University of Thessaly
Greece, Spring 2019

Sound Event Detection and Localization in domestic environments using Deep Learning

Εντοπισμός και αναγνώριση τύπου και προέλευσης ήχων σε εσωτερικούς χώρους με την χρήση βαθιάς μάθησης

Sotirios Panagiotis Chytas

Supervisor: Gerasimos Potamianos

Committee Members: Nikolaos Bellas, Michael Vassilakopoulos

Diploma Thesis

Department of Electrical and Computer Engineering

University of Thesssaly

Volos, Greece

This Thesis was written as part of the requirements for the Diploma of Electrical and Computer Engineering at University of Thessaly.

$$1.01^{365} = 37.8$$

$$0.99^{365} = 0.03$$

Declaration of Authorship

I, Sotirios Panagiotis CHYTAS, declare that this thesis titled, "Sound Event Detection and Localization in domestic environments using Deep Learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all the main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

University of Thessaly

Greece, June 2019

Sotirios Panagiotis Chytas

Περίληψη

Τα τελευταία χρόνια, ο τομέας της Τεχνητής Νοημοσύνης γνωρίζει μια τεράστια ανάπτυξη. Ευρενητές και εταιρίες παγκοσμίως προσπαθούν να επιλύσουν όλο και περισσότερα προβλήματα με ¨έξυπνους αλγορίθμους¨. Οι αλγόριθμοι αυτοί ανήκουν στον τομέα της **Μηχανικής Μάθησης** και οι πιο εκλεπτυσμένοι εξ΄ αυτών ονομάζονται νευρωνικά δίκτυα και ανήκουν στον υποτομέα της **Βαθειάς Μάθησης**.

Μια μεγάλη περιοχή έρευνας και ανάπτυξης είναι η περιοχή του ήχου με ανοιχτά ζητήματα όπως: πως μπορούμε να αναγνωρίσουμε την ταυτότητα του κάθε ήχου, πως μπορούμε να αναγνωρίσουμε την προέλευση του στον τρισδιάστατο χώρο, πως μπορούμε να καταλάβουμε πότε ακούγεται κάποιος ήχος κτλ. Συνεχώς προκύπτουν νέες βελτιωμένες λύσεις και τα τελευταία χρόνια η συντριπτική πλειοψηφία αυτών χρησιμοποιούν τεχνικές Βαθειάς Μάθησης.

Στην συγκεκριμένη εργασία προτείνουμε ένα διασυνδεδεμένο σύστημα νευρωνικών δικτύων, και πιο συγκεκριμένα *Συνελικτικών Νευρωνικών Δικτύων*, με σκοπό την επίλυση του δισυπόστατου προβλήματος της αναγνώρισης της ταυτότητας κάθε ήχου και της εύρεσης την προέλευσης του στον τρισδιάστατο χώρο. Η εργασία έγινε στα πλαίσια του παγκόσμιου διαγωνισμού DCASE19-task 3 [1] στον οποίον και λάβαμε μέρος με την μεθοδολογία που θα αναλύσουμε στα επόμενα κεφάλαια, όπως επίσης περιγράφεται στην τεχνική μας αναφορά [2] ως μέρος της υποβολής μας στον διαγωνισμό αυτόν.

Abstract

In recent years, the AI field has been growing rapidly. Researchers and industries have been trying to solve as many problems as they can with "smart" algorithms. Such algorithms are called **Machine Learning** algorithms and the most sophisticated of them are the Neural Networks that belong to the **Deep Learning** sub-field.

A wide area of research and development concerns audio related problems. Some of the most interesting and popular problems are sound classification, sound localization, sound detection, etc. New, improved solutions emerge all the time. In recent years, the vast majority of these solutions use Deep Learning techniques.

In this thesis, we propose an interconnected system of Neural Networks and, specifically, an interconnected system of Convolutional Neural Networks in order to solve two problems simultaneously: Sound Event Detection and Sound Event Localization. Our methodology was submitted to the DCASE19 Challenge - task3 [1], as also described in our technical report [2] that was included as part of our system submission to the Challenge.

Acknowledgments

Firstly, I would like to thank my thesis supervisor associate professor Gerasimos Potamianos for his guidance throughout my journey of thesis writing. With his help I managed to present a method with many unique ideas, submit it to a worldwide contest and write a paper, on top of my thesis' writing.

Secondly, I am thankful to associate professors Nikolaos Bellas and Michael Vassilakopoulos for being also my thesis supervisors as well as to all of the department's professors and staff who helped me the last 5 years.

Moreover, I should thank all these people who believed in me and helped me (even without realizing) and became a part of me. I am particularly thankful to my parents for their unconditional love and support. Thank you for bearing with me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Sound Event Detection and Localization

Sound Event Detection (SED) is an active area of research with many applications, such as medical telemonitoring [3] and surveillance [4]. Many solutions have been proposed using machine learning methods such as Gaussian mixture models and Hidden Markov Models [5], SVM [6], Random Forests [7], as well as Matrix Factorization techniques [8; 9]. In recent years, deep learning solutions, using mainly Convolutional Neural Networks (CNN) [10; 11] and Recurrent Neural Networks (RNN) [12; 13], outperform classic machine learning solutions. Not surprisingly, SED has been the subject of multiple evaluation campaigns in the literature, including the recent and well-established DCASE (Detection and Classification of Acoustic Sound Events) Challenges [14; 15; 16]

Moreover, alongside with the SED task, in many applications [17; 18] is also crucial to find the Direction of Arrival (DOA) of each source. Traditional methods are parametric and use techniques such as Enhanced Sound Localization [19] and Multiple Signal Classification [20]. In recent years, Deep Learning solutions have been proposed [21; 22], which seem to perform even better.

In this thesis, we present our developed SELD (Sound Event Detection and Localization) system for Task 3 of the 2019 DCASE Challenge [1]. As deep-learning based methods are well-established, outperforming traditional machine learning ones in both SED and DOA estimation, we adopt a deep-learning approach. Our proposed method is based on

CNNs and Ensembling. In particular, we employ convolutional neural networks (CNNs) to first address SED, i.e., determine the existence of each class at each time-frame, and to subsequently estimate the DOA for each of the audio segments predicted to exist. Notably, for SED we follow a hierarchical approach, where, first, a CNN operating over long-duration audio windows determines adaptive thresholds indicating how likely it is for each class to exist, and, subsequently, an ensemble of CNNs operating over shorter-duration windows determines the exact moments each class occurs.

1.2 Thesis Structure

The remainder of the thesis is organized as follows:

- In chapter 2, we provide a brief introduction to Deep Learning alongside with the main Neural Network architecture we are going to use in our methodology.
- In chapter 3, we give a short description of the DCASE community and the challenge we took part with this thesis.
- In chapter 4, we present the dataset we use for the development of our system. The dataset is provided by the DCASE Challenge.
- In chapter 5, we propose our method in order to solve the problem of Sound Event Detection and Localization.
- In chapter 6, we present the results of our system, alongside with graphs that analyze every design decision we made for our system.
- Finally, in chapter 7, we provide a conclusion to this thesis and some notes for future work that extends our methodology.

Chapter 2

Deep Learning

Deep Learning is a subset of the vast field called *Artificial Intelligence* (AI). In computer science, artificial intelligence, sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and animals. Colloquially, the term "artificial intelligence" is used to describe machines that mimic "cognitive" functions that humans associate with other human minds, such as "learning" and "problem-solving" [23].

Deep Learning, belonging to the field of Machine Learning or Statistical Learning, uses statistical methods in order to "learn" from data and generalize to new, unseen examples.

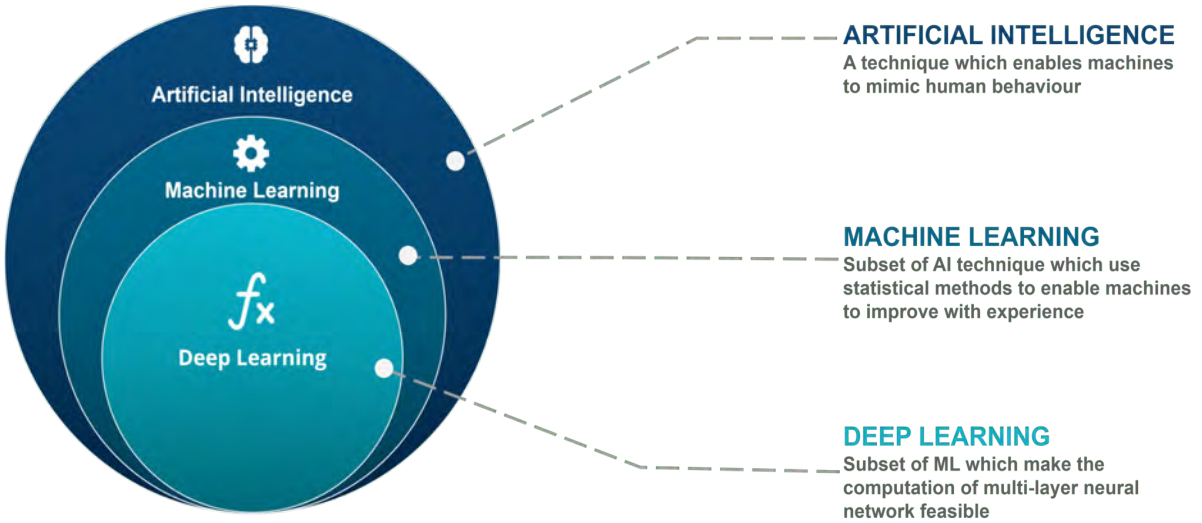


Figure 2.1: AI, Machine Learning, and Deep Learning (Figure from [24])

2.1 Artificial Neural Network

Artificial Neural Networks (ANN) are inspired by the biological neural networks which exist in all animals' brain. They are considered the most sophisticated Machine Learning algorithm we have, capable of solving many different and complex problems.

An ANN is composed of neurons, organized in layers. In its simplest form, the Feedforward Neural Network, each neuron is connected with all the neurons of the previous layer.

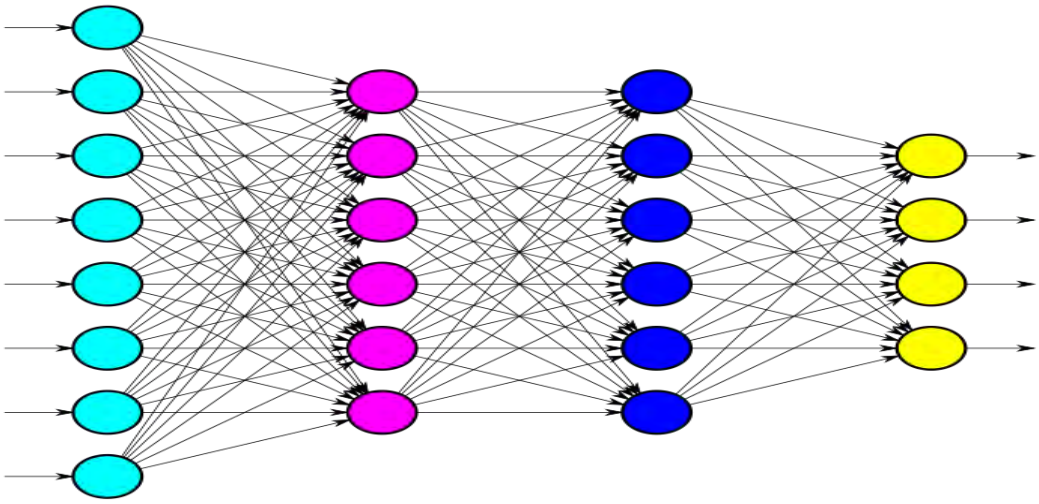


Figure 2.2: A FeedForward Neural Network (Figure from [25])

Each neuron is a small computational unit and its parts are:

1. Inputs (the outputs of the connected neurons - $\mathbf{x}$)
2. Weights (one weight associated with each input - $\mathbf{w}$)
3. Activation function (ϕ)
4. Output (y)

The output of each neuron is then calculated as:

$$y = \phi(\mathbf{w}^T\mathbf{x})$$

Common activation functions are:

1. Binary step

$$y = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{2.1}$$

2. Sigmoid

$$y = \frac{1}{1 + e^{-x}} \tag{2.2}$$

3. Identity

$$y = x \tag{2.3}$$

4. ReLU

$$y = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{2.4}$$

Usually, sigmoid and linear activations are used in the output neurons, while ReLU is used in all the hidden neurons due to the *Vanishing Gradient* problem [26].

