Gunshot Localization in Urban Reflective Scenario

Department of Electrical Engineering

Indian Institute of Technology Bombay

Dual Degree Project - Stage 1 Report

Author: Nandini Bhadoria (20d070055)

Guide: Rajbabu Velmurugan

October 25, 2024

Abstract

Gunshot localization systems are crucial for enhancing public safety and supporting law enforcement by detecting and tracking the sources of gunfire, especially in urban environments. A key challenge in these systems is multipath propagation, where gunshot acoustic signals reflect off surfaces and objects, leading to complex signal delays and distortions. This work delves into the impact of multipath propagation on the accuracy and reliability of gunshot localization in urban settings, presenting a proposed model designed to address these challenges. The model is tested on a complex, real-world dataset to evaluate its performance in reflective urban scenarios. In addition, we explore the model's robustness across various datasets and demonstrates how the approach can be extended to other audio localization tasks, with a particular focus on speech source localization in reverberant environments.

Contents

Chapter 1

Introduction

Gunshot localization, an integral component of public safety and law enforcement, plays a pivotal role in urban environments where the swift identification and tracking of gunfire sources can mean the difference between life and death. This technology relies on the precise determination of the origin of a gunshot using acoustic sensors, thereby enabling rapid response and enhancing security efforts. However, the accurate localization of gunshots in urban settings is a complex task fraught with challenges, one of which is the phenomenon of multipath propagation.

Gunshot localization systems function by employing an array of acoustic sensors to capture the acoustic signature of the gunshot. These sensors measure parameters such as direction-of-arrival (DOA) to calculate the source's location. In an ideal environment, this approach would yield precise results. However, urban settings are far from ideal. The presence of numerous surfaces, structures, and objects introduces the concept of multipath propagation, significantly complicating the process of gunshot localization.

Multipath propagation, occurs when a transmitted signal interacts with the environment, leading to signal reflections and delayed arrivals at the receiving sensors. In the context of gunshot localization, this means that the acoustic waves generated by a gunshot interact with the surrounding buildings, streets, vehicles, and other obstacles. These interactions give rise to a complex soundscape where signals take multiple paths before reaching the sensors, leading to reflections, diffractions, and signal interactions. As a result, the signals received at the sensors are a mixture of direct and reflected waves.

Gunshot localization, though critical for public safety, represents a relatively narrow research domain with specific applications in law enforcement and surveillance. The impact of multipath propagation, where sound waves—whether from a gunshot or another source—interact with surfaces, causing reflections, delays, and signal distortions is a primary concern. These challenges mirror those encountered in a broader and more widely applicable area of research: multi-channel and multipath speech localization.

1.1 Gunshot Signatures

1.1.1 Muzzle Blast

A muzzle blast is an explosive wave created at the muzzle of a firearm during shooting. It propagates at the speed of sound and it lasts for around 3ms. An array of sensors is used to detect the muzzle blast but the sensor is placed far away from the gun source resulting in losses. There is also some background noise associated with the propagation. These factors result in difficulty in the detection of muzzle blast.

The sound pressure resulting from the muzzle blast is strongest in the direction the gun barrel is pointing to. The energy of the sound pulse increases in direct correlation with the volume of gas flow rate (volume velocity) at the source. As the distance from the source increases, this sensitivity to background noise and external sources of interference also escalates. When the muzzle pulse meets a boundary, part of its energy is absorbed and part is reflected. The reflected pulse will have lower energy as compared to direct-path signal due to propagation losses and will have different amplitude-spectrum because its frequency content is not affected the same way throughout the entire bandwidth.

1.1.2 Ballistic Shockwave

Shockwave is a compression wave-generated by sudden increase in pressure followed by a sudden decrease-when the speed of bullet is greater than the speed of sound in air. It is characterized by *Mach Number*, which is the ratio of speed of bullet to speed of the sound in air. The mach number is greater than 1. A cone called *mach cone* is used to represent the shockwave
In our analysis, we have considered the effect of Muzzle Blast only, because not all firearms generate shockwave(only supersonic firearms like snipers produce shockwave significantly). A representation of mach cone for 2 different mach numbers is shown in Figure 1.1.

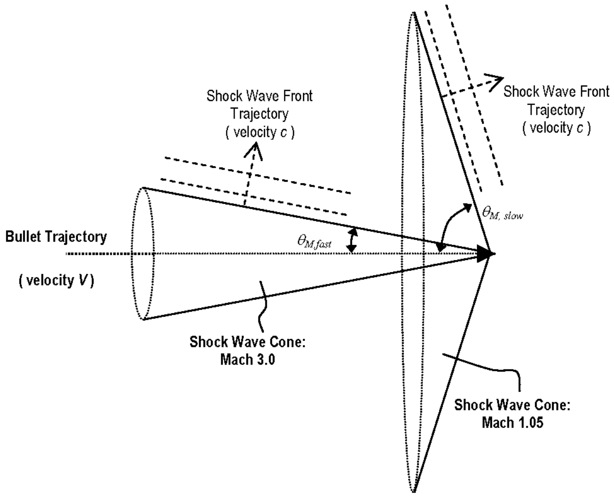

Figure 1.1: Shockwave geometry for a supersonic projectile with Mach number equal to 3 and a slower supersonic with Mach number equal to 1.05

The amplitude v/s time characterstics for a muzzle blast and a shockwave alongwith its spectrogram is shown in Figure 1.2.

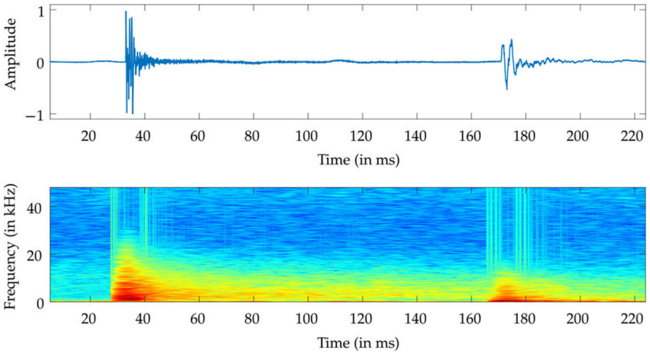

Figure 1.2: Signatures of a Gunshot

Chapter 2

Source Localization

The source can be estimated in many ways. The distributed sensor system typically consists of several microphones or acoustic sensors positioned at known locations in a specific area. Each sensor independently records the sound emitted by the source. When the source emits an acoustic signal, the sound wave travels through the air toward each sensor. Ideally, the signal would take a direct path to each sensor. However, in real-world scenarios, especially in urban or indoor environments, the sound wave can reflect off buildings, walls, objects, and other surfaces. This results in multipath propagation, where multiple copies of the signal, each arriving at different times and angles, are received by the sensors. The shortest route taken by the signal from the source to each sensor is called direct-path or Line-of-sight(LOS) path. The secondary signal path created by reflections off obstacles, which introduce delays and distortions is called reflected path or Non line-of-sight(NLOS) path.

Direction-of-arrival (DoA) estimation refers to the process of retrieving the direction information of gunshot signals from the outputs of a number of receiving microphones that form a sensor array. Direction-of-arrival can be estimated using Lead-Lag-Amplitude Ratio(LLAR) method, Generalised Cross-Correlation Phase Transform(GCC-PHAT) and Delay-and-Sum-beamforming and machine learning based approaches. We'll understand the trilateration based localisation and improve it by using machine learning approaches and later testing the model on a real-world dataset.

2.1 Trilateration based localisation

The trilateration-based approach used for gunshot localization works by estimating the position of the gunshot source using the distances from multiple sensors placed in known locations. The basic principle behind trilateration is to solve for the source's position based on the time delays of the signals arriving at each sensor, which correspond to the distances from the source to each sensor.

In the given scenario, multiple sensors (microphones) are placed at known locations in space and these sensors capture acoustic signals of gunshot. The sound wave from the gunshot takes different times to reach each sensor due to the varying distances between the source and the sensors. The distance between the source and each sensor is calculated using the time delays between them given by **Distance = Time delay x speed of sound**. Using the known positions of the sensors and the estimated distances, a system of linear equations is set up. These equations relate the unknown coordinates of the source to the known coordinates of the sensors and the distances. We are given eight sensors placed randomly in a given space. We need to find the location of the source, given that, it has two multipath reflections associated with it. We need to find the location of source assuming all other conditions to be ideal. We can find the source location using trilateration based approach.Trilateration is a technique used in geometry and navigation to determine the position of a point in space by measuring its distance from three or four known points. It can also be used for determining the gunshot location. First, we group sensors into 2 different groups assuming each group has one 4 different sensors and no two sensors are in both the groups. We also assume that each group has one multipath in it and the source in it, such that no both the multipaths are in different groups.

We first solve for the basic conditon where only 4 sensors and a source is present without any multipath. We can easily solve for this case using trilateration. Let the location of each of the four sensors be (x_n, y_n) where n belongs to sensor S_n, and let the location of the source be (x_s, y_s). There will be circles associated with each of the sensors such that each has its center at (x_n, y_n). There will be some delays associated with each of the sensors due to the source. Let that delay be τ_n with respect to sensor S_n. The radius of the circles can thus be written as $r_n = c\tau_n$, where c is the speed of light. The equation for each of the circles can be written as:

$$(x - x_n)^2 + (y - y_n)^2 = r_n^2 \tag{2.1}$$

for n = 1,2,3,4

$$2x_s(x_1 - x_2) + 2y_s(y_1 - y_2) = (x_1^2 - x_2^2) + (y_1^2 - y_2^2) + (r_1^2 - r_2^2) \tag{2.2}$$

$$2x_s(x_1 - x_3) + 2y_s(y_1 - y_3) = (x_1^2 - x_3^2) + (y_1^2 - y_3^2) + (r_1^2 - r_3^2) \tag{2.3}$$

$$2x_s(x_1 - x_4) + 2y_s(y_1 - y_4) = (x_1^2 - x_4^2) + (y_1^2 - y_4^2) + (r_1^2 - r_4^2) \tag{2.4}$$

This set of equations will give us the solution for the location with least square error.

