Environmental Sound Classification Using Deep Learning

SHANTHAKUMAR S, SHAKILA S, SUNETH Pathirana, JAYALATH Ekanayake

(*Dept. of Computer Science and Informatics*, *Faculty of Applied Sciences*,
Uva Wellassa University, *Passara Rd. Badulla*, 90000, *Sri Lanka*)

Abstract: Perhaps hearing impairment individuals cannot identify the environmental sounds due to noise around them. However, very little research has been conducted in this domain. Hence, the aim of this study is to categorize sounds generated in the environment so that the impairment individuals can distinguish the sound categories. To that end first we define nine sound classes--air conditioner, car horn, children playing, dog bark, drilling, engine idling, jackhammer, siren, and street music-- typically exist in the environment. Then we record 100 sound samples from each category and extract features of each sound category using Mel-Frequency Cepstral Coefficients (MFCC). The training dataset is developed using this set of features together with the class variable; sound category. Sound classification is a complex task and hence, we use two Deep Learning techniques; Multi Layer Perceptron (MLP) and Convolution Neural Network (CNN) to train classification models. The models are tested using a separate test set and the performances of the models are evaluated using precision, recall and F1-score. The results show that the CNN model outperforms the MLP. However, the MLP also provided a decent accuracy in classifying unknown environmental sounds.

Key words: Mel-Frequency Cepstral Coefficients, MFCC, Multi-Layer Perceptron, MLP, Convolutional Neural Network, CNN

1 Introduction

In this paper we present classification of environmental sounds using deep learningtechniques, as it is useful for noise suppression in many audio-processing systems. Particularly, the applications of environmental sound classification are very useful for hearing impairment individuals.

The sounds describe the context of our daily activities, ranging from the conversations we have when we interact with people, the music we listen to and all the other environmental sounds we hear daily, like a passing car, the crackling of rain, or any other type of background noise. The automatic classification of environmental sounds is a field of research in full expansion with many real applications. Although there is a great deal of research in related audio fields such as speech and music, work on the classification of environmental sounds is relatively rare(Demir, et al., 2020,Chu, et al., 2009, Mun 2016,Uzkent et al.,2012, Kim et al. 2019). Similarly, by observing recent progress in the field of image classification where convolutional neural networks are used to classify images with great precision and on a large scale, it raises the question of the applicability of these techniques in other areas, such as sound classification, where discrete overtime sounds occur. As people move through the activities of daily living at home, at work,

and in social or business situations, basic auditory abilities take on functional significance. Audition makes it possible to detect and recognize meaningful environmental sounds, to identify the source and location of a sound.

The ability of an individual to carry out auditory tasks in the real world is influenced not only by his or her hearing abilities, but also by a multitude of situational factors, such as background noise, competing signals, room acoustics, and familiarity with the situation. Such factors are important regardless of whether one has a hearing loss, but the effects are magnified when hearing is impaired. For example, hearing loss is a primary barrier for driving vehicle this will lead to reduce the confidence to live their life without any disability of those individuals with hearing loss. Because hearing loss people can't identify the environmental sounds around them. Inability in identifying environmental sounds not only affects while driving, but also affects the day-to-day life of the individual with hearing loss.

Not only this, in some real-world applications like smart home setup should be capable of acquiring and analyzing the environmental sounds very effectively. This will add performance level advantage to the home- monitoring environment which is now a days stands for assisting the elder people.There are lots of systems for identifying human speech but there is shortage of approaches for identifying environmental sounds. Environmental sounds consist of various non-human sounds in normal day-to-day life. As a solution, we classify the environmental sounds using deep neural networks. For doing this we select a specific feature called Mel-Frequency Cepstral Coefficients (MFCC).

Most existing researches in this area used available standard datasets (Salamon et al., 2020, Koenig, M., 2020) whereas we create real datasets for each environmental sound class considered in this project.

In this project we train MLP and CNN neural network models for sound classification from the datasets that we created andthen compare the performance of the models. The models are implemented using python together with TensorFlow Libraries.

2 Related Works

Recently, there has been an incremental interest on Environmental Sound Classification (ESC), which is an important topic of the non-speech audio classification task (Demir, et al., 2020).