Although a neuron has few capabilities, the combination of many neurons can lead to many and complex functions. In fact, it is proven that even a simple feedforward neural network with only one hidden layer is a universal approximator [27].

2.2 Convolutional Neural Network

A *Convolutional Neural Network* (CNN) [28] is a type of Neural Network based on the principle of *weight sharing*. Weight sharing is a technique of reducing the number of free parameters by setting many weights to the same value. CNNs are inspired by the primary visual cortex (V1 part) and are considered the state-of-the-art solution in tasks associated with image, video, and sound recognition.

Each layer of a CNN is usually either a *Convolutional* layer or a *Pooling* layer except for the final few layers which are fully connected layers (FeedForward NN) (see Figure 2.3).

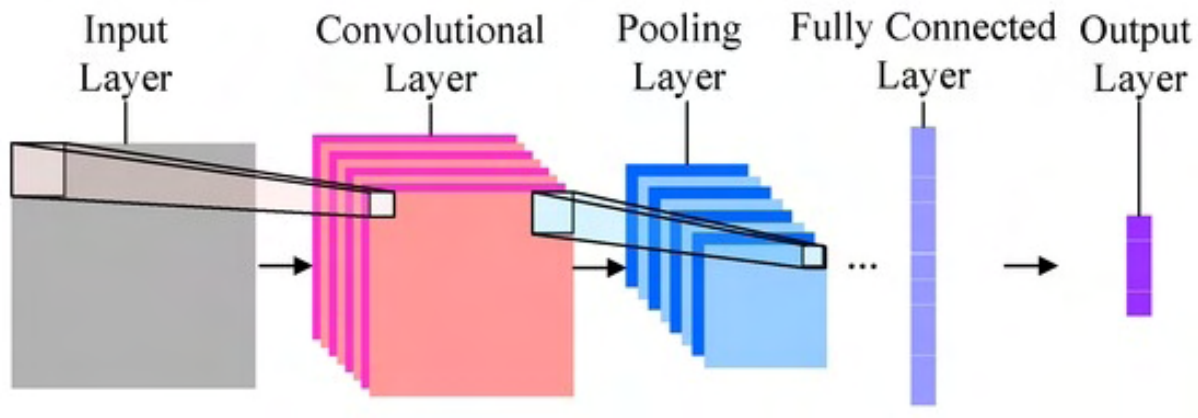


Figure 2.3: Convolutional Neural Network architecture (Figure from [29])

Convolution is the element-wise product of the layer's kernel (or filter) with the input elements (Figure 2.4). Pooling is used to reduce the size of the representation and prevent overfitting. Usually *Max Pooling* is used, but there is also the choice of *Average Pooling* (Figure 2.5).

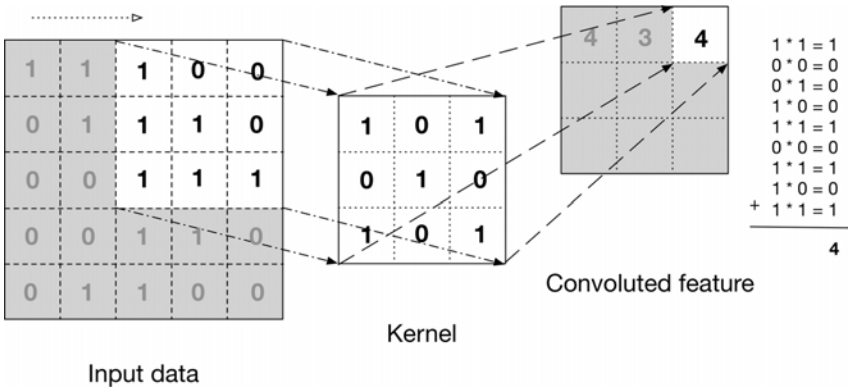


Figure 2.4: Convolution operation (Figure from [30])

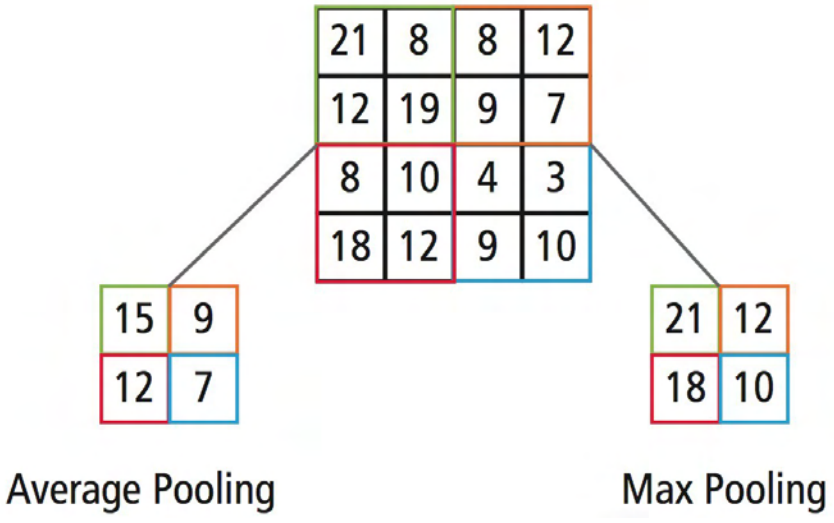


Figure 2.5: Pooling operation (Figure from [30])

Chapter 3

The DCASE Challenges

DCASE (Detection and Classification of Acoustic Sound Events) [31] is an effort by a rapidly grown worldwide community of researchers who are interested in all aspects of environmental sound classification and detection. DCASE is composed of researchers from both academia and industry, offering a platform for discussion and sharing of ideas.

Starting in 2016, DCASE has been hosting a yearly Challenge and a Workshop. This year's challenges were:

- **Task 1**, Acoustic scene classification
- **Task 2**, Audio tagging with noisy labels and minimal supervision
- **Task 3**, Sound event localization and detection
- **Task 4**, Sound event detection in domestic environments
- **Task 5**, Urban sound tagging

The current thesis is our proposed method to Task 3 [1; 2].

Chapter 4

Analysis of DCASE Task 3 Data

There are two available datasets of an identical sound scene, **TAU Spatial Sound Events 2019 - Ambisonic** and **TAU Spatial Sound Events 2019 - Microphone Array**, with the only difference being in the audio format [32]. We choose the **Microphone Array** which provides **four-channel** directional microphone recordings from a tetrahedral array configuration. The recordings are sampled at 48kHz, but in our system, we downsample them to 16kHz.

The development dataset consists of 400 one-minute long recordings, while the evaluation dataset consists of 100 one-minute long recordings. The development dataset is also associated with a pre-defined **four-way cross-validation split**. Each recording is synthesized using spatial room impulse responses (IR) in one out of five possible indoor locations. The indoor locations are in the Tampere University campus at Hervanta, Finland and the given description of each one is:

1. **Language Center** - Large common area with multiple seating tables and carpet flooring. People chatting and working.
2. **Reaktori Building** - Large cafeteria with multiple seating tables and carpet flooring. People chatting and having food.
3. **Festia Building** - High ceiling corridor with hard flooring. People walking around and chatting.
4. **Tietotalo Building** - Corridor with classrooms around and hard flooring. People

walking around and chatting.

5. **Sähkötalo Building** - Large corridor with multiple sofas and tables, hard and carpet flooring at different parts. People walking around and chatting.

The Eigenmike spherical microphone array [33] was used for the collection of the real-life IR recordings (see Figure 4.1).

Figure 4.1: The Eigenmike spherical microphone array (Figure from [34])

There are two tasks associated with this dataset (see Figure 4.2), which are:

1. Sound Event Detection (SED): Detect which classes exist at each time frame
2. Sound Event Localization (DOA - Direction Of Arrival): Localize the classes detected at SED task

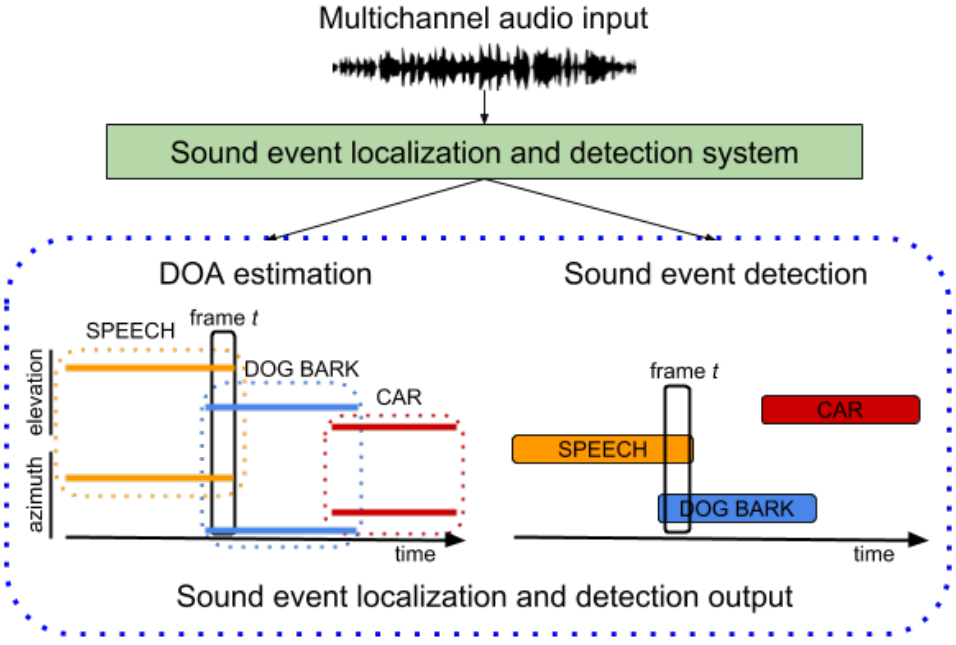


Figure 4.2: Overview of the two tasks (Figure from [1])

4.1 Sound Event Detection

At any given time, there are at most two different sound events, which are taken from the sound event dataset of DCASE16-task2 [15]. There are eleven sound event classes:

1. Knock
2. Drawer
3. Clearthroat
4. Phone
5. KeysDrop
6. Speech
7. Keyboard
8. Pageturn
9. Cough
10. Doorslam
11. Laughter