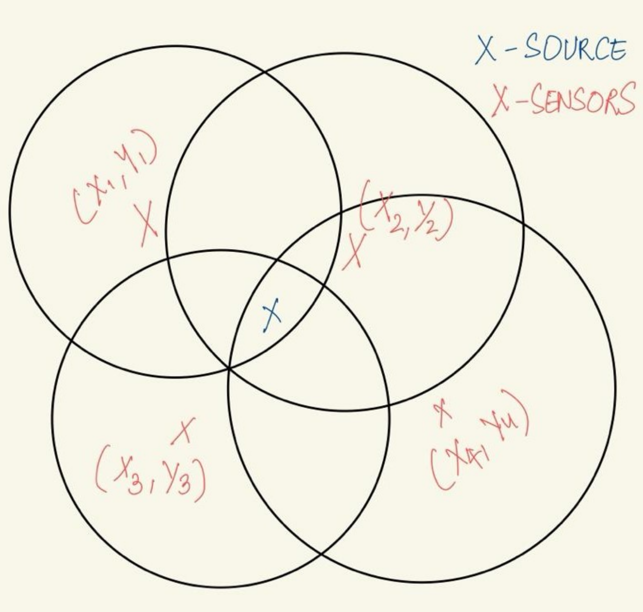

Figure 2.1: Trilateration for finding source location

The concept can be further be extended if the multipath due to the source is also present in the environment. In this case, we consider two multipath reflections associated with the source and eight sensors are used to find the true source locations. All other conditions are assumed to be ideal.

If we assume that the sensor location are known and the delays associated with source and both the multipaths are given for all the sensors, then we can group the sensors and multipath as shown

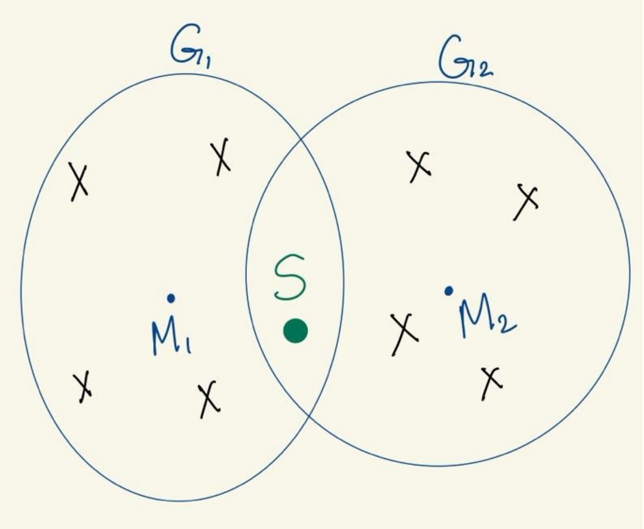

Figure 2.2: Setup to find source location

in figure in Figure 2.2. The groups G_1 and G_2 are made such that four of the eight sensors are in both the groups alongwith with one multipath in each group. The source is assumed to be in both the groups. The time delays associated with the source can be assumed as τ_n. Likewise the delays for multipath M_1 for G_1 is τ_{n_1} and M_2 for G_2 is τ_{n_2}. The radius of sensors is directly proportional to their delays, so each senor will have two radii corresponding to the delays. Now, since we have two radii for each sensor, we will get $2^4 = 16$ source locations for both the groups. The location which coincides for both the group will be our true source location.

Now, if the sensor groups is assumed to be 3, and it is given that the number of sensors per group are 4, then the total number of sensors is equal to 12 in the assumed scenario. One source of gunshot is considered and the number of multipath components associated with are assumed to be 2. The sensors are placed in known positions and are responsible for capturing the direct signal from the source as well as the reflected signals (multipath components). In our scenario, we have assumed that each sensor captures direct signal from source and one reflected depending on the group it belongs, i.e, first group is assumed to receive the direct signal and no multipath signal, the second receives direct signal and reflected signal from first multipath signal(M_1). Similarly, the third group receives direct signal and reflected signal from second multipath signal(M_2). The setup can be seen in figure Figure 2.3.

2.1.1 Simulation

To simulate a realistic scenario, we deploy an array of sensors in a 2D space, receiving both Line of Sight (LOS) signals and Non-Line of Sight (NLOS) signals from the source. We place a total of 12 sensors in a grid configuration, divided into 3 sensor groups with 4 sensors per group. These sensors record the direct signals and the multipath signals reflected off urban objects. The multipath reflection is simulated by generating 2 multipath components per sensor. For each reflection, the angle of incidence and the range are randomly generated to simulate a realistic urban scenario. The true source is located at an unknown position, and the time delays for both the direct and reflected signals are computed based on the speed of sound (343 m/s). The direct distance from the source to each sensor is used to calculate the LOS delays, while additional paths caused by reflections introduce NLOS delays. To simulate real-world conditions, noise is added to the recorded delays and gains using a Gaussian distribution, with a noise power of -50 dB.

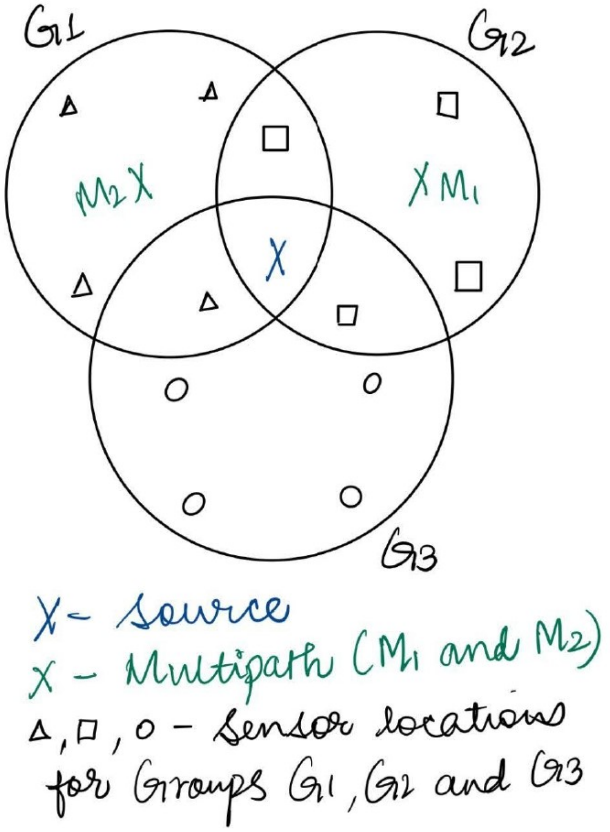

Figure 2.3: Setup to find source location

Trilateration is employed to estimate the source location using the range values derived from the delays at each sensor. The time delays are converted to distances (ranges) by multiplying the delay by the speed of sound. Using the range values from multiple sensors, we apply a **least squares (LS)** method to estimate the source position. This is done by solving a system of linear equations that relate the sensor positions and range values to the unknown source coordinates. The **findpos** function

2.2 Linear Regression Based approach

In addition to the physics-based trilateration approach, a machine learning model is introduced to improve the accuracy of source localization in multipath environments. The setup used is same as in trilateration and 1000 samples are generated, each consisting of randomly selected source locations and their corresponding sensor readings.

The input features to the model consist of the **(x, y)** positions of the sensors and the estimated range values from both the LOS and NLOS signals, which is equal to L+3, where L is the number of multipath components and the value 3 corresponds to number of sensor groups. For each sensor group, we compute the LOS and NLOS delays and use these values to form the feature vector. The model is trained to predict the true source location (x, y) from the sensor readings. The range values are randomized before training to prevent the model from learning a fixed pattern that always selects the same feature for source estimation. The output labels are the true source coordinates (x_s, y_s) which correspond to the actual location of the sound source.

2.2.1 Model Design

For each training sample, the input feature matrix X combines the sensor positions and the range values. The matrix X has the shape:

$$X = [\text{sensor positions } (x, y),\ \text{range estimates } (\text{LOS}, \text{NLOS})]$$

To avoid the model learning a fixed pattern, the range estimates are randomized in the feature matrix. This ensures that the model learns the relationships between sensor readings and the true source location, rather than relying on any fixed ordering of the features. The randomization is done by shuffling the columns corresponding to the range estimates before feeding the features into the model. This improves the model's generalization ability, as it prevents overfitting to a specific configuration of input data.
The linear regression model fits a linear equation to the observed data. The general form of the linear regression equation is:

$$\hat{Y} = X\beta + \epsilon$$

where $\hat{Y}$ is the predicted output (source location), X is the input feature matrix (sensor positions and range estimates) β is the vector of learned coefficients (weights assigned to each feature), ϵ and is the error term, representing the noise or residual differences between the observed and predicted values.
The performance of the model is evaluated using the **Root Mean Squared Error (RMSE)** between the predicted and true source locations. The model's performance is evaluated across different training sample sizes to assess its generalization capability. The RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$$

where n n is the number of samples, Y_i is the true location for the i-th sample, and $\hat{Y}_i$ is the predicted location for the same sample.
During the training process, a large number of samples are generated, each consisting of randomized sensor readings (including LOS and NLOS delays) and the corresponding true source locations. This serves as the labeled dataset for supervised learning. The linear regression model is trained on this dataset by finding the optimal values for the coefficients β. , which minimize the RMSE between the predicted and true source locations. After training, the model is tested on unseen data by predicting the source location from new sensor readings and evaluating its performance using RMSE which provides a measure of how close the predicted source locations are to the true locations. The lower the RMSE, the more accurate is the model. As the number of training samples increases, the RMSE decreases, indicating improved accuracy and robustness of the model. This behavior is plotted as shown in figure Figure 2.5. to observe the relationship between the size of the training set and the model's prediction accuracy.

