Sound Classification using Neural Network Models

Sanket Sonu — Student Id: x19206071
Sachin Muttappanavar — Student Id: x20144253
Saranya Varshni Roshan Karthikha — Student Id: x20154801
MSc. in Data Analytics
National College of Ireland
URL: www.ncirl.ie

***Abstract*—Recognizing different environmental sounds and segregating them into different categories is quite an easy and fun task for a human brain. But, when it comes to an artificial prediction model to do the same, it takes a lot of complex steps to classify the sound excerpts with the greatest level of accuracy. This study focuses on classifying different environmental sounds like a dog bark, drilling, gunshot, and many more using neural network models like Artificial Neural Network, Convolutional Neural Network 2D, and Recurrent Neural Network. Elaboration on these models will be done in further sections. This experiment showcases different rates like for ANN with 73.06% of accuracy for 10 fold cross-validation, CNN 2D using ReLu with 89.63%,96.24% of test accuracy for test and train accuracy. The Third model is the same CNN 2D performed using eLu that resulted in an accuracy of 76.4%. The Final fourth model LSTM which is used to compare 8 optimizers activation results in % of accuracy.**

***Index Terms*—Sound Classification, ANN, CNN, RNN, LSTM, Spectrogram**

I. Introduction

Infinite frequencies and sound waveforms exist everywhere and it is easily possible to identify every other feature in a sound wave clearly. Every single feature of a sound will be exactly distinguished by any normal human brain. It becomes more intriguing when the same activity could be performed by any smart device or artificial prediction models or deep learning models. These feature selections and identification helps us to interpret every sound form in and around us in this environment. Other complexities and challenges arrive when the accuracy level of this artificial model is expected to mimic as close as to the human level of identification and classification.

The objective of this study is to experiment with different neural network models and build different network methodologies. These models should be capable of extracting different sound features from sound excerpts or soundtracks and identify to which class of sound does the audio signals belong. In this project 4 different deep learning models like an artificial neural network, Convolutional Neural Network two dimensional model with ReLu activation, Convolutional Neural Network two dimensional model with eLu activation, and Long Short-Term Memory which is used to compare 8 optimizers activation.

Motivation for this experiment or project was inspired due to several significant applications of sound classifications like : The use of security systems will help in identifying any emergency situations like scenarios with broken glass or an alarm for emergency fire. These classification algorithms take the main role in recommendation algorithm systems. Whereas, application in other fields like speech-to-text translations, language conversions, and human speech classification is also predominant.

To elaborate the overview flow of this project, initially, the soundtracks are interpreted differently for feature extraction according to different deep learning models. Librosa library is used to extract such sound features from the audio signal and further, used in the input for deep learning models. These features are first in the form of spectrogram and later turned into data values for analysis.

The dataset that is considered for this project is from the Kaggle repository that contains different environmental sounds like children playing, gunshot, siren, street music and, many more.

In order to understand more on the deep learning models, around 20 literature were critically reviews and different methodologies of analysis were adopted from those research papers.

Research question of this project is : How accurately can we predict and classify the audio tracts based on the features using neural network model.

The upcoming sections will illustrate information regarding, dataset source, attribute information, methodology, different stages in the analysis, critical evaluation section of the results obtained, and finally the summary and future work related to this project.

II. LITERATURE REVIEW

Authors (Sang, Park, and Lee 2018) have proposed CRNN methodology that utilizes waveforms of the time domain as the only input. A hybrid model that involves convolutional neural networks and recurrent neural networks. CNN is chosen for different sound feature extraction and RNN is for the extracted features to be aggregated. The Classified Sound is passed as input into ESC. The proposed model outperforms the deep network with 7.38% of accuracy on absolute.

The author (Piczak 2015) proposed a model that comprises layers of the convolution along with max-pooling and training with two fully connected layers. This model outperforms with coefficients of Mel-frequency cepstral. Results show that CNN could opt for less complex data augmentation.

