Audio Event Detection Using Wireless Sensor Networks Based on Deep Learning

Jose Marie Mendoza, Vanessa Tan(✉), Jr. Vivencio Fuentes, Gabriel Perez, and Nestor Michael Tiglao

Electrical and Electronics Engineering Institute,
University of the Philippines - Diliman, Quezon City, Philippines
{jose.marie.mendoza,vanessa.tan,
vivencio.fuentes,gabriel.perez,nestor}@eee.upd.edu.ph

Abstract. Wireless acoustic sensor network is useful for ambient assisted living applications. Its capability of incorporating an audio event detection and classification system helps its users, especially elderly, on their everyday needs. In this paper, we propose using convolutional neural networks (CNN) for classifying audio streams. In contrast to AAL systems using traditional machine learning, our solution is capable of learning and inferring activities in an end-to-end manner. To demonstrate the system, we developed a wireless sensor network composed of Raspberry Pi boards with microphones as nodes. The audio classification system results to an accuracy of *83.79%* using a parallel network for the Urban8k dataset, extracting constant-Q transform (CQT) features as system inputs. The overall system is scalabale and flexible in terms of the number of nodes, hence it is applicable on wide areas where assisted living applications are utilized.

Keywords: Audio event detection · Ambient assisted living

1 Introduction

Advancements in sensor node technologies allowed the development of low-cost and low-power multipurpose devices. These are usually integrated in a Wireless Sensor Network (WSN), where multiple sensor nodes communicate with each other to monitor the environment and gather data periodically. WSNs are already being utilized in different applications such as environment sensing, health monitoring, and smart homes [1].

The main goal of this paper is to create a wireless sensor network that could help in assisted living applications. In this work, audio streams are gathered by the sensor nodes then the sink node analyzes the data to detect different events in the environment. This paper has two main contributions: the exploration

J.-L. Chen et al. (Eds.): WiCON 2018, LNICST 264, pp. 105–115, 2019.
https://doi.org/10.1007/978-3-030-06158-6_11

of techniques using Convolutional Neural Networks (CNNs) for audio and the utilization of different audio input representations in audio event detection.

The rest of the paper is organized as follows. In Sect. 2, the related works are presented. Section 3 describes the WSN set-up while Sect. 4 briefly mentions the dataset used in the audio event detection. Section 5 presents the different CNN architectures that were implemented and a thorough discussion of the results in Sect. 6. Lastly, a summary of the findings and recommendations can be seen in Sect. 7.

2 Related Work

Ambient Assisted Living (AAL) is being used to monitor elderly in health care institutions [2]. A variety of sensors could be used such as mobile, wearable or static (e.g. pressure sensors, cameras, and microphones) to create a reliable system. For this application, microphones are more recommended since they are less invasive compared to wearable and cameras. Early stages of AAL using Wireless Acoustic Sensor Networks (WASN) implemented algorithms to lower the computational complexity of audio recognition by introducing a hybrid time-frequency approach [3]. While other methods increase efficiency of the system in terms of distribution and power consumption using real-time and scalable networks [4,5]. Detecting a few audio classes is one of the prominent limitations of these methods. Fortunately, homeSound, a distributed network where each node deployed reports to a GPU-enabled concentrator, is capable of classifying fourteen different indoor events [6]. It results to a more secured reporting mechanism to the sink since detection and classification is done locally before reporting to the server. This type of implementation is scalable by enabling indoor appliances to have specific alerts to report to a sink. This is possible in a smart home with devices sending alerts to an Android application, providing assistance whenever elderly people are left alone in the house [7].

Recently, classification using deep learning show exceptional results compared to traditional machine learning algorithms. The simplest architecture that can be constructed is the fully connected Deep Neural Networks (DNN). It is used in the Detection and Classification Acoustic Scenes and Events (DCASE) challenge. The DNN is used as baseline, using Mel-Frequency Cepstral Coefficients (MFCCs) for input features [8]. The problem of using MFCC is that it is more appropriate in speech processing applications, because it discards characteristics of environmental sounds. For this reason, features such as spectrogram and Constant-Q Transform (CQT) gave better performance in the DCASE challenge while using convolutional neural networks (CNN) [9,10,12]. CNN proved to be more effective than DNN on audio classification tasks, achieving approximately 10% improvement in classification using spectrogram inputs [11]. However, CQT input features produced an equal error rate of 16.6%, which is the best system proposed for DCASE Challenge 2016 on domestic audio tagging [10].