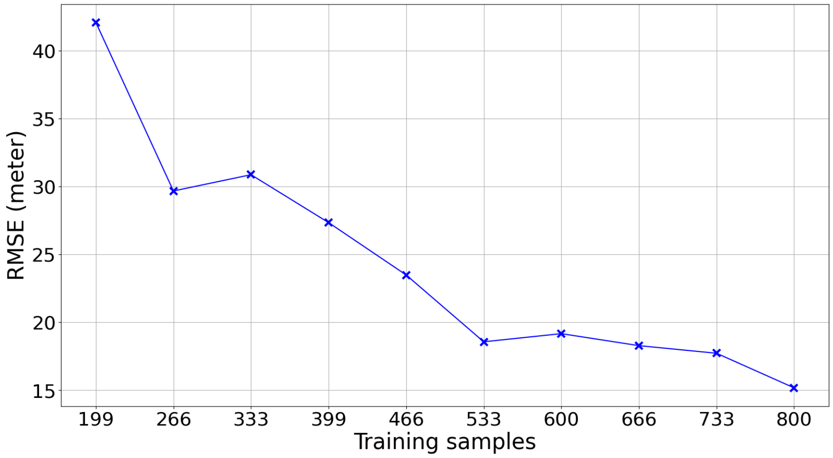

Figure 2.4: RMSE for linear regression model

2.3 CNN Based Approach

The use of CNNs is more effective in case of audio localisation and classification as compared to linear regression models. This is because CNNs are robust to noise and change in audio acoustics due to hierarchical feature extraction. They are effective in capturing spatio-temporal patterns of audio signal by convolving over both time and frequency axes of the audio spectrogram. Linear regression, on the other hand, treats all features equally and independently, without considering the temporal dependencies and spatial structure of the data. In audio localization, the input data includes features from multiple sensors with multipath components. CNNs are efficient in feature learning using layers extracting meaningful information in each layer. They can be combined with signal processing tasks to form a robust model because of their ability to capture complex and non-linear relationship between sound propagation, reflections, and the source location. Linear regression, limited by its linear assumptions, would struggle to model the complexity of these interactions and would lead to poor localization accuracy.

2.3.1 Model Design

To improve upon the proposed linear regression model, a CNN model is introduced which is robust to the given gunshot localisation task. For this the dataset is created first which aligns with the requirements of a real world gunshot data. The dataset consists of sensor readings and source location coordinates. Each sensor reading includes both the direct signal and their multipath components. The feature matrix contains sensor positions and the estimated time delays/ range estimates for LOS and NLOS components. Each sample has dimensions based on the number of sensor elements and multipath components. This matrix is reshaped and input to the model. The reshaped feature matrix is given by:

$$\text{Feature Data} = \text{Reshaped to } (-1, N_{\text{elements}}, L + 3, 1)$$

where -1 is the batch size, N$_{elements}$ is the number is sensor elements which is 12 in the given scenario, L+3 accounts for L = 2 multipath components plus the 3 additional features for LOS signal and other features.

The output is labeled and contains true source location coordinates (x_s, y_s). The CNN model is designed to process the input sensor data (in 2D format) and predict the coordinates of the sound source. The model architecture consists of the following layers:

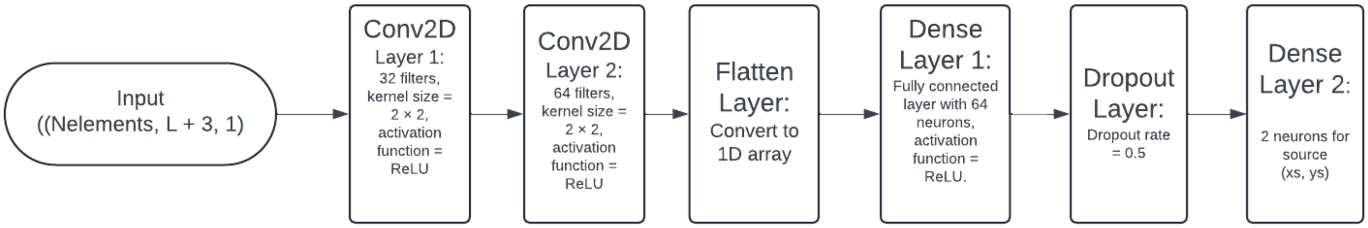
Input
((Nelements, L + 3, 1)
Conv2D Layer 1: 32 filters, kernel size = 2 × 2, activation function = ReLU
Conv2D Layer 2: 64 filters, kernel size = 2 × 2, activation function = ReLU
Flatten Layer: Convert to 1D array
Dense Layer 1: Fully connected layer with 64 neurons, activation function = ReLU.
Dropout Layer: Dropout rate = 0.5
Dense Layer 2: 2 neurons for source (xs, ys)

Figure 2.5: CNN model architecture for source localisation

Here, ReLU activation is used to capture non-linearity between sensors and source data. The model was trained for 10 epochs and was tested by predicting the source locations.

2.3.2 Results and Observations

The model was evaluated on metrics like **RMSE** and **R^2 score** which are given by Equation 2.5 and 2.6 respectively. The final RMSE was computed to be 8.34 meters, indicating that on average, the predicted source location was within this range of true source location. The model achieved an R^2 value of 0.82, suggesting that the model explains 82% of the variance in the true source location. Both these values are suggestive of the conclusion that the model works fine on the given dataset and gunshot localisation scenario. The plots for RMSE and R^2 score at each epoch are shown in figure Figure 2.6 and Figure 2.7. From the figures, it is observed that the RMSE decreases after each epoch and the R^2 score increases which is the ideal case.