Author (Huang et al. 2020) proposed CNN with 2-order dense which utilizes dual features. This new model has the ability to increase the convergence velocity while compared to other dense networks. This also helped in improving the classification accuracy rate. The results depict accuracy with 84.83% and also 85.17%. These accuracies have significant enhancement when compared to baseline

This paper (Shu, Song, and Zhou 2018) showcases the experiment with the resolution of the time-frequency of the sound inputs. In spite of a careful model design, results show that data augmentation has little impact on the performance of neural network design. It is also observed that the length of segmentation on the input signal window has a significant impact on output. With the improvement or increase in the segmentation window length parameter, a steady increase in classification accuracy leading to soon saturation.

The authors (Zhang et al. 2018) proposes CNN for ESC activities. This model combines convolutional that are stacked with pooling layers to retrieve feature presentations from different features that represent spectrogram. Different trails experimented on datasets like ESC-50 and other ESC-10. This methodology results show the achievement of 83.7% of performance accuracy with a dataset on ESCs.

This study (Rahmandani, Nugroho, and Setiawan 2018) focuses on the diagnosis of identifying the abnormalities in any human heartbeat sound waves. A method like MFCC was used in the extraction of different sound wave features for the input of analysis. With the use of the Artificial Neural Network method, this study results in up to 100% of the performance rate outperforming other models.

This research (Khamparia et al. 2019) aims to classify the ESC based on the different spectrogram images using neural networks. The model built was trained with the environmental sound wave spectrogram images. Both CNN and TDSN models were trained with the same train dataset. Two datasets like ESC-10 and ESC-50 were used in this model. Accuracies like 77% for CNN and 56% for TDSN were observed.

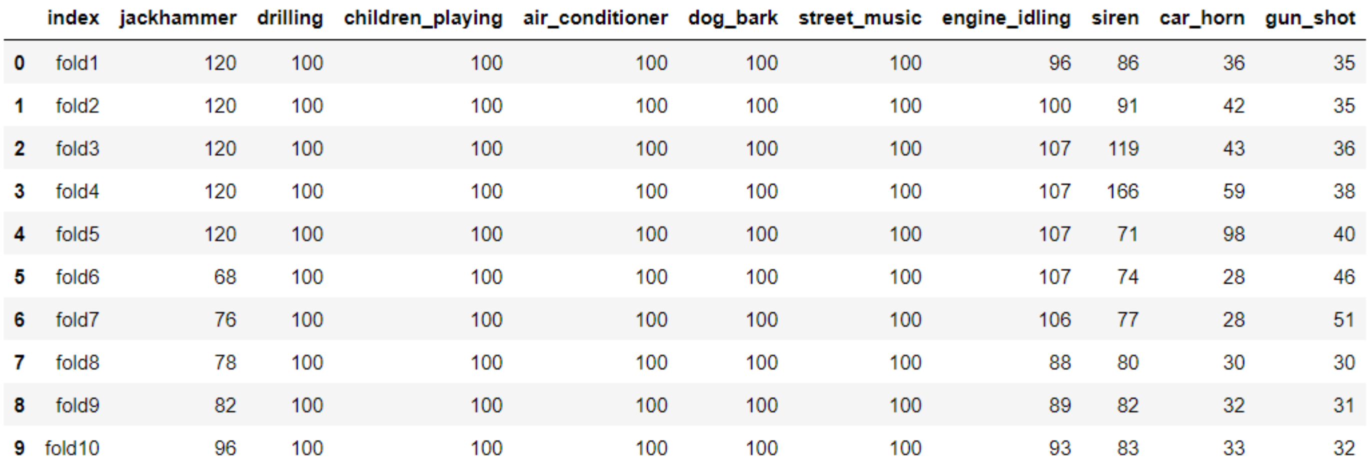
In this study (Demir, Abdullah, and Sengur 2020) extracted deep features are used for the ESC sound classification problem. These deep features are extracted by fully connected layers of a newly created CNN model trained by spectrogram images. Each feature vector comprises fully connected layers that are concatenated together. The results convey that the proposed CNN model records accuracies as 96.23% and 86.70%.

