through the signal. Now we have a FFT data information over the whole signal and we can use this to solve our classification tasks.

3.6 Detection of object presence states

Now we have all the raw ultrasonic data for the 8 materials when they were being entered into water. Since it is time-series data we can segment our data into isolated events to gain more information. This current signal can be divided into three main events: Still state (No object has yet entered the water), Entering state (Object being introduced into the water) and Stabilizing state (object was stopped being added to water and now the object will settle underwater). In actuality, the stabilizing state will have two sub-states: stabilizing and then being still. So, the original still state and the stabilizing still state can be harder to distinguish. These are visualized in Fig. 3.7. Hence, once we identify the state change of a signal, then we can say that an object has entered the water.

3.6.1 Approach

We can make use of the FFTs of time windows of the raw signal and extract information from them to determine if we are able to isolate these events based on the information. Since we have a lot of frequency and amplitude components after

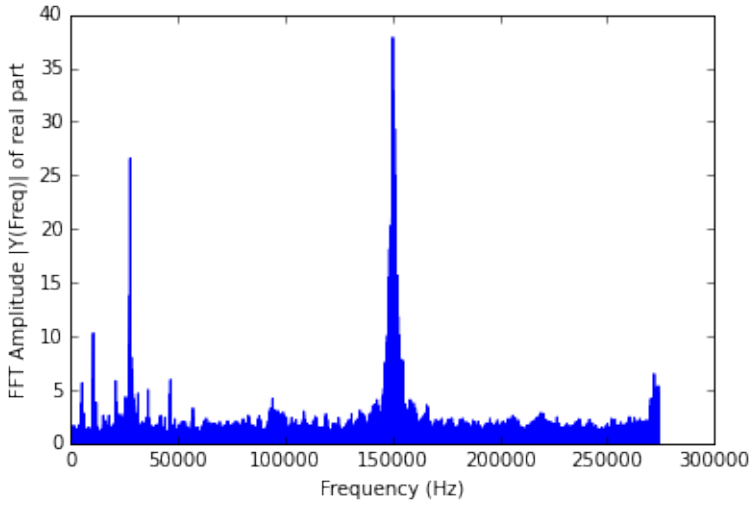


Figure 3.8. FFT of a time window of the signal.

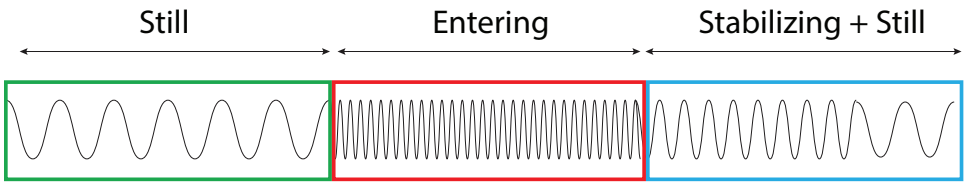
performing FFT, there are a lot of huge feature vectors to deal with since we get a lot of frequency responses by doing FFT. To deal with this problem, we utilize Principal Component Analysis on the FFT power spectrum to reduce the number of dimensions and help us extract more meaningful features which basically represent significant frequency components in the data set. These components can reflect, for example, natural frequencies of surface waves for the given container size, the base frequency of the sonar, or frequency response properties of the used materials.. The PCA features obtained result in a significantly compressed feature vector and can thus be used in simple clustering algorithms to combine similar events in the same cluster. This will help us identify the different states of the signal based on the similarity and dis-similarity of PCA features of those states with each other. Some common clustering algorithms such as K-Means clustering and Agglomerative clustering are tried on the PCA components of the FFT values to determine how effectively we can isolate the events. We also try clustering on different variances contained in the PCA components and show our results.

3.6.2 Clustering v/s Classification of states

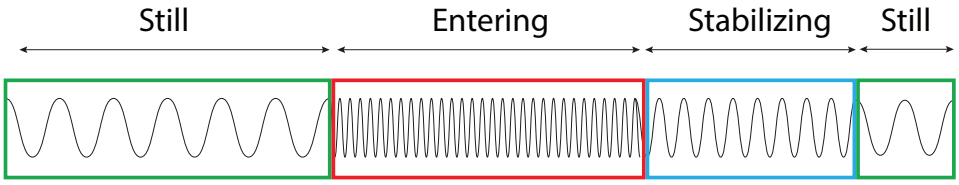
We prefer clustering over classification for segmenting the time stages because we are not sure if there are only 3 discrete stages of transition involved. Moreover, while the initial point in time at which the object is entered in the water might be easy to label, any transition from a stabilizing to a still state would generally be very difficult to label and thus labeling might be inherently incomplete, causing issues with training classification algorithms. There can be multiple stages and hence clustering can give us a better result in un-supervised classification as compared to classification with a fixed number of classes. As an initial assumption, we have considered 3 stages of transitions and we will learn about the stages from the results that we get and can understand if our assumption was correct about the stages. An example of comparison for classification and clustering based approach is illustrated in Fig. 3.9. If we choose classification, then we will always have specific classes and might not be able to explore whether there could be multiple states or if a pattern could repeat after some duration. Hence, clustering can let us understand the state transitions with more granularity even if we are not able to correctly label them. Moreover, data for cluster-based segmentation will be significantly easier to acquire when installing the system..

3.6.3 Outliers and temporal filtering

The main issues that we encounter in this approach is the uneven clustering of the PCA components. Since the data is time-series, some events can be incorrectly classified based on the PCA components, but they were supposed to be classified in an earlier state. For example, a FFT time window got classified as “entering”, but it was actually in “still” state. This problem is illustrated in Fig. 3.10 But, if we think about it, the simple principle of these time states is that they will always be in



(a) Classification when 3 states are assumed



(b) Clustering when 3 states are assumed

Figure 3.9. Comparison of clustering v/s classification.

sequence. So, if an entering state is mis-classified to be in the beginning of the still state, we know that's not possible because entering will always come after "still" state. Hence, we can leverage this to see the distribution and fix possible classification issues by looking which cluster is assigned the most in the beginning of the time sequence. Once we find that class, we know that no other state could be in the beginning "still" state and are misclassified. Hence, we can relabel these classes according to the most weightage they have in the specific region of the time window. For example, if most points of cluster "A" is in the beginning of the time window, then we know they all are in still state and other points of "A" which are clustered in the later time stages are misclassification. Thus we can relabel all points to class "A" from the beginning of the of the time window until the last point of class "A" where they are not far apart from the distribution of "A". We will do the same for other time states.

