Sound Source Localization and Distance Estimation in Open Environment using Simulation and AI

Master Thesis

Denis Rosset

University of Applied Sciences and Arts Western Switzerland

July 2023

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Supervisors:

Michael Mäder: Professor in computer science
Beat Wolf: Professor in computer science

Principals

Marc-Antoine Fénart: Professor in civil engineering
Gabriel Python: Scientific associate at Rosas

Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.[**?**]

Contents

1
Introduction

Within the framework of the research project ”NPR Teleoperation,” the engineers of the HEIA-FR have developed the first concept in Switzerland of a remote-controlled automated vehicle. However, teleoperation only makes sense if the vehicle is automated. There can be no teleoperation without automation (economic factors), just as there can be no automation without teleoperation (legal, technical, and social factors). ROSAS then created the Autovete (Automatisation de véhicules téléopérés) project. HEIA-FR finances them to build up vehicle automation expertise. Detecting other emergency vehicles is mandatory for a vehicle to be fully autonomous. V2V (Vehicle-to-Vehicle) communication is a solution but is not yet integrated into emergency vehicles. So, to detect such a vehicle, two signals need to be processed: the sound of the emergency siren and the blinking lights of the vehicle. The first use case of this project focuses only on sound source distance estimation and localization. Detecting excessively noisy vehicles on the street is a simpler use case created for this project to understand if sound source estimation and localization could work for vehicles in an open environment. The goal is to measure the sound level of the passing vehicles and compare it with the legal limits. If a vehicle exceeds the limit, the system can record its license plate and report it to the authorities. This way, the system can help reduce noise pollution and improve road safety. This system requires a microphone array, a camera, and a processing unit to achieve the needed detection. The microphone array captures the sound signals from different positions and sends them to the processing unit. The processing unit applies a sound source localization algorithm to estimate the direction and distance of the sound source. The camera captures the image of the vehicle and performs license plate recognition.

Big improvements in sound source localization with the help of machine learning are being made. They can be used to reliably localize the origin of a sound using one or more microphone arrays (multiple microphones operating in tandem). A non-negligible problem is the small number of real-world datasets with moving sources in an open environment. A solution is to create datasets in realistic sound propagation simulation and use them to augment the real-world datasets. A model can then be trained on the augmented dataset and tested for its performance in real-world data.

Using machine learning to solve the sound source localization problem can lead to a new attack vector for the system. The system can be attacked by modifying the sound source or the sound signal. Tests to understand how the system reacts to such attacks are necessary to understand the system’s robustness.

1.1 Motivation

1.1.1 Objectives

1.1.1.1 Objective n°1 Definition of a baseline

The first objective is to define a baseline for the project. The baseline should contain the problem statement, the project's scope, and objectives. The baseline will help us understand the project's context and the problem we are trying to solve.

1.1.1.2 Objective n°2 Dataset according to the baseline

The first objective is constructing a coherent dataset with the project's baseline. The dataset contains the target variable, features, and necessary pre-processing steps. This dataset will help represent and understand the problem.

1.1.1.3 Objective n°3 Realising a model for better sound source localization and distance estimation

The project uses a neural network model to detect the origin of a sound using a microphone array. The neural network uses the dataset created in objective 2 for training and can localize the sound source accurately. An evaluation of the trained neural network model in a real environment allows us to see how it performs. The model will also be evaluated to understand how it can be improved.

1.1.1.4 Objective n°4 Attacking the model to understand how it reacts

The trained neural network model needs to be evaluated by testing it on data modified in some way, such as by adding or removing noise or modifying the sound source. Tests against various attacks, such as masking, time-warping, and frequency-shifting, will help us understand how it performs.

1.1.2 Challenges

1.1.2.1 Challenge n°1: Realistic datasets

One of the main challenges in this project is to construct realistic datasets that accurately capture the data in a real-world scenario. The lack of open-source datasets that contain moving sound sources in open environments can make this difficult. The project should use realistic simulations of sound propagation to produce suitable datasets.

1.1.2.2 Challenge n°2: Robust models

Another challenge is to create a neural network model that can accurately detect the origin of a sound using a microphone array. The trained model must be accurate and robust to resist adversarial attacks. The model should localize the sound source accurately on altered data.