In this paper (Shuiping, Zhenming, and Shiqiang 2011), they have done an investigation on the Convolutional neural network and studied the impact of each parameter on the behavior of the network. As per their study Convolutional layer is a critical layer in the Convolutional neural network. Also, the layers count within the network also determines the performance of the network. Contradicting, the computation time for train and test increases as the number of layers increases. They also outlined the applications of CNN.

In another work (Albawi, Mohammed, and Al-Zawi 2017), different audio features like spectrogram, harmonic, percussion are extracted. The convolutional neural network model is trained on these extracted features. They also trained the pre-trained models like AlexNet, GoogleNet, Vgg-16, Vgg-19 over these features. Their study depicts that the ensemble of various fine-tuned CNNs improves the ability to classify the animal audio sound. Even their proposed model outperformed the state-of-the-art approaches. As future work, they suggested working with different audio classification tasks and fine tunning the parameters of the CNN model.

In research work (Wu, Mao, and Yi 2018), they have developed the CNN model over the spectrograms of audio files for classifications of sounds. Spectrograms are the visuals of the range of frequencies of audio that differ with time. Their model consists of one input and output layer, two hidden layers. Relu activation function is applied to every layer in the network. To update the weights of the network adam optimizer technique is used. The model ran for 25 epochs and to find the accuracy of the model accuracy metric is used. Model parameters are tuned through the Random Search CV algorithm. The model performed well with an accuracy of 85%.

Classifier-Attention-Based Convolutional Neural Network (CAB-CNN)is developed to classify the audio classification problem (Scarpiniti et al. 2021). The number of parameters needed by the model is reduced through CAB-CNN thus, model complexities. Hence, it was easy to train the CAB-CNN



	index	jackhammer	drilling	children_playing	air_conditioner	dog_bark	street_music	engine_idling	siren	car_horn	gun_shot
0	fold1	120	100	100	100	100	100	96	86	36	35
1	fold2	120	100	100	100	100	100	100	91	42	35
2	fold3	120	100	100	100	100	100	107	119	43	36
3	fold4	120	100	100	100	100	100	107	166	59	38
4	fold5	120	100	100	100	100	100	107	71	98	40
5	fold6	68	100	100	100	100	100	107	74	28	46
6	fold7	76	100	100	100	100	100	106	77	28	51
7	fold8	78	100	100	100	100	100	88	80	30	30
8	fold9	82	100	100	100	100	100	89	82	32	31
9	fold10	96	100	100	100	100	100	93	83	33	32

Fig. 1. Class Distribution of each fold

model and high performance was achieved. It is demonstrated that the CAB-CNN model produced 10% more accurate results compared to state-of-the-art algorithms.

A Deep Belief Network(DBN) model is developed to classify the audio files related to equipment and construction operations (Haoye Lu 2020). This model performed well on the trained data and the same is tested over the audio files of the vehicles and tools. Model is able to identify the classes with accuracy up to 98%. The model outperformed the existing machine/deep learning algorithms in the classification of construction site's audio files. Their proposed model is even proposed to use in the classification of the audio files of a variety of environmental scenarios.

Another research is done by (Jaiswal and Kalpeshbhai Patel 2018) in classifying the audio files based on the speakers' identities, accents, and emotional states. It was challenging to extract the features from the data because of high dimensionality. They first represented the source file in the spectrogram and later features are extracted from the spectrogram. They proposed Frequency Convolutional Network which is a task-independent model. Also, attention mechanisms are incorporated into the model to systematically improve the features from specific frequency bands. They evaluated their model over the three publically available speech databases. The model outperformed the state-of-the-art.

Support vector machines(SVM) are also implemented by (Nanni et al. 2020) to classify the audio files. Time and frequency domain features are fetched from the source audio files. SVM model is trained over the extracted feature set and model evaluated for its performance. The model produced favorable results with an accuracy of 90%.

