JACOBS
UNIVERSITY

Bachelor Thesis

# Sound Event Detection and Classification in Real Life Audio Using an Echo State Network with Deep Autoencoder Preprocessing

Fanlin Wang

*Supervisor* Prof. Dr. Herbert Jaeger

January 25, 2019

# Abstract

This paper investigates the sound event detection (SED) problem, which seeks to classify different sound activities as well as detect their onset and offset time in a given recording. The TUT Sound Events 2017 dataset used in this paper is a collection of recordings of street acoustic scenes with overlapping sound events. We investigate the performance gains brought about by extracting high-level features from the input signal, employing a deep auto-encoder for preprocessing. Those features are then fed into the echo state network (ESN), a version of recurrent neural network (RNN). The training is performed on the 4-fold validation dataset given in the DCASE2017 challenge. Our model compares very favorably with the baseline system and the published results. It also outperforms the winning entry in some metrics. This work is an extension of the author's semester project, including now a tailored, unsupervised preprocessing stage.

# Contents

# 1 Introduction

Humans not only have the ability to hear, but also are prepared to make sense of what they hear. For example, we are able to distinguish one voice from another, identify sounds with certain events, and separate speech from background noises. Ever since the introduction of the 'Cocktail Party Problem', which noted the exceptional human ability of perceiving speech in noisy settings, researchers from psychological fields have been trying to understand how the mixtures of sounds are analyzed by our ears [BC94]. An emerging field, called computational auditory scene analysis (CASA), concerns modeling the human auditory system and has been extensively researched in recent years. Two directions under this category are acoustic scene classification (ASC) and sound events detection (SED) [Mes+18; Gia+13], which have been recently promoted by many public challenges. An example is a series of IEEE AASP challenges, called Detection and Classification of Acoustic Scenes and Events (DCASE), which took place in 2013, 2016, 2017, and 2018.

The rest of the section is structured as follows: we will first explain the tasks of ASC and SED and outline their applications. Then a variety of features for audio signal processing is listed, followed by several feature extraction techniques in recent studies. We will then discuss a group of methods used in previous years and introduce various recent attempts.

Acoustic scene classification (ASC) aims to give the audio recording a general label without onset/offset time detection [Mes+18; Cak+17], whereas sound events detection (SED) addresses the problem of identifying every occurrence of a specific event in the recording [Mes+18; Gia+13]. Originally, SED deals with a simplified monophonic scenario, having non-overlapping sound events throughout the audio stream; current research interests move on to a more complicated polyphonic version, which requires a detection of overlapping sound events [Mes+18; Cak+17]. By analyzing audio recordings into meaningful sound events, we hope to gain insights in processing those components separately, and to arrive at a solution for more difficult variations such as the cocktail party problem.

Typical applications that fall into this category include surveillance in living or urban environments [McL+15; Cak+15; Cak+17; Fog+15; SG00]. [Zig+09] proposed a healthcare monitoring system to detect a fall of elderly people based on abnormal sound

detection, and had an advantage over wearable devices in terms of convenience. When it comes to security control in public areas, like the one installed at the central train station of Milan [Val+07], a sound monitoring system should be able to detect very unusual events like gunshots or screams. Other applications are, for example, environmental context detection [Par+16; Har+05], automatic audio indexing [Par+16; Cak+15; Bil98] and scene recognition for mobile robots [Cak+15]. In addition, combining visual information, audio events also assist in multimedia event detection [Wan+16], a task to detect certain events from a large set of video clips.

Numerous features have been used in the tasks of ASC and SED. They could be roughly classified as follows:

1. Temporal features: features in the time domain. They describe simple physical properties of a sound. Example: zero crossing rate (ZCR) [Val+07]

2. Spectral features: frequency-based features. Example: Mel Frequency Cepstral Coefficients (MFCCs), spectral moments, spectral distribution, log mel band energies, periodicity descriptors, constant-Q transform, variable-Q transform [Val+07; Cak+15; Mes+18]

3. Perceptual features: loudness, sharpness [Val+07]

4. Energy features: short time energy (STE) [Val+07]

5. Correlation features: correlation roll-off, correlation decrease, correlation slope, modified correlation centroid, correlation kurtosis [Val+07]

Recently, non-negative matrix factorization (NMF) is studied extensively as a learning feature from the time-frequency representation [Bis+17] and is used in preprocessing steps to create a dictionary from audio streams [Cak+15; Par+16; Hei+13]. Spectrogram-based image features achieved better performance in terms of noise robustness [McL+15]. Given such a variety of features, it is not trivial to decide which feature suits the problem best, and it usually needs to be done case-by-case. For example, abnormal sound detection contains sound events where energy is highly concentrated in a very short time period. Features like loudness and STE would be too sensitive to be meaningful for such task [Val+07]. However, in other cases, feature selection cannot be easily decided by the characteristics of features. To produce better results using these features with certain approaches is a major motivation behind those public challenges, which has led to numerous empirical studies for comparing performance.

One of the established approaches is to use standard features such as MFCCs with Gaussian mixture models (GMMs) or Hidden Markov models (HMMs). This group

of methods is popular among research work from, roughly speaking, 1995 to 2005. The work in [Bil98] describes how to find parameters for GMMs/HMMs using EM algorithms. In [Cai+06], the author proposed a framework, which first models the sound using an HMM, then obtains a probability representation of the HMM through a Grammar Network, and finally finds key audio effects by the Viterbi algorithm. The author in [Zha+05] addressed the issue of scarcity of training datasets by introducing a semi-supervised HMM with Bayesian adaptation techniques which measure how usual each observation sequence was. A more comprehensive evaluation of HMM based models can be found in [Mes+10]. Some earlier methods also explored decision trees to classify sound events [Hoi+05].

