Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Human-Computer Interaction

Learning to Tag Environmental Sounds in Nightlong Audio

Master's Thesis

Mohd Saif Khan
Born Jun 20, 1993 in New Delhi, India

Matriculation Number 120803

1. Referee: Prof. Dr. Benno Stein
2. Referee: Prof. Dr. Jan Ehlers

Submission date: February 14, 2022

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, February 14, 2022

. . . .
Mohd Saif Khan

Abstract

Environmental sound events are defined as sounds occurring naturally or produced due to human activity. Devices need to identify the environmental sounds for better perception of the environment. The emergence of environmental sound classification using machine learning has led to the development of more context-aware technologies like smart homes, multimedia search, etc. In this thesis, different sound events and their starting and ending times are determined using classification models. The audio data is provided by the German Aerospace Center and is composed of night-long audio clips that are recorded near an airport and contain different environmental sounds. Many challenges like noise in data, sounds of different lengths, and their effects on the classifier are also discussed. We investigated if different models prefer to identify sounds of different lengths in the same audio. An overlapping window approach is devised to improve the identification of starting and ending times of the predicted events. The results of the thesis are encouraging as the best model is able to identify various sound events with an accuracy of 0.94 on a balanced dataset of 4 different classes. Finally, a system is also conceptualized where the environmental sounds are identified, and their spans are visualized against time.

Acknowledgements

I am thankful to Dr. Jens Kersten and Mr. Johannes Kiesel for guiding me throughout my thesis. I want to thank Prof. Dr. Benno Stein for introducing me to the world of Machine Learning. I am grateful to Dr. Susanne Bartels for her constant input regarding the data and the studies at DLR. Finally, I want to thank my friends and family for their continuous support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Goals

Environmental sound contains information about our environment and physical events. We can hear sounds from miles away, even if we do not see a sound source. It was the primary source of information broadcasting for a long time, be it bells on the church telling time, drums on a battlefield, or the emergency sirens warning people of any trouble. We still use the sirens in police vans, ambulances, or fire trucks to convey the emergency of a situation. We can recognize and differentiate individual background noise and its sources with ease. The development of processing methods that automatically extract this information has excellent prospects for various applications. The automatic environmental sound detection will make the devices in the future more context-aware [5]. Examples include searching multimedia based on audio content, creating smartphones, robots, smart homes, and more that are better aware of our environment.

There is ongoing research at the German Aerospace Center (DLR) that identifies the effect of various sounds on people's sleep. In that research, a participant's physiological signals and the room's audio are recorded and then compared with any sound event that might have disturbed the sleep during the night. On average, the generated audio is 8-10 hours long, and then every sound event during the night is tagged manually by annotators. This thesis's main objective is to identify a sound event and estimate its starting and ending time in noisy night-long audio data generated by the above research. We will also investigate the classification performance with different window lengths and if overlapping and non-overlapping window steps might favor some classes. The outcome of this thesis will save a lot of valuable resources, and the methods can be modified and used in other applications where environmental sound classification is required.

The available open-source audio datasets for research are not noisy and have balanced classes. These datasets have few individual sound events. The length of the sound events in the dataset is fixed and contains a singular sound corresponding to its label. However, the data obtained at DLR is significantly noisy due to reasons like subject noise, equipment noise, and more. Moreover, there is a significant class imbalance among the identified sound events. Some have thousands of examples, and some have examples in single digits.

The thesis aims to create an effective audio tagging system that uses a classification model. The input audio will be processed by iteratively taking a window of n seconds. The model will classify this window and predict the most prominent sound present in that window. Then, all the same predictions adjacent to each other are grouped, and each group will be tagged with a sound event. The sound event's starting and ending times will correspond to the group's starting and ending times.

The thesis further investigates on the following points:

1. *Feature representation and the suitable model*
   The thesis finds a suitable feature and model for classifying noisy audio recordings. We have done an in-depth review of various methods that have already been used in environmental sound classification. Convolutional neural networks (CNNs) have been a popular building component for audio classification models in the past. But we have used a transformer-based Audio Spectrogram Transformer (AST) [3] to classify sound events. We also look at the features that are used in those classification models. After analysis, we use a time-frequency-based feature called Mel filter bank (fbank) as input to the classification models. It follows the Mel scale [6] that mimics the frequency perception of humans.

2. *Different window sizes, Overlapping windows, and performance metric*
   The research also investigates the classification of sound signals with different window sizes. The window size is the length of an audio signal that the classification model uses as an input. Different environmental sounds can have different average lengths. An airplane has an average sound of 65 seconds in the data, but the average size of car sounds is 16 seconds. We hypothesize that because of the different average lengths of sound events, different window sizes would directly affect the classification performance for different classes.

   There may be cases where a sound starts in one window for a few seconds and then ends in a few seconds in the next window. The partial sound events in different windows can result in the misclassification of the sound class. Therefore, we analyze an overlap moving window classification approach. We hypothesize that apart from remedying misclassification, this will also improve the starting point and ending point accuracy of the target sound present in the audio.

Polyphonic sound detection score (PSDS) [7] is used as the primary metric to identify the performance of the tagging system constructed using the classification model. PSDS will check the accuracy of a label of the sound event along with its accuracy for starting and ending points. The system achieved good results on classifying airplane noises in the night-long audio recordings for the window size of 30 and 15 seconds. Different window sizes of 5, 10, 15, and 30 seconds are used to find an optimal window size. The PSDS spans between 0 and 1, where 1 represent that a system has predicted all the sound events perfectly.

3. *Task Complexity and semi-automatic decision support system*
   The training data consist of 27 different classes and follows a hierarchy presented in a later chapter. The fully automated tagging system will be too complex if conceptualized with 27 classes; therefore, only four broad classes in the higher level of ontology covering 9 out of 27 classes in the lower level are used to train the AST model. The four broad categories are Airplane, Silence, Background Noise (Neighborhood Noise/ Subject noise), and Cars. The research will discuss the challenges like noise in the recordings and variable average lengths of different classes. The research will finally help design a decision support system that tags the audio recordings and help annotators label the recordings without listening to the night-long audios.

The final results of the thesis are pretty encouraging, even though the data is very noisy. On a test recording with a window size of 30 seconds and an overlap window size of 10 seconds, the PSDS score was 0.682. It means that the sound events are predicted correctly, and their start and end times obtained are reasonably accurate. The Airplane sound class achieved an F1 score of 0.744, and the silence label achieved an F1 score of 0.902. However, the Cars and Neighbourhood noise results are not encouraging with a 30-second window and we will discuss their performance with different window sizes. The highest PSDS score of 0.708 was achieved with a window size of 30 seconds without any overlap.

We will further discuss the implications of results in later chapters in the light of formulated hypotheses. The following chapter will discuss the background of Environmental sound event classification.

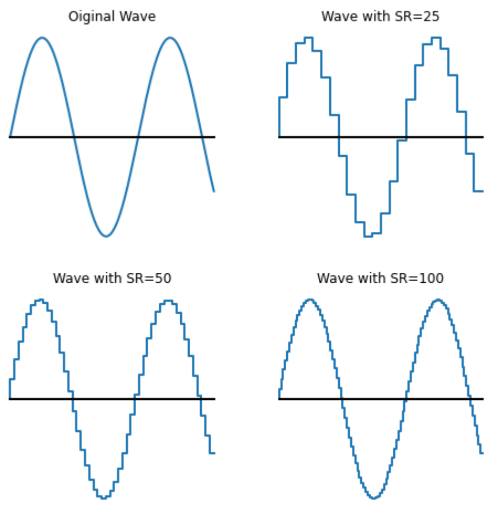
Chapter 2

Background

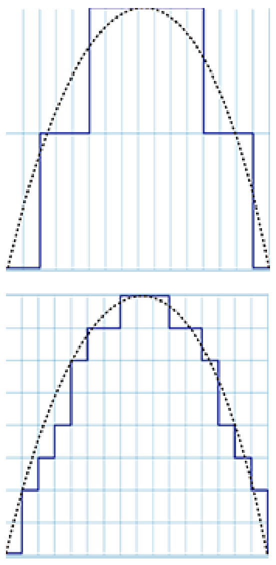
In this chapter, we will discuss various terminology in the audio domain. We will see how we store sound in modern electronic devices. We also discuss different features and the models that have already been used to classify environmental sounds. We will compare the advantages and disadvantages of various features and models.

The motivation to store sound gave birth to the first phonograph [8] by Edison in 1877. It was a rudimentary device used to store a sound sequence of a few seconds. The introduction of radios, telephones, and gramophones was a significant achievement in human history. With the improvement in technology, now we can process sound in several different ways and have numerous applications. The introduction of AI in audio analysis has created domains like Automated Audio Captioning, Automatic Speech Recognition, Sound Classification, Music information retrieval, Recommender systems, Music generation, and many more. Now audio analysis has limitless possibilities.

Sound is an analog signal, and it is stored and manipulated in modern devices as a stream of discrete numbers using Pulse Code Modulation (PCM). PCM is a de facto standard used to convert analog signals to digital signals [9]. A microphone is attached to an Analog to Digital Converter (ADC) that generates a sample. Each sample reflects the amplitude at a specific instant. The sample rate (SR) is the number of samples obtained every second. Higher sample rates can construct a sound signal more accurately and result in a higher quality of audio, as shown in 2.1a. Humans can listen to frequencies in the range of 20Hz-20KHz. Therefore, the optimal SR for humans covering all the frequencies is 44Khz, calculated using the Nyquist criterion [10]. On the other hand, Bit Depth determines the number of possible amplitude values of a sound signal. With more Bit Depth, we can store more distinct values of amplitude that result in better resolution of sound as shown in 2.1b.



(a) Original wave and its reconstruction with Sampling rate of 25, 50 and 100



(b) The upper signal with bit depth=4 and lower with bit depth=16

Figure 2.1: Sample rate and Bit Depth [1]

2.1 Audio Features

All Machine Learning tasks require the input to be converted into a vector of numbers, and these vectors are known as features. Therefore there is a need to convert the raw audio sequences to features. Various audio features have been adopted from the domain of signal processing. According to [11], in the signal domain, the audio features can be divided into three board domains: Time domain, Frequency domain, and Time-Frequency domain [12].

Some of the time domain features include

1. Zero-Crossing rate: It is the rate at which a signal changes from positive to zero to negative or from negative to zero to positive. [13]

2. Amplitude Envelope: It refers to the changes in the amplitude of a sound over time and affects our perception of timbre. [13]

3. Short-term energy: The short-term energy of a frame is defined as the sum of the squared absolute values of the amplitudes, normalized by the frame length. [14]

Some of the frequency domain features include

1. Band Energy Ratio: The Band Energy Ratio (BER) relates the lower and higher frequency bands and measures the dominance of low frequencies [11].

2. Spectral Centroid: It represents the center of gravity of a spectrum. It gives the frequency band where most energy is concentrated [13].

3. Spectral Flux: Spectral flux is a measure of how quickly the power spectrum is changing, calculated by comparing the power spectrum for one frame against that of the previous frame. [13]

Some of the time-frequency domain features include

1. Mel Filter Banks: The Mel scale attempts to replicate the nonlinear sound perception of the human ear. Human ears are more discriminative at lower frequencies, whereas at higher frequencies, they are less discriminative. Mel filter banks solve this by giving a better resolution at low frequencies and vice versa [15].

2. Mel-frequency cepstral coefficients (MFCC): It represents the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency [16].

3. Constant Q transform: It is very closely related to the Fourier transform. Like the Fourier transform, a constant Q transform is a bank of filters, but in contrast to the former, it has geometrically spaced center frequencies. [17]

2.2 Related Work

This section will discuss different features and the models that have already been used to classify environmental sounds. We will compare the advantages and disadvantages of various features and models. We will also discuss challenges like noisy data, different audio lengths, etc., and how other authors overcome those challenges.

The authors in [18] developed an approach for monophonic sound signals where the audio pattern recognition is done using the bag-of-frames (BOF) that are computed from the audio signals. They cut the signal into frames and calculated MFCCs. Then they model the distribution of MFCCs over the Gaussian Mixture model and obtain a probability density. Then they estimated the Kullback-Leibler (KL) divergence with Monte Carlo between different probability densities. The authors in [19] also used the bag-of-instances approach directly over the spectrograms. Then they applied clustering algorithms like KNN and k-medoids clustering to create bags of features and finally applied Support Vector Machine (SVM) to each class.

The Hidden Markov Model has been a popular algorithm for sequence modeling as it provides a simple and practical framework for modeling time-varying vector sequences. Most of the initial ASR systems were based on HMM [20], but they were also used for audio classification. In [21] the author has used HMM for sound event detection. The MFCCs were extracted as features, and then the author created a system to classify sounds using HMM. In [22] the author has used features like MFCC, LFCC (Linear Frequency Cepstral Coefficient), and LPC (Linear Predictive Coefficients) in a smart room environment setting. This system also used HMM and was designed to be a context-aware system.

