Indoor Location Sensing Using Smartphone Acoustic System

Impact of interferences on the performance of acoustic indoor location system

Filip Biliński[1]

Supervisor: Qun Song[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Filip Biliński
Final project course: CSE3000 Research Project
Thesis committee: Qun Song, Jorge Martinez Castaneda

Abstract

Indoor localization is an actively researched field due to there not being a universal solution found yet. Applications of such systems include but are not limited to indoor wayfinding and automated tour guides. In previous years multiple solutions were proposed. This work looks into the performance of an indoor location sensing system in the presence of background music and tries to improve the accuracy in such a scenario. To achieve that a denoising autoencoder is proposed as a preprocessing step aiming to remove the noise from the fingerprints used for localization. In the end, it is shown that the use of such a technique introduces a tradeoff between an accuracy drop in quiet environments but an accuracy increase in environments with music.

1 Introduction

Localization is a service used in many applications, ranging from location-targeted content to indoor navigation. The applications that make use of localization now mostly use it for positioning outdoors or in a setting where high accuracy is not needed. These types of applications already have a universal solution that provides satisfactory accuracy with the use of GPS or Cellular information.

However high-accuracy indoor location sensing does not have such a solution developed yet. The topic of indoor location sensing is relevant for the automatization of a wide range of tasks therefore developing such a solution is beneficial. This work focuses on recent developments with regard to acoustic indoor location sensing.

This introduction section will describe the motivation for the research. Motivation is included in subsection 1.1 followed by the statement of the research goal in subsection 1.2. Then subsection 1.3 will describe the structure of this report.

1.1 Motivation

Using GPS for indoor localization comes with multiple challenges[2]. In summary, GPS signals are suppressed by walls which drastically decreases the accuracy of GPS positioning making it unreliable for indoor room recognition.

However indoor location sensing is important in the automatization of many different tasks. Therefore finding a solution that would serve as a replacement for GPS positioning would unlock potential new automatizations or increase the reliability of systems that do use GPS.

Examples of possible use cases for indoor location sensing are indoor way-finding, hospital patient localization, automated tour guides, and smart building automatization[9]. It would also be possible in some cases to replace SLAM[1]-like algorithms by such a system for robotic solutions.

1.2 Problem statement

This work focuses on an acoustic localization system similar to the one described in [9]. The focus is put on the impact of interferences on the accuracy of such a system and tries to overcome some of the limitations that the interferences create.

The main research question for this work is: Can the robustness of the system against music containing environment be improved? To answer this research question smaller subquestions can be defined as follows:

1. How is the system affected by the presence of music in the environment?
2. Can deep learning methods be used to improve robustness against music in the environment?

1.3 Structure

To answer the stated research question and report the findings in an accessible way this work uses the following structure. First the upcoming section 2 explains background information on indoor acoustic sensing and presents related work in the field. Next section 3 shows some intuition behind why the proposed location sensing model work. Following section 4 explains the method of answering the stated research question followed by a section 5 on how was the system used for experiments implemented. Setup of the performed experiments as well as their results can be found in section 6. Next section 7 will talk about responsible research. And finally the last two sections section 8 and section 9 will contain a discussion on the research findings as well as conclusion and future work.

2 Background & Related work

This section contains an overview of background information needed to understand the work in subsection 2.1. Following the background subsection 2.2 presents the related work in the field of acoustic localization sensing including an overview of already studied impacts of various interferences on different types of systems

2.1 Background

As stated before indoor location sensing has multiple potential use-cases. Since the problem has multiple applications if solved, it has been well-studied over the past years. Multiple approaches were proposed for indoor localization which use a wide range of different sensors to perform the task. The data used for localization by different systems include WIFI data, cellular data, acoustic data, camera data, geomagnetism data, and atmospheric pressure data. It is also worth mentioning that some systems combine multiple types of data to increase localization accuracy. The solutions can also be divided into infrastructure-dependant and infrastructure-free. As the name suggests infrastructure-dependant systems require the use of additional infrastructure inside the building to perform localization. An example of an infrastructure-dependant system would be any system that makes use of WIFI data like for example, the one described in [3].