Recognizing environmental sounds is a basic audio signal processing problem. Consider, for example, applications in robotic navigation, assistive robotics, and other mobile device-based services, where context aware processing is often desired or required. Human beings utilize both vision and hearing to navigate and respond to their surroundings, a capability still quite limited in machine processing. Many of today's robotic applications are dominantly vision-based. When employed to understand unstructured environments (Pineau, et al., 2003).

Chu, et al., (2009) proposed an approach using Mel-frequency cepstral coefficients (MFCCs) integrated with matching pursuit (MP) algorithm to obtain effective time–frequency features and then applied Gaussian Mixture Model (GMM) to classify sounds. According to the results the system has shown to produce comparable performance as human listeners.

Demir, et al., (2020) developed a method to extract deep features in environmental sound using fully connected layers of Convolutional Neural Networks (CNN) model, which is trained in the end-to-end fashion with the spectrogram images. Finally, the sound classification was carried out using an SVM model recording the accuracies 94%, 81% and 78% on the predefined datasets; ESC 10, ESC 50 and UrbanSound8K, respectively.

Mun (2016) proposed a method to identify acoustic events using bottleneck features derived from a Deep Neural Network (DNN). The method was evaluated using a database of real life recordings of three sound categoriesand showed that the method outperforms the conventional methods and achieved the highest accuracy 90.8% on Office background

noise classification. This method is somewhat similar to our method. However, we define 9 sound classes and our method is more accurate than this method.

Uzkent et al. (2012) developed an approach to classify non-speech environmental sounds--gunshot, glass breaking, scream, dog barking, rain, engine, and restaurant noise--using Support Vector Machines (SVMs) and Radial Basis Function Neural Network classifiers. They used Mel Frequency Cepstrum Coefficients (MFCCs) together with pitch range (PR) to extract features from each of the sound classes and concluded that amalgamating the PR features into MFCC increased the classification accuracy by 4% to 35%.

Karol & Piczak (2015) trained CNN-based models to classify short audio clips of environmental sounds. They used three public datasets to evaluate the accuracy of the models and concluded that the convolutional network performs considerably better in recognizing sound classes--air conditioner, car horn, playing children, dog bark--whereas performing relatively poor for sounds drilling, engine idling, jackhammer. We also define these nine classes. However, we record sounds whereas Karol &Piczak usedthe UrbanSound8K dataset. Also, our approach is comparatively better than the above approach.

Addoli et al. (2019) developed an approach to classify audio sounds using CNN based models. Themodel was evaluated using UrbanSound8k dataset and the results showed that the models achieved 89% of mean accuracy.

Salamon et al. (2017) developed labeled dataset of different environmental sounds and trained models using CNN to classify the sounds. The outcome of this study showed that the CNN performs decently on the augmented datasets.

Wang et al (2014) presented a feature extraction method called nonuniform scale-frequency map for environmental sound classification in home automation. They trained SVM to predict 17 sound classes and achieved the prediction accuracy of 86.21%.

Most of the stated literature used standard datasets whereas we use our own dataset. We use Mel Frequency Cepstrum Coefficients (MFCCs) to extract features from sound clips as most of the studies in the literature.

3 Methodology

Data Collection

One common thing of previous researches of the environmental sound classification area is they used universal datasets. But in our case, we created our own dataset by collecting environmental sound from nine classes as shown in Table 1.

Table 1 Sound Classes

Class ID	Class Name
0	Air-Conditioner
1	Car-Horn
2	Children-Playing
3	Dog-Bark
4	Drilling
5	Engine-Idling
6	Jackhammer
7	Siren
8	Street-Music

Totally, we recorded nine hundred sound clips where 100 sounds recorded from each class. The sounds are recordedfrom the YouTube and the sound cloud. The dataset comprises two folders: audio and meta-data folders. The audio folder further divided into nine folders each representing a sound class containing around 100 files in WAV format. We used Matplotlib's spectrum to plot the spectrum of sound files. The meta-data of the sound files are recorded in a CSV file.

