Acoustic Event Detection and Sound Separation for security systems and IoT devices

Alexander I. Iliev [1,2], Mayank Dewli [1], Muhsin Kalkan [1], Preeti Prakash Kudva [1], Rekha Turkar [1]

[1] SRH University Berlin, Charlottenburg, Germany,
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria

ailiev@berkeley.edu, mayank.dewli@stud.srh-campus-berlin.de, muhsin.kalkan@stud.srh-campus-berlin.de, preeti.prakashKudva@stud.srh-campus-berlin.de, rekha.turkar@stud.srh-campus-berlin.de

Abstract

When we think of audio data, we think of music and speech. However, the set of various kinds of audio data, contains a vast multitude of different sounds. Human brain can identify sounds such as two vehicles crashing against each other, someone crying or a bomb explosion. When we hear such sounds, we can identify the source of the sound and the event that caused them. We can build artificial systems which can detect acoustic events just like humans. Acoustic event detection (AED) is a technology which is used to detect acoustic events. Not only can we detect the acoustic event but also, determine the time duration and the exact time of occurrence of any event. This paper aims to make use of convolutional neural networks in classifying environmental sounds which are linked to certain acoustic events. Classification and detection of acoustic events has numerous real-world applications such as anomaly detection in industrial instruments and machinery, smart home systems, security applications, tagging audio data and in creating systems to aid the hearing-impaired individuals. While environmental sounds can encompass a large variety of sounds, we will focus specifically on certain urban sounds in our study and make use of convolutional neural networks (CNNs) which have traditionally been used to classify image data, for our analysis on audio data. The model, when given a sample audio file must be able to assign a classification label and a corresponding accuracy score.

Keywords Acoustic Event Detection, Convolutional Neural Networks, Deep Learning, Spectrograms

1 INTRODUCTION

We can easily identify an event or a setting by looking at an image. Various objects within an image describe an event or a setting. Similarly, various sounds can describe an acoustic event. Residential, indoor or industrial settings can be defined by a set of various audio signals. Sound detection and classification using deep neural networks [1] and effectiveness of CNNs for large scale audio classification [2] has already been established. However, acoustic scenes constitute a vast multitude of overlapping sounds which must be separated and classified for us to be able to identify the acoustic event. These systems can be used in military applications for threat monitoring and situational awareness. Surveillance systems often rely on video-based camera systems which are vulnerable to bad weather and low light conditions. Acoustic event detection systems can be used in conjunction with the pre-existing defense and surveillance systems to boost their capabilities. They can also be used in industrial settings to identify defects in equipment and machinery. Furthermore, these systems are economical and more cost effective when compared to video-based systems and thus can be installed almost everywhere.

2 SYSTEM ARCHITECTURE

2.1 APPROACH

Figure.1 displays the flow diagram of the implementation methodology. The system takes random urban sounds as input, transforms the audio signals and extract certain features on basis of which the random sounds are classified. Figure 2 shows how a time domain input audio is transformed into a frequency domain signal and then assigned an output label based on the extracted features.