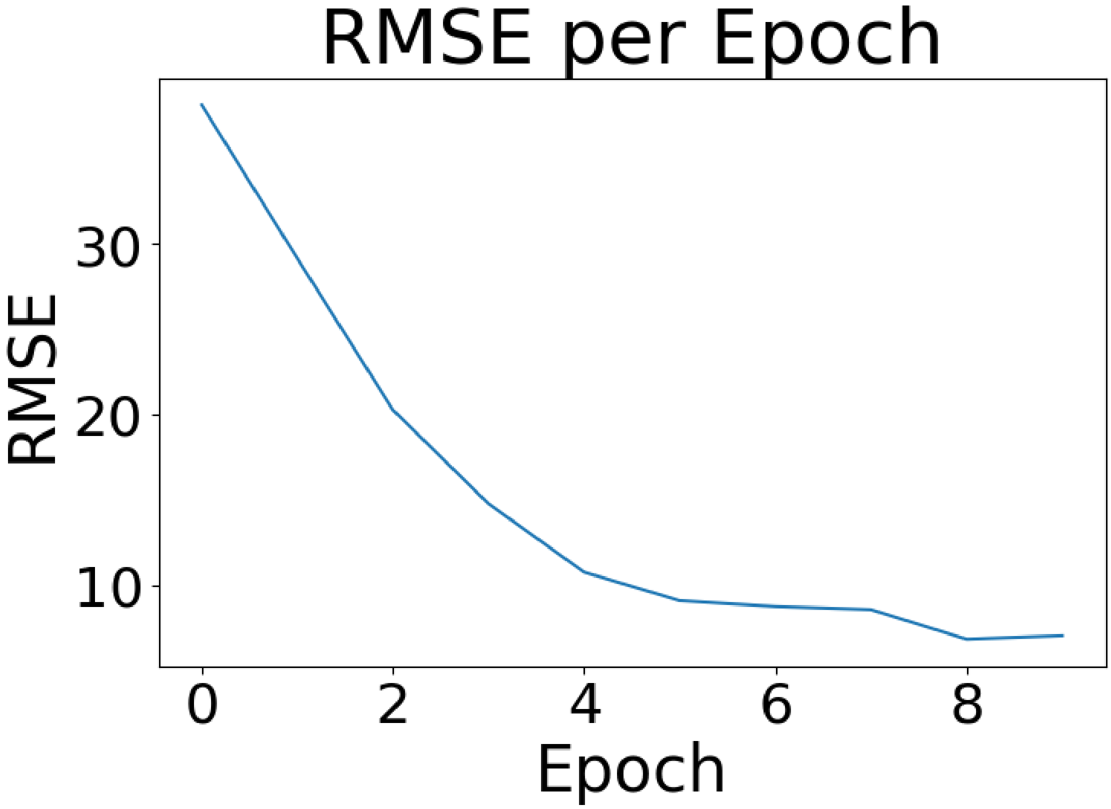

Figure 2.6: RMSE for CNN model

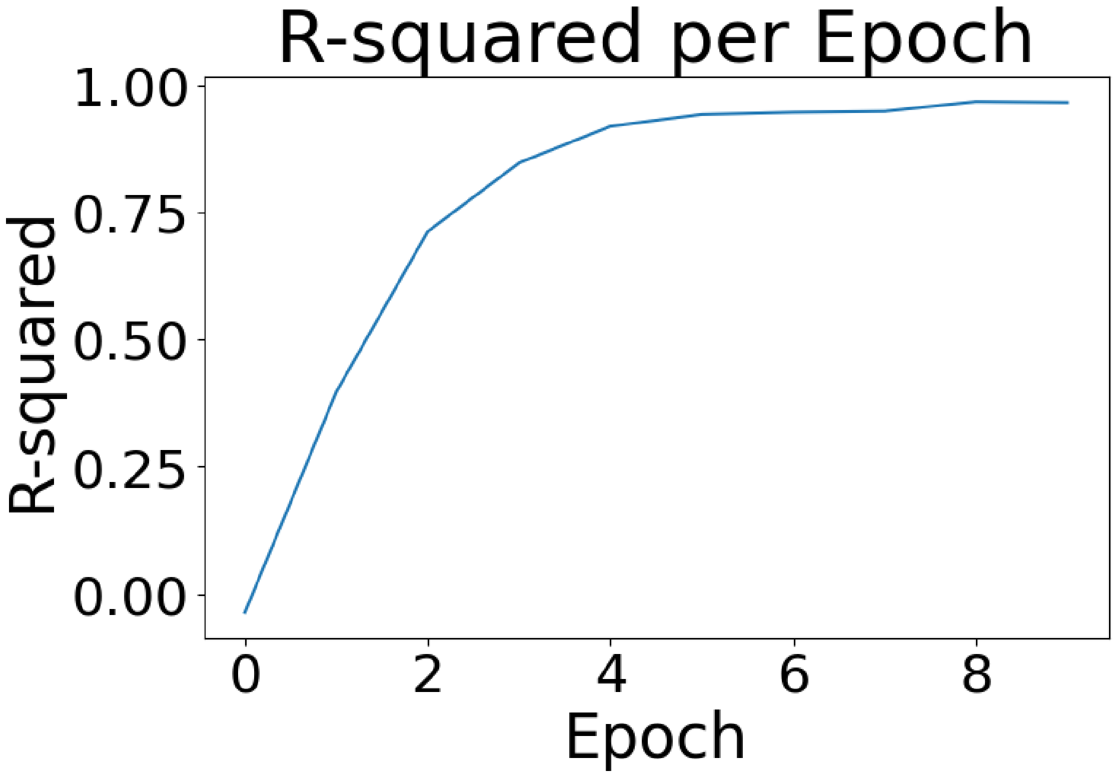

Figure 2.7: R^2 score for CNN model

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{2.5}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{2.6}$$

2.4 Model Testing on a Real Dataset

After testing the model on a synthetic dataset, it is important to test it on a real world data. For this a specific dataset is used which is discussed in later subsections.

2.4.1 Zenodo Gunshot\Gunfire Dataset[1]

The dataset aims to facilitate research in gunshot detection, firearm classification, and Direction of Arrival (DoA) estimation, which are crucial for both military and civilian situational awareness. It consists of gunshot recordings collected in a real-world outdoor environment. It offers gunshot audio files collected under noisy real-world conditions, including background noise like traffic, wildlife sounds, and minimal echoes due to the outdoor setting and it can be used to train machine learning models for DoA estimation in real-world conditions and validate gunshot detection and localization methods.

The given dataset consists of 2,148 audio files, recorded using various edge devices and microphones, a CSV labels file, and a CSV summary file. The former CSV file contains the audio file name, the number of gunshots contained within the file, and the estimated timestamps of gunshot occurrences within the file (in seconds). The latter CSV contains the audio file names, the number of gunshots contained within each file, the make and model of the firearm that fired the gunshots, the time when the device started recording this audio, the manufacturer and model of the device used to record the audio along with a unique identifier for that particular device, the latitude and longitude of the device that recorded the audio, and finally, the timestamps of gunshot occurrences within the file in seconds. Gunshots are treated as having a single corresponding timestamp that was estimated using peak amplitudes. For instance, a recording containing four gunshots contains four corresponding timestamps representing the time audio for that gunshot was first received by the device. The audio files include information about the make and model of the firearm, timestamps for gunshot occurrences, and the location and type of recording devices. The firearms used include an **AR-15 style semi-automatic rifle**, a **Remington 870 shotgun**, a **0.38 caliber revolver**, and a **9mm Glock 17 pistol**. Each firearm was fired in three styles: single shot, multiple shot with time intervals, and rapid fire.
Two sessions were conducted to capture gunshot audio. During session 1, all devices were set to record. Each device and microphone setup was positioned strategically across the firing range. During session 2, four of the devices were set to perform inference to test models that were trained from data collected during session 1. The devices included smartphones and Raspberry Pi systems equipped with different microphones. The setup used in both the sessions is shown in figures Figure 2.8 and Figure 2.9. The range is located with a highway to the west, a body of water surrounded by a wooded area to the east, and law enforcement training environments to the north and south. This setting provided an ideal combination of semi-controlled and real-world conditions for data collection. Its location at an outdoor firing range facilitated the collection of data amidst ambient noise, while its relative seclusion allowed some level of control over the type and amount of noise recorded by each device. The devices included a combination of Smartphones positioned in blue rectangles on the recording setup diagram, Raspberry Pi systems shown in green squares, some equipped with ReSpeaker 2-Mics Pi Hat arrays for multi-channel recording, Wi-Fi hotspots Used to create a mesh network that facilitated data synchronization between devices and allowed remote control of recordings using a mobile application.

Preprocessing was performed on the collected audio to extract only the audio around the gunshot sounds. This was a multi-step process. First, signals were labeled automatically by comparing the "peak" amplitudes of the audio to the Root Mean Square (RMS) of the whole audio file and labeling any point above an empirically determined threshold of 5 times the RMS of the entire file as a gunshot. Once the gunshot "peaks" were found and labeled, an additional step was applied to ensure that only one gunshot was labeled for every 0.25 seconds of audio (the observed average time between multiple shots). The audio was then separated into multiple clips by finding the best window that ensured that every 2 seconds of audio contained at least one gunshot (of length 0.25 seconds). If there was a gap larger than that amount, then the file was separated into multiple audio clips and the data in between was discarded. Each extracted audio clip was saved as its own file in the dataset. Finally, the final preprocessed audio files were manually verified again by a human listener. This extra step was taken to ensure that the final audio files were within an acceptable range of quality, preserving audio files containing real-world noise but excluding audio files where the gunshot was completely obscured by noise. An example of the gunshot event extraction is shown in figure Figure 2.11. In this situation 5 separate gunshots are observed with little ambiguity in where each gunshot begins.
Eight 8-channel circular microphone arrays were used during the recording sessions. Raw audio recorded by the microphone arrays is synchronized to produce multi-channel recordings that can be

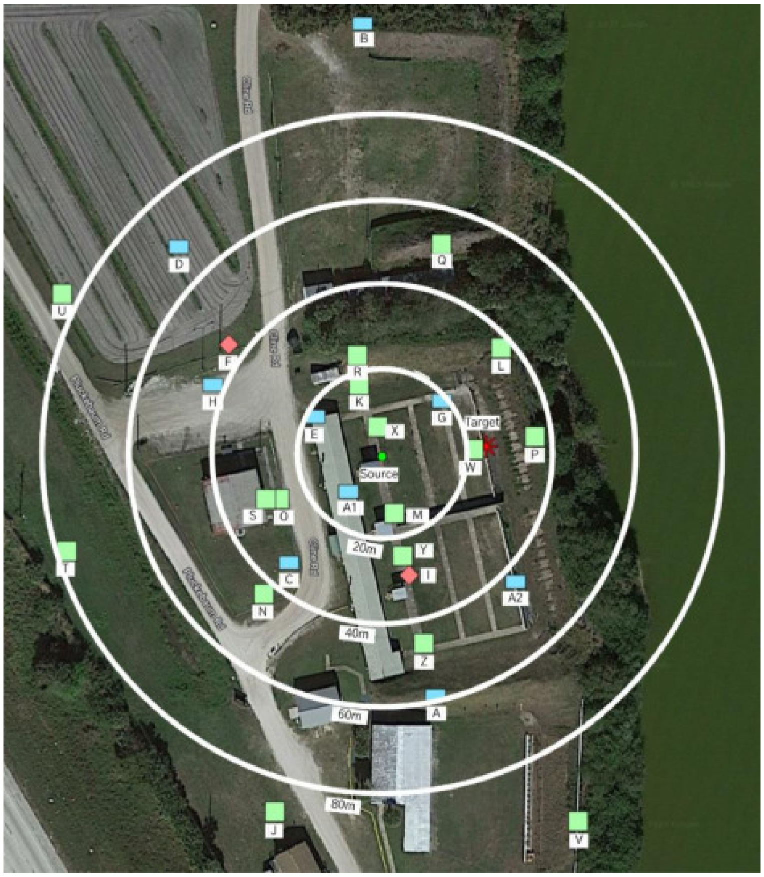

Figure 2.8: Session-1
[1]

used for DoA estimation. The recordings from these microphones are provided in the dataset both as their 8-channel WAV files and also as multiple individual mono WAV files. We have used the mono WAV files for our analysis.

2.4.2 Model Description

The metadata of the dataset is loaded from the CSV file containing information about each audio file, such as its unique identifier, coordinates, and recording device specifications. This data enables the extraction of ground truth location coordinates, which are used as target values for training the model. The function `load_and_preprocess_audio` is used to load each audio file and extract Mel Frequency Cepstral Coefficients (MFCC) features, a widely used feature representation in audio processing. This function uses a smaller window size for MFCC extraction because using a smaller `n_fft` (2048) allows more granular spectral detail in each time frame. Also, the `hop_length` is set to 512, allowing for an adjustable frame overlap and providing a finer temporal resolution of the MFCCs. The maximum length of each audio segment is capped to ensure uniform input shapes for the CNN model. Audio files exceeding this length are truncated, while shorter files are zero-padded. The `load_dataset` function iterates over each audio file in the dataset directory, applying the preprocessing function to generate flattened MFCC feature arrays. This process extracts MFCC features for each audio file and retrieves the latitude and longitude coordinates from the metadata to be used as target labels for training from the metadata using the UUID of the audio file. Finally, the MFCC data is reshaped to prepare for input into the CNN model, where each sample is represented as a 3D array with dimensions (number of MFCC coefficients, max time frames, 1).