CNN has been used majorly in the classification of audio signals. In [23] and [24] the authors have used Fully Convolutional Neural Network(CNN) and I-vectors and successfully applied them to the scene classification dataset. [24] computes the I-vectors by taking MFCC's as input features and extracting acoustic information in a fixed-length and low-dimensional format. [23] further improved the model by training four different models(1 i-vector and 3 CNN) at a time and then calibrating and averaging the predictions. In [25] the author has used scalograms as features. They are extracted from raw signals with simple wavelet transforms and perform better than Mel energy features as they are time-shift invariant.

The author in [26], has used transfer learning from the ImageNet-Pretrained standard deep CNN models like ResNet [27], DenseNet [28] for audio classification.They used a three-channel augmented Mel spectrogram as the main feature and obtained good results on ESC-50 and UrbanSound8K datasets. In [29], the author creates a new feature called Spectrogram image feature (SIF) that will convert arbitrary length spectrogram into denoised and smoothed image blocks. Then these blocks are trained on CNN to get a classification model.

In [30], the authors applied the inception v4 network for the classification of bird sounds. They converted the snippets of bird sounds into an RGB image. Three log-spectrograms were calculated using fast Fourier transform with three different window sizes of 128, 512, and 2048. Then they concatenated the spectrogram and obtained an RGB image. They also applied Time and Time-Frequency Attention mechanisms and transferred Inception-v4 from the image to the audio data. They achieved the highest results in all classes at the BirdClef2017 challenge.

One of the drawbacks of CNN is that they are not capable of modeling sequences. Authors in [31] have used deep Gated Recurrent Unit (GRU) based

Recurrent Neural Network (RNN) architecture to model the sequences. The audio scene is transformed into a sequence of label tree embedding (LTE) high-level feature vectors to make the training with RNN more efficient. Initially, the low-level features like MFCCs are extracted and then transformed into high-level LTE features. The author also experimented with other low-level features like 64 Gammatone cepstral coefficient [32] and log-frequency filter bank coefficients. The main drawback of this approach is that it is computationally expensive for a large dataset [33]. RNNs are also not efficient in processing long sequences; therefore, only a few seconds of audio snippets can be classified efficiently.

In [34], the author has proposed a scheme to use the best of both worlds by taking CNN and LSTM layers parallel and combining the results of both to get a prediction. The LSTM uses MFCCs to extract the temporal information from the signal, whereas the CNN uses SIF as features as used in [29].

Another disadvantage of CNN is its handling of variable-length audio sequences. In [35], the author tried to classify variable-length audio using CNN and Global Masked pooling. The sequences are usually padded to make their length equal. Although the pooling operation does not add any information, it takes many resources. It is problematic if the lengths of various audio events have considerable differences. Then we have two options either pad or cut the sequence, and cutting the sequence will result in loss of important information.

Most of the datasets used to classify sound events are composed of audio snippets that are clear to hear and contain the sound of only one class. Therefore the models require a vast amount of data and can easily overfit [36]. Even if they have a slight noise in some snippets, the variance of the classifier is less to identify the sound correctly. Therefore, the data is augmented to make the models more immune to noise. In [36], the authors have used Time Warping, Frequency masking, and Time masking. Authors in [30] used image augmentation techniques on the audio data and [37] used speed perturbation that creates a shift in the frequency component of the sound signal. Therefore a little noise in the data is beneficial for the model

In real-world scenarios, we cannot avoid unnecessary noise in the audio data. In [38], the author has proposed a system called SeqUential Self TeAchINg (SUSTAIN). The author says that using a weakly labeled dataset makes learning the sound in noisy conditions harder. Weakly labeled audio recordings are only tagged with the presence or absence of sounds without any temporal information about the event. The system is a sequential self-teaching, based on the teacher-student framework for improving learning over time.

In [39], the authors have used a completely different approach for recognizing harmonic sounds. They created a mask estimation algorithm that identifies spectral regions which contain reliable information for each sound source and then apply bounded marginalization to treat the inconsistent feature vector elements. They use log energy differences between spectral subbands as their features. In [40], the authors wanted to identify the pornographic content using the audio using Radon transform as their feature. They finally used a GMM to distinguish pornographic sounds. The model performed well and can be used as an alternative to image-based methods.

Recent advancements in the transformer architecture have also spilled into audio processing. The transformers were first introduced to better the Natural Language Processing (NLP) tasks. In [41], the author has put a transformer block on top of a CNN for Sound event detection. In [42], the authors have proposed a CNN-transformer model that takes log Mel spectrogram as input features. In [43], the authors have presented a novel method where they have added a convolution module in the transformer encoder block outperforming previous CNN-Transformer models and RNN models.

Audio Spectrogram Transformer (AST) presented by [3], is a pure attention-based, convolution free model to classify sound events. It can be directly applied to audio spectrogram and capture long-range global context. They also transferred the learning from Vision Transformer (ViT) that is pretrained on ImageNet. It achieved state-of-the-art results in Audioset, SpeechCommands, and ESC50 datasets.

In the next chapter, we will discuss the data and the methodology used to classify the sound events.

Chapter 3

Experimental Setup

This chapter will dive into the data, methodology, and evaluation metrics used for the final experiments. In section 3.1, the data obtained from the DLR in different studies is discussed. There is a discussion about the quality and quantity of data and the different classes present in 3.1.1 and 3.1.2. The challenges related to the night-long audio recordings are discussed in 3.1.3. Then the extraction of audio snippets of different classes from the recordings is discussed. It is followed by identifying the broad classes and creating the final dataset from the broad classes in 3.1.5 and 3.1.6.

In section 3.2, a methodology to achieve the final goal of the thesis is presented. In chapter 1, we want to investigate three questions that will help to create an audio tagging system. We have discussed the experimental setup for those three questions.

1. In sections 3.2.1 and 3.2.4, we will discuss the appropriate feature, baseline methods, and the primary classification method used to obtain the classification of audio snippets.

2. Sections 3.2.4 and 3.2.6 will discuss the thesis's hypothesis about the different window lengths and overlapping window approaches. We will also discuss the methods that will evaluate those hypotheses.

3. In section 3.2.5, we will look at the method to tag the classified audio snippets. The chapter ends with shedding some light on the evaluation metric in section 3.3 that is used to measure the performance of the final system.

3.1 Data

The DLR Institute of Aerospace Medicine collected the data as part of 3 different studies. We have combined the data from all three studies to create a single final dataset for evaluation. The institute is investigating nocturnal aircraft noise effects on sleep with polysomnography and an event-related analysis. They conducted three different studies: STRAIN, MIDAS, and Fluid21, the latter being the most recent. The First study was STRAIN, and it was done between 1999 and 2004. The second study, MIDAS, applied the same methodology to a children population and introduced new detailed sound event categories. The FluiD-21 study data come from a sample of the elderly where they investigated the effects of nocturnal aircraft noise on their sleep. They wanted to see differences in the effects between a middle-aged population, children, and the elderly. The FluiD-21 study tried to condense the sound event categories, e.g., they do not differentiate between different participants' sounds (coughing, talking, etc.), and tag it in a single category, "participant's sounds". The same applies to some neighborhood sounds.

They invited participants who consent to sleep in a controlled environment. They were given initial information about the study, where their physiological data and every sound they produced would be recorded. The participants gave written consent to contribute to the study. There was no violation of privacy as the participant's personal data was not disclosed.

The participant has to wear probes that are responsible for taking physiological data. Probes are attached to collect EEG with EMG and EOG, ECG, finger pulse oximetry, and breathing. For each night, the system records physiological and audio data (inside and outside the room). In addition, they also measured blood pressure for 24 hours during the previous day. The audio data is collected using two sound recorders, one is placed inside near the ear of the participant, and the other is outside the room. According to the study, participants have to sleep for about 8-10 hours, and the data is continuously recorded. Usually, a participant must sleep for one week, where the first night is the adjustment night. Sleeping with multiple medical probes on the body is hard, and for a night of good sleep, the participants should get used to the probes first. The data on the first night is recorded, but it is not used in the research. From the second night onwards, that physiological data is analyzed to check the variations due to environmental noise.

In this thesis, we will be working on the data recorded inside the room, which is more precise than the data outside the room. It records the sound events that are audible to the participant during the night. In Fluid-21, if the participant wakes up during a sound event, that sound event is accepted in the annotation file. Otherwise, it is discarded. So even if the sound event is present but did not disturb the participant will not be included in the final event file. In other studies, every sound event is saved in the annotation file.

3.1.1 Technical analysis

The STRAIN study used NC10, Cortex Instruments as a sound level meter. The acoustic equipment between the MIDAS and the FluiD-21 study was identical, and Class 1 sound level meters by NTi XL2 were used. One microphone was always placed outside and one near the ear to record the sounds perceived by the participant. The data was recorded with a sample rate of 24 kHz for 8-10 hours. The low sample rate is chosen to decrease the size of recordings on the disc. Apart from audio, two sound pressure parameters: LAFmax_dt (dB) and LASmax_dt (dB), were also recorded with a frequency of 10Hz. The total data collected during the three studies was about 235 GBs. The MIDAS had around 62 GBs, the Veu had around 137 GBs, and the Fluid21, had around 35GBs. The audio snippets extracted for all the classes based on the event files was around 88 GBs. The final data used by classification models was around 35.5 GBs.

3.1.2 Numerical analysis

There was a total of 27 different sound events from the three datasets. There was a vast difference in the number of sounds in different classes. The highest number of recordings was from car class came out to be 23448. The mean duration of the recordings for different classes varies greatly. The total number, sum of their duration, and average length of each class can be seen in table 3.1

3.1.3 Challenges with the data

Noise

There is only background noise in some moments, and in others, there can be a mixture of multiple sounds simultaneously. For example, there might be some crackling sounds from the room floor, airplane noise, or the subject noise simultaneously. In some recordings, the microphone is faulty and has pulsating noise every few seconds. Sometimes the participants probably used a fan or air-conditioning, so other noise got muffled. Some participants snored during the night, and others were quiet. In some recordings, the audio of the sound events was crystal clear, but in some cases, the sound intensity was extremely low. This noise can result in wrong classifications by the classifier.

Class	Count	Duration	
		Total (hh:mm:ss)	Mean (s)
Auto	23448	107:37:56	16.52
Probandengeraeusche	14061	139:28:02	35.71
Nachbarschaftslaerm	7088	51:36:58	26.22
Umdrehen im Bett	6920	39:48:37	20.71
Raumknacken	4848	6:27:14	4.79
LKW	3956	26:40:59	24.28
Transporter	2475	13:59:02	20.34
Flugzeug landend	1821	33:32:15	66.3
Flugzeug startend	1633	32:45:47	72.23
Flugzeug	1109	19:54:25	64.62
Vogelgezwitscher	921	13:03:49	51.06
Motorrad	452	2:23:46	19.08
Autobahn	450	7:15:12	58.03
Fahrzeugkolonne	318	3:18:08	37.38
Messung Start	199	0:10:22	3.13
Messung Ende	197	0:18:16	5.56
Gueterzug	169	2:46:34	59.14
Personenzug	105	1:29:18	51.03
lauter Regen	103	1:47:23	62.55
Aufstehen (Toilettengang etc	99	3:04:30	111.82
Husten Raeuspern	90	0:14:50	9.89
entgegenkommende Auto	84	0:33:07	23.65
Flughafenbodenlaerm	84	1:45:10	75.12
Sirene (Polizei, Notarzt Feuerwehr)	61	0:26:45	26.31
Strassenbahn	28	0:07:37	16.32
Gueterzug (langsam fahrend)	25	0:29:25	70.6
Bahn (Rangierfahrzeug etc	21	0:13:17	37.95
entgegenkommende Gueterzug	12	0:08:49	44.08
entgegenkommende Personenzug	11	0:14:28	78.91
Personenzug (langsam fahrend)	5	0:02:59	35.8
Personenzug (bremsend)	2	0:01:33	46.5
Gueterzug (bremsend)	1	0:00:26	26
Gewitter	1	0:00:21	21
Schnarchen	1	0:01:21	81
Wind	1	0:00:11	11

Table 3.1: Information of different classes corresponding to data of all the DLR studies. The table contains the total duration (hh:mm:ss) of each class extracted from all the nightlong audios. For every class, the mean duration is also given in seconds.

Weak labeling or misclassification

Since different people did the annotation, there is specific subjectivity to each annotator's perception. Sometimes the sound event is tagged from the very onset of the event, but sometimes it is tagged only when it is heard significantly. Therefore it was seen that the starting and the ending points are weakly labeled. Sometimes same sound events seem different to different people, and it is challenging to differentiate between them. For example, a car event and a truck event may sound similar. Therefore there is some misclassification of labels due to human error. The annotators also have to keep track if a sound event is disturbed by another sound event. For example, if airplane sounds are combined with a subject noise, indicating that the subject is disturbed by an airplane sound and vice versa.

