Urban Sound Classification Using Convolutional Neural Network and Long Short Term Memory Based on Multiple Features

Joy Krishan Das
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
joykrishan10@gmail.com

Arka Ghosh
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
arko.g96@gmail.com

Abhijit Kumar Pal
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
abhijitkp129@gmail.com

Sumit Dutta
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
somitdutt9@gmail.com

Amitabha Chakrabarty
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
amitabha@bracu.ac.bd

***Abstract*—There are many sounds all around us and our brain can easily and clearly identify them. Furthermore, our brain processes the received sound signals continuously and provides us with relevant environmental knowledge. Although not up to the level of accuracy of the brain, there are some smart devices which can extract necessary information from an audio signal with the help of different algorithms. Over the years several models like the Convolutional Neural Network (CNN), Artificial Neural Network (ANN), Region-Convolutional Neural Network (R-CNN) and many machine learning techniques have been adopted to classify sound accurately and these have shown promising results in the recent years in distinguishing spectra-temporal pictures and different sound classes. The novelty of our research lies in showing that the long-short term memory (LSTM) shows a better result in classification accuracy compared to CNN for many features used. Moreover, we have tested the accuracy of the models based on different techniques such as augmentation and stacking of different spectral-features. In such a way it was possible with our LSTM model, to reach an accuracy of 98.81%, which is state-of-the-art performance on the UrbanSound8k dataset.**

Keywords—Sound Classification, Spectrograms, Urbansound8k, CNN, LSTM, LibROSA

I. Introduction

Living in a world surrounded by different forms of sound from different sources, our brain and auditory system is constantly identifying each sound that it hears, in its way [1]. Classifying audio or sound has been a major field of research for many years now and there have been many tried and tested methods with different models and features which has proven to be useful and accurate. Classification of audio can range from fields like multimedia, bioacoustics monitoring, intruder detection in wildlife areas to audio surveillance, and environmental sounds [2].

There are three different stages that are attached to the classification of sound; pre-processing of the audio signal, specific spectral feature extraction, and finally the classification of the audio signal. Signal pre-processing samples the input audio signal into various fragments which are utilized for extricating essential features. Zero-crossing rate, spectral flux, and centroid, chroma vector, MFCC (Mel-frequency Cepstral Coefficient), poly features etc are among the most well-known handcrafted features for audio classification [3][4][8]. For the sake of our approach to audio classification, we have chosen the following features: Chromagram, Mel spectrogram, Spectral Contrast, Tonnetz, MFCC, Chroma CENS (Chroma Energy Normalized) and Chroma CQT (Constant-Q) [8]. Our main emphasis for comparison between the two models, which are Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM), was mainly based on the MFCC feature. The idea of MFCC is to convert time-domain signals into frequency domain signals and use Mel filters to mimic cochlea that has more filters at low frequency and fewer filters at high frequency. Thus it is safe to conclude that the feature MFCC and its characteristics are focused on the audibility of the human hearing system, that can accommodate the dynamic nature of true-life sounds with the way that they are treated with feature vectors for classification [3].

Once the steps regarding the feature extraction are all carried out, classification is required to be done using neural network models. There is a colossal number of potential techniques to construct Environmental Sound Classification (ESC) models utilizing distinctive sound or audio feature extraction procedures and AI or non-AI based models. SAI(Sensible Artificial Intelligence) and LPC(Linear Predictive coding) [4], DNN(Deep Neural Networks) [5], Decision Tree Classifier, Random Forest [6], ANN(Artificial Neural Networks) [7] are some of the popular models amongst other classifier models for audio.

For our purpose of audio classification, we are working with the Urbansound8K dataset and we are performing the classification via two different models which are CNN and LSTM. This has been done to put forth a comparison among the two models, using the same spectral features and the same dataset, to see which model gives a better accuracy result in classifying audio signal. The rest of our paper has been organized in the following manner: Section 2 enlists the previous research done on Convolutional Neural Network and other state-of-the-art techniques for sound classification. Section 3 contains the methodology of extracting different feature channels and augmentation techniques. Section 4 details our experimental setup and their implementation details. Section 5 displays our results on different feature channels. Finally, Section 6 concludes our research work.

978-1-7281-8084-7/20/$31.00 ©2020 IEEE

II. Literature Review

The potential of CNN to learn various spectro-temporal patterns has made them perfectly suited for the classification of environmental sound. In 2012 the work of Krizhevsky [9] CNN has been implemented in such an effective way that it changed the perception of researchers about this model. Since then it has been possible to significantly improve CNN in various pattern recognition tasks by replacing manually designed techniques. K.J Piczak [10] proposed a deep model that consists of two convolutional layers along with max-pooling layers and two fully connected layers which is trained on a low-level representation of audio data. The model was tested in a cross-validation scheme of 5 folds (ESC-10 and ESC-50) and 10 fold (UrbanSound8K) with a single regime fold used as a cross-validation method. In all the cases, the neural network-based model performed better while classifying ESC-50 datasets (baseline accuracy: 44%, best network: 64.5%). Another approach to ESC classification from sub-spectrum segmentation was proposed in [16]. In this paper, CRNN and score level fusion has been implemented jointly to improve the accuracy of classification which could accomplish 81.9% correct classification on the ESC-50 data set.