With the increasing availability of public audio database and the heated discussions on machine learning techniques, neural network approaches have become very popular in the field of audio processing. Better results compared to GMMs/HMMs have been achieved, especially for the polyphonic SED [Cak+17]. In feedforward neural networks (FNNs), inputs are pre-processed by context windowing: feature vectors are extracted from adjacent time frames and then concatenated to form a single training instance [Cak+15; Par+16]. In [Pic15], the author evaluated the possibility of applying convolutional neural networks (CNNs) to SED. The CNN is trained on log mel spectrogram features. The author concluded that CNN could be a viable solution for this problem, while admitting a much longer training time. Classifiers like support vector machines (SVMs) and deep neural networks (DNNs) can make use of auditory image features in such tasks [McL+15]. Currently, state-of-art approaches include long short-term memory networks (LSTMs), a form of recurrent neural networks (RNNs). RNNs can directly store the time sequential information that is nature to audio streams [Cak+17; Par+16]. Furthermore, a LSTM network is intended for enabling a longer short-term memory by substituting neurons with LSTM blocks, each consisting a) a self-connected memory cell, b) input/output activation form, c) 3 gating neurons (input, forget, output) [Par+16]. Inspired by good performance from both CNNs and RNNs, a deep learning architecture called CRNN is proposed in [Cak+17]. The author suggested that it can elevate two restrictions of FNNs: CNNs help to learn filters to shift time and frequency to increase invariance, while RNNs have the advantage of removing time window restrictions.

In this report, Section 2 states the motivation of our research. Section 3.1 investigates different commonly used features and describes a new model for feature extraction. Section 4 details the experiment setup, the result of which is discussed in Section 5. Section 6 summarizes and suggests the future work.

# 2 Motivation of Research

In this section, we will first give a quick re-introduction to echo state network (ESN). Then we will explain why we believe that ESN might be well suited for such tasks. The primary objective of this research is to explore the sound events detection (SED) problem using ESN.

## 2.1 Echo State Network: Structure

Echo state network is an approach to train recurrent neural networks (RNNs), which are characterized by feedback loops. Training an RNN was a notably difficult task before the deep learning invention, as cyclic dependencies of RNNs could potentially lead to non-convergence of gradient descent training methods [Jae02; Luk+12]. In addition, the convergence rates are slow because of vanishing gradient: training methods such as BackPropagation Through Time (BPTT) use gradients to update the neurons, which become very small after a few time steps. ESN bypasses this problem in a way that instead of using all the connections in the network to train the data, only the output weights are trained and it becomes a simple linear regression task. Formally speaking, the update function for the network state $x$ is [Jae02; Luk+12]:

$$x(n) = \tanh(W^{in}u(n) + Wx(n-1)), \tag{2.1}$$

where $n$ is a discrete number that refers to the $n$-th time step. If we have $K$ input units, then $u(n)$ is a $K$-dimensional vector, i.e, $u(n) = (u_1(n), ..., u_K(n))^T$. For a reservoir with $N$ neurons, $W^{in} \in \mathbb{R}^{N\times K}$ is the weight matrix between input units and neurons, whereas $W \in \mathbb{R}^{N\times N}$ is the one between neurons. The output is computed as follows:

$$y(n) = W^{out}[u(n); x(n)], \tag{2.2}$$

where $y(n) \in \mathbb{R}^L$, $L$ is the number of output units, and $W^{out}$ is the weight matrix between neurons and output units. $[\cdot;\cdot]$ stands for a vertical vector concatenation. By driving the network with training data, we collect $u(n)$ and $x(n)$ into a matrix $X$, where $X[:, n] = [u(n); x(n)]$, and the teachers into $Y^{target}$. Then we can obtain $W^{out}$ by ridge regression:

$$W^{out} = Y^{target}X^T(XX^T + \gamma^2 I)^{-1}. \tag{2.3}$$

From this equation, we can observe that the learning of an optimal output weight matrix is done in a closed-form manner, which makes the training possible on a normal computer to have a large input of thousands time steps. In addition, theoretically the input matrix can be arbitrarily large: we can decompose $X$ into a number of matrices and sum up the products of smaller matrices.

## 2.2 Echo State Network: Implications

As discussed in the previous section, ESN is very efficient in terms of computation and memory usage, making it particularly suitable for solving an SED problem. In order to be useful, ESN must have the echo state property (ESP) [Ver09; Jae02], the basic idea of which says that the network should "asymptotically forget its initial state". Figure 2.1 showcases this property.

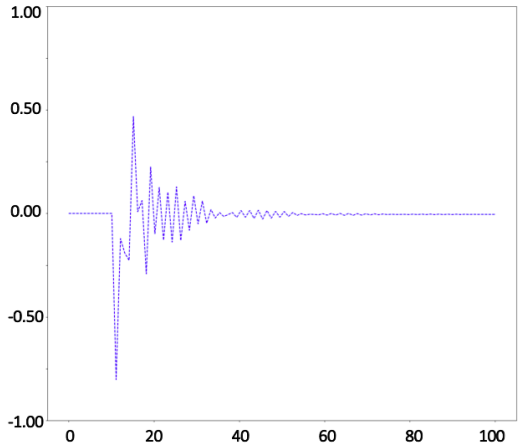



**Fig. 2.1:** A neuron's reaction to an impulse. The input signal remains 0 except at the time step 10, when it is set to 1. The neuron is activated and then dies out after several time steps.

