An artificial neural network based method for grid-free acoustic source localization using multiple input frequencies

Erik ten Oever

Delft University of Technology

TU Delft

An artificial neural network based method for grid-free acoustic source localization using multiple input frequencies

by

Erik ten Oever

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 01-04-2022.

Student name:	Erik ten Oever
Student number:	4741382
Date:	01-04-2022

Thesis committee:	Prof. dr. G.C.H.E. de Croon
	Prof. dr. ir. M. Snellen
	Ir. C. de Wagter

TU Delft

Acknowledgements

This thesis was written to obtain the MSc degree in Aerospace Engineering at the TU Delft. The research was conducted within the Control and Simulation department of the Faculty of Aerospace Engineering. During the thesis period I studied the potential of an artificial neural network based method for acoustic source localization. I am thankful for the opportunity to study this subject.

This thesis research was supervised by Guido de Croon. I would like to express my gratitude for his guidance during the research process. I would also like to thank Mirjam Snellen for her guidance in acoustic imaging. Before I started on this project I had little to no knowledge about this subject. Mirjam has guided me in obtaining a sufficient level of knowledge about acoustic imaging.

Finally, I would like to thank my family and friends for their support during the thesis period.

Erik ten Oever
Delft, March 2022

Contents

List of Figures

List of Tables

Introduction

In recent years the use of both commercial and consumer unmanned aerial vehicles ($UAVs$) has been increasing at a staggering pace. Due to the increase of UAV applications, the number of hazardous situations involving UAVs also increased [61]. A large part of these hazardous situations are caused by interference with other air traffic.
Several measures can be applied to minimize the interference with other air traffic. Many of these measures are so-called cooperative systems. These systems require both the UAV and the aircraft to be equipped with compatible systems [62]. Popular systems are the Traffic alert and Collision Avoidance System ($TCAS$) and Automatic Dependent Surveillance-Broadcast ($ADS-B$). However, many general aviation aircraft are not equipped with these systems.
A second category of systems that can be used for detect and avoid are known as non-cooperative systems. Non-cooperative systems only require the UAV to be equipped with a detection module. One of these non-cooperative systems is an air-to-air radar [49]. However, air-to-air radar systems are large and power consuming, making them physically infeasible or economically impracticable for many UAV applications. Therefore, efforts have been focused on finding lightweight, energy-efficient sensors suitable for non-cooperative autonomous detect and avoidance systems. Optical sensors have often been applied for obstacle detection on UAVs. Popular techniques that utilize an optical sensor are stereo vision [35], optical flow [8], and in recent years image classification using artificial neural networks [26]. However, optical sensors are less effective in several weather conditions, such as heavy rain and fog [66].
In recent years, research has been done in the use of microphones for detect and avoid. Microphones are lightweight, energy-efficient, and are able to receive information from all directions. A system that uses one or multiple microphones is referred to as a hear-and-avoid system. A hear-and-avoid system consists of two main aspects: acoustic event recognition and acoustic source localization. Acoustic event recognition is the task of detecting the presence of other air traffic based on their sound. Acoustic source location is the task of localizing other air traffic with respect to the UAV based on their sound. Earlier research has been done into acoustic event recognition and ego-noise suppression [57, 60].

This research is focused on the acoustic source localization aspect of hear-and-avoid. The aim is to study the potential of an artificial neural network based method for acoustic source localization. The model is trained to locate acoustic sources on a 2 dimensional plane.
This report consists of three main parts. The first part contains of the scientific paper. The scientific paper introduces the methodology used during this research. The methods are then applied on both simulated and recorded data. The second part contains the literature review. The literature review is both focused on sound event recognition and acoustic source localization. Part 3 contains appendices that support the findings during this research.

I

Scientific Paper

An artificial neural network based method for grid-free acoustic source localization using multiple input frequencies

Erik ten Oever*, Mirjam Snellen, Guido de Croon†
Delft University of Technology

ABSTRACT

In recent years, efforts are focused on developing an acoustic based autonomous detect and avoidance system for UAVs to minimize interference with other air traffic. The purpose of this research is to study the potential of artificial neural networks for fast, grid-free acoustic source localization. A multi-layer perceptron has been trained to localize simulated white noise acoustic point sources using a converted version of the cross spectral matrix. The ANN based method shows similar localization behaviour to different frequencies as conventional beamforming. A new ANN architecture is proposed that uses the converted cross spectral matrices of multiple different frequencies as input to improve the localization accuracy. The multi input model has shown to have a mean absolute error of approximately 0.27[m]. The proposed model has also been applied on real world recording data of an aircraft flyover. The ANN based method has shown to be able to obtain a prediction within approximately 0.05[s], compared to approximately 1000-2000[s] for conventional beamforming. However, the magnitude and inconsistency of the localization error for the recording is higher compared to the simulated white noise sources.

1 INTRODUCTION

In recent years the use of both commercial and consumer unmanned aerial vehicles ($UAVs$) has been increasing at a staggering pace. Due to the increase of UAV applications, the number of hazardous situations involving UAVs also increased [1]. A large part of these hazardous situations are caused by interference with other air traffic.
Several measures can be applied to minimize the interference with other air traffic. Many of these measures are so-called cooperative systems. These systems require both the UAV and the aircraft to be equipped with compatible systems [2]. Popular systems are the Traffic alert and Collision Avoidance System ($TCAS$) and Automatic Dependent Surveillance-Broadcast ($ADS-B$). However, many general aviation aircraft are not equipped with these systems.
A second category of systems that can be used for detect and avoid are known as non-cooperative systems. Non-cooperative systems only require the UAV to be equipped with a detection module. One of these non-cooperative systems is an air-to-air radar [3]. However, air-to-air radar systems are large and power consuming, making them physically infeasible or economically impracticable for many UAV applications. Therefore, efforts have been focused on finding lightweight, energy-efficient sensors suitable for non-cooperative autonomous detect and avoidance systems.
Optical sensors have often been applied for obstacle detection on UAVs. Popular techniques that utilize an optical sensor are stereo vision [4], optical flow [5], and in recent years image classification using artificial neural networks [6]. However, optical sensors are less effective in several weather conditions, such as heavy rain and fog [7].
In recent years, research has been done in the use of microphones for detect and avoid. Microphones are lightweight, energy-efficient, and are able to receive information from all directions. A system that uses one or multiple microphones is referred to as a hear-and-avoid system. A hear-and-avoid system consists of two main aspects: acoustic event recognition and acoustic source localization. Acoustic event recognition is the task of detecting the presence of other air traffic based on their sound. Acoustic source location is the task of localizing other air traffic with respect to the UAV based on their sound. Earlier research has been done into acoustic event recognition and ego-noise suppression [8, 9]. Wijnkers et al. have shown promising results in detecting an acoustic event caused by air traffic using a convolutional neural network. Van der Woude et al. have developed a method to suppress the UAV's ego-noise.

The next step for the hear-and-avoid system is to be able to localize the acoustic event caused by air traffic. Several different techniques for acoustic source localization already exist. One of the most popular techniques is conventional beamforming [10]. Conventional beamforming is an acoustic

*Msc student
†Supervisor

imaging technique that estimates a measure for the acoustic source level on a 2-dimensional scanning plane. On the beamform plot, acoustic sources are indicated by higher acoustic source levels. Calculating the sound levels for all grid points on the scanning plane is very computational demanding. Therefore, convolutional beamforming is less suitable for real time application in hear-and-avoid.
In recent years, artificial neural networks have rapidly become popular due to their non-linear modelling capabilities and low computational effort during application of a trained model [11]. Artificial neural networks have shown to be successful in computer vision and natural language processing [12, 13, 14, 15]. Also in the field of acoustic source localization, artificial neural networks have become more popular.
Kujawski et al. have applied a residual network to estimate the acoustic source coordinates using a phased microphone array [16]. The network uses a low resolution beamform plot as input and is able to predict the acoustic source coordinates with a higher accuracy than the grid resolution of the beamform plot. As this method still requires the creation of a beamform plot, it is less suitable for real time applications.
Castellini et al. propose a different method for acoustic source localization using artificial neural networks [17]. The method uses a multi-layer perceptron network to estimate the acoustic source coordinates. The network uses a converted version of the cross spectral matrix as input. The cross spectral matrix is a matrix containing measured phase information. By using a converted version of the cross spectral matrix, computational effort of preprocessing the input data is low.

This paper presents a study of the potential of an artificial neural network based acoustic source localization method suitable for a hear-and-avoid system. Methods for acoustic source localization often show different behaviour for different input frequencies. During this research the behaviour of the method proposed by Castellini et al. to different input frequencies is tested. Also a new artificial neural network model is proposed that combines the information of multiple frequencies.
The structure of this paper is as follows. Section 2 of this paper discusses the theory applied during this research. The working principle of conventional beamforming and the artificial neural network based for acoustic source localization proposed by Castellini et al. are explained in this section. Section 3 will explain the methodology followed during this research. The section explains how data has been generated and how the artificial neural network based models are applied. In this section a new model is proposed that uses more input information. The results obtained using the methodology from section 3 are presented in section 4. After the results possible influencing factors and improvements are discussed in section 5. Finally, section 6 will conclude the findings of this research.

2 Related Work

This research studies the capabilities of artificial neural networks for sound source localization. The artificial neural network based method uses similar input data as conventional beamforming, currently one of the most popular methods for acoustic source localization.

2.1 Conventional Beamforming

Conventional beamforming (CB) is one of the most popular acoustic source localization techniques, due to it being intuitive and robust [10]. CB is an acoustic imaging technique. Using CB, an estimate for the acoustic source levels can be calculated on a predefined scan plane at a distance parallel to the microphone array. The microphone array is used to capture sound in the time domain. Using the Fourier transform, the recorded time domain signals of each microphone are converted to the frequency domain. The frequency domain data of a specific frequency is stored in data vector $P(f_k)$, as shown in equation 1. The frequency data for a specific frequency for the n^{th} microphone is denoted by $p_n(f_k)$.

$$P(f_k) = \begin{bmatrix} p_1(f_k) \\ p_2(f_k) \\ \vdots \\ p_N(f_k) \end{bmatrix} \tag{1}$$

Using data vector P, the cross spectral matrix (CSM), denoted by C, can be created. The CSM is created by multiplying data vector P with its complex conjugate transpose, denoted by P^*. As a result of equation 2, the CSM is complex valued with size $N \times N$.

$$C(f_k) = P(f_k) \cdot P^*(f_k) \tag{2}$$

The scan plane of which an image is to be made is divided into multiple grid points. As the microphone is a predefined setup, the locations of all microphones with respect the center point of the microphone array are known. Using this knowledge, the distance from each microphone to a certain grid point can be calculated using equation 3. Equation 3 calculates the distance from the n^{th} microphone to the j^{th} grid point.

$$r_{n,j} = \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2 + (z_n - z_j)^2} \tag{3}$$

For each of the grid points a steering vector ($G_j(f_k)$) is constructed. This steering vector contains expected phase information for the location of the grid point at a specific frequency. Equation 4 is used to determine the different elements of the steering vector for each of the N microphones.

$$g_{n,j}\left(f_k\right) = \frac{e^{-2\pi i f_k \frac{r_{n,j}}{c}}}{r_{n,j}} \tag{4}$$

The CSM and steering vector can then be used to determine the beamform output for each of the grid points. Usually, when creating a beamform plot equation 5 is used. However, during this research equation 6 is used. Equation 5 increases the beamform output for locations further from the source. This research is focused on determining the source location. Therefore, equation 6 is used, as this equation puts more emphasis on matching the phase transition over the microphone array. Calculating the beamform output for all grid points results in a beamform plot containing the pressure levels on the scan plane.

$$B_j\left(f_k\right) = \frac{G_j^* C G_j}{\left\|G_j\right\|^4} \tag{5}$$

$$B_j\left(f_k\right) = \frac{G_j^* C G_j}{\left\|G_j\right\|^2} \tag{6}$$

With CB beamform plots can be made of different frequencies of interest. However, different frequencies show different behaviour in the beamform plot. Lower frequencies show a larger spreading of the main lobe around the sound source, making it more difficult to accurately locate a sound source. Higher frequencies show a smaller spreading of the main lobe, but show more side lobes in other areas of the scan plane, again making it more difficult to locate the sound source. These issues can be resolved by a technique called incoherent averaging [18]. For incoherent averaging separate beamform plots are made for frequencies in a frequency range of interest. The pressure levels of the separate plots are summed and the total is divided by the number of frequencies used, as shown in equation 7. The resulting plot shows a smaller spreading of the main lobe and a reduced amount of side lobes elsewhere in the scan plane.

$$B_{incoh} = \frac{1}{N_f} \sum_{k=1}^{N_f} B\left(f_k\right) \tag{7}$$

CB is a robust and intuitive acoustic source localization method. However, due to the iterative process over each of the grid points and creating images for multiple frequencies it is too slow for a real-time application such as hear-and-avoid.

2.2 Artificial Neural Networks for acoustic source localization

In recent years, artificial neural networks ($ANNs$) have become a popular techniques to solve complex tasks, such as image recognition and natural language processing. An ANN is in essence a function approximator that learns how to map inputs to a corresponding output. A popular type of ANN is known as a multi-layer perceptron [19]. A multi-layer perceptron consists of multiple layers: the input layer, the hidden layers, and the output layer. Each of these layers consists of multiple neurons. The neurons in the input layer and output layer are equal to the number inputs and outputs, respectively. The neurons in the hidden layers and output layers contain an activation function. An activation function is a simple non-linear function. The combination of these non-linear functions gives the ANN the ability to learn complex non-linear relations between input and output. The input of an activation function is a weighted sum of the outputs of the activation functions in the previous layer. These weights are changed during training to minimize a loss function. A loss function is used to quantify the error of the ANN's output predictions. The type of loss function is dependent on the application of the ANN. Training of the network is done using a process called backpropagation. The training process is guided by the optimizer. The optimizer is an algorithm that supervises the weight changes to minimize the loss function. Training an ANN can be computationally expensive and time consuming. However, ones an ANN has been trained it is able to quickly produce an output.

The ANN-based acoustic source localization technique used in this research is based on the work of Castellini et al. Castellini et al. developed a regression based method that directly predicts the x- and y-coordinate of the acoustic source location on the scan plane [17]. An important aspect in using ANNs is to establish a suitable input feature containing sufficient information to perform the task at hand. The CSM, discussed in section 2.1, contains sufficient information to be used as input feature. However, an issue arises when directly using the CSM. The elements of the CSM are complex valued, forming a problem to ANNs. Therefore, Castellini et al. propose the use of a converted version of the CSM that contains real valued elements. The first step to creating the converted CSM is to set the main diagonal of the CSM to 0. The CSM contains phase difference information between the received signals of different microphones in the array. The main diagonal of the CSM contains phase difference information between a signal received by a microphone and itself. This difference should be 0, therefore, the main signal should be set to 0. The next step is to convert the complex valued CSM to a real valued version. An important property of the CSM is that it is Hermitian in nature. Using this knowledge, the real and imaginary parts can be combined as follows: the upper right triangular part of $Re\left(C\right)$ becomes the upper right triangular part of the converted CSM , and the upper right triangular part of $Im\left(C\right)$ becomes the lower right triangular part of the converted CSM. For N microphones the resulting converted CSM is a real valued $N \times N$ matrix with its main diagonal set to 0. The conversion process to create the converted CSM is visualized in figure 1. Before the input feature is fed into the network it is flattened, giving the input layer a size of N^2 neurons for N microphones. The hidden layers

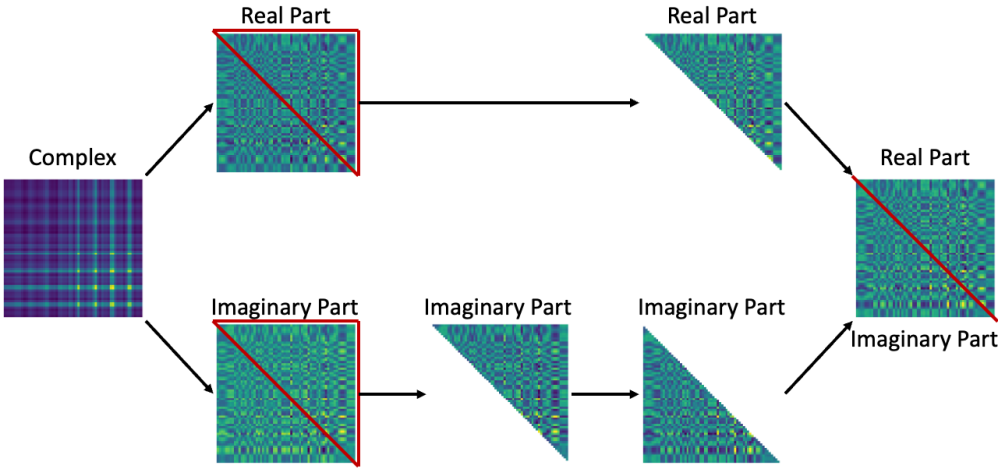


Figure 1: Conversion of CSM as described by Castellini et al.

of the network are fully connected and make use of the rectified linear units ($ReLu$) activation function, described by equation 8. The ReLu activation function is an often used activation function for the hidden layers due to its learning stability [20].

$$f(x) = \max(0, x) \tag{8}$$

The model directly outputs a prediction for the x- and y-coordinate of the sound source. As these coordinates can both be positive and negative, the neurons in the output layer make use of a linear activation function.

3 Methodology

3.1 Data Generation

UAVs that rely on a system such as hear-and-avoid mostly operate in the airspace used by general aviation aircraft, as these aircraft often lack cooperative collision avoidance systems. Measurements during this research have shown that the sound originating from these aircraft is mostly present between 0[Hz] to 3000[Hz]. Therefore, this research is focusing on frequencies within this range. Different frequencies within this range are used during this research. To obtain as much information as possible from the frequency range it is split into 10 evenly spread frequencies. The frequencies of interest during this research are: 300[Hz], 600[Hz], 900[Hz], 1200[Hz], 1500[Hz], 1800[Hz], 2100[Hz], 2400[Hz], 2700[Hz], and 3000[Hz]. For this research a scan plane of 100[m]x100[m] is created at a distance of 50[m] from the microphone array. The microphone array used consists of 64 different microphones. The configuration of the microphone array is shown in figure

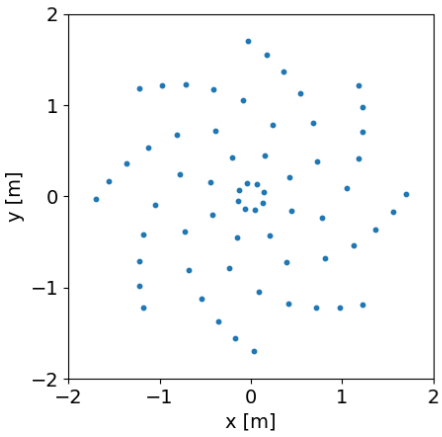


Figure 2: Microphone array configuration used during this research.

2. This is an initial study into acoustic source localization with neural networks. In order to employ this method on a UAV, a smaller microphone array will be necessary.

For neural network training, a large number of examples are required. Therefore, synthetic white noise is used during this research, instead of actual aircraft recordings. To create an example, a synthetic white noise signal of 0.5[s] is created with a sample frequency of 40,000[Hz]. This synthetic white noise signal is emitted from a point source that is located on the scan plane. The point source is placed on the scan plane with 0.1[m] accuracy. Knowing the location of the point source and each of the microphones in the array, a delayed

signal can be created for each of the microphones. In this manner the received signal for each of the microphones is simulated. The simulated received signals are in the time domain. By applying the fast Fourier transform algorithm the signals are converted to the frequency domain. Knowing the frequency information per frequency bin, the information belonging to the frequencies of interest can be extracted. For each frequency of interest a converted CSM is created with the steps described in section 2.2. These converted CSMs are standardized to have a mean of 0 and a standard deviation of 1. Standardizing the input feature improves stability during the training process of the ANN. In total 250,000 different examples are created. Each of these examples has its own point source location and synthetic white noise signal. Of the 250,000 examples 160,000 are used for training, 40,000 for validation, and 50,000 for testing. The synthetic white noise signals and corresponding converted CSMs were created using python 3.9.

3.2 Single Input Frequency

The method proposed by Castellini et al. takes a converted CSM of one frequency as input. Conventional beamforming has shown that different frequencies show a difference in accuracy for acoustic source localization. Here we study how an ANN-based method behaves to being trained on different input frequencies. The frequencies used in this test are: 600[Hz], 1200[Hz], 1800[Hz], 2400[Hz], and 3000[Hz]. The ANN model is based on the model proposed by Castellini et al.,described in section 2.2 The ANN architecture used is shown in figure 3 and further described in table 1. The input layer of the network uses 4096 neurons due to the microphone array consisting of 64 microphones, as the input layer is 64^2. During training the mean squared error (MSE) loss function is used. The MSE loss function is often used for a regression based ANN model. The optimizer used during training is the Adam optimizer with a learning rate of 0.001 [21]. The training data is fed into the network in batches of 128 examples. After each epoch the training data is shuffled to create new batches to obtain better generalization to the training data and prevent overfitting. The model is trained for 500 epochs. After training, the model of the epoch with the lowest validation loss is saved. The ANNs created during this research have been created using Keras with the TensorFlow backend [22, 23].

3.3 Multiple Input Frequencies

The model used in section 3.2 uses the converted CSM of a single frequency to make a prediction of the acoustic source location. Incoherent averaging of beamform outputs has shown that combining information of multiple frequencies yields more accurate localization results. This section proposes an ANN-based model that takes multiple converted CSMs of different frequencies as input. The model architecture used for multi-frequency acoustic source localization is shown in figure 4. The network first uses separate channels to process the converted CSM of a single frequency, this includes phase difference information between different microphones. After three hidden layers in each of the separate channels, the channels are combined in a concatenate layer. The configuration of one of the input channels is given in table 2. All input channels have a similar configuration. After the concatenate layer, the remainder of the network is similar to the hidden layers and output layer described in table 1.