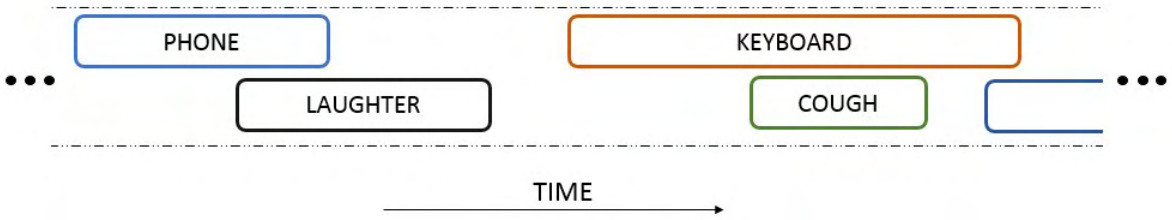
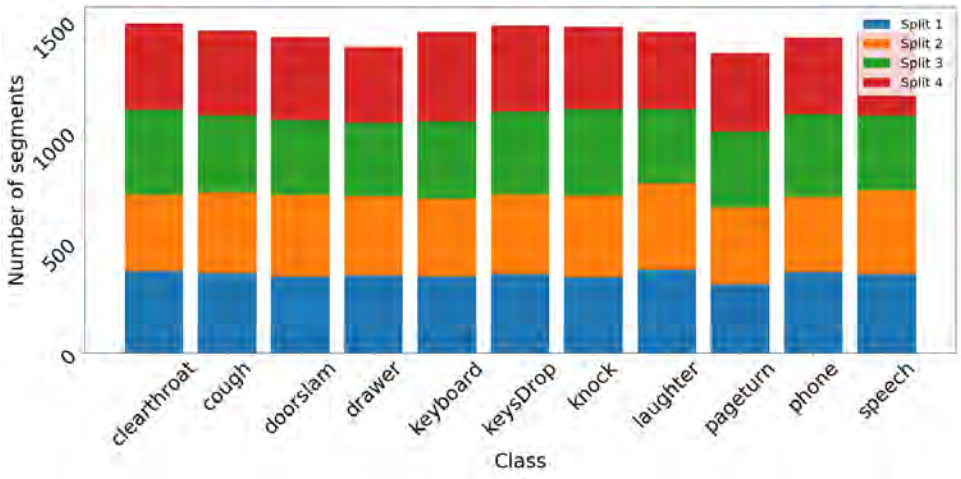


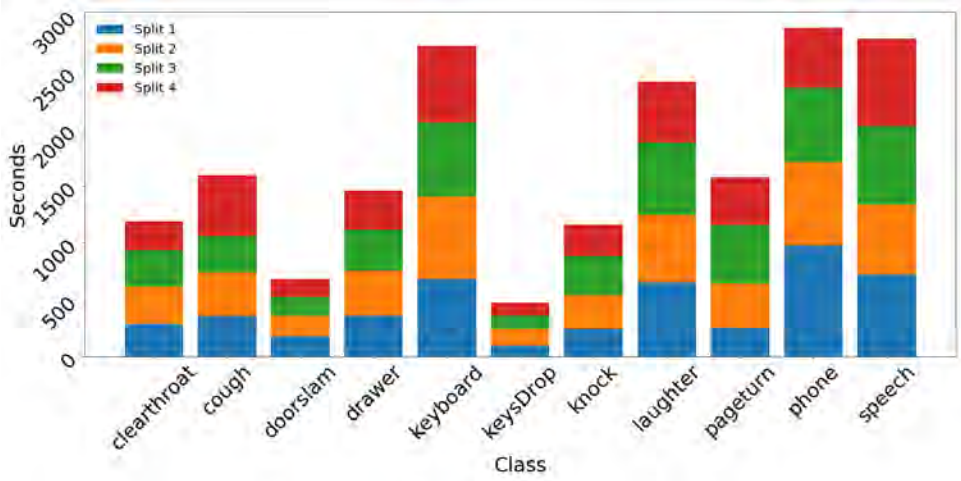
Figure 4.3: Overview of a one-minute recording. At any given time there are zero, one or two classes. Here we see one *Phone* segment overlapping with one *Laughter* segment, as well as a *Keyboard* segment overlapping with a *Cough* segment. Also, between *Laughter* and *Keyboard* there is a segment that does not belong to a class.

A one-minute recording consists of segments which belong to the above 11 classes. An overview of these recordings is given in Figure 4.3.

The duration of each class segment ranges from 205ms to 3.335sec. Although the number of segments in the recordings is almost the same for each class, there are great differences in the total duration of each class (Figure 4.4).



(a) Number of segments for each class



(b) Total duration of each class

Figure 4.4: Number of segments and total duration for each class in the 4 development splits

4.2 Sound Event Localization

The Direction of Arrival (DOA) of each segment is given in spherical coordinates. In Figure 4.5 we can observe the relationship between the spherical and the cartesian coordinate system. Each sound event is associated with one out of 324 possible combinations of azimuth and elevation values. Azimuth values lie in the range [-180°, 170°] while elevation values belong to the range of [-40°, 40°], both with a resolution of 10°. Note that the DOA of each segment remains the same throughout the whole duration of that segment.

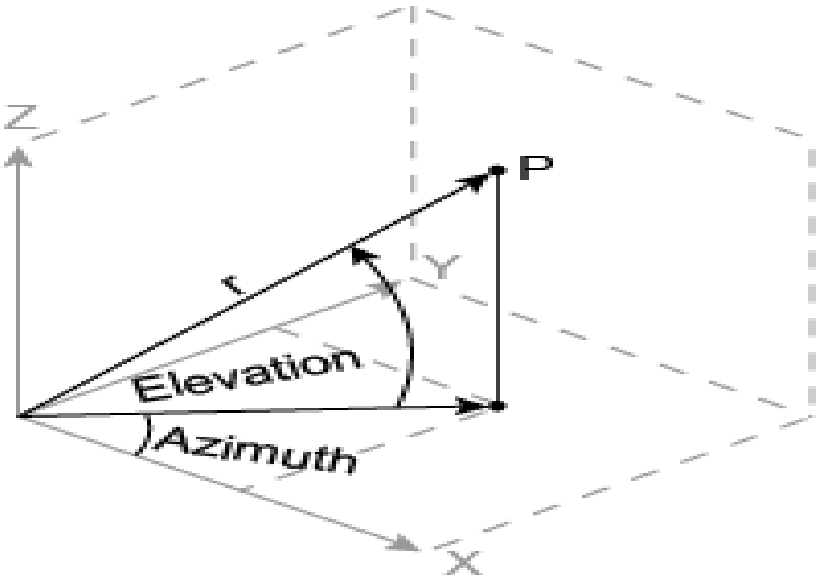
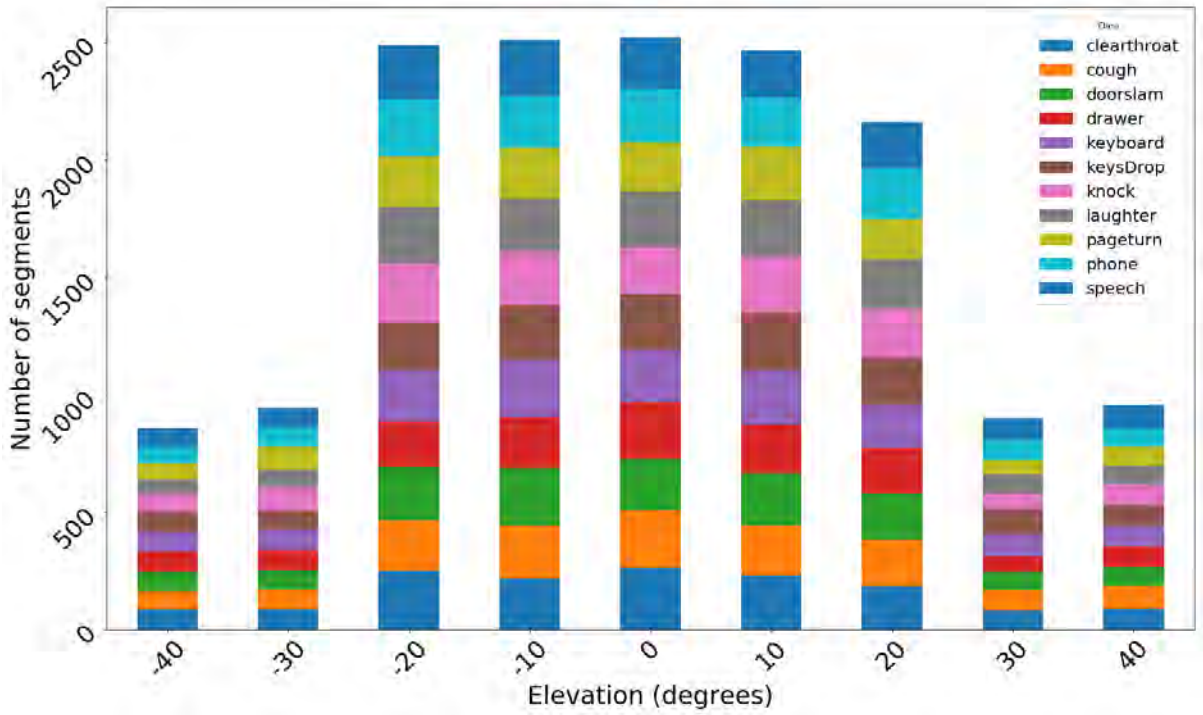
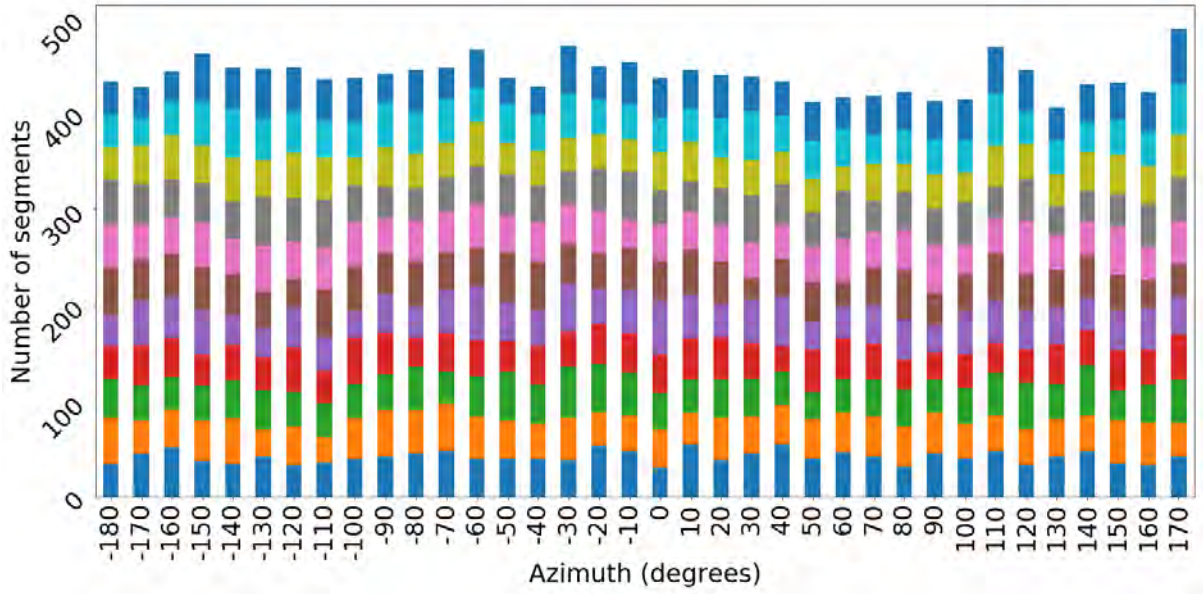


Figure 4.5: Azimuth and Elevation (spherical) coordinate system compared to the cartesian coordinate system (Figure from [35])

The elevation values are equally distributed between the segments of each class, but, in general, there are many more segments with elevations in the range of [-20°, 20°] (Figure 4.6a). On the other hand, there is about the same number of segments for each azimuth value (Figure 4.6b).