Echo state network also has the short-term memory effect, that is to say, the network will be able to store a mixture of information from the current and the past input [Jae02; JH04; Ver09]. In [Jae02], the author proved that with an ESN of $N$ neurons, it is possible to preserve information in the same order of $N$ and therefore we expect ESN to have a good performance on an SED problem.

## 2.3 Research Objectives

Taking into consideration the motivation behind using ESN for an SED problem, our research objectives are:

1. to preprocess the audio streams and the annotated labels, and set up an ESN that works with these preprocssed data,

2. to experiment on feature extraction techniques and compare which features work better with the ESN.

In addition, since our project is based on a challenge that ended in 2017, we can compare our model with the published results. Therefore, not only will we evaluate our models in terms of design choices, we also hope to achieve a better result than the existing approaches.

# 3 Feature Extraction

As discussed in Section 1, feature selection is key to effective system performance and has attracted significant research interests. In this section, we will give a brief introduction to log mel-band energies and MFCCs. Then our proposed architecture for a convolutional autoencoder on log mel-band energies will be explained.

## 3.1 Log Mel-band Energy and MFCC

According to the published results from the website [Dca], two spectral features are used extensively in the task: log mel-band energies and MFCCs. While MFCC has been a standard feature in speech recognition, log mel-band energy is used primarily in an SED problem. The initial steps for extracting both features are similar: The audio signals are first split into a sequence of fixed time frames with some overlaps, and then a Fast Fourier transform (FFT) is applied to those time frames. After obtaining spectrums from FFT, we pass the spectrums through mel-filters and convert them into a logarithmic scale [Chu+09]. For log mel-band energies, each of the mel-bands is normalized to $[0, 1]$ to account for different recording conditions. For MFCCs, additional transformations such as discrete cosine transformation (DCT) and principal component analysis (PCA) are applied.

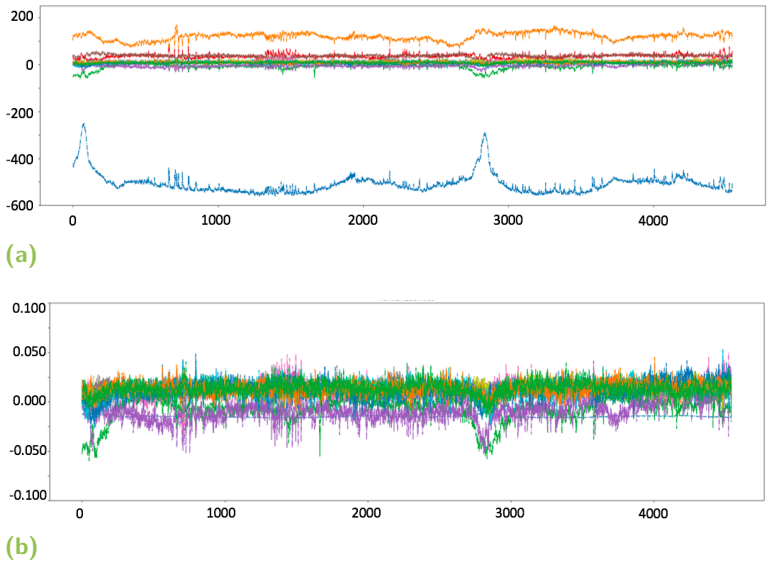



Fig. 3.1: a) the visualization of MFCC features; b) its normalized version.

Fig. 3.1.a shows the MFCC features extracted by librosa [Mcf+15]. It suggests that the original MFCC features have a wide range of value from roughly -600 to 200. Fig. 3.1.b illustrates the features after normalization, where the mean is 0 and the standard deviation is 1. Some studies suggest that MFCC does not work well with certain noise levels and types [Bha+16] and is more commonly used in speech recognition, where the frequency or energy level of the sound is more systematic.