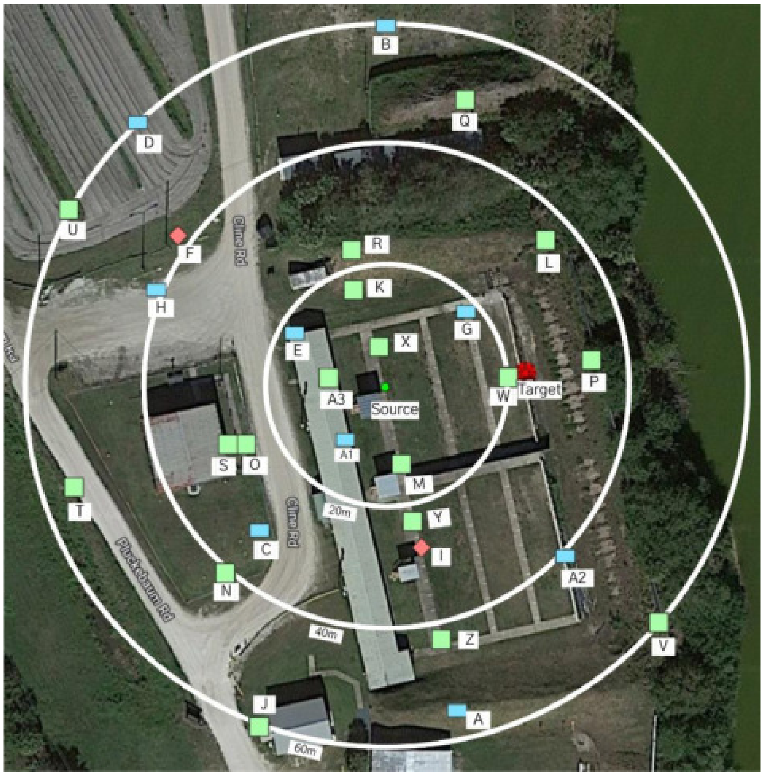

Figure 2.9: Session-2
[1]

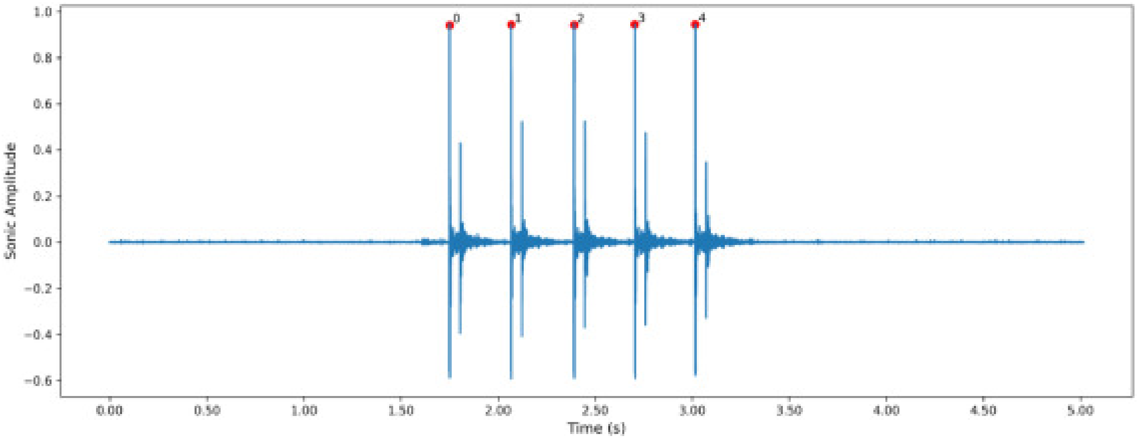

Figure 2.10: Example of a gunshot event extraction
[1]

Model Architecture

The model architecture is defined with the following layers:

- **Input Layer**: Accepts reshaped MFCC data with dimensions $(13, 100, 1)$, where 13 is the number of MFCC coefficients, and 100 is the maximum time frame length.
- **Convolutional Layers**:
  - `Conv2D` layer with 32 filters and a (2×2) kernel, activated by ReLU.
  - A second `Conv2D` layer with 64 filters and a (2×2) kernel, also activated by ReLU.

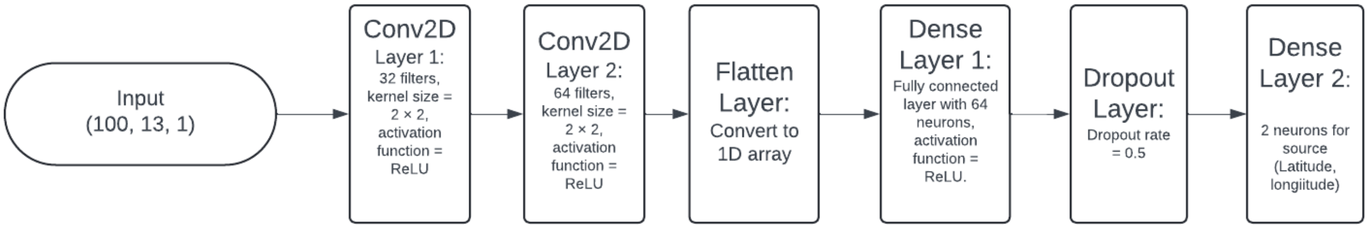

Figure 2.11: CNN model architecture for source localisation

- **Flatten Layer**: Converts the 2D output from the convolutional layers into a 1D array.
- **Fully Connected Layers**:
  - Dense layer with 64 neurons, activated by ReLU.
  - Dropout layer with a rate of 0.5 to mitigate overfitting.
  - Output layer with 2 neurons representing the estimated $(\text{latitude}, \text{longitude})$ coordinates.

The model is compiled using the Adam optimizer and mean squared error (MSE) as the loss function, an appropriate choice for regression tasks where the model outputs continuous values. Prior to training, the input data is normalized to ensure a zero mean and unit variance, which helps improve model convergence and performance. Using `StandardScaler`, the MFCC data is standardized and reshaped back to its original shape. A train-test split is performed to reserve a portion of the data for independent testing, ensuring an unbiased evaluation of the model. To thoroughly evaluate the model's performance and reduce the dependency on any single split of the dataset, **Monte Carlo cross-validation** is implemented. This leverages Repeated K-Fold cross-validation in which the data is split into training and test sets in each fold, with repeated iterations. In this case, 5 splits are used, repeated to reach a total of 1000 simulations. For each fold, the model is trained on a subset of data and tested on an independent subset and predicted coordinates for the test set are stored, along with the the MSE for each fold. This process provides a robust measure of the model's performance across different train-test splits.

The RMSE, derived from MSE, is calculated for each simulation to measure the average prediction error in meters. The RMSE values are averaged over all simulations to obtain the mean RMSE, and their standard deviation provides a measure of performance consistency. The mean and standard deviation of RMSE provide insights into the overall accuracy and stability of the CNN model in estimating source locations. The result is shown in figure Figure 2.12

2.4.3 Classification of firearm on Zenodo Dataset

In addition to localisation, the given dataset can also be used for classification tasks. The dataset used four different types of firearms. The dataset requires preprocessing steps, including, encoding categorical values, such as the uuid column, to numerical values; data cleaning to handle non-numeric values, ensuring that all numeric fields are correctly formatted and; Feature Selection based on relevant metadata fields that contribute to gunshot classification, like `gunshot_location_in_seconds`, `num_gunshots`, and `firearm`. A Random Forest Classifier is used due to its robustness and capability of handling complex, high-dimensional data. The model leverages multiple decision trees to classify the audio recordings, using metadata features for distinguishing gunshots. The model's performance is evaluated using 5-fold cross-validation, ensuring that the model's accuracy is tested across various