(Raguraman, Mohan, and Vijayan 2019) and (Vatolkin, Ginsel, and Rudolph 2021) have proposed models for proving solution in cases where the end users play music notes on musical instrument and evaluate the same for the performance check on benchmark music excerpts. These techniques uses loudness, tempo, rollover skewness and other associated features that are in relation to the music instrument.

III. Dataset

This project is using UrbanSound8K dataset, which is freely available on Kaggle: UrbanSound8K. This dataset has 8732 audio excerpts of urban sound in WAV format. The audio belongs to 10 classes, which are as follows:

1) air_conditioner
2) car_horn
3) children_playing
4) dog_bark
5) drilling
6) engine_idling
7) gun_shot
8) jackhammer
9) siren
10) street_music

The dataset has a CSV file which is metadata of audio excerpts. The dataset contains files that are divided into 10 folds (folder names as fold 1 to 10). (Figure 1) shows the table which explains the class distribution of each fold.

IV. METHODOLOGY & EVALUATION

This project has used ANN, CNN, and RNN models for the classification of audio excerpts. Deep Learning models learn from the images. The challenge was to convert audio files into images. This research paper has used the

‘Librosa’ library to create spectrograms of audio files. This process is called ’Features Extraction’. These features will be used by Deep Learning models for classification.

A. Features Extraction

Frequencies of sound and few other signals vary with time, and this can be visually represented using Spectrogram. (Patil and Nemade 2019) Mel-Frequency Cepstral Coefficients (MFCCs) has been used for feature extraction. Log-Mel spectrograms are the most popular for feature extraction using characteristics of the Mel scale. Mel scale has a perceptual pattern, which maps linear frequency to Mel frequency.

The Log-Mel spectrograms take perceptual patterns on the magnitude axis, and magnitudes can be expressed using the logarithmic axis. It also takes another dimension as a frequency axis.

Deep Learning models cannot understand the audio files directly. The features need to be extracted from the audio file, which can be used by models to understand. Each audio file 3-dimensional signals, which has axis such frequency, amplitude, and time. librosa.load(audiofile) will decode the audio file into 1-dimensional array. This array is time series x and the sampling rate is assigned for x. By default, sampling rate(sr) is 22kHz. The value of n_mfcc is 40 during the feature extraction.

librosa.display can be used to plot the audio plots based on the frequency with time. It shows multiple frequencies playing at different times along with their amplitude. (Figure 2) shows wave-plot of 1 particular audio from each class. (Figure 3) shows the spectrogram of 1 particular audio from each class.

librosa.display can be used to display wave-plots. (Figure 4) shows the wave-plot of dog barking and (Figure 5) shows wave-plot of children playing.

librosa.display can be used to display spectrogram.(Figure 6) shows the wave-plot of dog barking and (Figure 7) shows wave-plot of children playing.

A data-frame will be created with the extracted features and every row will be labelled with the corresponding class. This data-frame will be used further to create train and test sets, which will be used by deep learning models for prediction.

B. Artificial Neural Network (ANN)

The artificial neural network (ANN) is the first model used for this research paper. A data-frame is created using features extracted, which will be used for creating train and test sets. There are 2 ways for creating train and test sets. One is the normal train_test_split, which will be used for CNN and LSTM models. ANN model will use folds for creating train and test sets. There are 10 folds, in which all data are split. Each fold will be used as a test set once and the rest 9 folds will be used as a train set. For example, fold 1 will be used for the test set and 2 to 10 folds will be used for a train set, then fold 2 will be used for the test set and 1,3 to 10 folds will be used for a train set so on. The model will be fitted with each dataset 10 times and in the end, the mean of 10 fold cross-validation score will be computed as a final result.