J. Salamon and J.P Bello [11] proposed a deep CNN architecture for environmental sound classification consisting of three convolutional layers interleaved with 2 max-pooling operations and followed by 2 fully connected layers. They also proposed to use the audio data augmentation techniques to resolve the data scarcity problem throughout the UrbanSound8k dataset. As a consequence of the proposed augmentation, this proposed model significantly increases the efficiency by resulting in a mean accuracy of 79%. Sharma et al. proposed a model consisting of several feature channels provided as inputs to a Deep Convolutional Neural Network (DCNN) for the Environmental Classification Task (ESC) [2] with the help of different data augmentation techniques. This model is innovative in that it utilizes a variety of five feature channels which has been able in achieving state-of-the-art performance in all of the three different benchmark environment datasets used i.e. ESC-10 (97.25%) and ESC-50 (95.50%), UrbanSound8K (98.60%).

I. Lezhenin and N. Bogach proposed a model based on LSTM for urban sound classification because of its efficiency in time dependency learning [12]. The model is trained over the magnitude of the Mel-spectrogram derived from UrbanSound8k dataset. In this proposed approach, both of the models provide almost similar results while the accuracy of the CNN model is 81.67% and the accuracy of the proposed LSTM model is 84.25%. Sainath et al. also proposed a model in which they have used the complementarity of CNN, LSTM, and DNN through integrating them into a single architecture named CLDNN could provide a comparatively 4-6% boost in WER [13].

Y. Tokozume and T. Harada proposed a novel end-to-end method which is implemented for detection and classification of raw sound signals by CNN [17]. Overall the three benchmark baselines, their proposed model called EnvNet has achieved competitive performance. The authors employed a system called Before Class (BC) learning in the second version of EnvNet which is EnvNetv2. The accuracy of the ESC-50 has been 5.1% higher than the accuracy obtained with the use of static log Mel-CNN on their proposed model. An approach for classifying environmental sounds that stands on multi-temporal Convolutional Network in combination with multi-level features has been proposed in [16]. Raw waveforms are used and a variety of independent CNNs are used in different layers which have been able to achieve an average improvement of 3.0% and 2.0% on ESC-50 and DCASE 2017 respectively. Another method of training a raw audio signal filter bank has been proposed in a very groundbreaking and successful unsupervised approach in [15]. A CNN is used as a classifier together with ConvRBM filter bank which could accomplish 86.5% on the ESC 50 dataset.

Such outstanding research works discussed above have helped us by giving plenty of insights on different environmental datasets and achieving high efficiency in environmental sound classification.

III. Methodology

A. Dataset

Datasets used for audio classification contain short clips of environmental sounds that are extracted from Freesound [2]; a field recording repository containing more than 160,000 audio clips. Some of the known datasets are UrbanSound8k, ESC-10, ESC-50, and ESC-US. Since the datasets of ESC are not large enough we have decided to work with UrbanSound8k as we know from our preliminary knowledge that our Deep learning models will give better results with a large amount of data [36] as shown in Fig 1.

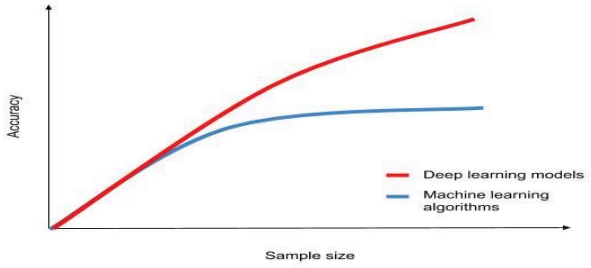

Fig. 1. Comparison of performance of Deep and ML models w.r.t sample size

In [18], an urban sound taxonomy is built which satisfies 3 rudimentary demands that include satisfying previously proposed taxonomies, detailing low-level sounds, and consist of sounds that contribute to noise pollution in urban areas. Besides, the UrbanSound8K dataset is built based on the proposed taxonomy in [18] which has a duration of 27 hours. The dataset contains 8732 labeled slices of audio that have varying audio length and sampling rate. The 10 classes are air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, police sirens, and street music. Most of the classes have approximately 1000 clips but the car horn and gunshot have fewer slices of 429 and 374 respectively. Hence, some of the baseline machine learning algorithms carried out on the dataset in [18] reveal less accuracy for car horn and gunshot. The CSV file of UrbanSound8K contains 8 columns which are slice file name, fsID, start, end, salience, fold, classID and class names.

B. Audio Exploratory

The audio samples are in the *.wav* format and these continuous waves of sounds are digitized. They are converted into a one-dimensional numpy array of digital values, as shown in Fig 2, by sampling them at discrete interim. Such

waves are one-directional and can be used to represent a specific frequency or amplitude at given time instances. Therefore a specific sampling rate is used and these sample values can be assembled if the audio is needed to be reconstructed [19]. The choice of the sampling rate depends on the Nyquist-Shannon theorem [20] that shows a relationship between the rate of sampling of the analog to digital converter and the sampling of max waveform frequency. This states that to collect and reconstruct all information present in a continuous waveform, the sample rates should be twice as large as the frequency of the continuous waveform. So the complete wave can only be recorded into an array without any noise or attenuation when we follow the Nyquist-Shannon theorem. However, we chose LibROSA which is a python package audio and music processing with predefined functions [8]. LibROSA normalizes the data so that the values can be represented in the array are between -1 and 1.

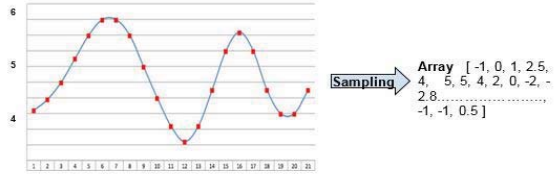

Fig. 2. Resulting Array after sampling (digital values are represented in red)

Fig. 2. shows the resulting one-dimensional numpy array after sampling, where the bit depth of the sound defines how comprehensive the sample will be. But LibROSA uses a default sample rate of 22050 Hz that assists in keeping the array size small by decreasing the audio quality but greatly reducing the training time. Therefore when we load the audio using *librosa.load()* function and the mono parameter is set to true, it combines the two channels of the stereo audio signal to make an one dimensional numpy array, thus creating a mono audio signal. Furthermore, the default frame and hop lengths are 2048 and 512 samples respectively [8]. Finally, after loading the audio and representing it in an array of values, various types of features are extracted from it.

