Interspeech 2018
2-6 September 2018, Hyderabad

Monitoring Infant's Emotional Cry in Domestic Environments using the Capsule Network Architecture

M. A. Tuğtekin Turan and Engin Erzin

Multimedia, Vision and Graphics Laboratory,
College of Engineering, Koç University, Istanbul, Turkey

[mturan,eerzin]@ku.edu.tr

Abstract

Automated recognition of an infant's cry from audio can be considered as a preliminary step for the applications like remote baby monitoring. In this paper, we implemented a recently introduced deep learning topology called capsule network (CapsNet) for the cry recognition problem. A capsule in the CapsNet, which is defined as a new representation, is a group of neurons whose activity vector represents the probability that the entity exists. Active capsules at one level make predictions, via transformation matrices, for the parameters of higher-level capsules. When multiple predictions agree, a higher level capsule becomes active. We employed spectrogram representations from the short segments of an audio signal as an input of the CapsNet. For experimental evaluations, we apply the proposed method on INTERSPEECH 2018 computational paralinguistics challenge (ComParE), crying sub-challenge, which is a three-class classification task using an annotated database (CRIED). Provided audio samples contains recordings from 20 healthy infants and categorized into the three classes namely neutral, fussing and crying. We show that the multi-layer CapsNet is competitive with the baseline performance on the CRIED corpus and is considerably better than a conventional convolutional net.

Index Terms: ComParE, computational paralinguistic, baby cry detection, capsule network, emotion recognition

1. Introduction

The automatic audio classification has gained an important attention in recent years after the availability of big datasets. Typical applications range from the understanding of a scene or context surrounding [1], the recognition of urban sound environments [2] and the audio stream segmentation [3].

Detection or classification of the sound signals from acoustic sensors is a challenging problem because the boundaries between different classes could be fuzzy in nature. This implies the need for developing reliable and robust algorithms for classification of acoustic events. Such solutions can be regarded as the first step for an automatic recognition or labeling of the audio content.

A large collection of the signal processing and machine learning approaches have been applied to the problem, including matrix factorization [4], unsupervised feature learning [5], wavelet filterbanks [6] and deep neural networks as well [7].

Specifically, convolutional neural networks (CNN) are mainly preferred routine for this particular audio classification problem owing to the following reasons. Firstly, they can effectively capture the energy transition patterns when used with spectrogram-like inputs [8]. Secondly, their convolutional filters arranged with a small receptive field are capable of learning and discriminating spectro-temporal patterns of different audio classes even if the sound is convolved by other sources (noise). Conventional audio features such as mel-frequency cepstral coefficients (MFCC) or line spectral frequencies (LSF) are noticeably unsuccessful with respect to these two reasons defined above [9].

In this context, INTERSPEECH 2018 ComParE challenge introduces a novel problem, which is to classify the three mood-related infant vocalizations. This is an interesting research which allows automatic monitoring of babies not only for research purposes but also for clinical or home applications [10]. The corpus provided for this challenge comprises 5587 vocalizations of 20 healthy infants (10 females and 10 males) recorded within a study for the early detection of neurodevelopmental disorders [11].

The training samples are categorised into the three classes: (i) neutral/positive mood, (ii) fussing, and (iii) crying where the categorization process is performed by two experts in the field of early speech-language development with visual inspection on the basis of audio-video clips.

A baby cry can be considered as rhythmic transitions between aspiration and expiration after periodic air pulses coming from the vocal cord vibration. The period of these pulses is typically varied in healthy babies between 250-600 Hz [12]. This cry signal is shaped by the vocal tract whose first two formants occur ordinarily around 1100 Hz and 3300 Hz respectively [13]. In fact, the vocal tract of a new-born child is shorter (68 cm) and has a different structure compared with an adult. Therefore, it has higher fundamental frequency and resonances than adults. More details about the baby speech production, as well as speech models and properties, can be found in Fort et al. [14].

In the literature, detection of cry signals is commonly followed through extraction of features from recorded audio segments. These include pitch and formants or other spectral features such as short-time energy, MFCCs and others [15]. In the second stage, the signal is mainly classified using the traditional algorithms such as nearest neighbor or support vector machines (SVM) [16].