(a) Distribution of elevation values among classes



(b) Distribution of azimuth values among classes

Figure 4.6: Elevation and Azimuth distributions

Chapter 5

Our Sound Event Detection and Localization Algorithm

As we mentioned before, we have to solve two problems, SED (Sound Event Detection) and DOA (Direction Of Arrival). In our method, we first address the SED sub-task and then the DOA one. Specifically, we develop a hierarchical approach to the former, determining the existence of each sound event class at each time-frame. For this purpose, first a "long SED model" estimates adaptive thresholds for each class, also taking into account the class prior probabilities. Then, an ensemble of "short SED models" determines the exact time-frames each class exists, exploiting the aforementioned thresholds. Following SED, we utilize a DOA model to localize the source of each detected event, estimating its elevation and azimuth values. All models are multi-channel CNNs, operating on raw waveforms or spectrograms over sliding windows of different durations, as detailed next. A schematic overview of the system is provided in Figure 5.1.

Our final predictions should be in the form:

frame number (int),active class index (int),azimuth (int),elevation (int)

where each time frame is of 20ms duration, resulting to 3000 time frames for each one-minute long recording.

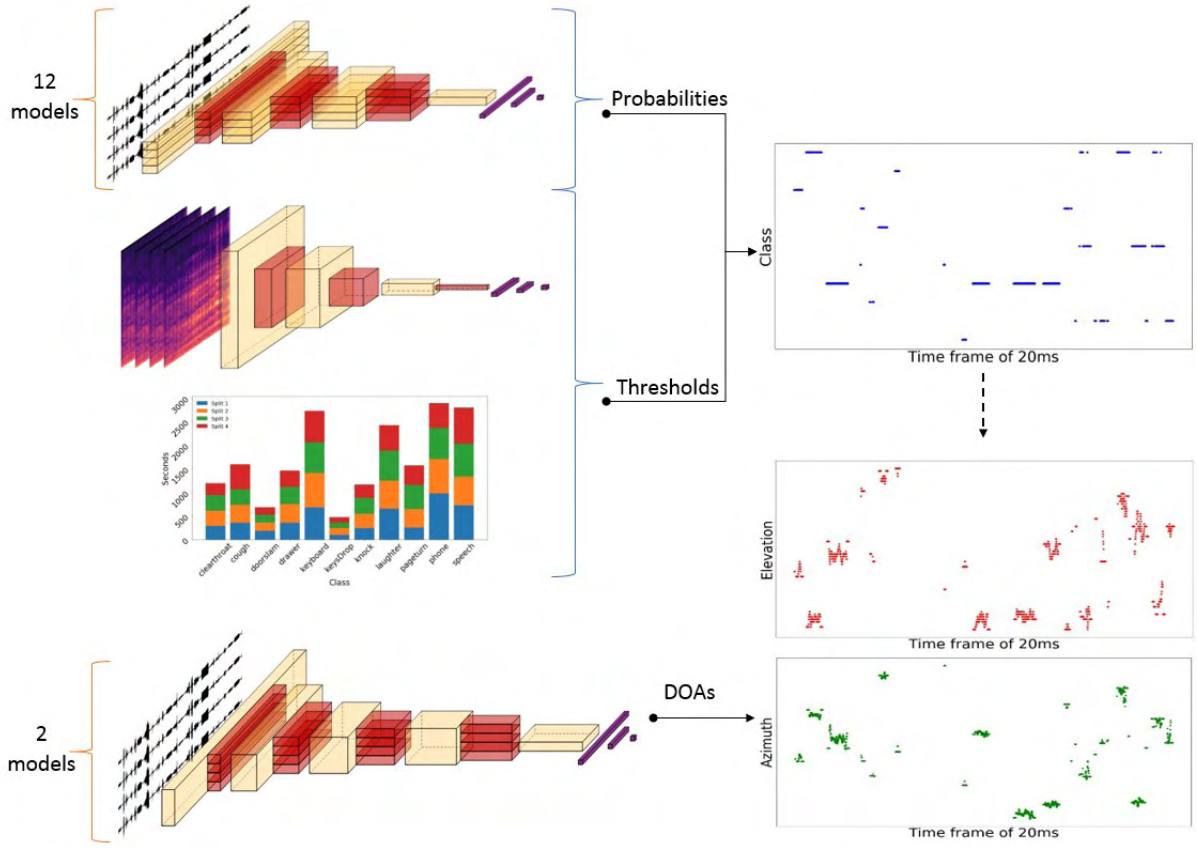


Figure 5.1: System overview (CNNs are drawn using the PlotNeuralNet software [36]).

5.1 SED Task

The objective of the SED task is to predict which classes exist in each time frame. As a result, our predictions from SED task are going to be of the form

frame number (int),active class index (int)

Given that there are overlapping sound events, our problem is a multi-label problem and not a softmax problem, resulting in a sigmoid activation function for each of the 11 output neurons (one for each class).

There are three metrics associated with the SED task [37; 38].

1. SED Error

$$\text{SED Error} = \frac{S + D + I}{N} \tag{5.1}$$

where:

- S: Substitutions
- D: Deletions
- I: Insertions
- N: Reference events

2. F-score

$$\text{F-score} = \frac{2PR}{P + R} \tag{5.2}$$

where:

- TP: True Positives
- FP: False Positives $(= S + I)$
- FN: False Negatives $(= S + D)$
- P: Precision,

$$P = \frac{TP}{TP + FP} \tag{5.3}$$

- R: Recall,

$$R = \frac{TP}{TP + FN} \tag{5.4}$$

3. Frame-Recall

$$\text{Frame-Recall} = \frac{\sum_{t=1}^{T} 1(D_R^t = D_E^t)}{T} \tag{5.5}$$

where:

- D_R^t: number of DOAs in time frame t
- D_E^t: predicted number of DOAs in time frame t
- T: Total time frames

We should improve all three metrics simultaneously. An ideal model would score

1. SED Error = 0
2. F-score = 1 (100%)
3. Frame-Recall = 1 (100%)

5.1.1 Short Models

5.1.1.1 Data Preprocessing

First of all, we apply the following preprocessing step to the original one-minute recordings. We create a different file for each segment of the recordings and use these new files as training data (we keep also the segments that belong to no class) as shown in Figure 5.2.

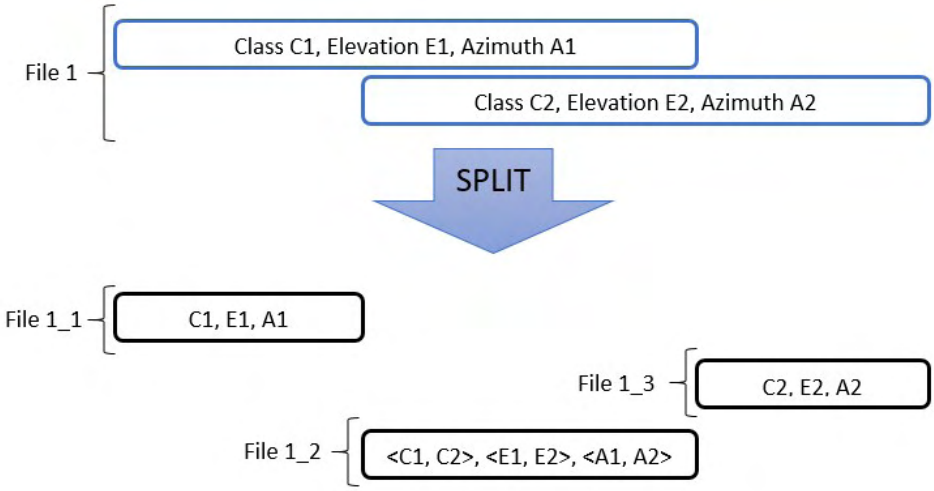


Figure 5.2: Splitting procedure of original one-minute recordings

5.1.1.2 Architecture

We create an ensemble of multi-channel CNNs (12 in total, as explained in the next paragraph), all with the architecture of Table 5.1. These operate on raw audio waveforms over short-duration windows of 100ms or 200ms, with these values determined after experimenting with various window lengths on the Challenge development data. We do not apply any preprocessing to the four channels (other than their downsampling to 16 kHz), and we use all four microphone data streams as input to the CNNs. The output layers of the models have 11 neurons (same as the number of sound event classes), each providing the probability of its corresponding class, following sigmoid activation. Note that, during training, windows with no sound events are kept, and windows with overlapping events are assigned to all occurring events inside them (maximum of two), while they slide in steps equal to half their duration, i.e. by 50 or 100ms. All CNNs are

trained with a binary cross-entropy (5.6) objective (as we deal with a multi-label problem) using the Adam optimizer and early stopping to prevent overfitting, employing the Keras API for development [39].

$$\text{Cross-Entropy} = -\Big(y_{true}\log(y_{pred}) + (1 - y_{true})\log(1 - y_{pred})\Big) \tag{5.6}$$

Input (4 x segment size)
100 filters, Conv 1x10, ReLU MaxPool 1x5
200 filters, Conv 1x10, ReLU MaxPool 1x6
300 filters, Conv 1x10, ReLU MaxPool 1x7
500 filters, Conv 4x1, ReLU
Flatten Dropout 0.6
1000 neurons, Dense, ReLU Dropout 0.3
11 neurons, Dense, sigmoid

Table 5.1: Architecture of the short SED models. Segment size is 1600 for 100ms windows and 3200 for 200ms ones.

5.1.1.3 Data augmentation and final models

In order to have more segments with overlapping sounds, we employ data augmentation as follows: we add segments, each belonging to only one class, two at a time. Concerning the Challenge evaluation metrics, we observed that datasets with more overlapping segments tend to yield better Frame-Recall results, while data with less overlapping segments tend to perform better in terms of SED error and F-Score. As we wish to improve all three metrics simultaneously, we choose to create different models, trained on data with various

degrees of artificial overlap, and then ensemble them. Thus, we create six datasets, having 0%, 5%, 10%, 20%, 30%, and 40% extra overlapping segments, and we train two different CNNs on each (i.e. with input window sizes of 100ms and 200ms length), thus resulting in 12 models. The process is repeated for each of the four given development data splits.

5.1.1.4 Predictions and Ensembling

After having the short models, we can proceed to the predictions for the original one-minute recordings. The predictions are floating point numbers in the range [0, 1] and stand for the probability of each class to exist. We are asked to predict the existence of each class in time frames of 20ms, so we convert each prediction for 100ms or 200ms (depending on the model) to 20ms time frames just by setting each subframe of 20ms to the value of the major frame of 100 or 200ms (for example, for the first 100ms time frame, we set the first five 20ms time frames in the same value). In order to have more robust predictions we predict with step 20ms, meaning that, for the case of 100ms models, for example, we predict for the 0-100ms, then for the 20-120ms, then for the 40-140ms, etc. and then we average the predicted probabilities for the same 20ms time frames. The same technique is applied for all 12 models we have and, after that, we average the predicted probabilities of all twelve models.