Using the ANN architecture visualized in figure 4, two different models were trained. The first model takes the converted CSMs of 5 different frequencies as input. These frequencies are: 600[Hz], 1200[Hz], 1800[Hz], 2400[Hz], and 3000[Hz]. The second model takes the converted CSMs of 10 different frequencies as input. These frequencies are: 300[Hz], 600[Hz], 900[Hz], 1200[Hz], 1500[Hz], 1800[Hz], 2100[Hz], 2400[Hz], 2700[Hz], and 3000[Hz].

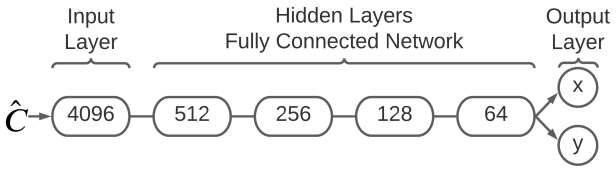


Figure 3: Single Input ANN model with indication of number of neurons for each layer.

Table 1: ANN architecture for single frequency acoustic source localization.

Layer	Neurons	Activation Function	Layer Type
1	4096	-	Input
2	512	ReLu	Fully Connected
3	256	ReLu	Fully Connected
4	128	ReLu	Fully Connected
5	64	ReLu	Fully Connected
6	2	Linear	Output

Table 2: Configuration of an input channel used in the multi input frequency model.

Layer	Neurons	Activation Function	Layer Type
1	4096	-	Input
2	512	ReLu	Fully Connected
3	256	ReLu	Fully Connected
4	128	ReLu	Fully Connected

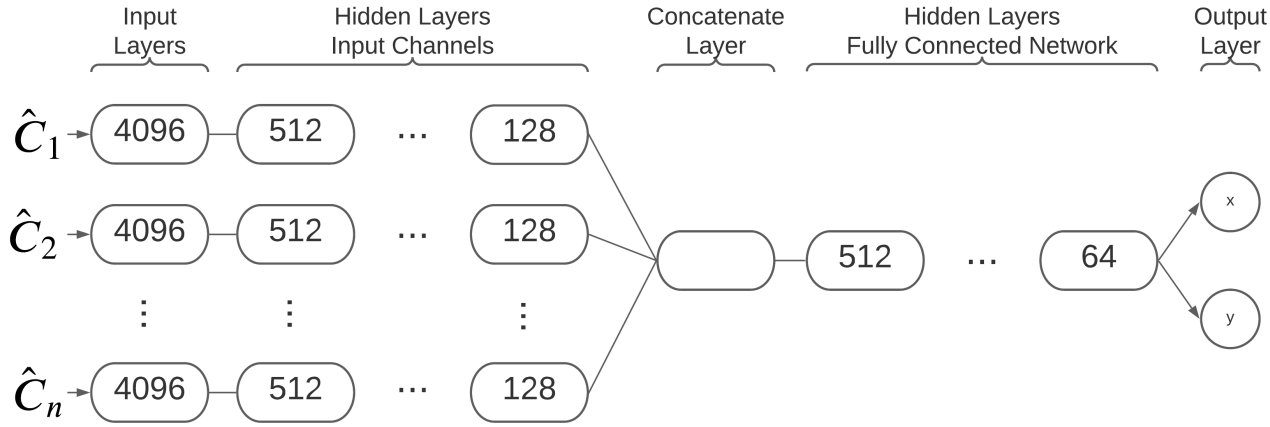


Figure 4: Multi Input ANN model. For 5 input frequencies n=5. For 10 input frequencies n=10.

The training procedure of these models is similar to the training procedure descibed in section 3.2.

3.4 Flyover Recording Data

The data generated in section 3.1 has been simulated without any background noise. For the large amount of data needed to train an ANN it is more efficient to simulate data compared to gathering recorded data that corresponds to the scanning plane. Still it is interesting to see how a network that has been trained using simulated synthetic white noise without other background noise copes with a real world recording. For this, the models created in section 3.3 are being presented with data that originates from a recording of an aircraft flyover. These recordings were made by Wijnkers et al. at Lelystad Airport [8]. The microphone array configuration used for these recordings is identical to the one shown in figure 2.

4 RESULTS

4.1 Single Input Frequency Results

The ANN based model from section 3.2 has been trained on five different input frequencies to get insight on the model's behaviour to different frequencies. After training, a test set consisting of 50000 examples has been fed to the model. The examples in the test set are white noise signals with source locations the model has not encountered during training. The prediction accuracy of the model is quantified using the mean absolute error and the standard deviation of the prediction error distribution. The mean squared errors and the standard deviation of the prediction error distributions are summarized in table 3. The mean absolute errors shown in table 3 are supported with figures 5 and 6. Figures 5 and 6 show the distribution of prediction errors of the x- and y-coordinate, respectively.

Table 3: Mean absolute error and standard deviation of the source location prediction errors for different input frequencies and average of separate models. Lower is better.

Input frequency	x MAE	x error std	y MAE	y error std
600	0.8982	1.5304	0.8949	1.5352
1200	0.5523	0.992	0.5422	1.0002
1800	0.4532	1.008	0.4527	0.989
2400	*0.4389*	0.9446	*0.4363*	0.9795
3000	0.4537	*0.9436*	0.4535	*0.9702*
Average	**0.3173**	**0.5282**	**0.3178**	**0.5307**

Looking at the mean absolute prediction errors in table 3 it can be seen that the average prediction error of all frequencies falls within one meter, stating that the ANN based model was able to find a relation between the converted CSM and the source location. Comparing the results of the different input frequencies, a resemblance can be seen to the behaviour of conventional beamforming. For lower frequencies, the mean absolute error decreases with increasing frequency. This is due to the decreasing size of the main lobe around the source with increasing frequency of interest. This is supported by a decreasing standard deviation in the prediction error distribution. From conventional beamforming it is known that the occurrence of side lobes increases with increasing frequency. The influence of these side lobes result

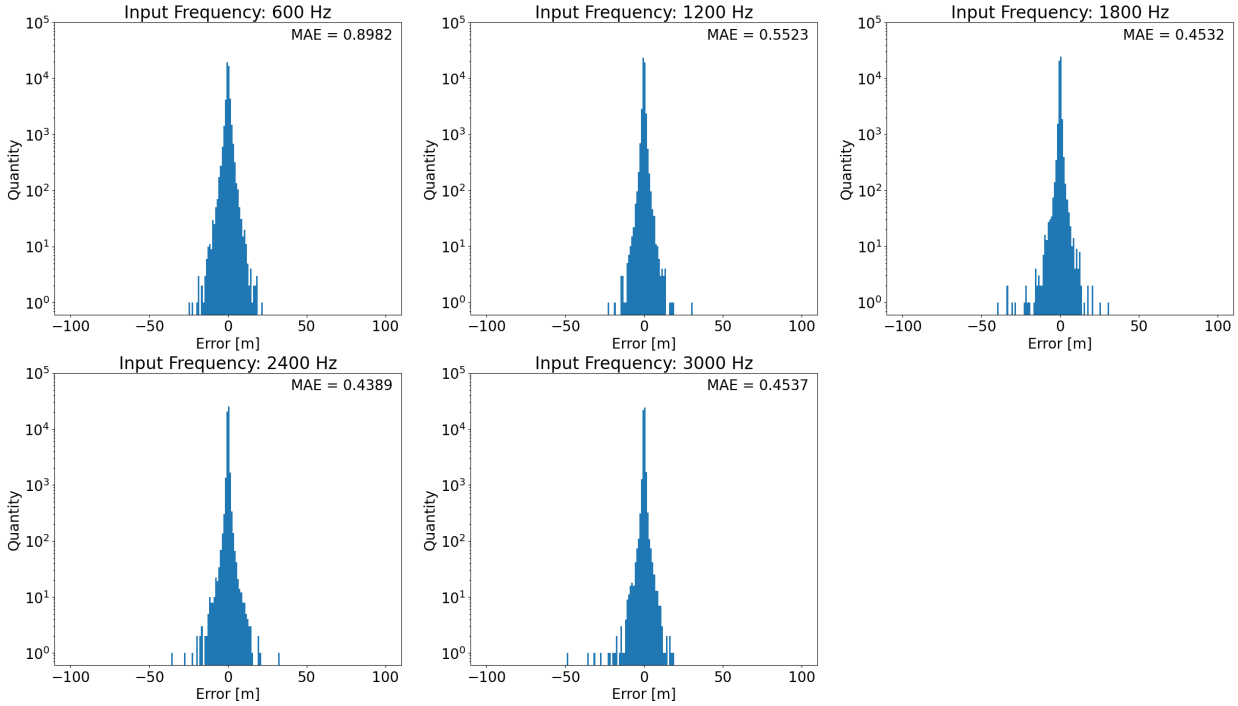


Figure 5: Prediction error distribution of x-coordinate.

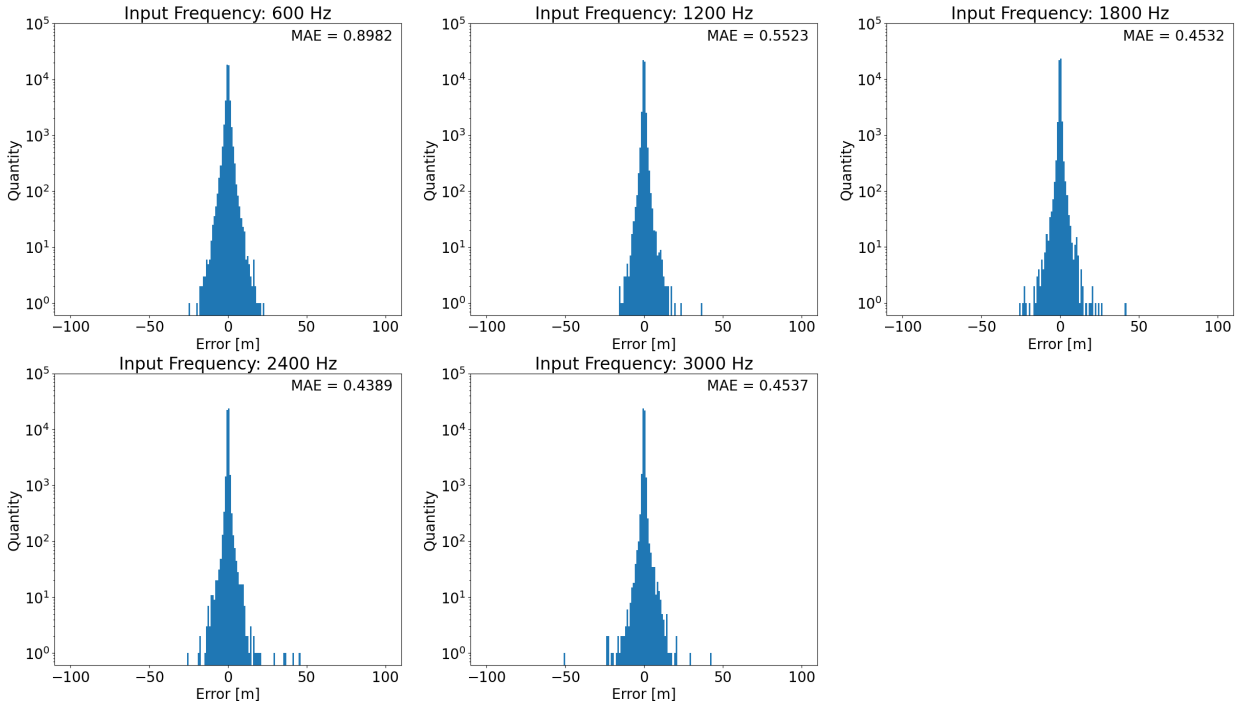


Figure 6: Prediction error distribution of y-coordinate.

in an increasing mean absolute errors of 3000[Hz] compared to the mean absolute error of 2400[Hz]. This is supported by figures 5 and 6, with increasing frequency it can be seen that the standard deviation of the error distribution decreases. Figures 5 and 6 also show that as the input frequency increases the presence and magnitude of larger errors due to side lobes increases.
Conventional beamforming has shown that applying incoherent averages reduces the size of the main lobe around the source and reduces the amount of side lobes. A straight forward approach for applying incoherent averaging in the ANN based method is to average the predicted source locations of the separate models. The bottom row of table 3 shows a drastic accuracy increase compared to the separate models. However, a hazard of this approach is that instabilities in one of the separate models will have an impact on the final prediction accuracy.

4.2 Multiple Input Frequencies Results

In section 3.3 a new ANN architecture was proposed that takes the converted CSMs of multiple frequencies as input and produces one prediction for the source location. Using the multi input ANN architecture a model was made that takes as input the same 5 frequencies as the separate models in section 4.1. Table 4 shows the results of the 5 input model. The mean absolute error and standard deviation show a significant decrease compared to the models discussed in section 4.1. Figure 7 also shows a decrease in larger errors caused by side lobes.

Table 4: Mean absolute error and standard deviation of the source location prediction errors for the 5 input model and 10 input model.

Model	x MAE	x error std	y MAE	y error std
5 input frequencies	**0.2681**	0.458	**0.2678**	**0.429**
10 input frequencies	0.2799	**0.4434**	0.2807	0.4638

The ANN architecture used to create the 5 input model has also been used to train a model that takes 10 different frequencies as input. The prediction accuracy achieved by the 10 input model is quite similar to that of the 5 input model. The prediction accuracy of the 10 input model is summarized in table 4 and the prediction error distributions are shown in figure 7. Figure 7 shows that the error distribution of the 5 input model and 10 input model are very similar. In table 4 it can be seen that the prediction accuracy of the 10 input model has slightly decreased compared to the 5 input model. This decrease in accuracy might be due to the addition of 300[Hz] to the input frequencies, as lower frequencies are often harder to localize.

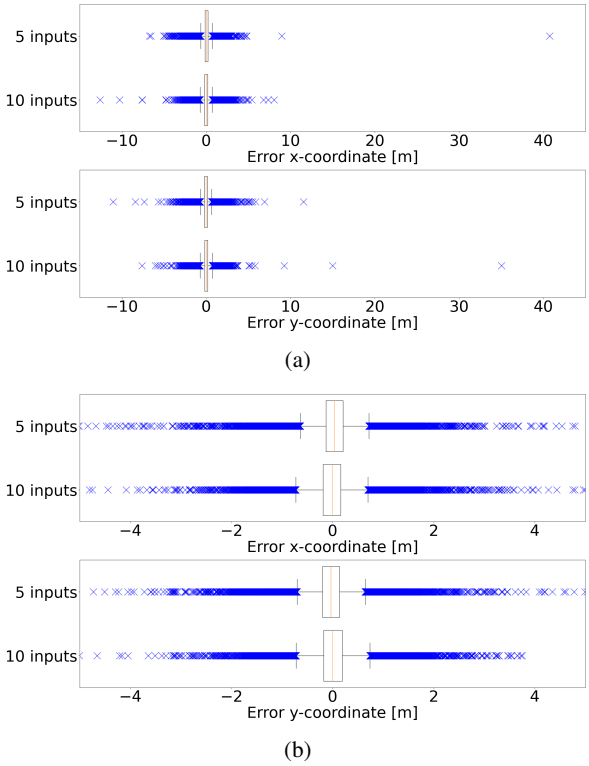


Figure 7: (a) Shows the prediction error distribution of the 5 input model and 10 input model of all prediction. (b) Shows the prediction error distribution of the 5 input model and 10 input model, zoomed in at the prediction errors within [-5,5].

4.3 Flyover Results

4.3.1 Single frame localization

Section 4.2 has shown that using 5 or 10 input frequencies shows little difference in prediction accuracy when using synthetic white noise. This section shows the performance of these models, which are trained on synthetic white noise, on real aircraft flyover recordings. The exact location of the aircraft during the flyover is unknown. Conventional beamforming is used to obtain an indication of the aircraft's position. A beamform plot is created at a distance of 50[m]. As the exact height of the aircraft is unknown, the beamform plot does not depict the exact location of the aircraft. This is due to a mismatch between the measured CSM and the expected phase differences in the steering vector. Figure 8 shows the beamform plots for 5 different frequenties: 600[Hz], 1200[Hz], 1800[Hz], 2400[Hz], 3000[Hz].

From the beamform plots in figure 8 several differences with synthetic white noise can be deduced. First, each of the frequencies does not originate from the same source location. The synthetic white noise training data uses point sources.

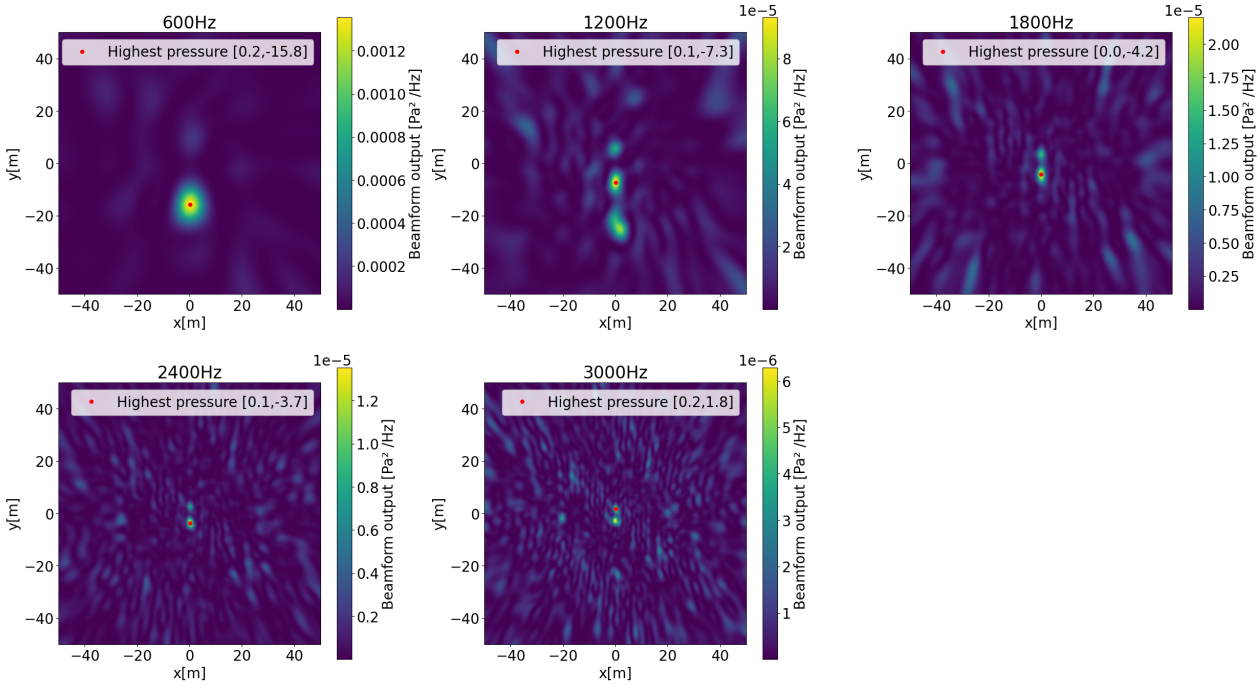


Figure 8: Beamform plot for the separate frequencies used as input for the 5 input model.

Using point sources causes each of the frequencies to originate from the same location. Second, the pressure level differences between the frequencies is much greater compared to synthetic white noise, as shown in figure 9 . In the synthetic white noise each frequency has comparable pressure levels. Figure 10a shows the averaged beamform plot and the predicted source location by the model. When averaging the different beamform plots, the averaging is done on basis of pressure levels. Looking at figure 10a it can be seen that the highest pressure point is closer to the location of the highest pressure point of lower frequencies in figure 8. Due to training using a point source and comparable pressure levels between different frequencies in the training data, each frequency has an equal contribution to the prediction of the source location by the ANN model.

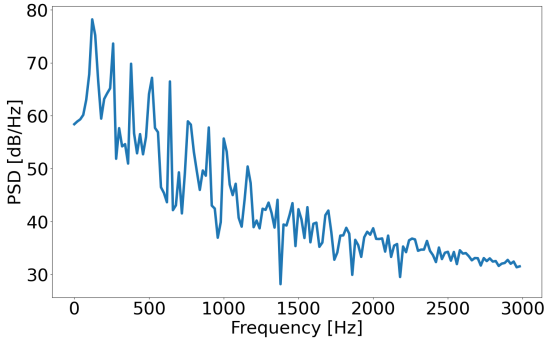
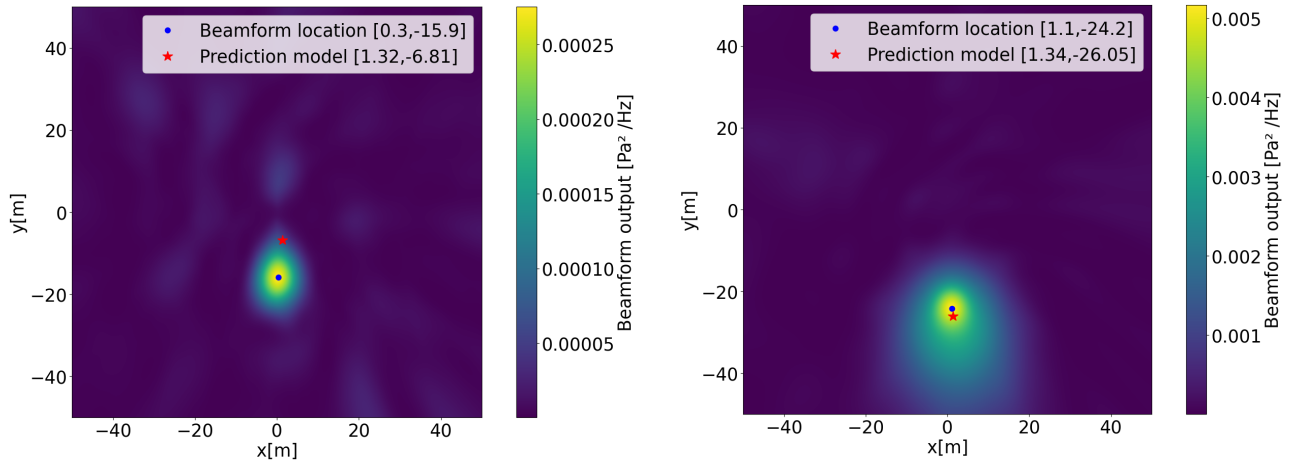


Figure 9: Power spectral density of aircraft flyover recording.