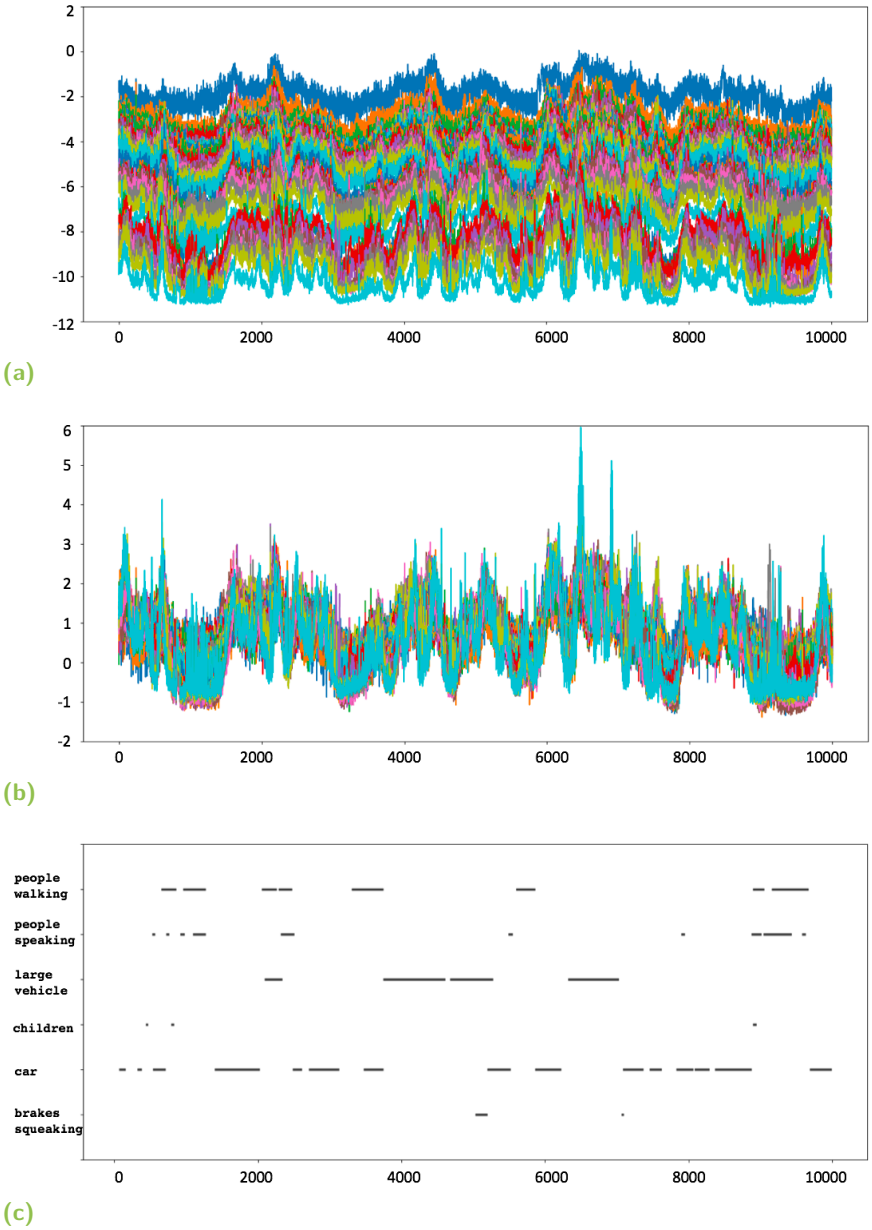



**Fig. 3.2:** a) the visualization of log-mel energies; b) its normalized version; c) labels. We noted that the dataset is not balanced in terms of occurrences of different events.

Another thing to consider is how to decide on the number of features. For MFCC features, typically 12 to 20 cepstral coefficients are chosen for speech analysis. There is a trade-off between accuracy and complexity in the model. [Val+07] proposed two

search algorithms to decide on feature number. The scalar approach is to start from the most discriminating feature and to add more features iteratively. It aims to maximize the objective function which concerns a class separability measure of a feature (e.g. Kullback-Leibler divergence, Fisher discrimination ratio), and the correlation coefficient between different features. A vectorial approach builds features iteratively. At each iteration, previously discarded features need to be reconsidered [Val+07]. However, such exhaustive searching would be too time-consuming to realize. Thus, we decide to follow the choice stated in [AV17], that is, we will use 13 cepsral coefficients for MFCCs and 40 bands for log mel-band energies. Figure 3.2 plots log-mel band energies along with labels.

## 3.2 Proposed Feature

Some existing approaches have explored a combination of features in hope of achieving a better characterization of audio signals, however, [Chu+09] pointed out that adding more features does not always produce better performance, as irrelevant features potentially have negative impacts. Therefore, our aim is to extract high-level features that would yield a more effective representation of audio signals. Towards this goal, we propose the use of autoencoder as an unsupervised learning and dimension reduction technique on the 40 log mel-band features extracted previously.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv1d_1 (Conv1D) | (None, 40, 32) | 128 |
| max_pooling1d_1 (MaxPooling1 | (None, 20, 32) | 0 |
| conv1d_2 (Conv1D) | (None, 20, 4) | 260 |
| max_pooling1d_2 (MaxPooling1 | (None, 5, 4) | 0 |
| flatten_1 (Flatten) | (None, 20) | 0 |
| reshape_1 (Reshape) | (None, 5, 4) | 0 |
| conv1d_3 (Conv1D) | (None, 5, 4) | 36 |
| up_sampling1d_1 (UpSampling1 | (None, 20, 4) | 0 |
| conv1d_4 (Conv1D) | (None, 20, 32) | 416 |
| up_sampling1d_2 (UpSampling1 | (None, 40, 32) | 0 |
| conv1d_5 (Conv1D) | (None, 40, 1) | 97 |

Total params: 937
Trainable params: 937
Non-trainable params: 0

Fig. 3.3: A summary of our model obtained from the Python Keras package.

Autoencoders are a category of neural networks where input is the same as output so that the training can be done in an unsupervised manner. An autoencoder is composed

of encoder layers which result in a reduced representation of input, and decoder layers that decode the resulting vector back to the original dimension. Depending on the degree of nonlinearity of the data, one can design a deep autoencoder by adding more hidden layers.

Our proposed method uses convolution layers in the autoencoder because a convolutional neural network can benefit the SED problem by introducing time and frequency invariance [Cak+17]. Since our input signal is a vector that preserves a spatial ordering, we choose to use a 1D CNN. In our model, we use 2 1D-convolutional layers for the encoding part which lead to decrease of dimension from 40 to 20. The first convolution layer uses a kernel of size 3 and a filter of size 32, i.e., the size of convolution window is 3 while the number of output filters in the convolution is 32. We use a rectifier as an activation function, and the output is padded with zeros to keep the dimension unchanged. The second convolution layer has a filter of size 4 and a convolutional window of size 2. A max-pooling layer is applied after each convolutional layer to reduce the dimension. After the encoder layers, the output is a vector of length 20. Symmetrically, we use 2 CNN layers for decoding, after which the resulting vector is reconstructed back to 40 dimensions. The neural network is implemented using the Python Keras [Cho15] package. Fig. 3.3 is the summary of the neural network architecture.