Model contains below mentioned layers:

- **Input Layer:** The first layer is the input layer, which contains the spectrogram images, that is processed through the model. The first layer is of node 512, with an input shape of (40,) because during feature extraction, n_mfcc was 40. The activation function used is ’ReLu’ and the dropout is 0.5. This will ignore few neurons which are chosen randomly at the training phase.
- **Two Hidden Layers:** There are 2 sub-layers in hidden layers. The first, hidden layer is node 256. Activation function is ’ReLu’ and dropout is 0.5. The second hidden layer is just node 128. ’ReLu’ activation functions are used which will help in tackling the vanishing gradient problem. The negative part of the argument will be removed by using the ’ReLu’ activation function.

  f ReLU(x)=max(0, x)
- **Output Layer:** The final layer is the output layer. The activation function used is ’SoftMax’. The output layer will be used to generate the output based on the number of labels, i.e., 10. SoftMax will give a probability of each class that will sum up equal to 1.

The optimizer used was ’Adam’. Adam optimizing algorithm is an extension of stochastic gradient descent, which is used to update the weights of the network in the training data. In Stochastic gradient descent, a single learning rate is used for all weights, whereas, Adam will maintain a different learning rate for each network weight.

The training process used 10 epochs. In the end, the mean of 10 fold cross-validation score will be computed to achieve the final result.

Accuracy for 10 fold cross validation: 73.60 %

(Figure 8) shows the plot which contain loss of train and test sets. (Figure 9) shows the plot which contains accuracy of train and test sets. The observed accuracy is less and loss is more for test sets. This can be improved by changing layers and models.

C. Convolutional Neural Network (CNN 2D) - ReLu

The second model is Convolutional Neural Network (CNN). This is a 2-dimensional convolutional neural network model using the ’ReLu’ activation function.(Shamsaldin et al. 2019) After feature extraction, data-frame has been created. This