Figure 10b shows the averaged beamform plot and the predicted location using the 10 input model. The frequencies used as input are: 300[Hz], 600[Hz], 900[Hz], 1200[Hz], 1500[Hz], 1800[Hz], 2100[Hz], 2400[Hz], 2700[Hz], and 3000[Hz]. The predicted location of the model is closer to the source location indicated by the averaged beamform plot in figure 10b. This might be due to adding frequencies that contain information corresponding to the location of the acoustic source. However, adding frequencies to the input of the ANN model does not always result in more accurate results, as shown in figures 11a and 11b. This snapshot is from the same recording, but is 0.5[s] earlier than the prior discussed snapshot. As the aircraft is flying at the edge of the scan plane it might occur that the highest pressure point of one of the frequencies added for the 10 input model falls outside of the scan plane, this introduces an error for the model.

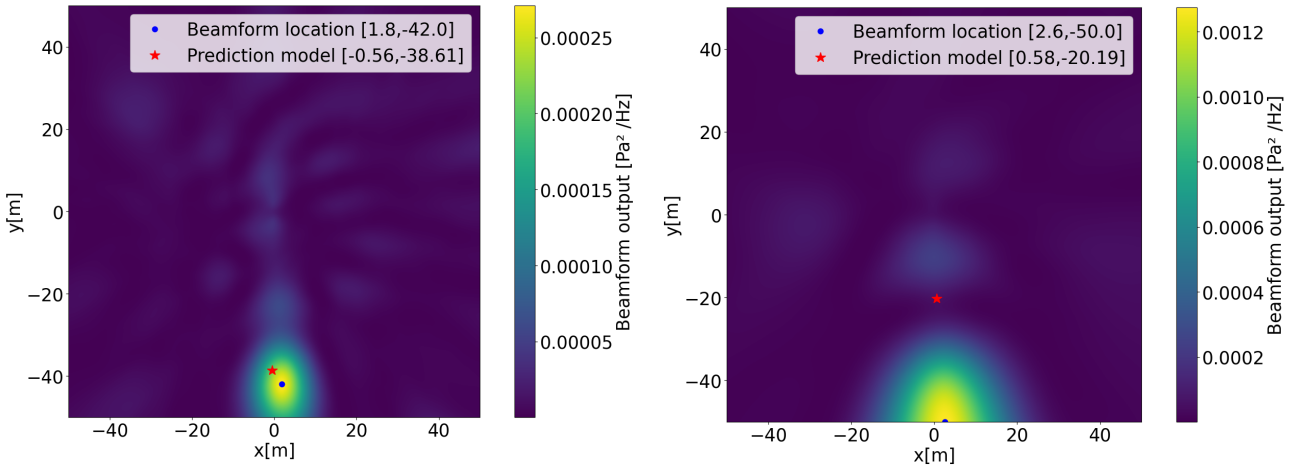
4.3.2 Consecutive localization

This research is focused on studying the potential of an ANN based method for real-time acoustic source localization. Short processing time is important for real-time operations. A program has been build that predicts the acoustic source location every 0.1[s] for 1.5 seconds of flyover recording. Ev-



(a) (b)

Figure 10: Incoherent averaged beamform plot vs. model prediction using 5 input frequencies (a) and 10 input frequencies (b) of aircraft flyover.



(a) (b)

Figure 11: Incoherent averaged beamform plot vs. model prediction using 5 input frequencies (a) and 10 input frequencies (b) of aircraft flyover.

ery 0.1[s], the next 0.05[s] is segmented from the recording. This 0.05[s] is converted to the frequency domain and the converted CSM is created. The converted CSM is used as input for both the model that uses 5 input frequencies and the model that uses 10 input frequencies. The processing time is measured from the point at which the recording segment is obtained up to the point at which a prediction is received from the model. The calculations were done using Python 3.9 on a 2021 Macbook Pro with M1 Max processor and 64GB of unified memory. Table 5 shows the average processing times of 15 segments for both models compared to the processing time of incoherent averaging for beamforming. The beamform plot used for this comparison are those shown in figures 10a and 10b. These beamform plot use a grid size of 0.1×0.1 [m], totaling to 1 million grid points per acoustic image for a single frequency.

Table 5: Processing time in [s] of both multi input ANN models vs. incoherent averaging for beamforming.

5 input model	5 frequency beamforming	10 input model	10 frequency beamforming
0.04[s]	1326.52[s]	0.05[s]	2199.28[s]

Using the discussed program to localize the sound source for a period of 1.5[s], a trajectory can be seen. Figure 12 shows the consecutive predictions of the acoustic source location performed by the ANN model that uses 10 input frequencies. Using conventional beamforming it is known that the aircraft flies in positive y-direction. This is supported by the predictions done by the ANN model, as shown in figure 12.

Contrary to the results in section 4.2, using a different number of input frequencies does greatly influence the accuracy of the acoustic source location, as larger differences are present between different frequencies. Using more input frequencies does not necessarily mean that the prediction accuracy improves.

5 Discussion

During the first phase of this research, synthetic white noise was used for both training and testing. As a result, the training data is a good representation of the test data. Section 4.1 shows that the model using a single input frequency is able to find a relation between the input feature and the acoustic source location. Training the single input model on different frequencies shows similar behaviour as conventional beamforming.
Even though section 4.1 shows the model is able to find a relation between the input feature and the source location, results can still be improved by taking a couple of actions. First, all datasets used in this research contain 160000

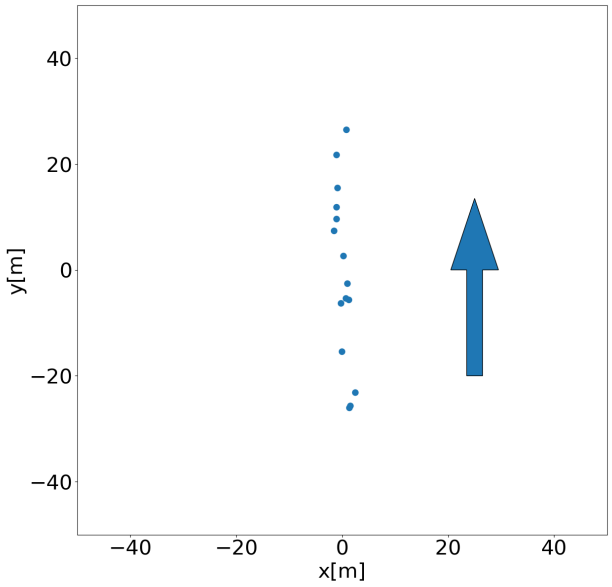


Figure 12: Acoustic source location predictions for 1.5[s] of flyover recording. The arrow indicated the direction of flight.

training examples, 40000 validation examples, and 50000 testing examples. This quantity was limited by computer limitations. Increasing the amount of training data has shown to be beneficial in many ANN applications. Second, by further tuning the number of hidden layers and their neurons higher prediction accuracy might be obtained.
Section 4.2 has shown that using 5 input frequencies makes little difference to using 10 input frequencies. As the model is both trained and tested on synthetic white noise, different frequencies contain similar information. Section 4.3, however, shows that using a different number of input frequencies has significant influence on the prediction accuracy when using real world recordings. In real recording data different frequencies are more varied in terms of source location and sound level. To gather sufficient training data it is hard to avoid the use of simulated data. To achieve better results on real recording data, further research has to be done into mimicking real world situations using simulated data. For each application the characteristics of real world recordings might differ. For hear-and-avoid application the simulated data should be comparable to different aircraft flybys.
The microphone arrays used during this research consist of 64 microphones. The dimensions of the microphone array are not suitable for hear-and-avoid. Further research is to be done in a microphone array that is suitable for a UAV platform. When the number of microphones is altered, the size of the input layer of the network also changes. With the changing size of the input layer, the network's architecture should again be optimized.

6 Conclusion

This paper has described the research into the potential of an artificial neural network based method for fast acoustic source localization. A multi-layer perceptron model was used to localize an acoustic source on a 2-dimensional scanning plane. In the first phase the acoustic source consisted of a simulated point source emitting synthetic white noise. A multi-layer perceptron model was trained to localize the acoustic source using a converted version of the CSM of one frequency. The predictions show similar behaviour to conventional beamforming for different frequencies. Concluding that the model is able to understand the input data and find a relation between the input feature and the acoustic source location. From conventional beamforming it is known that inchorent averaging the results of multiple acoustic images improves the prediction accuracy. Based on this principle a new network architecture is created that uses the converted CSMs of multiple different frequencies as input. The multi input model has shown to have a mean absolute error of approximately 0.27[m] for both the x- and y-coordinate, which is a reduction of approximately 0.2 to 0.6 compared to the the single input model, depending on the input frequency. Increasing the number of input frequencies from 5 to 10 input frequencies has to have little effect to the prediction accuracy when localizing synthetic white noise point sources. The models using the information of multiple frequencies were tested on recording data of an aircraft flyover. For real world recordings it was shown that the input frequencies used greatly influence the prediction accuracy, due to a larger difference in source location and pressure levels of different frequencies. The ANN based method has shown to be able to obtain a prediction within approximately 0.05[s], compared to approximately 1000-2000[s] for conventional beamforming. This opens the door for real time acoustic source localization and tracking.

This research has shown promising results in terms of prediction accuracy and processing time. However, further research is required to improve the magnitude and consistency of the prediction accuracy for real world acoustic source localization.

References

[1] Graham Wild, John Murray, and Glenn Baxter. Exploring civil drone accidents and incidents to help prevent potential air disasters. *Aerospace*, 3:22, 07 2016.

[2] Gilbert Wu, Andrew Cone, Seungman Lee, Christine Chen, Matthew Edwards, and Devin Jack. Well clear trade study for unmanned aircraft system detect and avoid with non-cooperative aircraft. 06 2018.

[3] RTCA (Firm). SC-186. *Minimum Operational Performance Standards (MOPS) for Aircraft Surveillance Applications (ASA) System*. RTCA, Incorporated, 2011.

[4] Raman K. Mehra, Jeffrey Byrne, and Jovan D. Boskovic. Flight testing of a fault-tolerant control and vision-based obstacle avoidance system for uavs. 2005.

[5] Antoine Beyeler, Jean-Christophe Zufferey, and Dario Floreano. optipilot: control of take-off and landing using optic flow. 2009.

[6] Sunyou Hwang, Jaehyun Lee, Heemin Shin, Sungwook Cho, and David Hyunchul Shim. Aircraft detection using deep convolutional neural network in small unmanned aircraft systems. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, page 2137. 2018.

[7] Xiang Yu and Youmin Zhang. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences*, 74:152–166, 2015.

[8] Dirk Wijnker, Tom van Dijk, Mirjam Snellen, Guido Croon, and Christophe De Wagter. Hear-and-avoid for unmanned air vehicles using convolutional neural networks. *International Journal of Micro Air Vehicles*, 13:175682932199213, 01 2021.

[9] Mark van der Woude. Acoustic-based aircraft detection and ego-noise suppression: for micro aerial vehicles. 2021.

[10] R. Merino-Martínez, P. Sijtsma, M. Snellen, T. Ahlefeldt, J. Antoni, C. J. Bahr, D. Blacodon, D. Ernst, A. Finez, S. Funke, and et al. A review of acoustic imaging methods using phased microphone arrays. *CEAS Aeronautical Journal*, 10(1):197–230, 2019.

[11] Ben Luijten, Regev Cohen, Frederik J de Bruijn, Harold AW Schmeitz, Massimo Mischi, Yonina C Eldar, and Ruud JG van Sloun. Deep learning for fast adaptive beamforming. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1333–1337. IEEE, 2019.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[13] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation, 2017.

[14] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[15] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[16] Adam Kujawski, Gert Herold, and Ennes Sarradj. A deep learning method for grid-free localization and quantification of sound sources. *The Journal of the Acoustical Society of America*, 146:EL225–EL231, 09 2019.

[17] Paolo Castellini, Nicola Giulietti, Nicola Falcionelli, Aldo Franco Dragoni, and Paolo Chiariotti. A neural network based microphone array approach to gridless noise source localization. *Applied Acoustics*, 177:107947, 2021.

[18] D. G. Simons. Introduction to aircraft noise reader, 2019.

[19] M.W Gardner and S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636, 1998.

[20] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units, 2018.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[22] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[23] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning, 2016.

II

Literature Review

Graded under AE4020

1
Introduction

In recent years the development of unmanned aerial vehicles (UAV) has been advancing in a staggering pace. With this the number of applications also has been increasing rapidly, from parcel delivery to environment mapping [4, 38]. The development of autopilots for UAVs has sped up the application of UAVs even more by allowing them to fly autonomously in open areas [6]. Due to the increased amount of UAVs, the amount of hazardous situations is also increasing [58]. These hazardous situations can be split into two categories: collisions with static obstacles and interference with other air traffic. This research is focused on finding a solution for the second category. Many regulations and restrictions exist to prevent hazardous situations between UAVs and other air traffic. For instance, UAVs are not allowed to get close to airports. However, hazardous situations still occur.

To avoid collision between UAVs and air traffic several technical security measures can be applied. These measures become more important if a UAV is flying autonomously. One of the systems used in aircraft avoidance is Automatic Dependent Surveillance-Broadcast ($ADS-B$). ADS-B is a satellite based aircraft localisation technique and is seen as the successor of radar. However, ADS-B is vulnerable to loss of connection and security breaches [33].

A new initiative called Single European Sky ATM Research ($SESAR$) aims to safely increase the use of airspace within Europe. One of the projects within SESAR is called PercEvite. percEvite is a collaboration between the TU Delft, KU Leuven, Parrot, AeroVinci and SESAR Joint Undertaking, that focuses its efforts on developing a sensor, communication and processing suite for small UAVs. One of the goals of the PercEvite project is to enable small UAVs to avoid objects using lightweight, energy-efficient sensor. An often used sensor for avoidance is a camera. These sensors are lightweight and consume little energy. Stereo vision and optical flow are often applied methods that employ cameras. However, both of these methods have their flaws. Stereo vision is only suitable for detecting objects located within a short range. Optical flow relies on textures in the image. Within the PercEvite project the TU Delft is developing a method of avoiding other aircraft by means of a small acoustic sensor. An algorithm which is able to avoid other aircraft by means of acoustic sensors is called a hear-and-avoid algorithm. Hear-and-avoid includes two major aspects: sound event recognition and sound source localization. Sound event recognition is the science of detecting a sound and classifying the source of that sound. Sound source localization is determining the location of the sound source with respect to the receiver.

This report is a review of the literature available on sound event recognition and sound source localization suitable for hear-and-avoid. Before it possible to go deeper in the aspects of sound event recognition it is important to know what sound event recognition is and how it differs from other sound processing applications. Chapter two will discuss the essence of sound event recognition and what modules it includes. Chapter three will explain the fundamentals of acoustics and sound representations often applied in sound event recognition. An often applied classifier in sound even recognition is the artificial neural network. Chapter four explains the working principle of artificial neural networks and variants often applied in sound event recognition. Chapter five discusses round event recognition algorithms with promising results which have been proposed in recent years. Eventually, several methods for sound source localization are explained in chapter six, including methods based on artificial neural networks.

2
Sound Event Recognition

In recent years the applications of audio processing has been making leaps forward with fields such as Automated Speech Recognition (*ASR*) and Sound Event Recognition (*SER*). Untill a few years ago, the field of SER did not get as much appreciation as ASR. However, due to the progress in the field of ASR, the possibilities of SER have grown. This chapter will give an overview of the structure and applications of SER.

2.1. Overview

Sound Event Recognition is the science of recognizing and classifying the source of a sound event. A sound event is a sound that originates from an object, such as a car passing, a phone ringing, and in this research aircraft sound. These sound events carry all sorts of information and cues, which can be used to obtain knowledge about the source. This information consists of sound event duration and characteristic frequencies.

Compared to speech and music, it is much harder to obtain characteristic information from sound events. This is due to the many different origins of a sound event. In comparison, speech is confined as the sound humans are able to vocally produce. Table 2.1 shows a comparison of the characteristics of speech, music and sound events.

Table 2.1: Comparison of the acoustical characteristics of speech, music, and sound events [65]

Acousitcal Characterisitcs	**Speech**	**Music**	**Sound Events**
Number of Classes	Number of Phonemes	Number of Tones	Undefined
Length of Window	Short (fixed)	Long (fixed)	Undefined
Bandwidth	Narrow	Broad	Broad & Narrow
Harmonics	Clear	Clear	Clear & Unclear
Repetitive Structure	Weak	Weak	Strong & Weak

From table 2.1 it can be denoted that it is hard to define the acoustic characteristics for sound events, compared to speech and music. Due to the extreme variety in characteristics of sound events, it is usual the scope for the problem of SER that is narrowed. This is done by choosing one specific type of sound to classify, such that some characteristics can be defined. To give more structure to the field of SER, it is important to categorize sounds. This creates sub-fields within SER that enables more specific understanding of the data domain. An example of these categories can be seen in figure 2.1, in which hearable sound is divided into several categories and sub-categories.

The different classes described in figure 2.1 follow how humans would classify sounds. This research is focused on classifying aircraft sound, which falls under the category of Artificial Sounds. However, Noise and Natural Sounds can also be present in audio recordings.

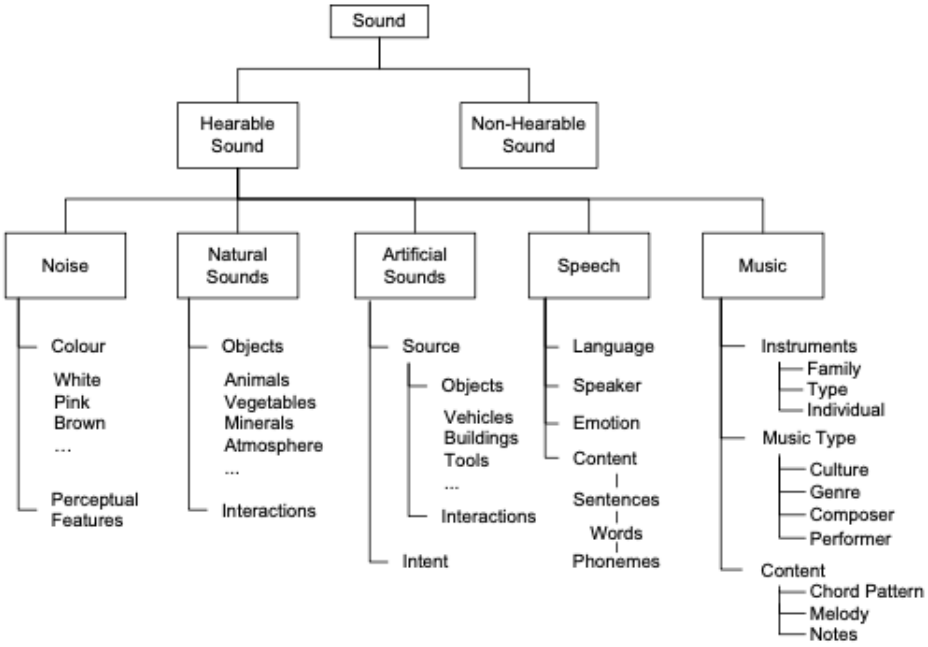


Figure 2.1: Taxonomy of sound [20]

2.2. Applications of Sound Event Recognition

SER can be applied is many different fields, due to the general definition of a sound event. Because of its widespread application, it is hard to compare different methodologies of SER. Therefore, many researchers focus on one specific field of interest. In general SER can be defined in three different categories: environmental sound event recognition, acoustic surveillance, and environmental classification.

2.2.1. Environmental Sound Event Recognition

In this category the research is focused on recognizing a specific subset of sounds that are found in a given environment. The environment in which the sound event is taking place defines the recognition problem. It has to be noted that recognition of speech in a certain environment could also be included in this category. However, the content of speech is not interpreted, but is passed on to an ASR module. Some examples of environmental sound event recognition are: counting of coughs for healthcare purposes [5] and classification of meeting room sounds [55].

2.2.2. Acoustic Surveillance

An often adopted manner of surveillance is the use of cameras. However, in some situations it is only possible to make use of acoustics for surveillance. Overall the field of acoustic surveillance can be divided into two sub-categories: human and animal surveillance. Examples of human surveillance include detection of aggressive sounds, such as: screams or gunshots [21]. Acoustic surveillance is used in biological environments as it allows for long term monitoring. The main obstacle of acoustic surveillance is their robustness to noise.

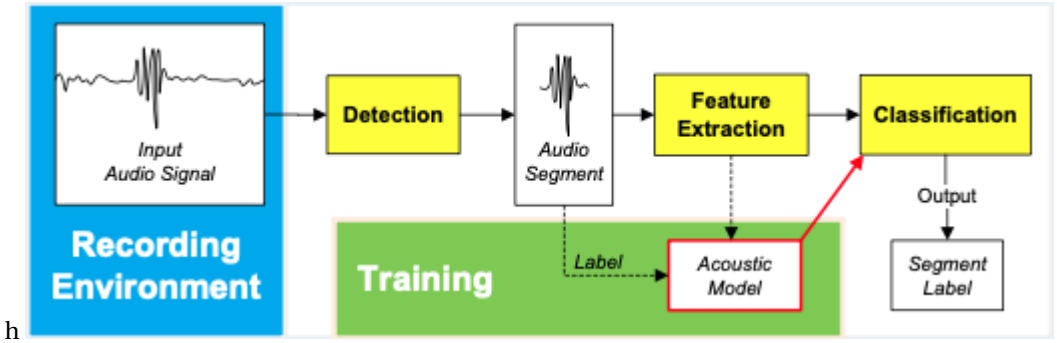
2.2.3. Environmental Classification

Environmental Classification is the technique of recognizing the surrounding environment from the audio recording [32]. It can be applied to tune the settings of a device depending the context in which it is used [18]. An example is tuning parameters of hearing aid devices to improve the listening experience in different environments [44].