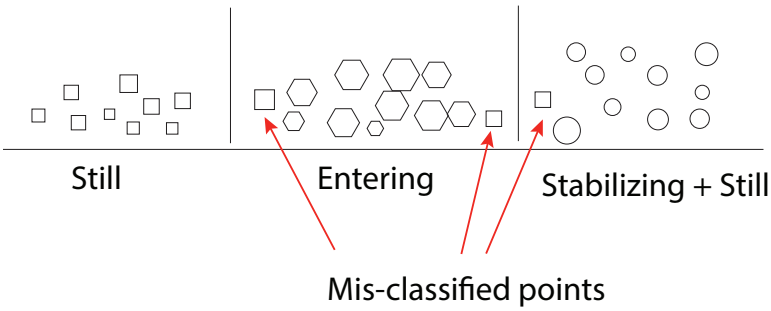


Figure 3.10. Mis-classified points for time states.

3.7 Classification of materials

The classification of materials underwater is a continuation of the previous problem. To solve this problem, we will first segment the states for all signals of all classes. Now, as soon as we have separate data for each state (i.e. still, entering or stabilizing), we will use the data to train separate neural networks for separate states, for example, we use only "entering" state data of all classes to train our neural network and see the results. We do the same for the other states and then compare the results on which state is more significant in identifying the materials.

3.7.1 Input dataset for Neural Network

The vector that we design for feeding the Neural Network contains the actual raw signal of both the ultrasonic sensors and the PCA components of the FFT response for the raw signal window. The reason for including the raw data is the assumption that in order to determine material characteristics, more detailed information regarding the absorption and reflection of the actual 300KHz signal might be useful and it is important to be able to utilize the differences in amplitude, phase, and shape of the raw signal in addition to the larger frequency response information provided by the FFT spectrum. This vector is made for all time windows for only a specific state (ex. entering). The vector is illustrated in Fig. 3.11. So, each time

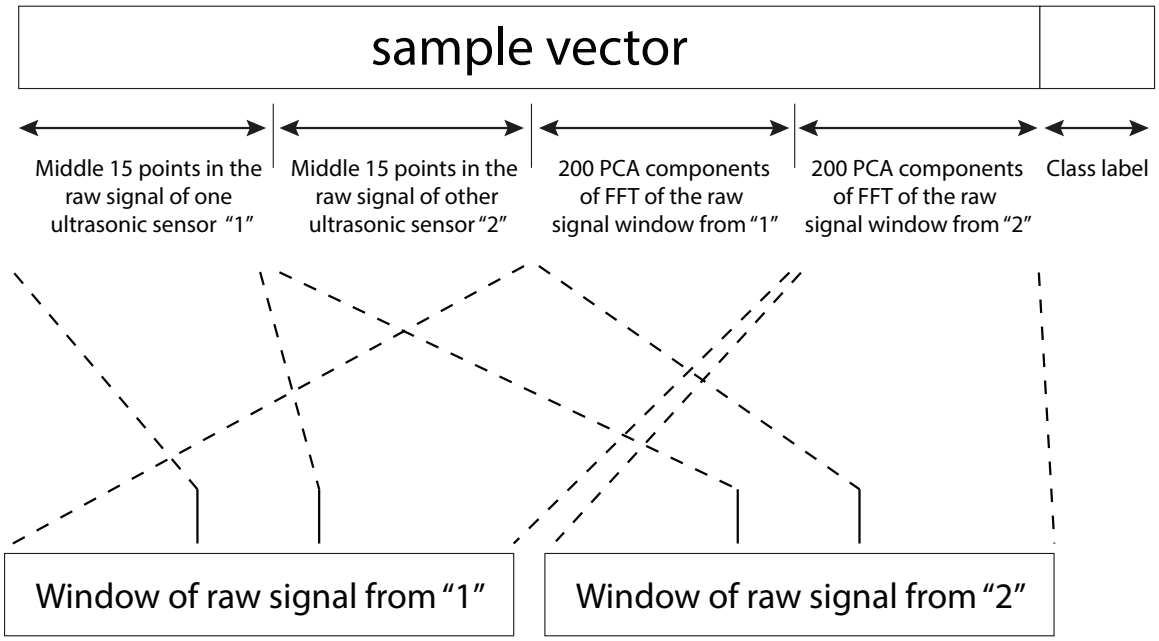


Figure 3.11. Sample vector for Neural Network.

window will have one sample vector and we will have 431 points in the sample vector (15 points for raw signal from sensor '1' + 15 points for raw signal from sensor '2' + 200 points of PCA components from FFT response of time window of sensor '1' + 200 points of PCA components from FFT response of time window of sensor '2' + 1 point of class label of the respective class). All these sample vectors for time windows in "Entering state" for all classes will be combined in a single dataset and the last column of class labels will be separated as 'target' values to train for. So our dataset will be of the shape $(\sum_{i=1}^{8} N_i, 430)$, where N_i is the number of time windows in a class i and our target labels array will be of size $\sum_{i=1}^{8} N_i$.

3.7.2 Neural Network Architecture

We build a Dense Neural Network with 5 dense layers having activation functions as tanh, relu, sigmoid, and softmax. We have 2 dropout layers with 20% dropout each to enforce robustness of the solution and reduce the risk of overfitting. The loss

function used is Sparse Categorical Cross-entropy and optimizer is Adam with a learning rate of 0.001. The neural network is illustrated in Fig. 3.12. We have used dropout to avoid over-fitting and since we are performing multi-class classification, we used categorical cross-entropy loss.