Multiple infrastructure-free systems were also already proposed however this work focuses on only audio-sensing systems. These systems rely only on the built-in smartphone audio system to perform the localization. Such systems were also proposed with multiple different approaches that can be

[1] SLAM - Simultaneous localization and mapping

divided into categories based on some properties of the systems:

First of all, we can classify the systems based on the type of audio sensing used by each one of them. Then we can distinguish two different approaches:

- Passive sensing - these types of systems only make use of audio recordings to perform the localization. However, these systems perform worse in the presence of background noises and raise privacy concerns as they have to take longer audio recordings to perform the localization[9].
- Active sensing - compared to passive systems these systems not only record the environment but they send signals to the environment and record the room's response to these signals and based on that perform the localization. The signals are either in the audible range or inaudible. These systems reduce the needed recording times to avoid privacy concerns as well as provide more robustness against background noises[9].

Secondly, we can categorize the systems based on the underlying algorithms that perform localization. In that case, we can distinguish the following categories of algorithms:

- Analytical-based approach - this approach can be used both in active and passive scenarios and it uses different sound features to perform localization with mathematical methods. For example in a passive sensing scenario angle of arrival can be used for localization as in the case of [8]. Compared to active sensing scenarios where time-of-flight can be used for localization as described in [5].
- Fingerprint-based approach - this approach creates a fingerprint of a location based on sound features. This again can be used in active and passive scenarios where active would create the fingerprint based on the environment's response to the emitted signal, and passive would create the fingerprint just by looking at sounds present in the environment. The localization is then changed into a classification problem that tries to classify newly collected fingerprints based on earlier training. For the classification multiple models were used ranging from shallow models like knn[10] to deep learning models like cnn[9].

2.2 Related work

Since the focus of this research is on acoustic sensing this section presents some already developed acoustic systems. The section presents the system and talks about its known limitations with regard to different types of interferences.

Examples of passive sensing systems include:

- BatPhone[10] - BatPhone uses an acoustic background spectrum to perform room recognition. The acoustic background spectrum is an ambient sound fingerprint that allows the system to distinguish rooms. The fingerprint is made by recording a 10-second audio sample in the frequency interval of $0 - 7kHz$. To perform room recognition BatPhone uses the k nearest neighbor algorithm with the acoustic background spectrum as an input. As a result, it was shown that BatPhone achieves 69% accuracy. However, it was also shown that the passive fingerprinting model is very susceptible to noise. Interferences like chatter or conversations, while the audio sample is taken, reduce the accuracy significantly.
- SorroundSense[1] - SorroundSense is actually a hybrid system that uses passive acoustic sensing as one of the features based on which logical localization is performed. Besides the use of acoustic data, the system also uses optical and motion data to perform the localization. For the acoustic fingerprint of this system, two approaches were considered namely using the frequency domain as the fingerprint of the room, which proved ineffective, and using the amplitude of the signal in the time domain, which was chosen for the final system design. However, this type of fingerprint is not effective as the final accuracy of the system bearly outperformed random guessing[9].

Examples of active sensing systems include:

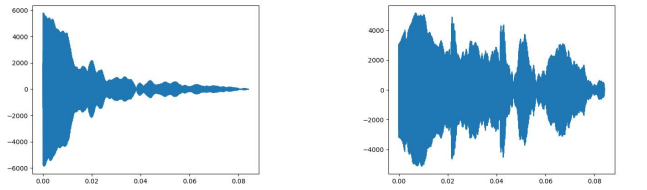
- RoomRecognize[9] - This system works by emitting an inaudible chirp into the environment and recording the room's response to the chirp. Based on the taken audio sample a spectrogram is created. This spectrogram is used as the fingerprint of the room. The room recognition is done with the use of a convolutional neural network. As a result, it was shown that the system can recognize 22 rooms with an accuracy of 99.7%. The paper also includes a study on the impact of different types of interferences on the system. Interferences considered in the work are sounds, phone position and orientation, surrounding moving people, and layout changes.
- RoomSense[7] - Compared to RoomRecognize this system uses a different approach for creating the chirp to be emitted by the system and a different classification algorithm. For chirp creation, RoomSense uses the Maximum Length Sequence technique which results in a longer audio signal of 0.68s compared to 0.002s used by RoomRecognize, the signal is also audible and described as noisy[7]. For classification, RoomSense uses Support Vector Machine with a Gaussian kernel. In the end, the work claims an accuracy 98% accuracy for room recognition between 20 rooms which is a similar result to the one achieved by RoomRecognize. This work also pointed out the impact of noise within the environment on the accuracy with an accuracy drop from 98.2% to 66.6% resulting from increasing noise within the recorded signal.

