Urban Sound Classification using Convolutional Neural Networks

Abstract— **The purpose of this research paper is to determine whether the classification through convolutional neural networks of urban sounds can be made more efficient and accurate. In the recent past, it has been observed that pedestrians all over the world are becoming increasingly prone to accidents due to the lack of awareness of incoming obstacles in their nearby surroundings. Our research aims to curtail this threat by using sound as a tool of detecting potential hazards, making the accuracy of classification imperative. As a result, pedestrians will be made aware of the approaching threats in the environment. For this purpose, a deep-learning model consisting of 3 convolutional layers with max-pooling and the last convolution layer with the GlobalAveragePooling2D has been implemented. The accuracy of this network has been evaluated on UrbanSound8k, a publicly available dataset with over 8,700 audio samples.**

Keywords—

I. Introduction

In the field of audio classification, there has been abundant research in the domain of music and speech recognition. However, urban sound data has been largely overlooked as a worthwhile area of research. Nowadays, audio classification applies Convolutional Neural Networks as a tool for detecting features from a set of audio data. CNNs are also widely used in the fields of image processing, image stitching, document analysis etc.

Over the years, extensive research has been carried out in the domain of audio classification via implementation of convolutional networks. In this regard, researchers have used a variety of datasets with ESC-10, ESC-50, UrbanSound8k etc. being the prominent ones offering access to a large variety of audio samples. In spite of the considerable success of the existing networks, one still feels that by implementing an alternative approach, a higher level of accuracy can be achieved. This shortcoming becomes even more prominent when we focus on the aspect of pedestrian safety. According to a study conducted by researchers in the United States [4], the number of accidents occurring due lack of awareness of the surroundings has increased by roughly 200%. This alarming statistic brings the question into mind *"Can sound classification help in injury prevention?"*

II. Litrature Review

In this section a few related works will be discussed.

[1] The paper employs a CNN model having 2 convolutional layers with max pooling and 2 fully connected layers. The model has been tested on 3 datasets and achieved a highest of 73.3% accuracy on UrbanSound8K dataset. [2] The proposed solution is a deep CNN model and data augmentation. This model supports a total of 5 layers; the first 3 layers are convolutional layers with max pooling on 2 of them and the last 2 are fully connected layers. Data augmentation part includes 5 different deformations, which include time stretching, 2-time shifts, dynamic range compression, and background noise mixing. 73% accuracy is achieved without augmentation, and a highest of 79% is achieved after data augmentation. [3] This paper proposes a solution based on CNN and the Tensor Deep Stacking Network (TDSN). Both solutions are applied to two different datasets. The CNN model consists of 4 layers, sub-divided into 2 convolutional layers, 1 fully connected and 1 output layer. TDSN model consists of 2 parallel hidden layers, which connects to 3rd order weight tensor for combining the results of hidden layers to output layer. Network training is divided into 3 stages, network training is done using 2 stacked blocks, then 3 and finally with 4 stacked blocks. On ESC-10 and ESC-50, CNN achieved the accuracy of 77% and 49% respectively. And 56% accuracy on ESC-10 using TDSN. [4] This paper uses a combination of 2 four layered CNN with different preprocessing on data, which are then combined using Dempster-Shafer evidence theory to compose TSCNN-DS model. Log-mel spectrogram and MFCC are extracted from the data and preprocessed data is fed to 2 CNN models with the same architecture. CNN models have 4 convolutional layers and 1 fully connected layer. The proposed model achieved a very high level of accuracy, but at double the cost both in terms of time and processing power. [6] The proposed solution is to use a dilated CNN by replacing the max pooling layer with dilated convolution. It consists of 3 groups, with 2 models in each group: one with the max pooling layer and the other model with the dilated convolution. Experiments are also conducted for different dilation rates and different numbers of layers in models. Of all the experiments conducted, the best model achieves an accuracy of 78%.

CNN based approaches for sound classification have garnered a lot of attention over the years, and have been accepted as reliable diagnostic tools. The models discussed above have all shown the utility of such methods. However, the proposed solutions have failed to achieve a trade-off between optimum accuracy and cost effectiveness. Thus, this paper presents a deep learning model that aims to fill these gaps.