Looking at the SER categories above, it can be said that an hear-and-avoid algorithm falls within acoustic surveillance. In an hear-and-avoid algorithm, aircraft sound has to be distinguished from background sound.

2.3. Sound Event Recognition Structure

Eventhough many different forms of SER systems exist, most of the algorithms follow a common structure. SER systems typically consist of three different modules: detection, feature extraction, and classification.



h

Figure 2.2: Typical structure of a sound event recognition system [19]

To get a clear overview of the structure of an SER system, figure 2.2 illustrates the typical structure of an SER system. Each of these modules forms its own phase in recognizing an audio segment. The following subsections will give a brief explanation on the different SER modules.

2.3.1. Detection

The focus of the detection module is finding the start and end point of a sound event from a continuous stream. Within sound event detection two different approaches could be applied: detection-and-classification and detection-by-classification [56].
The detection-and-classification approach consists of two separate modules. A detection module which detects and extracts sound event segments from a continuous. The module does not interpret the data, but only separates the sound event segment from the background noise. In the segmentation process, a sliding window slides through the audio signal and determines a similarity measure between neighbouring regions. This similarity measure is compared to a threshold, witch is determined previous noise profiles. The benefit of this approach is that the segmentation window is variable, which is better for recognition tasks in which the duration of sound events differs substantially. However, it is difficult to determine the similarity threshold in audio recordings with inconsistent background noise.
In the second approach, detection-by-classification, the detection and classification modules are combined into one module. It classifies sequential segments from the audio stream, in which the detection window slides over the audio segments [55]. At each time step the module determines whether the segment contains only noise or one of the pre-trained sound events [69]. Figure 2.3 illustrates the approach used by detection-by-classification.

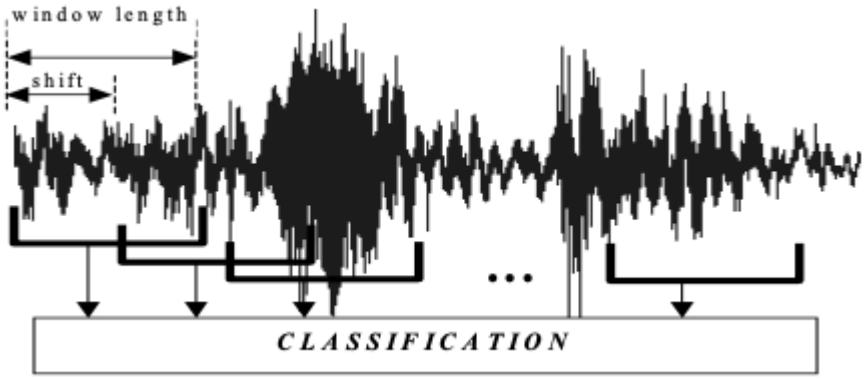


Figure 2.3: Detection-by-Classification [56]

The benefit of the detection-by-classification approach is that it is easier to implement as no separate modules have to be constructed for detection and classification. The problem, however, comes with determining the proper window size that works for varies different sound events.

2.3.2. Feature Extraction

The objective of the feature extraction module is to highlight the characteristics of the audio signal that contain the information on the sound event. Not all features are well suited for recognising some specific sound event classes compared to others. Therefore, it is necessary to use a sound feature that fits to the specific

recognition task. A well chosen audio feature is able to distinguish between different sound classes. Sound features have several different properties: signal representation, domain, temporal scale, semantic meaning, and underlying model [36].
Signal representation describes how the signal is coded. The decoded sound can be one of two categories: linear coded or lossily compressed. Most feature extraction methods use linear coded signals, as these signals contain all information present in the original audio signal. However, in recent years more research is done into using lossily compressed audio features. The benefit of using lossily compressed audio features is that the audio signal is transformed into a frequency range that is audible by humans. Therefore, sounds that cannot be heard by humans are removed from the audio signal. This reduces the computational efforts for extracting features significantly.
Knowledge about the domain makes interpretation of the sound feature easier. In total seven different domains exist: temporal, frequency, correlation, cepstral, modulation frequency, reconstructed phase space, and eigen-domain. A feature expressed in the temporal domain, also known as the time domain, is directly given as a waveform of the signal. The frequency domain on the other hand describes the spectral information of the signal. The correlation domain gives the relation between signals. In audio processing, the autocorrelation is mainly used as this compares the signal to a time-shifted copy of itself. By doing this it is able to find repeating patterns and its periodicity. The cepstrum domain is used to show the envelope of the spectrum. The cepstrum is defined as the power spectrum of the logarithm of the power spectrum of a signal [10]. The modulation frequency domain informs on the temporal modulations in a signal. Reconstructed phase space is able to extract non-linear features from a signal [29]. lastly, a representation in the eigen-domain is constructed of eigen- or singular vectors.
The third feature property is the temporal scale. generally, audio is considered as a non-stationary signal depending on time. This is why multiple feature extraction methods use short frames (10 to 40 ms) of an audio signal, in which a signal is said to be locally stationary. Each of these frames is used to generate a feature vector, containing information such as volume and Fourier transform coefficients. Features that use the information extracted from the stationary audio frames are called intraframe features [59]. On the other hand, interframe use longer fixed length (usually 1 or 2 seconds) clips to extract dynamic information of the signal, such as rhythm and modulation information [59]. Lastly, there are global features. These features use the entire audio signal to extract information.
The semantic meaning of an audio feature indicates if the feature represents the human perception. A feature is perceptual if the feature is known by humans, examples are: harmonicity, loudness, pitch, and rhythm [67]. The semantic meaning of a feature can also be physical. Physical features are described by mathematical or physical properties, Fourier transform coefficients for example.
The last property of an sound feature is the underlying model. Lately, researchers have used psychoacoustic models in feature extraction. Psychoacoustic models use a mathematical representation of a signal that correspond to the human auditory system [54]. For example, a psychoacoustic model implements filters that simulate the frequency humans are able to hear.
All audio features used for sound event recognition can be characterized in these feature properties.

2.3.3. Classification

The third and final module of a sound event recognition system is the classification module. The goal of this module is to classify the features extracted from the audio signal. The main working principle of the classification module is to assign a label to audio segments corresponding to the classes used to train the classifier. Many different techniques can be applied to classify an audio segment. The techniques mainly use one of two approaches. The first approach uses a database containing training features and a distance measure to compute similarities between the training and test features. This approach is used in k-Nearest Neighbours (kNN) and Dynamic Time Warping (DTW), among others. However, this approach is often not favorable as it is more computationally expensive [39]. The second approach is a model-based approach. The model-based approach generates a feature vector while training the model. When testing the model, the distance between the trained model en the extracted training features is measured. The model-based approach is applied in various classification methods, such as Gaussian Mixture Model (GMM), Hidden Markov Models (HMM), and Artificial Neural Networks (ANN) [16]. This research mainly focuses on using Artificial Neural Networks as classifier. In recent years the use of various types of Artificial Neural Networks has shown promising results in terms of acoustic surveillance [1**?**]

3
Sound

Sound can be seen as a disturbance of air which propagates as a longitudinal pressure wave. For sound event recognition it is important to understand the fundamentals of sound. This chapter will explain the basics of acoustics and aircraft sound in particular. An essential aspect of sound event recognition are the features used. Section 3.3 will discuss sound representations and what features can be extracted from these representations.

3.1. Basics of Acoustics

Due to vibrations of surfaces, the sound source, a longitudinal wave is created which displaces particles in the air. When the displacement of particles reaches the ear, a sound is heard. The displacement of the air particles is around their local equilibrium position, which produces a local increase in pressure following by a decrease in pressure. This fluctuation in pressure is measured in sound pressure, denoted by p'. The fluctuation in pressure on the eardrum will make a sound audible to a listener. Over distance the amplitude of the Sound Pressure decreases. Next to distance, the sound pressure is dependent on several other variables. The sound pressure variations with time and distance is expressed in the following equation as a function of time t and distance r.

$$p'(r,t) = \frac{A}{r} cos\left[\omega\left(t - r/c\right)\right] \tag{3.1}$$

In which, A is the amplitude at 1 meter of the sound source, c is the speed of sound (generally 340 $[m/s]$) and ω is the radial frequency in radians per second. The effective (sound) pressure is the most often used measure of amplitude, and is denoted by p_e. Effective (sound) pressure is the root-mean-square of the instantaneous sound pressure over one period. Equation 3.2 shows the relation between the sound pressure and the effective sound pressure.

$$p_e = \left[\frac{1}{T}\int_0^T \left[p'(t)\right]^2 dt\right]^{1/2} = \left[\frac{1}{T}\int_0^T \left[\frac{A}{r} cos\left[\omega\left(t - r/c\right)\right]\right]^2 dt\right]^{1/2} = \frac{A}{r\sqrt{2}} \tag{3.2}$$

Equation 3.2 shows that the amplitude of the sound pressure and effective sound pressure are related by a factor of $\sqrt{2}$. The effective sound pressure is the most commonly measured quantity because most microphones measure sound pressure. A microphone's read-out generally gives the sound pressure level (SPL) in decibel (dB). Equation 3.3 show the relation between sound pressure level and the effective sound pressure.

$$SPL = 10log\left(\frac{p_e^2}{p_{e_0}^2}\right) \tag{3.3}$$

In which p_{e_0} is the reference pressure of 2x10^{-5} $[N/m^2]$. In figure 3.1 it is shown that using the sound pressure level instead of the effective sound pressure reduces an audible range.

3.2. Aircraft Sound

As this research is focused on detecting the presence of aircraft, it is important to understand the characteristics of sound produced by aircraft. Sound produced by aircraft mainly originates from the propulsion system. However, other components such as airframe and landing-gear also contribute. Various different type of aircraft use different types of propulsion systems. As the hear-and-avoid algorithm is developed for the implementation on UAVs, not all types of aircraft will be encountered during flight. The type of aircraft most likely to form a hazard for UAVs are small aircraft. Almost all aircraft in this category use piston engines for propulsion. Sound produced by the piston engine propulsion system can be divided into two categories: sound produced by the piston engine and sound produced by the propeller.
A piston engine usually exists of four to six cylinders that operate on the four-stroke cycle principle, explained in figure 3.2.

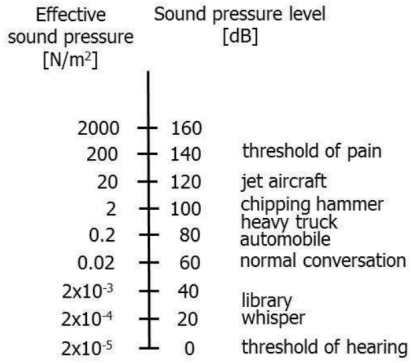


Figure 3.1: Relation between effective sound pressure and sound pressure level

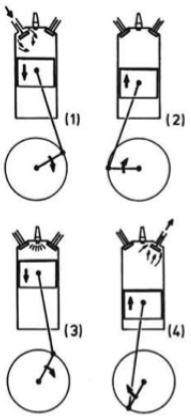


Figure 3.2: The working cycle of a four-stroke piston engine. (1) The piston travels downwards to allow a mixture of vaporized fuel and air to be sucked in. (2) The piston travels upwards to compress the fuel and air mixture. (3) The compressed fuel and air mixture is ignited, pushing the piston back down. (4) The piston travels upwards to push out the burned gas inside the cylinder. [46]

The piston engine itself produces sound at discrete frequencies which are integer multiples of the cylinder firing frequency (f_c), equation 3.4. In which n is the engine's rotational speed in rpm. However, the most dominant frequency is the exhaust firing frequency (f_e) ant its harmonics, equation 3.5. In which N is the number of cylinders and n is the engine's rotational speed

$$f_c = \frac{n}{120} \tag{3.4}$$

$$f_e = \frac{Nn}{120} \tag{3.5}$$

The propeller of the piston propulsion system converts shaft power from the engine into thrust by pushing air backwards. The sound produced by the propeller consists of rotational sound and vortex sound. Rotational sound is caused by the propeller rotating through air. The rotational sound is harmonically related to the blade passage frequency (f_1), equation 3.6.

$$f_1 = \frac{Bn_p}{60} \tag{3.6}$$

In which B is the number of propeller blades and n_p is the rotational speed of the propeller. Vortex sound is caused by random air disturbances caused by the propeller blades. The vortex sound is significantly weaker compared to the rotational sound, causing the propeller sound spectrum to show peaks at integer multiples of the blade passage frequency.

During flight the sound observed on the ground is influenced by three different factors. The first factor is the influence of motion of the aircraft on the directional pattern of the noise field. This influence is also known as the Doppler effect. The Doppler effect is known as the change of the observed frequency due to the sound source moving with respect to the observer. The observed frequency can be calculated using equation 3.7.

$$f' = \frac{f}{1 + \frac{dr/dt}{c}} \tag{3.7}$$

In which f' is the observed frequency, f the true frequeny, dr/dt is the change of distance to the source with respect to time, and c is the speed of sound in air.
The second factor of influence is the effect of the ground and other obstacles reflecting sound waves. The third factor is the effect of the airframe, known as aerodynamic sound. Aerodynamic sound is caused by a turbulent boundary layers over the outer surfaces of the aircraft. It is most present during landing when landing-gear and high-lift devices are deployed.

3.3. Sound Representations

Recorded sound contains a lot of information. Different representations can be used to extract this information from a sound recording. This section will discuss sound representations often applied in sound event recognition.

3.3.1. Frequency Domain

As sound is actually a wave travelling through the air it can also be defined in the frequency domain. The frequency domain expresses a signal in terms frequency instead of time. Frequency can be explained as the number of waves per second, expressed in hertz $[Hz]$. Sound recordings contain a huge amount of information in the frequency domain. This is because most sound events contain a wide range of frequencies, as can be seem in figure 3.3.

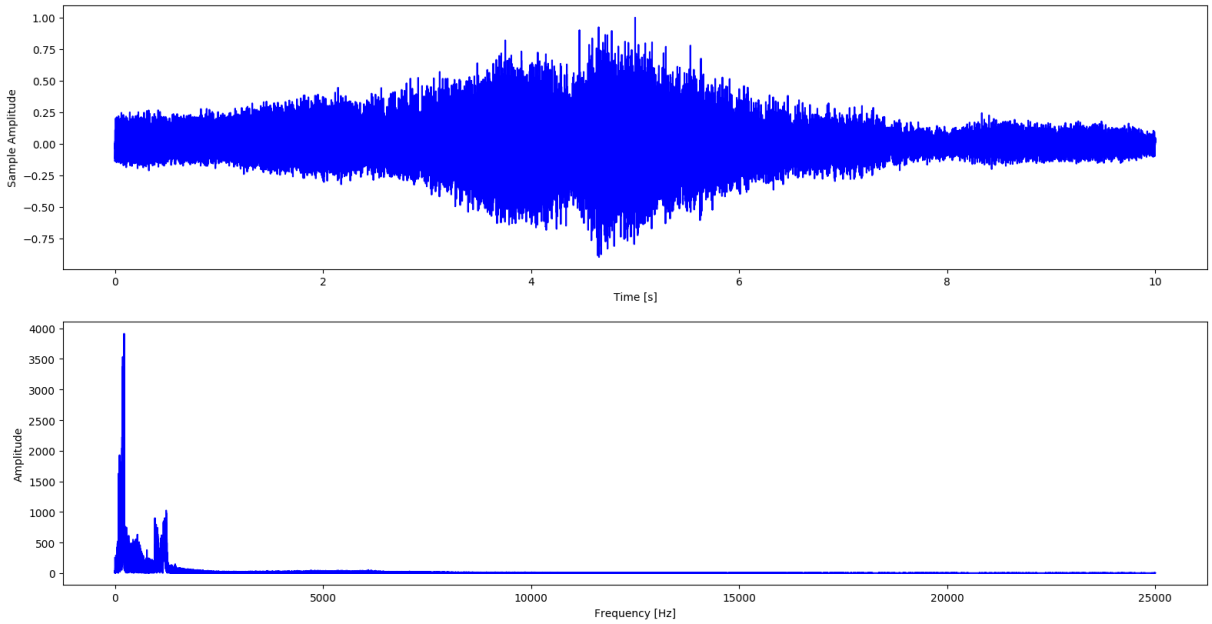


Figure 3.3: Comparison between time domain and frequency domain

Figure 3.3 shows the sound recording of am aircraft flying over. The sample frequency of this signal is 50000 $[Hz]$, meaning the recorder measures 50000 data-points every second. A sound wave is continuous of nature, while a computer is not. For this reason, the wave is reduced from a continuous-time signal to a discrete-time signal. In theory, the most ideal sample frequency produces samples corresponding to the instantaneous value of the continuous-time signal. To obtain an optimal sampling the Nyquist-Shannon theorem is applied [51]. This theorem states that if a signal does not contain any frequencies higher than W $[Hz]$, it is determined by giving its ordinates at a series of $0.5W$ seconds apart. So, the minimal sampling rate should be twice as much as the highest frequency of the signal. This explains why the frequency domain plot of figure 3.3 is in the range of $[0, 25000]$.
To obtain frequency information from a time signal, the signal has to be transformed from the time domain signal to the frequency domain signal. This can be achieved by applying the Fourier Transform. The Fourier transform applies the Fourier theorem for the transformation. According to the Fourier theorem it is possible to write every periodic waveform in a trigonometric series representation. The resulting trigonometric series

is called the trigonometric Fourier series. Equation 3.8 gives the mathematical representation of the Fourier series [45].

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n cos\left(n2\pi f_0 t\right) + \sum_{n=1}^{\infty} b_n sin\left(n2\pi f_0 t\right) \tag{3.8}$$

In which f_0 is the fundamental frequency of the wave. The Fourier transform coefficients, denoted by a_0, a_n, and b_n, can be used as features for audio classification [59]. The mathematical expression of the Fourier transform coefficients are shown in equations 3.9 to 3.11

$$a_0 = \frac{1}{T_0}\int_0^{T_0} x(t)\,dt \tag{3.9}$$

$$a_n = \frac{2}{T_0}\int_0^{T_0} x(t)\,cos\left(n2\pi f_0 t\right)dt \tag{3.10}$$

$$b_n = \frac{2}{T_0}\int_0^{T_0} x(t)\,sin\left(n2\pi f_0 t\right)dt \tag{3.11}$$

Using the trigonometric series representation above it is possible to determine the signal in the frequency domain. This is done by applying the Fourier transform given in 3.12. To transform from the frequency domain back to the time domain, the inverse Fourier transform (equation 3.13 is used. Together, equations 3.12 and 3.13 are known as a Fourier transform pair.

$$X\left(f\right) = \int_{-inf}^{\infty} x(t)\,e^{-i2\pi f_0 t}\,dt \tag{3.12}$$

$$x(t) = \int_{-inf}^{\infty} X\left(f\right)e^{i2\pi f_0 t}\,dt \tag{3.13}$$

The function described above are all defined in continuous-time. The audio recordings are sampled data signals. These sampled data signals are defined in discrete-time. For these signals the discrete Fourier transform (DFT) can be used to convert a signal from the time domain to the frequency domain [45], given in equation 3.14

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}}, \quad k = 1,2,...,N \tag{3.14}$$

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\frac{i2\pi kn}{N}}, \quad n = 1,2,...,N \tag{3.15}$$

In which N is the total number of samples in the within the recording interval. As with the continuous-time Fourier transform is the DFT part of a discrete Fourier transform pair together with the inverse DFT, given in equation 3.15. Algorithms used to calculate the discrete Fourier transform are often referred to as fast Fourier transform (FFT) algorithms, which efficiently compute the discrete sum of the DFT on a computer.

3.3.2. Psychoacoustics

Many studies on sound recognition consider humans to be superior in recognising sound sources compared to computers. Therefore, many studies look at the human perception of sound. A widely known fact is that the frequencies audible to humans range from 20 [Hz] to 20,000 [Hz]. This narrows the frequencies of interest for sound recognition algorithms. Various sound recognition methods also make use of mathematical representations of the human perception of sound to make recognition more robust. These representations are in the form of mathematical models and are called psychoacoustic models. As explained in section 2.3.2, psychoacoustic models can be used to extract features from an audio signal. One of the most used psychoacoustic sound features are Mel Frequency Cepstral Coefficients ($MFCCs$). The Mel frequency cepstrum is a scaled representation of the short-term power spectrum of sound. MFCCs are the coefficients that construct the Mel frequency cepstrum. The first step in obtaining the MFCCs is to divide a sound recording into smaller frames (typically 20 [ms]) in witch the signal is considered statistically stationary [31]. These frames are converted to the frequency domain using the discrete Fourier transform. The frequencies belonging to the spectrum are scaled to the Mel scale. The Mel-scale is a partially logarithmic scale that is based on relationship between frequency and perceived pitch. It appears that the human perception of pitch is not completely linear. The

Mel scale maps the frequency approximately linear until 1,000 $[Hz]$ and logarithmic after. The mapping of frequencies to the Mel-scale is performed using equation 3.16 [68].

$$M(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \tag{3.16}$$

Next, the logarithm of the resulted spectrum is taken as the perceived loudness of a signal is approximately logarithmic. The final step in obtaining the MFCCs is applying the discrete cosine transform (DCT) [31]. MFCCs have mainly been used in speech modeling for automatic speech recognition . However, several studies into sound event recognition have shown positive results in using the MFCCs and the Mel-scale [1, 52**?**].

3.3.3. Image Representation

As has been explained in this chapter, sound has various different representation. One of these representations is an image representation. Images of sound have been widely used for sound classification as it offers the opportunity to use image classification algorithms. An often used image representation of sound is the spectrogram image. A spectrogram is a time-frequancy representation of an audio signal. Spectrograms show frequency and energy distributed over time. This makes it possible to distinguish different sound elements and their harmonic structure. Various different types of spectrograms exist [19].