3 Measurement study

This short section presents the intuition behind the workings of the active acoustic sensing location system. First, the section proves the presence of an echoic response to the emitted chirp within the environment, after which examples of spectrograms from different locations are shown to see that the echoic response is a feature that can uniquely identify rooms. Lastly, some experiments were also run to see how music impacts the system.

3.1 Echoic response analysis

As stated earlier the system performs the localization based on the fingerprint of the response of the room to the emitted chirp. We can see that this fingerprint is in fact the analysis of echos within the environment. We can see that such echoes are present within the environment by performing a cross-correlation of the recorded sample with a 20kHz sine wave representing the chirp.



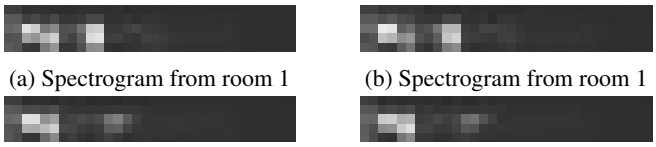
(a) Cross-correlation example 1

(b) Cross-correlation example 2

Figure 1: Cross-correlation between recorded samples and 20 kHz sine wave

Figure 1 shows the plotted cross-correlation from two different rooms. The figures prove that there are in fact echoes within the environment. Furthermore, we can also see that the two different rooms produced different echo responses giving the base for the classification based on the echoic response.

3.2 Between rooms spectrogram analysis



(a) Spectrogram from room 1

(b) Spectrogram from room 1

(c) Spectrogram from room 2

(d) Spectrogram from room 2

Figure 2: Pairs of spectrograms from different rooms

To see that rooms can be distinguished by their respective spectrograms multiple spectrograms from multiple different locations were created. From these spectrograms it can be seen that spectrograms are consistent within the same spot and different across different spots therefore they will allow for room recognition. Pairs of spectrograms from two rooms are presented in Figure 2. From the figure, it can again be seen that spectrograms are consistent within a room but different between rooms.

3.3 Impact of music on the system

One of the main objectives of this work is to improve the system's robustness against the presence of music in the environment. First, an analysis of how the system is impacted is performed. The hypothesis that was considered is the impact of audio wave interference on the collected audio sample based on which the localization is performed.

It is known that having more than one audio source within the environment will result in interaction between the waveforms produced by these sources. The interference can be either constructive or destructive depending on the phase shift and relative frequency of the two signals.

It should be noted that this interference is dependent on the frequency of the two signals. The next step to showing the impact of audio interference on the echoic response of the room is done by the following experiment. In the experiment, 200 samples are collected from the exact same spot within the room with the exact same phone orientation. Of the 200 samples 100 are gathered in the presence of music and 100 without music. Spectrograms are constructed from the samples. To show that music has an impact on the spectrograms average value of each spectrogram is tracked. Analysis of the averages shows that in the presence of music the average value of the spectrogram is more varied as the standard deviation increases.

(a) Mean spectrogram without music

(b) Mean spectrogram with music

Figure 3: Mean spectrograms of the experiment

To present the findings average spectrograms are created. These spectrograms are created by averaging the value of each data point on the spectrogram across the 100 samples. These spectrograms can be seen in Figure 3. From the result, it can be seen that the spectrograms differ in the presence of music.

4 Methodology

This section explains the methodology connected to the research. It explains both the method of answering the stated research question which is described in subsection 4.1 as well as the general design for the localization system implemented which can be seen in subsection 4.2.

4.1 Research methodology

To answer a research question one needs to establish a method that will lead to finding the answer. The method should also be documented well enough to make the conducted research reproducible for others. This section will explain how the research was conducted to find the answer to the stated research question.

The aim of this research is to study the impact of different types of interferences on the accuracy of active acoustic sensing systems. In principle, the method can be in the form of the following list of tasks.

- Create proof-of-concept application for location sensing.
- Gather dataset for the classifier training.
- Establishing baseline accuracy for the proof-of-concept application.
- Design experiments for different types of interferences.

- Gather datasets for experiments.
- Analyze experiment data.

Breaking down each of the listed components separately results in a full description of the methodology used for this research. First, the proof-of-concept application which was developed as a mobile app for Android smartphones with the classifier design closely following the one described in [9]. The dataset used for the classifier was gathered from four different rooms in a residential building. For the dataset, 500 samples from each room were collected.