Pre-processing techniques introduce additional computation time to arrive to a prediction. Because of this, end-to-end networks using CNN extracts useful

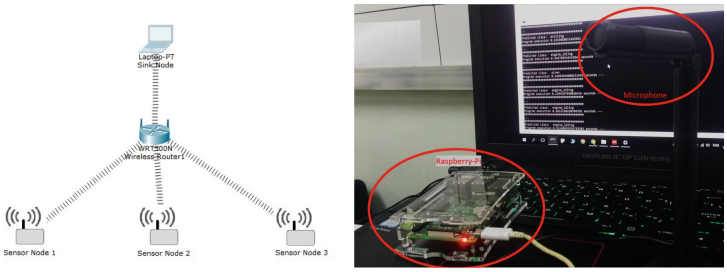
feature maps from input data on image classification architectures [12]. Similarly, this option is explored for environmental audio classification, where instead of 2-dimensional inputs, a time-domain windowed signal is used [13]. This paper discusses experiments using different input data to evaluate CNN architectures in audio classification.

3 WSN Raspberry-Pi Setup

A Wireless Sensor Network (WSN), composed of multiple nodes and a central base station, was implemented to facilitate the gathering and processing of audio signals over an enclosed area. These wireless acoustic sensor nodes would capture environmental audio signals, while a desktop computer would act as a base station which receives all signals captured. Data processing techniques would then be applied in the base station to detect specific audio events. A sample of the implemented network is illustrated in Fig. 1a showing three nodes and the main computer connected to the router.

Wireless sensor nodes used by the system are Raspberry Pi 3 devices, which are interfaced with a microphone using an integrated sound card. Raspberry Pi devices were chosen as the nodes for the network as it runs on a free open-source Linux operating system. These devices offer good computing power and are easily interfaced with audio components [14]. The devices record audio data via a microphone and is set to do so continuously. Recordings are done every 3 s, and audio is generated with a sampling rate of 44.1 kHz as Waveform Audio File Format. The audio data have meta-data, such as time stamps, in order to prepare the data for analysis in the server.

Transfer of the audio files are done via Secured Shell (SSH) through a WiFi network. Files are then retrieved by the server where classification using DNN takes place. As each audio file comes from different source nodes, their energy signatures would then be estimated to generate a consensus on which file would be used for the analysis of the signal at a given time [15]. The set-up of the system is shown in Fig. 1b, where a microphone is connected to the Raspberry-Pi and encodes the audio data received that is immediately sent to the computer.



(a) Diagram. **(b)** Setup using Raspberry- Pi.

Fig. 1. Wireless sensor network implementation

4 Dataset

The audio dataset used for training the classification model is the Urban Sound Dataset [16]. It contains 8,732 sound sources with 10 classes. The sounds included in the dataset are listed in Table 1. The audio data was split into 80% training set and 20% test set. All audio files were converted to mono, sampled at 44.1 kHz, and reduced to 3 s if the original data is too long and appended with silence if shorter than 3 s.

Table 1. Classes of the urban sound dataset

Class	Label	Class	Label
0	Air conditioner	5	Engine idling
1	Car horn	6	Gunshot
2	Children playing	7	Jackhammer
3	Dog bark	8	Siren
4	Drilling	9	Street music

5 Audio Classification

5.1 Sound Representations

Features are extracted using the Short-Time Fourier Transform (STFT) where each frame is 30 ms long. Overlap of each frames are 15 ms and extracts 1024 Fast Fourier Transform (FFT) points to compute for the spectrogram at the same frequency resolution.

Spectrogram Features are used as representation of a signal strength over time at different frequencies. It expresses the signal in its time-frequency representation, where the presence of frequencies are extracted per time frame.

Spectrogram Representative Images are used for visualization of the spectrogram which is common in audio processing, where time frames and frequencies are placed in the x-axis and y-axis respectively. To indicate the intensity of the frequency components, a color scheme is used in spectrogram images [18]. The lighter the color, the higher the amplitude of the intensity is present. This type of input turned out to be more effective than spectrogram features alone.

Constant-Q Transform is a time-frequency representation where the frequency bins are geometrically spaced. Its frequency resolution is better for low frequencies and the time resolution is better for high frequencies [10]. Different parameters are set in this feature, with a hop length of 512 samples with 12 bins per octave. Hamming window is used for short-time processing.

5.2 CNN Architectures

The following CNN architectures use 2-D convolutional layers for feature mapping. It is followed by fully connected layers which increases the capacity of the network during classification.