5.1.2 Long Model

A major issue in multi-label problems concerns the choice of class thresholds, used to decide if a class exists or not. A simple approach is to set all thresholds to 0.5, as in the Challenge baseline system [40], however, their careful tuning may yield significant improvements. For example, in [10] exhaustive search is utilized to yield a single optimal threshold for all classes, whereas in [11; 12] separate thresholds are employed for each class, found by exhaustive search. Nevertheless, both approaches may be prone to overfitting due to the exhaustive search used.

To prevent overfitting, we opt to create a SED model operating on longer-duration data windows. Our motivation stems from the expectation that such a model will provide a “bigger picture” concerning class existence, and thus can help in determining class

thresholds adaptively. These can then be utilized in conjunction with the outputs of the short SED models to predict the exact timeframes in which each sound event occurs. For this purpose, we create a multi-channel CNN that operates on power spectrograms over signal windows of one-second duration (sliding in 100ms steps during training), with the spectrograms generated by the libROSA package under its default parameters [41]. We use all available channels, ending up with four spectrograms as input. For data augmentation, we consider all permutations of the four channels, resulting in 24 times more training data. The details of the long SED model architecture are provided in Table 5.2.

Input (4x128x32)
40 filters, Conv 1x6x1, ReLU MaxPool 1x3x1
60 filters, Conv 1x1x6, ReLU MaxPool 1x1x3
80 filters, Conv 1x6x6, ReLU MaxPool 1x3x3
Flatten Dropout 0.5
500 neurons, Dense, ReLU Dropout 0.3
11 neurons, Dense, sigmoid

Table 5.2: Long SED model architecture.

5.1.3 Adaptive thresholds and SED predictions

To determine the class thresholds we work at a time-resolution of 20ms, exploiting the long SED model predictions. These fine-resolution predictions are obtained by averaging the coarser-resolution probabilities of each class over all 1s-long windows that contain the given 20ms time-frame, while sliding by 200ms. A first approach to determine the desired

thresholds is to simply set them to

$$\theta_c^t = 1 - \mathrm{lp}_c^t \tag{5.7}$$

where lp_c^t denotes the long SED model prediction (probability) of class c at time-frame t , and θ_c^t is the corresponding threshold.

In general, however, we do not wish the thresholds to be too close to 1, in order to guard against false negatives of the long SED model. Thus, we choose to smooth (5.7) by multiplying the thresholds with a number within the [0.6, 0.9] range. This number is different for each class, and it is based on its total duration in the training data (class prior), meaning that less frequent classes tend to have lower thresholds. The resulting thresholds are given by

$$\theta_c^t = \left(1 - \mathrm{lp}_c^t\right)\left(0.6 + 0.3\frac{p_c - p_{min}}{p_{max} - p_{min}}\right) \tag{5.8}$$

where

$$p_c = \frac{duration_c}{\sum_{k=1}^{n_c classes} duration_k} \tag{5.9}$$

denotes the prior of class c (based on duration), while p_{min} and p_{max} represent the minimum and maximum of all class priors, respectively.

The desired SED results are finally derived at a time-resolution of 20ms, by employing the ensemble of the 12 short SED models of Section 5.1.2 and the adaptive thresholds of (5.8). Specifically, let sp_c^t denote the combined short model prediction of class c at time-frame t as described in Section 5.1.1.4. To determine if class c exists in time frame t:

$$\mathrm{exist}_c^t = \begin{cases} 1 & \mathrm{sp}_c^t \geq \theta_c^t \\ 0 & otherwise \end{cases} \tag{5.10}$$

5.2 DOA Task

Following SED, we proceed to the DOA sub-task. Having our SED predictions, we have to augment them with elevation and azimuth values. The metric which is associated with the DOA sub-task is called DOA Error [38]:

$$\text{DOA Error} = \frac{1}{\sum_{t=1}^{T} D_E^t} \sum_{t=1}^{T} \mathcal{H}(\mathbf{DOA}_R^t, \mathbf{DOA}_E^t) \tag{5.11}$$

where

- T: number of time frames
- $\mathbf{DOA}_R^t$: list of all reference DOAs at time-frame t
- $\mathbf{DOA}_E^t$: list of all estimated DOAs at time-frame t
- D_E^t: number of estimated DOAs
- $\mathcal{H}$: the Hungarian algorithm for solving the assignment problem

An ideal model would score DOA Error = 0.

For this purpose, and similarly to the short SED models, we create short models for DOA estimation that provide 22 numbers at their output layer, i.e. the elevation and azimuth for each of the 11 classes. The goal is, given a raw multi-channel audio segment of short duration, to predict the DOA of each class, no matter if it exists or not (SED results will determine what to keep). Specifically, we create two CNNs, with their architecture detailed in Table 5.3. The CNNs operate on four channels of raw audio over windows of 100ms or 200ms in duration that, during model training, slide at steps of 50ms or 100ms, respectively. For training the two networks, we use the same data as in the SED sub-task, but exclude audio with no sound events, as such data are not associated with DOA values. We employ the mean squared error (5.12) loss as training objective,

$$\text{Mean-Squared-Error} = \left(y_{true} - y_{pred}\right)^2 \tag{5.12}$$

but slightly modified, as we calculate it only in the 2 (in the case of one class) or the 4 (for two overlapping classes) output neurons of interest. As before, we use the Adam

optimizer and early stopping to prevent overfitting. DOA estimation occurs at a time resolution of 20ms, first by averaging the elevation and azimuth predictions for the 20ms time frame of interest within the model sliding windows, and subsequently averaging the predictions across the two models.

A problem arises in this approach towards the boundaries of each segment. To prevent noisy DOA estimates there, these are smoothed by setting predictions for the first and last 300ms of each segment to the minimum or maximum of that sub-segment (depending on the relative position to the zero), thus preventing steep DOA ascents or descents. An example of this process is depicted in Figure 5.3.

Input (4 x segment size)
100 filters, Conv 4x10 (same padding), ReLU MaxPool 1x3
200 filters, Conv 4x10 (same padding), ReLU MaxPool 1x5
300 filters, Conv 4x10 (same padding), ReLU MaxPool 1x5
400 filters, Conv 4x10 (same padding), ReLU MaxPool 1x5
500 filters, Conv 4x1 (same padding), ReLU
Flatten Dropout 0.5
1000 neurons, Dense, ReLU Dropout 0.3
22 neurons, Dense, linear

Table 5.3: Architecture of the DOA models.

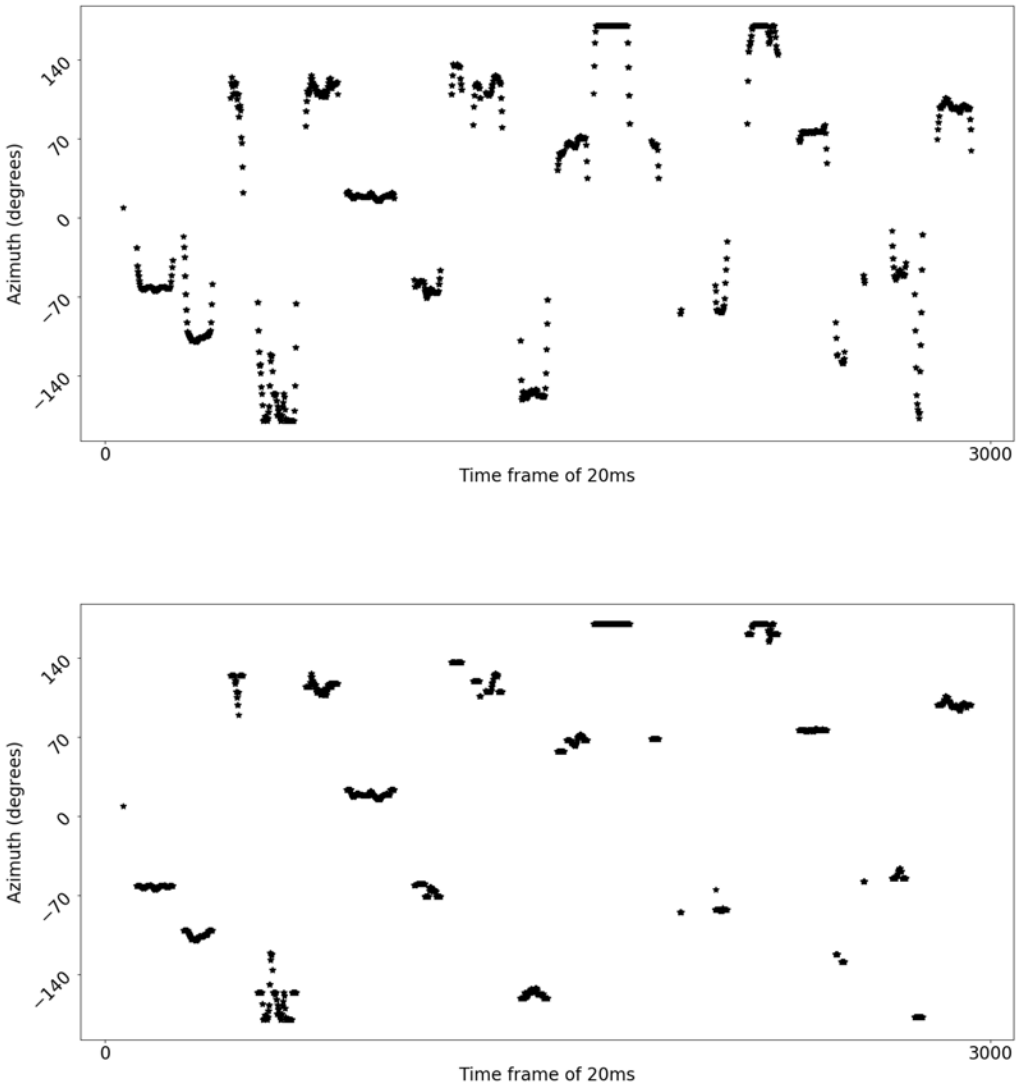


Figure 5.3: Example of DOA (here, azimuth) estimate smoothing at segment edges: (top) before smoothing; (bottom) after smoothing.

5.3 Baseline

The baseline system, which was the best submission in the task3 of DCASE17 Challenge [14], solves both problems (SED and DOA) simultaneously. The overview of the system, called SELDnet (Sound Event Detection and Localization network), is provided in Figure 5.4.