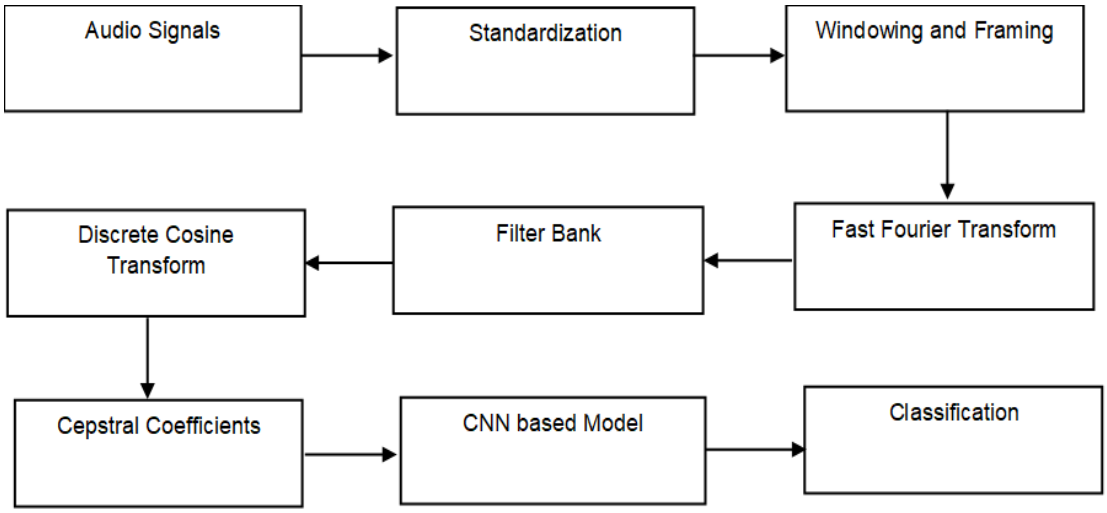

Figure 1. Overall Implementation Methodology

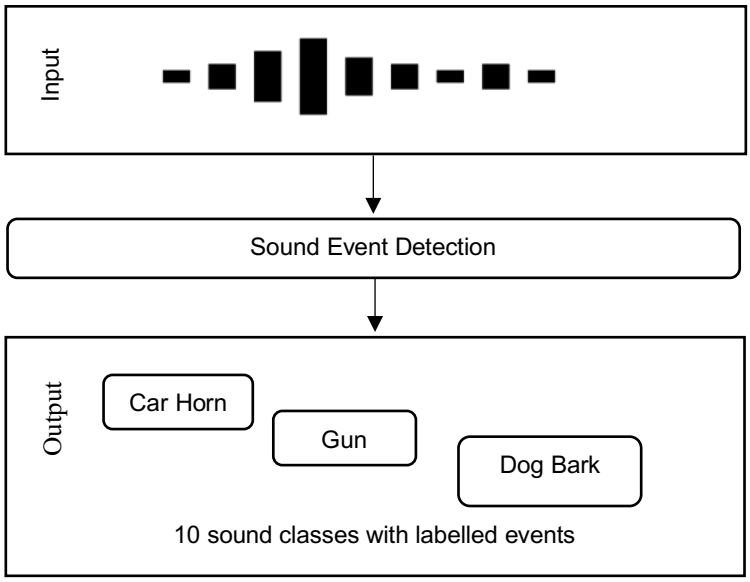

Figure 2. A Simple Sound Event Architecture [3].

3 DATA DESCRIPTION AND PREPROCESSING

3.1 DATASET

The dataset we used for urban sound detection is the Urban-Sound8k dataset which comprises of 8732 sound clips from field recording of duration <= 4 seconds each, in length and a varying sampling rate of 16kHz to 44:1kHz. All the excerpts in the dataset have been taken from field recordings uploaded to www.freesound.org while the classes are drawn from urban sound taxonomy [3]. The Urban-Sound8k dataset has 10 classes of urban sounds clips. Class IDs and number of samples for each class are shown in table 1.

Table 1. Name of ten labelled classes with its description

Class	Class ID	Number of Audio Samples
air_conditioner	0	1000
car_horn	1	429
children_playing	2	1000
dog_bark	3	1000
drilling	4	1000
engine_idling	5	1000
gun_shot	6	374
jackhammer	7	1000
siren	8	929
street_music	9	1000

3.2 ANALYSIS

3.2.1 DATA AUDITORY INSPECTION

Figure 3 shows the time domain waveforms associated with each of the 10 classes.