The meta-data of sound files indicate the class labels--car horn, dog bark, engine idling, jackhammer, air conditioner, street music, children playing, drilling, and siren-- andMFCC features of each audio file of our dataset.

Data Pre-processing

In the process of data pre-processing first, we

manually trimmed all the audio files with time unique duration. As our data is audio data, some audio properties require preprocessing to ensure their consistency across the dataset. Those audio properties are audio channels, bit depth and sample rate.We used the librosa python package for pre-processing of our dataset. Because, mainly, librosa's load function is automatically doing the pre-processing of above audio properties. Like librosa's load function by default converts the sampling rate of audio files to 22.05 KHz which we used as our comparison level and also librosa's load function removes the complication of the dataset having a wide range of bit-depths. Librosa also converts the audio signal to mono so the number of audio channels of the dataset audio channel always is one.

Feature Extraction

We extracted Mel-Frequency Cepstral Coefficients (MFCC) from the audio samples. The MFCC summarizes the frequency distribution across the window size, so it allowed us to analyses both the frequency and time characteristics of sounds. These audio representations will allow us to identify features for classification. We use Librosa's mfcc() function, which generates an MFCC from time series audio data. Then, we extracted an MFCC for each audio file from our dataset and store it in a Panda Data frame along with its classification label. We also generated a .csv file which includes the extracted MFCC of each audio files.

Next the categorical text was converted into model-understandablenumerical data using sklearn. preprocessing. LabelEncoder. Subsequently, we used sklearn. model_selection.train_test_split to split the dataset into training and testing sets. The size of the test set is 20% of the dataset and test sample selection is set to random.

We constructed the Multilayer Perceptron (MLP) Neural Network model using Keras and a TensorFlow backend. We used a sequential model to build the model layer by layer. We created a model with three layers, an input layer, a hidden layer and an output layer. All three layers are the dense layer type. The first layer will receive the input shape. As each sample contains 40 MFCCs. So, we have a shape of (1x40). The first two layers have 256 nodes.

We used the activation function for our first 2 layers were the ReLU (Rectified Linear Activation). We also applied a Dropout value of 50% on our first two layers. This randomly excluded nodes from each update cycle which in turn results in a network that is capable of better generalization and is less likely to overfit the training data. Our output layer has 9 nodes (num_labels) which matches the number of possible classifications. The activation is for our output layer is softmax. The softmax makes the output sum up to 1 so that the output can be interpreted as probabilities. The model makes its prediction based on which option has the highest probability.

Next, we constructed the Convolutional Neural Network (CNN) Model using Keras and a TensorFlow backend. The model is defined as a Sequential Keras model, for simplicity. We created a model with reshape layer because we had a sparse matrix as input.so it made dimension of shape in conv1D. We defined the model as having two 1D CNN layers, followed by a pooling layer. We used a standard configuration of 64 parallel feature maps and a kernel size of 3. The feature maps are the number of times the input is processed or interpreted, whereas the kernel size is the number of input time steps considered as the input sequence is read or processed onto the feature maps.

Then we defined the model as having two 1D CNN layers, followed by a GlobalAveragePooling1D then dropout layer for regularization, then a pooling layer. We used a standard configuration of 128 parallel feature maps and a kernel size of 3.We used the activation function for conv1D layers were the ReLU (Rectified Linear Activation). We also applied a dropout value of 50% on our first two layers. This randomly excluded nodes from each update cycle which in turn results in a network that is capable of better generalization and is less likely to overfit the training data. Our output layer has 9 nodes (num_labels) which

matches the number of possible classifications. The activation is for our output layer is softmax. Softmax makes the output sum up to 1 so that the output can be interpreted as probabilities. The model makes its prediction based on which option has the highest probability.

Model Compilation

We used three parameters for compiling our model. Those are Loss function, Metrics and Optimizer. We used categorical_crossentropy as a loss function. We used accuracy metric which will allow us to view the accuracy score on the validation data when we train the model. Also, we used adam optimizer.

Model Training

We cycled the data through the model in specific number of times until to reach an accuracy score, we are happy with. We started with 100 epochs which is the number of times the model will cycle through the data. The model improved on each cycle until it reaches a certain point. We also started with a low batch size, as having a large batch size can reduce the generalization ability of the model.