1.1.2.3 Challenge n°3: Adequate evaluation metrics

The last challenge is to create an evaluation metric that adequately reflects the model's performance. The evaluation metric should consider the accuracy of sound source localization in a real environment and its ability to resist adversarial attacks. The metric should also capture how well the model can provide reliable results in various environments.

1.2 Structure of the thesis

- **Chapter 2: Background and Litterature**: This chapter provides an overview of the background knowledge necessary to understand the project, such as the theory of machine learning and sound propagation, and a brief introduction to sound source localization.
- **Chapter 3: Methodology**: This chapter describes the methodology used to achieve the project's objectives, such as the dataset creation, the simulation, the neural network model, and the evaluation metrics.
- **Chapter 4: Realization**: This chapter describes the work done to achieve the project's objectives, such as creating a dataset, creating a simulation, building a neural network model, and evaluating it using an appropriate metric.
- **Chapter 5: Results and Analysis**: This chapter presents the project's results, such as the performance of the neural network model and the evaluation metrics. It also analyzes the results and discusses the findings.
- **Chapter 6: Conclusion and future work**: This chapter concludes this project by discussing the main results and summarizing the key findings.

2
Background and Litterature

This chapter introduces technical concepts and background used in the conceptualized solution of the thesis. It also explains the analysis of the needs of the thesis and finds relations with the current state of research in sound source localization systems.

2.1 Baseline analysis

During the first weeks of the thesis, we had the opportunity to place an installation of microphones on the HEIA-FR main building roof. We took that opportunity to design the baseline and analyze how to build a system around it.

After analyzing the road in front of the HEIA-FR main building, we decided to use the baseline to detect the position of vehicles driving on the road. The road is moderately busy, and the vehicles drive at a reasonable speed. The road is also straight, which makes it easier to detect the position of the vehicles. The baseline is shown in Figure 3.1. This analysis helps provide an intuitive understanding of the sound source localization system. The baseline comprises a vehicle as the sound source we want to record, multiple microphones recording the sound of the street, and an embedded system that manages the microphones.

2.2 Sound Source Localization

Sound Source Localization (SSL) is the process of determining the position of a sound source. It usually uses a microphone array that captures the sound signals from multiple directions. Various applications use SSL [1], such as speech recognition [2], source separation [3], human-robot interaction [4] or room acoustic analysis [5]. In this thesis, SSL is used to estimate the distance and direction of a sound source to detect excessively noisy vehicles.

2.2.1 Spectrograms for sound visualization

Spectrograms are a visual representation of the frequency content of a sound signal. They are often used in sound source localization to identify the direction of a sound source. The spectrogram is a two-dimensional representation of the frequency content of a sound signal (figure 2.1).

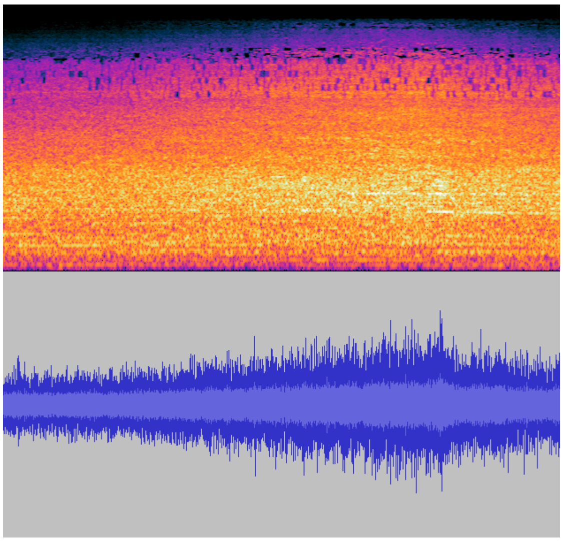


Figure 2.1: Spectrogram of a sound signal

The x-axis represents time, and the y-axis represents frequency. The intensity of the color at each point in the spectrogram represents the amplitude of the frequency component. A matrix of spectrograms allows the representation of multiple channels, such as the ones recorded by a microphone array. On that matrix, each spectrogram represents the frequency content of a single channel (figure 2.2).