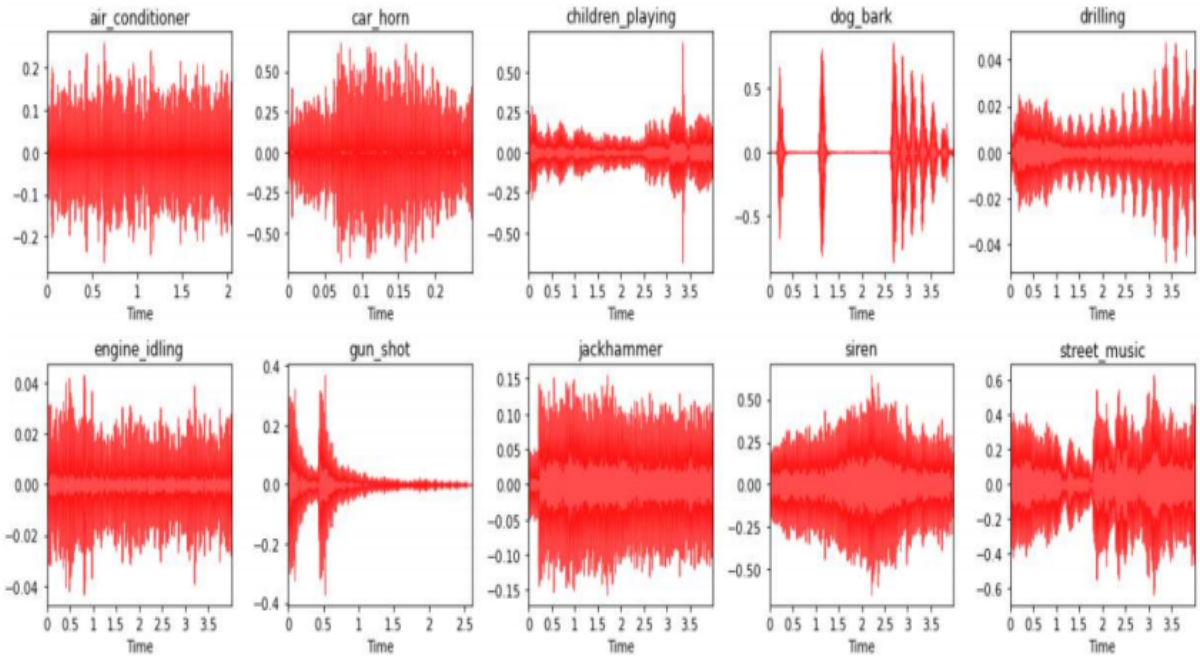

Figure 3. Time domain representation of different classes in the dataset

From visual inspection of time domain representations of the audio signals, we can see that it is quite difficult to ascertain the difference between some of the classes. Especially, the waveforms for repetitive sounds such as air conditioner, drilling, engine idling and jackhammer are similar in shape. It is possible to train a classification model using samples of amplitude values that approximate the signal in time domain, however, as a rule of thumb, models based on frequency domain features tend to outperform those based on time domain features, on audio classification tasks. We thus need to convert the time domain audio signals into a log scaled spectrogram so that we can identify various audio classes by analyzing the power spectral density of various frequency components in each audio file. Figure 4 shows the log scaled spectrograms of each of the 10 classes.