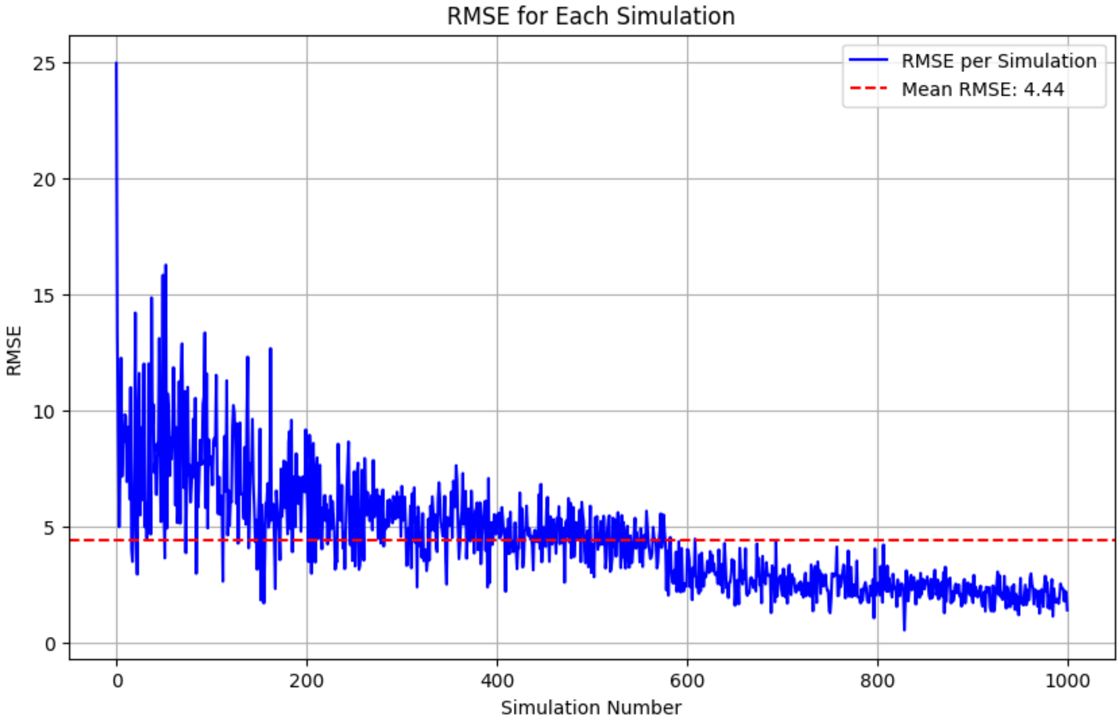

Figure 2.12: RMSE evaluated on the Zenodo Dataset

subsets of the dataset. This method provides a reliable measure of model performance and minimizes the risk of overfitting. The evaluation process resulted in an accuracy of 88.6%. The confusion matrix for the classification is shown in figure Figure 2.13. This shows that the model is able to correctly identify the firearm with an accuracy of 88.6%. The accuracy can be increased further by using more complex CNN models. This can be done in future by integrating the classification task with localisation.

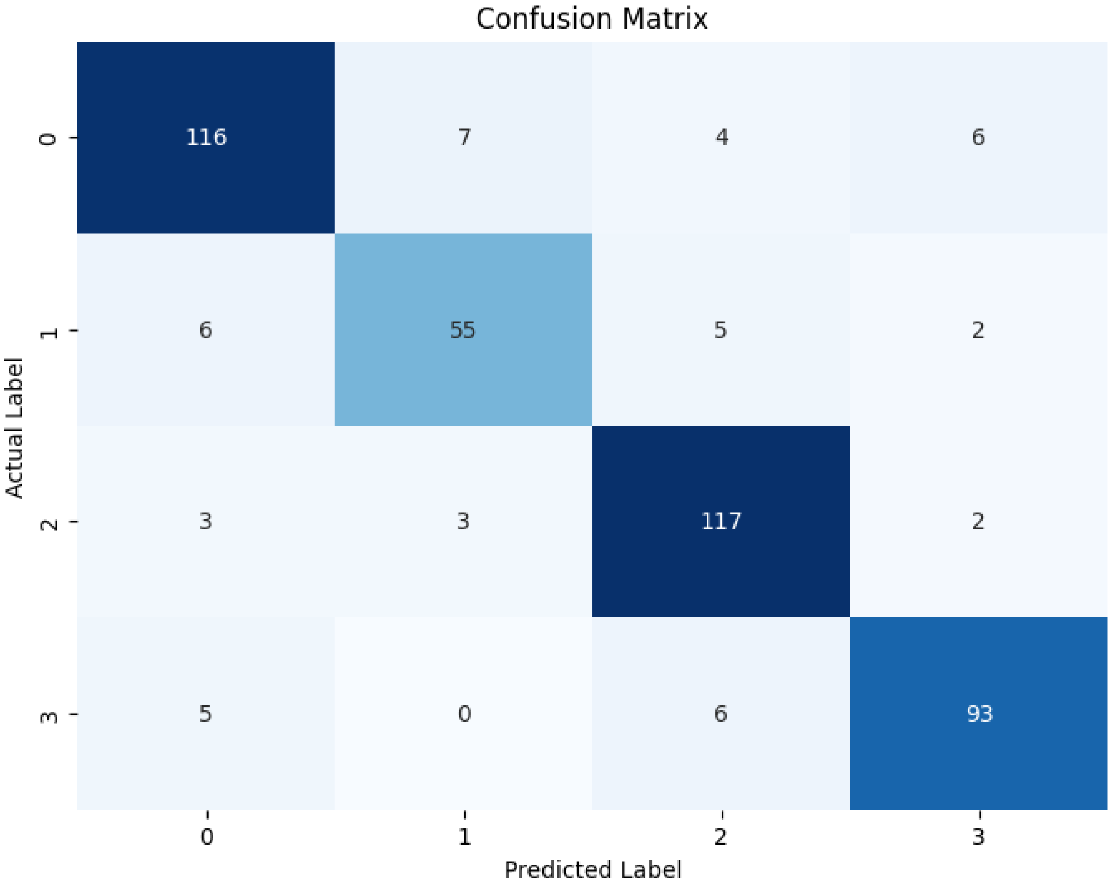

Figure 2.13: Confusion Matrix for classification of Firearms

Chapter 3

Literature Survey

The Gunshot localisation problem is a very narrow domain of research. however it can further extended to classification of generalised audio specifically speech localisation. Researches have been conducted on this problem, we can integrate the problem of localisation in multipath environment into speech with this ongoing research on speech localisation.

3.1 Sound Source Localization Using a Convolutional Neural Network and Regression Model[2]

Sound source localization (SSL) is critical in various applications, such as navigation, human-computer interaction, and surveillance, especially in indoor environments where GPS technology is ineffective. Traditional localization technologies like Bluetooth, Wi-Fi, and infrared each face limitations such as range, line-of-sight requirements, or high installation costs. Sound, however, offers strong penetrative power and can be used efficiently for localization based on acoustic features. This study presents a novel SSL model that combines a convolutional neural network with a regression function (CNN-R) to accurately estimate the sound source angle and distance by processing interaural phase difference (IPD) features extracted from sound signals.

The proposed system leverages CNNs for automatic feature extraction to increase robustness, targeting indoor localization by processing sound signals through a regression model rather than classification, thus supporting continuous value predictions. The study uses the CMU_ARCTIC speech database, combined with simulated room impulse responses (RIR) from the Pyroomacoustics platform and real RIRs from the Multichannel Impulse Response Database. Sound signals are transformed into the time-frequency domain using Short-Time Fourier Transform (STFT) and then processed into IPD feature maps for input to the CNN-R model. The CNN consists of two convolutional layers for feature extraction, followed by a regression model with fully connected layers. This setup reduces overfitting while maintaining high accuracy by utilizing a smaller architecture. The architecture diagram is shown in figure Figure 3.1

Three main experiments evaluate the performance of the CNN-R model in different environments: **single acoustic environment** in which training and testing are conducted in rooms with consistent acoustic parameters, varying only the signal-to-noise ratio (SNR); **multiple acoustic environments** in which the model is trained in varied room configurations, and the testing environment incorporates different reverberation times (RT60) and SNR levels; **real acoustic environment** where experiments are conducted in a real-world setting using real room impulse responses to validate the model's practical application.

The performance metrics used to assess the model's effectiveness include accuracy (Acc.), mean absolute error (MAE), and root mean square error (RMSE). In the simulated environment, the model

achieved an average accuracy of 98.96% in angle estimation at 30 dB SNR and 0.16 s RT60. In real environments, accuracy improved further, reaching 99.85% under similar conditions. The model's accuracy in distance estimation was consistent, with real-environment accuracy reaching 99.38% at 30 dB SNR and 0.16 s RT60. The results are shown in figure Figure 3.2
The CNN-R model outperformed several existing methods, including CNN-SL and CRNN, for angle and distance estimation accuracy in both simulated and real environments, highlighting its capability for precise and robust SSL. The results suggest that CNN-R is highly effective for SSL in both simulated and real environments, achieving high accuracy and low error in favorable acoustic conditions. However, performance declines in high-reverberation scenarios.

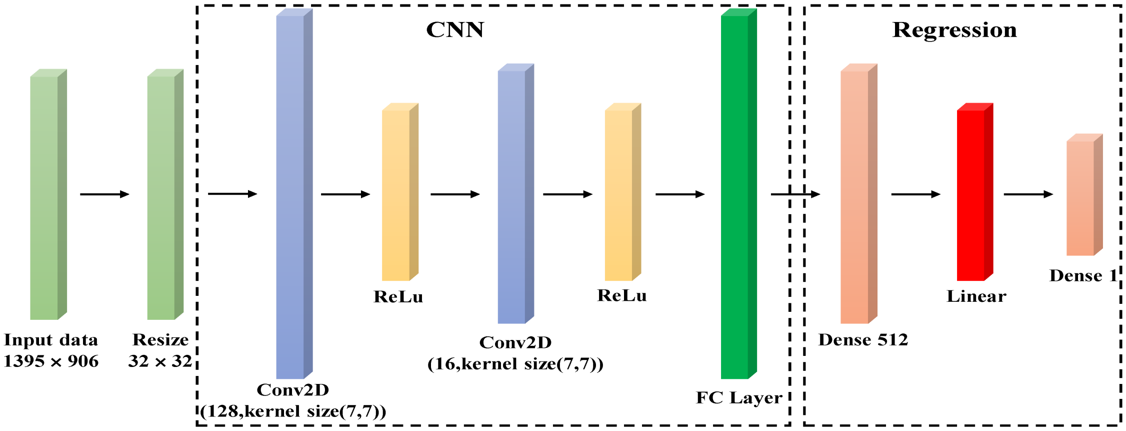

Figure 3.1: The architecture of CNN-R
[2]