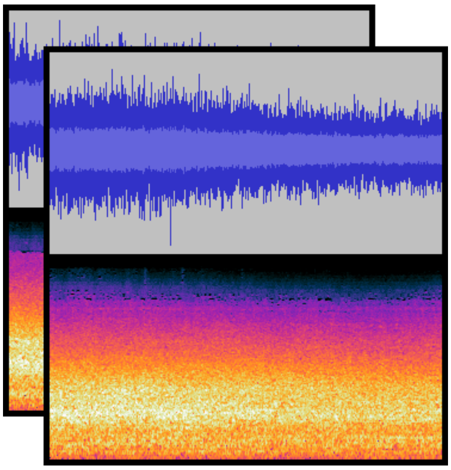


Figure 2.2: Dual channel spectrogram matrix of a sound signal

Looking at the frequency content of the sound signal allows us to identify the time delta of a recorded sound by using a multi-channel spectrogram. The bright spot on the spectrogram will indicate a jump in the amplitude and determine the start time of the recording of a loud sound. By comparing this time with the other channel, we can find the direction of the sound source by comparing the sound signal's time delta with the other channels' time delta (figure 2.3).

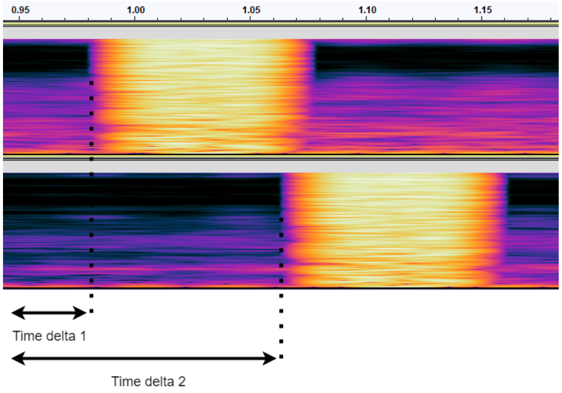


Figure 2.3: Spectrogram of two sound signals with time their delta

Since we know the distance between the microphones, we can determine the direction of the sound.

2.2.2 Origin of sound using two microphones

Admitting the following setup (figure 2.4), if the time delta 1 is greater than the time delta 2 of the other channels (setup 1), the sound source is closer to microphone 2. If the time delta 1 equals the time delta 2 (setup 3), the sound source is at the same distance to both microphones. If the time delta 2 is greater than the time delta 1 (setup 2), the sound source is closer to the microphone 1.

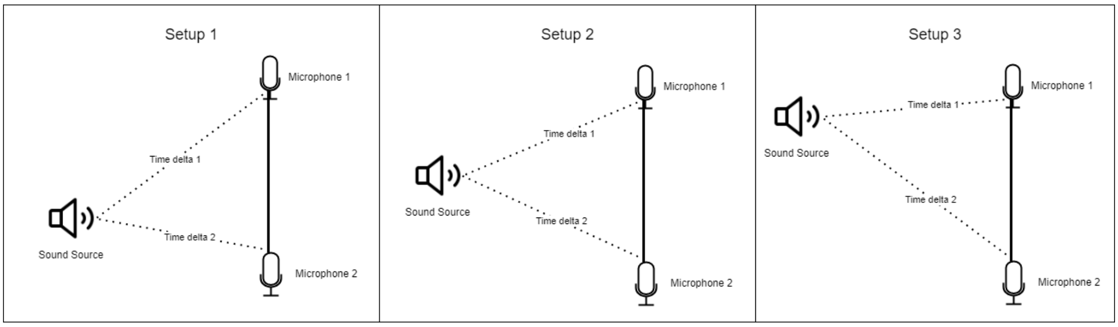


Figure 2.4: Sound source localization setup

This concept can be formalized and is better explained in [6]. Once the delay between the two microphones is known, the equation allows us to find the direction of the sound source by using trigonometric calculations. As in the figure 2.5, considering point M as the sound source and point A and B as microphones, the distance between the two microphones is d and the time delta between the two microphones is Δt, the angle α can be calculated.