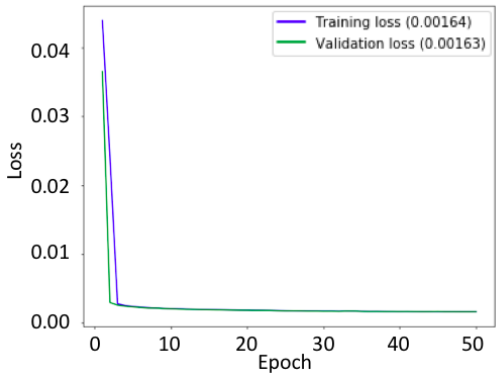



**Fig. 3.4:** The training and validation loss after 50 epochs.

Fig. 3.4 shows the loss of training after 50 epochs with the mean square error as loss function and an Adam [KB14] optimizer with a learning rate of 0.001. Both training and validation loss decrease to 0.0016 after 10 epochs, which suggests that the compressed features can be well reconstructed from the neural network. Fig. 3.5 is the visualization of our proposed features from a clip of an audio recording.

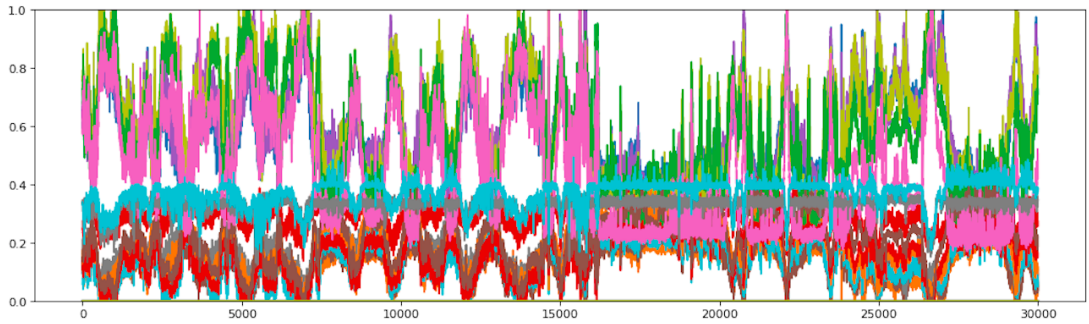
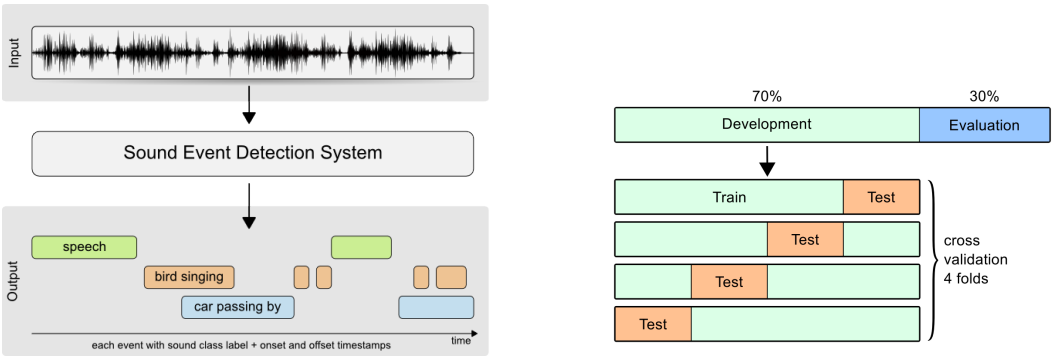



Fig. 3.5: Visualization of the feature obtained from our proposed model.

# 4 Experiment

We choose to conduct our experiment on the TUT Sound Events 2017 dataset. This publicly available dataset is used in the task 3 of DCASE2017 Challenge which is organized by IEEE Audio and Acoustic Signal Processing (AASP) community. A detailed description of this polyphonic SED task can be found in [Dca]. Figure 4.1 illustrates the task and the dataset.




(a) (b)

**Fig. 4.1:** An overview of the task and the dataset. The development dataset is provided with the ground truth and the cross-validation setup shows that each recording is used exactly once as test data. Source: [Dca; Mes+16b]

This dataset contains real-life recordings captured in different streets [Mes+16b]; therefore as shown in Figure 4.2, these recordings are very different in terms of amplitude range. Each recording is around 3-5 minutes long, using a 44.1 kHz sampling rate and 24 bit resolution. Annotation and quality checking process is described in [Mes+16b]. There are 6 sound classes selected for this task: 'brakes squeaking', 'car', 'children', 'large vehicle', 'people speaking' and 'people walking'. A 4-fold cross-validation setup is provided by the authors of the dataset, which helps to avoid splitting train and test set from using the exact same recording conditions.