Sequential CNN is illustrated in Fig. 2. This implementation is similar with the sequential layers in [9]. Batch normalization and dropout layers are also included to explore their effects on the network. Moreover, dropout is implemented in the fully connected layers to provide regularization for training.

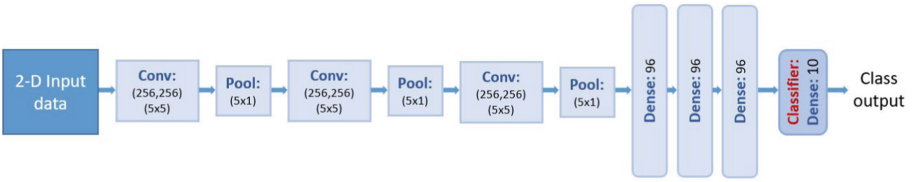


Fig. 2. A sequential CNN using a combination of 2D convolutional and fully-connected layers.

Parallel CNN is based on DeXpression architecture, a facial expression recognition system [19]. Some of the convolutional and pooling layers in the architecture are done in parallel. In this case, batch normalization replaced the local response normalization layer of the architecture. Dropout was also added at the fully-connected layers similar to sequential CNN.

CNN for End-to-End Classification is also explored, which uses raw audio as input by implementing a 1-D convolutional layer that acts similarly as a short-time pre-processing technique [13]. The network is shown in Fig. 3. It involves pooling layers for regularization and reshaping to allow the signals to be processed using 2-D feature mapping. In this case, the audio feature representation is extracted within the CNN architecture.

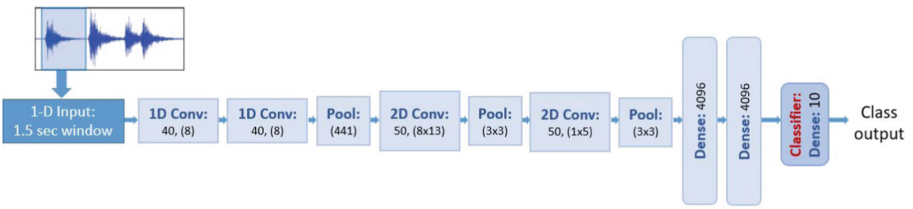


Fig. 3. An end-to-end CNN accepting a 1.5 s window for classification.

5.3 Experiments

A 5-fold cross validation was done on the training set to tune and evaluate the performance of the networks on different inputs. The batch size is set to 64, running for 30 epochs. A dropout of 0.5 on the fully-connected layers is also implemented. Rectified Linear Units (ReLU) are used for activation and the ADAM optimizer to boost training. All architectures are implemented in Keras using a NVIDIA GTX 960Ti GPU [20]. Training time took 2–3 h to finish for each architecture.

Classification. The 3-second audio recorded at the nodes are sent to the sink for audio event classification. The input features are then extracted and fed to the network to predict the audio event sensed by the nodes. The predicted class belongs to one of the classes in the Urban8k dataset. Moreover, the sink gives an output for each 1 s interval.

For the end-to-end network, the system accepts the 3-second data while the network extracts a 1.5-s window of raw audio from the initial input with an overlap of 200 ms. The class of the initial audio input is determined using probability voting [21]. Furthermore, the latency for providing output classes depends of the number of windows processed by the system, $n_{windows} \times delay_{network}$.

6 Results and Analysis

6.1 Classification Accuracy

For evaluation, the average accuracy of each class is calculated to get an overall accuracy of the network. Each CNN architecture will test the effects of batch normalization (BN) and dropout (DO) to the performance of the system.

Feature-Based: Sequential Convolutional Neural Network. It can be seen in Table 2 that the network with CQT input and with batch normalization yields the highest mean accuracy of 72.98%. This is the effect of accelerating the training that has regularization given that the model is trained for 30 epochs.

Table 2. Classification accuracies of the sequential CNN architecture on the Urban8K test set

Input	BN	DO	Mean Acc.
Spectrogram	✓		62.41%
Spec. Images	✓		57.91%
CQT	✓		72.98%
Spectrogram		✓	10.30%
Spec. Images		✓	34.83%
CQT		✓	21.62%

Feature-Based: Parallel Convolutional Neural Network. Different results were obtained using the parallel CNN. In Table 3, CQT input with dropout in training yields 86.17%, beating the previous network with batch normalization. It is also observed that spectrogram inputs are not effective when using dropout. This regularization technique does not work if inputs are not images causing it to have accuracies below 15%.