Deep neural networks, which have a high model capacity, are particularly dependent on the availability of large quantities of training data in order to learn a non-linear function from input to output that generalizes well and yields high classification accuracy on unseen data [17]. Hence, recent studies have explored the use of the CNNs tailored to baby cry detection.

In [18], the authors propose two learning algorithms for binary detection of baby cry in audio recordings (cry or no-cry). The first algorithm is a low-complexity logistic regression classifier, used as a reference. The second algorithm uses CNN, operating on log Mel-filter bank representation of the recordings. Performance evaluation of the algorithms is carried out using an annotated database containing several tens of hours recordings and their best configuration yielded 82.5% accuracy for the CNN classifier. In another study, similar network design is adapted to classify the crying into three categories including

10.21437/Interspeech.2018-2187

hungry, pain, and sleepy [19]. Their network achieves 78.5% validation accuracy collected from a balanced dataset.

The organizers of the challenge provide four baseline systems mainly composed of a set of features and a commonly used SVM classifier except for one system. All of their components can be reproduced via freely available, open source tools[1]. Their lowest performance system applies brute-forced segmental acoustic features extracted using the openSMILE tool [20] which achieves 57.5% unweighted average recall for test samples. Indeed, the tool gives a general purpose feature set that satisfies a wide range of paralinguistic problems. However, there is also a need for alternative representations achieving state-of-the-art results on many paralinguistic tasks.

CNNs have become the dominant approach to object recognition problem. They use translated replicas of learned feature detectors which allows them to translate knowledge about good weights acquired at one position in an image to other positions [21]. On the other hand, small groups of neurons called "capsules" make a very strong representational assumption: at each location in the image, there is at most one instance of the type of entity that a capsule represents [22]. Motivated this new approach, we apply an interesting alternative called capsule network (CapsNet) [23] to recognize baby cry spectrogram inputs. This network topology replaces the scalar-output feature detectors of CNNs with vector-output capsules and max-pooling with routing-by-agreement. As with CNNs, higher-level capsules in CapsNet cover larger regions of the image, but unlike max-pooling, it does not throw away information about the precise position of the entity within the region.

For the ComParE Crying Sub-Challenge, we paid particular attention to improve the classification performance according to the presented baselines under limited and unbalanced vocalizations. We implemented the capsule architecture which is designed to have both activation and pose components. In particular, we investigated a deep CapsNet architecture with localized (small) kernels for baby cry sound classification. This new structure is beneficial for the restricted number of samples because it nicely preserves the variations in the detected entity.

Rest of the paper is structured as follows. Section 2 gives a brief summary of the employed structure, then Section 3 presents the detailed methodology including pre-processing, feature extraction and network architecture. Experimental evaluations are then given in Section 4.

2. Capsule Networks

The concept of capsules was first introduced by Hinton et al. [22] as a method for learning robust unsupervised representation of images. Capsules are locally invariant groups of neurons that learn to recognize the presence of visual entities and encode their properties into vector outputs, with the vector length (limited to being between zero and one) representing the presence of the entity. For example, each capsule can learn to identify certain objects or object-parts in images. Within the framework of neural networks, several capsules can be grouped together to form a capsule-layer where each unit produces a vector output instead of a conventional scalar activation.

The output vector length of a capsule represents the probability that the entity represented by the capsule is present in the current input. Sabour et al. [23] use a non-linear function called "squashing" to ensure that short vectors get shrunk to almost zero length and long vectors get shrunk to a length slightly below 1,

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1+||\mathbf{s}_j||^2}\frac{\mathbf{s}_j}{||\mathbf{s}_j||} \qquad (1)$$

where $\mathbf{v}_j$ is the vector output of capsule j. In other words, the capsule j performs the non-linear squashing activation for the given input vector $\mathbf{s}_j$ and output vector $\mathbf{v}_j$. The orientation of vector $\mathbf{s}_j$ is preserved, but the length is squashed between 0 and 1. The parameters in $\mathbf{v}_j$ represent the various properties (like position, scale or texture) of a particular entity, and the length are used to represent the existence of the entity.