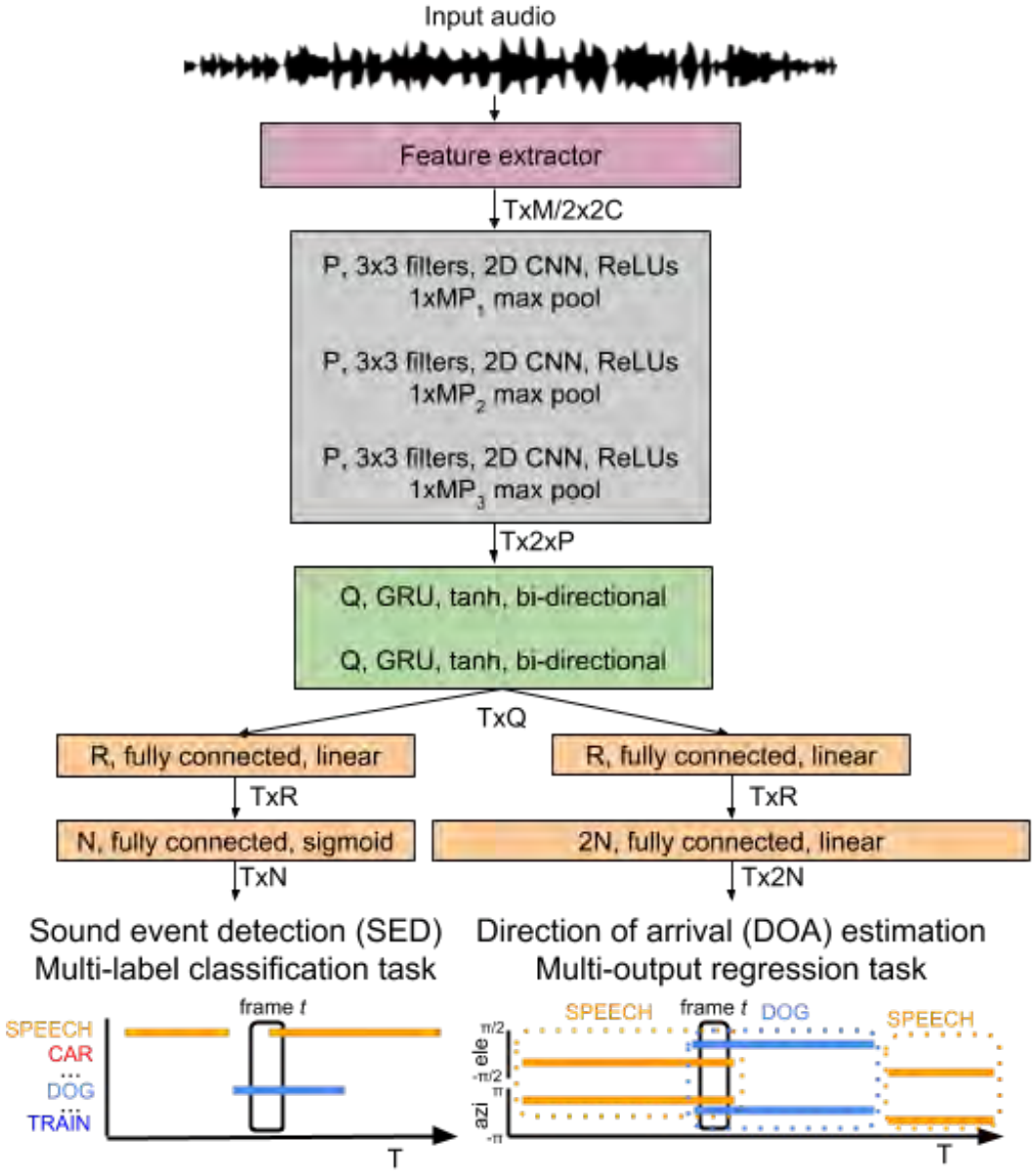


Figure 5.4: SELDnet overview. In the benchmark method, the variables in the image above have the following values, T = 128, M = 2048, C = 4, P = 64, MP1 = MP2 = 8, MP3 = 4, Q = R = 128, N = 11. (Figure from [42])

The major differences compared to our proposed method are:

1. Solves both problems simultaneously
2. Uses both Convolutional and Recurrent Neural Network
3. Uses only spectrograms as input features
4. Outputs directly for time frames of 20ms and there is no need for post-processing
5. Uses *mean_squared_error* to train the DOA outputs, by setting to zero the DOAs of the inactive classes
6. Uses 0.5 threshold for all classes
7. Introduces a trade-off between SED and DOA accuracy

Chapter 6

Results

6.1 Overview

We first present in Table 6.1 a summary of our system results on the Challenge development set over the four data splits, in terms of the four Challenge metrics and their combination (SELD score). We can readily observe that, compared to the baseline that is provided by the Challenge organizers, we achieve a 3% relative reduction in the SELD score (from 0.22 to 0.213). In terms of the individual metrics, we obtain relative improvements of 12% in SED error (from 0.35 to 0.309), 2% in F-score, 36% in DOA error (from 30.8 to 19.8), but trail significantly in the frame-recall metric, where we achieve only 75.3% over 84.0% of the baseline.

system	SED error	F-score	frame-recall	DOA error	SELD score
baseline	0.350	80.0%	**84.0%**	30.8°	0.220
proposed	**0.309**	**81.2%**	75.3%	**19.8°**	**0.213**

Table 6.1: Results of the proposed system on the development set compared to the baseline, in terms of the five Challenge metrics.

In the figures below we present results to highlight performance differences between the various design choices of our developed system components. We focus primarily on the relative merits of the various short SED models, of approaches to class threshold estimation, and of the DOA models and the smoothing of DOA estimates.

First, in Figure 6.1 we depict performance of the short SED models of Section 5.1.2 and

their ensembles in terms of SED error, F-score, and frame-recall (difference from 1 is shown for the latter two). We also depict results for additional window sizes, namely 300ms and 400ms. Each bar shows results of the ensemble of six models, trained on various data augmented sets (from 0% to 40%, as discussed in Section 5.1.2), with the error bars indicating the range of the individual model results. Note that the 12-model ensemble results are also shown ("100+200 ensemble"). We can readily observe that shorter window sizes (100ms) yield the best results in terms of frame-recall, mainly because two sounds may overlap for very short periods of time, but have much worse results in SED error and F-score, because short windows may not carry adequate class information. On the other hand, medium window sizes (200ms) yield the best results in SED error and F-score, but worse frame-recall as they may fail to detect very short segments. Combining the two window sizes by model ensembling exploits the relative advantages of both, improving SED error and F-score significantly, but at minor detriment in frame-recall. Longer windows (e.g. 300ms or 400ms sizes) significantly degrade frame-recall, thus are not used in our system.

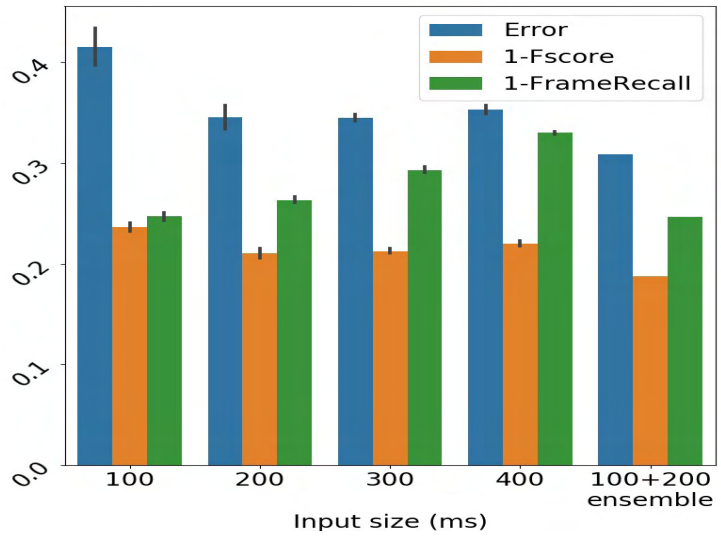


Figure 6.1: Results for the three SED metrics for 4 different input sizes, against our final ensemble results. The error bars on the top indicate the differences between the 6 datasets we used. We rejected the models with sizes 300 and 400ms mostly because they perform much worse in terms of Frame-Recall.

Next, in Figure 6.2 we examine the effect of class thresholds to the SED system performance. Thresholds fixed to 0.5 for all classes perform the worst, whereas adaptive thresholds estimated by means of (5.7) – labeled as "long" in the graph, perform better in all three

metrics (SED error, F-score, and frame-recall). Results further improve when adaptive thresholds are computed by (5.8) – labeled as "long and prior" in the bar-plot.

Finally, in Figure 6.3 we consider the DOA estimation component. There, we can readily observe the importance of DOA estimate smoothing, as systems "without" smoothing perform significantly worse than systems "with" it. Also, DOA models operating on windows of 100ms or 200ms in duration outperform systems built on 300ms windows. The ensemble of both 100ms and 200ms systems performs even better in terms of the DOA error metric.

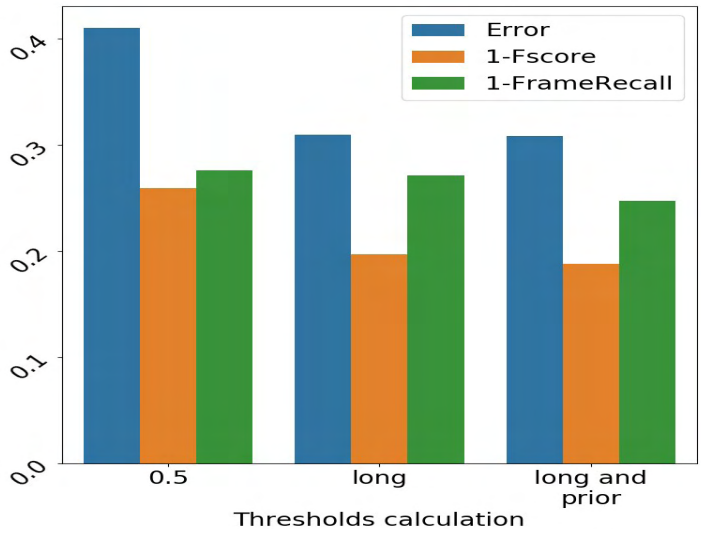


Figure 6.2: The results for the three SED metrics for 3 different types of thresholds.

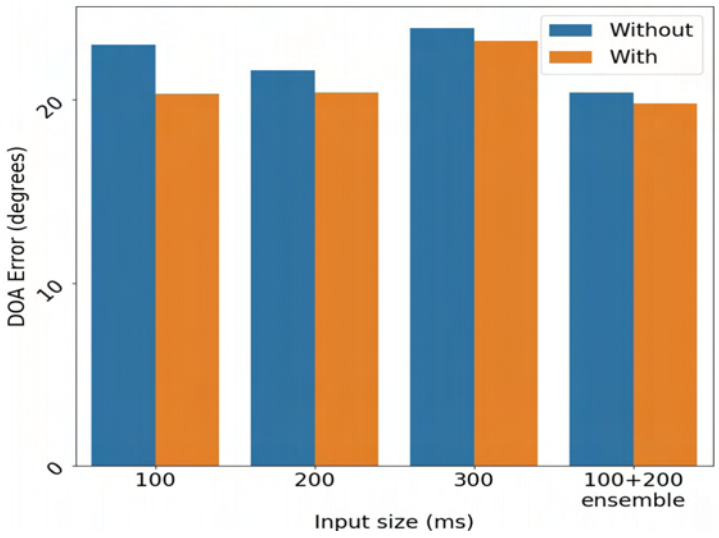


Figure 6.3: Results for the DOA error. We can see the importance of smoothing in the difference between "with" and "without", especially for input size 100ms.