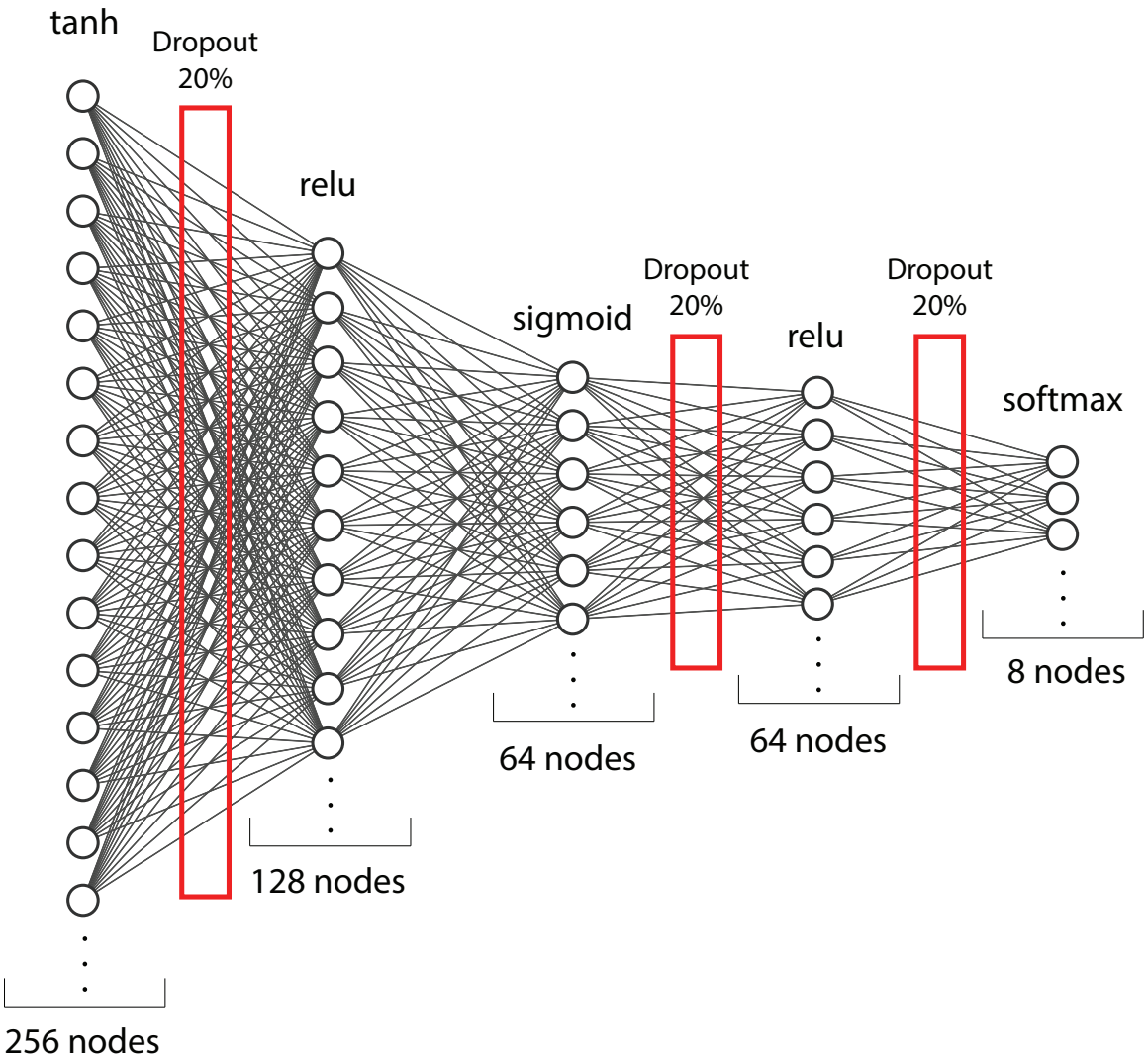


Figure 3.12. Neural Network Architecture.

CHAPTER 4

RESULTS

4.1 Overview

To validate the working of our object detection solution, we have created a test from ground truth. We marked the points when the states changed (from "still" to "entering" and from "entering" to "stabilizing") while recording the data from the ultrasonic sensors. This helped us validate the performance of our clustering algorithms.

4.2 Detection of object presence states

We classified the states into when the object was introduced in water and when the object was stopped being introduced into water. We assumed that there were three states: 'Still', 'Entering' and 'Stabilizing + Still'. We performed PCA on the obtained FFT responses to reduce the number of dimensions for clustering. We tested with different variances in PCA and compared their results. We tested with 85%, and 95% variance in PCA, resulting in the sue of 170 and 320 PCA components, respectively. We performed K-Means clustering and all four linkage types of Agglomerative clustering on our data. The results obtained are illustrated in the next sections.

4.2.1 PCA 85% Variance

In this case, we can observe from the confusion matrix Fig. 4.1 that a lot of points are misclassified as 'still' when they were actually pre-labeled as 'stabilizing'. So, here comes the problem of classification, since we do not definitely know if the

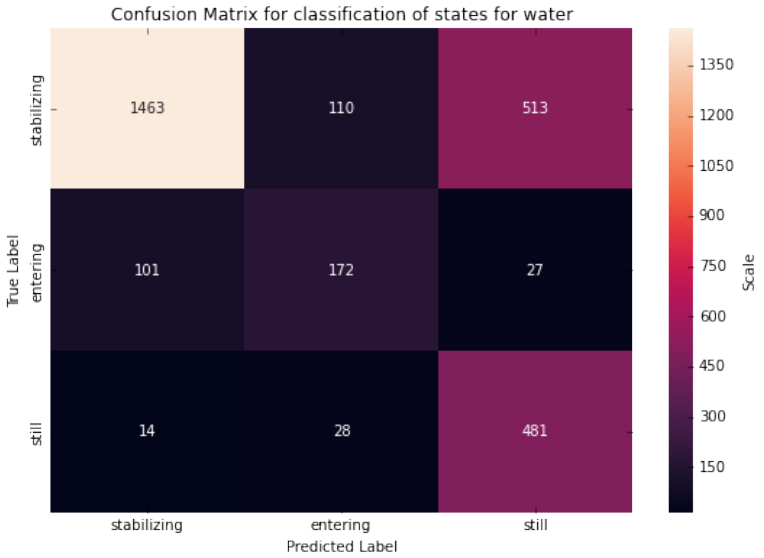


Figure 4.1. Confusion Matrix for K-Means clustering for 85% PCA.

states are actually well-separated or there are definitely 3 states, our underlying assumption that there are three states might not be correct. So after plotting the cluster assignments for the time series, we can observe in Fig. 4.2 that there are many points classified as 'entering' when they should have been included in 'stabilizing' and there are also some other points which are misclassified. Now, we apply the 'relabeling technique' as mentioned in Section 3.6.3. After relabeling, we get clean results of contiguous clusters as we know that the states 'still' and 'entering' will be contiguous. But, we cannot relabel stabilizing since the states might actually be trying to transition from 'stabilizing' to 'still' and vice-versa and hence the number of clusters should be increased to accommodate for multiple stages that can occur. The relabeled states are illustrated in Fig. 4.3.