Model Evaluation

We tested our model accuracy on both the training and test data sets, which separate audio .wav files.

4 Results and Discussion

We trained two models, Multilayer Perceptron (MLP) Neural Network and Convolutional Neural Network (CNN) models. The datasets are evenly distributed as 100 instances from each class are representing the datasets. Hence, the accuracy of models--number of correctly classified instances vs. the total number of instances--also an acceptable matric to evaluate the performances of the models. Table 2 shows the overall models' accuracy on tainting and testing sets. Accordingly, Multilayer Perceptron (MLP) Neural Network model provides 99%and 98%on the training and testing sets respectively. Also, the Convolutional Neural Network (CNN) model achieves 100% and 99% on training and testing sets respectively. Hence, both models provide very decent accuracies on classifying environment sounds.

According to Fig.1, the training time of CNN model is faster than the MLP model. The CNN model reaches the maximum performances using less than 20 epoches whereas MLP uses arround 60 epoches.

Tables 3 and 4 dipic the perfromances of the models in each sound class. The models are tested using separate test sets. The prediction accuracy is measured using precision, recall and F1-score. Accordingly the CNN outperforms the MLP. However, the MLP is also provies a very decent accuracy on classifing sound classes.

Table 2 Classification Accuracies of Models

Model	Training Accuracy	Testing Accuracy
Multilayer Perceptron (MLP) Neural Network Model	99%	98%
Convolutional Neural Network (CNN) Model	100%	99%

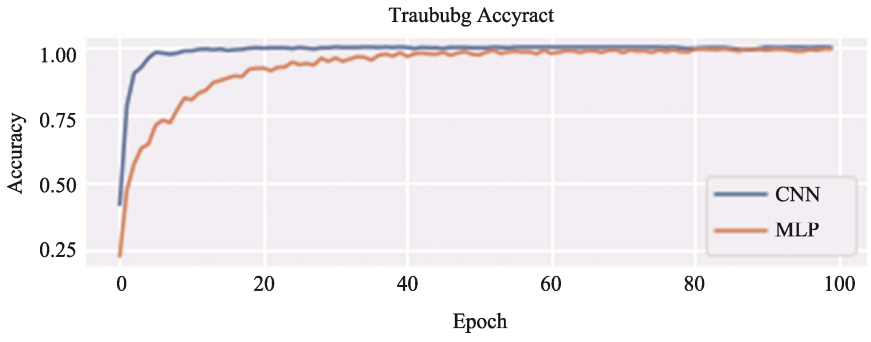

Fig.1 Comparison between Training Time of Models

Fig.2 and Fig.3 show the results of the confusion matrixs of MLP and CNN respectively. According to the confusion matrix, siren results confuse the most among all classes in both the MLP and CNN classifications. Furthermore, the total number of dog bark audio clips in the test fold is 22. Among them MLP predicted 21 dog bark audio clips correctly and 1 audio clip is confused with car horn class whereas the CNN predicted the total number of dog bark audio clips correctly.

Table 3 Performance of MLP

Type	Precision	Recall	F1 Score	Support
Air-Conditioner	1.00	1.00	1.00	25
Car-Horn	0.93	1.00	0.96	13
Children-Playing	0.96	1.00	0.98	22
Dog-Bark	1.00	0.95	0.98	22
Drilling	1.00	1.00	1.00	22
Engine-Idling	1.00	1.00	1.00	23
Jackhammer	1.00	1.00	1.00	16
Siren	1.00	0.94	0.97	18
Street-Music	1.00	1.00	1.00	19
Accuracy			0.99	180
Macro Average	0.99	0.99	0.99	180
Weighted Average	0.99	0.99	0.99	180

Table 4 Performance of CNN

Type	Precision	Recall	F1 Score	Support
Air-Conditioner	1.00	1.00	1.00	25
Car-Horn	1.00	1.00	1.00	13
Children-Playing	0.96	1.00	0.98	22
Dog-Bark	1.00	1.00	1.00	22
Drilling	1.00	1.00	1.00	22
Engine-Idling	1.00	1.00	1.00	23
Jackhammer	1.00	1.00	1.00	16
Siren	1.00	0.94	0.97	18
Street-Music	1.00	1.00	1.00	19
Accuracy			0.99	180
Macro Average	1.00	0.99	0.99	180
Weighted Average	0.99	0.99	0.99	180