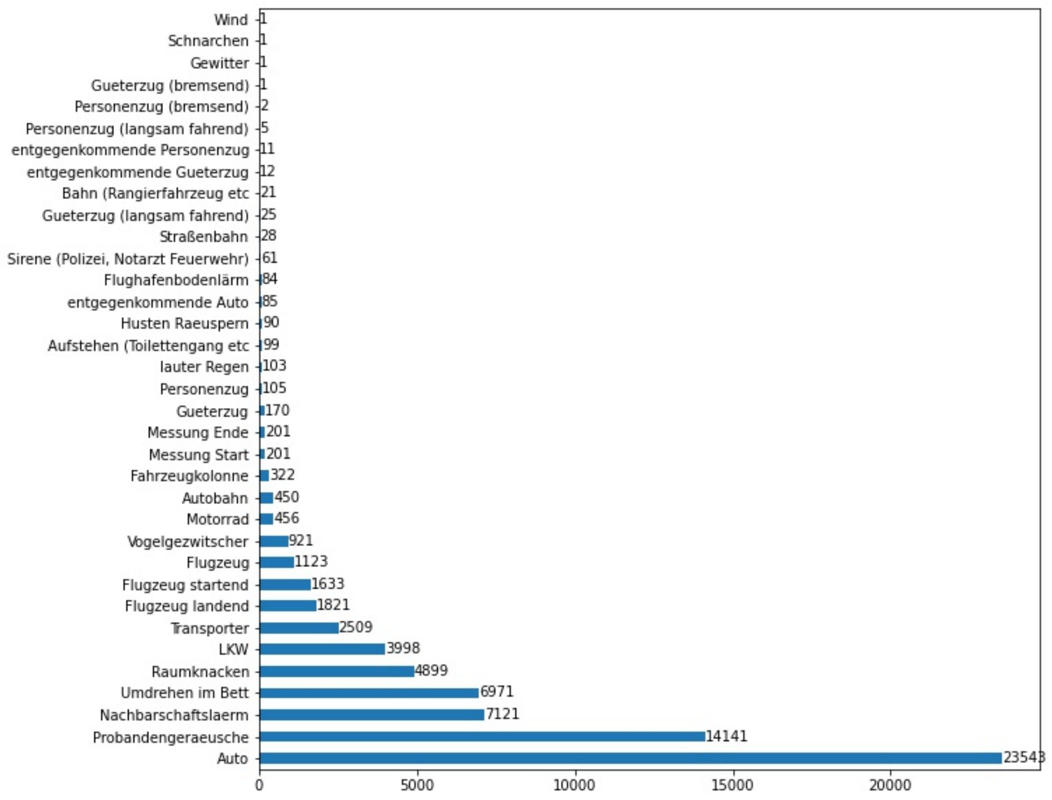


Figure 3.1: A horizontal bar plot of the class presented in the Table 3.1. The figure illustrates a high class imbalance in the dataset.

Class imbalance

There are 27 different class categories in which the data is annotated. There is a significant class imbalance among the identified sound events. Some have thousands of examples, and some have examples in single digits. Therefore it was necessary to create a class ontology in which classes are grouped based on similar characteristics. The class imbalance can be seen the figure 3.1. The thesis tries to group different sound classes into broad categories as seen the figure 3.2.

Average audio lengths of different classes

There is a big difference in the average lengths of various sound classes. The average length of an airplane sound was about 65 seconds, whereas the average sound of cars was about 16 seconds. The length of subject noise differs significantly from one recording to other. The subject may sometimes sneeze or cough, which creates an instantaneous noise for 2 seconds. Sometimes, the length may run into tens of seconds when the subject speaks. We have hypothesized that different classifiers can favor sounds of different lengths. If all the sounds in the dataset were equivalent, we wouldn't need to experiment with multiple models.

3.1.4 Extraction of sound events from audio files

The data is annotated and stored in a file with various sound events along with their starting and ending times, and it will be called the annotation file in the rest of the text. The time mentioned in the annotation file is when the event occurred. For example, if the audio recording starts at 11:30 PM and an airplane passes at 11:45 PM for 1 minute, then the starting time of the sound event will be 11:45 PM, and the ending time will be 11:46 PM. This 1-minute snippet will be extracted and stored in a folder containing similar events. A sample annotation file can be seen in A.1. The audio recording must be processed following the time stamps mentioned in the annotation file. Otherwise, there will be a mismatch between the actual and the extracted sound events. The snippet of the sound event is extracted and stored in a folder corresponding to its label. Therefore each folder will correspond to labels mentioned in the annotation file and contain snippets of audios of that label. Sometimes the labels have different spellings in different annotation files. For example, the label 'Probandengeraeusche' is also spelled as *'Probandengeräusche'* in some annotation files.

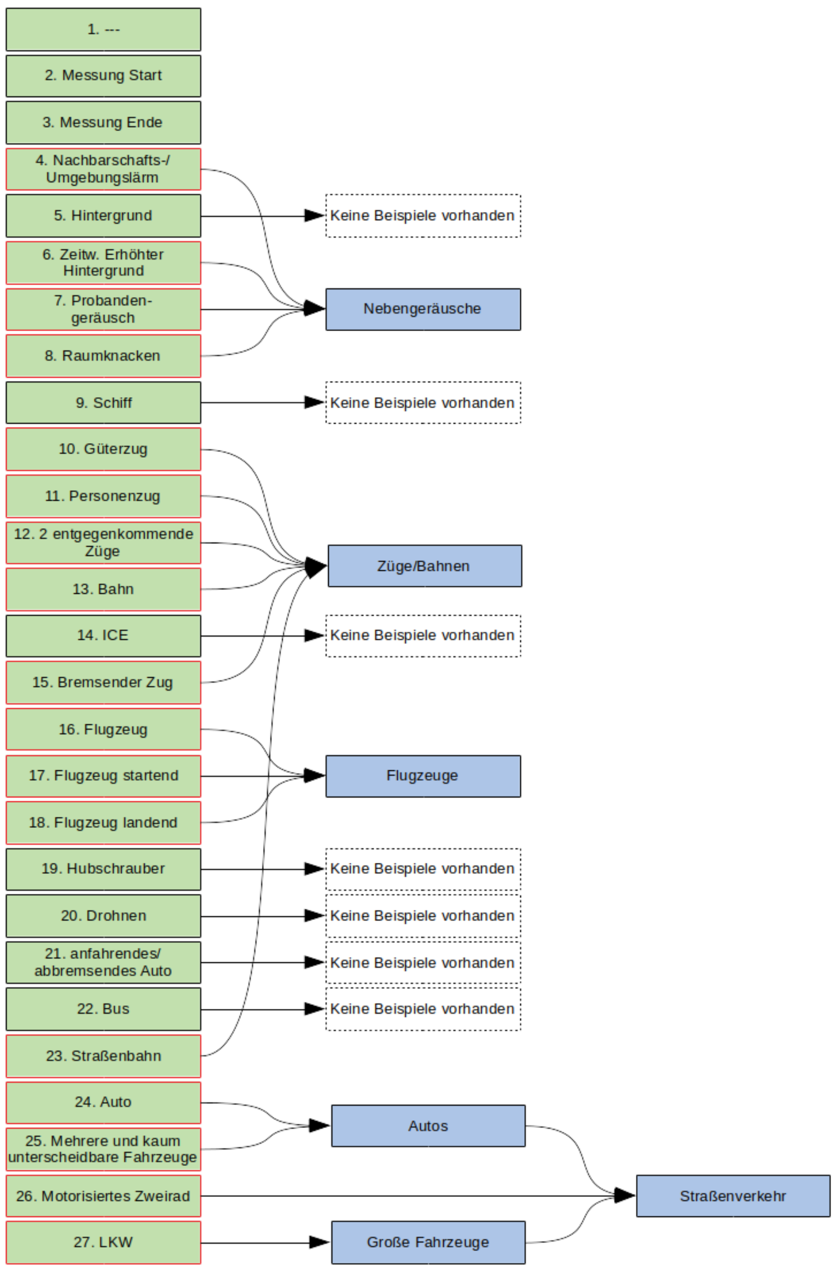


Figure 3.2: Class Ontology [2]. The green boxes in the figure represent the 27 lower-level classes. The blue boxes represent the classes at a higher level. The classes in lower levels are combined to create higher-level classes. The classes that are not connected with the arrows are not required to be classified.

3.1.5 Identification of broad categories in the data

At the beginning of the project, the DLR Institute of Aerospace Medicine in Cologne wanted to differentiate between over 40 classes. It was apparent that the number of classes was too large, so the classes were reworked [2]. The existing classes were partially combined, and some new classes were created that were considered useful. In this way, the 27 classes were created that can be seen in figure 3.2. It can be seen from the ontology that all the airplane sounds(Flugzeug, Flugzeug landend, and Flugzeug startend) were grouped into a single class as they sound similar. The other major group was Strassenverkehr, composed of cars, motorcycles, trucks, and other land vehicles. All the ICE, trams (Strassenbahn), freight train (Entgegenkommende Gueterzug), and more are grouped as Trains. The subject Noise (Probandengeraeusche), Neighborhood noise, and room cracking (Raumknacken) sounds were grouped as Background noise (Nebengeraeusche).

Some sounds like starting alarm (Messung Start) and ending alarm (Messung Ende) are not grouped as they are not required to be identified. This thesis considers only three significant groups for classification that can be seen in Table 3.2. Since the classifier has to differentiate the three groups from the rest of the remaining sounds in the recording, we also include a Silence class. The Silence class is extracted from the recording where the annotators did not tag label and are considered as void of any sound. So the final classes that are considered for classification are Airplane (Flugzeug), Silence, Nebengeraeusche (Neighborhood Noise/ Subject noise), and Cars (Autos).

Class	**Count**	**Duration**	
		Total (hh:mm:ss)	**Mean (s)**
Autos	23448	107:37:56	16.52
Flugzeug	4563	86:12:27	68.01
Nebengeraeusche	25997	197:32:14	27.35

Table 3.2: This table represent the broad categories which are taken from the higher levels of the ontology as seen in figure 3.2. These 3 classes have the highest individual samples among other classes in the same level.

3.1.6 The Final dataset

As we will consider only three classes, we took out all the examples of the sound events that belong to these classes. We can see that there was still a vast class imbalance among the class in Table 3.2. The models used for classification can only process a fixed length of an audio clip at a time, and the mean duration of all the classes differs significantly. Therefore it would be difficult to accommodate all the data samples using one standard window. We hypothesized in the introduction that different window sizes might favor different classes due to their average lengths. We will classify a maximum window size of 30 seconds (discussed in 3.2.4) apart from other window sizes of 5, 10, and 15 seconds. Therefore, all the audio files extracted from the long audio files shorter than the window size were either cut into the appropriate window size or padded to match the window. For example, if a window was twice an audio clip size, we cut it into the audio into two separate audios and included both in the final dataset.

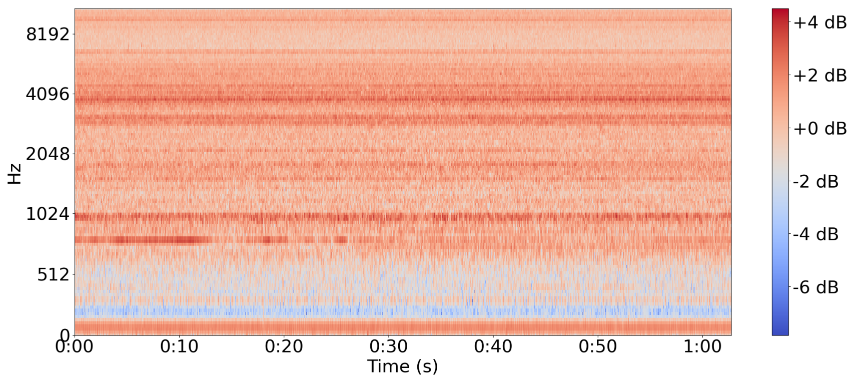
As we decrease the window size, the number of examples increases as more smaller-length snippets will be available in the dataset. After cutting and padding, we look at the lowest number of examples in each class. Then we take an equal number of examples from other classes to create a balanced dataset. For the window size of 30 seconds, we have a total of 50200 examples (12550 instances for each class).

3.2 Methodology

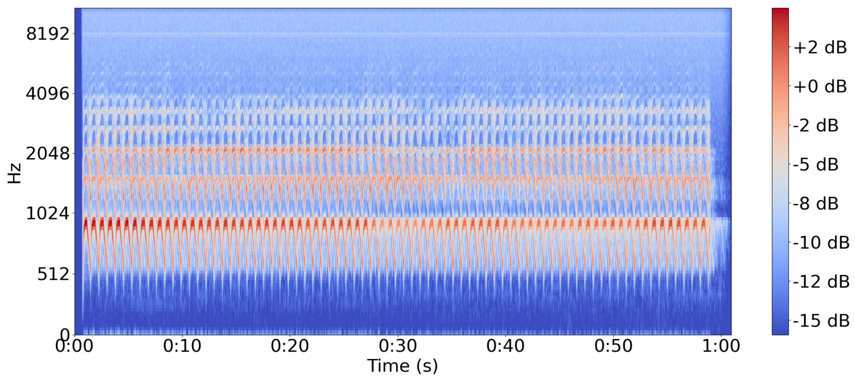
In this section, we will discuss the methodology we take to achieve our final results. We will start with the feature used for classification. Then we will discuss the baseline models and the primary classifier used. Then we will explain the setup of different experiments to validate our hypotheses and evaluate the end system.