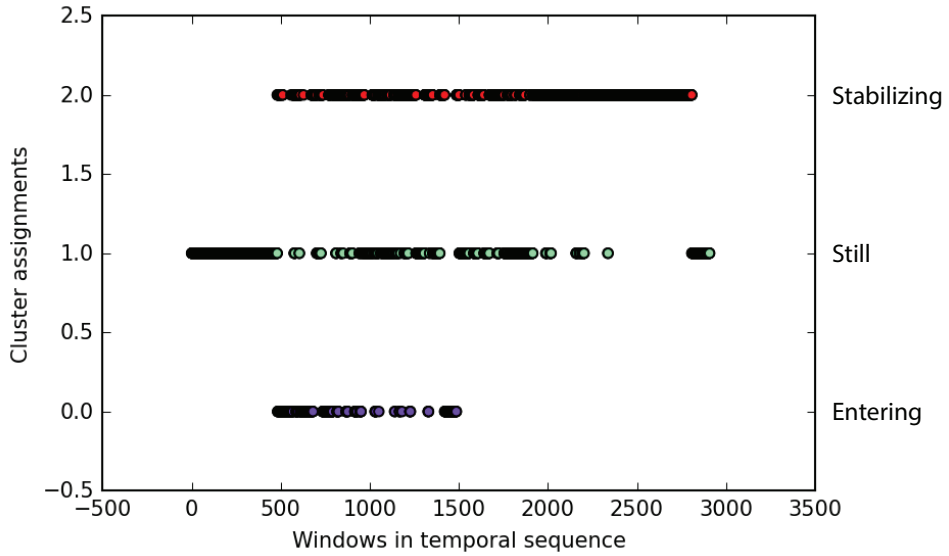


Figure 4.2. Cluster assignments of time windows for water.

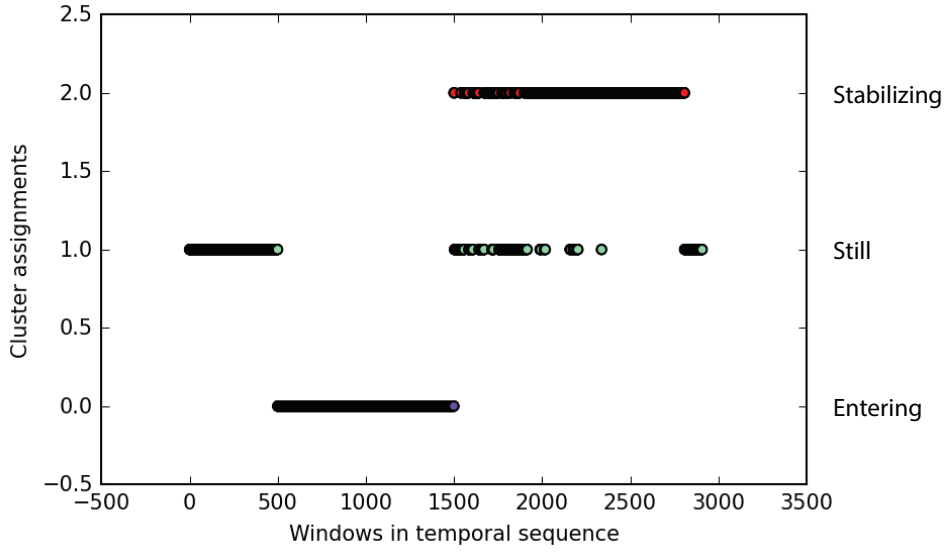


Figure 4.3. Relabeled cluster assignments of time windows for water.

Hence, to account for the new unknown state, we can assign the transitioning and unlabeled state as 'still_after' and the original 'still' state becomes 'still_before'. These new states are illustrated in Fig. 4.4.

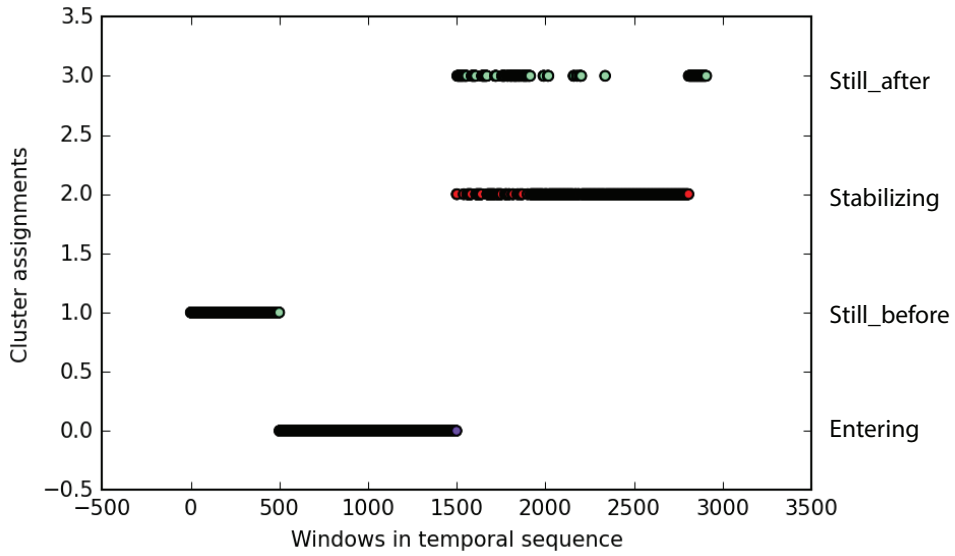


Figure 4.4. After relabeling and creating new state cluster for water.

Further results from other clustering algorithms for 85% and 95% variances are depicted in the following figures, this same relabeling technique can be applied in other cases to get accurate cluster representations. We can see from this confusion matrix in Fig. 4.5 that Agglomerative clustering with ward linkage is comparable to KMeans clustering and has the most accurate results among other linkage functions. It can be fixed similarly using the 'relabeling' technique as mentioned in previous case.

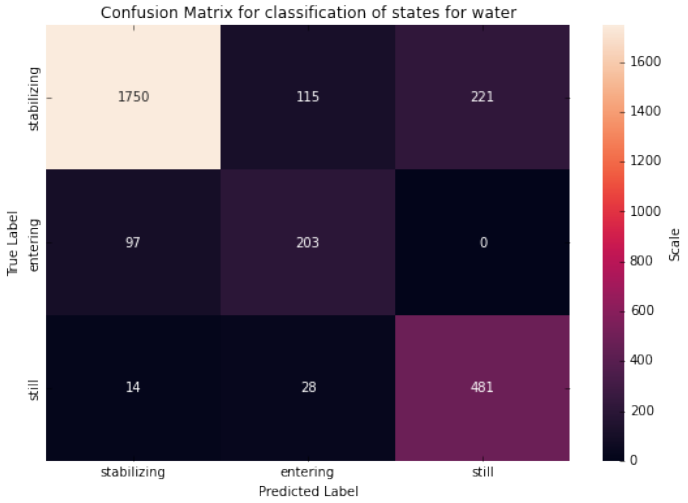


Figure 4.5. Confusion Matrix for Agglomerative clustering: Ward for 85% PCA.