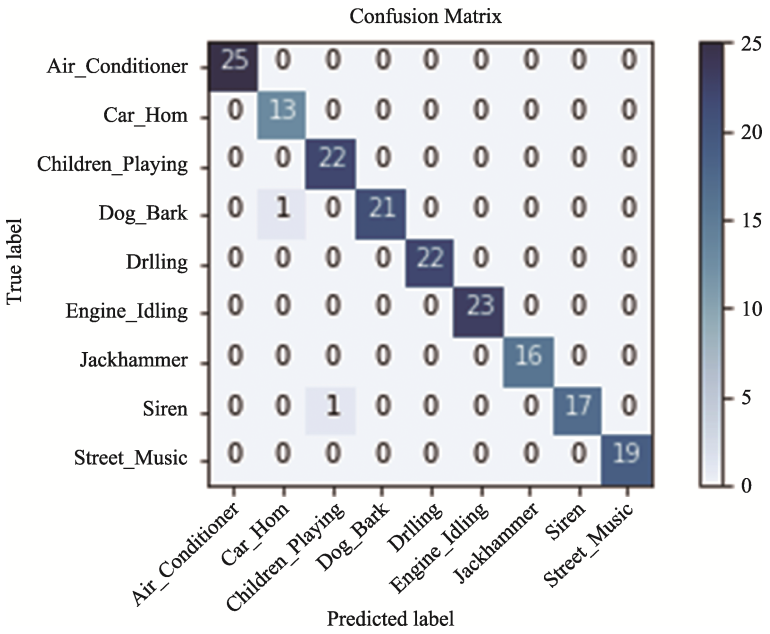

Fig.2 MLP Confusion Matrix

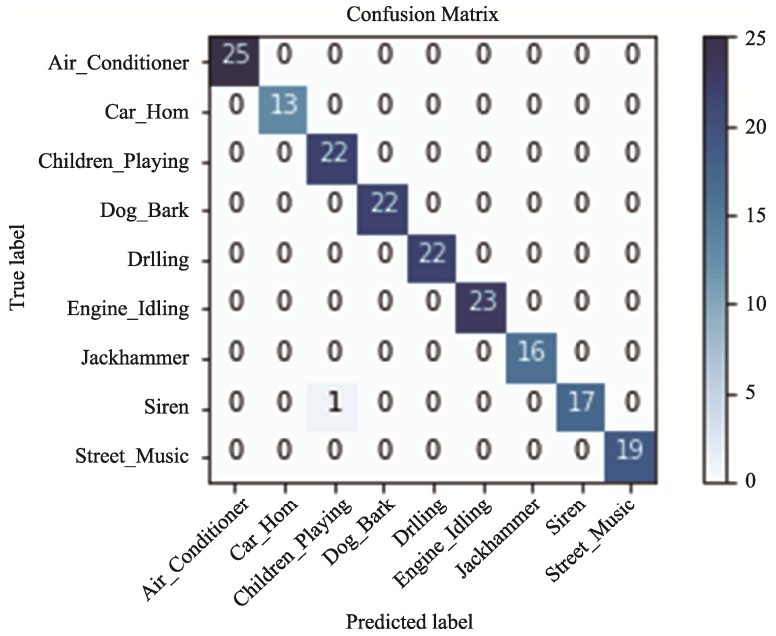

Fig.3 CNN Confusion Matrix

5 Conclusion

Operating audio files are not that hard as it appeared in the previous time. In the form of time series data, we can easily represent audio files. We have python libraries that have been predefined, which makes our job easier. The aim of this work is to extract audio features and then determine which environmental sound class the audio belongs to. We obtained900 audio samples without noises from some sources like YouTube, sound cloud, etc. The Mel-Frequency cepstral coefficients (MFCC) have been selected as one common audio feature to extract from each audio samples that we collected.We trained both models MLP and CNN respectively on the same dataset. Although we trained 2 distinct models for this, there was very little difference in accuracy between them. We analyzed whether such a blend is feasible and whether it can lead to improved results in the classification of environmental sounds.In summary, this work shows that the audio samples can be extracted with a reasonable accuracy based on different MFCC and then categorized into pre-defined categories. Finally, we come up with the conclusion as CNN is the better model compared to MLP based on the accuracy of both models. We classified sound into 9 classes however this can be further extended.