Table 3. Classification accuracies of the parallel CNN architecture on the Urban8K test set

Input	BN	DO	Mean Acc.
Spectrogram	✓		62.27%
Spec. Images	✓		77.36%
CQT	✓		73.32%
Spectrogram		✓	11.91%
Spec. Images		✓	82.59%
CQT		✓	83.79%

End-to-End: 1-D and 2-D Combination of Sequential CNN. The end-to-end network performance is shown in Table 4. It is noticeable that if the model is trained with dropout, it gives the best performance at 36.81% but not near the performance of systems with initial pre-processing. The reason for its low performance is the uneven distribution of data of Urban8k.

Table 4. Classification accuracies of the end-to-end CNN architecture on the Urban8K test set

Input	BN	DO	Mean Acc.
Raw audio	✓	✓	28.70%
Raw audio	✓		11.14%
Raw audio		✓	36.81%

6.2 Discussion

Among the three input representations, Constant-Q Transform is the most significant feature for urban sound classification. Because of its advantage of mimicking the human auditory system, this feature captures the low and mid-to-low frequencies better than spectrogram. This results to a classification system achieving the best performing accuracy.

As shown in Tables 2 and 3, the parallel CNN architecture is more stable and yields high accuracies for models with batch normalization and dropout. However, it fails the configuration where the model has dropout and uses spectrogram features as input. The regularization caused by dropout have been proven to work on images, but if it is a spectrogram which is time-frequency data important information may be omitted at times leading to unstable training. Similar experiments are also performed for the end-to-end network. The highest accuracy is observed with batch normalization shown in Table 4. Good performance with batch normalization is caused by the reduction of the dependencies of the gradients to the scale of the parameters when it is added.

A brief summary of the highest accuracies per architecture is shown in Table 5. The CQT-based parallel architecture achieves the highest accuracy while the end-to-end architecture has the lowest accuracy. The architectures are also compared to an implementation of a very deep CNN classifying the Urban8k dataset. For the end-to-end network, it is anticipated that it could be improved if CQT related features could be extracted from the raw audio signal.

Latency of each network is also observed to determine the practicality of the system. Compared to feature-based networks, the end-to-end approach have its pre-processing qualities within the network which lessens the run time of the system. Execution speed are seen in Table 5 where the parallel architecture performed slowest because of the wide structure of the network.

6.3 Overall Performance

Figure 4 shows the confusion matrix for the evaluation of the test set using the highest performing model which is the parallel CNN architecture with CQT as input and dropout for regularization. The class with the highest accuracy is *engine idling* (class 5) while the class with the lowest accuracy is *children playing* (class 2). The network finds difficulty in classifying *children playing* due to its wide difference of samples and relates it to *street music* because of the same set-up environment during data collection.

Table 5. Classification accuracies and execution time of the best architectures per implementation. Note that for the end-to-end network, multiple labels are obtained from 1 s of audio

Network	Mean Acc.	Time
Sequential	72.98%	0.67 s
Parallel	**83.79%**	0.75 s
End-to-end	36.81%	**0.56 s** (0.055 s/win)
Deep CNN [22]	71.80%	–

The combination of the deep learning systems and the wireless sensor network is illustrated in Fig. 1b. This scene depicts real-time processing, where the audio event is captured by the microphone and the predicted class label is displayed on the monitor.

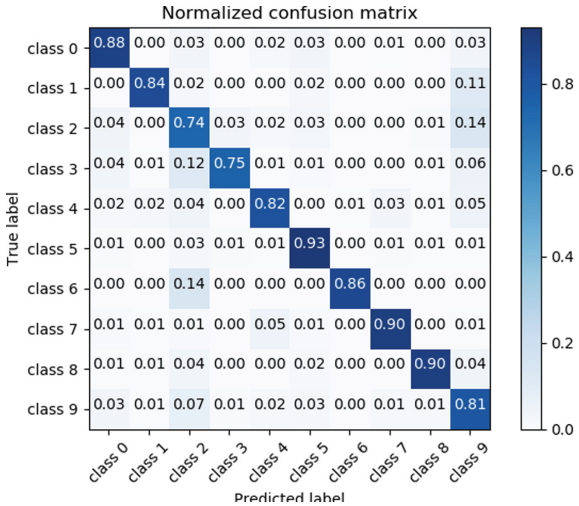


Fig. 4. Confusion matrix of parallel CNN architecture, best performing system, on the Urban8K dataset using CQT input features and dropout layers.