Establishing the baseline accuracy of the app is an important step as there might be deviations from the established 99% accuracy. The application was implemented from nothing only by following the mentioned research therefore it might not achieve the same accuracy and use of different hardware and dataset may also change the accuracy. Therefore for later evaluation of experiment data a new baseline accuracy needs to be established.

To answer the research question a series of experiments will be performed to study whether the robustness against music of the system can be improved. To perform the experiments a dataset from the same rooms will be gathered in the presence of background music and with no background music.

The final step is to analyze the experiment results. This analysis and interpretation of results is what in the end answers the stated research question. When analyzing the results it is important to take into account and study the randomness of the experiment. This will be done by running experiments multiple times and looking at the average accuracy and standard deviation of the accuracy between experiment runs.

4.2 System design

To design the system first the input data needs to be established. Based on section 3 it can be concluded that spectrograms of the room's echoic response can be used to classify rooms. However, there are important details regarding the preprocessing and other parameters that have to be established to create the input spectrograms. This subsection discusses how exactly the spectrogram is created.

Chirp frequency

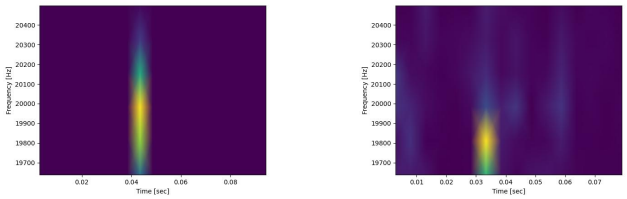
The system performs the classification based on a fingerprint of the response of the room to the emitted chirp. It is worth mentioning at this point that the frequency of the chirp is of high importance. We would like the chirp to be outside of the audible range, so above 20kHz, because of two reasons.

First of all the chirp being inaudible gives a better experience to the user by reducing how much the user can hear while using the system. A completely silent system however is not achievable due to imperfections in the smartphone audio systems. Therefore although the chirp itself is inaudible the user will be able to hear some noise during recognition.

Secondly, the advantage of having the chirp at such high frequency reduces the exposure to sound pollution in the environment. It was shown that the frequency of the recorded sample matters for the robustness of the system for example in the case of BatPhone reducing the sample frequency band from the original [0kHz, 7kHz] to [0kHz, 0.3kHz] improved the accuracy in the presence of background conversations[10]. The system improved the performance by reducing the hearing range to low frequencies which, similar to high frequencies, are also less polluted by the environment's background noise. Different systems did use different chirp frequencies for example [4] uses a range of 8 chirps in frequencies from 0.5kHz to 4kHz and the RoomRecognize uses chirps of 20kHz[9].

On the other hand, one needs to be aware of the limitations of smartphone acoustic systems. Different smartphones may have different capabilities for producing and recording high-frequency audio data. In general, we can see a trend of smartphones being less capable of producing such data above 20kHz. Therefore the system uses a chirp on the border of the audible range to reduce the impact of smartphone hardware. Smartphones high-frequency capabilities can be tested using Near Ultrasound Tests[2]

Fingerprint creation



(a) Spectrogram with chirp (b) Offset spectrogram

Figure 4: Spectrograms of one room with different offsets

To create a fingerprint of the echoic response the data is first transformed into a spectrogram in a narrow band range of [19.5kHz, 20.5kHz]. A spectrogram is a graph representing the strength of different frequencies of signal over time. For example Figure 4a Shows a spectrogram of a recording of the chirp emitted by the phone. We can clearly see the emitted chirp at around 0.02 seconds from the recording start. However, to use the spectrogram for classification we actually limit the sample interval to not include the original chirp as it would skew the data too much. So for the actual spectrogram, the window is shortened and offset to not include the original chirp as can be seen in Figure 4b.

Figure 5: Example final spectrogram

The last step in the data preparation process is to rescale the spectrogram. The final spectrogram is a grayscale image of size 5x32 as seen in Figure 5

[2]Near Ultrasound Tests - https://source.android.com/docs/compatibility/cts/near-ultrasound

Music removal

An important part of the system is removing noise from the spectrograms that occurs due to background music being present in the environment. This part is the actual extension of the system described in [9]. This addition is meant to improve the robustness of the system against background music which directly calls back to the main research question of this work.