References

[1] Abdoli, S., Cardinal, P. & Koerich, A. L., 2019. End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications,* Volume 136, pp. 252-263.

[2] Chu, S., Narayanan, S. & Kuo, C.-C. J., 2009. Environmental Sound Recognition With Time–Frequency Audio Features. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING,* Volume 17, pp. 1142-1158.

[3] Demir, F., Abdullah, D. A. & Sengur, A., 2020. A New Deep CNN Model for Environmental Sound Classification. *IEEE Access,* Volume 8, pp. 66529-66537.

[4] Demir, F., Turkoglu, M., Aslan, M. & Sengur, A., 2020. A new pyramidal concatenated CNN approach for environmental sound classification. *Applied Acoustics,* Volume 170.

[5] J. Salamon, C. Jacoby, and J. P. Bello. 2020.A dataset and taxonomy for urban sound research. Justinsalamon.com. [online] Available at: <http://www.justinsalamon.com/

uploads/4/3/9/4/4394963/salamon_urbansound_acmmm14.pdf> [Accessed 28 April 2020].

[6] Karol, J. & Piczak, 2015. ENVIRONMENTAL SOUND CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS. *2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING,* pp. 17-20.

[7] Kim, T., Lee, J. & Nam, J., 2019. Comparison and Analysis of SampleCNN Architectures for Audio Classification. *IEEE Journal of Selected Topics in Signal Processing,* Volume 13, pp. 285-297.

[8] Koenig, M., 2020. Free Sound Clips | Soundbible.Com. [online] Soundbible.com. Available at: <http://soundbible.com//> [Accessed 25 April 2020].

[9] Mun, S., Shon, S., Kim, W. & Ko, H., 2016. Deep Neural Network Bottleneck Features for Acoustic Event Recognition. *INTERSPEECH 2016.*

[10] Pineau, J. et al., 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems,* Volume 42, pp. 271-281.

[11] Uzkent, B., Barkana, B. D. & Cevikalp, H., 2012. NON-SPEECH ENVIRONMENTAL SOUND CLASSIFICATION USING SVMS WITH A NEW SET OF FEATURES. *International Journal of Innovative Computing, Information and Control.*

[12] Salamon, J. and Bello, J.P., 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, *24*(3), pp.279-283.

[13] Wang, J. Lin, C. Chen, B and Tsai, M. "Gabor-Based Nonuniform Scale-Frequency Map for Environmental Sound Classification in Home Automation," in IEEE Transactions on Automation Science and Engineering, vol. 11, no. 2, pp. 607-613, April 2014, doi: 10.1109/TASE.2013.2285131.

Author Biographies

SHANTHAKUMAR S, is currently pursuing her B.Sc. (Sp) Degree in Computer Science & Technology from UvaWellassa University of Sri Lanka. Her main research interests include Neural Networks and Deep Learning.

Email: sampavi.shanthakumar@gmail.com

SHAKILA S, is currently pursuing her B.Sc. (Sp) Degree in Computer Science & Technology from Uva Wellassa University of Sri Lanka. Her main research interests include Neural Networks and Deep Learning.

Email: shakila8995@gmail.com

SUNETH Pathirana, is a PhD Senior Lecturer attached to the Department of Computer Science and Informatics, Uva Wellassa University of Sri Lanka. His main research interests include Brain-Machine Interfacing and Neurorobotics.

Email: vajira@uwu.ac.lk

JAYALATH Ekanayake, is currently working as a Lecturer in Computer Science at the Uva Wellassa University, Sri Lanka. His main research interest includes pattern recognition.

Email: jayalath@uwu.ac.lk