C. Spectral Features Extracted

For our classification problem, we have used spectral features where we convert the audio from the time domain to frequency domain using Fourier transformation. There are numbers of spectral features like MFCC, Spectral Centroid, Spectral Roll-off, Mel spectrogram etc. Amongst all these spectral features we have decided to work with 7 of the features which are the following: MFCC, Mel Spectrogram, Chroma STFT, Chroma CQT, Chroma CENS, Spectral Contrast, Tonnetz as they provide distinguishable information and representations that are effective in classifying audio signals more accurately. The process of extracting features from an audio signal is made far easier with the help of LibROSA library where there are some built-in functions in python to generate the required spectrogram [8]. To illustrate that, for generating MFCCs we have to pass a few parameters into the *librosa.feature.mfcc()* function which are the loaded audio, the sample rate of the audio (in our case the default sample rate of 22050 Hz) and number of MFCCs to return. However, the function returns 40 MFCCs over 173 frames if we do not specify the number of MFCCs to return. Furthermore, we have taken the mean of 173 frames which converts the two-dimensional array into one-dimensional array with the mean values of the frames.

Such feature extraction functions of LibROSA library has made it possible for us to extract the following required features for our research work.

1) MFCC

Mel frequency cepstral coefficients in Fig. 3. are compact representations of the spectrum are typically used to automatically identify speech and it is also used as a primary feature in many research areas that include audio signals [21].

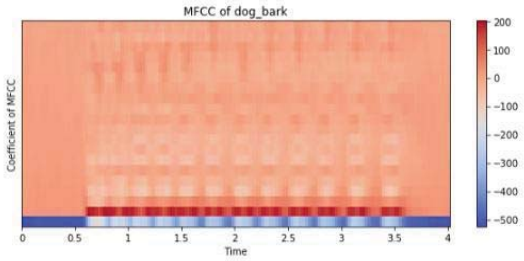

Fig. 3. Mel Frequency Cepstral Coefficient

2) Mel Spectrogram

A Mel spectrogram in Fig. 4 is the collaboration of Mel scale and spectrogram where Mel scale represents the non-linear transformation of the frequency scale [14]. In this case, the audio signal is first broken down into smaller frames and a Hamming window is applied on each frame. Then DFT is applied to switch from time domain to the frequency domain. A logarithm is used at the final stage to generate the spectrum that helps in the generation of the Mel spectrogram shown in Fig. 4.

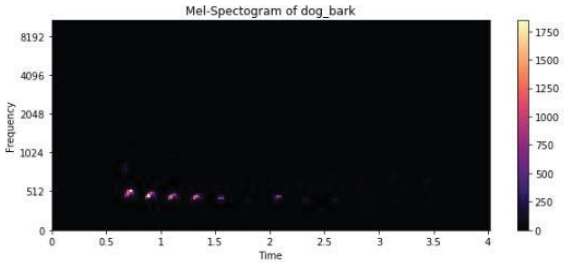

Fig. 4. Mel Spectrogram

3) Chroma STFT

The Chroma value of an audio basically represent the intensity of the twelve distinctive pitch classes that are used to study music. They can be employed in the differentiation of the pitch class profiles between audio signals. Chroma STFT in Fig. 5. used short-term Fourier transformation to compute Chroma features. STFT represents information about the classification of pitch and signal structure. It depicts the spike with high values (as evident from the color bar net to the graph) in low values (dark regions).

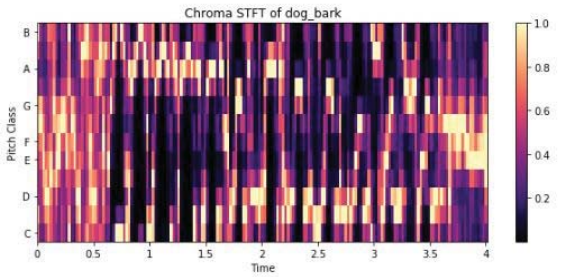

Fig. 5. Chroma STFT

4) Chroma CQT

Another approach of Chromagram is Chroma Constant-Q (CQT) transform in Fig. 6. which adjusts the STFT to logarithmically create space between the frequency bins. However, the constant bin size for all frequencies leads to some problems when you map frequency on a logarithmic scale [22]. CQT seeks to solve this problem by increasing the buffer size for lower frequencies and alleviate some of the computational strain caused by this by reducing the buffer size used for high frequencies [23].

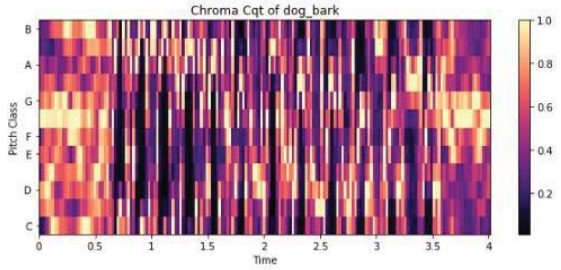

Fig. 6. Chroma CQT

5) Chroma CENS

Thus we can obtain CENS shown in Fig. 7. CENS functions can be interpreted effectively due to their low spatial resolution. The Chroma characteristics are centered on just the 12 pitch sounding properties that are familiar in the western music notation in which each Chroma variable demonstrates how the intensity of audio is dispersed through the dozen Chroma bands [24].