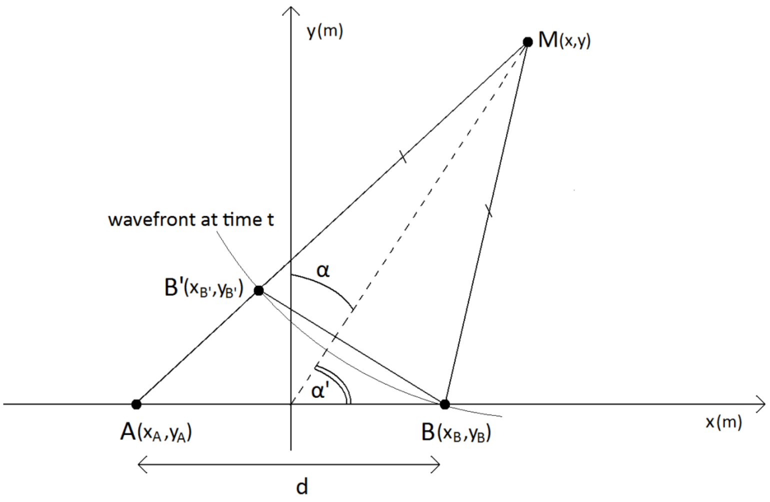


Figure 2.5: Equation formalization. Original image from [6]

Looking at the graphic allows us to find the following equation:

$$AB' = AM - B'M \tag{2.1}$$

With Pythagorean theorem:

$$AM = \sqrt{(X_a - X)^2 + (Y_a - Y)^2} \tag{2.2}$$

$$BM = \sqrt{(X_b - X)^2 + (Y_b - Y)^2} \tag{2.3}$$

The two microphones have the same Y coordinate, so $Y_a = Y_b = Y$ and $Y_a - Y_b = 0$ and $X_a = -X_B$ The equation becomes:

$$y = \pm\sqrt{\frac{AB'^2}{4} - x_B^2 + x^2(\frac{4 \cdot x_B^2}{AB'^2} - 1)} \tag{2.4}$$

This setup shows that two microphones are enough to determine the direction of a sound source.

2.3 Neural networks

The report [A survey of sound source localization with deep learning methods][1] shows that deep neural networks achieve good scores in sound source localization. Neural networks are machine learning algorithms based on biological neurons used to solve various problems, including image recognition, speech recognition, and natural language processing. Neural networks learn from provided data to solve a problem without explicitly programming the solution. Many domains, like self-driving cars, facial recognition, and medical imaging, achieve state of the art results using neural network models.

A neural network (figure 2.6) is composed of multiple neurons (the circles) that are organized in layers and connected to the neurons in the previous and next layers.

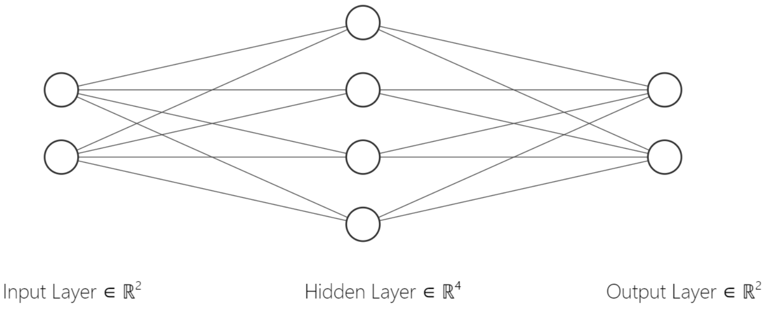


Figure 2.6: Neural network

Neural networks comprise multiple neurons. Neurons are mathematical functions with activation functions and weights. These determine how the neurons respond to inputs and connect to other neurons. Neural networks are trained by adjusting the weights to minimize the error between the predicted and desired outputs, using methods like gradient descent[7] and backpropagation[8].

2.3.1 Deep Neural Networks

Deep neural networks are a type of neural network composed of multiple layers of neurons[9]. They are trained on a large dataset of images and then used to classify new images. There are countless architectures [10] and implementations of neural networks, but they all share the same basic principles. The most known architectures of neural networks include CNNs[11], transformers[12], and many others.

2.3.2 Convolutional Neural Networks for sound source localization

Convolutional Neural Networks (CNNs)[11] are deep neural networks specifically used for image recognition. They are often composed of convolutional, subsampling, and fully connected layers (Figure 2.7).

- Convolutional layers are used to extract features from images. These features are then fed into fully connected layers to perform classification. Each convolutional layer comprises multiple filters convolved with the input image to produce a feature map. The filters are trained to extract specific features from the input image.

- Subsampling layers are used to reduce the size of the feature maps. The most common subsampling layer is the max-pooling layer, which takes the maximum value of a specific region of the feature map.

- Fully connected layers are trained to classify the features extracted by the convolutional layers. The output of the fully connected layers is a probability distribution over the possible classes.