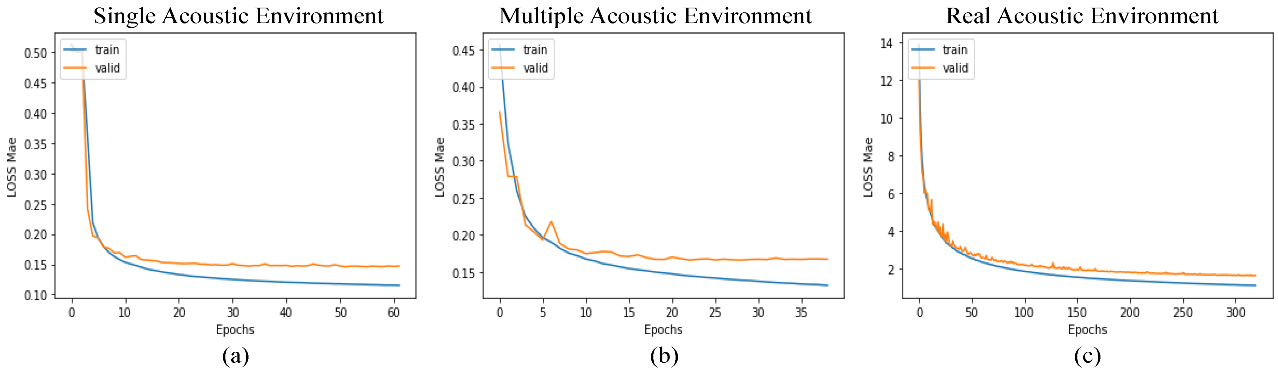

Figure 3.2: he training–validation loss curves of CNN-R, where (a) is the performance of CNN-R in a single acoustic environment. (b) is the performance of CNN-R in a multiple acoustic environment. (c) is the performance of CNN-R in a real acoustic environment.
[2]

3.2 Beamnet: End-to-End Training of a Beamformer-Supported Multi-Channel ASR System[3]

This paper introduces Beamnet, an end-to-end approach for a multi-channel Automatic Speech Recognition (ASR) system that incorporates a statistically optimal beamformer for noise suppression. Unlike conventional ASR models, Beamnet integrates a neural network for mask estimation with an acoustic model, propagating gradients from the back-end acoustic model to the front-end beamformer. This setup allows for training directly on noisy multi-channel data without requiring clean stereo data. The model achieves competitive word error rate (WER) performance on the CHiME 4 dataset.

Beamnet's architecture comprises a mask estimation network that computes time-frequency masks for the GEV beamformer and an acoustic model that processes the enhanced signals to predict phoneme states. Key components include: **Multi-Channel ASR** in which input from multiple microphones undergoes short-time Fourier transform (STFT) to separate speech and noise. The beamformer weights are optimized based on signal-to-noise ratio (SNR) and feature extraction using filterbanks and **End-to-End Backpropagation** in which the model propagates gradients from the ASR objective function (cross-entropy loss) through the GEV beamformer to optimize the entire system jointly.

Using the CHiME-4 dataset with both real and simulated noisy data, the study evaluates Beamnet under various configurations. The experiments aimed to test whether end-to-end training could eliminate the need for clean-noisy parallel data, and results were benchmarked against BeamformIt, a conventional beamformer.

Beamnet demonstrated significant reductions in WER compared to baseline models such that the models initialized with pre-trained components achieved the best WER, while models trained from scratch had slightly worse results. Importantly, the end-to-end trained Beamnet outperformed the conventional BeamformIt and demonstrated the feasibility of training without clean-noisy parallel data.

Beamnet showcases the advantages of end-to-end training for multi-channel ASR, particularly its flexibility with noisy data and independence from specific microphone configurations. The findings suggest that Beamnet can improve recognition performance without requiring clean stereo data, making it applicable in real-world, noisy environments.

3.3 Deep Beamforming for Speech Enhancement and Speaker Localization with an Array Response-Aware Loss Function[4]

This paper addresses the challenge of enhancing and localizing speech in adverse acoustic environments. Leveraging deep neural networks (DNN), the proposed model integrates a novel ARROW (Array Response-Aware) loss function, which utilizes relative transfer functions (RTFs) to jointly improve speech enhancement and speaker localization accuracy. The research focuses on convolutional recurrent networks (CRNs) for this dual-purpose task, showcasing how ARROW loss, combined with scale-invariant source-to-noise ratio (SI-SNR) metrics, can outperform conventional models.

The multi-channel signal model uses an array of microphones to capture both speech and noise signals, which are then transformed into the frequency domain. The proposed model applies a filter-and-sum beamformer that isolates the speech signal, enhancing the target and reducing interference. By computing optimal beamforming weights through CRN-based architecture, the model achieves both enhancement and localization.

The ARROW loss function is designed to target both enhancement and localization by balancing speech clarity and interference reduction. This function combines SI-SNR and ARROW with linear weighting, where SI-SNR reduces noise while ARROW maintains signal integrity. The weighted parameters are optimized to minimize localization errors, particularly in the presence of reverberation and variable room impulse responses (RIRs).

The model was tested using both simulated and real RIRs. Speech data from LibriSpeech and noise samples from MS-SNSD were used, with noise levels ranging from -10 to +15 dB for training and various SNRs for testing. Performance was compared against two baseline models, one solely using

SI-SNR loss and another employing an SPLM (signal processing-based localization module). The proposed model is shown in figure Figure 3.4

Results indicate that the proposed model with ARROW loss achieves superior performance in both speech quality and localization as shown in figure Figure **??**. In speech enhancement, ARROW loss improves overall quality (OVRL) and speech clarity (SIG) at optimal weighting factors. For localization, the model demonstrated a 5% accuracy improvement over baseline methods, maintaining robustness even in high reverberation environments. This demonstrates ARROW's effectiveness in balancing noise reduction and signal preservation.

The deep beamforming system with ARROW loss effectively enhances speech and accurately localizes speakers in complex environments. This joint optimization approach eliminates the need for traditional grid search localization, making it efficient for real-world applications.

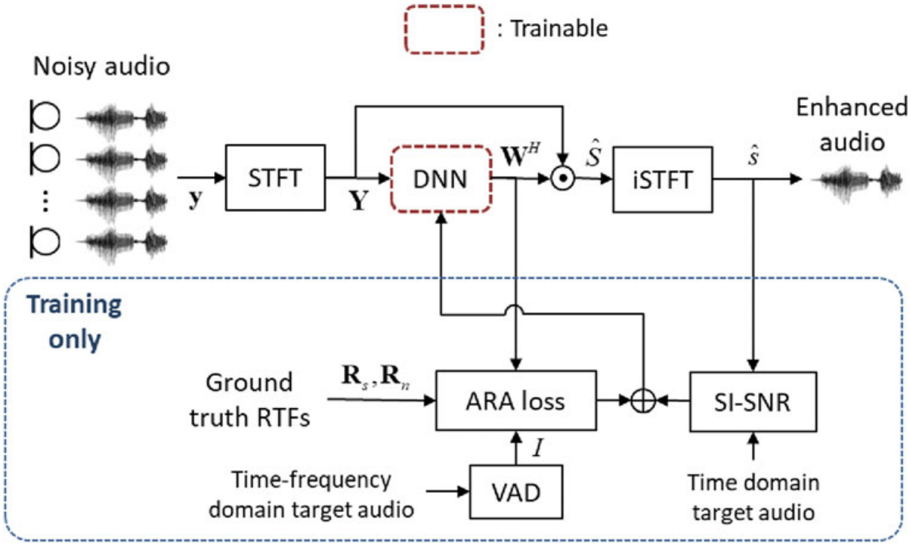

Figure 3.3: The proposed deep beamformer.
[4]

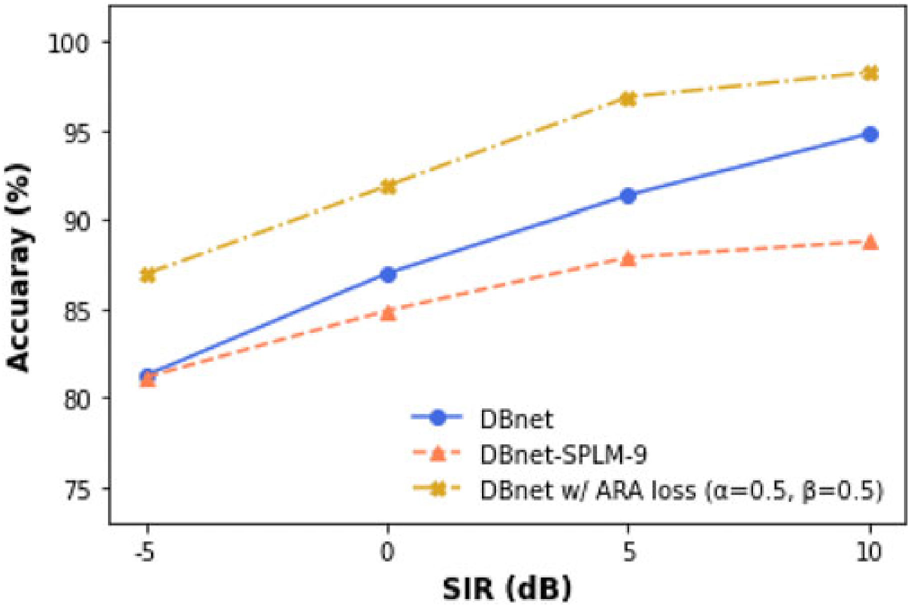

Figure 3.4: Localization performance of the proposed method and the baselines.
[4]

3.4 Towards End-to-End Acoustic Localization Using Deep Learning: From Audio Signal to Source Position Coordinates [5]

The paper presents a novel approach for indoor acoustic source localization (ASL) using convolutional neural networks (CNN) in an end-to-end framework. Traditional ASL techniques rely heavily on hand-crafted features, such as time-delay estimation, beamforming, or spectral estimation, which limit adaptability and often struggle in challenging environments. This research aims to estimate the three-dimensional coordinates of a sound source directly from raw audio input, bypassing traditional preprocessing. The model's strength lies in its ability to generalize across varied acoustic environments without extensive manual feature engineering, making it highly applicable in real-world settings.