The music removal from audio samples can be generally classified as a denoising task. To perform denoising one might use a denoising autoencoder (DAE) or a convolutional denoising autoencoder (CDAE). In principle, music removal with the use of CDAE was already done in [11] with satisfying results. However, in [11] the CDAE was used on the audio data while in this work the CDAE is used on the spectrogram. Therefore the network architecture will differ for the autoencoder used for noise removal from the spectrogram.

Autoencoder design

Since the autoencoder will be run on spectrograms rather than raw audio data the network design differs from the one described in [11].

Encoder network layers:

1. Convolutional layer - 16 5 x 5 filters
2. Max pooling layer - 2 x 2 filter
3. Convolutional layer - 16 5 x 5 filters
4. Max pooling layer - 2 x 2 filter

Decoder network layers:

1. Transposed convolution layer - 16 5 x 5 filters
2. Transposed convolution layer - 16 5 x 5 filters
3. Convolutional layer - 1 5 x 5 filter

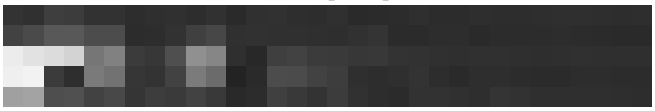
Additionally, every layer has zero padding added so that the output shape is the same as the input shape. The output layer uses sigmoid activation and all other layers use Tanh, all layers use stride equal to 1.

For training the autoencoder two equal size datasets of spectrograms have to be taken from environments with and without music. For the purpose of the proof-of-concept application the dataset gathered looked as follows. Data was gathered from 4 rooms and in each room 200 samples were taken from the same locations where the localization is performed from. The setup was performed twice with and without the presence of music. Spectrograms were matched on the room they were taken from so that samples with and without noise from a given room are fed for training as example input and output. This results in 800 pairs of input-output for training from which 1/5 pairs are taken for the validation dataset.

Resulting model

The resulting model takes as input a sample and denoises it with regard to interferences occurring due to wave interference with music. Figure 6 shows example input and example output generated by the autoencoder. Although the spectrograms differ the output is still unique per room. This change in spectrograms requires training the main room classifier on the outputs of the autoencoder. Note that training on autoencoder outputs and original spectrograms on the same dataset resulted in very similar validation accuracies of the two models.

(a) Example input



(b) Example outputs

Figure 6: Example input and output for the autoencoder

5 Implementation

This section is a concrete explanation of the implementation of the proof-of-concept application used for the experiments. The application code base can also be found at https://github.com/filip-bilinski/IndoorNavigationRP. Note that this repository is a fork as the frontend of the application was developed in collaboration with other students following the project on Indoor Location Sensing Using Smartphone Acoustic System.

5.1 General architecture

The application is split between the front-end and the back-end. The front-end is a client application implemented in Java for Android smartphones. The back-end provides a REST API for the client that was written in Python using Flask[3] and is connected to a self-hosted MongoDB[4] database for dataset saving.

The front-end and back-end perform different tasks to make the whole system functional. The front-end is mainly responsible for gathering the dataset, sampling for classification, and displaying the classification results. In general, the back-end is responsible for the classification as this is where the model is deployed. The back-end also provides endpoints for creating new models, training models, saving models, and loading saved models into the program.

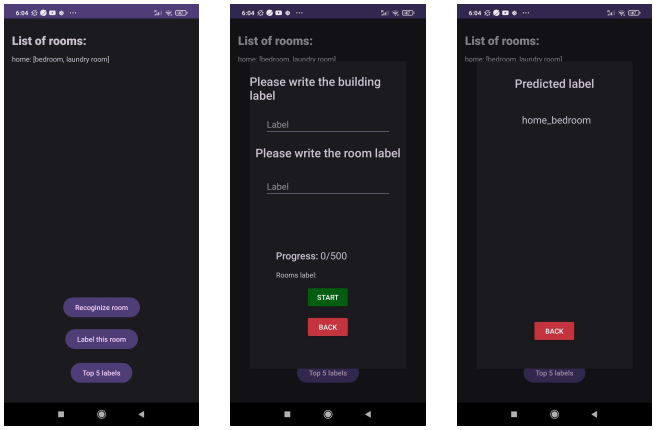
5.2 Front-end

The main functionality of the front-end is to emit the chirp and record the response. This functionality is needed both for dataset creation as well as performing localization. This was implemented using the standard Android API for emitting and recording audio. To emit the chirp a sine wave of 20kHz frequency and 2ms length is generated and written to a buffer that then replays that data with phone's loudspeaker.