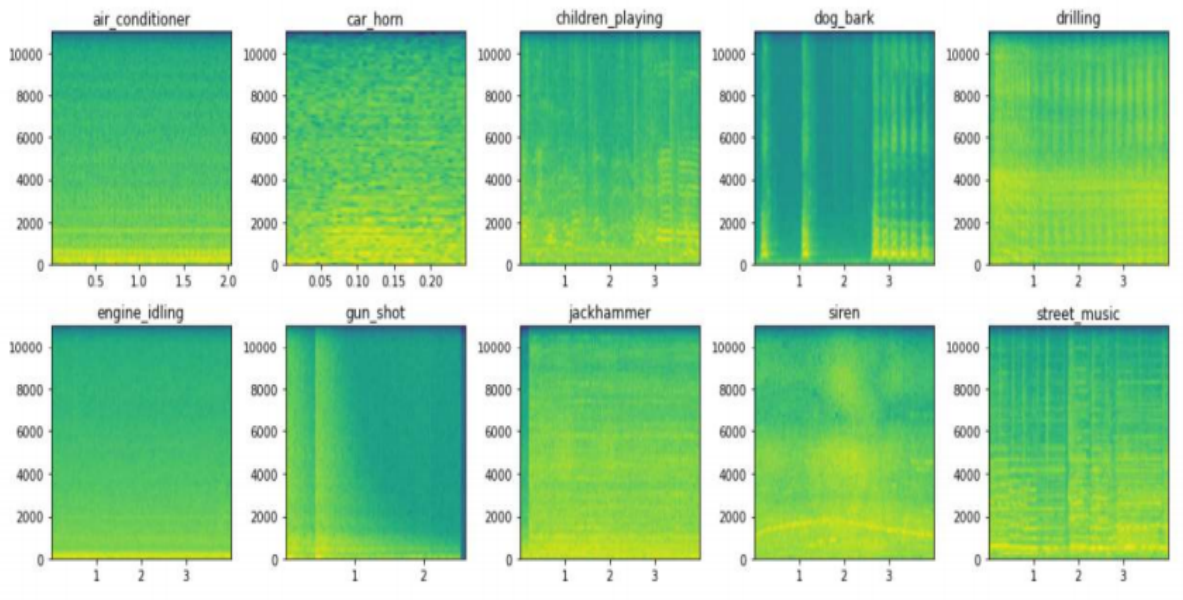

Figure 4. Log scaled spectrogram of a 10 audio classes.

The bright yellow color indicates higher magnitude and darker green color is indicative of lower magnitude. The data from the spectrograms of all the audio files in the input dataset can be converted to a numerical array or tensor which can then be fed to a convolutional neural network to train the model.

3.3 DATA PRE-PROCESSING

Audio data for preprocessing is obtained by sampling the analog audio signal at different time intervals and assigning the amplitude value to each sample. Sampling rate is defined as the number of samples per second of audio data. Thus, each audio sample is stored as a time series of numbers, where each value represents the amplitude. There was a wide variation in the Sample rates of various audio samples in our dataset, ranging from 96k to 8k. The bit depth determines how many possible amplitude values a given sample can take. A bit-depth of 16 which implies that each sample could take on 2^{16}or 65536 values. Most of the audio samples in the dataset had two audio channels (stereo) while a few audio samples had just one channel (mono). The bit-depth also varied between audio samples ranging from 4bit to 32 bits. We set the sampling rate to 22.5k and bit depth to 16 bits for all audio files, thus normalizing the data for further analysis. We used a sampling rate of 22,500 samples per second. Thus, one second of audio was represented as an array of 22,500 numbers.

3.4 MEL-FREQUENCY CEPSTRUM

Fourier transform provides an elegant method to represent a complex audio signal as a sum of sinusoids of various frequencies. However, valuable information about the time variation of signals is lost after applying a Fourier transform. STFT (Short Time Fourier Transform) makes use of a short time domain window that slides along the time domain and performs a FFT (fast fourier transform) of each segment, thereby giving us a frequency domain representation, while preserving the information about time domain variations of the audio signal. Humans perceive sound logarithmically and not linearly at higher frequencies. Thus, humans can easily differentiate between 100 Hz and 200 Hz signals as opposed to 1000 Hz and 1100 Hz signals.

To approximate our signal representation to match human perception of audio signals, we make use of Mel scale. A Mel Spectrogram plots the signal in terms of Mel Scale instead of Frequency scale, on the y-axis. Colors vary as per the logarithmic Decibel Scale instead of linear Amplitude Scale. For deep learning models, Mel spectrograms generally outperform Spectrograms.

The Mel-Frequency Cepstral Coefficients (MFCC) has been a popular feature extraction technique in recent years [4]. The development of MFCC was inspired by the way humans perceive audio data. It differs from other cepstral features in the frequency bands which are on the mel-scale.

Mel Scale is a non-linear representation of audio signals where frequency is represented as m mels, defined as:

$$m = 2595log_{10}(1 + f/700) \quad (1)$$

Mel frequency is F is defined as follows:

$$\mathrm{F}= [1000/\log_{10}(2)] * [\log_{10}(1 + \mathrm{f}/100)] \quad (2)$$

Here, f represents the frequency in Hz and F represents non-linear Mel-frequency. MFCCs can be obtained using the formula:

$$MFCC_m = \sum_{k=1}^{13}((log_{10}E_k)[(\frac{m\left(k-\frac{1}{2}\right))^n}{20})] \quad (3)$$

Here, m takes on values from 1 to N, where N is the number of Mel cepstral coefficients and E_k represents the energy emitted by the k^{th} filter [5].