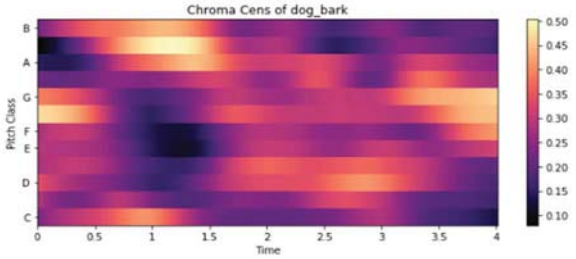

Fig. 7. Chroma CENS

6) Spectral Contrast

Another form of extraction technique that has been used in our proposed model is Spectral Contrast, which has been represented in Fig. 7. Spectral Contrast takes into account the spectral peak, the spectral valley, and the disparity between each sub-band frequency.

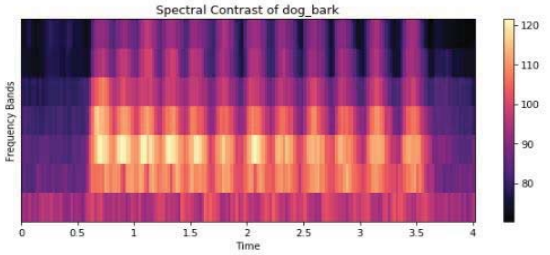

Fig. 8. Spectral Contrast

7) Tonnetz

The last audio extraction technique, shown in Fig. 8 we have worked with Tonnetz. Tonnetz computes the tonal centroid features by detecting the harmonic changes in musical audio clips. It is a well-known planar representation of pitch relations in an audio clip which is considered as an infinite planar.

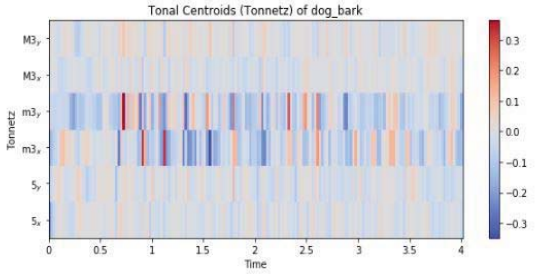

Fig. 9. Tonal Centroids

D. Data Augmentation

The concept of data augmentation is very straightforward; duplicate the existing datasets with variation to provide the models with more samples from which our two deep learning models can be trained better to provide a more accurate classification of the sound [25]. There are some augmentation techniques such as pitch shift and time-stretching which are effective in terms of data augmentation [11]. So we have adopted these techniques to augment the UrbanSound8K dataset of 8732 audio files. After augmentation, we achieved a total of 78588 audio clips which were labeled automatically. Since we have seen in Fig .1 that the number of data increases the performance of the deep learning models, we took fairly enough data to achieve a state of art performance using our model which is shown in the result part. We have taken three different audio data augmentation techniques that are mentioned below:

1) Pitch Shift

The factors of {-2, +2} are used to raise and lower the pitch (in semitones) of an audio sample in the dataset through which we could create 17464 samples using pitch shift. Fig. 10 represents how the signal varies when the pitch is changed. In comparison with the original signal *blue*, the *orange* audio signal has less frequency since the pitch is reduced by a factor of 2. Whereas the *green* signal has a higher frequency due to pitch increment by a factor of 2.

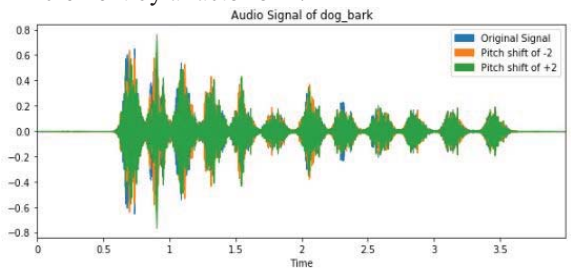

Fig. 10. Audio Signal of Dog_bark after Pitch Shifting

2) Time Stretch

In this technique, we slow down or speed up the sound clips with a rate of 0.9 and 1.1. Here the rate of 0.9 slows down the audio therefore the spikes of the *orange* signal in Fig. 11 occur after few seconds compared to the original signal *blue*. On the other hand, the *green* audio signal happens to occur before the *blue* sample audio. In this way, we could generate more 17464 new audio clips for our augmented dataset.

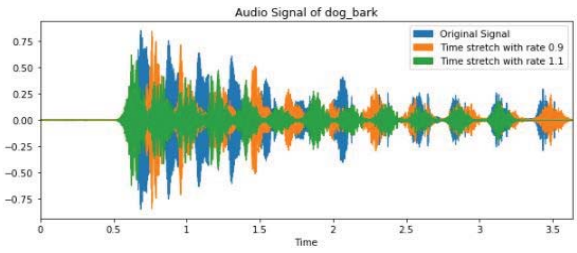

Fig. 11. Audio Signal of Dog_bark after time-stretch

3) Pitch Shift along with Time Stretch

In this augmentation step, a sound clip is manipulated using both pitch shift and time stress to generate a total of 34928 novel audio clips. In Fig. 12 We can see that the *orange* and *green* signal's variation concerning time and normalized amplitude when it is pitch-shifted along with time stretch with the factors mentioned above. To illustrate this, a sample of dog bark is shown below in Fig. 12.