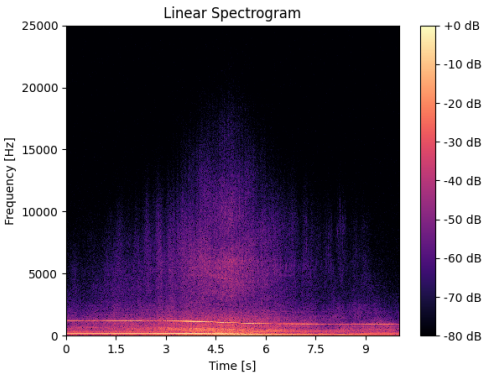


Figure 3.4: Linear-scale spectrogram

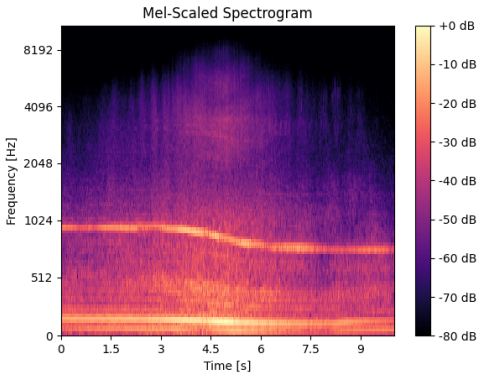


Figure 3.5: Mel-scale spectrogram

The most popular approach in obtaining a spectrogram is based on the short time Fourier transform $(STFT)$, which is based on equations 3.17 and 3.18. Where N is the number of samples per frame, $f = kf_s/N$ is the frequency bin for $k = 1, ..., N/2+1$, w is the window function, and t is the time frame index. Figure 3.4 shows an STFT based spectrogram of the waveform shown in figure 3.3.

$$S_{lin}(f,t) = \left| \sum_{n=0}^{N-1} x_t[n]\, w[n]\, e^{-i2\pi \frac{f}{fs} n} \right| \tag{3.17}$$

$$S(f,t) = S_{log}(f,t) = max\left[log\left(S_{lin}(f,t)\right), max_{f,t}\left(log\left(s_{lin}(f,t)\right)\right) - 80dB\right] \tag{3.18}$$

Another spectrogram which is often used in sound event recognition is the Mel-scale spectrogram, shown in figure 3.5. In a Mel-scale spectrogram the frequency axis is scaled to the Mel-scale, which is discussed in section 3.3.2.

4
Classification

In section 2.3.3 it was mentioned that several different classification techniques can be used to classify a sound's origin. The method discussed in this report is the use of artificial neural networks. Artificial neural networks are based on the immense parallel computation power of the human brain. The human brain can be mathematically modeled by a weighted, directed graph of interconnected neurons. In previous years, artificial neural networks have become increasingly popular as function approximators. Therefore, artificial neural networks are well suited for recognition/classification tasks, including sound event recognition. Numerous different types of artificial neural networks, such as convolutional neural networks, recurrent neural networks and autoencoders. This chapter will explain how neural networks work and how they can be applied for classification tasks.

4.1. Artificial Neural Network

A neural network is a form of machine learning in which a function is created that maps inputs to a corresponding output. Using this function outputs can be predicted for a new inputs. Neural Networks have the form as illustrated in figure 4.1. A neural network consist of multiple layers which contain neurons. The neurons of a layer are connected to all neurons of the neighbouring layers.

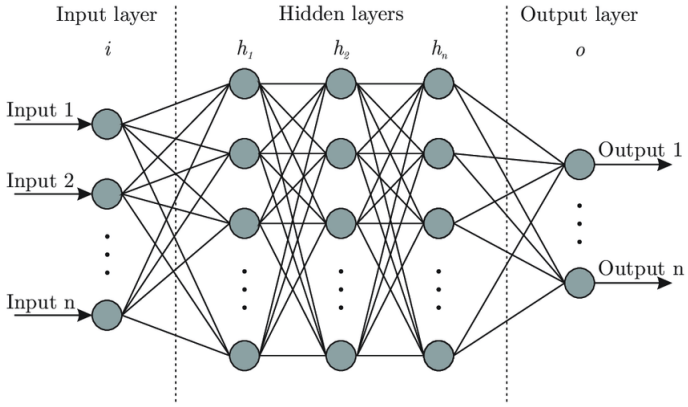


Figure 4.1: Basic form of artificial neural networks [11]

As indicated by figure 4.1, the first layer of a neural network is the input layer. This layer takes the input values of the function. The output layer produces the eventual output of the function. This can consist of multiple neurons in the case of classification. Each neuron produces the probability of the input belonging to a certain class. All layers in between the input and output layers are called the hidden layers. Each neuron consists of an activation function, which takes the outputs of the neurons in the previous layer as input. In this manner an intricate equation can be composed from simple linear and non-linear functions. Some often used activation functions are the linear function, the rectified linear unit function, logistic function and the hyperbolic tangent function, described by equation 4.1 to 4.4 respectively.

$$f(z) = az \tag{4.1}$$

$$f(z) = max(0, z) \tag{4.2}$$

$$f(z) = \frac{1}{1+\exp(-z)} \tag{4.3}$$

$$f(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)} = \tanh(z) \tag{4.4}$$

The connections between the neurons of neighbouring layers all have their own weight. These weights indicate the influence each of the neurons of the previous layer have on a neuron in the current layer. As a result of the weights, the input of a neuron is the outputs of the neurons in the previous layers multiplied by their specific weights. In some cases an additional bias neuron can be added, as illustrated by figure 4.2. Bias neurons make it possible to translate the activation function to the left or right.

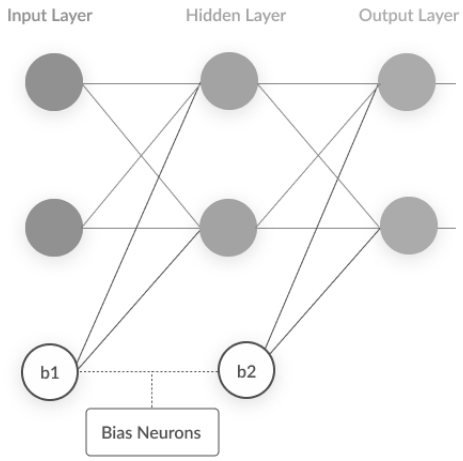


Figure 4.2: Addition of bias neurons

To create a function that maps the input to the output, the neural network has to learn the relationship between the input and the output. The most applied method of learning is called supervised learning. In supervised the neural network is given a dataset containing input data (such as images) and output labels (such as categories). During the training of the network it is shown the items in the dataset. It then tries to determine the corresponding label. The network is highly likely unable to correctly determine the label before training. To improve the accuracy of the network an objective function is established. This objective function measure the error between the network's output and the desired output. This objective function is used to properly adjust the weights and biases of the network. The main goal is to minimize the error between the network's output and the desired output. To do so backpropagation is used. The backpropagation procedure calculates the gradient of the objective function with respect to the weights in the network. In essence backpropagation is no more than an application of the chain rule for derivatives. The gradient of the objective function with respect to the input of a certain neuron can be calculated by computing the gradients of each neuron's output with respect to it's input. Calculating the gradients is done repeatedly through all neurons, starting with the output of the network and continuing till the input neurons. Once these gradients have been calculated, it is easy to calculate the gradients with respect to the weights of each neuron.

One of the biggest challenges in learning a neural network is over- and underfitting. Overfitting appears when the network performs well on the training dataset, but performs bad on net unseen inputs. Underfitting appears when the network is unable to fit a function to the dataset.Several methods can be applied to prevent over- and underfitting. Both the network and the dataset can help to prevent overfitting. Reducing the number neurons and layers in the network can help to prevent overfitting as this produces a function which is less prone to variances. A version of this is applying drop-out. When drop-out is applied in training the network, every training round several neurons are randomly selected and ignored. The output of these neurons is not passed to the following neurons. This makes sure that that particular training round the features corresponding to the dropped-out neurons are not trained. Prepossessing the training data can also help to prevent overfitting. By removing irrelevent features from the data, the network becomes less complex and less prone to overfitting. Where simplifying the network can help to prevent overfitting, increasing the complexity of the network can help to prevent underfitting.

4.2. Convolution Neural Network

Convolutional neural networks (ConvNets) are a type of neural networks designed for processing data that contains multiple arrays. ConvNets are ofted used in classifying colored images. Just as a general neural network, a ConvNet starts with an input layer. The second layer in a ConvNet is the convolutional layer. In a smaller window moves over image. This filter is a an array of weights. The convolutional layer usualy consists of multiple different filters. These filters can be seen as feature identifiers, each detecting a different feature

such as: edges, colors or curves. When the filter moves over the image, element wise matrix multiplication is applied between the numbers in the filter and the input values it covers. The results of the multiplications is summed up to a single value. If a filter detects the feature it is looking for in the input data, the resulting value will be higher. The result the filter moving over the input data is stored in an activation map. These activation maps contain data about the location of the features. Adding more filters will add depth to the activation map, such that it stores more information about the input data. The basic form of a ConvNet is shown in figure 4.3.

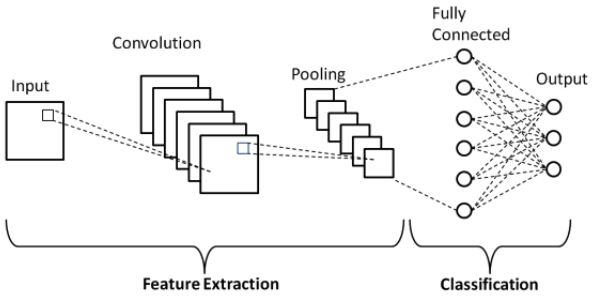


Figure 4.3: Basic form of convolutional neural network [41]

Using convolutional layers has two main benefits [28]. Local values in the input data are often highly correlated with their neighbouring values, forming distinctive features which are easier to detect. Secondly, Features are not bound to one particular location, but can be present in multiple locations in the input data. After the convolutional layers of the network a pooling layer is implemented. The pooling layer serves to decrease the spatial size of the data to reduce the number of parameters and computational effort of the network. The most often type of pooling layer is max pooling. In a max pooling the maximum value of a small local patch of data points is taken, as shown in figure 4.4. A different type of pooling, called average pooling, works in a similar fashion, but it takes the average value of the local patch of data. When multiple convolutional layers are implemented in the network it is common practise to implement pooling layers between convolutional layers

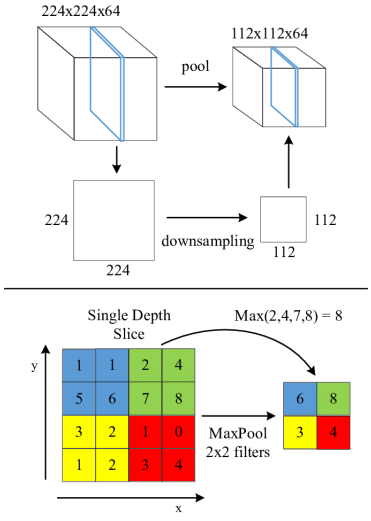


Figure 4.4: Max pooling operation using 2x2 window [27]

subsequently to the convolutional and pooling layers, the resulting data is passed to a fully connected layer for classification. The fully connected layer works similar to the neural network explained in section 4.1. Updating the weights of the convolutional and fully connected layers is done by backpropagating gradients through the ConvNet similar to updating the wieghts of a normal neural network.
ConvNets are most often used for image classification, such as described in [27, 41]. Hoewever, some studies have applied ConvNets in detecting sound sources. Several of these studies will be further explained in chapter 5.

4.3. Recurrent Neural Network

Recurrent neural networks (RNNs) are used for classification tasks involving sequential data, such as speech. The neurons in an RNN are not only trained to recognize the current input, but also its relation to previous inputs of a sequence. Using RNNs it has become possible to predict the next word in a sequence, but also to find a good representation of a thought expressed by a sentence. The basic principle of RNNs is illustrated by figure 4.5. The neurons in the network get updates from neurons at previous time steps. Using this principle, an RNN is able to map input x_t with output o_t. Each output o_t is dependant on all previous inputs x'_t (for $t' \leq t$).

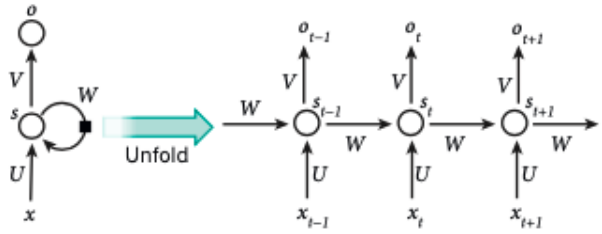


Figure 4.5: A recurrent neural network and the unfolding in time of the computation involved in its forward computation [28]

To update the weights of an RNN a version of backpropagation is applied, call backpropagation through time ($BPTT$). In BPTT an RNN is unfolded and represented as a multi-layer network. Figure 4.6 shows an unfolded RNN in time with a copy of the network at each time step, in which x_t is the state of the network at time t, u_t is the input at time t, and ε_t is the error of the output at time t.

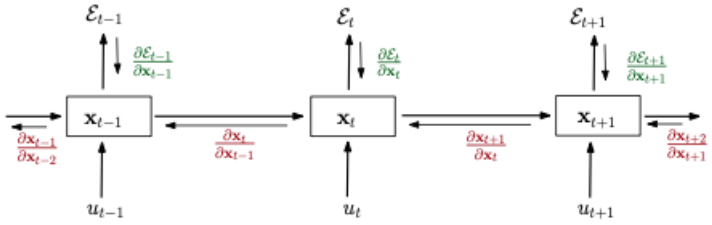


Figure 4.6: Unfolded recurrent neural network with a copy of the network at each time step [40]

A big disadvantage of RNNs is the problem of exploding and vanishing gradients [7]. In updating an RNN a distinguish between long term and short term contributions to the error gradient. Long term refers to the contribution of a copy of the network from a large amount of time steps ago. Exploding gradients occur when the error gradients of long term contributions grow exponentially more compared to short term contributions, causing the model to become unstable. Vanishing gradients occur when the error gradients of long term contributions go exponentially fast to 0, causing the network to be unable to learn correlation between temporally distant data. A popular solution to the problem of exploding and vanishing gradients is the use of a long short-term memory ($LSTM$) network. In an LSTM network the neurons are replaced by memory blocks [22]. These memory blocks are able to control the flow of information by the use of so-called 'gates'. A memory block consists of three gates: the forget gate, input gate, and output gate. The forget gate determines which information from previous time steps is important. The input gate determines which information from the current time step is important to add to the network's memory. The output passes the cell state on to the next time step.

4.4. Stacked Convolutional Recurrent Network

In recent years ConvNets and RNNs have been combined into stacked convolutional recurrent neural networks ($CRNN$). Especially in the field of sound event recognition stacked convolutional recurrent neural networks have shown promising results [1, 2, 64]. Audio recordings often contain a lot of background noise and events can occur randomly along the recordings with varying lengths. The convolutional layer is used to extract robust features from short frames and disregard the background noise by using the pooling operation. Recurrent layers are able to obtain information from a sequence of short frames. Eventually the information

is fed through a fully connected layer to classify the sound event. Using the CRNN architecture provides the opportunity to use a wide range of features that require less pre-processing.

4.5. Autoencoder

An autoencoder is a type of neural network that is able to copy the input to the output. With this ability an autoencoder can compress data while the reconstruction error is minimized. An autoencoder usually consists of three components: encoder, compressed representation, and decoder. Autoencoders are often used to denoise the input data [34]. During training of the autoencoder noise is added to the input data. For normal autoencoders the cost function is focused in minimizing the difference between the input and output. In denoising autoencoders the added noise is subtracted from the cost function, such that the cost is lower when the noise is not present in the output. Using denoising autoencoders can help to remove various types of unwanted objects from the input, such as a drone's ego-sound from attached microphones for hear-and-avoid purposes.

4.6. Evaluating Classifier

A neural network will always produce results when it is fitted to a training dataset. However, due to training flaws, such as over- and underfitting, the model might not produce desired results on new data. Several measures exist to minimize the occurrence of such training flaws, as discussed in section 4.1. To check whether training flaws have occurred it is important to evaluate model. Several metrics exist to evaluate the fitted model.

The most popular evaluation metric is the classification accuracy. The classification accuracy is the ratio between the amount of correct predictions and the total amount of predictions, as shown in equation 4.5

$$Accuracy = \frac{Correct \quad Predictions}{Predictions} \tag{4.5}$$

Using the classification accuracy as evaluation metric works well when the classes contain an similar amount of samples. However, when the class sizes are not equal a bias can arise to the largest class. The classification accuracy is a great evaluation metric for initial indication on the model's performance, but other metrics are needed to give conclusive results

Another often used evaluation metric is the confusion matrix. The confusion matrix produces a matrix which describes the model's performance. The matrix contains four important terms to illustrate the model's performance:

- True Positives: The situation in which something, an aircraft for example, is detected and is also actually present.
- True Negatives: When no aircraft is detected and no aircraft is present in the area.
- False Positives: When an aircraft is detected, but no aircraft is present in the area.
- False Negatives: When no aircraft is detected, but an aircraft is actually present in the area.

The terms used in the confusion matrix form an important basis for other evaluation metrics.
A metrics which is often used for binary classifiers is the receiver operating characteristic (ROC) curve and its corresponding area under curve (AUC). An ROC curve plots the true positive rate (TPR) against the false positive rate (FPR).

- True Positive Rate: The amount of correct positive classifications with respect to the total amount of truly positive samples.
- False Positive Rate: The amount of false positive classifications with respect to the total amount of truly negative samples.

Figure 4.7 illustrates how the ROC curve is used to evaluate a model's performance. The AUC is the area underneath the ROC curve and corresponds to the probability that the model will classify a truly positive sample as positive compared to classifying a truly negative sample as positive. A higher AUC means a better performing model.

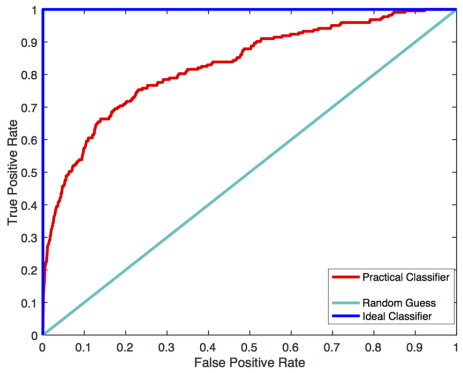


Figure 4.7: ROC curve [15]

A metric which can be derived from the ROC curve is the equal error rate (EER), also known as the cross over rate. EER refered to as the point where false negative rate (FNR) is equal to the false positive rate [37]. As $FNR = 1 - TPR$, it is possible to determine the EER by drawing a line from top left to bottom right. The FPR value where the ROC curve intersects the line is the EER value. A lower EER value is better.
The F1 score is metric that describes the accuracy of model independent of the size of the classes. The F1 score is constructed of two different terms: the precision and recall.

- Precision: The amount of true positives with respect to the total amount of positively classified samples.
- Recall: The amount of true positives with respect to the total amount of truly positive samples.

The F1 score is the harmonic mean of precision and recall, according to equation 4.6

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.6}$$

An evaluation metric often applied in multi-class classifiers is the logarithm loss. The logarithm loss metric penalizes false classifications. In a multi-class classification the classifier calculates the probability to each class for the samples. If the dataset contains N samples belonging to M classes, the logarithm loss is calculated according to equation 4.7.

$$Logarithm \quad Loss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} * log\left(p_{ij}\right) \tag{4.7}$$

In which:

- y_{ij} states if sample i belongs to class j
- p_{ij} is the probability calculated by the classifier that sample i belongs to class j

A lower logarithmic loss indicates a higher accuracy.

5
Sound Event Recognition For Hear And Avoid

Sound event recognition is a novel attribute to hear and avoid. Over the years many different algorithms for sound event recognition have been developed. In these algorithms classification using artificial neural networks has become more popular. Algorithms used for sound event recognition use different types of artificial neural networks. To develop a good working sound event recognition algorithm, the features and the classifier should be a matching combination. This chapter will discuss multiple sound event recognition algorithms which use this type of classifier.

5.1. ConvNet Classifier

Convolutional neural network classifiers are often used in sound event recognition algorithms. Different types of features can be used as input for a convolutional neural network classifier. This section will discuss sound event recognition methods that apply a convolutional neural network classifier in combination with different features.

5.1.1. ConvNet And Spectrogram Image Features

An often applied approach is the use of a ConvNet and spectrogram image features (*SIF*). In this method the waveform signal is converted to a spectrogram image representation, as explained in section 3.3.3. Mel-scaled spectrograms are often used as they zoom in on the human auditory range. These spectrogram images are fed into a Convnet for image classification. This method has both been applied for sound event detection and sound classification.
A prior research into hear-and-avoid , performed by D. Wijnker et al., has applied the combination of a ConvNet classifier and spectrograms for the detection of an aircraft fly-over. In this approach one spectrogram is created of an entire recording lasting up to 120 seconds. Each second in this recording is labeled for containing aircraft sound or not. The classifier is then trained to detect the location of the sound event in the spectrograms. Other research has applied spectrograms in combination with ConvNet classifiers to separate different sound sources. K. J. Piczak has applied a combination of spectrograms and ConvNets to classify isolated sound event recordings from the ESC-10 [43], ESC-50 [43] and UrbanSound8K [48] datasets [42]. These datasets contain short recording of environmental sound events, such as church bells, coughing, airplanes, and barking dogs. Each of these recordings has been divided into smaller segments with 50% overlap to increase the amount of data. All segments have been converted to log-scaled Mel-spectrograms. The spectrograms together with their deltas are used as a two-channel input for a ConvNet. The kernel size of the filters in the convolutional layers is rectangular (57x6) allowing for slight frequency invariances.

5.1.2. ConvNet And raw-waveform

Other research has tried to eliminate the need for pre-processing altogether. W. Dai et al. has proposed a method to classify sound using raw waveform as input [17]. W. Dai uses a fully convolutional neural network without any fully connected layers. The network proposed by W. Dai is shown in figure 5.1. The fully connected layers are replaced by one global average pooling layer. By using the global average pooling layer each feature map is reduced into a float by averaging the activation over the time dimension. Similar to the research performed by K. J. Piczak, has the model been trained on the UrbanSound8K dataset. The audio recordings have are fed into the network without any pre-processing.