6.2 SED task

In Figures 6.4, 6.5 we can observe the SED predictions for two of the 400 one-minute long recordings of the development dataset. First, in Figure 6.4 we have the predictions for a recording with no overlapping segments (*split1_ir3_ov1_70.wav*), while in Figure 6.5 the predictions come from a recording with overlapping segments (*split3_ir3_ov2_74.wav*). We observe, especially in the case of overlapping segments, that our model tends to have more *Deletions* instead of *Insertions* and, most importantly, *Substitutions*. This is mainly the reason our Frame-Recall is lower than the baseline while our SED_Error and, mostly, F-score are higher.

Also, in the case of overlapping segments, we can mention that sometimes one of the two classes dominates and the other one is not detected. This problem is greatly reduced by the 12 models ensembling.

Finally, there is a relatively high level of confusion between classes *Clearthroat* (class 2), *Cough* (class 5) (major confusion) and *Laughter* (class 10) (minor confusion). In fact, even a human listener sometimes confused these classes (especially the *Cough* and *Clearthroat*) in these recordings so we guess that with another dataset this problem may not exist.

The results for the three SED metrics are given in Table 6.2.

overlap	SED error	F-score	frame-recall
no	0.282	84.5%	87.0%
yes	0.32	79.3%	64%

Table 6.2: SED results on the development set for recordings with no overlapping segments and recordings with overlapping segments.

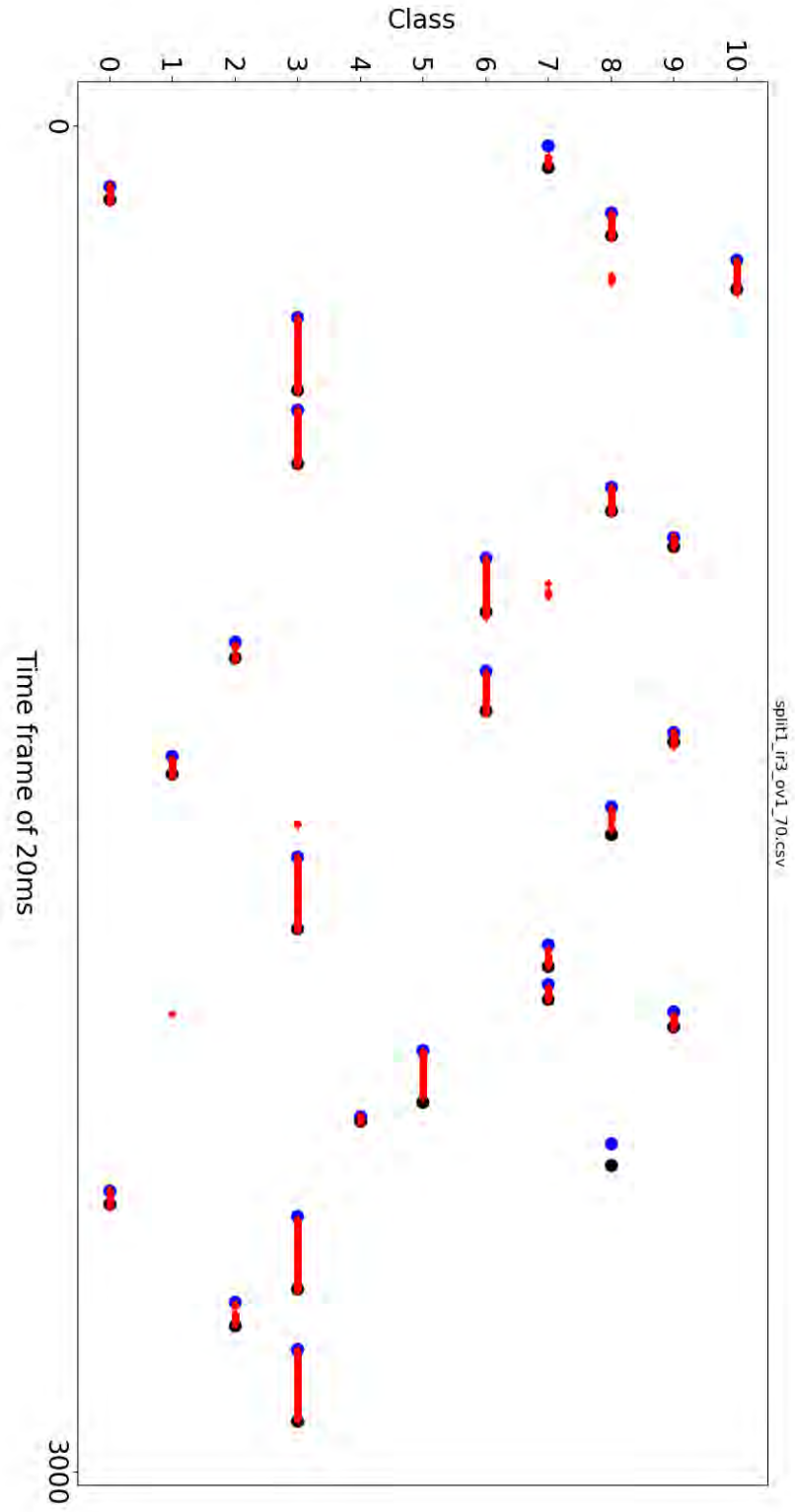


Figure 6.4: SED predictions with no overlapping segments. The blue dots indicate the start of each segment and the black dots its end.

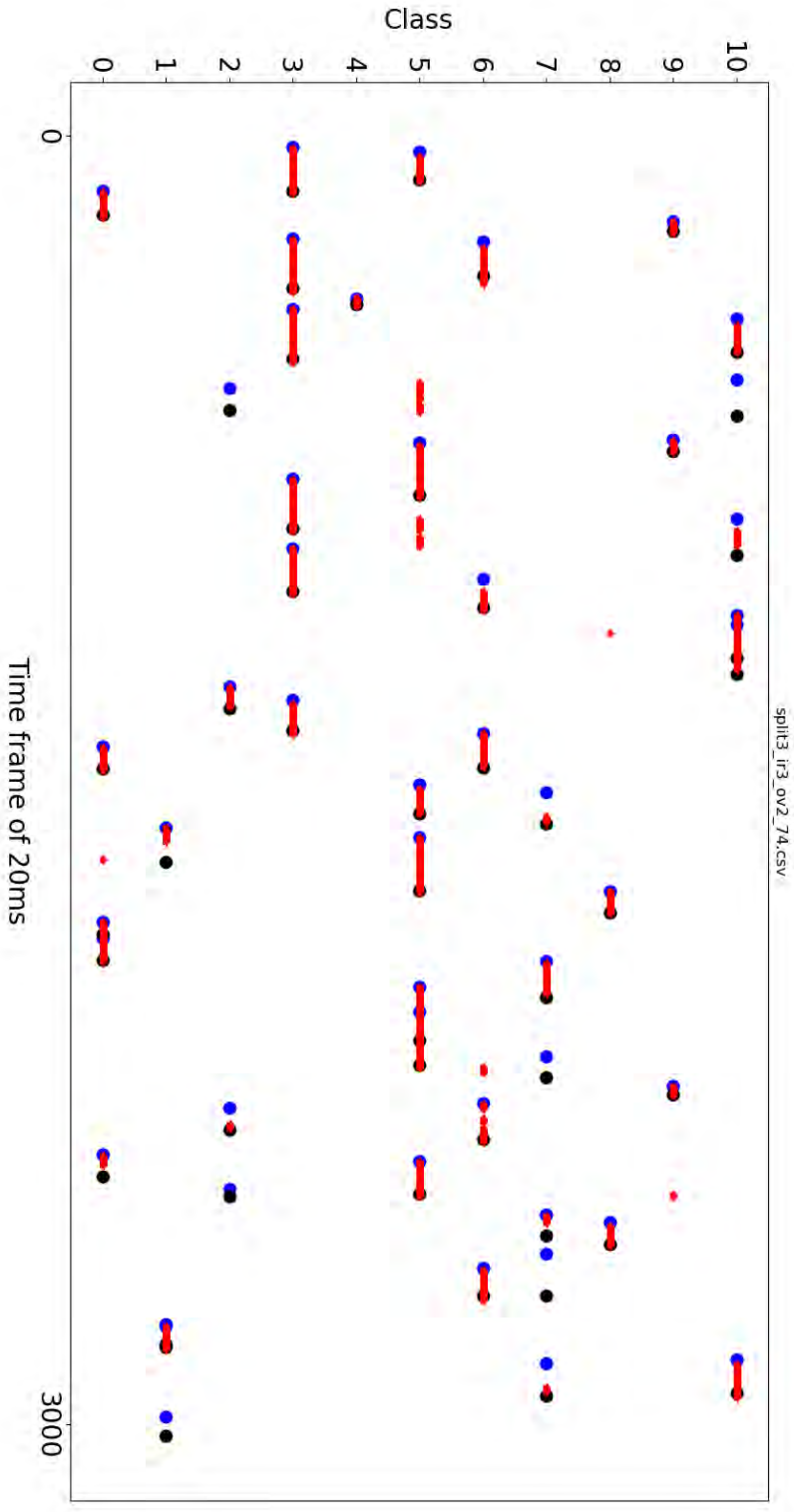


Figure 6.5: SED predictions with overlapping segments. The blue dots indicate the start of each segment and the black dots its end.

6.3 DOA task

In the DOA detection sub-task, things are simpler. Only one metric is associated with this task (DOA Error). Here again, as expected, results for recordings with no overlapping segments are better than results for recordings with overlapping segments. The exact scores are given in Table 6.3.

overlap	DOA Error
no	13
yes	24

Table 6.3: DOA results on the development set for recordings with no overlapping segments and recordings with overlapping segments.

In Figures 6.6, 6.7 we can observe the DOA predictions for two of the 400 one-minute long recordings of the development dataset. First, in Figure 6.6 we have the predictions for a recording with no overlapping segments (*split2_ir2_ov1_48.wav*), while in Figure 6.7 the predictions come from a recording with overlapping segments (*split1_ir2_ov2_51.wav*).