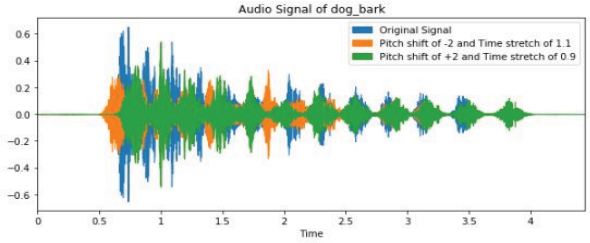

Fig. 12. Audio Signal of Dog_bark after pitch-shift along with time-stretching

IV. Model Architecture

We have used CNN and LSTM which takes the spectrograms as input in our proposed approach which is explained below:

A. CNN Model

In a typical CNN, there is a series of various kind of layers which are combined in the overall architecture. The training of a CNN requires a different kind of decisions which need to be taken in terms of both architectural patterns such as number of convolution and pooling layers, input data format, filter dimension, etc, as well as hyperparameters such as learning rate, dropout probability, number of epochs, batch size, etc [10]. Our proposed model is a sequential model consisting of two Conv2D layers followed by three dense layers among which the final layer is the output layer as shown in Fig. 13.

1) Convolutional Layers

The first Conv2D layer which is receiving the input shape consists of 64 filters, a kernel size of 5, and stride set to 1. The parameter determines the kernel window's size which in our case is 5 that results in a 5x5 filter matrix. As stride is set to 1 in the first layer, the filter converges around the input volume by moving one unit at a time. This convolutional layer with the 'same padding' operation produces an output of the same height and weight as the input. The activation function we have used for this layer is called ReLU which has got several benefits compared with conventional units such as effective gradient propagation and quicker computation rather than sigmoid units [15]. The first Conv2D layer is followed by the MaxPooling2D layer which is used in reducing the dimension of the input shape. The second Conv2D layer consists of 128 filters, a kernel or of size 5, stride set to 1 along with 'same padding operation' and ReLU as an activation function. The second Conv2D layer is a dropout of 30% to reduce overfitting.

2) Flatten Layer

The flatten layer converts the output of the convolutional layers into a one-dimensional array to be inputted into the next hidden layers.

3) Dense Layers

Two dense layers and an output layer have been used for further processing of the model. The two dense layers will have 256 and 512 nodes correspondingly. The non-linear function which is followed in the first two dense layers is ReLU that simply switches all the negative activations to zero greatly reducing the time required for gradient descent.

4) Output Layer

The output layer having activation function softmax will consist of 10 nodes in our model that refers to the possible classification numbers. The model then predicts the choice with the highest probability.

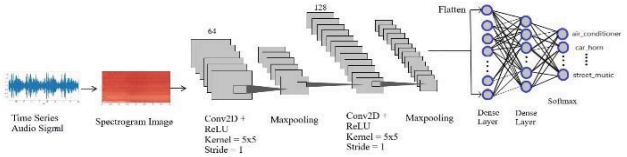

Fig. 13. Model Architecture of our Proposed CNN Model

B. LSTM Model

Audio classification with LSTM has become popular recently and LSTM has shown an impressive accuracy rate in the classification process. For the sake of our approach towards the audio classification, we have used two LSTM layers, followed by two Time Distributed Layer, and a flatten layer and lastly a dense layer shown in Fig. 14.

1) LSTM Layers

In our model, we are stacking two LSTM layers. In both the LSTM layer there are 128 hidden units and return sequence has been set to "true" because to stack LSTM layers, both the LSTM layers require to output a 3D array which will be used by the following Time Distributed Dense layers as input. In the first LSTM layer of our model input shape provided is (20,5), where 20 represents the time steps which lets an LSTM layer know how many times it should repeat itself once it is applied to the input. A dropout of 0.3 has been used to reduce overfitting.

2) Time Distributed Layers

Then comes the two time distributed dense layers in the model. The size of the input in the first layer is 128 and the number of nodes in the output layer of the first time distributed dense layer is 256. For the second Time Distributed layer the size of the input is 256 and the output layer contains 512 nodes. In both layers, ReLU has been used as the activation function.

3) Flatten Layer

The 3D output from the time distributed dense layer is taken in as input by the flatten layer. After the flattening is

done, we end up with a long vector of input data which is then passed onto the last layer which is the dense layer.

4) Dense Layer

The output dense layer feeds all the output that it has fed from the previous flatten layer to all the neurons in the hidden layer which is 10 in our case. We have used a softmax activation function in the output dense layer, which converts its inputs into a discrete probability distribution.

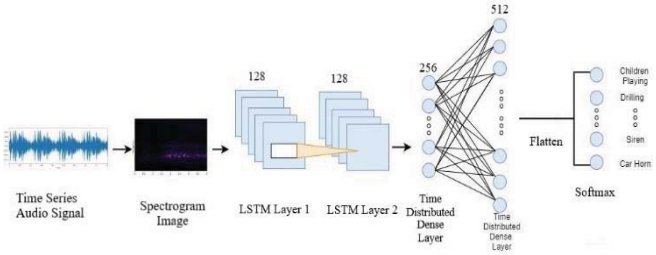

Fig. 14. Model Architecture of our Proposed LSTM Model

C. Model Summary of CNN & LSTM

For training our CNN and LSTM model, we have started with 30 epochs. The Fig. 15 and Fig.16 displays the validation loss of CNN and LSTM respectively in the Y-axis and number of epochs in the X-axis. In the figure, we can see that with the increase in the number of epochs the validation error of the model is decreasing exponentially for both the training and testing data. We have chosen 30 epochs for our model because the validation error stops improving somewhere around the 25th epoch. The entire dataset is passed forward and backward through the model in one epoch. We split the dataset into a batch size of 50.