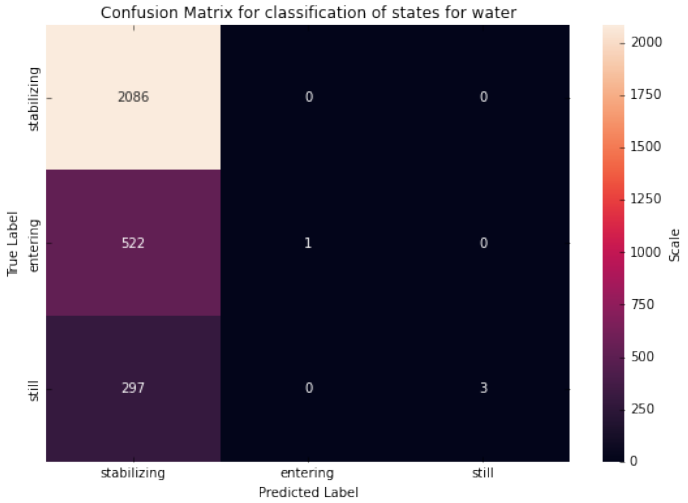


Figure 4.6. Confusion Matrix for Agglomerative clustering: Single for 85% PCA.

The single linkage for Agglomerative clustering does not work well as shown in Fig. 4.6 and the accuracy for classification is worse.

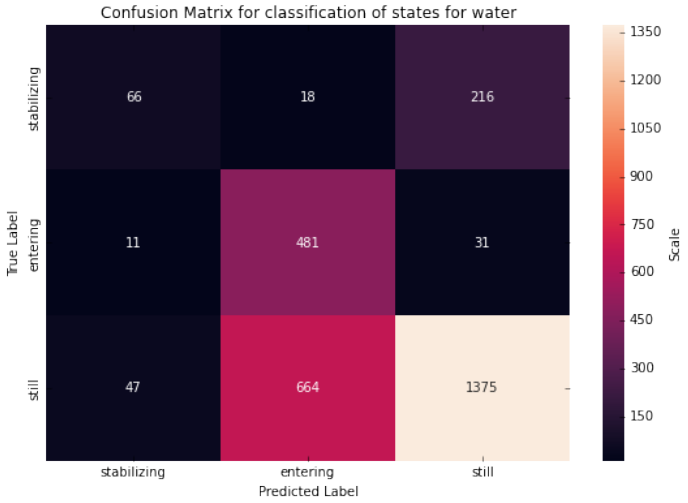


Figure 4.7. Confusion Matrix for Agglomerative clustering: Complete for 85% PCA.

Similarly, the complete and average linkages in Fig. 4.7 and Fig. 4.8 for Agglomerative clustering also do not work well and the accuracy is worse. This indicates that these linkage criteria do not correctly capture the similarity relation of the object presence stages in our problem and are thus not appropriate for the stage classification problem.

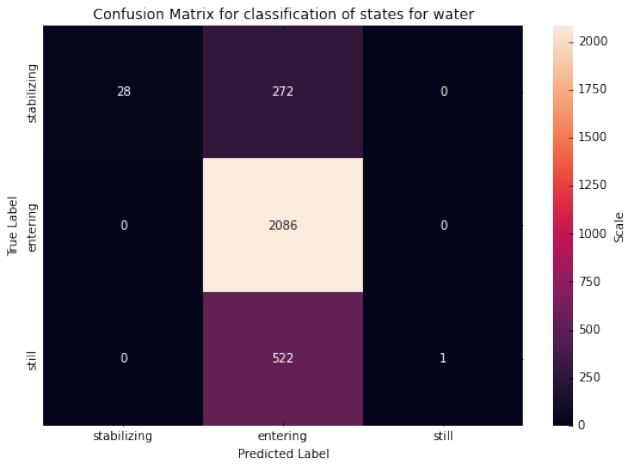


Figure 4.8. Confusion Matrix for Agglomerative clustering: Average for 85% PCA.

4.2.2 PCA 95% Variance

The results for PCA with 95% Variance are similar to those of 85% Variance and hence the same techniques can be applied. The only drawback was that since we are preserving more variance in this case, we have more data in 95% variance and thus, it takes a little longer to cluster.

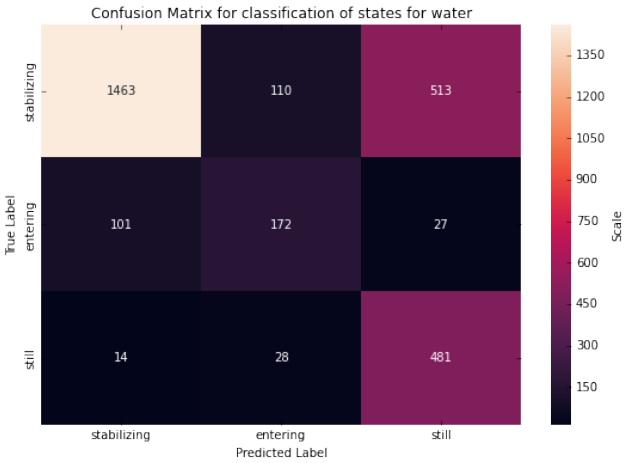


Figure 4.9. Confusion Matrix for K-Means clustering for 95% PCA.

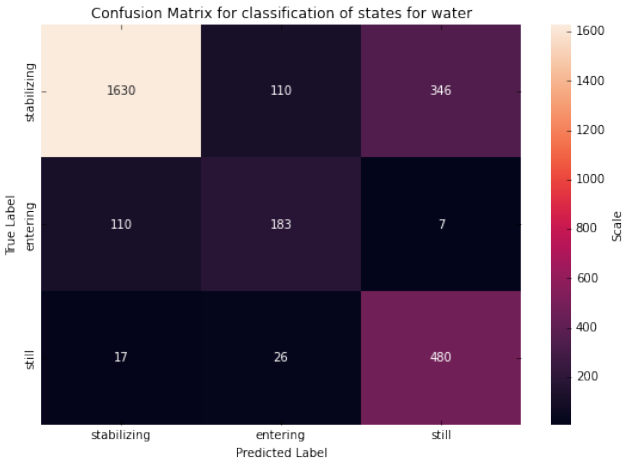


Figure 4.10. Confusion Matrix for Agglomerative clustering: Ward for 95% PCA.