III. CNN

Convolutional Neural Networks (CNNs) have become increasingly popular in the field of deep learning as they offer the innovative ability to simultaneously learn a variety of filters pertaining to a dataset which is being modeled to a specific problem, for instance sound classification, image recognition, image stitching etc. They are specifically designed to work with two-dimensional images, although one-dimensional and threedimensional images may find themselves to be applicable as well. As a result, particular features can be extracted from an input, commonly known as kernels or filters. The filter is smaller as compared to the input data and is multiplied to the filter-sized patch by an inner product.

A. *Convolution Layer*

Convolutional layers are central to the working principle of CNNs. Inside a convolution layer, arrays of sound data are multiplied with a two-dimensional array of weights referred to as a filter, resulting in a scalar product. Therefore, convolution is a linear operation.

CNNs traditionally have a series of convolution layers by which they can extract features from the incoming data and lay them out on a feature map. Filters pass over the sound data searching for patterns. In the instance where the filter finds a corresponding pattern in the input data, it returns a positive value, otherwise returning a minimum value.

B. *Pooling layer*

To generate a new collection of the same number of pooled feature maps, the pooling layer operates separately on each feature map. Pooling requires the selection of a pooling operation to be applied to feature maps, much like a filter. The size of the filter is much smaller than the size of the feature map; in particular, usually 2×2 pixels are used with a stroke of 2 pixels.

One of the key characteristics of the pooling operation is that it is not learned; rather it has to be specified beforehand. Few of the functions used in this process are:

- **Average Pooling**: This function helps us to compute the mean values on the feature map
- **Maximum Pooling (also known as Max Pooling)**: This function enables us to compute the ceiling values on the feature map

Typically, max pooling is preferred over average pooling. The pooling layer is implemented separately on each of the convolution layers, giving a concise version of the features and thus, reducing the computation in a CNN.

C. *Dropout Layer*

In each of the layers of the CNN, a regularization strategy called dropout is often implemented involving the parallel training of several neural networks with a variety of architectures. Some layers are temporarily removed from the network or are *“dropped out”*. Consequently, the layer has the appearance of having a different number of nodes and connectivity to the preceding layer. In effect, during the training process, each update to a layer is performed with a different *“view”* to the optimized layer.

D. *Softmax*

The SoftMax function transforms a vector of K real values into a vector of K real values that adds to 1. The input values may be positive, negative, zero, or greater than one. This function converts them into values between 0 and 1, so that they can be interpreted as probabilities.

E. *MFCC*

Mel-frequency cepstrum (MFC) is a depiction of the power spectrum of sound for limited intervals. It can be computed by taking linear cosine transform of a log power spectrum. The operation is performed on a nonlinear mel scale of frequency.

Mel-frequency cepstral coefficients (MFCCs), represent the coefficients that constitute MFC. These are basically features used in speech recognition across a plethora of applications. MFC and cepstrum primarily differ due to the spacing between the frequency bands. For MFC, it is equally spaced as represented on the mel scale. This allows for more accurate approximation of human auditory response as compared to the linearly spaced frequency bands of cepstrum. In other words, it is a type of frequency warping, which may be used for better representation of sound signals.

MFCCs are commonly derived as follows:

1. First, we calculate the Discrete Fourier transform (DFT) of the input signal.
2. Then magnitude of DFT is wrapped in mel frequency.
3. Take the logs for each of the mel frequencies.
4. Take the Inverse Discrete Fourier transform (IDFT) at each frequency.
5. The MFCCs are the amplitudes of the resulting spectrum.

IV. Experimrnt Setup

- The sampling rate, which is defined as the digitization of sound waves by examining them at discrete intervals (naturally 44.1 kHz). Individually, samples represent the amplitude of the wave at a specific time span, where bit depth governs the detail within the sample, otherwise called the dynamic range of the signal (frequently 16bit i.e., range of 65,536 amplitude values).
- The sample rate is set to 22.05 KHz for additional processing.
- Normalized data contains bit depth values between -1 and 1.
- Signal conversion to mono, thus the number of channels will be constant i.e., 1.
- MFCC is generated from time series audio data.
- The purpose of convolution layers is primarily feature detection. The working principle can be described as placing a filter window over the input data and following it with matrix multiplication. The results are kept in a feature map. The operation is acknowledged as Convolution.
- Filter parameters determine the number of nodes per layer. These nodes increase in magnitude for independent layers as multiples of 16 i.e., 16, 32, 64 & 128. The kernel size parameter stipulates the size of the kernel window. For our case, it is 2, the resultant of which is a 2x2 filter matrix.
- The information state of (40, 174, 1) is relayed to the initial layer, where 40 is the quantity of MFCC's. In

total, the frames number 174 including padding. The 1 here implies that the sound is mono.