The input vector $\mathbf{s}_j$ is a weighted sum over all prediction vectors $\hat{\mathbf{u}}_{j|i}$ that is produced by multiplying the output $\mathbf{u}_i$ of a capsule in the layer below by a weight matrix $\mathbf{W}_{ji}$,

$$\mathbf{s}_j = \sum_i c_{ij}\,\hat{\mathbf{u}}_{j|i} \quad , \qquad \hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ji}\,\mathbf{u}_i \qquad (2)$$

where the c_{ij} are "coupling coefficients" that are determined by the iterative dynamic routing process.

The coupling coefficients between capsule i and all the capsules in the layer above sum to 1 and are determined by a "routing softmax" whose initial logits b_{ij} are the log prior probabilities that capsule i should be coupled to capsule j.

$$b_{ij} = b_{ij} + \hat{\mathbf{u}}_{j|i}\,\mathbf{v}_j \quad , \qquad c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \qquad (3)$$

The initial coupling coefficients are then iteratively refined by measuring the agreement between the current output $\mathbf{v}_j$ of each capsule j, in the layer above and the prediction $\hat{\mathbf{u}}_{j|i}$ made by capsule i using the scalar product (cosine similarity) $\mathbf{v}_j \cdot \hat{\mathbf{u}}_{j|i}$. In convolutional capsule layers each unit in a capsule is a convolutional unit. Therefore, each capsule will output a grid of vectors rather than a single vector output (see the original paper for the details of routing by agreement algorithm).

The coupling coefficients inherently decide how information flows between pairs of capsules. For a classification task involving K classes, the final layer of the CapsNet can be designed to have K capsules, each representing one class. Since the length of a vector output represents the presence of a visual entity, the length of each capsule in the final layer can then be viewed as the probability of the image belonging to a particular class k.

3. Methodology

In this study, we target to detect cry event classes from audio recordings. We utilize the CapsNet structure to learn time-frequency features of cry sounds. Our event detection system is based on a three-class classification model which is defined as a classification problem of events neutral/positive, fussing, and crying over a temporal window. The ground truth event label is taken as a single label for each audio sample. In other words, our event detection system will determine only one label for a given input test data regardless of their duration. We first pre-process the audio samples to enhance quality and to remove redundancy of the data that will be fed into network. We then apply feature extraction procedure based on the time-frequency analysis. Finally, we train the CapsNet architecture on the extracted feature representations. The overview of our methodology is illustrated in Figure1.

3.1. Pre-processing

In Section 1, we emphasized that the frequency content of an infant's cry is higher than an adult's. The whole content of CRIED corpus includes not only the cry vocalizations but also human speech or environmental noises recorded during data collection sessions. Although the duration of other sound types is much more smaller compared to the cry samples, it is required to clean the data to achieve more accurate performance. Thus, we first

[1] http://emotion-research.net/sigs/speech-sig/is18-compare

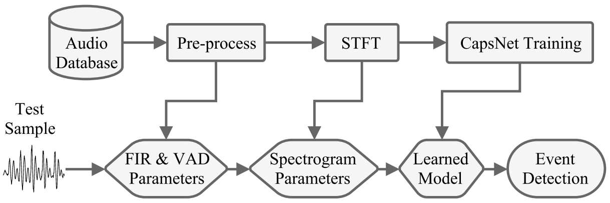

Figure 1: *Block diagram of the proposed classification scheme*

apply a high-pass FIR filter to remove the speech sounds and other low-frequency noise on the signal.

On the other hand, baby cry sounds don't have a fully continuous characteristics. Instead, impulse-like sequences recorded with different size of duration. Therefore, it is required to perform segmentation of all vocalizations before the feature extraction step. We then apply a voice activity detection (VAD) algorithm as a front-end processing of sound signals.

We implemented a very basic VAD algorithm which uses short-time features of audio frames and a decision strategy for determining sound/silence frames. The main idea is to vote on the results obtained from two discriminating features namely spectral flatness (SF) and short-term energy (STE). Energy is the most common feature for the VAD problem, however using only the STE is not enough for a robust detection. We therefore use the SF in addition to the STE which is calculated using the following equation,

$$SF_{\,dB} = 10\ \log_{10}(G_S/A_S) \tag{4}$$

where are G_S and A_S are geometric and arithmetic means of the audio spectrum respectively. For each incoming frame these two features are computed and the particular frame is marked as a cry sound, if both of the feature values fall over the predefined threshold. We fix up the threshold parameters based on the visual inspection of how the VAD performs discrimination efficiently.