## 4.1 ESN Setup

After the feature extraction, for training data in each 4-fold cross-validation set, we obtain an input size of about $170000 \times N$, where $N$ refers to the size of features. $N$ is 13, 40, and 20 for MFCC, log mel-band energy and our proposed higher-level feature respectively. For each time step of the input data, we extract the corresponding output

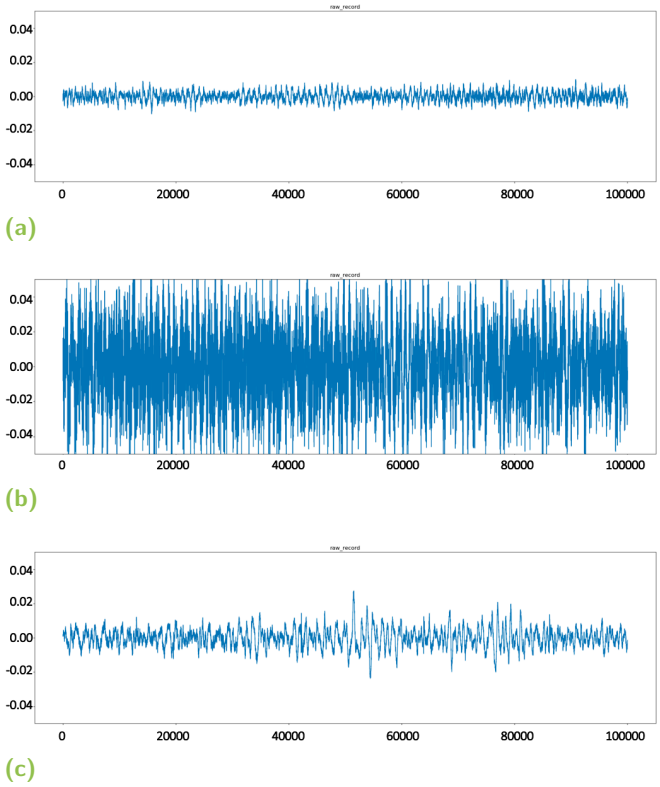



**Fig. 4.2:** Plots of some raw audio streams. The amplitude is normalized, with -1 corresponding to the minimum voltage, and 1 to the maximum voltage.

labels from the text file provided, which contains information on the onset/offset time of an event. The value is then converted to a vector of size 6 (because there are 6 classes of events in total), with each component indicating a corresponding class label. In other words, the training teacher matrix has a size of about $170000 \times 6$.

During the first run of creating the echo state network, we choose 500 internal neurons, and generate an input weight matrix $W^{in}$ of size $500 \times N$, and an internal weight matrix $W$ of size $500 \times 500$. The matrix $W$ is normalized such that its spectral radius equals 1. These weight matrices are reused in the future runs.

## 4.2 Evaluation Metrics

Evaluation metrics for sound event detection usually use segment-based error rate and F-score [Mes+16a]. A segment of one second length is used to compare the ground truth and the system output [Dca].

### 4.2.1 F-score

An event is considered to be correctly detected, or true positive (TP), if both the ground truth and the output indicate it as active. False positive (FP) refers to the case when the output suggests something inactive as active, while false negative (FN) is when the output indicates something active as inactive. Based on total counts of TP, FP, FN, the precision (P) is:

$$P = \frac{TP}{TP + FP} \tag{4.1}$$

whereas the recall (R) is:

$$R = \frac{TP}{TP + FN} \tag{4.2}$$

and the F-score (F1) is defined as:

$$F1 = \frac{2PR}{P + R} \tag{4.3}$$

### 4.2.2 Error Rate

The error rate is decided by the amount of errors in terms of insertions (I), deletions (D) and substitutions (S). A substitution is defined as the case when the system detects an event in a given segment, but gives it a wrong label. This is equivalent to the system output containing one FP and one FN. After counting the number of substitutions per segment, the remaining false positives in the system output are counted as insertions, and the remaining false negatives as deletions. For K segments, the error rate is calculated as follows:

$$ER = \frac{\sum_{k=1}^{K} I(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} S(k)}{\sum_{k=1}^{K} N(k)} \tag{4.4}$$

where $N(k)$ refers to the number of active labelled events in segment k.

## 4.3 Parameter Optimization

The scaling factors for $W^{in}$ and $W$ need to be chosen carefully in order to preserve the echo state property. The regularization term $\gamma$ in Equation 2.3 and the number of wash-out steps can also be tuned to obtain a better performance. The idea behind wash-out steps is based on the echo state property that we stated in Section 2.3. Wash-out refers to the observation that if we initialize a neuron with a different value, these value will converge after some time steps, which is demonstrated in Figure 4.3. Only the states

after these time steps will be collected. This is beneficial in reducing errors introduced by initial fluctuations.

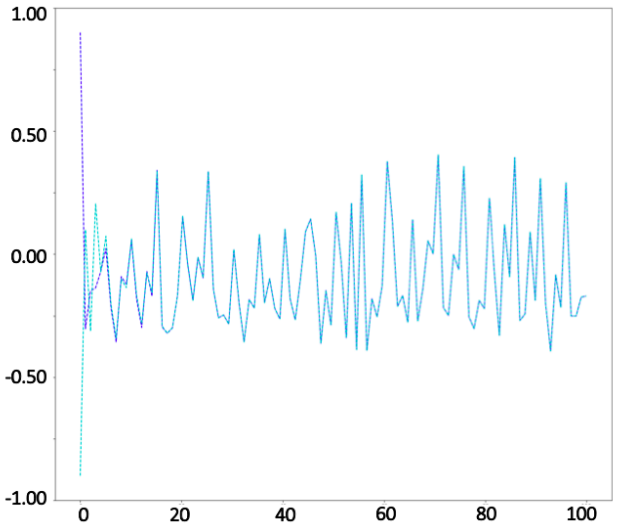



**Fig. 4.3:** The blue dashed line depicts the change of a neuron when its initial state is set to 0.9, while the cyan line shows that of the same neuron with a different initial value (-0.9). After about 10 time steps, the difference becomes so small that they are visually indistinguishable.