3.5 DATA CLASSIFICATION USING CNN

CNN based models are being used for a variety of tasks from Audio Generation, Music Genre Classification to Environment Sound Classification in many studies. The ability of capturing patterns of time and frequency makes deep convolutional neural networks (CNNs) well suited for image and sound classifications models. When CNNs applied spectrograms shown to be very important distinguishing between noises and the actual sounds that we want such as car horn and sirens. The networks can successfully learn and identify Spectro-temporal patterns by using convolutional filters with small receptive fields, even the sound is masked by noises. Deep neural networks have a limitation, they are as good as the data we feed into them, they are dependent on large quantities of labelled training data to learn classification on untrained data. While new datasets are becoming available in recent years, the UrbanSound8k dataset is relatively low for research in sound classification. Data augmentation [6] is an effective solution to this problem, applying deformations to available training data without changing the semantic meaning of labels results in new additional training samples. For example, in computer vision scaled, translated, rotated, and mirrored images create new sets of data and would still be coherent images. However, the application of data augmentation in case of environmental sound classification is relatively limited, simple sound augmentation techniques like time stretching, time shifting, and pitch shifting have proven to be unsatisfactory in increasing model accuracy and decreasing training time. We use sound data augmentation to overcome the data scarcity problem and helps us to explore different types of sound deformations, data augmentation in combination with CNN provides increased performance for urban sound classification.

Each audio sample of duration 4 secs was divided into 4*22500 or 90,000 samples. Hamming window size of 2048 and hop length of 512 was used in the audio preprocessing stage. Each audio sample was further divided into 4 segments, to increase the size of training data. Samples with length less than 1 seconds were discarded. The total number of inputs were increased from 8,732 to 29,300 audio samples of 1 sec duration each. Thus, we had 22500 samples for each audio signal. The frequency domain transformations are applied not on the entire time duration of the audio signal (1 second), but on a small chunk of the audio whose length is equal to the hop length. Thus, we had 22500/512 or 44 segments for each second of every audio signal on which we apply frequency domain transformation. Using MFCCs we were able to extract 13 features for each of the 44 segments per second in the audio signal. The data was normalized between 0 and 1 as MFCC's are susceptible to noise. The shape of each input sample of 1 sec duration was now fixed to (44, 13). Since, convolutional neural networks are traditionally used for image data, a new axis was added to mimic 3 dimensional RGB image data. The new shape of each audio sample was set to (44,13,1). 29,300 sample were further divided into train, validation and test sets. 20 percent of the sample were reserved for the test set while 25 percent of the remaining samples were reserved as validation set. Therefore, we had 17,580 audio samples in the training set.

Figure 5. shows the architecture of the convolutional neural network used to train the model. The model had 3 Conv2d layers,3 MaxPooling2d and 3 batch normalization layers one after another in sequence. The output was flattened thereafter and fed to a dense layer before the dense output layer.

Batch Normalization: Batch normalization layer was added after each Conv2D and MaxPooling layer to stabilize the inputs and improve the learning rates.

Activation There were a total of 5 activation functions one for each of the three conv2d, one for dense layers after the succession of 3 conv2d and max pooling layers, and 1 for the output layer were used out of which 4 were Relu and 1 was SoftMax.