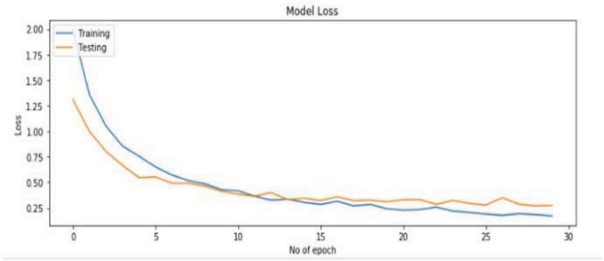

Fig. 15. Validation Loss of CNN

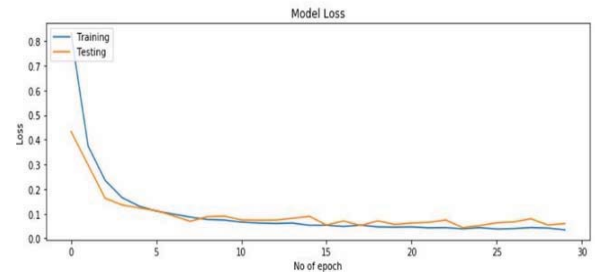

Fig. 16. Validation Loss of LSTM

D. Model Compilation of CNN & LSTM

1) Loss Function

Among different loss functions, the loss function we will be using in both of our models is Categorical Crossentropy because of the increased number of classes in our model which determines the categorization of a single label. The equation of Categorical Crossentropy:

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij})) \qquad (1)$$

2) Metric

The metric we will be using in both of our models is an accuracy metric which is the fraction of our model's predictions. It will let us see the validation data accuracy score during the model training. Accuracy is generally obtained by the following equation:

$$Accuracy = \frac{No.of\ Correct\ Predictions}{No.of\ Total\ Predictions} \qquad (2)$$

3) Optimizer

Here we will be using the optimizer Adam that measures the adaptive learning rate for each parameter. This uses the adaptive models of learning rates to determine specific learning levels for each parameter. Adam prefers the flat surface error minima for which it is proven to be a good optimizer in many cases.

V. RESULTS

Our approaches to the research included the identification of features that work effectively and produce high accuracy for both the models. In such scenarios, MFCC was the best feature that showed the highest accuracy for both the models compared to other features. Many types of research use MFCC as the only feature to classify sound [3]. MFCC single-handedly gives the highest performance boost and stacking it with features like Chroma and Mel gives a slight increase in the performance which is shown in Table II. The other approach to research was to stack different features together to find out how the classification accuracy of the model's changes in accordance. Some of the stacking worked significantly well in boosting the performance of our model, for example, stacking MFCC and Chroma STFT helped us to reach a validation accuracy of 98.81% shown in Table I, whereas the best performance till date was 98.60% in [2]. On the other hand, augmentation plays a significant role in increasing the performance of both systems shown in Table II and Table III.

TABLE I. PROPOSED MODEL VS THE PREVIOUS STATE OF ART MODELS

Model	**Accuracy (%)**
EnvNet [28]	66.30
EnvNet + logmel-CNN [28]	71.10
EnvNetv2 [17]	76.60
EnvNetv2 + Strong Augment [17]	78.30
M18 [29]	71.68
PiczakCNN [10]	73.70
AlexNet [30]	92.00
GoogleNet [30]	93.00
SB-CNN [11]	79.00
CNN + Augment + Mixup [31]	83.70
GTSC ⊕ TEO-GTSC [32]	88.02
1D-CNN Random [33]	87.00
1D-CNN Gamma [33]	89.00
LMCNet [34]	95.20
MCNet [34]	95.30
TSCNN-DS [34]	97.20
ADCNN-5 [2]	98.60
Multiple Features with CNN and LSTM (Proposed)	98.81

TABLE II. PERFORMANCE OF ALL IMPLEMENTATIONS USING CNN

Implementation with CNN			
Features	***Input Dimension***	***Accuracy without Augmentation (%)***	***Accuracy with Augmentation (%)***
MFCC	(20,5,1)	90.78	96.78
Mel-spectogram	(20,5,1)	83.11	94.42
Chroma_CQT	(20,5,1)	64.68	69.41
Chroma_CENS	(20,5,1)	31.37	32.04
Chroma_STFT	(20,5,1)	72.98	79.36
MFCC, Mel-spectogram	(20,5,1)	89.63	94.97
MFCC, Chroma_STFT	(20,5,1)	89.86	95.90
MFCC, Chroma_CQT	(20,5,1)	88.32	96.02
Mel-spectogram, Chroma_STFT	(20,5,1)	82.25	95.02
Mel-spectogram, Chroma_CQT	(20,5,1)	79.91	93.17
MFCC,Mel-spectogram, Chroma_STFT, Chroma-CQT	(20,5,1)	91.64	95.36
MFCC,Mel-spectogram, Chroma_STFT, Tonnetz,Spectral_contrast(174 features)	(29,6,1)	86.09	94.09

TABLE III. PERFORMANCE OF ALL IMPLEMENTATIONS USING LSTM

Implementation with LSTM			
Features	***Input Dimension***	***Accuracy without Augmentation (%)***	***Accuracy with Augmentation (%)***
MFCC	(20,5)	93.30	98.23
Mel-spectogram	(20,5)	81.85	96.25
Chroma_CQT	(20,5)	57.53	69.23
Chroma_CENS	(20,5)	30.51	32.40
Chroma_STFT	(20,5)	65.48	78.92
MFCC, Mel-spectogram	(20,5)	91.01	98.03
MFCC, Chroma_STFT	(20,5)	93.76	**98.81**
MFCC, Chroma_CQT	(20,5)	91.36	97.95
Mel-spectogram, Chroma_STFT	(20,5)	82.94	95.18
Mel-spectogram, Chroma_CQT	(20,5)	79.68	94.11
MFCC,Mel-spectogram,Chroma_STFT,Chroma-CQT	(20,5)	91.76	98.34
MFCC,Mel-spectogram, Chroma_STFT, Tonnetz, Spectral_contrast(174 features)	(29,6)	89.87	95.69

A. *Comparison between CNN and LSTM model*

CNN is a very powerful technique for classification based on image, designed in a way to exploit "spatial correlation" in data images and speeches. In scenarios where time-series data or sequential data is involved, LSTM does a better job as LSTM memory cell includes constant error backpropagation that allows LSTM to deal with data noise that is very useful for us, as the models have been applied with expanded datasets after augmentation using techniques of time stretch and pitch shift function from LibROSA [8].