Since the value of each neuron in the output layer is between 0 and 1, we set a threshold value of 0.5 to decide whether a sound event is present. We use cross validation to find the optimal learning parameters that generalize well. Unfortunately, doing it automatically requires a lot of computational power. Therefore, to pick the parameters manually, we start with an educated guess and observe whether neurons get under-excited or over-excited (Fig.4.4). After manually running the cross-validation, checking the validation error in each iteration, we reach a set of parameters that gives us the best error rate.

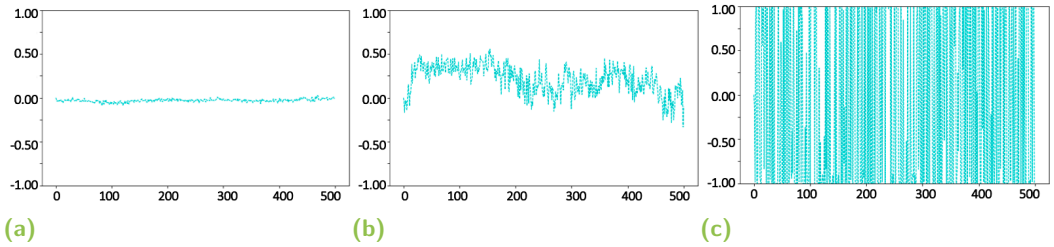



**Fig. 4.4:** Some neurons with a) under-excited, b) normal, c) over-excited states.

# 5 Result and Analysis

According to the task description, the TUT Sound Events 2017 dataset consists of the development dataset and the evaluation dataset. The development dataset is a 4-fold validation dataset that should be used in training, while the evaluation dataset (without labels), commonly understood as test dataset, was provided only one month before the challenge submission deadline. The outcome of our models is summarized as follows:

| | Development Dataset | | Evaluation Dataset | |
|---|---|---|---|---|
| | ER | F1 | ER | F1 |
| Baseline system | 0.69 | 56.7% | 0.94 | 42.80% |
| 1st ranked submission | 0.25 | 79.3% | 0.79 | 41.70% |
| 2nd ranked submission | 0.51 | 67.0% | 0.81 | 42.90% |
| 1st model:<br>log mel-band energies + ESN | 0.57 | 48.6% | 0.71 | 33.48% |
| 2nd model:<br>MFCCs + ESN | 0.78 | 37.6% | 0.83 | 31.22% |
| 3rd model:<br>convolutional autoencoder + ESN | 0.60 | 45.3% | 0.63 | 40.91% |

Our 3rd model which uses a convolutional autoencoder gives the best segment-based error rate on the test dataset, and it improves the second best error by 8%. In terms of the F1 score, it also improves the first model significantly. We also observe that our 3rd model has a relatively lower performance on the development dataset. The small difference between the error rate for the development dataset and the evaluation dataset suggests that our 3rd model achieves a better generalization. By contrast, the submission which ranked first seems to suffer from over-fitting issue since its performance on evaluation dataset is significantly worse than that on the development dataset. The 2nd model which uses MFCC as features is not better than the baseline system, which confirms our previous assumption that MFCC might work better with speech recognition than with an SED problem.

There is still room for further parameter tuning, and due to time limitation of our work, further improvements on some design choices are noted but not yet implemented. For example, from Fig.4.1 we recognized that it is an imbalanced dataset. This could lead to the situation where the system tends to be less likely to indicate an event as active if the event is a rare case.

In addition, the decision making method used in our approach is a simple thresholding. Other methods appeared in the submissions include median filtering, adaptive thresholding, top output probability, etc. There is also a novel approach on decision making among these submissions: in [Xia+17], the author used the same network structure as the baseline system, except for introducing a class-wise distance based approach in determining the output. It gets 4% reduction in terms of the error rate.

# 6 Conclusion

Sound events detection (SED) as a subfield of computational auditory scene analysis (CASA) is explored in this guided research project. The main objectives of this research are to achieve a basic understanding of the related work in the field of CASA, to set up an echo state network and to experiment on feature extraction techniques. We have successfully applied an unsupervised feature extraction method using deep learning techniques. By comparing our system with other published results, we can conclude that ESN can handle this problem well. The experimental result of our proposed feature shows promising performance which might be due to a better representation on the high dimensionality and nonlinearity of the dataset.

# Bibliography


[AV17] S. Adavanne and T. Virtanen. *A Report on Sound Event Detection with Different Binaural Features*. Tech. rep. DCASE2017 Challenge, 2017. URL: https://www.cs.tut.fi/sgn/arg/dcase2017/documents/challenge_technical_reports/DCASE2017_Adavanne_130.pdf.

[BC94] G. J. Brown and M. Cooke. „Computational auditory scene analysis“. *Computer Speech  Language* 8.4 (1994), pp. 297–336.

[Bha+16] U. Bhattacharjee, S. Gogoi, and R. Sharma. „A statistical analysis on the impact of noise on MFCC features for speech recognition“. *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. 2016, pp. 1–5.

[Bil98] Jeff Bilmes. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Tech. rep. 1998. URL: http://melodi.ee.washington.edu/people/bilmes/mypapers/em.pdf.

[Bis+17] V. Bisot, R. Serizel, S. Essid, and G. Richard. „Feature Learning With Matrix Factorization Applied to Acoustic Scene Classification“. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.6 (2017), pp. 1216–1229.