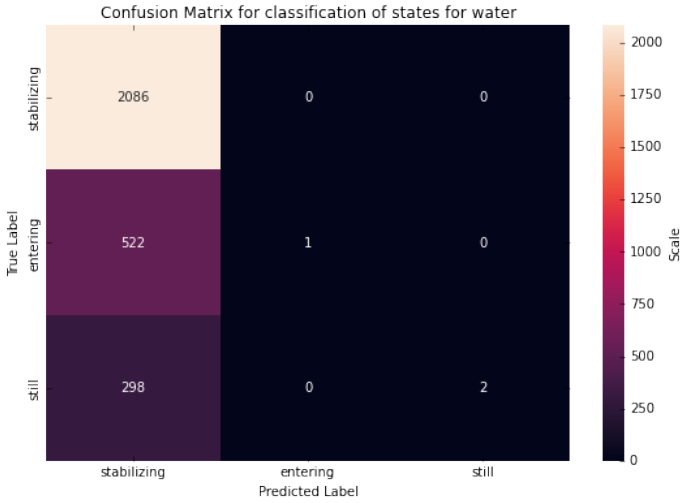


Figure 4.11. Confusion Matrix for Agglomerative clustering: Single for 95% PCA.

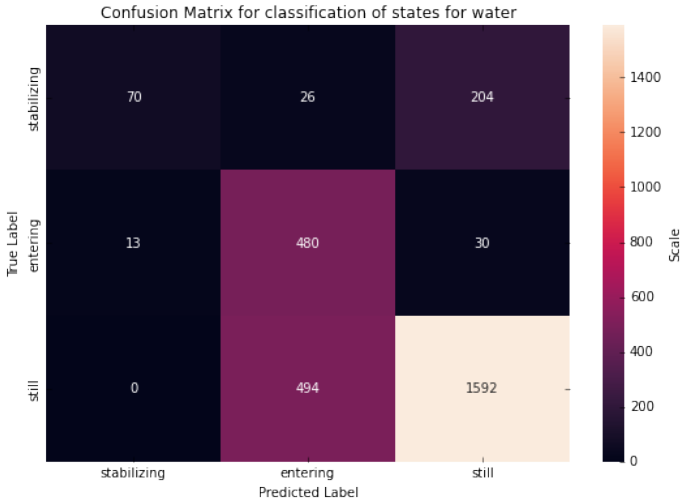


Figure 4.12. Confusion Matrix for Agglomerative clustering: Complete for 95% PCA.

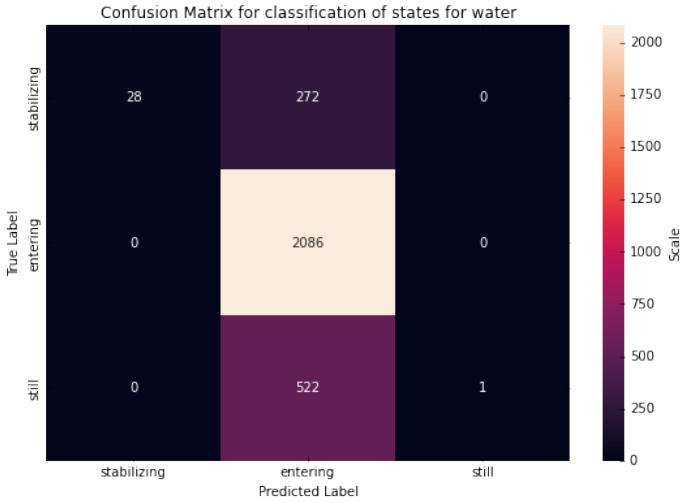


Figure 4.13. Confusion Matrix for Agglomerative clustering: Average for 95% PCA.

4.3 Classification of materials

After segmenting the clusters in the previous step as mentioned in Section 3.7, we built vector inputs for our designed Neural Network by taking data from the 'Entering' state and 'Still_after' state. For this, PCA components for the data in this state were computed and the feature vectors of length 430 indicated previously were constructed for all time windows, combining raw and PCA components of the FFT spectrum. The breakdown of our sample vector for the Neural Network was as follows:

- 15 points from the middle of raw signal for the current time window for sensor '1'.
- 15 points from the middle of raw signal for the current time window for sensor '2'.
- 200 PCA components from FFT response of the time window for sensor '1'.
- 200 PCA components from FFT response of the time window for sensor '2'.

Using this data, an 8-class classifier Neural Network was trained and its performance was evaluated.

4.3.1 Entering State

Fig. 4.14 shows the training and validation accuracy and loss function, illustrating the strong performance of the classifier, converging to a validation accuracy of 95.51%. While the loss graphs indicate some overfitting which might be addressable with a larger data set, the test accuracy still illustrates the soundness of the approach.

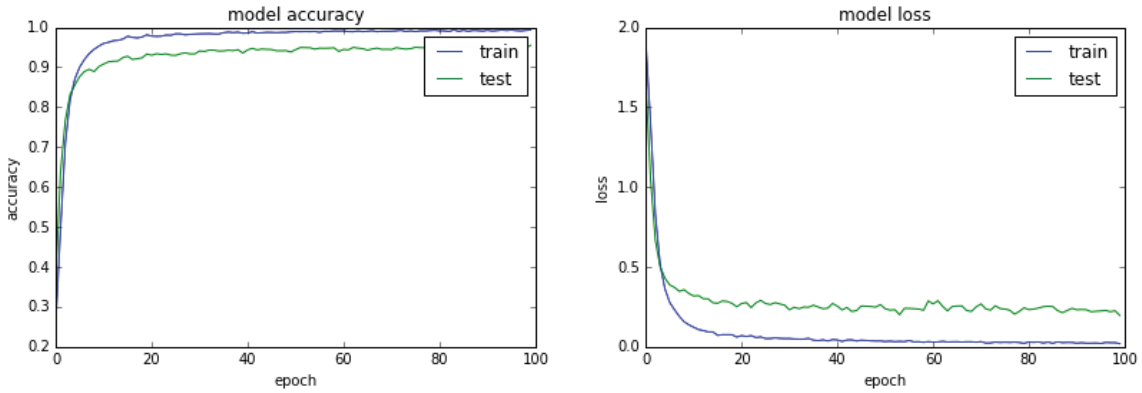


Figure 4.14. Model accuracy and loss graphs for Entering state.

Classification results for 8 classes namely: water, rock, soap bar, metal mesh, bad milk solids, flour, bread crumbs and sand are illustrated in a confusion matrix in Fig. 4.15, illustrating a very high accuracy for all the materials except for rock which also had the smallest number of data samples as it exhibited a very short entering phase.