3.2. Feature Extraction

Spectrogram representation visualizes time-frequency energy distribution on a two-dimensional graph. The signal energy at a particular time and frequency is represented by the color-map intensity in which higher amplitudes are represented by brighter reddish colors. Spectrograms are extracted from the input signal using the fast Fourier transform (FFT). In order to represent the temporal resolution, the signal is broken up into overlapping windows in the time-domain, and FFT transformed magnitude of the frequency spectrum for each window is calculated. This process generally corresponds to the squared log-magnitude calculation of the short-time Fourier transform (STFT) of the signal. Each cry signal is converted from waveform into spectrogram using Librosa library [24] with 256 FFT size. Within each temporal window, the spectrograms are computed over 15 msec frames (equal to 240 samples for 16 kHz sampling rate) with 50% overlap.

3.3. Network Architecture

Spectral capsule networks consist of spatial coincidence filters that detect entities based on the alignment of extracted features on a linear subspace. For this challenge, the proposed CapsNet architecture is shown in Figure 2. It has three layers where ”ConvReLU” has 128, 9 x 9 convolutional kernels with a stride of 2 and ReLU activation. This layer converts pixel intensities to the activities of local feature detectors that are then used as inputs to the ”primary” capsules.

The primary capsules are the lowest level of multi-dimensional entities and activating them corresponds to inverting the rendering process. This is a special type of computation than putting parts together to make familiar units, which is what capsules are designed to be good at.

The second layer, ”PrimaryCaps”, is a convolutional capsule layer with 32 channels of convolutional 8D capsules (in other words each primary capsule contains 8 convolutional units with a 9 x 9 filter and a stride of 2). Each primary capsule output sees the outputs of all 128 x 28 x 28 ConvReLU units whose receptive fields overlap with the location of the center of the capsule. In total, PrimaryCaps has $[32, 10, 10]$ capsule outputs (each output is an 8D vector) and each capsule in the $[6, 6]$ grid is sharing their weights with each other. Furthermore, PrimaryCaps can be regarded as a convolutional layer with Eq. (1). The final Layer, ”CryCaps”, has one 16D capsule per event class and each of these capsules receives input from all the capsules in the layer below. The length of the activity vector of each capsule in CryCaps layer indicates presence of an instance of each class and is used to calculate the classification loss. $\mathbf{W}_{ji}$ is a weight matrix between each $\mathbf{u}_i$, $i \in (1, 32 \text{ x } 10 \text{ x } 10)$ in PrimaryCaps and $\mathbf{v}_j$, $j \in (1, 3)$. The last CryCaps layer is connected with dropout to a 3 class softmax layer with cross entropy loss.

4. Results

4.1. Dataset

The provided corpus, CRIED, comprises 5587 audio recordings with alternating durations from 0.4 up to 41 seconds. Although all vocalizations were extracted from sequences of up to 5 minutes in duration, vegetative sounds such as breathing, smacking, hiccups, etc., were not segmented and included in the dataset. The summary of the corpus is given in the following table.

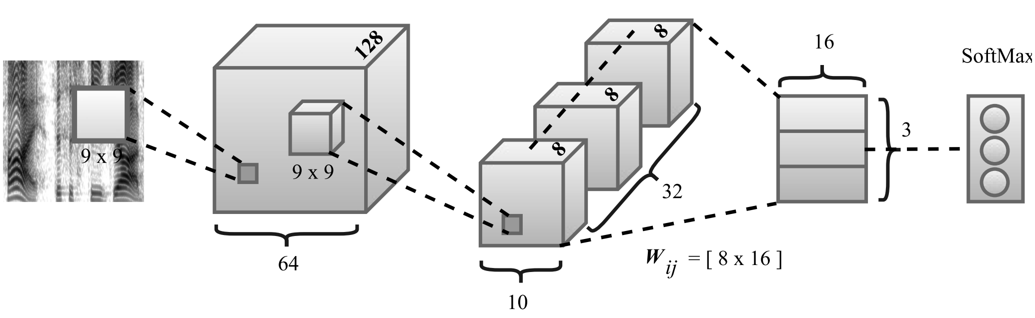

Figure 2: *Proposed CapsNet architecture with three layers*

Table 1: *Number of instances and durations per class*

	Instance	*Dur. (sec)*
Neutral/Positive	2292	3539
Fussing	368	960
Crying	178	812
Total	2838	5311

