2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India

Minimally Supervised Sound Event Detection Using a Neural Network

Aditya Agarwal, Syed Munawwar Quadri, Savitha Murthy, Dinkar Sitaram
Center for Cloud Computing and Big Data
Computer Science Department
PES University
Bangalore - India

***Abstract*—This paper proposes a sound event detection system that is trained using a minimally annotated data set of single sounds to identify and separate components of polyphonic sounds. The system uses a Feed Forward Neural Network with a single hidden layer that is pre-trained using an autoencoder. Single sounds, represented as Mel Frequency Cepstral Coefficient (MFCC) feature vectors, are used to train the neural network using back propagation. Polyphonic sounds are preprocessed using Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF) to obtain source separated sounds. These source separated sounds are then tested for sound classification using the feed-forward algorithm. Our system is able to achieve reasonable accuracy of source separation and detection with minimal training set. The ultimate goal of our system is to bootstrap from minimal data and learn new sounds leading to a better sound detection system.**

I. Introduction

An auditory scene could have either a general context that consists of multiple/polyphonic sounds (basketball match or a busy street), or a characteristic context containing singular/monophonic sounds (such as siren sounds, footsteps or a car horn). In an auditory scene, when a recognizable event is present; this event is known as a sound event. Computational auditory scene analysis (CASA) is a field of research that studies the detection of sound events in auditory scenes.

In real life auditory scenes, sound events tend to occur as overlapping events. Despite this, humans easily are able to recognize the sound events present. For example, one can easily identify the sound of a car horn on a noisy street. For automatic sound event detection, modeling overlapping sound events require a great deal of human annotation to identify the number of sound events present at a particular instance. Furthermore, annotations could differ in terms of determining the start and end times of sound events. The automatic detection of overlapping sound events in an auditory scene still remains an unsolved problem. This is partially due to the sound event becoming more complex to detect as the number of overlapping sound events present at a particular instance increases [5].

The same context does not always have the same sound events. One match might have overlapping sound events of the crowd, cheerleaders and score board at one instance; while another match might just have the crowd and score board at an instance. Though contextual information is useful in predicting anticipated sound events [3], [4] (such as, in a football match, one can expect to hear the sound of a crowd cheering), it is not useful for predicting unexpected sound events [5]. The sparsity of examples - either due to the varied amount of sound events present at one instance or unexpected events occurring within the same context; makes modeling context problematic.

The primary aim of this paper is to explore accuracy in sound event detection for overlapping noises with minimal use of training data, contextual information and annotations. The paper proposes a system that does not use context to identify and separate sound events in an auditory scene. This is an unsupervised method for detecting sound events. As this is a work in progress, this paper presents preliminary results of the system. Since multiple overlapping sounds require a great deal of human annotation, the system is trained using an initial seed corpus of monophonic sounds. The system is then tested on polyphonic sounds, to give the output of individually identified and separated sound events. The ultimate end goal of this work in progress is to create a system that is trained on a small seed corpus, and can continuously add to this corpus from successfully classified test examples. The rest of the paper consists of *2. Background*, the *3. System Description*, *4. Results*, and *5. Conclusion*.

II. Background

Previous work in CASA are primarily concerned with modeling context in the system. for example, in [3] and [4], different audio contexts are recorded with the aim of detecting predetermined sound events. In [3], the co-occurrence of overlapping events are modeled using probabilistic latent semantic analysis (PLSA); which facilitates learning the relationship between individual sound events in a polyphonic sound. In [4], the two stage approach consists of a context recognition stage using a Gaussian mixture model (GMM) and a sound event detection stage using a 3 state HMM.

For many papers such as [3], [6] and [4], the input is a polyphonic sound in an unidentified context, and the output is one identified sound event at each instance of the polyphonic sound. This one sound event is determined to be the most prominent sound at that instance. Evaluation is considered to be correct if the sound event predicted at that instance is present in the collective sound events at that instance. However [4] also reports an alternative output of polyphonic or multiple sound events at a particular instant.

In [5], a system is proposed to model unusual events. This is because systems dependent on prior contextual information; such as [3] and [4], tend to mis-classify unusual or rare events. [5] takes a number of well established usual events and adds a set of unusual events in an unsupervised manner using Bayesian adaptation techniques to overcome the sparsity of data. However, this technique still requires large amounts of annotated data for usual events.