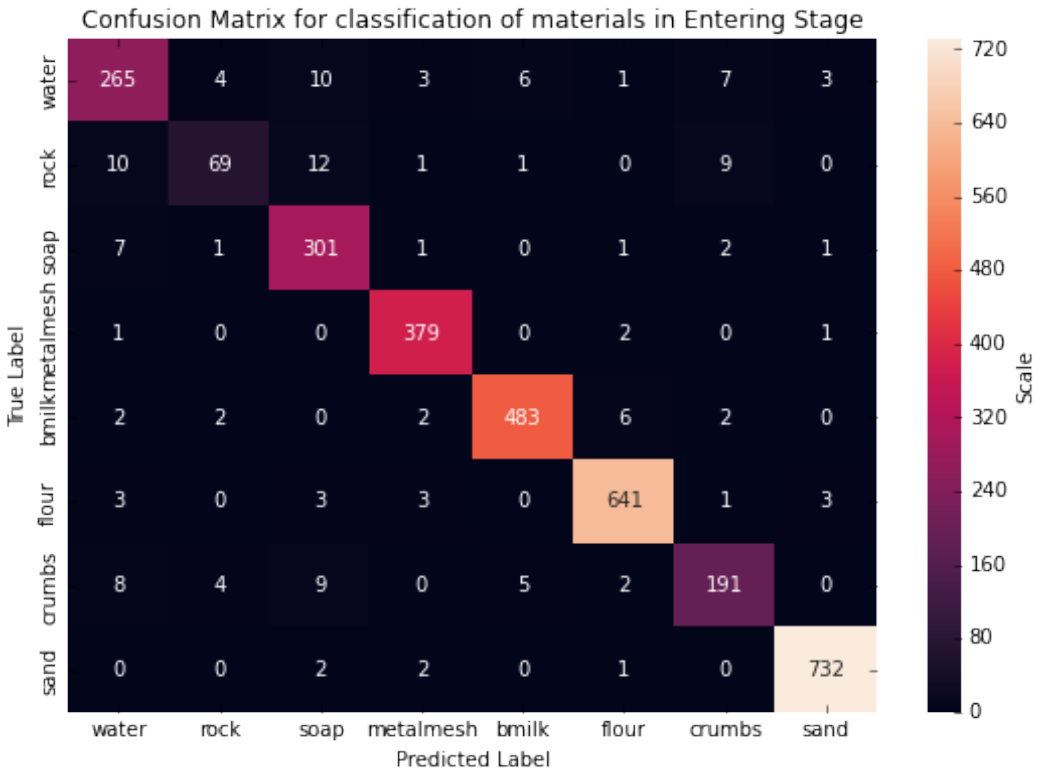


Figure 4.15. Confusion Matrix for classification of materials in Entering state.

4.3.2 Still_after state

The Still_after state is the 'still' equilibrium state which the system transitions to when it has stopped stabilizing as mentioned in Section 4.2.1. The same procedure as in the previous Section 4.3.1 is performed and the accuracy and loss graphs are illustrated in Fig. 4.16. The results for this classification in this state are mostly similar to the 'Entering' state and the accuracy comes out to be 95.31% which indicates that our model again performs good in this task. The classification results in the form of a confusion matrix are illustrated in Fig. 4.17. We had good classification for all the classes here because we had approximately the same number of samples for

all classes. The loss graph again might indicate some overfitting, but still indicates good test performance which might be further improvable with a larger dataset.

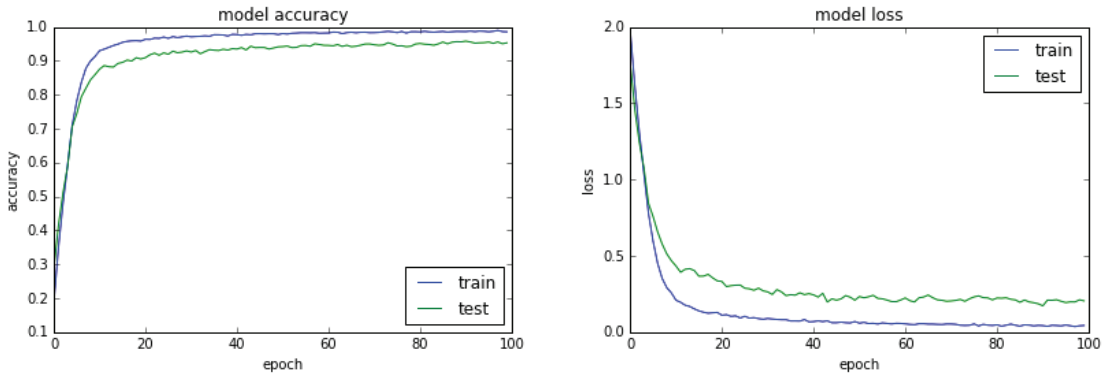


Figure 4.16. Model accuracy and loss graphs for Still_after state.

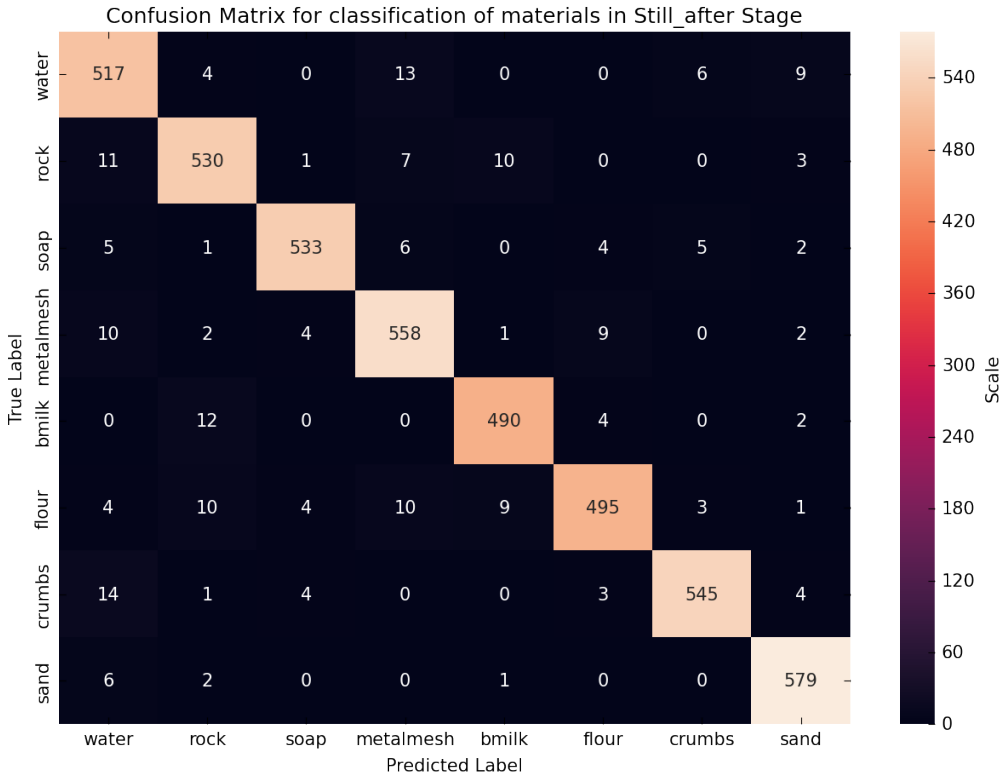


Figure 4.17. Confusion Matrix for classification of materials in Still_after state.