3.2.1 Feature

In the section 2.1, we saw many audio features that can be used as input features in a classification model. These features can be used to identify audio components(like frequency, sound intensity, etc.) responsible for differentiating various sound events. We have decided to use Mel Filter banks (fbank) as the main feature in this thesis. The fbank contains information about both time and frequency components of an audio signal; therefore, they are used widely in audio research [3], [31], [41], [42]. The Mel scale [6] tries to replicate the nonlinear sound perception of the human ear. Human ears are more discriminative at lower frequencies and less discriminative at higher frequencies. The filter banks are computed by applying triangular filters in a Mel-scale to the power spectrum, and the frequency bands are extracted. The Mel Filters



(a) Mel Filter Bank representation of some background noise



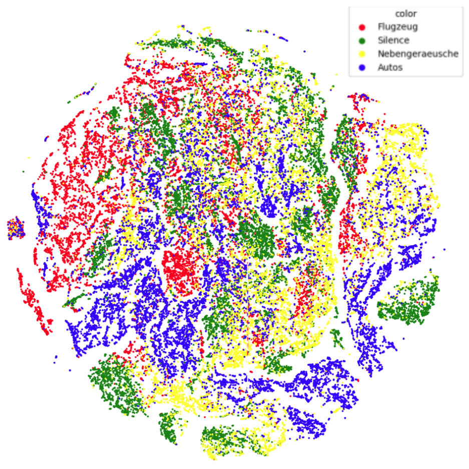
(b) Mel Filter Bank representation of a police siren

Figure 3.3: Mel Filter Bank representation of 2 different sound events. The x-axis represent the time (seconds) and y-axis represent the frequency (Hertz). We can see the repetitive nature of a police siren with time in the spectrum of figure *b*

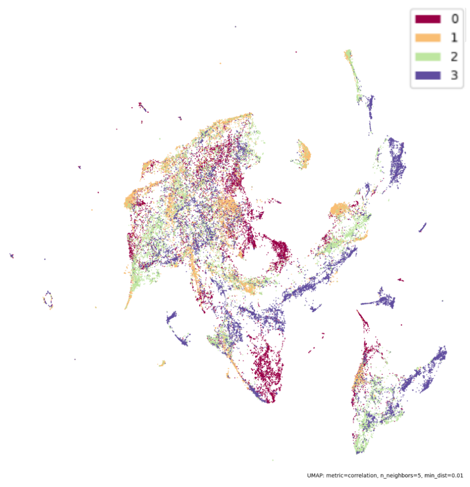
can be represented in a 2D image with frequency components on the y axis. Fbanks for two different sounds can be seen in figure 3.3.

3.2.2 Data insights by dimensionality reduction techniques

In this section we will visualize our dataset (in 2-D) to see the relationship between the different classes. There are many techniques that can transform higher dimensional data into lower dimensions. The representation and visualization help to observe the overlap between different classes. The fbank breaks



(a) TSNE: The red is Airplane, blue is car, green is Silence and yellow is Nebengeraeusche.



(b) UMAP: 0 (Red) is Airplane, , 1 (Yellow) is Silence, 2(Green) is Nebengeraeusche and 3(Blue) is car.

Figure 3.4: Dimensionality reduction techniques where different colors belongs to different classes. It visualizes the relationship between the different classes.

a sound into 128 distinct components, which means a window of sound can be represented as a single point in a higher-dimensional space. The dimension reduction technique helps to represent the same data in 2 dimensions and helps us understand the relationship between different classes. In the thesis, we used TSNE [44] and UMAP [45] techniques to visualize our data. The results can be seen in figure 3.4.

We discussed several challenges in the section 3.1.3 relating to the data, and we will discuss the visualization in light of those challenges. We can see that we do not have a completely separate cluster of any class. Some small clusters of airplanes and cars can be distinguished among the other events. However, it is tough to distinguish between cars and airplanes in some cases. For example, sometimes very fast-moving cars sound like an airplane if listened to from a distance. We can also observe that the Silence class and the Background Noise class are very tightly coupled in some areas and slightly separated in others. Many noises in recordings could sound similar to Background Noise sounds. For example, the sound of a person snoring is not explicitly annotated and therefore could be included while extracting the Silence class. Similarly, a rotating fan can have a sound signature similar to a moving car. Therefore we can say that sound events in the dataset are not mutually exclusive and have much noise.

The size of clusters relative to each other is essentially meaningless in both TSNE and UMAP [46]. They use local notions of distance to construct its high-dimensional graph representation. Similarly, the distances between clusters are meaningless also due to the usage of local distances when constructing the graph. The dimensionality reduction techniques usually require multiple iterations to find optimum hyperparameters that give good results. Therefore multiple iterations with different hyperparameters were performed to obtain the final results.

3.2.3 Baseline Model

In this section, we will discuss the baseline methods used in the experiments. The baseline models give a benchmark to compare our primary classification model. We choose three different baseline models: K-Nearest Neighbor, Random Forest, and Support Vector Machine. The three models use different approaches as discusse below to classify the data . The classifiers use Mel filterbank as the feature for classification. The filter bank is made using 128 filterbank values for a window of 30 seconds.

K-Nearest Neighbor (KNN)

It is a supervised Machine Learning algorithm used for regression and classification [47]. It uses the principle that similar things remain in close proximity. The prediction of the class is dependent on the nearest K neighbors present in the training examples and is the average of the values of K neighbors. The neighbors to a point are selected with the least distance from the point.

Support Vector Machine (SVM)

It is a supervised Machine Learning algorithm [48]. It finds a plane in n-dimension that maximizes the margin between the plane and the training examples of different classes.

Random Forest

It is a supervised Machine Learning algorithm that creates an ensemble of many decision trees [49].

SVM handles outliers better than KNN. KNN is better than SVM if the training data is larger than the features [50]. SVM outperforms KNN with a significant number of features and lesser training data. KNN needs to track all training data and find adjacent nodes, which consumes much memory and is costly to train [50]. Random forests do not expect linear functionality or functionality that interacts linearly and is inherently suitable for multi-class problems, while SVMs are suitable for two-class problems [51], [52].

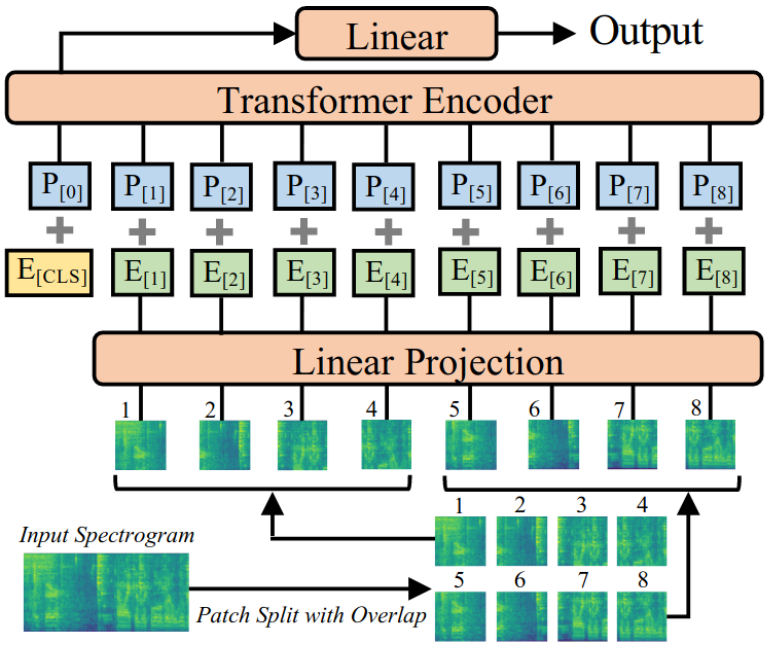


Figure 3.5: AST Architecture [3]: The 2D audio spectrogram is split into a sequence of 16*16 patches with overlap and then they are converted into 1-D patch embeddings. Positional embedding is added to each patch embedding. A special token [CLS] is added to the sequence. Then it is fed into the Transformer, and the [CLS] token's output is used to get prediction after passing through a linear layer

3.2.4 Audio Spectrogram Transformer (AST)

We use AST as our primary classifier as it is the current SOTA method in audio event classification. It is based on the transformer architecture [53] and the authors find that transformer-based models can have better results than CNN. It is inspired by Vision Transformer [54], where the transformer model is applied on an image dataset and proposed the approach for transfer learning from the image processing domain. The architecture of the AST can be seen in figure 3.5. The data pipeline in the AST generates 128 Mel filter banks over a fixed audio length. Then fbanks 2-D spectrogram is split into 16*16 patches with an overlap. In the thesis, we implemented AST with different window lengths. We hypothesized that different window lengths would favor different classes due to their average audio lengths. The transformer model's space complexity increases with the window length of the input audio sequence. Therefore after conducting many experiments, 30 seconds came out to be the optimal maximum length that could fit in the memory of the GPU. We then also trained models with 15, 10, and 5 seconds window sizes.

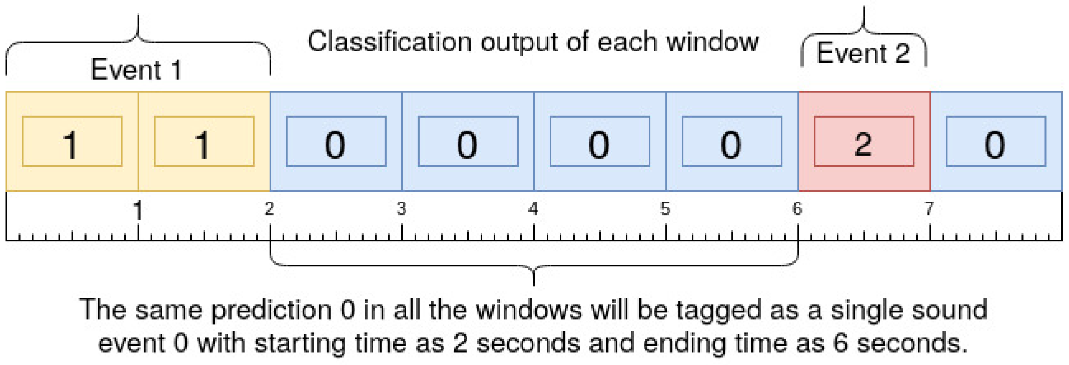

Event 1
Classification output of each window
Event 2
1
1
0
0
0
0
2
0
1
2
3
4
5
6
7
The same prediction 0 in all the windows will be tagged as a single sound event 0 with starting time as 2 seconds and ending time as 6 seconds.

Figure 3.6: Audio tagging: Different colors indicate different classes

3.2.5 Audio Tagging

According to the objective of this thesis, we have to identify the sound events present in night-long audio and find the starting and the ending time of the identified sound events. The audio is passed through a classification step where a window of n seconds is classified and labeled. After the classification is completed, we have to find the starting and ending times.

We have used a grouping approach to group the contiguous blocks of classified windows with the same label. Suppose if five contiguous windows have the label Silence and are preceded by Cars and succeeded by Airplane, then these five windows will make one sound event, i.e., Silence. The event's starting time will be the starting time of the first window, and the ending time will be the ending time of the last window as seen in figure 3.6. We found that a sound event can last only a single window or span tens of windows contiguously. We have implemented a tagging mechanism to tag every predicted sound event present in the audio.

3.2.6 Tagging audio with different techniques

In this section, we will discuss the implementation setup of one of the two main hypotheses. In the first hypothesis mentioned in section 3.2.4, we will observe if different window lengths favor sounds of different lengths. Now we create an implementation to test our second hypothesis. This hypothesis will observe if an overlapping window approach improves the starting and ending times of the predicted sound event.

Non-Overlapping Windows

It is the most straightforward approach to tag the recorded audio. We divide the audio into segments of n seconds corresponding to the window size of the AST. We then take each segment iteratively and stored the classification result in a raw prediction file containing every window's prediction. We then use this raw prediction file and tag the audio with the approach mentioned in section 3.2.5. This experiment is an example of simple audio tagging and has fewer computational requirements. The starting and ending points have higher errors when we take large window lengths. For example, if a sound event is classified using a 30-second window and a sound event is started after 20 seconds in the window. If the classification label is predicted correctly, then there will be an error of 20 seconds in the starting time of the sound event. We have analyzed this approach further in section 4.2.2

Overlapping Windows

Looking at the non-overlapping window above, we hypothesized that we could get better results about the classified class with the Overlapping Windows approach. There may be cases where a sound starts in one window for a few seconds and then ends in a few seconds in the next window. When the windows are classified in this state, the classifier may misclassify the sound event.