In Fig. 17 and 18, we can see that LSTM outperforms CNN model in most cases when it comes to the classification of audio. In the Fig. 18 we have shown the significance of data augmentation which increases the testing accuracy by 4%. Our LSTM model has also achieved state-of-the-art performance with the help of data augmentation on a benchmark dataset of UrbanSound8k.

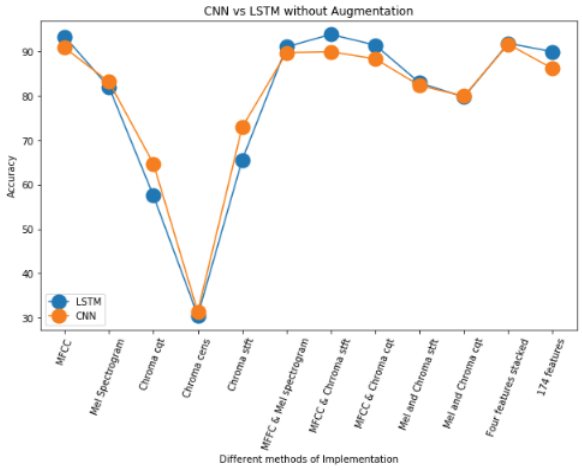

Fig. 17. CNN VS LSTM without Augmentation

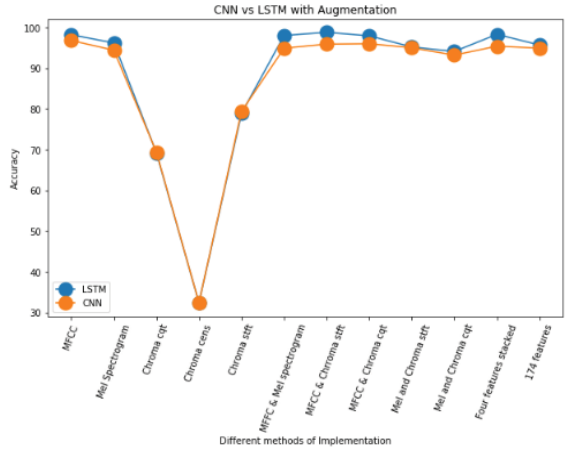

Fig. 18. CNN VS LSTM with Augmentation

B. *Analysis of State-of-the-Art Performance*

In Fig. 19, we demonstrated the confusion matrix of our best performing approach which is LSTM using stacked features of MFCC and Chroma STFT. Since it outperformed all the previous models, we declare our approach as state-of-the-art in audio classification.

After training our LSTM model with 62870 data samples, our model tested with a lot of other new samples. In the confusion matrix, we can see that out of 15718 of augmented testing data, our model has incorrectly classified only 186 testing samples reaching an accuracy of 98.81%. Furthermore, it performs well for all the classes with 97.7% being the least accurate for the class Jackhammer which is shown in Fig. 20. Moreover, we can also notice that jackhammer is misclassified as drilling and children playing is misclassified as street music most of the time because these samples sound similar.

However, the execution time of our LSTM model with MFCCs and Chroma STFT as features was 37.14 minutes and for implementing our model we have used GeForce RTX 2060 GPU with 6 Gigabytes of VRAM and boost clock of 1.68 GHz. On the other hand, approximately 8 Gigabytes of

RAM was consumed while training both models of CNN and LSTM.

In total, our model has performed outstandingly for all the classes in the UrbanSound8k dataset with the aid of augmentation and stacking techniques.

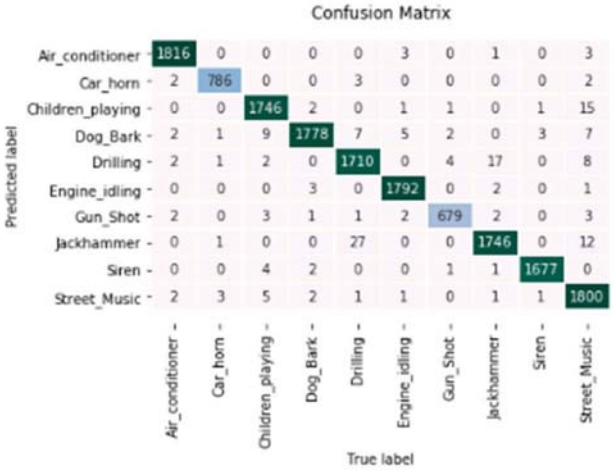

Fig. 19. Confusion Matrix of our state-of-the-art system

CLASS	ACCURACY
Engine Idling	99.666296
Air Conditioner	99.616018
Siren	99.525223
Street Music	99.118943
Car Horn	99.117276
Children Playing	98.867497
Drilling	98.050459
Dog bark	98.015436
Gun Shot	97.979798
Jackhammer	97.760358

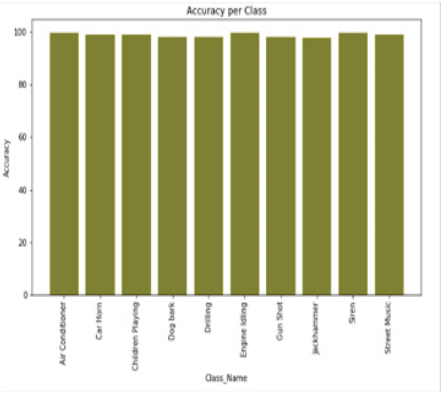

Fig. 20. Accuracy of each class for our State-of-the-Art

VI. Conclusion

We have presented an approach to sound classification, which consists of multiple features stacking and two different neural network models which are CNN and LSTM. Both the models have been trained and tested with original UrbanSound8K and its augmented dataset. Among these, using the MFCC and Chroma STFT stacked together which is used as a feature for the LSTM model and in this way we were able to achieve a state-of-the-art result. Furthermore, we would like to proceed with this in our future work where we have planned to use novel unsupervised learning techniques that can be adopted to train, test the models, and check their accuracy.