Our system tries to automatically identify and separate the number of sound events present in a polyphonic sound without any contextual information; and hence processes unannotated data. The objective from this system is different; as given an input of polyphonic sound, the sound events are separated using source separation and then identified as individual sounds. All sound events in the polyphonic sound are identified, but the polyphonic sound is decomposed into component sound events. For example, if given an input sound from a noisy street, our system tries to automatically detect the number of sound events in this sound (lets say, 3 sound events consisting of footsteps, speech and cars). Then the system decomposes this sound into 3 separate sounds through source separation, each depicting an individual sound event. The system then tries to classify each individual sound. The output is thus 3 source separated and individually classified sounds.

Work has been done previously on source separation for sound event detection. A common method of source separation of a polyphonic sound such as in [6] and [7] is using Non-negative matrix factorization (NMF). This method is mostly unsupervised, except for the initial input of estimating the number of sound events present in the multi source environment. This paper's system additionally uses Principal component analysis (PCA) to automatically estimate the number of sources present, then uses NMF to do source separation.

A typical way to classify sound events is using an HMM [3][4], with new work utilizing neural networks (NN) as a replacement [7]. In [7], the classification of isolated acoustic events with 61 distinct classes is done using a 5 layer NN pre-trained with RBMs, and an overall accuracy of 64.6% is achieved. The same data tested on a GMM+HMM system achieved 54.8% [7]. [2] uses a DNN with two hidden layers to learn MFCC features. Accuracies are reported in terms of contexts as well as sound events. This system utilizes a Feed-Forward NN due to the promising results of using NNs, combined with an NNs ability to learn new data (which is ideal for our bootstrapping approach). Moreover, the recent approaches based on neural networks have been quite successful motivating us to use a Feed-Forward NN for our proposed implementation.

III. System Description

The System description consists of Phase 1 - which gives the description of the current implemented system, and Phase 2 - which gives the plans for future implementation.

A. Phase 1

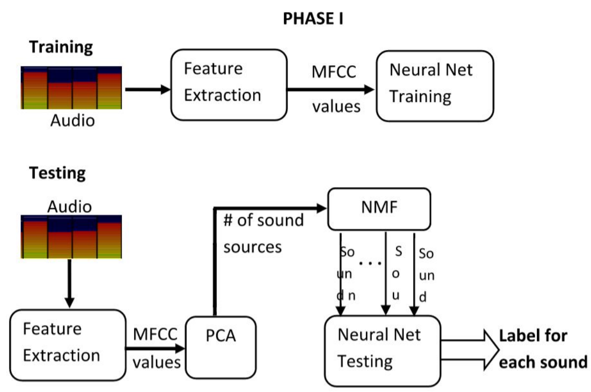

Fig. 1. Phase 1 architecture

Figure 1 shows the overall architecture for Phase 1.

1) The Dataset: Many papers such as [2], [6] and [7] have multiple classes of sound events for training data, some over 61 classes. Since annotated data is sparse, the system presented in this paper tries to utilize an initial seed corpus of 7 single sounds present in a publicly available data set. The objective is to add to this corpus as new sound events are correctly classified in the testing stage; a bootstrapping procedure. Since single sounds are used, no manual annotations of the sounds are required.

The sound classes in the publicly available Urban sound data set [11] comprises of single sounds, which we define as sounds without the presence of other overlapping sounds and with minimal silences. Of these, the number of minutes used for each class and the corresponding sound class is as follows: 30.58 minutes of air conditioner, 25.13 minutes of dog barking, 42.48 minutes of drilling, 7.32 minutes of gun shots, 28.01 minutes of jack hammer recording 37.76 minutes of recording of siren, and 36.32 minutes of engine idling. These single sounds were used as the initial seed corpus (Urb_data) to train the system.