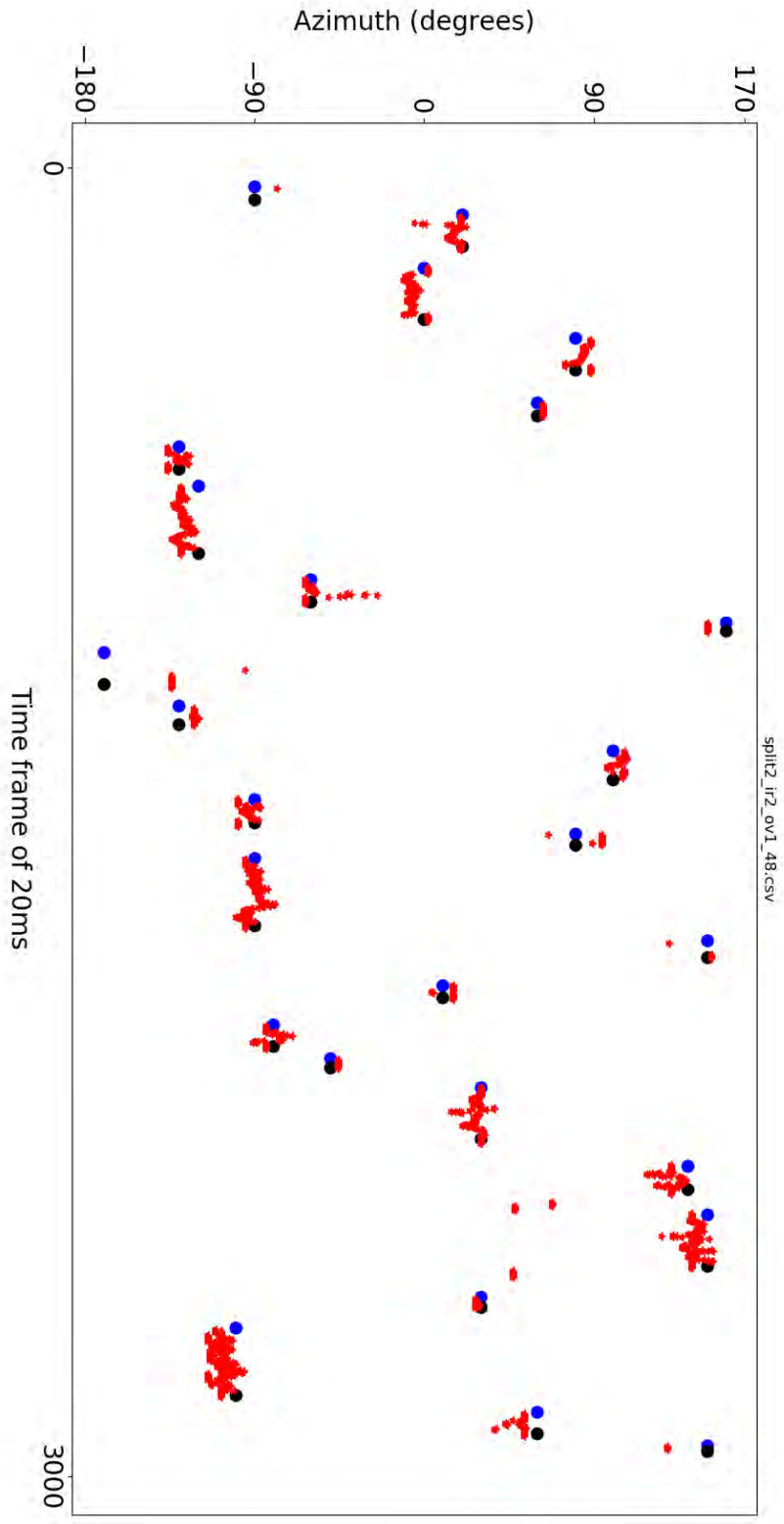


Figure 6.6: DOA predictions with no overlapping segments. The blue dots indicate the start of each segment and the black dots its end. Note that the frames and classes for which DOA predictions exist depend only on SED predictions.

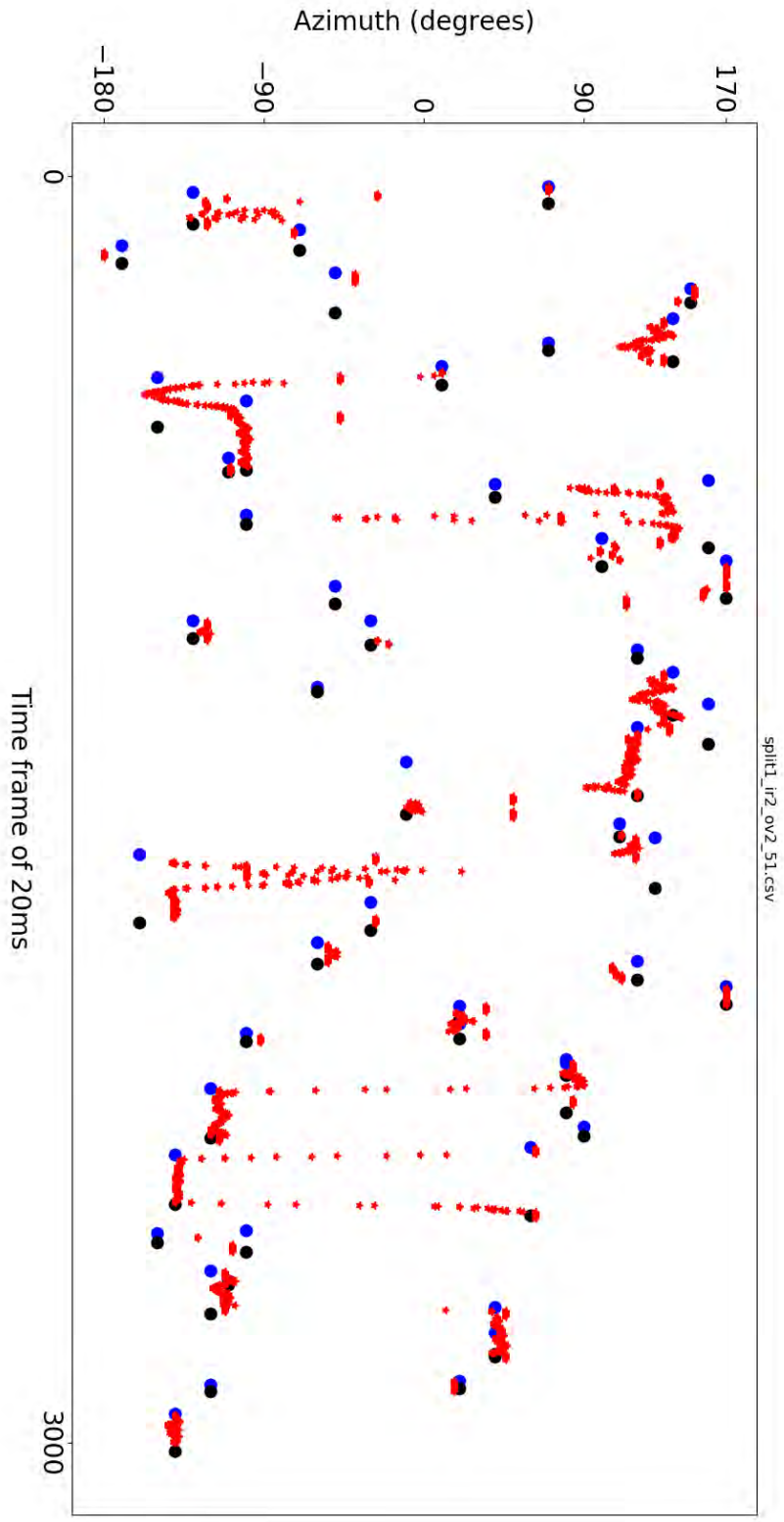


Figure 6.7: DOA predictions with overlapping segments. The blue dots indicate the start of each segment and the black dots its end. The steep descent or ascent which we observe is because some times one of the two segments dominates and leads to an immediate change to predictions, or because one of the two overlapping segments starts or stops to exist.

Chapter 7

Conclusion

In this thesis, we presented a solution for the SELD task (Sound Event Localization and Detection) using only CNNs, separately addressing SED and DOA estimation. In particular, we followed a hierarchical approach to SED, first determining adaptive class thresholds based on a CNN operating over longer windows, which we subsequently utilized in an ensemble of CNNs operating on shorter windows, also exploiting data augmentation techniques in their training. Our proposed method was submitted to the DCASE19 Challenge (task 3). We evaluated our system on the Challenge development dataset, outperforming the baseline in all Challenge metrics but the frame-recall one.

Nevertheless, in our research, we did not explore many other approaches. First of all, the use of Recurrent Neural Networks may further improve our results. Also, for the DOA task, it is possible to use Denoising Autoencoders to separate overlapping sound events and then use a simpler model for DOA estimation for each separated sound event. Finally, our approach should be also tested in datasets with no upper limit in the number of overlapping sound events.

Bibliography

[1] http://dcase.community/challenge2019/task-sound-event-localization-and-detection.

[2] S. P. Chytas and G. Potamianos, "Hierarchical detection of sound events and their localization using convolutional neural networks with adaptive thresholds," DCASE2019 Challenge, Tech. Rep., 2019.

[3] N. C. Phuong and T. D. Dat, "Sound classification for event detection: Application into medical telemonitoring," in *Proc. International Conference on Computing, Management and Telecommunications (ComManTel)*, 2013, pp. 330–333.

[4] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proc. IEEE International Conference on Multimedia and Expo*, 2005, pp. 1306–1309.

[5] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, 2013.

[6] A. Temko and C. Nadeu, "Acoustic event detection in meeting-room environments," *Pattern Recognition Letters*, vol. 30, pp. 1281–1288, 2009.

[7] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 20–31, 2015.

[8] P. Giannoulis, G. Potamianos, P. Maragos, and A. Katsamanis, ""Improved Dictionary Selection and Detection Schemes in Sparse-CNMF-Based Overlapping Acoustic Event Detection"," DCASE2016 Challenge, Tech. Rep., 2016.

[9] P. Giannoulis, G. Potamianos, and P. Maragos, "Multi-channel non-negative matrix factorization for overlapped acoustic event detection," in *Proc. 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 857–861.

[10] Y. Chen, Y. Zhang, and Z. Duan, "DCASE2017 sound event detection using convolutional neural network," DCASE2017 Challenge, Tech. Rep., 2017.

[11] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network," DCASE2017 Challenge, Tech. Rep., 2017.

[12] C.-H. Wang, J.-K. You, and Y.-W. Liu, "Sound event detection from real-life audio by training a long short-term memory network with mono and stereo features," DCASE2017 Challenge, Tech. Rep., 2017.

[13] R. Lu and Z. Duan, "Bidirectional GRU for sound event detection," DCASE2017 Challenge, Tech. Rep., 2017.

[14] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio.

[15] http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-synthetic-audio# audio-dataset.

[16] http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio.

[17] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Comput. Surv.*, vol. 48, pp. 52:1–52:46, 2016.

[18] C. J. Grobler, C. P. Kruger, B. J. Silva, and G. P. Hancke, "Sound based localization and identification in industrial environments," in *Proc. 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2017, pp. 6119–6124.

[19] B. Mungamuru and P. Aarabi, "Enhanced sound localization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1526–1540, 2004.

[20] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[21] X. Zhang and D. Wang, "Deep learning based binaural speech separation in reverberant environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1075–1084, 2017.

[22] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 405–409.

[23] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[24] https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/.

[25] http://deeplizard.com/learn/video/FK77zZxaBoI.

[26] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*, J. F. Kolen and S. C. Kremer, Eds. IEEE, 2003, pp. 237–243.

[27] M. Stinchcombe and H. White, "Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions," in *Proc. International 1989 Joint Conference on Neural Networks*, vol. 1, 1989, pp. 613–617.

[28] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. MIT Press, 1998, pp. 255–258.

[29] M. Peng, C. Wang, T. Chen, and G. Liu, "NIRFaceNet: A Convolutional Neural Network for Near-Infrared Face Identification," *Information*, vol. 7, no. 4, 2016. [Online]. Available: https://www.mdpi.com/2078-2489/7/4/61

[30] "Convolutional Neural Networks for Text Classification," http://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets/.

[31] http://dcase.community/.

[32] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," in *Proc. Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, (Submitted to). [Online]. Available: https://arxiv.org/abs/1905.08546

[33] https://mhacoustics.com/products#eigenmike1.

[34] https://mhacoustics.com/home.

[35] https://www.mathworks.com/help/matlab/ref/sph2cart.html.

[36] https://github.com/HarisIqbal88/PlotNeuralNet.

[37] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[38] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," in *Proc. 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 1462–1466.

[39] https://keras.io/.

[40] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2019.

[41] https://librosa.github.io/librosa/.

[42] https://github.com/sharathadavanne/seld-dcase2019.