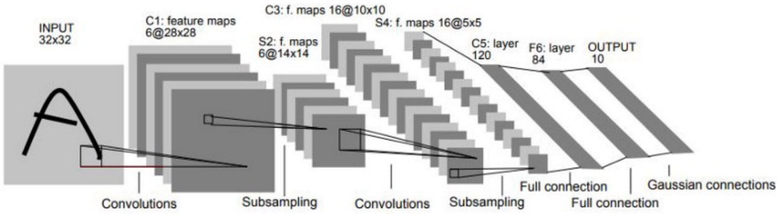


Figure 2.7: CNN architecture example with LeNet-5 [13] composed of two convolutional layers, two subsampling layers, and finishing with two fully connected layers.

Even if CNNs are mainly used to classify photography, they can classify any images, including sounds[1]. Based on section 2.2.1, CNNs can use spectrograms as input since they also are images. The spectrograms are converted into images and fed into the CNN.

An approach for sound source localization is using classes to define zones where the sound can come from as the classes. The CNN will output a probability distribution over the possible classes. The class with the highest probability is the predicted class. The predicted class can then refer to a zone.

The CNN then outputs a probability distribution over the possible classes. The possible classes need to be defined before training the CNN. In[14], they approach the problem with 15 classes, using angles 0, 30, and 60 degrees and distances 1, 2, and 3 meters (Figure 2.8).

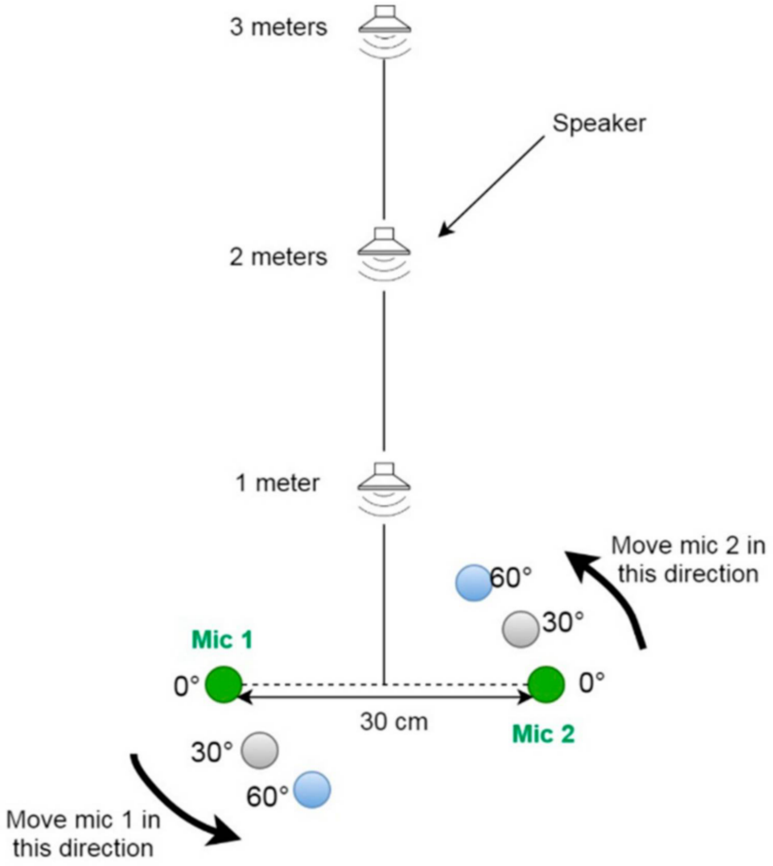


Figure 2.8: CNN for source localization

Based on the baseline defined in chapter **??**

2.3.3 Transfer learning

Transfer learning is a machine learning technique where a model trained on a specific task is reused as the starting point for a model on a different task. The model is trained on a large dataset and then used as a starting point for a model on a different dataset. The model is then trained on the new dataset, and the weights are adjusted to minimize the error between the predicted and desired outputs. Transfer learning is used to train models on smaller datasets and achieve better results than training the model from scratch.

2.4 Datasets for sound source localization

Datasets are needed to train and test neural networks. They are composed of data and labels. The data is the neural network input, and the label is the expected output. In the case of sound source localization, the data is audio, and the labels are the zones of the sound source.