To remedy this, we can take an approach where after we classify the first window, we move forward with a step of only one-third of the window and create an overlap with the previous window. Then classify this overlapped window. In this way, we can classify a snippet of audio three times if the step forward is one-third. Now we have three classifications for a snippet of length $window/3$. We can take the result as the majority of the votes of the three classifications. In this way, we can iteratively take n seconds window of the audio and classify it. Then move forward with a step of $n/3$ seconds and classify n seconds again. The approach can be seen in the figure 3.7. This approach will take almost thrice the computational resources compared to the non-overlapping window as we have to classify more windows. We can also create an overlap of 1/5, but it will be more computation-intensive, though it could be explored in the future. The benefit of the approach is that we will have a fine-grained starting and the ending point since the effective classification window is one-third of the window used by the classifier.

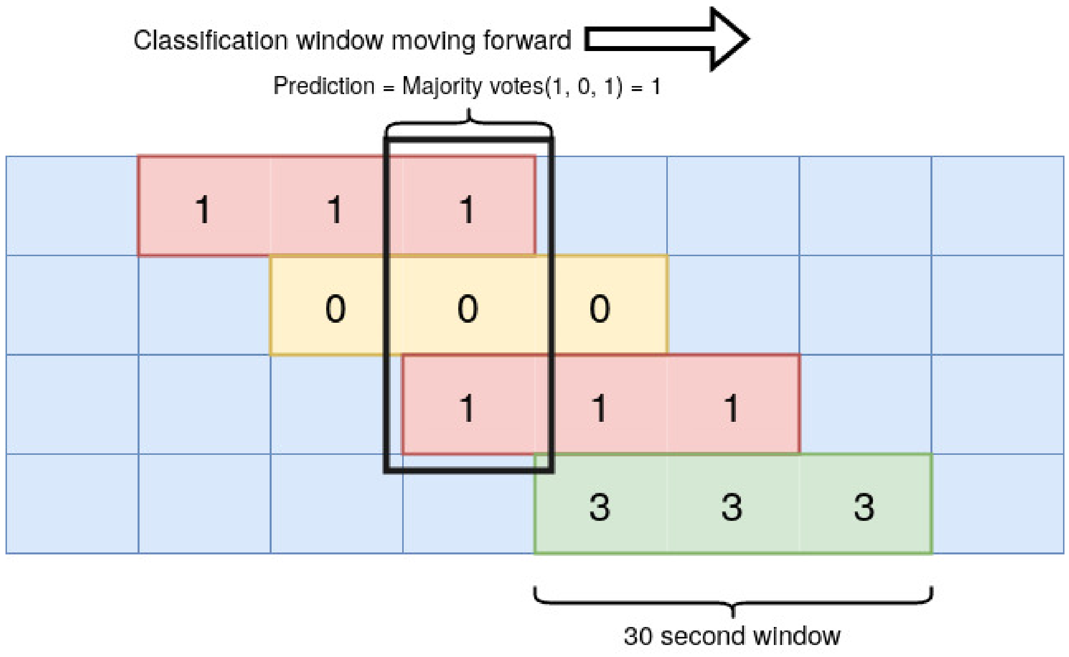

Classification window moving forward
Prediction = Majority votes(1, 0, 1) = 1
1
1
1
0
0
0
1
1
1
3
3
3
30 second window

Figure 3.7: A representation of classification with an overlapping window. We can see that the window is moving forward with a step of $n/3$, where n=30 seconds

3.3 Evaluation

We have used Polyphonic Sound Detection Score (PSDS) to evaluate the efficiency of classification models, and the different window approaches to tag the audio events in the long recording correctly [7]. The PSDS was developed by an organization called Audio Analytics, where they recognized some limitations in the existing metrics for sound event detection tasks. The PSDS predictions can be seen in figure 3.8. Current metrics limit the start and end timings of the prediction to a certain distance from the ground truth. The 'collar' is a constrain that allows for some tolerance in the assessment for either human or system, but it also raises the necessity for highly exact ground realities to match the detections closely [4]. For example, if an audio sample comprises two extremely near sound events (e.g., 200ms) (3rd item in figure 3.8), and if the annotator generates two ground truths. A system should not punish if it detects both annotations with a single detection. PSDS relaxes these constraints and makes the predictions more flexible. The background rectangles are ground truths, while the smaller rectangles illustrate the system detections. Vertical dashed defines the collar constraints. It can be inferred from figure 3.8 that PSDS is very flexible and relaxes the constraint to make the detection of a sound event more robust.

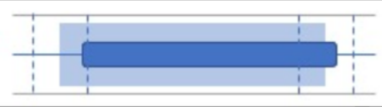
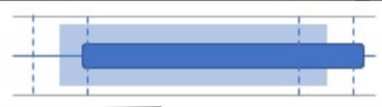
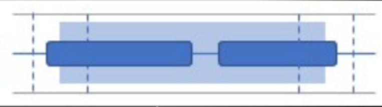
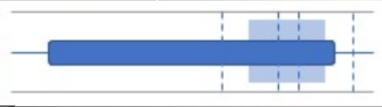
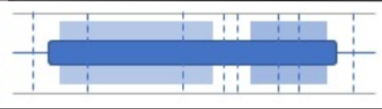
Collars			True Positive	
Left	Right		Collars	PSDS
✓	✓		✓	✓
✓	✗		✗	✓
✓ ✗	✗ ✓		✗	✓
✗	✓		✗	✗
✓ ✗	✗ ✓		✗	✓

Figure 3.8: PSDS Predictions [4]: The background rectangles are ground truths, while the smaller rectangles illustrate the system detections. Vertical dashed defines the collar constraints. The PSDS has relaxed the constraints and is flexible in predicting a sound event.

The PSDS spans between 0 and 1, where 1 represent that a system has predicted all the sound events perfectly. F1-score and True Positive rate(TPR) based on the PSDS are also generated. The metric is capable of finding efficacy in polyphonic sound events. They have argued that an intersection between the ground truth and the valid sound events should be evaluated instead of collars around start and end times. The metric has also been beneficial in the case of cross-triggers. In our case, we will use PSDS only for a single sound event in a single time range.

The following sections will discuss results acquired during experiments. The Baseline models will be compared with AST, and finally, the approach of Overlapping and Non-Overlapping windows will be evaluated.

Chapter 4

Results

In chapter 3 data, features and the models used in this thesis were discussed. We also discussed the non-overlapping and overlapping window approach. This chapter will present the results of our experiments, overall performance, and observations. In this chapter, we will discuss the results of baseline models and compare AST results with them. We will talk about the test train split of the data, the optimal hyperparameters of various algorithms, and finally, discuss the performance of each model.

We will evaluate our hypotheses mentioned in chapter 1. In the first hypothesis, we will compare the performance of models with different window lengths on different classes. Then we compare the results of the overlapping and non-overlapping window approaches and evaluate our second hypothesis. Finally, we will analyze our decision support system, which will help annotate the recordings.

4.1 Baseline models

We used three different supervised machine learning models to evaluate the data to create a baseline. We started with the Support vector machine (SVM) and used the data with a length of 30 seconds as discussed in section 3.1.6. The final dataset was split into train and test set with 75% of the data in the train set and the remaining in the test set. The data was converted into Mel filter banks (fbank) and used as the primary feature in the three baseline models. The SVM was initialized with different parameters and achieved the best results with c=0.5 and the 'RBF' kernel. We used K-Nearest neighbor (KNN) as a baseline and used 20 neighbors to achieve the best results. We created a Random Forest Classifier with 200 decision trees to create an ensemble in the third model. We then calculated Accuracy, Recall, Precision, and F1 score for all the classification models. All the hyperparameters in the baseline models are chosen after multiple runs, and the models with the best results are chosen. The Random forest classifier achieves the highest performance in baseline classification with an accuracy of 0.654. The KNN classifier followed it with an accuracy of 0.618, and finally, the SVM got an accuracy of 0.508. The results of the classification can be seen in figure 4.1. We have also compared the baseline model results with our primary classification model Audio Spectrogram Transformer (AST).

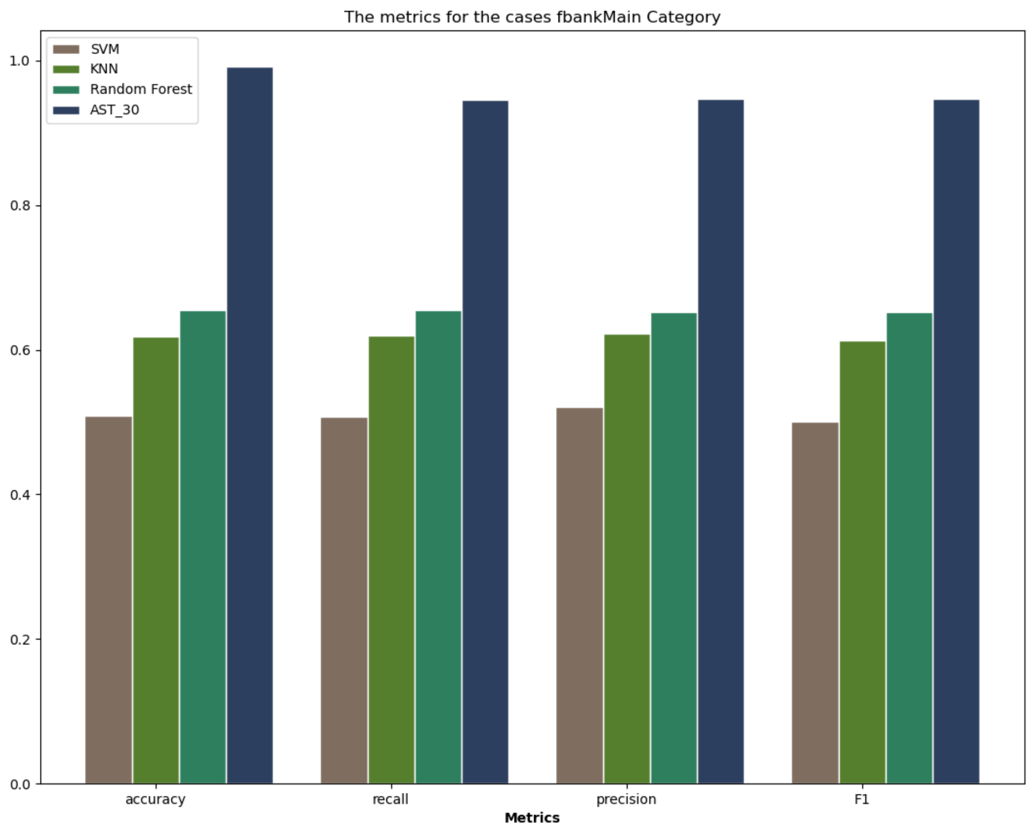


Figure 4.1: Accuracy, Recall, Precision, and F1 score of Baseline models (SVM, KNN, Random Forest) in comparison with the AST model with a window size of 30 seconds

	Accuracy	Recall	Precision	F1-score
SVM	0.508	0.507	0.521	0.514
KNN	0.618	0.619	0.621	0.620
Random Forest	0.654	0.654	0.651	0.653
AST_5	0.908	0.908	0.908	0.908
AST_{10}	0.918	0.918	0.918	0.918
AST_{15}	0.924	0.924	0.924	0.924
AST_{30}	**0.946**	**0.946**	**0.946**	**0.946**

Table 4.1: The table shows the performance metrics of the baseline models and the performance metrics of AST with different window sizes. AST_{30} has the highest overall performance.

4.2 AST results

Several experiments were performed with our primary classifier AST and our tagging system. The final dataset was split into training and test set with 75% of the data in the training set and the remaining in the test set. Then, we used different window sizes and observed the performance of each model. The models were named based on their window lengths, where if a model uses a window size of 30 seconds, then the model will be known as AST_{30}, where postscript represents the window size. We found out that the AST_{30} model receives the highest accuracy among all the models. It achieved an accuracy of 0.94, but as we decreased the window size of the model, the performance took a slight dip. The smallest window size used was 5 seconds, and it achieved an accuracy of 0.90. We observed from the results that a larger window size could classify smaller lengths sounds better, but the reverse is not valid. We will discuss more on this in the next section. The performance of all the models, along with the baseline models, can be seen in Table 4.1. All the AST models performed better than the baseline models. It was expected as the transformer-based architecture of the AST is better suited to learn sequences from the audio data. There is a significant performance jump of around 25% between the best performing baseline and least performing AST_5 models.