[Cai+06] R. Cai, L. Lu, A. Hanjalic, H. J. Zhang, and L. H. Cai. „A flexible framework for key audio effects detection and auditory context inference“. *IEEE Transactions on Audio, Speech, and Language Processing* 14.3 (2006), pp. 1026–1039.

[Cak+15] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen. „Polyphonic sound event detection using multi label deep neural networks“. *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015, pp. 1–7.

[Cak+17] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen. „Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection“. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.6 (2017), pp. 1291–1303.

[Cho15] F. Chollet. *Keras v1.1.2*. 2015. URL: https://github.com/fchollet/keras.

[Chu+09] S. Chu, S. Narayanan, and C. . J. Kuo. „Environmental Sound Recognition With Time–Frequency Audio Features“. *IEEE Transactions on Audio, Speech, and Language Processing* 17.6 (2009), pp. 1142–1158.

[Dca] *Detection and Classification of Acoustic Scenes and Events - An IEEE AASP Challenge*. 2017. URL: http://www.cs.tut.fi/sgn/arg/dcase2017/ (visited on Nov. 20, 2018).

[Fog+15] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento. „Reliable Detection of Audio Events in Highly Noisy Environments“. *Pattern Recogn. Lett.* 65.C (2015), pp. 22–28.

[Gia+13] D. Giannoulis, E. Benetos, D. Stowell, et al. „Detection and Classification of Acoustic Scenes and Events: An IEEE AASP Challenge“. *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 2013.

[Har+05] A. Harma, M. F. McKinney, and J. Skowronek. „Automatic surveillance of the acoustic activity in our living environment“. *2005 IEEE International Conference on Multimedia and Expo*. 2005, p. 4.

[Hei+13] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj. „Supervised model training for overlapping sound events based on unsupervised source separation“. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 8677–8681.

[Hoi+05] D. Hoiem, K. Yan, and R. Sukthankar. „SOLAR: sound object localization and retrieval in complex audio environments“. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 5. 2005, pp. 429–432.

[Jae02] H. Jaeger. *Short term memory in echo state networks*. Tech. rep. 2002. URL: `https://www.researchgate.net/publication/247514367_Short_Term_Memory_in_Echo_State_Networks`.

[JH04] H. Jaeger and H. Haas. „Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication“. *Science* 304.5667 (2004), pp. 78–80.

[KB14] Diederik P. Kingma and Jimmy Ba. „Adam: A Method for Stochastic Optimization“. *CoRR* abs/1412.6980 (2014).

[Luk+12] M. Lukoševičius, H. Jaeger, and B. Schrauwen. „Reservoir Computing Trends“. *KI - Künstliche Intelligenz* 26.4 (2012), pp. 365–371.

[Mcf+15] B. Mcfee, C. Raffel, D. Liang, et al. „librosa: Audio and Music Signal Analysis in Python“. *Proc. of the 14th Python in Science Conference*. 2015, pp. 1–7.

[McL+15] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao. „Robust Sound Event Classification Using Deep Neural Networks“. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.3 (2015), pp. 540–552.

[Mes+10] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen. „Acoustic event detection in real life recordings“. *2010 18th European Signal Processing Conference*. 2010, pp. 1267–1271.

[Mes+16a] A. Mesaros, T. Heittola, and T. Virtanen. „Metrics for Polyphonic Sound Event Detection“. *Applied Sciences* 6.6 (2016), p. 162.

[Mes+16b] A. Mesaros, T. Heittola, and T. Virtanen. „TUT Database for Acoustic Scene Classification and Sound Event Detection“. *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*. Budapest, Hungary, 2016.

[Mes+18] A. Mesaros, T. Heittola, E. Benetos, et al. „Detection and Classification of Acoustic Scenes and Events: Outcome of the DCASE 2016 Challenge“. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.2 (2018), pp. 379–393.

[Par+16] G. Parascandolo, H. Huttunen, and T. Virtanen. „Recurrent neural networks for polyphonic sound event detection in real life recordings“. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6440–6444.

[Pic15] K. J. Piczak. „Environmental sound classification with convolutional neural networks“. *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. 2015, pp. 1–6.

[SG00] C. Stauffer and W. E. L. Grimson. „Learning Patterns of Activity Using Real-Time Tracking“. *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8 (Aug. 2000), pp. 747–757.

[Val+07] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti. „Scream and gunshot detection and localization for audio-surveillance systems“. *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. 2007, pp. 21–26.

[Ver09] David Verstraeten. „Reservoir Computing: computation with dynamical systems“. eng. PhD thesis. Ghent University, 2009, pp. XXII, 178.

[Wan+16] Y. Wang, L. Neves, and F. Metze. „Audio-based multimedia event detection using deep recurrent neural networks“. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 2742–2746.

[Xia+17] X. Xia, R. Togneri, F. Sohel, and D. Huang. *Class Wise Distance Based Acoustic Event Detection*. Tech. rep. DCASE2017 Challenge, 2017. URL: https://www.cs.tut.fi/sgn/arg/dcase2017/documents/challenge_technical_reports/DCASE2017_Xia_136.pdf.

[Zha+05] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. „Semi-supervised adapted HMMs for unusual event detection“. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, pp. 611–618.

[Zig+09] Y. Zigel, D. Litvak, and I. Gannot. „A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound—Proof of Concept on Human Mimicking Doll Falls“. *IEEE Transactions on Biomedical Engineering* 56.12 (2009), pp. 2858–2867.