Multiple datasets exist in sound source localization for neural networks. The most common are the DCASE 2019 task 3 dataset[15] and the DCASE 2020 task 3 dataset[16]. These datasets are composed of audio files and the corresponding labels. The labels are the zones of the sound source. The audio files are recorded in a room with a microphone array and a sound source. The sound source is moved around the room, and the audio is recorded. The audio files are then annotated with the zones of the sound source. The annotations are done manually by listening to the audio files and annotating the zones. Multiple annotators then verify the annotations to ensure the quality of the annotations.

Although these datasets are good baselines for sound source localization, they do not suit the needs of this project. The datasets are recorded in a closed environment and do not reflect the baseline defined in this project. Still, these datasets are good baselines for sound source localization and help to understand how to create a dataset.

2.4.1 Dataset augmentation for audio classification

Since recording many audio files is time-consuming and costly, and since the dataset needs to be large to train a neural network, we decided to use dataset augmentation techniques.

Dataset augmentation is a technique used to increase the size of a dataset. It is used to improve the performance of a neural network by training on more data, thus becoming better at generalizing. The most common techniques are flipping, rotating, and cropping images, but since the classification in this project is realized on audio, other techniques are needed to augment the dataset.

Some techniques that work well on audio are adding noise, changing the pitch, or simulating new data.

2.5 Dataset simulation for sound source classification

Simulating a dataset is a technique used to create a dataset without recording audio from real life. It helps to create a dataset with many samples and labels. Since the goal is to generate sounds in an open environment, a 3D-capable engine is necessary. Game engines are increasingly used for simulation since they are optimized for real-time rendering and can simulate complex 3D scenes. The game engine must simulate sound propagation for a realistic audio simulation.

2.5.1 Sound propagation

Sound propagation is the physical process by which sound waves propagate in a given environment. Multiple factors affect the propagation of sound waves, including reverberation, occlusion, doppler effect, and obstruction.

The strength of the sound wave depends on various factors, including the frequency, environment, and distance from the sound source. These factors make an accurate identification and localization of a sound source difficult; thus, we need a more accurate and robust sound source localization system.

2.5.2 Microsoft Project Acoustics

Microsoft Project Acoustics is a sound propagation engine that simulates the propagation of sound waves in a given environment. Various applications, including video games, virtual reality, and physics simulation, use this engine. It simulates wave effects like obstruction, reverberation, and occlusion in complex 3D scenes without requiring zone markup or raytracing. It works similarly to a raytracing engine but is precomputed and optimized for real-time performance.

2.5.2.1 Sound Propagation in game-engine

2.6 Adversarial Attacks

Adversarial attacks are a manipulation technique that aims to fool a neural network by modifying the input data. The goal is to make the neural network misclassify. Adversarial attacks allow us to test the robustness of neural networks and understand how neural networks work and how we can improve them.

2.6.1 Adversarial attacks categories

Adversarial attacks can be categorized into white-box attacks and black-box attacks. White-box attacks are attacks where the attacker has access to the neural network's parameters and architecture. Conversely, black-box attacks are attacks where the attacker cannot access the neural network's parameters and architecture.

2.6.2 Fast Gradient Sign Method

One of the most common adversarial attacks is the Fast Gradient Sign Method (FGSM)[17]. It is a white-box attack that uses the gradient of the loss function to find the adversarial example. The adversarial example is calculated using the following equation:

$$X_{adv} = X + \epsilon \cdot sign(\nabla_X J(\theta, X, y)) \tag{2.5}$$

X is the input, y is the target class, ϵ is the magnitude of the perturbation, and $J(\theta, X, y)$ is the loss function. The loss function is the function that the neural network normally tries to minimize, but here is used to maximize the loss. The gradient of the loss function is calculated for the input X. The sign of the gradient is then calculated and multiplied by the magnitude of the perturbation ϵ. The result is added to the input to create the adversarial example X_{adv}. The adversarial example is then fed into the neural network, which outputs the adversarial class (Figure 2.9).

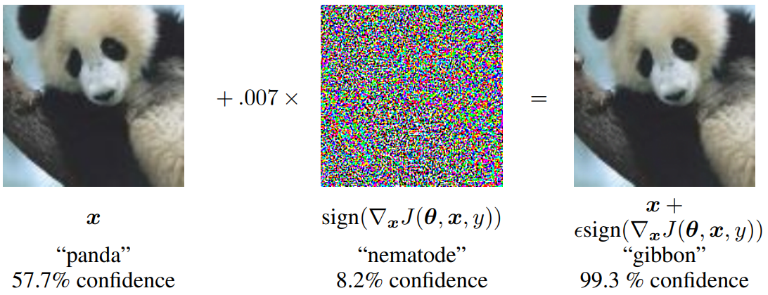

$+ .007 \times$
$=$
$\boldsymbol{x}$
“panda”
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“nematode”
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“gibbon”
99.3 % confidence

Figure 2.9: FGSM example in [17] with a neural network classifying a panda as a gibbon because of the attack.