State-of-the-Art research primarily divides into three categories. Time-Delay Methods use time-difference-of-arrival (TDOA) but are sensitive to environmental noise and reverberation. Beamforming Techniques, like Steered Response Power-Phase Transform (SRP-PHAT), optimize a spatial statistic but struggle in dynamic or complex sound fields. Spectral-Estimation Methods, such as MUSIC, perform well in ideal settings but are prone to modeling errors and lack robustness under varied conditions. Machine learning, particularly deep learning, has begun addressing ASL, but existing models often rely on extracted features (like spectral data or GCC coefficients) instead of learning directly from raw data. This work is the first to explore end-to-end CNNs trained on raw acoustic signals, which simplifies and potentially enhances localization accuracy.

The CNN model estimates the source's position by learning a function that maps raw signals from multiple microphones to 3D coordinates. The architecture consists of five convolutional layers for automatic feature extraction, learning spatial and temporal patterns directly from the input signals and two fully connected layers to synthesize extracted features and predict precise location coordinates. Given limited real-world ASL data, training is split into two stages. First is semi-synthetic training in which initial training uses semi-synthetic data generated by simulating source signals in various positions. Noise and delay variations mimic real-world acoustic properties, allowing the CNN to learn foundational spatial features. Other is fine-tuning on real data in which final tuning occurs with real recordings, enhancing the model's performance and robustness in actual environments.

Experiments leverage the AV16.3 corpus, which includes synchronized multi-microphone audio in controlled environments. The CNN model is tested using three main sequences, each varying in window size (80 ms, 160 ms, and 320 ms) to examine model sensitivity to temporal granularity. Performance is evaluated using multiple Object Tracking Precision (MOTP), measuring Euclidean distance between predicted and ground truth locations. Relative Improvement in MOTP (%) over SRP-PHAT and a Generative Model-Based Fitting (GMBF) approach, which utilizes sparse constraints to fit a generative model to signal data . Key findings highlight the robustness and adaptability of the proposed CNN model. Baseline Performance: Initial results without fine-tuning show that the CNN underperforms compared to SRP-PHAT and GMBF. However, fine-tuning with real data significantly improves accuracy. Effectiveness of Fine-Tuning: With limited real-world data, the model's performance surpasses SRP-PHAT and approaches GMBF's accuracy. Fine-tuning enables the CNN to adjust to speaker movements, noise variations, and environmental differences; & Generalization: The model exhibits resilience against different speakers, room conditions, and speaker positions, making it adaptable for diverse ASL applications.

The CNN approach provides a robust ASL solution, particularly in environments with minimal labeled data. Fine-tuning on real data allows the model to adapt to specific room characteristics and audio distortions, enabling it to maintain high accuracy. While performance in complex environments remains an area for improvement, the CNN-based method shows promise for real-time ASL systems that require minimal setup and calibration.

3.5 Multiple Sound Source Localization in Three Dimensions Using Convolutional Neural Networks and Clustering Based Post-Processing [6]

This paper introduces an innovative method for three-dimensional multiple sound source localization (SSL) by leveraging convolutional neural networks (CNN) in conjunction with a clustering-based post-processing approach. SSL plays a vital role in numerous fields, including surveillance, navigation, and human-machine interaction. Conventional SSL systems often require large and distributed microphone arrays, which can be challenging to deploy and maintain. To address this, the authors propose a more streamlined solution utilizing a single tetrahedral microphone array capable of capturing spatial cues from sound sources within a 3D space. This compact system is designed to overcome the limitations of traditional SSL algorithms like Steered Response Power-Phase Transform (SRP-PHAT) and Generalized Cross-Correlation with Phase Transform (GCC-PHAT). Instead of conventional methods, the proposed model uses phase spectrum components derived from Short-Time Fourier Transform (STFT) frames as input features for training the CNN, aiming to achieve efficient 3D localization of sound sources.

The methodology begins by formulating SSL as a classification problem, where each possible position of a sound source within a 3D grid is treated as a distinct spatial class. The input feature extraction involves using the STFT phase component from the microphone signals, which are processed into 3D matrices encoding spatial probabilities. This approach allows the model to handle sound sources as continuous locations rather than relying solely on directional estimates. The CNN architecture designed for this task includes three 2D convolutional layers for capturing spatial features from the phase input, followed by several fully connected layers that produce a 3D output matrix. This matrix represents the probability of a sound source existing at each point within the grid. A unique aspect of this model is the clustering-based post-processing technique applied to the CNN output. K-means clustering is utilized to refine the output by identifying clusters within the 3D probability matrix, thereby pinpointing the positions of one or multiple sound sources with greater accuracy.

To evaluate the model, the authors conducted experiments using both simulated and real-world data collected from tetrahedral microphone arrays. They systematically varied parameters such as spatial resolution (Q) and Gaussian kernel spread to understand their impact on localization accuracy. The training and testing datasets included both semi-synthetic data, generated using the pyroomacoustics simulation toolkit, and real-world recordings. This hybrid approach allowed the researchers to test the model's effectiveness in controlled acoustic environments as well as in practical, real-world settings. The simulations provided a controlled setup for model training, while real recordings allowed the researchers to observe the model's performance in live environments with natural reverberation and noise.

The results indicate that the proposed CNN model with clustering-based post-processing achieves high localization accuracy in both single- and multiple-source scenarios. In single-source localization, the CNN and clustering approach yielded mean absolute error (MAE) values of less than one meter, showcasing its precision. In cases involving two sound sources, the clustering method accurately localized both sources with an average MAE of around 1.08 meters. The clustering algorithm, used to enhance the CNN output, significantly improved accuracy, providing a 31% reduction in MAE compared to a method that simply identified the maximum value in the output matrix. Furthermore, the model effectively estimated azimuth, elevation, and distance of the sound sources. Although the model's azimuth and distance estimates were accurate, elevation estimation posed challenges, particularly when sources were near the enclosure's boundaries. This reduced accuracy was attributed to the tetrahedral microphone configuration, which has a single vertically non-coplanar microphone, limiting the model's vertical spatial resolution.

The results demonstrate that CNNs can effectively learn to map raw phase components to three-dimensional spatial positions, particularly when supported by clustering-based post-processing. This model provides a practical and cost-effective SSL solution, eliminating the need for large and complex microphone arrays. Additionally, the method's adaptability and precision make it a promising option for various SSL applications in domains that require portability, such as security monitoring and human-computer interaction. However, the model's performance is challenged in highly reverberant environments and when sound sources are positioned close to the boundaries of the space. Future work could explore advanced clustering algorithms, incorporate sound source separation techniques, and improve localization under changing acoustic conditions to address these challenges.

This study presents a novel approach to 3D SSL using CNNs with STFT phase components as input and clustering-based post-processing. The model achieves precise sound source localization with a compact, single-array setup, reducing the complexity associated with traditional distributed microphone arrays. The system's accuracy in both simulated and real environments illustrates its potential for broad application across different SSL use cases, from surveillance to interactive systems. This research marks a significant step forward in SSL technology, demonstrating that CNNs can provide robust localization with minimal preprocessing and equipment.

Chapter 4

Summary and Future work

We discussed about the problem of gunshot localisation in urban environments, focusing on challenges related to multipath propagation, where gunshot sounds reflect off surfaces, causing delays and signal distortion. The gunshot signatures, including muzzle blasts and ballistic shockwaves were explained, and methods for source localization using trilateration, linear regression, and CNNs were discussed. We then discussed model testing on real-world datasets, specifically the Zenodo gunshot dataset, and evaluated firearm classification performance. The literature review compares relevant studies in sound localization and how the ongoing gunshot localisation problem can be expanded to speech localisation.

For the future, the problem of localisation can be expanded to speech signals which is a correlated audio. Other than speech, the localisation task can be generalised to all types of audio. Alongwith it, sound classification and localisation can be combined for a better performance. The existing model can also be tested on varying datasets, other than the one used. In addition to speaker localisation, the problem of problem of speaker diarisation can also be looked into in which the different speakers present in the same environment or setup and differentiated and evaluate the performance of different models in multipath and noisy environment.

Bibliography

[1] Ruksana Kabealo and Steven J. Wyatt, "Gunshot/gunfire audio dataset," Aug. 2022.

[2] Tan-Hsu Tan, Yu-Tang Lin, Yang-Lang Chang, and Mohammad Alkhaleefah, "Sound source localization using a convolutional neural network and regression model," *Sensors*, vol. 21, no. 23, 2021.

[3] Jahn Heymann, Lukas Drude, Christoph Boeddeker, Patrick Hanebrink, and Reinhold Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel asr system," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5325–5329.

[4] Hsinyu Chang, Yicheng Hsu, and Mingsian R. Bai, "Deep beamforming for speech enhancement and speaker localization with an array response-aware loss function," *Frontiers in Signal Processing*, vol. 4, 2024.

[5] Juan Vera-Diaz, Daniel Pizarro, and Javier Macias-Guarasa, "Towards end-to-end acoustic localization using deep learning: From audio signal to source position coordinates," 07 2018.

[6] Saulius Sakavičius, Artūras Serackis, and Vytautas Abromavičius, "Multiple sound source localization in three dimensions using convolutional neural networks and clustering based post-processing," *IEEE Access*, vol. PP, pp. 1–1, 01 2022.