Acknowledgment

We author express our gratitude to Dr. Amitabha Chakrabarty, Associate Professor, CSE, BRAC University for his constant guidance that helped greatly to move forward with our research. This work has been (Partially) supported by the Bachelor program in CSE, BRAC University, Bangladesh

References

[1] Chachada, S., & Kuo, C. (2014). Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processing, 3*, E14. doi:10.1017/ATSIP.2014.12

[2] Sharma, Jivitesh & Granmo, Ole-Christoffer & Goodwin, Morten. (2019). Environment Sound Classification using Multiple Feature Channels and Deep Convolutional Neural Networks.

[3] Esmaeilpour, M., Cardinal, P., & Koerich, A. L. (2019). Unsupervised feature learning for environmental sound classification using cycle consistent generative adversarial network. *arXiv preprint arXiv:1904.04221*.

[4] Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... & Jaitly, N. (2018, April). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4774-4778). IEEE.

[5] Tzinis, E., Wisdom, S., Hershey, J. R., Jansen, A., & Ellis, D. P. (2020, May). Improving universal sound separation using sound classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 96-100). IEEE.

[6] Yu, C. Y., Liu, H., & Qi, Z. M. (2017). Sound event detection using deep random forest. In *Detection and Classification of Acoustic Scenes and Events*.

[7] Yang, A. Y., & Cheng, L. (2020). Two-Step Surface Damage Detection Scheme using Convolutional Neural Network and Artificial Neural Neural. *arXiv preprint arXiv:2003.10760*.

[8] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015, July). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (Vol. 8).

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[10] Piczak, K. J. (2015, September). Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1-6). IEEE.

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

[12] Lezhenin, I., Bogach, N., & Pyshkin, E. (2019, September). Urban Sound Classification using Long Short-Term Memory Neural Network. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 57-60). IEEE.

[13] Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015, April). Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4580-4584). IEEE.

[14] Qiao, T., Zhang, S., Zhang, Z., Cao, S., & Xu, S. (2019). Sub-Spectrogram Segmentation for Environmental Sound Classification via Convolutional Recurrent Neural Network and Score Level Fusion. *arXiv preprint arXiv:1908.05863*.

[15] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).

[16] Zhu, B., Xu, K., Wang, D., Zhang, L., Li, B., & Peng, Y. (2018, September). Environmental sound classification based on multi-temporal resolution convolutional neural network combining with multi-level features. In *Pacific Rim Conference on Multimedia* (pp. 528-537). Springer, Cham.

[17] Tokozume, Y., Ushiku, Y., & Harada, T. (2017). Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*.

[18] Salamon, J., Jacoby, C., & Bello, J. P. (2014, November). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 1041-1044).

[19] Ridley, M., & MacQueen, D. (2004). Sampling plan optimization: A data review and sampling frequency evaluation process. *Bioremediation journal*, *8*(3-4), 167-175.

[20] Song, Z., Liu, B., Pang, Y., Hou, C., & Li, X. (2012). An improved Nyquist–Shannon irregular sampling theorem from local averages. *IEEE transactions on information theory*, *58*(9), 6093-6100.

[21] Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, *28*(4), 357-366.

[22] Brown, J. C. (1991). Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, *89*(1), 425-434.

[23] Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, *66*(1), 51-83.

[24] Miiller, M. (2007). Information retrieval for music and motion. *Springer, Berlin Heidelberg*.

[25] N. Davis and K. Suresh, "Environmental Sound Classification Using Deep Convolutional Neural Networks and Data Augmentation," 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, 2018, pp. 41-45, doi: 10.1109/RAICS.2018.8635051.

[26] Sang, J., Park, S., & Lee, J. (2018, September). Convolutional recurrent neural networks for urban sound classification using raw waveforms. In *2018 26th European Signal Processing Conference (EUSIPCO)* (pp. 2444-2448). IEEE.

[27] Yuan, J., & Tian, Y. (2019). An Intelligent Fault Diagnosis Method Using GRU Neural Network towards Sequential Data in Dynamic Processes. *Processes*, *7*(3), 152.

[28] Tokozume, Y., & Harada, T. (2017, March). 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.

[29] Dai, W., Dai, C., Qu, S., Li, J., & Das, S. (2017, March). Very deep convolutional neural networks for raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 421-425). IEEE.

[30] Boddapati, V., Petef, A., Rasmusson, J., & Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia computer science*, *112*, 2048-2056.

[31] Zhang, Z., Xu, S., Cao, S., & Zhang, S. (2018, November). Deep convolutional neural network with mixup for environmental sound classification. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)* (pp. 356-367). Springer, Cham.

[32] Agrawal, D. M., Sailor, H. B., Soni, M. H., & Patil, H. A. (2017, August). Novel TEO-based Gammatone features for environmental sound classification. In *2017 25th European Signal Processing Conference (EUSIPCO)* (pp. 1809-1813). IEEE.

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

[34] Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors*, *19*(7), 1733.

[35] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

[36] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

.