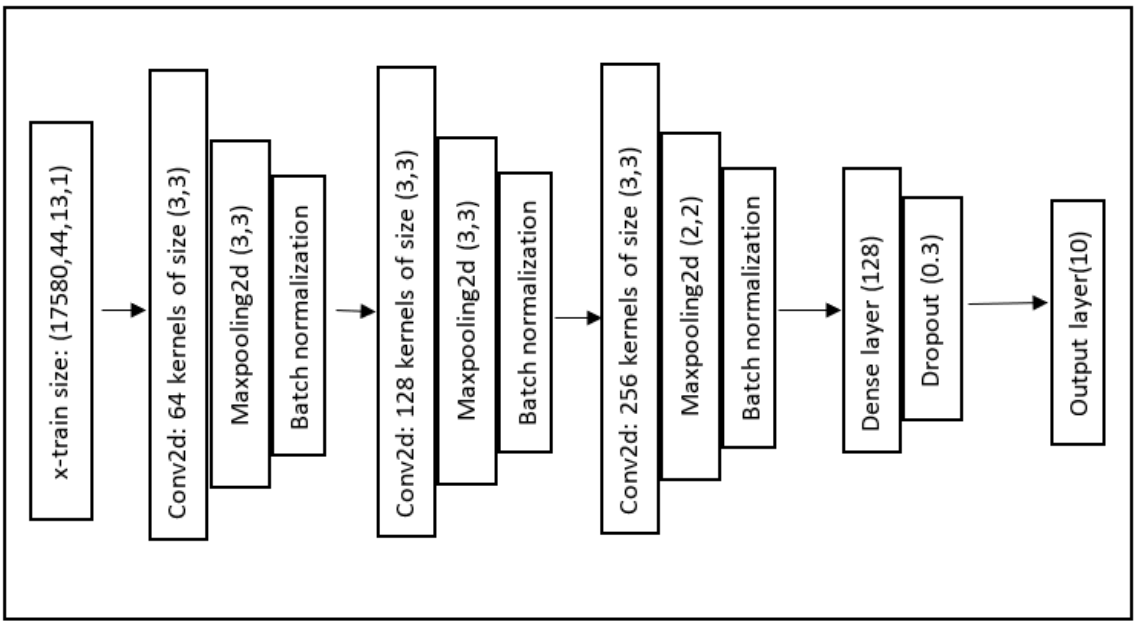

Figure 5. CNN architecture

Dropout Random dropout of a certain percentage of neurons during the training process is a regularization technique that is used to avoid overfitting [7]. We used a dropout rate of 0.3 for our dropout layer.

Max Pooling Max Pooling is a dimensionality reduction technique and reduces the number of computations during the training stage. We used max pooling layer thrice, immediately after conv2d layer each time. The size of kernels used was (3,3) and (2,2). The stride size was (2,2) and 'same' padding was used.

Flatten This layer flattens the matrix into one column vector. We used this layer after a succession of 3 conv2d and max pooling layers.

Dense The output after the flattening operation was fed to a fully connected dense layer with Relu activation followed by a dropout layer. This output was then fed to the output layer.

*Relu O*r Rectified Linear Unit is one of the most used activation functions. It was used as an activation after each conv2d layer and the dense layer. Relu is preferred over sigmoid function as it is simple, fast, computationally efficient [8] and less prone to exploding and vanishing gradient problems. It is a nonlinear function defined as follows:

$$f(x) = max(0; x) \quad (4)$$

Softmax Is often used at the multiclass classification problems as it assigns a probability score to each output label. In our model it assigns one out of 10 labels to each output.

4 RESULTS

Figure 6. shows the accuracy and error curve for 100 epochs plotted on the jupyter notebook using matplotlib library. We were able to achieve an accuracy of 86 percent for our validation set data and an accuracy of 85.7percent on the test data. However, we were able to significantly improve the accuracy scores after making use of data augmentation techniques and normalization of MFCC feature values. The validation accuracy increased to 92.2 percent and the model was able to predict 92.9 percent of the outputs in the test set correctly, which was a significant improvement over the previous results. The accuracy of the model diminished due to an unbalanced dataset with low sample size of gunshot, siren, and car horn audio samples. The model can achieve greater accuracy with a well-balanced dataset.