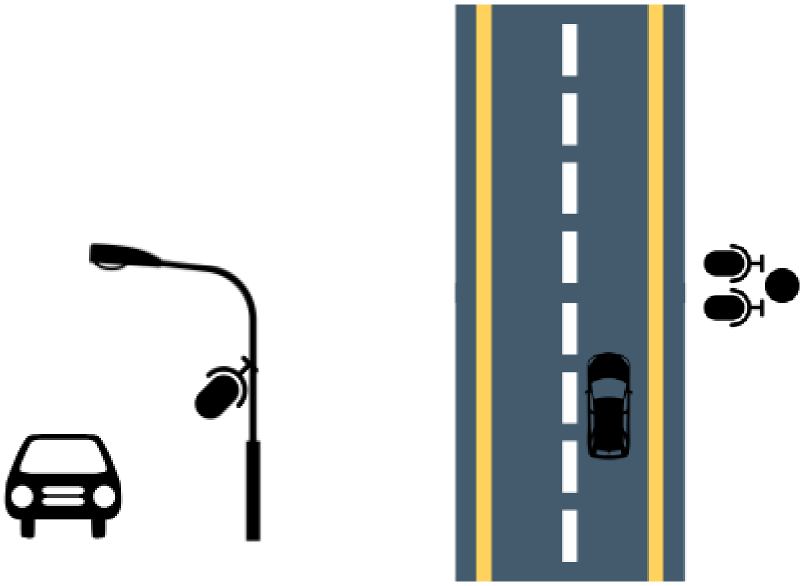
3
Methodology

This chapter presents the methodology used to create the dataset, train the neural network, and perform the adversarial attack.

3.1 Baseline conception

We need to conceptualize a baseline since we are starting the project without previous work. The baseline is the starting point of the project. It is the simplest system that we can create to solve the problem. We can then use the baseline to compare our results and improve the system.

We conceptualized the baseline of the system as follows: we place the microphones on the side of the road, and the sound source is the vehicle. The microphones record the sound emitted by the vehicles driving on the road. The sound source localization system then detects the vehicle's position. The setup of the baseline is shown in Figure 3.1.



(a) Side view

(b) Top down view

Figure 3.1: Setup of the baseline

Using microphones and an embedded system to detect vehicle positions is an efficient and effective way of providing a real-time sound source localization system. This idea can be further developed to incorporate other sound sources for movement tracking in a generalized environment, such as emergency vehicle detection. The baseline is a valuable starting point to develop and test a system that can accurately identify and track sound sources.

Overall, the baseline provides a context to develop further a concept of an accurate sound source localization system for outdoor space. The setup can be easily replicated in many other environments, such as other city streets, traffic intersections, etc. This setup is a valuable starting point to develop and test a system that can accurately identify and track sound sources.

3.1.1 Dataset conception

The dataset is the most crucial part of the baseline. We can determine the dataset's characteristics based on the analysis of the section 2.4. The dataset needs to contain the sound recorded by the microphone and the position of the sound source. To simplify the problem, we will use four classes as the main classification challenge in the project. The classes are the following:

- *left_to_right*: The vehicle goes from the left to the right of the microphone.
- *right_to_left*: The vehicle goes from the right to the left of the microphone.
- *no_cars*: No vehicles pass by the microphone.
- *multiple_cars*: Multiple vehicles pass by the microphone.

By adding a camera to the system, we can use the image captured by the camera to determine the ground truth of the sound source's position. The camera's position is the same as the microphone's position, and the camera is facing the road. These classes allow the creation of a dataset without precisely recording the vehicle's position. The *no_cars* and *multiple_cars* are here to ensure we will have a complete dataset, as with these four classes, we can cover every possible scenario recorded by the microphones and don't need to cherry-pick only the recordings that match our classification system.