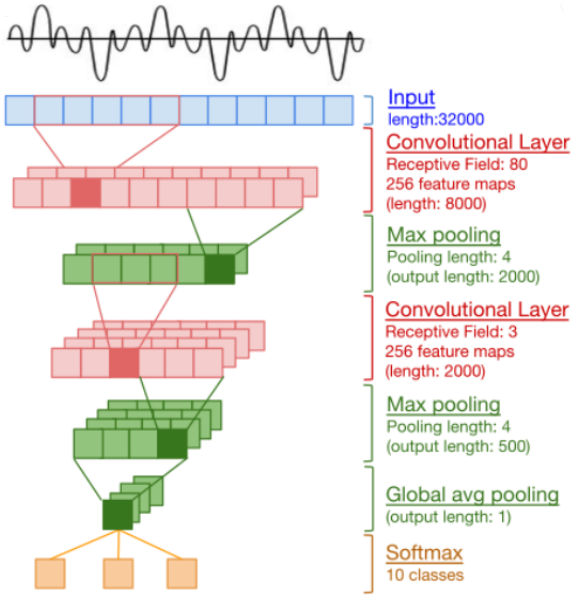


Figure 5.1: Fully convolutional neural network proposed by W. Dai et al. [17]

5.2. CRNN Classifier

An alternative for convolutional neural network based classifiers is the stacked convolutional recurrent neural network based classifiers. These type of classifiers are able to extract small-scale features using the convolutional layers and learn context between these features using the recurrent layers.
Y. Xu et al. has proposed a convolutional gated recurrect neural network based method for audio tagging. The input of the network are small time windows of 32 [ms]. The convolutional layer is used to extract robust features from the small time window. A recurrent layer containing gated recurring units ($GRUs$) is used to detect long-term information from each sound event. GRUs are similar to LSTMs, but are less complex and faster to compute. A GRU memory block only uses 2 gates, compared to three gates in an LSTM block. The first gate, the update gate, determines which information from previous time steps is important for the next time step. The second gate, the reset gate, determines of the information from previous time steps can be forgotten. Figure 5.2 shows the architecture proposed by Y. Xu et al.

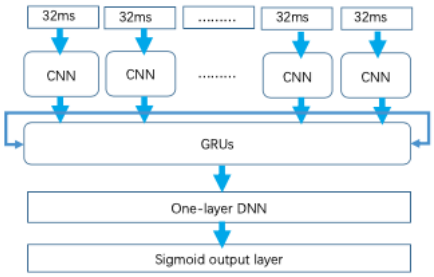


Figure 5.2: Convolutional gated recurrent neural network proposed By Y. Xu et al. [64]

The dataset used consists of seven different audio classes with 4 second sound recordings. The network is able to train on different features. The features used in this research are spectrograms, raw waveform, and mel-filterbanks.

S. Adavanne et al. have used a crnn based method to detect a bird call [2]. The network applied is a stacked convolutional and bi-directional recurrent neural network ($CBRNN$), shown in figure 5.3. The complete network contains two separate convolutional neural networks. Both of these ConvNets process a different input channel. The input to the entire logarithm is a sound signal of 10 seconds. This signal is divided into frames of 40 [ms] with 50% overlap, to a total amount 500 frames for a 10 second recording. The first feature extracted from each frame is the log mel-band energy (mbe). The mbe features are stored in a volume of 500x40x1. The second feature is the local dominant frequency ($dom-freg$) and their respective magnitudes. From each frame the three most dominant frequencies and their magnitudes in the range of 500-800 [Hz] are

extracted. The dom-freq's are stored in a volume of 500x3x2. The two volumes with the extracted sound features are fed into the network as shown in figure 5.3.

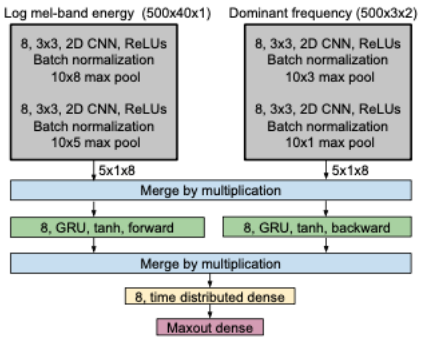


Figure 5.3: Stacked convolutional bi-directional recurrent neural network proposed by S. Adavanne et al. [2]

5.3. Data Augmentation

For training a classifier it is essential to have a large dataset. By applying data augmentation it is possible to increase the current dataset by applying small changes to copies of the current data. In addition to increasing the dataset, data augmentation can also help to increase generalization and robustness. Data augmentation for each type of data is different and research into effective forms is very active. J. Salamon and J.P. Bello have performed research into the influence of data augmentation for environmental sound classification [47]. Four different types of augmentation have been applied, resulting in five augmentation sets. The augmentation sets applied are listed below:

- Time Stretching (TS): The samples are slowed down or sped up. Each sample is stretched using four different factors: 0.81, 0.93, 1.07, and 1.23.

- Pitch Shifting ($PS1$): The pitch of the sound sample is raised or lowered. In this set the pitch of each sample is shifted by four different semitone values: -2, -1, 1, and 2.

- Pitch Shifting ($PS2$): In this set the pitch of each sample is shifted by four larger semitone values: -3.5, -2.5, 2.5, and. 3.5.

- Dynamic Range Compression (DRC): The dynamic range of the samples is compressed using four different parameterizations, three originate from the Dolby E standard [9] and one from the icecast online radio streaming server: music standard, film standard, speech , and radio.

- Background Noise (BG):The samples are mixed with recordings of real-world background noise: street-workers, street-traffic, street-people, and park. Each mix is created using $z = (1-w) * x + w * y$, where x is the original sample, y is the noise recording and w is a weighting facter which is randomly chosen from [0.1,0.5].

The augmentations were applied to the raw sound signal, after which the samples were converted to log mel-spectrograms. The network used is similar to the one applied by J. K. Piczak and has comparable performance without augmentation. The results achieved with these augmentation sets are presented in figure 5.4. It is interesting to see that the influence of data augmentation is class dependent. For some classes data augmentation has a positive effect, while for others it has a negative effect. Overall, pitch shift has the most positive influence on the classification accuracy.

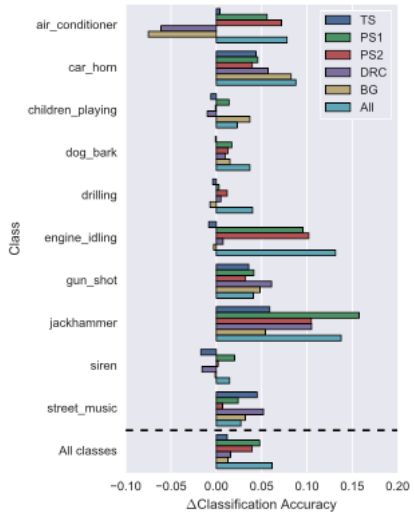


Figure 5.4: Difference in classification accuracy for each class as a function of the augmentation applied [47].

5.4. Algorithm Results

All of the approaches explained in sections 5.1 and 5.2 all show promising results. This section will compare the results of these algorithms. The results are summarized in table 5.1. Looking at the results it becomes clear that some features perform better than other no matter what classifier is used. It can be seen that raw waveform features perform less compared to the other features. It is hard to say which of these algorithms is most suitable for hear-and-avoid, further testing will be conclusive.

Table 5.1: Comparison proposed sound event recognition algorithms

Research	**Type**	**Classifier**	**Feature**	**Performance**
K. J. Piczak [42]	Classification	CNN	log mel-spec + deltas	accuracy: 73.7%
J. Salamon [47]	Classification	CNN	log mel-spec	accuracy: 79%
W. Dai [17]	Classification	CNN	raw waveform	accuracy: 71.68%
Y. Xu [64]	Classification	CRNN	spectrogram	EER: 0.110
Y. Xu [64]	Classification	CRNN	raw waveform	EER: 0.127
Y. Xu [64]	Classification	CRNN	mel-filterbanks	EER: 0.119
D. Wijnkers [60]	Detection	CNN	log mel-spec	AUC: 0.91
S. Adavanne [2]	Detection	CBRNN	mbe	AUC: 0.88
S. Adavanne [2]	Detection	CBRNN	mbe + dom-freq	AUC: 0.87

6
Sound Source Localization

Another important aspect for hear-and-avoid is localizing where an aircraft sound originates from. Within sound localization many different parameters can be determined. One of these parameters is direction of arrival (DOA). Determining DOA is defined by identifying the general direction from which a sound originates with respect to a microphone [3]. Many different techniques of determining the DOA exist. This chapter will discuss the most used techniques for DOA determination as well as employing deep learning.

6.1. Time Delay Of Arrival

A commonly used technique for DOA determination is to measure the time delay of arrival ($TDOA$) between microphones. In essence is the TDOA problem a mathematical problem [25]. Assume a microphone array consists of $N+1$ of which the location is defined by equation 6.1. The first microphone with $i = 0$ is considered as the reference microphone and is placed at $r_0 = (0,0,0)^T$. The sound source is located at $r_s \triangleq \left(x_s, y_s, z_s\right)^T$. The distance between the reference microphone and the ith microphone and the sound source are R_i and R_s, respectively, as shown in equation 6.2 and 6.3.

$$\mathbf{r}_i \triangleq \left(x_i, y_i, z_i\right)^T, \quad i = 0, ..., N \tag{6.1}$$

$$R_i \triangleq ||\mathbf{r}_i|| = \sqrt{x_i^2 + y_i^2 + z_i^2}, \quad i = 1, ..., N \tag{6.2}$$

$$R_s \triangleq ||\mathbf{r}_s|| = \sqrt{x_s^2 + y_s^2 + z_s^2} \tag{6.3}$$

Following equation 6.2 and 6.3, the distance between the ith microphone and the sound source can be calculated using equation 6.4. Using multiple microphones the difference in distance between the ith microphone and the jth microphone to the sound source is given by equation 6.5. The difference in distance between the ith and jth microphone to the sound source is often referred to as the range difference.

$$D_i \triangleq ||\mathbf{r}_i - \mathbf{r}_s|| = \sqrt{(x_i - x_s)^2 + \left(y_i - y_s\right)^2 + (z_i + z_s)^2} \tag{6.4}$$

$$d_{ij} \triangleq D_i - D_j, \quad i, j = 0, ..., N \tag{6.5}$$

The difference in distance between the ith and jth microphone to the sound source is a function function of the time delay of arrival τ_{ij}, as shown in equation 6.6 in which c is the speed of sound.

$$d_{ij} = c * \tau_{ij} \tag{6.6}$$

To localize the sound source $\mathbf{r}_s$ has to be calculated given a set if microphone coordinates and measured time delay of arrival. Note that when using $N+1$ microphone there are $(N+1)\,N/2$ different estimates of τ_{ij}.
An often used approach to estimate the time delay of arrival is the generalized cross correlation (GCC) framework [24]. The generalized cross correlation framework is build on single-path propagation of the plane wave model. The plane wave model describes a received signal as a delayed and weakened version of the original signal, which originates from a point source and is influenced by white Gaussion noise. Following the GCC framework, equation 6.7 describes the ith microphone's received signal.

$$x_i(t) = \alpha_i s\left(t - T - \tau_{ij}\right) + n_i(t), \quad i = 1, ..., N \tag{6.7}$$

In which α_i is the weakening factor due to signal propagation ($0 < \alpha_i < 1$), $s(t)$ is the reference signal, T is the delay between the sound source and the first microphone receiving the signal, τ_{ij} is the relative time delay between two microphones, and $n_i(t)$ is the added noise of the ith microphone. Using equation 6.7 in combination with the GCC framework the time delay can be estimated as follows:

$$R^{g}_{x_1x_2}(\tau) = E\left[\left(x_1(t)^* h_1(t)\right).\left(x_2(t)^* h_2(t-\tau)\right)\right] \tag{6.8}$$

$$\tau^{'} = argmax\left[R^{g}_{x_1x_2}(\tau)\right] \tag{6.9}$$

In which $E[.]$ is the expected statistical average over time, $h_1(t)$ and $h_2(t)$ are filters to improve the estimation accuracy of τ, * indicates a convolution operation, and $\tau^{'}$ is the estimate of τ. In the frequency, the GCC is related to the Power Spectral Density (PSD) φ as shown in equation 6.10. In the frequency the product of $h_1(t)$ and $h_2(t)$, $H_1(f)H_2^*(f)$, is referred to as the generalized frequency weighting function $\psi_g(f)$.

$$R^{g}_{x_1x_2}(\tau) = \int_{-\infty}^{+\infty} H_1(f)H_2^*(f)\varphi_{x_1x_2}(f)e^{j2\pi f\tau}df \tag{6.10}$$

Multiple generalized frequency weighting functions can be applied. The easiest weighting function is the classic cross correlation (CCC). CCC states that the weighting function is equal to 1. However, this results in a high estimation error in noisy environments. Other generalized frequency weighting functions include: the maximum likelihood weighting function [23], the cross-power spectrum phase weighting function [12], and the PHAT-$\rho\gamma$ [30].

6.2. Multiple Signal Classification

Multiple signal classification ($MUSIC$) is an often applied technique in sound localization. MUSIC is a technique used for determining the several parameters of multiple wavefronts arriving at a microphone array. The determined parameters include: number of signals, directions of arrival (DOA), strength and cross correlations among the directional waveforms, polarizations, and strength of noise/interference [50].
According to the MUSIC method is the signal received of an array consisting of M microphones a linear combination of the waveforms and noise, as shown in equation 6.11. In which X_m is the received signal per microphone, $\mathbf{a}(\theta_i)$ is the steering vector corresponding to the ith signal, s_i is the incident signal of the ith signal, and W_m is the noise received by each microphone.

$$\begin{bmatrix}\mathbf{x}_1(t)\\ \mathbf{x}_2(t)\\ \vdots\\ \mathbf{x}_M(t)\end{bmatrix} = \begin{bmatrix}\mathbf{a}(\theta_1) & \mathbf{a}(\theta_2) & \cdots & \mathbf{a}(\theta_I)\end{bmatrix}\begin{bmatrix}\mathbf{s}_1(t)\\ \mathbf{s}_2(t)\\ \vdots\\ \mathbf{s}_I(t)\end{bmatrix} + \begin{bmatrix}\mathbf{w}_1(t)\\ \mathbf{w}_2(t)\\ \vdots\\ \mathbf{w}_M(t)\end{bmatrix} \tag{6.11}$$

$$\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{W}(t) \tag{6.12}$$

To determine the DOA, θ_i has to be estimated. The first step is to compute the covariance matrix of the microphone arrray data, given in equation . In which $\mathbf{A}$ is the direction matrix containing the steering vectors, $\mathbf{R}_{SS}$ is the autocorrelation matrix of the signal, and σ^2_W is the noise covariance.

$$\mathbf{R}_{xx} = \mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H + \sigma^2_w\mathbf{I} \tag{6.13}$$

$$\mathbf{R}_{ss} = \mathbf{E}\left[\mathbf{s}\mathbf{s}^H\right] \tag{6.14}$$

If the microphone array consists of M microphones, then the covariance matrix $\mathbf{R}_{xx}$ is M rank square matrix whose eigenvalues (λ_k) can be calculated using equation 6.15.

$$\left|\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H + \sigma^2_w\mathbf{I} - \lambda_k\mathbf{I}\right| = \left|\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H - \left(\lambda_k - \sigma^2_w\right)\mathbf{I}\right| = 0, \quad k = 1,...,M-1 \tag{6.15}$$

Using equation 6.15 it is possible to determine the eigenvalues of $\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H$ (υ_k):

$$\upsilon_k = \lambda_k - \sigma^2_w \tag{6.16}$$

Then the eigenvectors corresponding to the maximum eigenvalues of $\mathbf{R}_{xx}$ are used to determine the subspace of the signal. The eigenvector associated with noise ($\mathbf{q}_k$) is determined using equation 6.17.

$$\left(\mathbf{R}_{xx} - \sigma^2_w\mathbf{I}\right)q_k = \mathbf{A}\mathbf{R}_{ss}A^Hq_k + \sigma^2_w\mathbf{I}q_k - \sigma^2_w\mathbf{I}q_k = 0 \tag{6.17}$$

The signal subspace is orthogonal to the noise subspace. The steering vectors that belong to the DOA lie in the signal subspace and are orthogonal to the noise subspace. By determining which steering vectors are

orthogonal to the noise subspace the DOA can be determined. To summarize the noise subspace a new matrix is formed:

$$\mathbf{Q}_w = [\mathbf{q}_I, \mathbf{q}_{I+2}, ..., \mathbf{q}_M] \tag{6.18}$$

The DOAs of multiple signals can be estimated by determining the location of the peaks of the spacial spectrum using equation 6.19. When the steering vectors and noise are orthogonal the denominator will minimize, created peaks in the spacial spectrum.

$$P_{MUSIC}(\hat{\theta}) = \frac{1}{\mathbf{a}^H(\hat{\theta})\mathbf{U}_w\mathbf{U}_w^H\mathbf{a}(\hat{\theta})} \tag{6.19}$$

6.3. Beamforming

To get a more detailed map of where sound originates from an acoustic image can be created. One of the most popular techniques is delay-and-sum beamforming. Delay-and-sum beamforming is based on the difference in phase of the emitted sound signal and the received sound signal of each microphone in an array [53]. To generate an acoustic image, first a scan grid should be determined over which the beamformer output is calculated. The microphone can be electronically steered towards each of the grid-points by applying an electronic delay (τ_n) to the signal of each of the N microphones. Beamforming is most often applied in the frequency domain due to lower computational times. The Fourier transform of delayed signal $x(t-\tau)$ is $X(f)e^{-2\pi i f\tau}$. The beamformer output of a specific grid-point for a fixed frequency f_k becomes:

$$B(\theta_s, f_k) = \left|\sum_n X_n(f_k)\, e^{-2\pi i f_k \tau_n}\right|^2 \tag{6.20}$$

with:

$$\tau_n = \frac{d}{c} sin(\theta_s) \tag{6.21}$$

if the microphone array is a line array with spacing d, which is steered in direction θ_s.
For a 2D acoustic image the microphone array has to have a 2D configuration with a 2D scan grid at a determined distance parallel to the microphone array, as shown in figure 6.1.

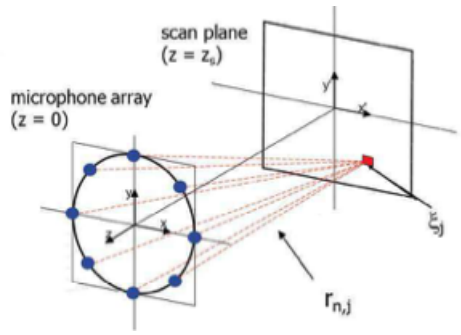


Figure 6.1: Geometry 2D beamformer [14].

The beamformer output for a 2D scan grid is often described as:

$$B(\zeta_j, f_k) = \frac{\mathbf{g}^*(\mathbf{X}\mathbf{X}^*)\mathbf{g}}{\|\mathbf{g}\|} \tag{6.22}$$

in which $\zeta_j = (x', y')$ is the coordinate of grid-point j in the scan grid. $\mathbf{X}$ is the data vector of which the components are $X_n(f_k)$ and $\mathbf{X}^*$ is the complex transpose of $\mathbf{X}$. The product of $\mathbf{X}$ and $\mathbf{X}^*$ is known as the cross-spectral-matrix (CSM). g is the steering vector of which the components are described in equation 6.23.

$$g_n(\zeta_j, f_k) = \frac{e^{-2\pi i f_k \frac{r_{n,j}}{c}}}{,} r_{n,j} \tag{6.23}$$

in which c is the speed of sound and $r_{n,j}$ is the distance from the grid-point to the nth microphone given by

$$r_{n,j} = \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2 + z_s^2} \tag{6.24}$$

(x_n, y_n) is the location of the nth microphone and z_s is the distance between the microphone plane an the scan plane.

6.4. Deep Learning For Sound Source Localization

In recent years, the application of artificial neural networks has been popular in an variety of detection and classification tasks. Also in sound source localization ANNs have shown promising in minimizing the computational efforts compared to conventional methods. In this section two different approaches for sound source localization are discussed.

The first method has been developed by Pengwei Xu et al. [63]. Pengwei Xu et al. have developed a method that computes the source strength at each grid-point of the scanning grid. Similar to conventional beamforming cross-spectral-matrix is calculated. The CSM is chosen as the input feature for the network, due to its ability to capture the structure of the microphone array signals. From each of the CSM elements only the real part is taken to convert the complex CSM into a real CSM. The network used is based on the architecture of the DenseNet-201 model developed by G. Huang et al. [25]. The network contains 201 layers. The architecture of DenseNet-201 is shown in table 6.1. Each of the conv layers consists of the sequence: batch normalization-relu-convolution.

Table 6.1: Architecture of DenseNet-201 [25]

Hidden Layers	Output Size	DenseNet-201
Convolution	112x112	7x7 conv, stride 2
Pooling	56x56	3x3 max pool, stride 2
Dense Block (1)	56x56	$\begin{bmatrix} \text{1x1 conv} \\ \text{3x3 conv} \end{bmatrix}$ x6
Transition Layer (1)	56x56 28x28	1x1 conv 2x2 average pool, stride 2
Dense Block (2)	28x28	$\begin{bmatrix} \text{1x1 conv} \\ \text{3x3 conv} \end{bmatrix}$ x12
Transition Layer (2)	28x28 14x14	1x1 conv 2x2 average pool, stride 2
Dense Block (3)	14x14	$\begin{bmatrix} \text{1x1 conv} \\ \text{3x3 conv} \end{bmatrix}$ x64
Transition Layer (4)	14x14 7x7	1x1 conv 2x2 average pool, stride 2
Dense Block (4)	7x7	$\begin{bmatrix} \text{1x1 conv} \\ \text{3x3 conv} \end{bmatrix}$ x48
Classification Layer	1x1	7x7 global average pool 1000D fully-connected, softmax

The output of the network is a vector (size Nx1), know as $\mathbf{q}$ containing the estimated source strength over the scanning grid (size $\sqrt{N}$x$\sqrt{N}$. The input data is labeled with vector $\mathbf{q}_0$ (size Nx1). Each grid point in $\mathbf{q}_0$ containing a sound source is equal to one, all other elements are equal to zero. The loss function used for training is given in equation 6.25. $\bar{\mathbf{q}}$ is the mean predicted source strength, $\bar{\mathbf{q}}_0$ is the mean true source strength , and $\mathbf{q}_{0,max}$is the maximum true source strength (1).

$$loss = \frac{\left|\bar{\mathbf{q}} - \bar{\mathbf{q}}_0\right|}{\mathbf{q}_{0,max}} \tag{6.25}$$

The most complex model constructed by Pengwei Xu et al. is able to detect a random amount of sound source up to 25. The model is only able to detect the source for one specific frequency as the CSM is constructed for one frequency bin. The error of the network does increase as the number of source increases. This increase in error is due the existence of side lobes. However, compared to conventional beamforming the NN model shows better resolution between source.