- ReLU is employed as the activation function for the convolutional layers. We have made use of a smaller dropout value i.e. 20%.
- All individual convolutional layers are associated to a pooling layer of MaxPooling2D type. However, the final layer is linked to a GlobalAveragePooling2D type. These layers work to decrease the dimensionality of the model (by diminishing boundaries and resulting calculation prerequisites). This gives shortened training time and reduced overfitting. The working can be described as taking the maximum and average size for each window, for Max Pooling and Global Average Pooling type, respectively. This is apt for feeding into the dense output layer.
- There are 10 nodes for the output layer which matches the number of conceivable classifications. SoftMax is used as the activation for the output layer, which causes the output sum to be 1. This can be viewed as a probability. A prediction is made by the model based on the option with the highest probability.
- The model is trained for 72 epochs with a batch size of 256.

V. DATA SET

The model is implemented on dataset “UrbanSound8K” [10]. The dataset encompasses urban sounds from 10 classes i.e., 8732 labeled sound excerpts (<=4s).
These are:

- Air conditioner
- Car horn
- Children playing
- Dog bark
- Drilling
- Engine idling
- Gun shot
- Jackhammer
- Siren
- Street music

All classes have been obtained from the urban sound taxonomy. [7]
Excerpts used have been acquired from field recordings found at www.freesound.org. There are 10 folders which cater to all the files, systematically arranged i.e. fold1-fold10. These are very useful for the recreation of the automatic classification results detailed above. In addition to these, a CSV file containing metadata about each excerpt is additionally given.

VI. RESULTS

This section presents the results of the classification of urban sounds using CNNs. Model evaluation was conducted in a 10-fold (UrbanSound8K) cross-validation regime with a single training fold used as an intermittent validation set.

Figures 3-6 represent various tools that have been used to measure the accuracy of the model.