Testing was done on 2 data sets: a synthetically created one (Synth_data) and a data set obtained from recording in a natural environment (Nat_data). In the Synth_data, pure sounds from the urban dataset were synthetically combined

TABLE I
SUMMARY OF TEST DATA SET

Dataset	Overlapping Sounds	Duration	Sources
Synth_data	Dog + Air Conditioner	5 min	2
Synth_data	Engine idling + Unknown sound1	5 min	2
Synth_data	Gun Shots + Drilling	5 min	2
Synth_data	Jackhammer + Siren	5 min	2
Nat_data	Siren + Unknown Sound2	5 min	2
Nat_data	Jackhammer + Unknown Sound3	5 min	2

to obtain a polyphonic sound. Each synthetic file has a maximum of 2 overlapping sound events present at an instance. The combination of sounds for the Synth_data is random and as follows: 5 minutes each of dog+air conditioner, gunshots+drilling, jackhammer+siren, and engine+an unknown sound not present in the Urb_data set. Hence about 20 minutes of synthetic sounds were produced; which is roughly 10% of the minutes of the training data used.

In Nat_data, a total of 10 minutes of naturally recorded noises were obtained from the internet. The objective for using Nat_data in testing was to observe whether our system could recognize a sound event present in Urb_data from a polyphonic sound. Therefore, in Nat_data, each polyphonic sound contains a sound that is present in Urb_data, combined with an unknown sound(s). It consists of approximately 5 minutes each of: a siren+unknown noise(s) and a jackhammer+unknown noise(s). The Nat_data has contextually occurring sounds, meaning the siren sound has city noises in the background, and the jackhammer sound has construction noises in the background. It is important to note that the Nat_data is a completely unseen data set, and though the Nat_data sounds correspond to some similar class labels of Urb_data; this does not guarantee any similarity of the sounds themselves. A summary of the test data can be seen in Table 1.

2) Feature Extraction: The first step in the processing of an audio file is to extract features, that is to identify the components of the audio signal that are good for identifying the acoustic content. We use openSMILE [14] to extract Mel-frequency Cepstral Coefficients from 25 milliseconds audio frames sampled at a rate of 10ms (Hamming window) with no frame overlap. It computes 13 MFCC (0-12) from 26 Mel-frequency bands, and applies a cepstral filtering filter with a weight parameter of 22. 13 delta and 13 acceleration coefficients are appended to the MFCC. The log-energy is appended to the MFCC 1-12 instead of the 0th MFCC. As though, they are more used in the context of speech recognition due to cepstral analysis, they have also achieved good accuracy in sound event detection [2], [3], [4] We thus obtain a 39 attribute feature vector for each time frame of 10ms. Hence for each frame, MFCC vectors are generated for each sound event class.

3) Training the Network: In this system, a Feed-Forward NN is used for sound event detection using labeled data. A neural network is composed of an input layer, multiple layers of hidden units and an output layer. In the proposed framework, we have implemented a neural network with one hidden layer, one input layer and 1 output layer essentially making it a 2 layered network. There are 39 neurons in the input layer (corresponding to 39 MFCCs) and 30 neurons in the hidden layer. The output layer consists of 7 neurons which classifies 7 different known classes, corresponding to 7 single sounds from Urb_data. The NN system is created using [13].

The energy of activation of the neurons in the hidden layer and the output layer are calculated utilizing the feed-forward algorithm that uses the $sigmoid$ function for activation. The energy of activations for the neurons in the output layer is calculated and the $argmax$ function is used to output the labels corresponding to the neuron with the highest energy of activation. Pre-training using auto-encoding is used to initialize the weights and bias of the network. The error associated with the output layer is calculated using the backpropagation algorithm which provides us with a way of computing the gradient of the cost function. In the implementation, the backpropagation algorithm is combined with the stochastic gradient descent algorithm for learning the weights and bias of the neural network.

The neural network was trained for 100 epochs with a mini batch size of 10, a learning rate of 3 and with an accuracy of average 70% (for correctly classifying single sounds from the Urban data set).

The NN parameters such as number of hidden units, mini batch size, learning rate and initial weight and bias etc. are selected by a grid search over the parameter values. The most successful results were obtained for the hyperparameters mentioned above.

4) Source estimation and separation: In a self learning environment, it is good to be able to estimate the number of sound sources computationally; which would make source separation completely unsupervised. A drawback of solely using NMF is that it still requires the number of sound sources to be specified explicitly. In an environment with multiple sounds, manually estimating the number of sound events is time consuming and often inaccurate. Hence the implemented system additionally uses PCA to automatically estimate the number of sources present on the test data, then uses NMF to do source separation.