[3]Flask - https://flask.palletsprojects.com/en/2.3.x/

[4]MongoDB - https://www.mongodb.com/

At the same time, a recording is taken. Since the audio play and audio recording happen on different threads of the application a synchronization with a barrier was added to decrease the offset between the start of the recording and the playing of the chirp as much as possible.



(a) Main scene (b) Add label scene (c) Prediction scene

Figure 7: Differnt screens of the application UI

Besides the chirp emission and audio recording the front-end can also communicate its data to the back-end to perform localization or add new room labels to the dataset. This can be done by the user with the designed UI. Main UI scenes can be seen in Figure 7.

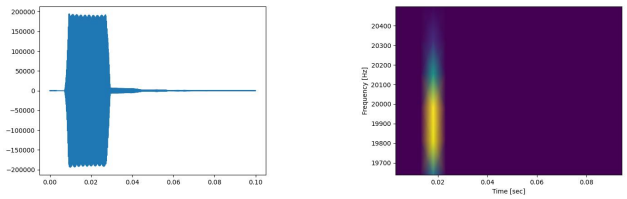
Figure 7a shows the design of the main scene. This is where the user can choose between adding a new room to the dataset or running the localization. The user can also see a list of all rooms organized per building. If the user chooses to add to the data set they will see the scene as the one in Figure 7b where they may add labels of the building and room and then by pressing start collect samples from the room. If the user wants to run the localization instead they will see a popup as the one in Figure 7c where the label resulting from running the classifier on a newly collected sample will be shown.

5.3 Back-end

The back-end of the application is responsible for two main tasks. First of all it is responsible for the creation of the fingerprint spectrograms for the classifier as well as saving them to the database. The other task is to manage the classifier itself which includes creating new models, training models, making predictions from the model.

The creation of the spectrograms is a non-trivial task. Even though the clientside attempts to synchronize the recording start with chirp emission with the barrier the synchronization is far from optimal. And as we have seen before a spectrogram that includes the original chirp does not show any visible echos due to them being of much lower magnitude than the chirp. Therefore the server needs to find the original chirps within the recorded data to then construct a proper timeframe between two chirps.

To find the chirps a cross-correlation of the recorded audio with 20kHz sine wave is calculated. The software can



(a) Cross-corellation plot (b) Offset spectrogram

Figure 8: Cross correlation over time and corresponding spectrogram

then find the chirp position as well as length by finding an area of high cross-correlation. This can be seen in Figure 8 where both spectrogram and cross-correlation plot was created for the same recording period. From those two figures, we can see that the chirp corresponds to the area of high cross-correlation on the plot. The recording frame is then taken by finding two consecutive chirps and taking the time frame in between as the echo period from which the fingerprint is created.

The model underlying the classifier is a convolutional neural network exactly the same as the one described in [9]. In short summary, the model uses two convolutional layers with 16 and 32 4x4 filters each of which is followed by a pooling layer with a 2x2 filter. The model is finished by adding two dense layers one with 1024 ReLUs and the output layers with K ReLUs with K representing the number of different rooms the model can recognize.

As mentioned the server is capable of initializing new models as the one described, training them, saving them to a file and performing predictions.

6 Experimental Setup and Results

This section presents the experimental setup of the run experiment and the achieved results of the experiment. subsection 6.1 explains the setup while subsection 6.2 presents the results.

6.1 Experimental setup

The experiment's aim is to answer whether the addition of the CDAE for music interference removal from the spectrograms improves the robustness of the system. To achieve this the following experiment design is proposed. From each room, 100 samples are taken with and without music in the background. Next two system setups are created. First, where the model was trained without the use of autoencoder, and classification is done without its use either. Second, where the model was trained on autoencoder outputs and classification is performed on autoencoder outputs.

To evaluate the robustness we evaluate the accuracy of both models on three datasets:

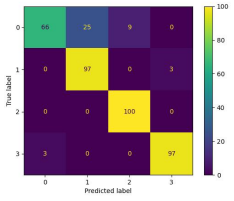
1. Dataset without background music.
2. Dataset with background music.
3. Mixed dataset.