Whereas the method proposed by Pengwei Xu et al. is able to locate a sound source on, a method developed by Paolo Castellini et al. is able to localize a sound source in a grid-less area [13]. As with the method of Pengwei Xu et al., the CSM is calculated for a certain frequency. This CSM is converted to a new form, $\hat{\mathbf{C}}$, as illustrated in figure 6.2. The real values of the upper triangular part of the original CSM becomes the upper triangular part of $\hat{\mathbf{C}}$, while the imaginary part of the upper triangular part of the original CSM becomes the lower triangular part of $\hat{\mathbf{C}}$. Similar to the original CSM, the diagonal is set to zero as this contains the microphone self-noise.

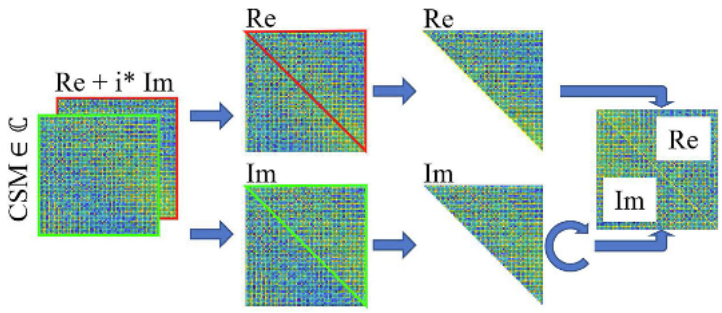


Figure 6.2: Rearrangement of CSM to move from complex-valued to real-valued data to be used as input model [13].

The model itself is constructed out of fully connected layers. The $\hat{\mathbf{C}}$ matrix is first standardized, after which it is flattened into an one dimensional array. This input array is fed into the network as can be seen in figure 6.3. The of an input layer of the network has the same size as the flattened $\hat{\mathbf{C}}$ array. The input layer is followed by four fully connected layer, which use the ReLu activation function. The output the network is an estimation of the source location in Cartesian coordinates. The output layer has two neurons to produce this output, one for the x-coordinate and one for the y-coordinate. The output of the model be in the range of $[-\infty,\infty]$, therefore the linear activation function is used.

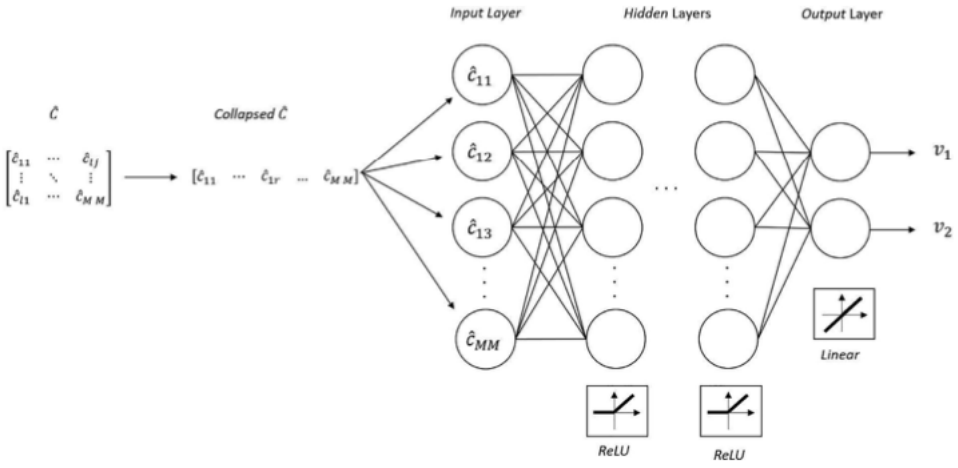


Figure 6.3: Flattened $\hat{\mathbf{C}}$ is fed into the network [13].

Unlike the method developed by Pengwei et al., the model is able to predict only one source at a time. However, the same architecture can be use to train different models to estimate the location of the strongest source, second strongest source, and third strongest source. The method proposed by Paolo Castellini et al. shows to have comparable accuracy to conventional beamforming.

7
Conclusion

This report has reviewed researches performed in the field of sound event recognition and sound source localization. As explained in the introduction, this research is performed as an addition to the larger hear-and-avoid project. The final goal of hear-and-avoid is to be able to detect and avoid other air traffic based on sound.
A vital element of hear-and-avoid is sound event recognition. Chapter 2 explained the basic principle of sound event recognition. A sound event recognition algorithm mainly consists of three modules: detection, feature extraction, and classification. Often, the detection and classification modules are combined, called detection-by-classification. In detection-by-classification each segment is classified to see if an event is present. Feature extraction is extraction of characteristics from an audio segment, which the classifier used to classify the segment.
In chapter 3 discusses the principle of sound in general, sound originating from aircraft, and representations. Sound is in essence a disturbance of air particles originating from a vibrating surface exciting the surrounding air. In aircraft sound originates from the engine and airframe. When sound is recorded it can be defined with multiple representations. Sound representations often applied for sound event recognition are: frequency representation, psychoacoutic representation, and image representation.
Sound extracted using one of the sound representations are used to train a classifier to classify a certain sound source. Chapter 4 goes deeper into applying artificial neural networks as classifier. An artificial neural network is a function approximator that an input to an output. Artificial neural networks are constructed of multiple layers each containing neurons. Each neuron contains a so-called activation function and connected to all neurons in adjacent layers. Many different types of artificial neural networks exist, but not all are suitable for all tasks. For sound event recognition convolutional neural networks and stacked convolutional recurrent neural networks are often employed for their ability to classify images and learn sequential information.
Chapter 5 takes a closer look into what sound event recognition algorithms have been proposed by different researchers. An often applied approach is the combination of spectrogram image features in combintation with a convolutional neural network. In more recent studies stacked convolutional recurrent neural networks have shown to work with different types of input, which might decrease prepocessing efforts. To increase accuracy and robustness data augmentation can be applied to the training data. Data augmentation increases the training data by applying small changes to copies of the current training data.
Next to detecting the presence of other air traffic, for hear-and-avoid it is also important to where it is located in order to avoid. Chapter 6 discusses different methods to estimate the relative location of a sound source. Sound source localization is a long researched topic with well established methods, such as: time delay of arrival, multiple signal classification, and beamforming. However, in recent years the development of artificial neural network based methods has shown promising results, which may be more suitable for hear-and-avoid.

As said before, sound event recognition and sound source localization have been popular topics of research. For sound event recognition the use of artificial neural networks has been the default for many years. However, in the field of sound source localization artificial neural network have only been applied in recent years. Therefore, the remainder of this thesis will focus on researching the suitability of artificial neural network based sound source localization for hear-and-avoid.

III

Appendices

A
Batch size selection

This appendix describes how the ANN model proposed by Castellini et al. is trained on a white noise dataset using different batch sizes. This is done to find the most optimal batch size.

A.1. Method

A.1.1. Dataset

An important component of applying an artificial neural network based approach is constructing a dataset. To create a dataset with a significant amount of data, synthetic data generator has been developed. The synthetic data generator able to perform the following tasks:

- Load the microphone array;
- Determine a random source location in the following range: x=[-25,25], y=[-25, 25], and z=50;
- Create a white noise signal originating from the source;
- Determine the received signal for each of the microphones;
- Create the cross spectral matrix;
- Convert the cross spectral matrix to a version suitable for an artificial neural network.

Microphone array

The microphone array using in this test consists of 64 microphones. The orientation of the microphone array is shown in figure A.1

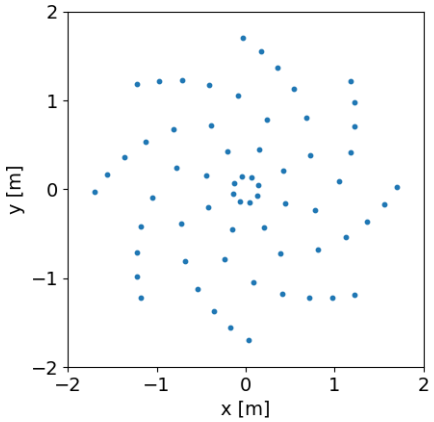


Figure A.1: Microphone array configuration used during this research.

Source location

The sound source is given a random location on the scan plane. The source location is determined with an accuracy of 0.1[m].

Generate white noise source signal

The source signal is generated as a signal with a normal distribution. The resulting signal is filtered to have a sample frequency of 40000[Hz].

Determine received signals

Knowing the orientation of the microphone array and the source location, the time delay of the signal per microphone is determined. The time delay is added in front of the source signal, resulting in each microphone's received signal.

Create cross spectral matrix

The first 0.5[s] of the received signals is used to construct the cross spectral matrix. The time domain signal is converted to the frequency domain by using the fast Fourier transform. The cross spectral matrix is made for one specific frequency. Similar to the article of Castellini et al. a cross spectral matrix is made for a frequency of 4000[Hz].

Convert cross spectral matrix

The elements of the CSM are complex valued, forming a problem to ANNs. Therefore, Castellini et al. propose the use of a converted version of the CSM that contains real valued elements. The first step to creating the converted CSM is to set the main diagonal of the CSM to 0. The CSM contains phase difference information between the received signals of different microphones in the array. The main diagonal of the CSM contains phase difference information between a signal received by a microphone and itself. This difference should be 0, therefore, the main signal should be set to 0. The next step is to convert the complex valued CSM to a real valued version. An important property of the CSM is that it is Hermitian in nature. Using this knowledge, the real and imaginary parts can be combined as follows: the upper right triangular part of $Re(C)$ becomes the upper right triangular part of the converted CSM , and the upper right triangular part of $Im(C)$ becomes the lower right triangular part of the converted CSM. For N microphones the resulting converted CSM is a real valued $N \times N$ matrix with its main diagonal set to 0. The conversion process to create the converted CSM is visualized in figure A.2. Before the input feature is fed into the network it is flattened, giving the input layer a size of N^2 neurons for N microphones. The converted CSMs are standardized to have a mean of 0 and a standard deviation of 1. Standardizing the input feature improves stability during the training process of the ANN. In total 100000 different samples are created. Each of these samples has its own point source location and synthetic white noise signal. Of the 64000 samples 16000 are used for training, 20000 for validation, and 50000 for testing.

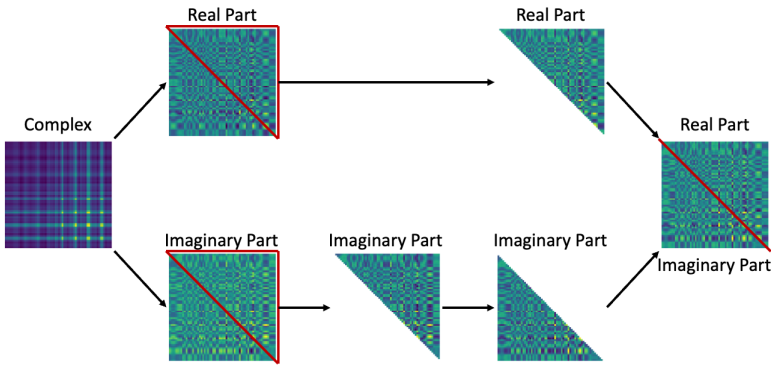


Figure A.2: Conversion of CSM as described by Castellini et al.

A.1.2. ANN architecture

The network used in this test is similar to the network described in the article of Castellini et al. The network architecture is shown in figure A.3 . The hidden layers of the network use the rectified linear unit (ReLU) activation function. The ReLU activation function is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zeros. Since the network is modeled for regression, outputs can be both positive and negative. Therefore, the linear activation function is applied in the output layer of the network.

A.1.3. Training

The model has been trained using the training and validation data sets. Different batch sizes have been tested to find the optimal training results, these batch sizes are: 32, 64, 128, 256, 512, 1024, 2048, and 4096. The

Figure A.3: ANN architecture as described by Castellini et al.

models have been trained for 10000 epochs. Early stopping is applied such that the best model during the training process is saved as a separate model. During training the mean squared error (MSE) loss function is applied. The Adam optimizer is applied as optimizer during training.

A.2. Results

Each model withe a different batch size has been trained for 10000 epochs. The training history over this period is shown in figure A.4. Figure A.4 shows that models trained using smaller batch sizes converge faster. However, training is much more unstable.

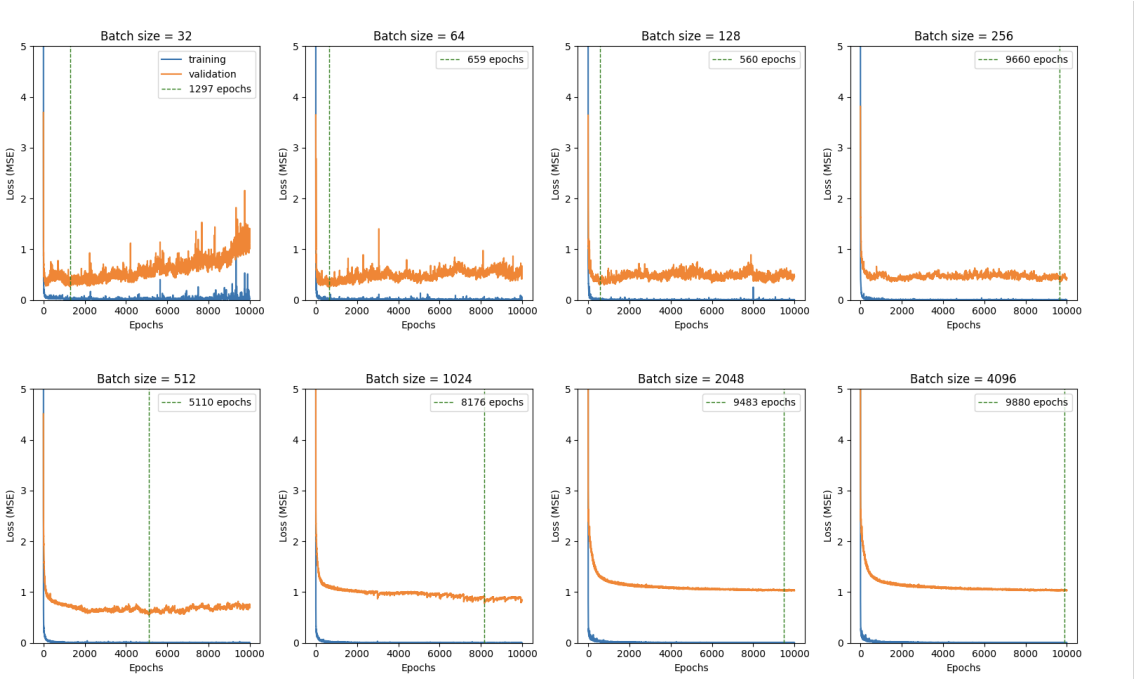


Figure A.4: Training and validation losses and epoch of lowest validation loss.

The results of the models saved at lowest validation loss are shown in table A.1. Table A.1 also shows at which epoch the model has been saved. From table A.1 it can be concluded that batch sizes of 64 and 128 produce the most accurate results. This is supported by the prediction error distributions shown in figures A.5 and A.6. Figures A.5 and A.6 show that training using batch sizes of 64 and 128 give comparable results. However, the mean sqaured error for batch size of 64 is slightly higher due to slightly larger errors. Therefore, a batch size of 128 is used during this research.

Table A.1: Mean absolute error and standard deviation of the source location prediction errors for the 5 input model and 10 input model.

Batch size	Epoch saved	Overall MSE	Overall MAE	x coordinate MAE	y coordinate MAE	Max x-error	Max y-error
32	1297	0.4382	0.2507	0.2524	0.2491	27.3	20
64	659	0.3268	0.2315	0.2356	0.2274	14.2	35.8
128	560	0.3156	0.2317	0.2324	0.2309	12.3	15.3
256	9660	0.5039	0.2360	0.2351	0.2370	24.2	31.1
512	5110	0.6276	0.2646	0.2644	0.2647	32.5	33
1024	8176	0.8267	0.3148	0.3158	0.3137	22	34.4
2048	9483	1.1511	0.3690	0.3712	0.3669	29.1	32.9
4096	9880	1.3717	0.4372	0.4410	0.4334	24.8	25.6

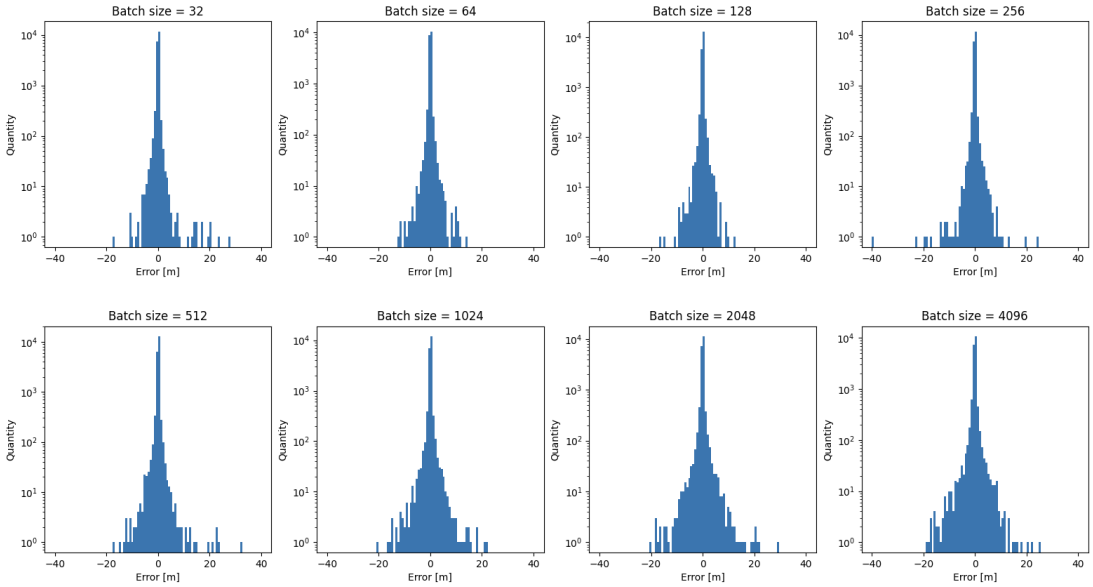


Figure A.5: Prediction error of x-coordinate using different batch sizes during training.

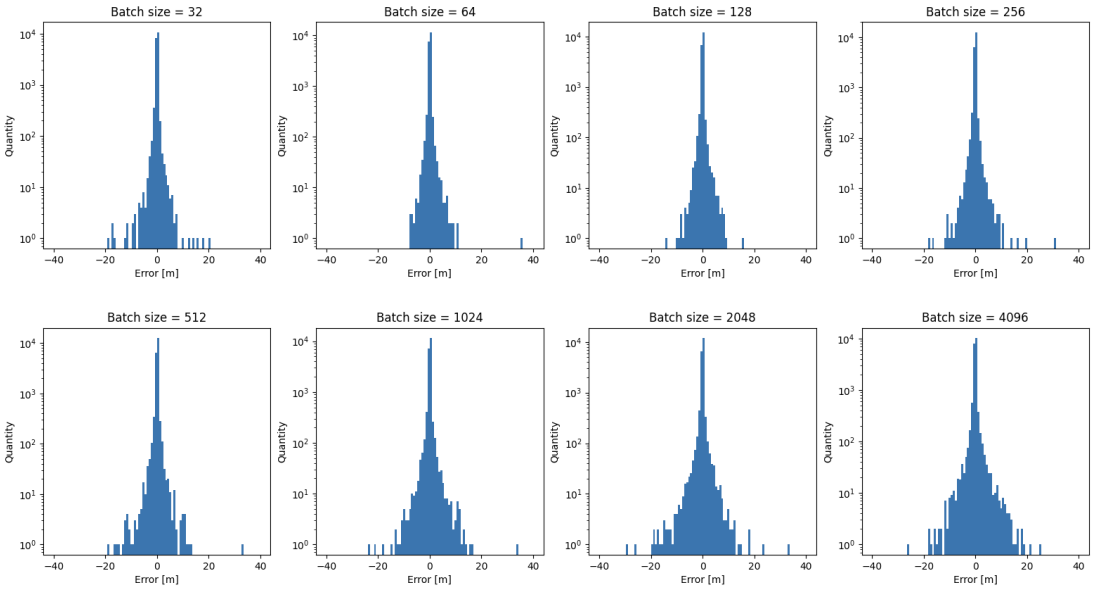


Figure A.6: Prediction error of x-coordinate using different batch sizes during training.

B
Neuron tuning

In this appendix the neurons in the model are trained to obtain more accurate results. The starting position of the neuron tuning is the architecture proposed by Castellini et al., shown in figure A.3. Table B.1 shows the different model architecture and there accuracy. The accuracy is quantified using the mean squared error and mean absolute error. The neurons in the input and output layers are bounded by the application of the network. Using 64 microphones results in an input layer consisting of 4096 neurons. As the model is used to predict the acoustic source location on a 2-dimensional plane, the output layer consists of 2 neurons. The data set used for neuron tuning is similar to the data set created in section A.1.1. Both model 11 and 12 show good results. However, the lower mean squared error of model 12 shows that there are less large errors compared to model 11. Therefore, the network architecture of model 12 is used during this research.