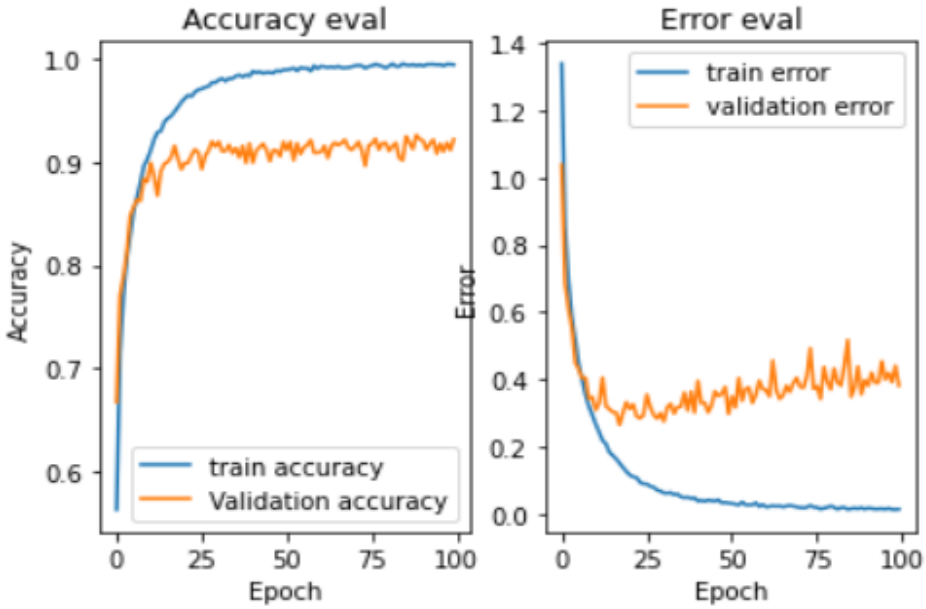

Figure 6. CNN architecture

5 CONCLUSION AND FUTURE WORK

Acoustic event detection can play a crucial role in improving the efficacy of surveillance, military and industrial systems. In this paper we attempted to create such system using random urban sounds. However, in real life scenarios, the sound data may be a combination of numerous overlapping audio signals and noise. Thus, it becomes necessary to perform necessary digital signal processing to separate different signals. Furthermore, the way in which audio features vary over time, can be important in identify various acoustic events. Therefore, models that preserve the time domain sequence of feature vectors to memory can be useful. Therefore, the efficacy of the systems can be improved by using LSTMs (long short-term memory networks) [9] or RNNs (recurrent neural networks) in conjunction with CNNs [10]. MFCCs can be combined with GFCCs, and various other frequency and time domain features to increase the accuracy scores. Data Augmentation techniques in the data preprocessing stage can further improve the accuracy scores.

6 REFERENCES

[1] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey and P. Tiwari, "Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network," in IEEE Access, vol. 7, pp. 7717-7727, 2019, doi: 10.1109/ACCESS.2018.2888882..

[2] S. Hershey et al., "CNN architectures for large-scale audio classification," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 131-135, doi: 10.1109/ICASSP.2017.7952132.

[3] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A Dataset and Taxonomy for Urban Sound Research. In Proceedings of the 22nd ACM international conference on Multimedia (MM '14). Association for Computing Machinery, New York, NY, USA, 1041–1044. DOI:https://doi.org/10.1145/2647868.2655045

[4] M. Rahmandani, H. A. Nugroho and N. A. Setiawan, "Cardiac Sound Classification Using Mel-Frequency Cepstral Coefficients (MFCC) and Artificial Neural Network (ANN)," 2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE), 2018, pp. 22-26, doi: 10.1109/ICITISEE.2018.8721007.

[5] A. I. Iliev and M. S. Scordilis, "Emotion recognition in speech using inter-sentence Glottal statistics," 2008 15th International Conference on Systems, Signals and Image Processing, 2008, pp. 465-468, doi: 10.1109/IWSSIP.2008.4604467.

[6] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," in IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279-283, March 2017, doi: 10.1109/LSP.2017.2657381.

[7] P. Dileep, D. Das and P. K. Bora, "Dense Layer Dropout Based CNN Architecture for Automatic Modulation Classification," 2020 National Conference on Communications (NCC), 2020, pp. 1-5, doi: 10.1109/NCC48643.2020.9055989.

[8] Y. Guo, L. Sun, Z. Zhang and H. He, "Algorithm Research on Improving Activation Function of Convolutional Neural Networks," 2019 Chinese Control And Decision Conference (CCDC), 2019, pp. 3582-3586, doi: 10.1109/CCDC.2019.8833156.

[9] I. Lezhenin, N. Bogach and E. Pyshkin, "Urban Sound Classification using Long Short-Term Memory Neural Network," 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), 2019, pp. 57-60, doi: 10.15439/2019F185.

[10] J. Sang, S. Park and J. Lee, "Convolutional Recurrent Neural Networks for Urban Sound Classification Using Raw Waveforms," 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 2444-2448, doi: 10.23919/EUSIPCO.2018.8553247.