We can see the confusion matrix for the AST_{30} model in figure 4.2. The silence class achieves the maximum true positives of all the classes. Therefore we can say that our model can predict the presence of Silence with a very high probability. We can also see that the Silence class is sometimes misclassified

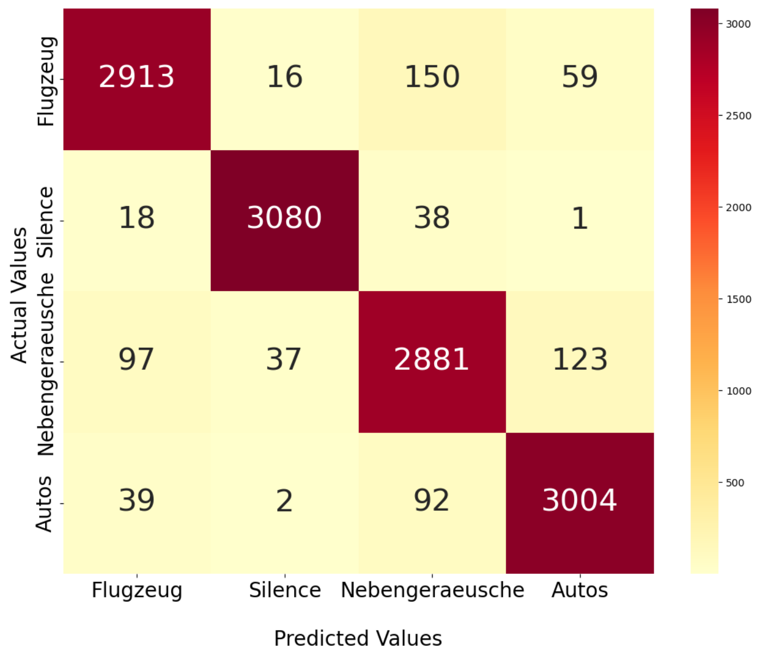


Figure 4.2: Confusion Matrix for AST_{30} model

as the Nebengeraeusche. We saw this pattern when we visualized the audio examples using the dimensionality reduction technique in section 3.2.2. The clusters of the two classes were tightly coupled and were not easily separable. The airplane sound is often misclassified with Nebengeraeusche and Auto classes. After inspecting some misclassified samples, we found some participants used fans in some of the recordings during the night. The fan has sound signatures similar to that aircraft, and similarly, the Auto sounds are also misclassified as Nebengeraeusche. A distant fast-moving truck or a heavy car can be perceived as an aircraft. The unwanted noise present in the recordings can lead to misclassifications of some sounds.

4.2.1 Performance comparison of each class with different AST models

This section will remark on different window sizes used in the AST. We hypothesized that different window lengths might favor the sound class with different average lengths. We applied four different models corresponding to four window sizes and obtained each class's precision, recall, and F1-score. The observations can be seen in table 4.2. We observe that the AST_{30} achieves the

	Model	Flugzeug	Silence	Nebengeraeusche	Autos
Recall	AST_5	0.921	0.944	0.869	0.898
	AST_{10}	0.908	0.979	0.876	0.909
	AST_{15}	0.925	0.976	0.857	0.936
	AST_{30}	0.928	0.981	0.918	0.957
Precision	AST_5	0.897	0.942	0.882	0.910
	AST_{10}	0.922	0.936	0.892	0.921
	AST_{15}	0.914	0.959	0.915	0.907
	AST_{30}	0.949	0.982	0.911	0.942
F1-Score	AST_5	0.909	0.943	0.875	0.904
	AST_{10}	0.915	0.957	0.884	0.915
	AST_{15}	0.919	0.968	0.885	0.921
	AST_{30}	0.938	0.982	0.914	0.950
Average audio length (seconds)		68.01	-	27.35	16.52

Table 4.2: Precision, Recall and F1-score of each class corresponding to the AST models of different window sizes. AST_{30} achieves the highest performance in each class.

best performance in every metric for all the classified classes. The Silence class archives the highest F1-score of 0.982 among all the classes, followed by Autos and Flugzeug. We can say that the classifier can at least distinguish between any sound and Silence appropriately.

We also observed that the F1-score for each model slightly improves as we increase the window size of the AST. The AST_5 is worst in all the cases. We see an increase of almost 0.01 when we move from AST_5 to AST_{10}. But the increase from AST_{10} to AST_{15} only sees a slight increase in the performance. Then as we move from AST_{15} to AST_{30}, we see a significant jump in the performance. Almost all the metrics of every class improved by at least 0.02. Therefore we can conclude that larger window sizes of the AST can classify each class better than AST with a smaller window. Our hypothesis does not hold in this case.

4.2.2 Audio tagging using non-overlapping windows

As discussed in the section 3.2.6, in the classification with a non-overlapping window, the model iteratively classifies a window of audio snippets at one time. We will use PSDS as a measure of performance to assess the correct

Recording	Window Size	PSDS	Airplane		Car		Nebengerausche		Silence	
			F1	TPR	F1	TPR	F1	TPR	F1	TPR
135-0061-210521	5	0.151	0.101	0.942	-	-	0.013	0.35	0.177	0.142
-225750-indoors	10	0.431	0.176	0.885	-	-	0.051	0.55	0.53	0.392
	15	0.565	0.312	0.942	-	-	0.054	0.3	0.767	0.678
	30	0.708	0.688	0.914	-	-	0.222	0.4	0.92	0.857

Table 4.3: Metric for Non-Overlapping window evaluated on a sample recording. The Autos class was not available in the manually tagged audio file therefore its metrics cannot be calculated.

Recording	Window Size (Overlap)	PSDS	Airplane		Car		Nebengerausche		Silence	
			F1	TPR	F1	TPR	F1	TPR	F1	TPR
135-0061-210521	15	0.516	0.251	0.885	-	-	0.041	0.25	0.755	0.66
-225750-indoors	30	0.682	0.744	0.914	-	-	0.2	0.35	0.902	0.821

Table 4.4: Metric for overlapping window valuated on a sample recording is calculated only for 15 and 30 seconds window size. The Autos class was not available in the manually tagged audio file therefore its metrics cannot be calculated.

identification of starting point and the ending point of the identified events. The highest PSDS achieved during the tagging of an example recording was 0.708, with a window size of 30 seconds. The classes Airplane and Silence individually achieve the F1-scores of 0.688 and 0.923. The class Nebengeraeusche got an F1-score of 0.222 and can be attributed to the fact that not all the background sounds were tagged in the raw annotation file. The window size with 15 seconds for the same recording achieved a PSDS score of 0.566. The F1-score for class airplanes dropped significantly to just 0.3128. The window sizes with 5 and 10 seconds achieved PSDS scores of 0.151 and 0.431. The F1-score for all the classes for these two window sizes is very low. The different metrics for a tagged recording can be seen in table 4.3

4.2.3 Audio tagging using overlapping windows

As discussed in the section 3.2.6, the classification with an overlapping window iteratively classifies a window (size n) of an audio snippet and moves forward with a step of n/3. The classification of every n/3 sized window is done three times. With a window size of 30 seconds and a 10-second step, we achieved a PSDS of 0.682 for the example test recording. The PSDS based F1-score is calculated for different classes. The classes Airplane and Silence individually achieve the F1-scores of 0.7442 and 0.902. We observed that the overlapping window approach improved the starting and ending point of the sound events, which are longer in length. Table 3.2.6 shows that the F1 score for the Airplane increases significantly from the non-overlapping window approach with a 30-second window.

But if the sound events predicted are smaller in length, then the overlapping approach applies an operation similar to smoothing. Therefore any shorter sound event gets smoothed out. The class Nebengeraeusche achieved an F1-score of 0.20, less than the non-overlapping window approach. The window size with 15 seconds and the step of 5 seconds for the same recording achieved a PSDS score of 0.516. The F1-score for class airplanes dropped significantly to just 0.251. The overlapping window approach was not performed with 5 and 10 seconds smaller window sizes as the step size would be too small. The different metrics for a tagged recording can be seen in table 4.4. Therefore our hypothesis that the overlapping window approach will improve the starting and the ending time of the predicted sound event is partially valid, only for longer duration events.

4.2.4 Decision support system

Finally, a decision support system is created that will make the audio tagging easy. The system will perform the following steps to tag an audio recording.

1. Take an audio recording.
2. Identify the various sound events with iterative classification.
3. Identify the sound event's starting and ending points with overlapping and non-overlapping windows separately.
4. Generate a CSV for each window size separately. (6 total: 4 for non-overlapping windows, and 2 for overlapping windows)
5. Create a visualization based on a CSV file generated in the step above. figure 4.3 shows an example visualization where the tagged audio events are shown corresponding to a CSV file.

The visualization shows a graph of pressure values(y-axis) measured w.r.t the time (x-axis) during the night. The horizontal spans of different colors show various sound events identified, and their starting and ending points can be identified on the x-axis. The pressure values are the sound intensity of the the recorded sounds and therefore represented in decibels (dB). The time axis on the x-axis can be zoomed in and therefore can have values in seconds, minutes or even hours. The visualization system has many additional features like Horizontal zooming, where we can zoom only in the time axis, and the values in the y axis remain the same. The mouse pointer shows the pressure and time values in a small bubble to identify exact values more easily. We will see the advantage of using overlapping window using decision support system in the next section.

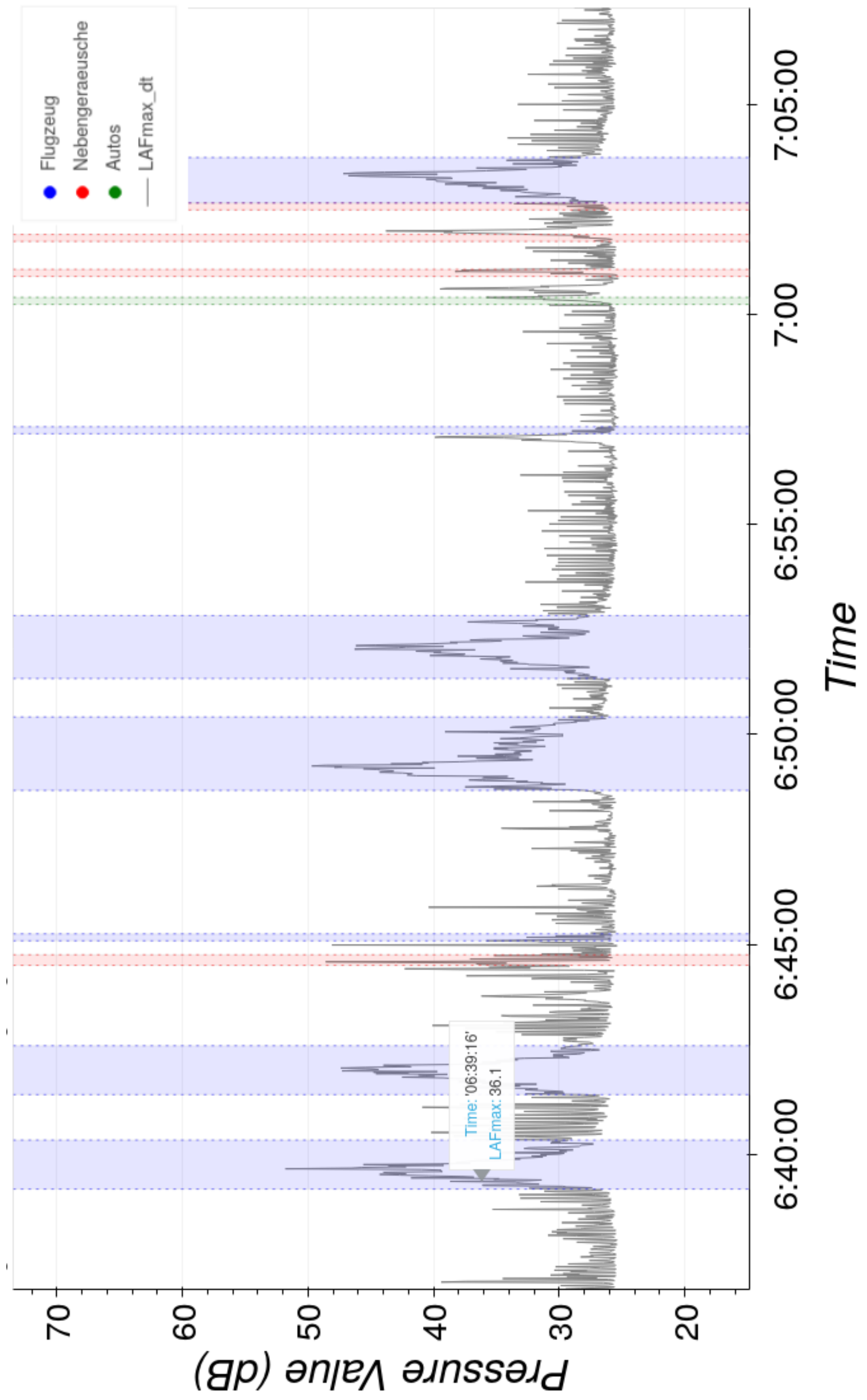
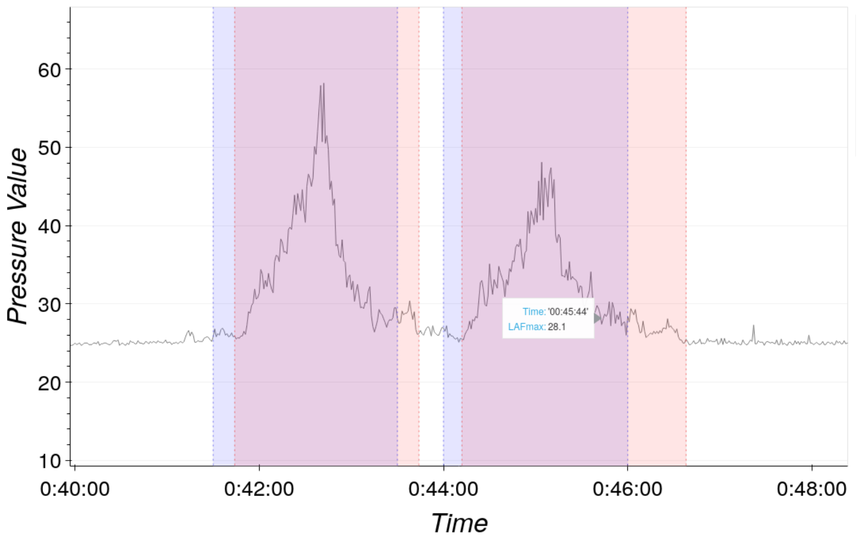