4.2. Baseline Experiments

Similarly to previous years, official baseline system proposed for this challenge employs the ComParE features set comprises 6373 features resulting from the computation of various functionals over low-level descriptor (LLD) contours. The features are computed with openSMILE toolbox [20]. The classifier used is a SVM implemented in WEKA [25]. Another feature set is obtained through unsupervised representation learning with recurrent sequence to sequence autoencoders, using the AUDEEP toolkit [26]. These feature vectors are concatenated to obtain the final feature vector for SVM classifier. A different baseline framework provides Bag-of-Audio-Words (BoAW) features computed using OPENXBOW [27]. Again SVM is used for classification of the BoAW descriptors. Last baseline approach uses a CNN to extract features from the raw time representation and then a subsequent recurrent network with Gated Recurrent Units (GRUs) performs the final classification. For this purposes the END2YOU toolkit was utilized [28].

4.3. Proposed System

We employed a stride of $1/128$ sec to obtain square spectrogram images with 256 FFT sizes. Training is performed on 128 x 128 normalized spectrograms that have been downsampled by scale 2 on each direction to achieve faster learning. No other data deformation is used. We train our proposed system with sequence mini batches of size 128. We also use the Adam optimizer with a small learning rate of 0.001. The network is trained for 50 epochs on a single NVidia GeForce Titan XP GPU with 12 GB onboard memory implemented using PyTorch[2]. All hidden layers use RELU activation functions, the output layer use softmax function, and the loss is calculated using cross-entropy function. Dropout and $L2$ regularization were also used to prevent extreme weights.

In order to compare the CapsNet performances, we also employed a standard CNN as a benchmark evaluation. The CNN is designed with three convolutional layers of 256, 256, 128 channels. Each has 5 x 5 kernels and stride of 1. The last convolutional layers is followed by two fully connected layers of size 328, 192. The last fully connected layer is connected with dropout to a 3-class softmax layer with cross entropy loss.

In the experimental evaluations, we utilized leave-one-subject-out (LOSO) cross-validation to get the subject independent evaluations. We use 64 x 64 spectrograms as an input for the network, our event detection system uses majority rule for instance based decision. As evaluation measure, unweighted average recall (UAR) is used mainly since the beginning of the first challenge held in 2009, because it is more adequate especially for unbalanced multi-class classifications than weighted average recall (accuracy).

Although, we augmented the audio data using a stride to the spectrograms, the neural network structure is very sensitive to the training dimensions for each input type. In Table 1, it can be observed that the duration of fussing or crying events are around four times less than the neutral/positive samples. In other words, spectrograms of the fussing and crying classes are formed by overlapping four times than the neutral/positive class which compensates the number of training spectrograms of each event.

Table 2: *Performances of the LOSO experiments*

	UAR [%]	Acc. [%]
Baseline: END2YOU	-	70.8
Baseline: OPENSMILE	75.6	82.6
Baseline: OPENXBOW	76.9	84.2
Baseline: AUDEEP	74.4	83.5
CNN	66.3	75.1
CapsNet	68.6	77.9
+ VAD	69.2	80.4
+ Equalization	**71.6**	**86.1**

Table 2 presents the results obtained by all these configurations with baseline performances. Results show that the CapsNet with LOSO protocol achieves 68.6% UAR and 77.9% accuracy. In order to improve network performance, we apply VAD and spectrogram equalization where both enhancements yield 71.6% UAR and 86.1% accuracy respectively. However, the CNN does not add any perceivable progress over the CapsNet systems which demonstrates the improvement of the new topology clearly.

Table 3: *Confusion matrix for the CapsNet LOSO experiment*

		Predicted		
		Neutral	Fussing	Crying
Actual	Neutral	**93%**	5%	2%
	Fussing	23%	**59%**	18%
	Crying	11%	27%	**62%**

Event class based evaluations are given as confusion matrices in Table 3 for CapsNet system. Although we observe consistent predictions in all event classes for the LOSO experiment, the most significant improvement appears in the neutral/positive event class. Although we implemented an spectrogram equalization for the classes that have fewer instances than neutral/positive, the unbalance problem still causes limited performance for both fussing and crying classes.