Table B.1: Network architecture and resulting accuracy on test data set.

Model	Hidden layer 1	Hidden layer 2	Hidden layer 3	Hidden layer 4	Overall MSE	Overall MAE
Original	200	100	50	20	0.3156	0.2317
1	256	128	64	32	0.3772	0.2384
2	128	64	32	16	0.4704	0.2389
3	256	64	32	16	0.4606	0.2402
4	256	128	32	16	0.4304	0.2416
5	256	128	64	16	0.4626	0.2313
6	512	64	32	16	0.4373	0.2399
7	512	128	32	16	0.4706	0.2316
8	512	128	64	16	0.4217	0.2412
9	512	128	64	32	0.4053	0.2403
10	512	256	64	32	0.3316	0.2388
11	512	256	128	32	0.3037	0.2222
12	512	256	128	64	0.2650	0.2268

C
Beamform plots

Figures C.1 and C.2 show the beamform plots for multiple frequencies during the same snapshot. The recording used is made during an aircraft flyover. The frequencies of which the beamform plots are shown are: 300[Hz], 600[Hz], 900[Hz], 1200[Hz], 1500[Hz], 1800[Hz], 2100[Hz], 2400[Hz], 2700[Hz], and 3000[Hz]. The beamform plots show a clear difference in main lobe and side lobes between different frequencies.

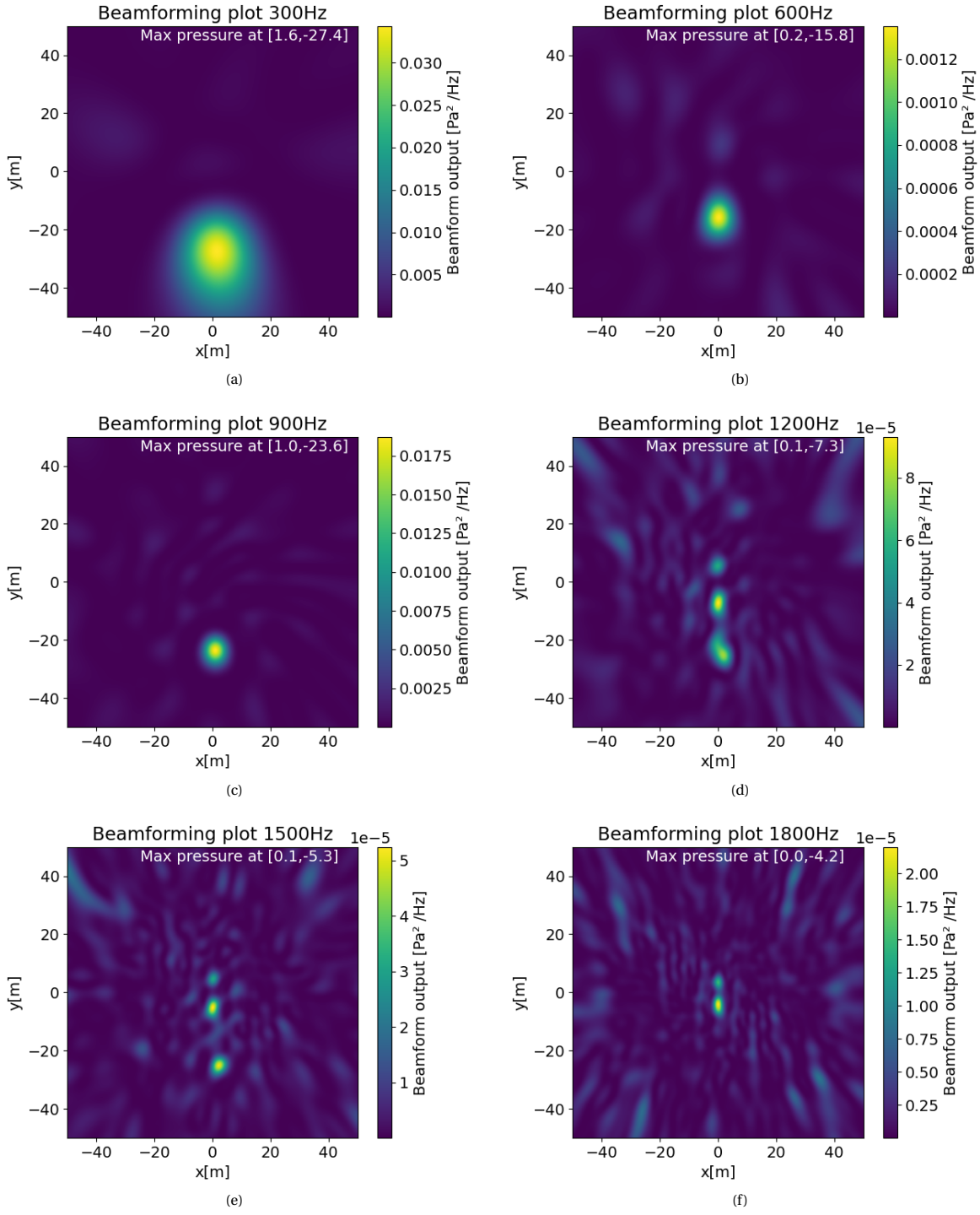


Figure C.1: Beamform plots for 300[Hz], 600[Hz], 900[Hz], 1200[Hz], 1500[Hz], and 1800[Hz]

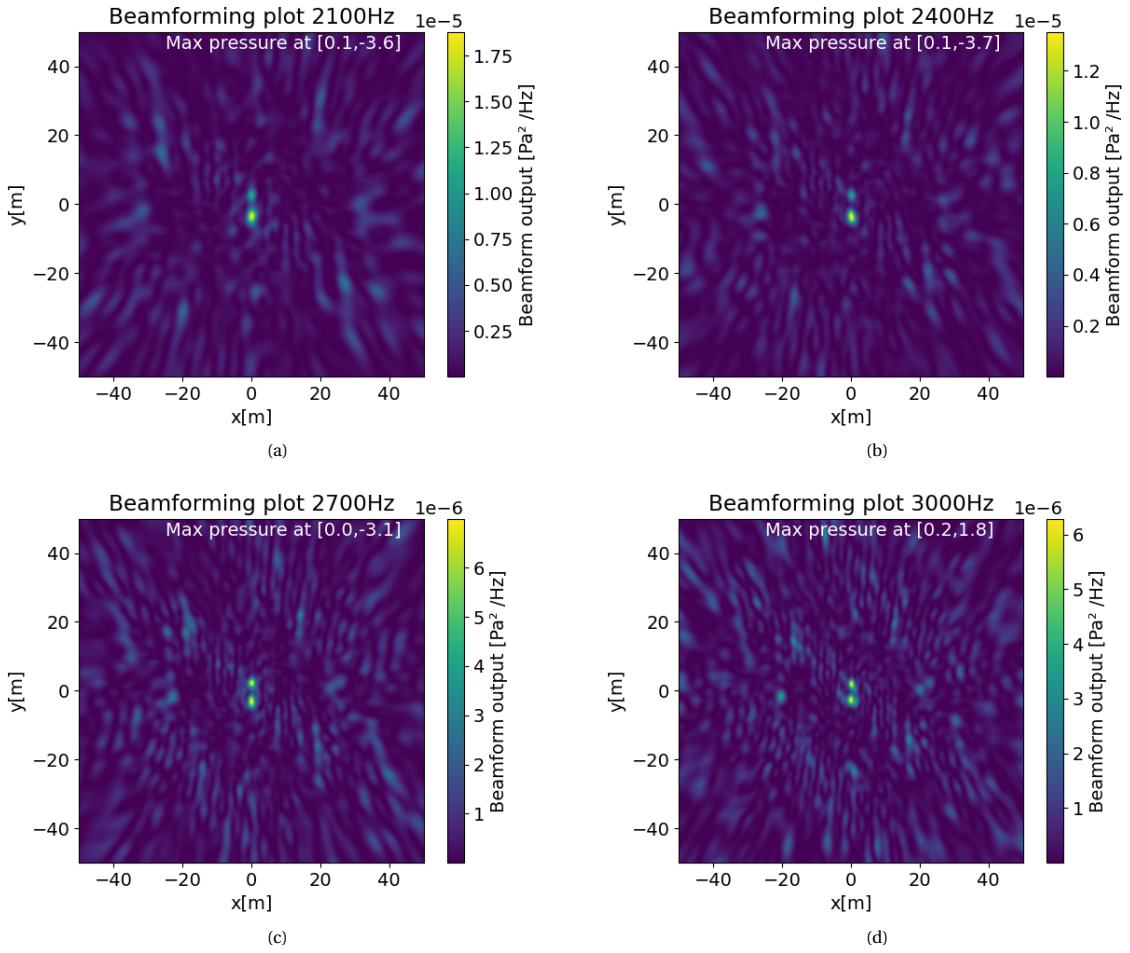


Figure C.2: Beamform plots for 2100[Hz], 2400[Hz], 2700[Hz], and 3000[Hz]

Reference List

[1] Sharath Adavanne, Konstantinos Drossos, Emre Çakır, and Tuomas Virtanen. Stacked convolutional and recurrent neural networks for bird audio detection. 06 2017. doi: 10.23919/EUSIPCO.2017.8081505.

[2] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen. Sound event detection using spatial features and convolutional recurrent neural network. 03 2017. doi: 10.1109/ICASSP.2017.7952260.

[3] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. IEEE Journal of Selected Topics in Signal Processing, 13(1):34–48, Mar 2019. ISSN 1941-0484. doi: 10.1109/jstsp.2018.2885636. URL http://dx.doi.org/10.1109/JSTSP.2018.2885636.

[4] Tiberiu Banu, Gheorghe Borlea, and Constantin Banu. The use of drones in forestry. Journal of Environmental Science and Engineering B, 5, 11 2016. doi: 10.17265/2162-5263/2016.11.007.

[5] Samantha Barry, Adrie Dane, Alyn Morice, and Anthony Walmsley. The automatic recognition and counting of cough. Cough (London, England), 2:8, 02 2006. doi: 10.1186/1745-9974-2-8.

[6] Randal Beard, Derek Kingston, Morgan Quigley, Deryl Snyder, Reed Christiansen, Walt Johnson, Tim McLain, and Michael Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. Journal of Aerospace Computing, Information, and Communication, 2, 02 2005. doi: 10.2514/1.8371.

[7] Y. Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, 5:157–66, 02 1994. doi: 10.1109/72.279181.

[8] Antoine Beyeler, Jean-Christophe Zufferey, and Dario Floreano. optipilot: control of take-off and landing using optic flow. 2009.

[9] Dolby E Bitstreams. Standards and practices for authoring dolby digital and dolby e bitstreams. 2002.

[10] Bruce P. Bogert. The quefrency analysis of time series for echoes : cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. 1963.

[11] Facundo Bre, Juan Gimenez, and Víctor Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. Energy and Buildings, 158, 11 2017. doi: 10.1016/j.enbuild.2017.11.045.

[12] G. C. Carter, A. H. Nuttall, and P. G. Cable. The smoothed coherence transform. Proceedings of the IEEE, 61(10):1497–1498, 1973. doi: 10.1109/PROC.1973.9300.

[13] Paolo Castellini, Nicola Giulietti, Nicola Falcionelli, Aldo Franco Dragoni, and Paolo Chiariotti. A neural network based microphone array approach to grid-less noise source localization. Applied Acoustics, 177:107947, 06 2021. doi: 10.1016/j.apacoust.2021.107947.

[14] Soumitro Chakrabarty and Emanuël Habets. Broadband doa estimation using convolutional neural networks trained with noise signals. 10 2017. doi: 10.1109/WASPAA.2017.8170010.

[15] Hongge Chen. Novel machine learning approaches for modeling variations in semiconductor manufacturing. PhD thesis, 01 2017.

[16] Michael Cowling and Renate Sitte. Analysis of speech recognition techniques for use in a non-speech sound recognition system. 01 2002.

[17] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. pages 421–425, 03 2017. doi: 10.1109/ICASSP.2017.7952190.

[18] Waltenegus Dargie. Adaptive audio-based context recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 39:715–725, 07 2009. doi: 10.1109/TSMCA.2009.2015676.

[19] Jonathan William Dennis. Sound Event Recognition in Unstructured Environments using Spectrogram Image Processing. Thesis, Nanyang Technical University, jan 2014.

[20] David Gerhard. Audio signal classification: History and current techniques. Technical report, A. TEMKO, C. NADEU / PATTERN RECOGNITION 39 (2006) 682 – 694, 2003.

[21] L. Gerosa, Giuseppe Valenzise, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. Scream and gunshot detection in noisy environments. 09 2007.

[22] Felix Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. Neural computation, 12:2451–71, 10 2000. doi: 10.1162/089976600300015015.

[23] E. J. HANNAN and P. J. THOMSON. Estimating group delay. Biometrika, 60(2):241–253, 08 1973. ISSN 0006-3444. doi: 10.1093/biomet/60.2.241. URL https://doi.org/10.1093/biomet/60.2.241.

[24] Saber Hoseini, Amirhossein Rezaie, and Yousef Zanjireh. Time difference of arrival estimation of sound source using cross correlation and modified maximum likelihood weighting function. Scientia Iranica, 24, 08 2017. doi: 10.24200/SCI.2017.4355.

[25] Yiteng Huang, Jacob Benesty, G.W. Elko, and R.M. Mersereati. Real-time passive source localization: A practical linear-correction least-squares approach. Speech and Audio Processing, IEEE Transactions on, 9:943 – 956, 12 2001. doi: 10.1109/89.966097.

[26] Sunyou Hwang, Jaehyun Lee, Heemin Shin, Sungwook Cho, and David Hyunchul Shim. Aircraft detection using deep convolutional neural network in small unmanned aircraft systems. In 2018 AIAA Information Systems-AIAA Infotech@ Aerospace, page 2137. 2018.

[27] Patrik Kamencay, Miroslav Benco, Tomas Mizdos, and Roman Radil. A new method for face recognition using convolutional neural network. Advances in Electrical and Electronic Engineering, 15, 11 2017. doi: 10.15598/aeee.v15i4.2389.

[28] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. Nature, 521:436–44, 05 2015. doi: 10.1038/nature14539.

[29] A.C. Lindgren, Michael Johnson, and Richard Povinelli. Speech recognition using reconstructed phase space features. volume 1, pages I – 60, 05 2003. ISBN 0-7803-7663-3. doi: 10.1109/ICASSP.2003.1198716.

[30] Hong Liu and Miao Shen. Continuous sound source localization based on microphone array for mobile robots. pages 4332 – 4339, 11 2010. doi: 10.1109/IROS.2010.5650170.

[31] Beth Logan. Mel frequency cepstral coefficients for music modeling. Proc. 1st Int. Symposium Music Information Retrieval, 11 2000.

[32] Ling Ma, Ben Milner, and Dan Smith. Acoustic environment classification. TSLP, 3:1–22, 07 2006. doi: 10.1145/1149290.1149292.

[33] Mohsen Riahi Manesh and Naima Kaabouch. Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ads-b) system. International Journal of Critical Infrastructure Protection, 19, 10 2017. doi: 10.1016/j.ijcip.2017.10.002.

[34] Erik Marchi, Fabio Vesperini, Stefano Squartini, and Björn Schuller. Deep recurrent neural network-based autoencoders for acoustic novelty detection. Computational Intelligence and Neuroscience, 2017, 09 2016. doi: 10.1155/8483.

[35] Raman K. Mehra, Jeffrey Byrne, and Jovan D. Boskovic. Flight testing of a fault-tolerant control and vision-based obstacle avoidance system for uavs. 2005.

[36] Dalibor Mitrovic, Matthias Zeppelzauer, and Christian Breiteneder. Features for content-based audio retrieval. Advances in Computers, 78:71–150, 01 2010.

[37] Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012. ISBN 0262018020.

[38] Chase Murray and Amanda Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies, 54, 05 2015. doi: 10.1016/j.trc.2015.03.005.

[39] Douglas O'Shaughnessy. Invited paper: Automatic speech recognition: History, methods and challenges. Pattern Recognition, 41:2965–2979, 10 2008. doi: 10.1016/j.patcog.2008.05.008.

[40] Razvan Pascanu, Tomas Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. 30th International Conference on Machine Learning, ICML 2013, 11 2012.

[41] V.H. Phung and E.J. Rhee. A deep learning approach for classification of cloud image patches on small datasets. Journal of Information and Communication Convergence Engineering, 16:173–178, 01 2018. doi: 10.6109/jicce.2018.16.3.173.

[42] Karol Piczak. Environmental sound classification with convolutional neural networks. pages 1–6, 09 2015. doi: 10.1109/MLSP.2015.7324337.

[43] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification, 2015. URL `https://doi.org/10.7910/DVN/YDEPUT`.

[44] S. Ravindran and David Anderson. Audio classification and scene recognition and for hearing aids. pages 860 – 863 Vol. 2, 06 2005. doi: 10.1109/ISCAS.2005.1464724.

[45] D. Ronald Fannin Rodger E. Ziemer, William H. Tranter. Signals & Systems Continuous and Discrete. Pearson, fourth edition.

[46] G.J.J Ruijgrok. Elements of aviation acoustics. VSSD, 2007.

[47] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters, 24(3):279–283, 2017. doi: 10.1109/LSP.2017.2657381.

[48] Justin Salamon, Christopher Jacoby, and Juan Bello. A dataset and taxonomy for urban sound research. 11 2014. doi: 10.1145/2647868.2655045.

[49] RTCA (Firm). SC-186. Minimum Operational Performance Standards (MOPS) for Aircraft Surveillance Applications (ASA) System. RTCA, Incorporated, 2011.

[50] R. Schmidt. Multiple emitter location and signal parameter estimation. IEEE Transactions on Antennas and Propagation, 34(3):276–280, 1986. doi: 10.1109/TAP.1986.1143830.

[51] C. E. Shannon. Communication in the presence of noise. Proceedings of the IRE, 37(1):10–21, 1949.

[52] Roneel V. Sharan and Tom J. Moir. An overview of applications and advancements in automatic sound recognition. Neurocomputing, 200:22 – 34, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.03.020. URL http://www.sciencedirect.com/science/article/pii/S0925231216300406.

[53] Dick. G. Simons. Introduction to aircraft noise. Course Reader for AE-4431 of Faculty of Aerospace Engineering, Tu Delft, Kluyverweg 1, 2629HS Delft, 2018.

[54] Ravindran Sourabh, Schlemmer Kristopher, and David Anderson. A physiologically inspired method for audio classification. EURASIP Journal on Advances in Signal Processing, 2005, 06 2005. doi: 10.1155/ASP.2005.1374.

[55] Andrey Temko and Climent Nadeu. Acoustic event detection in meeting-room environments. Pattern Recognition Letters, 30:1281–1288, 10 2009. doi: 10.1016/j.patrec.2009.06.009.

[56] Andrey Temko, Climent Nadeu, Dušan Macho, Robert Malkin, Christian Zieger, and Maurizio Omologo. Acoustic Event Detection and Classification, pages 61–73. 01 2009. doi: 10.1007/978-1-84882-054-8_7.

[57] Mark van der Woude. Acoustic-based aircraft detection and ego-noise suppression: for micro aerial vehicles. 2021.

[58] Ryan Wallace, Jon Loffi, Samuel Vance, Jamey Jacob, Jared Dunlap, and Taylor Mitchell. Pilot visual detection of small unmanned aircraft systems (suas) equipped with strobe lighting. Journal of Aviation Technology and Engineering, 7, 04 2018. doi: 10.7771/2159-6670.1177.

[59] Yao Wang, Zhu Liu, and Jincheng Huang. Multimedia content analysis using both audio and visual cues. Signal Processing Magazine, IEEE, 17:12 – 36, 12 2000. doi: 10.1109/79.888862.

[60] Dirk Wijnker, Tom van Dijk, Mirjam Snellen, Guido Croon, and Christophe De Wagter. Hear-and-avoid for unmanned air vehicles using convolutional neural networks. International Journal of Micro Air Vehicles, 13:175682932199213, 01 2021. doi: 10.1177/1756829321992137.

[61] Graham Wild, John Murray, and Glenn Baxter. Exploring civil drone accidents and incidents to help prevent potential air disasters. Aerospace, 3:22, 07 2016. doi: 10.3390/aerospace3030022.

[62] Gilbert Wu, Andrew Cone, Seungman Lee, Christine Chen, Matthew Edwards, and Devin Jack. Well clear trade study for unmanned aircraft system detect and avoid with non-cooperative aircraft. 06 2018. doi: 10.2514/6.2018-2876.

[63] Pengwei Xu, Elias Arcondoulis, and Yu Liu. Deep neural network models for acoustic source localization. 03 2020.

[64] Yong Xu, Qiuqiang Kong, Qiang Huang, and Mark Plumbley. Convolutional gated recurrent neural network incorporating spatial features for audio tagging. 05 2017. doi: 10.1109/IJCNN.2017.7966291.

[65] Nobuhide Yamakawa, Tetsuro Kitahara, Toru Takahashi, Kazunori Komatani, Tetsuya Ogata, and Hiroshi Okuno. Effects of modelling within- and between-frame temporal variations in power spectra on non-verbal sound recognition. pages 2342–2345, 01 2010.

[66] Xiang Yu and Youmin Zhang. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. Progress in Aerospace Sciences, 74:152–166, 2015. ISSN 0376-0421. doi: https://doi.org/10.1016/j.paerosci.2015.01.001. URL https://www.sciencedirect.com/science/article/pii/S0376042115000020.

[67] Tong Zhang and C.-C. Jay Kuo. Content-Based Audio Classification and Retrieval for Audiovisual Data Parsing. 01 2001. doi: 10.1007/978-1-4757-3339-6.

[68] Fang Zheng, Guoliang Zhang, and Zhanjiang Song. Comparison of different implementations of mfcc. J. Comput. Sci. Technol., 16:582–589, 11 2001. doi: 10.1007/BF02943243.

[69] Xiaodan Zhuang, Xi Zhou, Mark Hasegawa-Johnson, and Thomas S. Huang. Real-world acoustic event detection. Pattern Recognit. Lett., 31:1543–1551, 2010.