	No Autoencoder		With Autoencoder	
	Average	Standard deviation	Average	Standard Deviation
No Music	0.91	0.07	0.81	0.05
With Music	0.70	0.10	0.76	0.09
Mixed	0.81	0.06	0.79	0.04

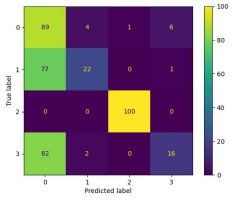
Table 1: Accuracies from the experiment

Analysis of the accuracies together with confusion matrices on each of the datasets for different setups will be used to conclude whether the autoencoder improves the robustness of the system against background music.

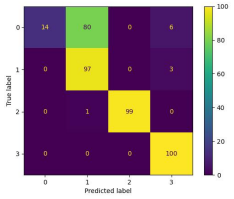
6.2 Results



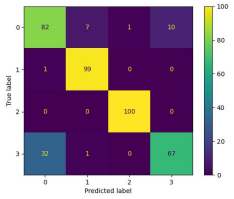
(a) No music, No autoencoder



(b) Music, No autoencoder



(c) No music, Autoencoder



(d) Music, Autoencoder

Figure 9: Result confusion matrices of the experiment

Figure 9 presents the resulting confusion matrices of the introduced experiment. Furthermore, the accuracies of the experiment are presented in Table 1. From the resulting accuracies, we can see that the best performance for no music and mixed dataseta is achieved without the use of an autoencoder but in the dataset with music introducing the autoencoder achieves higher accuracies.

7 Responsible Research

This section will talk about ethical issues connected to this project as well as discuss the reproducibility of the research. First subsection 7.1 discusses the ethical issue of recording audio data for the system and then subsection 7.2 talks about the reproducibility of this research.

7.1 Ethical concearns

As with any research, this work raises some ethical concerns. In particular, the ethical issue with this research is the recording of audio data.

Audio recordings may carry speech recordings within themselves. Speech is considered personally identifiable information and therefore recordings of speech are privacy-violating. Some legislations even forbid recordings of speech of third parites[6] which would make gathering a dataset in public spaces illegal. However, there are some considerations to be taken when evaluating whether the recordings taken within the proof-of-concept applications invade people's privacy.

First of all the saved recordings are around 80ms long. This in itself provides more privacy as such short recording times decrease the chance of speech being present in the recording as well as such short part of speech does not carry a lot of information. While this is true in the database for training a batch of 500 samples is saved from one recording. While there is no information about the order of the samples in the database and there is a 20ms gap between any two samples in theory a longer recording with gaps could be reconstructed by brute force.

That's why the saved spectrograms only correspond to the frequencies around the emitted chirp frequency. Limiting the frequencies gets rid of any trace of speech in the saved spectrograms. Therefore we can conclude that all the data saved by the application and used for training is privacy-preserving.

For this proof-of-concept app saved data being privacy respecting is sufficient. However in general it is not perfect as the privacy-sensitive data is sent from the client to the server. To improve systems privacy the data should be filtered on the front-end and only then sent to the server. Limiting the frequencies before sending would require the system to create spectrograms on the client side which is at the moment done on the server side. This could be changed or alternatively, a speech filtering similar to the one described in [6] could be applied before the data is sent. This should be taken into account when developing a full solution for location recognition with active acoustic sensing however is out of scope for this project.

7.2 Reproducability

This report was designed with reproducibility in mind. There are areas in which the description could be improved for easier reproducibility however given the time period of the project not everything can be fully developed.

The only part of this research that will not be shared is the gathered dataset. Sharing of this dataset however should not be mandatory for the reproducibility of this research. Since the full application codebase is available at github one can create a very similar dataset for their purposes.

Experiments done should also be able to be reproduced. The description of each experiment is included in section 6 and the scripts for result analysis are available on github.

8 Discussion

The main objective of this research was to investigate the impact of background music on the accuracy of active acoustic localization systems and answer whether it is possible to increase the robustness of such systems against background music. The experiments performed to conclude on these topics can be summarized as follows.

Regarding the impact of background music, it can be seen that the system is negatively impacted by the presence of music in the environment. In particular, the results from Table 1

show around 10% accuracy drop between datasets with and without music. This coincides with the intuition given in section 3 where it was shown that on average the spectrograms from the exact same location differ in the presence of music and with the study done in [9].