5. Conclusion

For the ComParE Crying Sub-Challenge, we implemented the capsule architecture, which is designed to have both activation and pose components with localized (small) kernels for baby cry sound classification. We applied pre-processing to filter out low frequency content as well as eliminating non-vocalized segments with a VAD. Furthermore, spectrograms of the minority classes were sampled more frequently to overcome the data unbalance problem during the training. Although we still observe effects of the data unbalance in our LOSO experiments, we got competitive and promising results with the proposed CapsNet system.

[2] http://pytorch.org/

6. References

[1] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[2] C. Mydlarz, J. Salamon, and J. P. Bello, "The implementation of low-cost urban acoustic monitoring devices," *Applied Acoustics*, vol. 117, pp. 207–218, 2017.

[3] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on speech and audio processing*, vol. 10, no. 7, pp. 504–516, 2002.

[4] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 151–155.

[5] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6445–6449.

[6] J. T. Geiger and K. Helwani, "Improving event detection for audio surveillance using gabor filterbank features," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 714–718.

[7] K. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[8] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 724–728.

[9] M. Dong, "Convolutional neural network achieves human-level accuracy in music genre classification," *arXiv preprint arXiv:1802.09697*, 2018.

[10] B. W. Schuller, S. Steidl, A. Batliner *et al.*, "The INTERSPEECH 2018 Computational Paralinguistics Challenge: Atypical & Self-Assessed Affect, Crying & Heart Beats," in *Proceedings of Interspeech*, 2018.

[11] P. Marschik, F. Pokorny *et al.*, "A novel way to measure and predict development: a heuristic approach to facilitate the early detection of neurodevelopmental disorders," *Current neurology and neuroscience reports*, vol. 17, no. 5, p. 43, 2017.

[12] L. L. LaGasse, A. R. Neal, and B. M. Lester, "Assessment of infant cry: acoustic cry analysis and parental perception," *Developmental Disabilities Research Reviews*, vol. 11, no. 1, pp. 83–93, 2005.

[13] A. Fort and C. Manfredi, "Acoustic analysis of newborn infant cry signals," *Medical Engineering and Physics*, vol. 20, no. 6, pp. 432–442, 1998.

[14] A. Fort, A. Ismaelli, C. Manfredi, and P. Bruscaglioni, "Parametric and non-parametric estimation of speech formants: application to infant cry," *Medical Engineering and Physics*, vol. 18, no. 8, pp. 677–691, 1996.

[15] R. Cohen and Y. Lavner, "Infant cry analysis and detection," in *Convention of Electrical & Electronics Engineers in Israel (IEEEI)*. IEEE, 2012, pp. 1–5.

[16] J. Saraswathy, M. Hariharan, S. Yaacob, and W. Khairunizam, "Automatic classification of infant cry: A review," in *International Conference on Biomedical Engineering (ICoBE)*. IEEE, 2012, pp. 543–548.

[17] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[18] Y. Lavner, R. Cohen, D. Ruinskiy, and H. IJzerman, "Baby cry detection in domestic environment using deep learning," in *International Conference on the Science of Electrical Engineering (ICSEE)*. IEEE, 2016, pp. 1–5.

[19] C.-Y. Chang and J.-J. Li, "Application of deep learning for recognizing infant cries," in *International Conference on Consumer Electronics (ICCE)*. IEEE, 2016, pp. 1–2.

[20] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *International Conference on Multimedia*. ACM, 2010, pp. 1459–1462.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NIPS)*, 2012, pp. 1097–1105.

[22] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 44–51.

[23] S. Sabour, N. Frosst, and G. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 3859–3869.

[24] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[26] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, "audeep: Unsupervised learning of representations from audio with deep recurrent neural networks," *arXiv preprint arXiv:1712.04382*, 2017.

[27] M. Schmitt and B. Schuller, "openxbowintroducing the passau open-source crossmodal bag-of-words toolkit," *Journal of Machine Learning Research*, vol. 18, no. 96, pp. 1–5, 2017.

[28] P. Tzirakis, S. Zafeiriou, and B. W. Schuller, "End2you–the imperial toolkit for multimodal profiling by end-to-end learning," *arXiv preprint arXiv:1802.01115*, 2018.