CHAPTER 5

CONCLUSIONS

Detection and characterization of soluble, diffuse, and solid objects in water has a number of important applications in water management as well as in incontinence monitoring for health applications in the context of a smart toilet. Especially for the latter, a non-intrusive solution is required that does not inconvenience a user, is easy to maintain, and does not invade a person's privacy. Use of ultrasonic sensors is one possible way to achieve such a non-intrusive system. In this thesis, we aimed to explore the potential for classifying materials introduced underwater based on raw ultrasonic data. We built an experiment setup for collecting data for 8 different materials namely water, rock, soap bar, sand, metal mesh, bad milk solids, flour, and bread crumbs when they were submerged underwater in a ceramic container. We built a high speed data acquisition device using a Teensy microcontroller to record the ultrasonic data on our laptop in the backend. The same microcontroller was used as a wave generator for generating 300kHz signals for ultrasonic transmitters. The whole interface of transmitters and receivers was built around a Teensy, implementing a level shifter and operational amplifier-based amplification and filtering setup that allowed the sonar sensors to operate at a 12V range with a 6V center voltage to optimize their performance while maintaining range and resolution of the analog to digital conversion.

After data collection, we pre-processed data for our major tasks. The two major tasks were: Segmenting discrete states when the object was introduced in the water and classifying the object properties when inserted underwater in one of the 8 classes

based on an individual state. In the first task, we calculated FFT components for time windows of the raw signal for an object. Then we extracted PCA components from the FFT data and performed clustering over the PCA components to divide the signal into different states of the object presence. Experiments showed that with additional temporal filtering this approach provided high quality segmentation results with very few outliers. After segmenting the states, we used individual state to classify the object properties. A Neural Network model was built to classify the objects based on the chose state's PCA components and a fraction of raw signal. Building separate classifiers for each state is here aimed at increasing accuracy by removing variation in the signal due to other effects present in the state such as surface variance caused by wave formation.

Results obtained from the classification in the entering and the still_after state both had good accuracies up to 95% and serves as a proof of concept that ultrasonic waves can be used to classify object's characteristics and can have potential applications in the field of health monitoring through its application to non-intrusive incontinence monitoring.

5.1 Future Work

This thesis work can be extended to use more than two ultrasonic receivers and transmitters to measure and record data from various angles. With the use of multiple sensors, more physical characteristics can potentially be extracted from the object. Also, various different ultrasonic sensors could be used with higher frequencies going up in the Mega Hertz range to get a stronger signal and possibly more information. After studying it on more types of objects with different characteristics, a finalized attachment could be fabricated to finally begin testing on toilet systems and experimenting to analyze the possibility of application in healthcare.

REFERENCES

[1] M. G. Silk, "Defect detection and sizing in metals using ultrasound," *International Metals Reviews*, vol. 27, no. 1, pp. 28–50, 1982. [Online]. Available: https://doi.org/10.1179/imr.1982.27.1.28

[2] S.-M. Park, D. D. Won, B. J. Lee, D. Escobedo, A. Esteva, A. Aalipour, T. J. Ge, J. H. Kim, S. Suh, E. H. Choi, A. X. Lozano, C. Yao, S. Bodapati, F. B. Achterberg, J. Kim, H. Park, Y. Choi, W. J. Kim, J. H. Yu, A. M. Bhatt, J. K. Lee, R. Spitler, S. X. Wang, and S. S. Gambhir, "A mountable toilet system for personalized health monitoring via the analysis of excreta," *Nat. Biomed. Eng.*, vol. 4, no. 6, pp. 624–635, June 2020.

[3] M. Ecemis and P. Gaudiano, "Object recognition with ultrasonic sensors," in *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No.99EX375)*, 1999, pp. 250–255.

[4] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54 776–54 788, 2020.

[5] K. Petty, "Smart toilet uses artificial intelligence to monitor bowel health," May 2021. [Online]. Available: https://today.duke.edu/2021/05/smart-toilet-uses-artificial-intelligence-monitor-bowel-health

[6] T. J. Ge, C. T. Chan, B. J. Lee, J. C. Liao, and S.-m. Park, "Smart toilets for monitoring covid-19 surges: Passive diagnostics and public health," Mar 2022. [Online]. Available: https://www.nature.com/articles/s41746-022-00582-0

[7] D.-H. Kim, S.-R. Lee, and C.-Y. Lee, "Material classification using reflected signal of ultrasonic sensor," *J. Control Autom. Syst. Eng.*, vol. 12, no. 6, pp. 580–584, June 2006.

[8] "Analog-to-digital converter," *Wikipedia*, May 2022. [Online]. Available: https://en.wikipedia.org/wiki/Analog-to-digital_converter

[9] G. Stanley, "Sampling, aliasing, and analog anti-alias filtering." [Online]. Available: https://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Filtering/Aliasing/aliasing.htm

[10] Mathuranathan, "Interpret fft, complex dft, frequency bins & fftshift," Oct 2021. [Online]. Available: https://www.gaussianwaves.com/2015/11/interpreting-fft-results-complex-dft-frequency-bins-and-fftshift/

[11] "Cos 324: Introduction to machine learning." [Online]. Available: https://www.cs.princeton.edu/courses/archive/fall18/cos324/

[12] Jovana and Kassambara, "Agglomerative hierarchical clustering," Oct 2018. [Online]. Available: https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/

[13] D. Asanka, "Implement artificial neural networks (anns) in sql server," May 2020. [Online]. Available: https://www.sqlshack.com/implement-artificial-neural-networks-anns-in-sql-server/

[14] "Product specification datasheet for murata ma300d-1 ultrasonic sensor." [Online]. Available: https://www.murata.com/products/productdata/8797589241886/MASPHFE.pdf?1508297407000

[15] "Datasheet for tl08xx fet-input operational amplifier." [Online]. Available: https://www.ti.com/lit/ds/symlink/tl084.pdf

BIOGRAPHICAL STATEMENT

Mehul Vishal Sadh was born in Mumbai, India, in 1997. He graduated from University of Mumbai in 2020 with a degree in Bachelors of Engineering in Computer Engineering and he received his Masters of Science in Computer Science degree from The University of Texas at Arlington in 2022. He worked as a Graduate Teaching Assistant from January 2022 - May 2022 at The University of Texas at Arlington in the Computer Science and Engineering Department. His current research interests are in Machine Learning and Artificial Intelligence.