Principal Component Analysis (PCA) helps us estimate the number of different sound components computationally. PCA is a dimensionality reduction technique that enables identification of the most important features. [8] have used PCA to determine discrete cluster indicators for the K-means clustering. We perform PCA [12] on the MFCC features on the test dataset (both Synth_data and Nat_data) to determine the number of sound events in each polyphonic sound. This estimate is then given to the NMF thus making source separation completely unsupervised.

The FASST toolbox [9] and [10] was used to implement the NMF algorithm. The NMF algorithm accepts the number of sound sources (in our case, estimated from PCA), or the rank of the NMF matrix as input. This results in the separation of the overlapping sounds into a number of individual sounds specified from the input. Essentially, the polyphonic sound is decomposed into composite single sounds.

5) Testing: This source estimation and separation procedure can be thought of as a fundamental pre-processing tool of the test data for our NN system; as the NN system currently is trained on/ better able to classify input in the form of single sounds. After obtaining these single sounds using PCA and NMF from the test data set , they are given to the NN for classification. The NN generates a label from among the set of labels corresponding to the output neuron with the highest energy of activation. Additionally generated is the $precision$ (number of instances correctly classified) and $recall$ (percentage of the correctly classified instances that are retrieved). With this a balanced $F-score$ or F_1 score is calculated (the harmonic mean of $precision$ and $recall$).

B. Phase 2

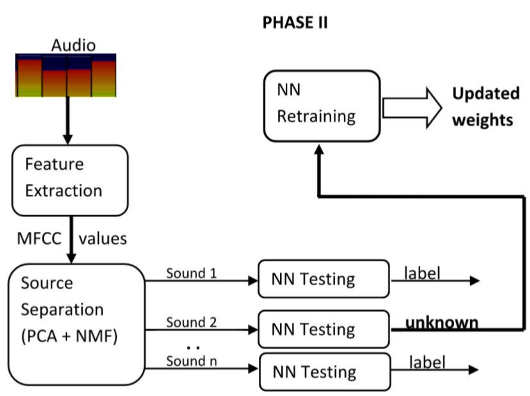

Fig. 2. Phase 2 architecture

Figure 2 shows the overall architecture for Phase 2. In the *4. Results* section of the paper, preliminary results are reported and discussed. A few ideas planned for future implementation are:

- *Obtain a larger amount of test data*, preferably for the Nat_synth set.
- *Thorough cleaning of the Urb_data dataset*: As far as possible the dataset has been cleaned and outliers removed; but there are certain sound events that are mislabeled (such as jackhammer for drill). This is not easily rectified as a few audio files for these two categories are similar to the extent that human annotators also cannot distinguish between the two. A more careful cleaning of Urb_data to train the NN could lead to improved accuracy
- *Further preprocessing of the dataset*: Urb_data contains a mixture of continuous sounds and sounds interspersed with silences. Since we have relatively noise free data, we can assume that there is a difference in the energy of a frame of sound compared to energy in a frame of silence. Using a method such as a Root mean square (RMS) energy calculation could help remove frames of silence (and low level noise) if the predetermined energy threshold is below a certain point.
- *Further classification of unknown sounds*: The ultimate objective of the system is to learn new sounds as the system is exposed to them. Currently, if a sound event is classified as one amongst the known classes and the F_1 score is below a certain threshold (currently 0.25), we categorize this as an unknown sound. If a sound is classified as unknown, then give it a new label and add it to Urb_data for retraining the NN. This would enable our system to learn new sounds. Human intervention maybe later on needed to give a meaningful label.

  Currently, 3 distinct unknown sounds have been obtained from testing; which human annotators would categorize as construction sounds, traffic/city sounds and noise (unidentifiable). Implementation is planned by utilizing a distance measure to determine the sub-categorization of the unknown sounds and a similarity measure to determine how close different samples of one sound event are to each other.

IV. Results

A. Evaluation Procedure

As the evaluation metric, F_1 score is calculated as the harmonic mean of precision and recall. We have used F_1 score as a measure to test the accuracy of the proposed framework. For each sound event, if the event is:

- Detected in the labeled output by the neural network and matches the annotation, then that event is regarded as correctly detected.
- Not detected in the labeled output by the neural network but is present in the annotation, then that event is regarded as missed.
- Detected in the labeled output by the neural network but is not present in the annotation, then that event is regarded as false alarm.