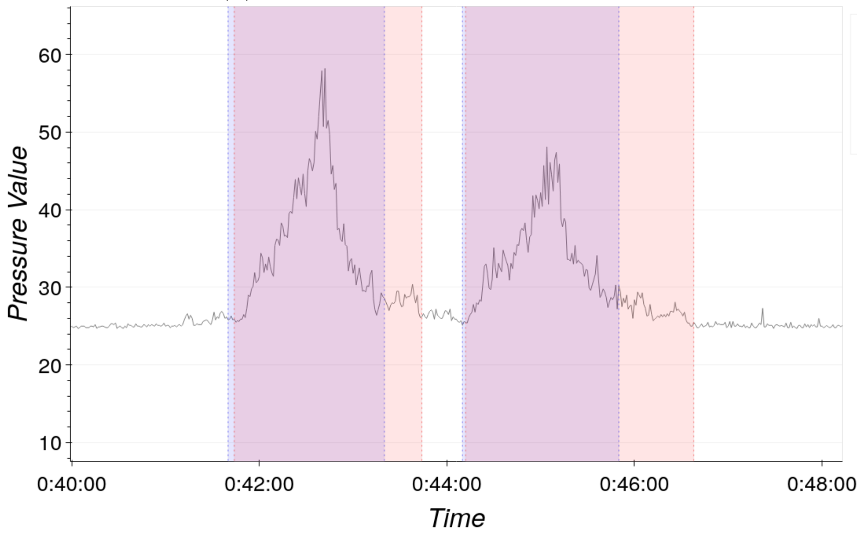
Figure 4.3: Decision Support System: The horizontal spans of different colors show various sound events identified, and their starting and ending points can be identified on the x-axis.

4.2.5 Advantage of using Overlapping Window

We can see in the figure 4.4 that the overlapping window approach improves the starting time accuracy of the sound event identified in the recording. The blue span is the predicted event, and the red is the actual event. We can see in the figure that the starting point of the blue and red span are closer in the overlapping window approach than in the non-overlapping window.



(a) Non-overlapping window tagging



(b) Overlapping window tagging

Figure 4.4: The blue span is the predicted event, and the red is the actual event. We can see that the starting point of the blue and red span are closer in the overlapping window approach than in the non-overlapping window.

We have achieved quite good results using the AST model. The decision support system can tag different sounds and then visualize them in usable visualization. In the next section, we have discussed the results achieved in this section in detail.

Chapter 5

Discussion

This chapter will discuss the various aspects of the thesis. We will start with the data and its effect on the final system. We will comment on the features and the models used during the classification. The efficacy of using the overlapping window approach will also be evaluated. Finally, we will discuss our approach for audio tagging and the final decision support system developed as part of this thesis.

The audio recordings in the first two studies (STRAIN and MIDAS) are noisy, as discussed in the section 3.1.3. The multiple sound events in the data make it very challenging for the classifier to predict the correct class and are one of the reasons that the performance on the final test recordings is not good. However, in real-world scenarios, we cannot avoid noise. It is a part of the environment, and we should be able to better deal with noise and design classifiers that are resistant to noise.

The night-long recording were annotated with many different classes. We tried to group them into four major groups to identify the aircraft noises in the audio recordings. The earlier studies at DLR discussed in section 3.1 used a vast number of categories. In the most recent study, FluiD-21, the number of classes has been reduced and focused mainly on aircraft noise. The categories have been condensed, and it negatively affects data quality. The Subject noise label is composed of a lot of sub-labels that do not have a similar audio fingerprint eg. (Room cracking and subject's speech). This leads to highly diverse sounds under a single label that is challenging for the model. The category reduction has also led to less tagging for sound events that are not deemed necessary.

The feature we used in the classification model was inspired by the domain of Computer Vision. The Mel filter bank creates a spectrogram, and the AST uses these filter banks and classifies the sound events. The AST is used with pre-trained weights from the AudioSet dataset. Therefore the AST has already been trained on an audio dataset with more than 500 different classes. This pretraining might affect our classification scheme as we have decreased the possible output labels to only four. We could have retrained some of the layers of the AST with our dataset to suit our classification scheme. It might have further increased the performance of the final model. Also, the current classifier classifies and predicts only a single class for each classification. Therefore a multi-label output from the AST can be considered for evaluation.

The difference in the average length of the sound events led us to use different window sizes of 5, 10, 15, and 30 seconds. For example, the average length of the class Autos is 16 seconds, and the expectation is that classifiers with smaller window sizes performed better in this case. But as we observed the results in table 4.2, the expectation does not hold. The main reason to include a 5-second window is to identify instantaneous noises like cough or a sneeze. However, we can see from the observations in table 4.2 that the classifier with a 5-second window has the worst performance among all the other classifiers. Although the overall metrics are high, we noticed that all the AST models have a low performance for Nebengeraeusche. It could be due to many unrelated classes(Subject noise, Cracking noises of the floor) that are combined into this class. We can conclude that models with larger window sizes may be suitable for classifying different sound events.

The hypothesis with the overlapping window approach was partially successful as we achieved higher accuracy in terms of starting and the ending timings of the sound events that are longer in duration. The 30-second window approach best identifies all the aircraft noises present in the audio recordings. Another effect observed with the 30-seconds window is that the short-duration sound events are removed from the final tagging. The overlapping approach applies an operation similar to smoothing. Therefore any shorter sound event gets smoothed out. This approach helps in noisy audio recordings where these short sounds are unwanted. However, it has a negative effect on sound events like Autos that are inherently very short in duration, as seen in the table 3.2. We can also develop an approach to penalize some sound events while taking the majority votes in the process.

As discussed in section 3.1, the annotators have not tagged every sound event in the audio recording. Only the sound event that wakes up the participant is included in the annotation file, or else it is discarded. So even if the sound event is present but did not disturb the participant will not be included in the final event file. However, the classifier can identify most of the sound events in the recording, which leads to a high number of misclassifications and is one of the reasons for less PSDS scores on the test recordings. However, it is better for the final decision support system because all the sound events will be tagged during the night, and the annotator can decide which sound events to keep and discard the others.

The decision support system introduced in section 4.2.4 can be beneficial to the annotators as they now have to spend less time manually annotating the recording. The support system can identify the sound events in the audio recording and show them in a pretty and descriptive visualization. They can use the output from the decision support system, listen to the sound event mentioned in the output, and decide if they want to keep the sound event. If we also introduce the multi-label classification in the system, we can directly identify whether a sound event should be accepted without listening to the audio.

This concludes our discussion on various investigations that this thesis has conducted. Our main hypotheses have been evaluated, and further improvements to the system have also been identified. The following section will conclude our work and discuss the possible work that could follow this thesis.

Chapter 6

Conclusion and Future Works

The objective of this thesis was to identify a sound event and estimate its starting and ending time in a noisy night-long audio clip. We also investigated the classification performance with different window lengths and if overlapping and non-overlapping window steps might favor some classes. The thesis started with the investigation of the data and the categories of sound events defined in various DLR studies. The sound events were extracted and stored to create a single dataset of different sound events. The number of categories in the data was reduced to four to simplify the classification model. The thesis also pondered upon the effect of noise present in the data and the challenges it poses for classification models.

The thesis also reviewed various audio features that could be used for classification and found out that Mel filter banks were more suitable for the classification model. We studied various models used for classification and found a transformer-based model suitable for our data. The final results of the classification are encouraging. The AST model achieved an accuracy of 0.94 on the extracted dataset with a window size of 30 seconds. The highest PSDS score of 0.70 was achieved for tagging an 8-hour long test recording. The high PSDS shows superior sound event identification and accurate starting and ending points.

The thesis started with two main hypotheses of different window sizes and overlapping window approaches to increase the efficiency of final audio tagging. By the end of the experiments, the overlapping window approach to improve starting and end timings of the predicted sound event stands partially true. The approach favors longer sound events and smoothed out the events with smaller lengths. Another hypothesis that different window sizes may favor different classes does not hold. We observed that a larger window size could classify smaller lengths sounds better.

In the thesis, we developed a decision support system that will help the annotators tag different audio events with ease. The system gives an interface to check the various sound events predicted in the audio recordings and obtain pressure values and timings of the events. The annotators can choose the appropriate resources for their study, resulting in a decrease in the effort required to conduct quality research at DLR. The system can be modified and used in similar application where the final goal is to tag different sound events.

6.1 Future Works

The thesis only used four classes to create the final system. More classes can be included in the future classification scheme. The subclasses of the broad classes can also be examined. Separate classifiers for the broad classes and the subclasses can be created. There can also be a different hierarchy where we group the sounds differently. Different approaches can be explored to decrease the amount of noise in data or find models resistant to noise. Newer classification models that are better than the previous models should be explored. Other audio features like constant Q-transform and many more can be employed in the classification models.

The classifier in the system can only identify a single sound event at a time. The classifier can be organized to predict two different classes instead of one in the future. This scheme can be beneficial to detect if the mixture of the sounds affected human sleep. A threshold can be fixed, and the second output should only be picked if the probability of that outcome is greater than the threshold. As discussed in 3.1, the sound event is accepted only if it wakes the participant. For example, if the classifier outputs two labels: an event and the subject noise, this will indicate that the sound event wakes up the participant. In this way, this approach will even remove the need for manual examination.

We can create an ensemble of classifiers that will classify the same audio recording in parallel. The results can therefore be merged based on some conditions. Another approach could be targetted multi-classification where the ensemble only gives output if a particular class is the output of the initial classification. After the initial classification on an audio snippet and obtaining a particular class, multiple smaller windows can be classified using different classifiers.

In the thesis, we explored the overlapping window approach where we use the majority of votes to decide a label. We can also use a weighted approach where some classes are rewarded, and others are penalized based on the type of audio data. For example, if the data is noisy, then the noisy classes should be penalized. The number of overlaps can also be increased from 3 to more overlaps. In this thesis, due to the limited resources, the maximum length of

30 seconds can be identified. It can be investigated if this window size can further increase to classify even longer audio lengths.

The decision support system designed in the thesis has very few features. The system can be further improved by adding some features. Right now, the class events can only be visualized in the system on a graph. Adding the audio support in the visualization will remove the need to listen to the audio recording separately. The user can click on the graph, and the audio can start playing from that point. The system generates a CSV file with the final sound events. Therefore, functionality could be included to show a table with the sound events, their starting and ending times, and the visualization. The user should change the data in the table and tag the audio after a manual check if necessary. Another feature could be added to adjust the starting and the ending timing by dragging the spans along the time axis. The changes should also reflect in the table displayed. Then finally, the event data should be exported in different file formats.

Appendix A

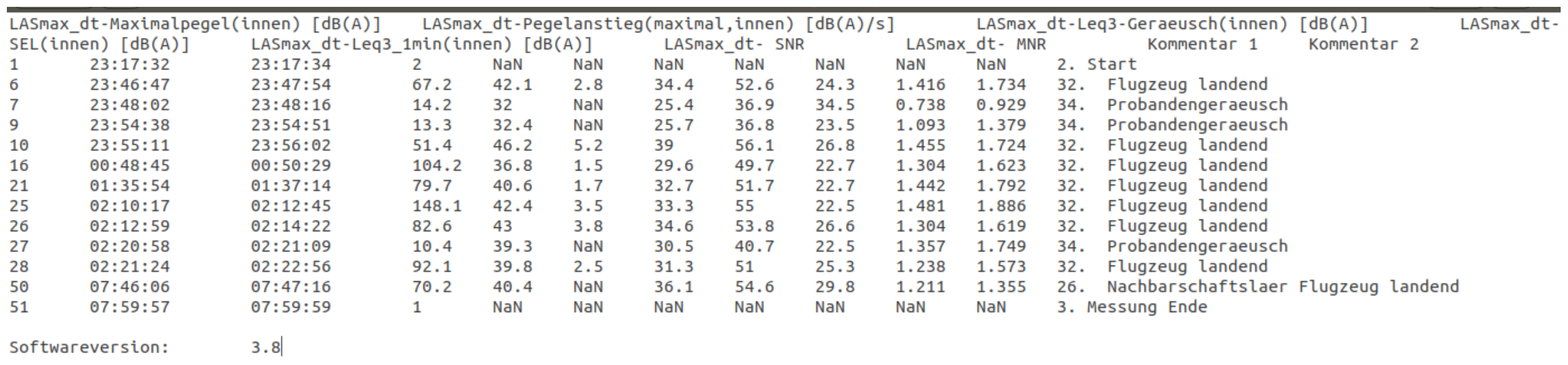
Apendix

A.1 Sample Event file

The event file contains the starting time, Ending time, event type, and the pressure values corresponding to that event.