Coming to the second part of the research is to improve the robustness of the algorithm against background music autoencoder was considered for denoising the spectrograms. From the results it can be seen that in fact, introduction of the autoencoder comes with a trade-off. On average the system performs better on the dataset with music when autoencoder is used, however, it also performs worse on the dataset without music. In the mixed dataset, both setups perform similarly to the point where it is not possible to unambiguously determine which one performs better due to randomness in the experiment. Therefore, in general, it can be concluded the robustness in this particular setup stayed relatively the same.

It has to be taken into account that the performed experiment had randomness involved. The randomness comes from the random train, validation dataset split, and random initialization of the model weights. As a result, each re-train of the models will achieve slightly different results in the experiment. The standard deviations of the presented results are also presented in Table 1. The averages and standard deviations are calculated from 6 runs of the experiment.

9 Conclusions and Future Work

Summarizing, this work aimed to answer the following research question: Can the robustness of the system against the music-containing environment be improved? To answer this question first the impact of the music on the classifier was studied and the reasoning behind the impact was given. To improve the robustness against music a denoising autoencoder was used as an attempt to remove noise in the spectrograms coming from interference with music.

In the end, the robustness of the system was not improved. However, the use of autoencoder increased the accuracy in music containing the environment even though the average accuracy of the classifier was lower than that of the system without the autoencoder. While the system without the autoencoder experiences around 10% accuracy drop in the presence of music with the autoencoder the accuracy drop was reduced by a factor of two.

The reduced accuracy drop with the use of autoencoder shows that with more research autoencoder could actually be used to improve the overall accuracy of the system. To achieve that the following areas could be further explored:

- Developing a more generic autoencoder - The autoencoder used for the purpose of this research was trained on data points from the same spots as the localization is performed. Therefore it isn't a generic denoising autoencoder as it would not be able to remove noise from a spectrogram taken from a random location. During the research, an attempt to train on random data points was performed however with the current network design such an autoencoder performed worse. Developing such an autoencoder could increase the overall accuracy.
- Devloping a smart switching system - Since the accuracy of the system in the presence of music was higher with the use of autoencoder but lower in a quiet environment one could develop a smart system that would switch between the two classifiers based on the current state of the environment. To develop such a system an analysis of audio could be done before creating the fingerprints to determine which classifier should be used. This would certainly improve the robustness of the system
- Using autoencoder on different types of data - For the purpose of this work, the autoencoder was used on the specific 32x5 spectrograms that are used by the system. One could consider using the autoencoder on higher resolution spectrograms or even different types of data like for example the original audio signal.
- Use of signal processing methods - Another area that could be explored to improve the current version of the system is signal processing. Applying signal processing techniques together with the autoencoder or instead of the autoencoder would give a comparison of it's performance and could also increase the accuracy.

References

[1] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: Mobile phone localization via ambience fingerprinting. MobiCom '09, page 261–272, New York, NY, USA, 2009. Association for Computing Machinery.

[2] G. Dedes and A.G. Dempster. Indoor gps positioning - challenges and opportunities. In *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, volume 1, pages 412–415, 2005.

[3] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, and Dan S. Wallach andLydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 70–84, 2004.

[4] Kai Kunze and Paul Lukowicz. Symbolic object localization through active sampling of acceleration and sound signatures. volume 4717, pages 163–180, 09 2007.

[5] Jie Lian, Jiadong Lou, Li Chen, and Xu Yuan. Echospot: Spotting your locations via acoustic sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(3), sep 2021.

[6] Daniyal Liaqat, Ebrahim Nemati, Mahbubur Rahman, and Jilong Kuang. A method for preserving privacy during audio recordings by filtering speech. In *2017 IEEE Life Sciences Conference (LSC)*, pages 79–82, 2017.

[7] Mirco Rossi, Julia Seiter, Oliver Amft, Seraina Buchmeier, and Gerhard Tröster. Roomsense: An indoor positioning system for smartphones using active sound probing. pages 89–95, 03 2013.

[8] Sheng Shen, Daguan Chen, Yu-Lin Wei, Zhijian Yang, and Romit Roy Choudhury. Voice localization using nearby wall reflections. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Qun Song, Chaojie Gu, and Rui Tan. Deep room recognition using inaudible echos, 2018.

[10] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *MobiSys '11: Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 155–168, 2011.

[11] Mengyuan Zhao, Dong Wang, Zhiyong Zhang, and Xuewei Zhang. Music removal by convolutional denoising autoencoder in speech recognition. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 338–341, 2015.