Each labeled output is thus classified as correct, missed, or a false alarm. The F_1 score for each sound event class is calculated as follows:

$$F_1 score = 2 * \frac{precision * recall}{precision + recall}$$

$$precision = \frac{correct}{correct + falsealarm}$$

$$recall = \frac{correct}{correct + missed}$$

The resultant F_1 score is calculated by taking the average F_1 scores of all the sound event classes.The percentage of the resultant average F_1 score is taken to get the accuracy; and hence results are reported in terms of accuracy.

For an unknown sound to be classified as such, if the F_1 score for the annotation is below a threshold of 0.25 for all 7 classes of sound events present in the NN, then the label is discarded and is classified as an unknown sound.

B. Results and Discussion

The model was tested against Synth_data; the synthetically created overlapping sounds of dog+air conditioner, gunshots+drilling, jackhammer+siren, and engine+an unknown sound not present in the Urb_data set. The number of sound sources was given as input to the NMF algorithm as the number of Principal Components specified by the PCA. As we can see from Table 2, PCA correctly estimated the number of sources in each overlapping sound. The individual separated sounds were then tested on the trained neural network and the resulting accuracy scores obtained are presented in Table 3.

TABLE II
ACTUAL SOURCES COMPARED TO PCA ESTIMATE

Overlapping sounds	Number of components	PCA estimated components
Dog + Air Conditioner	2	2
Engine idling + Unknown sound1	2	2
Gun Shots + Drilling	2	2
Jackhammer + Siren	2	2
Siren + Unknown Sound2	2	2
Jackhammer + Unknown Sound3	2	2

TABLE III
ACCURACY FOR SYNTH_DATA

Overlapping sounds	Sound	Accuracy
Dog + Air Conditioner	Dog	48.56%
Dog + Air Conditioner	Air conditioner	6.7%
Engine idling + Unknown sound1	Engine	11.4%
Gun Shots + Drilling	Gun shots	21.5%
Gun Shots + Drilling	Drilling	24%
Jackhammer + Siren	Jackhammer	15.4%
Jackhammer + Siren	Siren	66.8%
Engine idling + Unknown sound1	Unknown Sound1	85%

We presume that the poor accuracies obtained in the classification of drilling, jackhammer, air-conditioner and engine idling sounds can be partly attributed to the mislabeled training data set for training the NN. As mentioned before, some of these sound events are not only mislabeled, bit also difficult to manually distinguish. Another reason for overall bad accuracy could be due to the method of synthesizing the data; some critical information about the sound could have been lost.

Our model was also tested the Nat_data consisting of the following sounds: a siren+unknown noise(s) and a jackhammer+unknown noise(s). As we can see from the Table 2, PCA estimated for both these overlapping sounds is 2 components. This could be due to the presence of a predominant sound event (such as the siren) combined with an overlapping background noise. Our gold standard evaluation also estimates the number of components for each overlapping sound to be 2; as it was found to be hard to subcategories the overlapping background noise. The source separated sounds of the Nat_data were given to the NN for testing and the corresponding accuracy scores mentioned in Table 4. Any sound is classified as unknown if the F_1 scores obtained for the known sound classes are lower than a confidence score of 0.25, that is; an accuracy lower than 25%.

TABLE IV
ACCURACY FOR NAT_DATA

Overlapping Sounds	Sound	Accuracy
Siren + Unknown Sound2	Siren	66.2%
Jackhammer + Unknown Sound3	Jackhammer	43.6%
Siren + Unknown Sound2	Unknown Sound2	82%
Jackhammer + Unknown Sound3	Unknown Sound3	78%

As mentioned earlier, the objective for using Nat_data in testing was to observe whether our system could recognize a sound event present in Urb_data from a polyphonic sound, and successfully classify unknown sounds. As we can see in the Table 4, the accuracies of the source separated Siren and Jackhammer sound events are correspondingly 66.2% and 43.6%. This is an unseen dataset, and though the sound event has the same annotated class in our training set, this does not mean that the sound events are the same. We estimate that the distinguished noise of the siren helps in the recognition accuracy regardless of whether the data set is seen or unseen, and is reflected in the results. In the instance of the jackhammer, though the accuracy is still encouraging for a completely unseen data set; lower accuracy could be due to mislabeled training data. This also importantly gives correctly classified examples to grow two sound event classes in our current Urb_data. The 2 unknown noises were also classified with a high accuracy of 82% and 78%.