A.2 Confusion Matrices

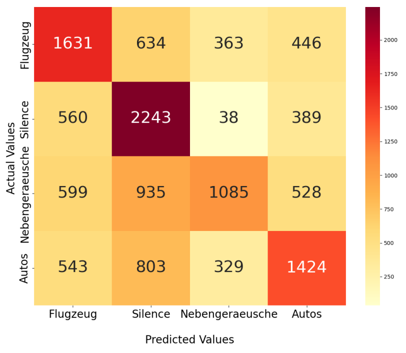
The confusion matrices for various models used in the thesis are shown in Figure A.2



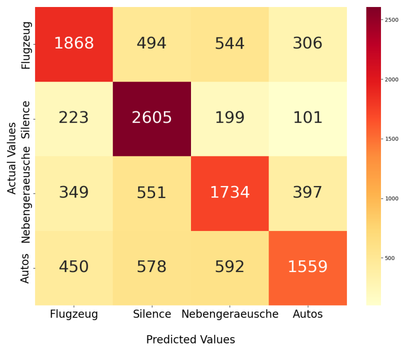
```
LASmax_dt-Maximalpegel(innen) [dB(A)]    LASmax_dt-Pegelanstieg(maximal,innen) [dB(A)/s]       LASmax_dt-Leq3-Geraeusch(innen) [dB(A)]       LASmax_dt-
SEL(innen) [dB(A)]          LASmax_dt-Leq3_1min(innen) [dB(A)]          LASmax_dt- SNR          LASmax_dt- MNR          Kommentar 1     Kommentar 2
1       23:17:32        23:17:34        2       NaN     NaN     NaN     NaN     NaN     NaN     NaN     2. Start
6       23:46:47        23:47:54        67.2    42.1    2.8     34.4    52.6    24.3    1.416   1.734   32.  Flugzeug landend
7       23:48:02        23:48:16        14.2    32      NaN     25.4    36.9    34.5    0.738   0.929   34.  Probandengeraeusch
9       23:54:38        23:54:51        13.3    32.4    NaN     25.7    36.8    23.5    1.093   1.379   34.  Probandengeraeusch
10      23:55:11        23:56:02        51.4    46.2    5.2     39      56.1    26.8    1.455   1.724   32.  Flugzeug landend
16      00:48:45        00:50:29        104.2   36.8    1.5     29.6    49.7    22.7    1.304   1.623   32.  Flugzeug landend
21      01:35:54        01:37:14        79.7    40.6    1.7     32.7    51.7    22.7    1.442   1.792   32.  Flugzeug landend
25      02:10:17        02:12:45        148.1   42.4    3.5     33.3    55      22.5    1.481   1.886   32.  Flugzeug landend
26      02:12:59        02:14:22        82.6    43      3.8     34.6    53.8    26.6    1.304   1.619   32.  Flugzeug landend
27      02:20:58        02:21:09        10.4    39.3    NaN     30.5    40.7    22.5    1.357   1.749   34.  Probandengeraeusch
28      02:21:24        02:22:56        92.1    39.8    2.5     31.3    51      25.3    1.238   1.573   32.  Flugzeug landend
50      07:46:06        07:47:16        70.2    40.4    NaN     36.1    54.6    29.8    1.211   1.355   26.  Nachbarschaftslaer Flugzeug landend
51      07:59:57        07:59:59        1       NaN     NaN     NaN     NaN     NaN     NaN     NaN     3. Messung Ende

Softwareversion:        3.8
```

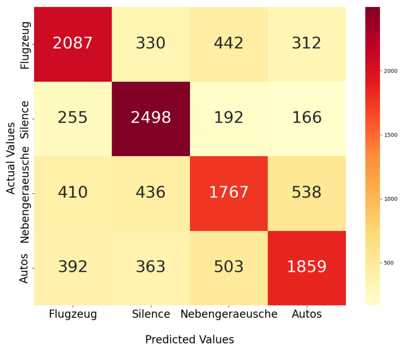
Figure A.1: Sample event file. The second last and the last columns represent the sound event. Starting and ending times are given by second and third column.



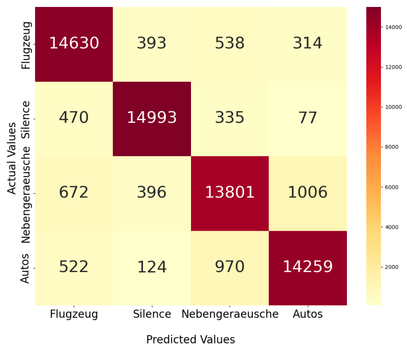
(a) SVM



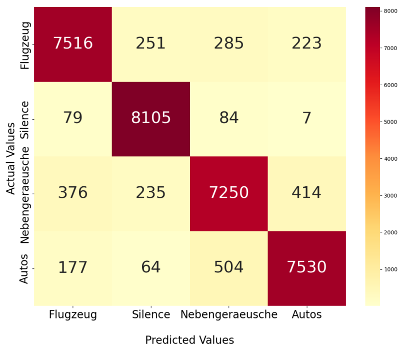
(b) KNN



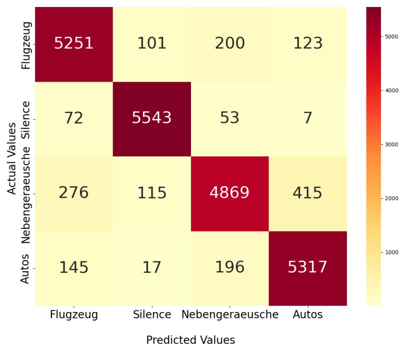
(c) Random Forest



(d) AST_5



(e) AST_{10}



(f) AST_{15}

Figure A.2: The confusion matrices for different classification models

Bibliography

[1] presonus presonus. Digital audio basics: Sample rate and bit depth. (document), 2.1

[2] Martin Immel. Potentiale des maschinellen lernens zur klassifikation von verkehrs und umgebungsgeraeuschen. Master's thesis, FRIEDRICH SCHILLER UNIVERSITAET JENA, 02 2021. (document), 3.2, 3.1.5

[3] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. *arXiv:2104.01778 [cs]*, April 2021. arXiv: 2104.01778. (document), 1, 2.2, 3.2.1, 3.5

[4] Audioanalytic Audioanalytic. Audioanalytic/psds$_{e}val : Polyphonicsounddetectionscore(psds).(document), 3.3, 3.8$

[5] L. Ma, Dan Smith, and Ben Milner. Context awareness using environmental noise classification. 01 2003. 1

[6] S. S. Stevens, John E. Volkmann, and Edwin B. Newman. A scale for the measurement of the psychological magnitude pitch. *Journal of the Acoustical Society of America*, 8:185–190, 1937. 1, 3.2.1

[7] Cagdas Bilen, Giacomo Ferroni, Francesco Tuveri, Juan Azcarreta, and Sacha Krstulovic. A framework for the robust evaluation of sound event detection, 2020. 2, 3.3

[8] History of the Cylinder Phonograph | History of Edison Sound Recordings | Articles and Essays | Inventing Entertainment: The Early Motion Pictures and Sound Recordings of the Edison Companies | Digital Collections | Library of Congress. 2

[9] wikipedia. Pulse-code modulation, Dec 2021. 2

[10] Nyquist frequency, Apr 2021. 2

[11] Peter Knees and Markus Schedl. *Basic Methods of Audio Signal Processing*, pages 33–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016. 2.1, 1

[12] Valerio Velardo The Sound of AI. Types of audio features for machine learning, Jul 2020. 2.1

[13] Theodoros Giannakopoulos and Aggelos Pikrakis. *Introduction to Audio Analysis: A MATLAB Approach*. Academic Press, Inc., USA, 1st edition, 2014. 1, 2, 2, 3

[14] Simple experiments with speech detection. 3

[15] yasseryasser 3122 silver badges88 bronze badges and Thilina DissanayakeThilina Dissanayake 18188 bronze badges. How to obtain mel filterbank?, Dec 1966. 1

[16] Mel-frequency cepstrum, May 2021. 2

[17] Judith Brown. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, 89:425–, 01 1991. 3

[18] Jean-Julien Aucouturier, Boris Defreville, and FranÃ§ois Pachet. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America*, 122(2):881–891, 2007. 2.2

[19] Forrest Briggs, Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Fern, Raviv Raich, Sarah Frey, Adam Hadley, and Matthew Betts. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *The Journal of the Acoustical Society of America*, 131:4640–50, 06 2012. 2.2

[20] Mark Gales and Steve Young. 2008. 2.2

[21] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. Context-dependent sound event detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013:1, 01 2013. 2.2

[22] Michel Vacher, Jean-FranÃ§ois Serignat, and StÃ©phane Chaillol. Sound classification in a smart room environment: an approach using gmm and hmm methods. 05 2007. 2.2

[23] M. Dorfer, B. Lehner, Hamid Eghbal zadeh, Christoph Heindl, Fabian Paischer, and G. Widmer. ACOUSTIC SCENE CLASSIFICATION WITH FULLY CONVOLUTIONAL NEURAL NETWORKS AND I VECTORS Technical Report, 2018. 2.2

[24] Hamid Eghbal-zadeh, Bernhard Lehner, Matthias Dorfer, and Gerhard Widmer. A hybrid approach with multi-channel i-vectors and convolutional neural networks for acoustic scene classification. 08 2017. 2.2

[25] Hangting Chen, Zuozhen Liu, Zongming Liu, Pengyuan Zhang, and Yonghong Yan. Integrating the Data Augmentation Scheme with Various Classifiers for Acoustic Scene Modeling. *arXiv:1907.06639 [cs, eess]*, July 2019. arXiv: 1907.06639. 2.2

[26] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking cnn models for audio classification, 2020. 2.2

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2.2

[28] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. 2.2

[29] Haomin Zhang, Ian McLoughlin, and Yan Song. Robust sound event recognition using convolutional neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 559–563, 2015. 2.2

[30] Antonio Sevilla and Hervé Glotin. Audio bird classification with inception-v4 extended with time and time-frequency attention mechanisms. In *CLEF*, 2017. 2.2

[31] Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maaß, Radoslaw Mazur, and Alfred Mertins. Audio scene classification with deep recurrent neural networks. In *INTERSPEECH*, 2017. 2.2, 3.2.1

[32] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Tut database for acoustic scene classification and sound event detection. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1128–1132, 2016. 2.2

[33] Jia Deng, Sanjeev Satheesh, Alexander Berg, and Li Fei-Fei. Fast and balanced: Efficient label tree learning for large scale object recognition. *NIPS*, 01 2011. 2.2

[34] Soo-Don Hyun, Inkyu Choi, and Nam Soo. Acoustic scene classification using parallel combination of lstm and cnn. 2016. 2.2

[35] Lars Hertel, Huy Phan, and Alfred Mertins. Classifying Variable-Length Audio Files with All-Convolutional Networks and Masked Global Pooling. *arXiv:1607.02857 [cs]*, July 2016. arXiv: 1607.02857. 2.2

[36] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *Interspeech 2019*, pages 2613–2617, September 2019. arXiv: 1904.08779. 2.2

[37] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *INTERSPEECH*, 2015. 2.2

[38] Anurag Kumar and Vamsi Krishna Ithapu. A Sequential Self Teaching Approach for Improving Generalization in Sound Event Recognition. *arXiv:2007.00144 [cs, eess]*, June 2020. arXiv: 2007.00144. 2.2

[39] Dimitrios Giannoulis, Anssi Klapuri, and Mark D. Plumbley. Recognition of harmonic sounds in polyphonic audio using a missing feature approach. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8658–8662, 2013. 2.2

[40] Myung Jong Kim and Hoirin Kim. Automatic extraction of pornographic contents using radon transform based audio features. In *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 205–210, 2011. 2.2

[41] Koichi Miyazaki, Tatsuya Komatsu, Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, and K. Takeda. Convolution-augmented transformer for semi-supervised sound event detection technical report. 2020. 2.2, 3.2.1

[42] Qiuqiang Kong, Yong Xu, and Mark Plumbley. Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, PP:1–1, 08 2020. 2.2, 3.2.1

[43] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition, 2020. 2.2

[44] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003. 3.2.2

[45] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. 3.2.2

[46] Coenen Andy and Pearce Adam. Understanding UMAP — pair-code.github.io. `https://pair-code.github.io/understanding-umap/`. [Accessed 27-Oct-2021]. 3.2.2

[47] Evelyn Fix and Joseph L. Hodges. Discriminatory analysis - nonparametric discrimination: Consistency properties. *International Statistical Review*, 57:238, 1951. 3.2.3

[48] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. 3.2.3

[49] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. 3.2.3

[50] Danny Varghese. Comparative study on classic machine learning algorithms, May 2019. 3.2.3

[51] Xamat. What are the advantages of different classification algorithms? 3.2.3

[52] lanenok lanenok. When to use random forest over svm and vice versa?, Aug 2015. 3.2.3

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 3.2.4

[54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 3.2.4