7 Conclusion and Recommendation

Deep learning techniques such as CNN used in wireless acoustic sensor networks for audio event detection proved that the system could be reliable. Results show that Constant-Q transform inputs are more appropriate to use in the system. This feature transform may also improve the end-to-end implementation if the features can be extracted within the network. The number of Raspberry-Pi nodes could also be increased to check for the reliability of the system if it is scalable. The continuous operation of the sensor nodes suggests that an energy efficiency algorithm could also be explored.

Acknowledgments. The authors would like to acknowledge the support of the University of the Philippines Diliman and the Department of Science and Technology through the Engineering Research and Development for Technology (ERDT) Consortium.

References

1. Ramson, S.R.J., Moni, D.J.: Applications of wireless sensor networks-a survey. In: 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), pp. 325–329. IEEE (2017)

2. Erden, F., Velipasalar, S., Alkar, A.Z., Cetin, A.E.: Sensors in assisted living: a survey of signal and image processing methods. IEEE Signal Process Mag. **33**(2), 36–44 (2016)
3. Martalò, M., Ferrari, G., Malavenda, C.: Wireless Sensor Networks and Audio Signal Recognition for Homeland Security. CRC Press (2012)
4. Dhawan, A., Balasubramanian, R., Vokkarane, V.: A framework for real-time monitoring of acoustic events using a wireless sensor network. In: 2011 IEEE International Conference on Technologies for Homeland Security (HST), pp. 254–261. IEEE (2011)
5. Sruthy, S., George, S.N.: WiFi enabled home security surveillance system using raspberry Pi and IoT module. In: 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), pp. 1–6. IEEE (2017)
6. Alsina-Pagès, R.M., Navarro, J., Alías, F., Hervás, M.: HomeSound: real-time audio event detection based on high performance computing for behaviour and surveillance remote monitoring. Sensors **17**(4), 854 (2017)
7. Nisar, K., Ibrahim, A.A.A., Wu, L., Adamov, A., Deen, M.J.: Smart home for elderly living using wireless sensor networks and an android application. In: 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–8. IEEE (2016)
8. Kong, Q., Sobieraj, I., Wang, W., Plumbley, M.: Deep neural network baseline For DCASE challenge 2016. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016) (2016)
9. Cakir, E., Heittola, T., Virtanen, T.: Domestic audio tagging with convolutional neural networks. In: IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2016) (2016)
10. Lidy, T., Schindler, A.: CQT-based convolutional neural networks for audio scene classification and domestic audio tagging. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016), DCASE2016 Challenge, vol. 90 (2016)
11. Hershey, S., Chaudhuri, S., Ellis, D.P.W., Gemmeke, J.F., et al.: CNN architectures for large-scale audio classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 131–135. IEEE (2017)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
13. Tokozume, Y., Harada, T.: Learning environmental sounds with end-to-end convolutional neural network. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2721–2725. IEEE (2017)
14. Vujović, V., Maksimović, M.: Raspberry Pi as a wireless sensor node: performances and constraints. In: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1013–1018 (2014)
15. Bahari, M.H., Plata-Chaves, J., Bertrand, A., Moonen, M.: Distributed labelling of audio sources in wireless acoustic sensor networks using consensus and matching. In: 2016 24th European Signal Processing Conference (EUSIPCO), pp. 2345–2349. IEEE (2016)
16. Salamon, J., Jacoby, C., Bello, J.P.: A dataset and taxonomy for urban sound research. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 1041–1044. ACM (2014)

17. Mysore, G., Smaragdis, P.: Relative pitch estimation of multiple instruments. In: IEEE International Conference on 2009 Acoustics, Speech and Signal Processing, ICASSP 2009, pp. 313–316. IEEE (2009)
18. Nanni, L., Costa, Y.M.G., Lucio, D.R., Silla, C.N. Jr., Brahnam, S.: Combining visual and acoustic features for audio classification tasks. In: Pattern Recognition Letters, pp. 49–56, vol. 88 (2017)
19. Burkert, P., Trier, F., Afzal, M.Z., Dengel, A., Liwicki, M.: DeXpression: deep convolutional neural network for expression recognition (2015). arXiv preprint arXiv:1509.05371
20. Chollet, F. et al.: KERAS. In: GitHub (2015). https://github.com/fchollet/keras
21. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2015)
22. Dai, W., Dai, C., Qu, S., Li, J., Das, S.: Very deep convolutional neural networks for raw waveforms. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 421–425. IEEE (2017)