Work done by [2] has achieved an overall accuracy of 63.8% for detecting temporally overlapping sound events in realistic environments using a deep neural network. Our system has achieved a reasonable high accuracy in detecting the presence of unknown sounds as mentioned above.

Work done by [15] has achieved an overall accuracy of 24% in classifying isolated events into 61 classes using a network of hidden Markov models.

V. CONCLUSION

The following observations have been made:

- PCA was successfully implemented on our test data sets. Currently, only a maximum of 2 sound events per instance were used, further work will be to increase the number of sound event classes per instance.

- In a system trained on single sounds, the Nat_data overlapping sounds with known and unknown sound events were classified with reasonable accuracy provided preprocessing of the overlapping sounds using PCA and NMF is successfully done.
- Contextual information is useful to predict the overall context of a polyphonic sound. However, for our system, the Nat_data results indicate that contextual information is not essential when classifying single sound events of a polyphonic sound. Further testing will have to be done on an expanded Nat_data.
- The primary aim of this paper was to observe accuracy in sound event detection for overlapping noises with minimal use of training data, contextual information and annotations. With this objective in mind, currently interim results look promising.

REFERENCES

[1] Heittola, Toni, et al. "Sound event detection in multisource environments using source separation." Proc CHiME (2011): 36-40.

[2] Cakir, Emre, et al. "Polyphonic sound event detection using multi label deep neural networks." Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, 2015.

[3] Mesaros, Annamaria, Toni Heittola, and Anssi Klapuri. "Latent semantic analysis in sound event detection." In Signal Processing Conference, 2011 19th European, pp. 1307-1311. IEEE, 2011.

[4] Heittola, Toni, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. "Context-dependent sound event detection." EURASIP Journal on Audio, Speech, and Music Processing 2013, no. 1 (2013): 1-13.

[5] Zhang, Dong, Daniel Gatica-Perez, Samy Bengio, and Iain McCowan. "Semi-supervised adapted hmms for unusual event detection." In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 611-618. IEEE, 2005.

[6] Heittola, Toni, Annamaria Mesaros, Tuomas Virtanen, and Moncef Gabbouj. "Supervised model training for overlapping sound events based on unsupervised source separation." In ICASSP, pp. 8677-8681. 2013.

[7] Gencoglu, Oguzhan, Tuomas Virtanen, and Heikki Huttunen. "Recognition of acoustic events using deep neural networks." In Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European, pp. 506-510. IEEE, 2014.

[8] Ding, Chris, and Xiaofeng He. "K-means clustering via principal component analysis." In Proceedings of the twenty-first international conference on Machine learning, p. 29. ACM, 2004.

[9] Y. Salan, E. Vincent, N. Bertin, N. Souvira-Labastie, X. Jaureguiberry, D. T. Tran, and F. Bimbot, The Flexible Audio Source Separation Toolbox Version 2.0, in Show & Tell, IEEE International Conference on Acoustics, Speech and Signal Processing, 2014.

[10] Ozerov, Alexey, Emmanuel Vincent, and Frdric Bimbot. "A general flexible framework for the handling of prior information in audio source separation." Audio, Speech, and Language Processing, IEEE Transactions on 20, no. 4 (2012): 1118-1133.

[11] Salamon, Justin, Christopher Jacoby, and Juan Pablo Bello. "A dataset and taxonomy for urban sound research." In Proceedings of the ACM International Conference on Multimedia, pp. 1041-1044. ACM, 2014.

[12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[13] Team, The Theano Development, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas et al. "Theano: A Python framework for fast computation of mathematical expressions." arXiv preprint arXiv:1605.02688 (2016).

[14] Florian Eyben, Felix Weninger, Florian Gross, Bjrn Schuller: Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor, In Proc. ACM Multimedia (MM), Barcelona, Spain, ACM, ISBN 978-1-4503-2404-5, pp. 835-838, October 2013. doi:10.1145/2502081.2502224

[15] A. Mesaros, T. Heittola, A. Eronen and T. Virtanen, "Acoustic event detection in real life recordings," Signal Processing Conference, 2010 18th European, Aalborg, 2010, pp. 1267-1271.