We also used only two classes at the beginning of the project to ensure the concept's functionality when installing the system. These classes are the following:

- *left_to_right*: The vehicle goes from the left to the right of the microphone.
- *right_to_left*: The vehicle goes from the right to the left of the microphone.

The results and comparison of this task are available in the appendix ??????????.

3.1.2 Real data retrieval system conception

To record real data that suits our baseline

3.1.3 Data recording and storage

3.2 Convolutional Neural Network for Sound Source Localization

3.3 Adversarial Attack conception

3.4 Sound Propagation Simulation

3.4.1 Audio reconstruction from spectrograms

Since the adversarial example is a spectrogram, it needs to be converted back into audio to be recorded again through the microphone. The conversion is done using the Griffin-Lim algorithm[**?**]. The Griffin-Lim algorithm is an algorithm that reconstructs an audio signal from a spectrogram. It is an iterative algorithm that uses the spectrogram to estimate the phase of the audio signal. The algorithm starts with a random phase and iteratively updates the phase until the spectrogram converges to the original spectrogram. The algorithm is defined as follows:

4
Realization

4.1 Simulation model creation

4.1.1 Microsoft Project plugin

4.1.2 Unity plugin

4.2 Dataset creation

4.2.1 Microphone installation

4.2.2 Embedded system setup and access

4.2.3 Data transmission

4.2.4 Data recording and storage

4.3 Neural Network for Sound Source Localization

4.3.1 Dataset preparation

4.3.2 Neural Network architecture

4.3.3 Training

4.4 Adversarial Attack

4.4.1 Fast Gradient Signed Method implementation

5
Results and Analysis

5.1 Sound Propagation Simulation

5.1.1 Dataset Augmentation

5.2 Neural Network model results

5.3 Adversarial Attack results

5.3.1 Adversarial Attack mitigation

6
Conclusions and Future Work

6.1 Conclusions

6.1.1 Specification Fulfillment

6.2 Future Work

6.2.1 Sound Propagation Simulation

6.2.2 Sound Propagation Simulation bachelor's thesis

A thesis named *SimSound3D is*

6.2.3 Advanced adversarial attack

6.2.3.1 Patch attack

6.2.3.2 Targeted attack

6.2.4 Dataset publication

6.2.4.1 Dataset annotation

6.2.5 Loxo Ears model

A
Appendix

A.1

List of Tables

List of Figures

Bibliography

[1] *Pierre-Amaury Grumiaux, Srdjan Kitic, Laurent Girin, and Alexandre Guerin. A survey of sound source localization with deep learning methods.* The Journal of the Acoustical Society of America*, 152(1):107–151, jul 2022.*

[2] *Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In* 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.*

[3] *Shlomo E. Chazan, Hodaya Hammer, Gershon Hazan, Jacob Goldberger, and Sharon Gannot. Multi-microphone speaker separation based on deep doa estimation. In* 2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019.*

[4] *Xiaofei Li, Laurent Girin, Fabien Badeig, and Radu Horaud. Reverberant sound localization with a robot head based on direct-path relative transfer function. In* 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2016.*

[5] *Amengual Garamp. Spatial analysis and auralization of room acoustics using a tetrahedral microphone, Apr 2017.*

[6] *Carlos Fernández Scola and María Dolores Bolaños Ortega. Direction of arrival estimation : A two microphones approach. In* 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.*

[7] *Jiawei Zhang. Gradient descent based optimization algorithms for deep learning models training, 2019.*

[8] *Ch Sekhar and P Meghana. A study on backpropagation in artificial neural networks.* Asia-Pacific Journal of Neural Networks and Its Applications*, 4:21–28, 08 2020.*

[9] *Jürgen Schmidhuber. Deep learning in neural networks: An overview.* Neural Networks*, 61:85–117, jan 2015.*

[10] *Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications.* Neurocomputing*, 234:11–26, 2017.*

[11] *Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015.*

[12] *Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.*

[13] *Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition.* Proceedings of the IEEE*, 86(11):2278–2324, 1998.*

[14] Mariam Yiwere and Eun Joo Rhee. Sound source distance estimation using deep learning: An image classification approach. Sensors*, 20(1), 2020.*

[15] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen. A multi-room reverberant dataset for sound event localization and detection. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, pages 10–14, New York University, NY, USA, October 2019.*

[16] Archontis Politis, Sharath Adavanne, and Tuomas Virtanen. A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, pages 165–169, Tokyo, Japan, November 2020.*

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.