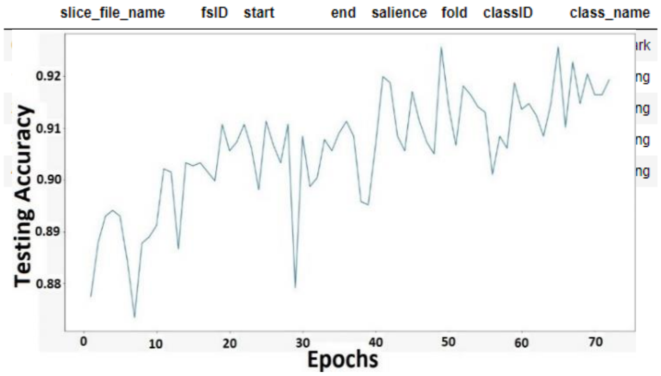

Fig. 3

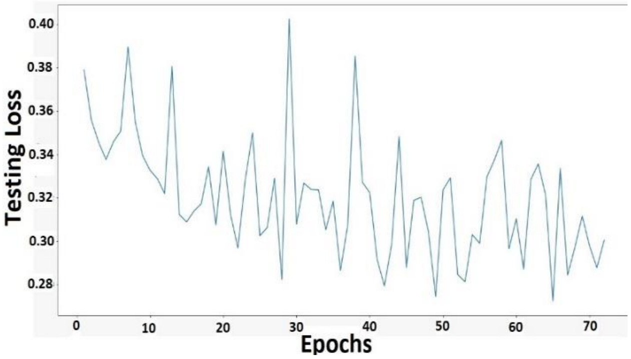

Fig. 4

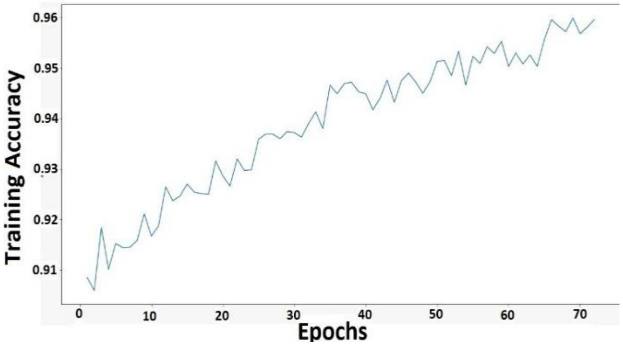

Fig. 5

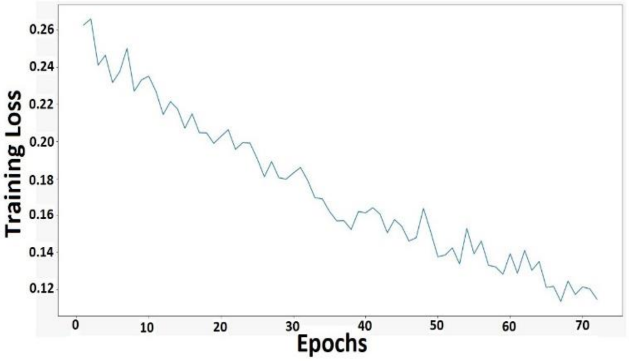

Fig. 6

Figure 3 shows that the testing accuracy increases with the increase in number of epochs. The final accuracy achieved by our model defined is 91.57% for the test data. Figure 4 displays the correlation between the testing loss and the number of epochs. An overall decrease is seen with a final loss value approximately equal to 30%. Figure 5 depicts the training accuracy, which is an overwhelming 95%, avoiding overfitting. While Figure 6 shows the training loss, which is a meager 12%. These are encouraging results which have led to an increased efficiency of our model.

VII. Summary

The purpose of this paper was to determine whether the classification through convolutional neural networks of urban sounds could be made efficient and accurate for various applications.

The deep-learning model consisting of 4 convolutional layers with max-pooling and GlobalAveragePooling2D had been implemented and proven to be efficient enough for further applications with sound classification. The accuracy of the network has been evaluated on UrbanSound8k, a publicly available dataset with over 8,700 audio samples.

A possible question open for future inquiry is whether the model is efficient for more diverse data and real time applications.

References

[1]"Environmental sound classification with convolutional neural networks - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7324337?casa_token=Uj2FqCEqs7IAAAAA:RdiJNMjfFhfJkhoUgOF1iKPmThf2WB3ndsCgTw2aQBYEVYi2IAIyp59Hw5g32imqrgRXAmnCux4. [Accessed: 13- Nov- 2020].

[2]"Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification - IEEE Journals & Magazine", *Ieeexplore.ieee.org*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7829341?casa_token=FKq0YpNqH1QAAAAA:gYTODj9gs6OnTJbWpTP9rcO5g8UuSuiq9cFYSw3TA7V5QnSNoxA44RgPYDKF1jOL3ogAhDSdD_o. [Accessed: 13- Nov- 2020].

[3]"Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network", *Ieeexplore.ieee.org*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8605515. [Accessed: 13- Nov- 2020].

[4]"Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion", 2020. [Online]. Available: https://www.mdpi.com/1424-8220/19/7/1733/htm. [Accessed: 13- Nov- 2020].

[5] S. Zhou, D. Rueckert and G. Fichtinger, *Handbook of Medical Image Computing and Computer Assisted Intervention*. 2020, pp. 379-399.

[6] K. Ming Li, *Applied Acoustics*. 2019, pp. 123-132.

[7]"Taxonomy", *Urban Sound Datasets*, 2020. [Online]. Available: https://urbansounddataset.weebly.com/taxonomy.html. [Accessed: 13- Nov- 2020].

[8]"Sound Classification using Deep Learning", *Medium*, 2020. [Online]. Available: https://mikesmales.medium.com/sound-classification-using-deep-learning-8bc2aa1990b7. [Accessed: 13- Nov- 2020].

[9] R. Lichenstein, D. Smith, J. Ambrose, and L. Moody, "Headphone use and pedestrian injury and death in the United States: 2004–2011", *Injury Prevention*, vol. 18, no. 5, pp. 287-290, 2012. Available: 10.1136/injuryprev-2011-040161.

[10]"Download UrbanSound8K", Urban Sound Datasets, 2020. [Online]. Available: https://urbansounddataset.weebly.com/download-urbansound8k.html. [Accessed: 13- Nov- 2020].