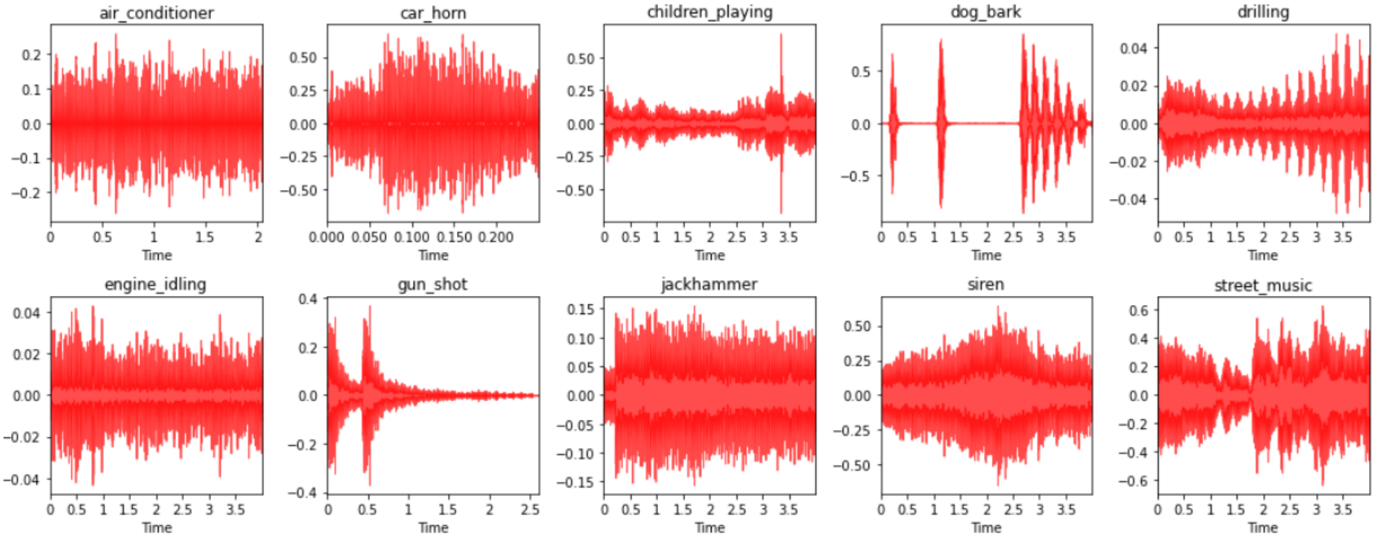


Fig. 2. Waveplot of audio files

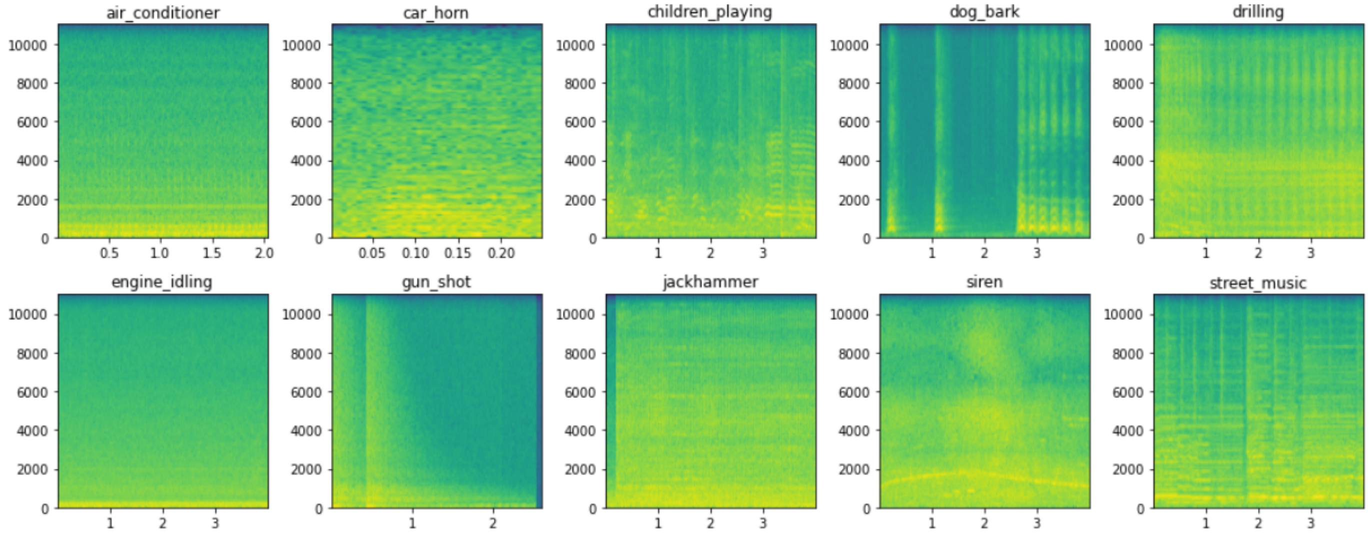


Fig. 3. Spectrogram of audio files

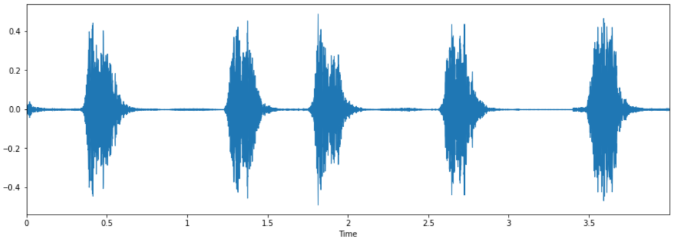


Fig. 4. Waveplot of dog barking

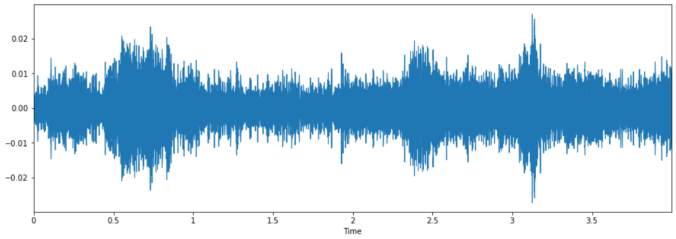


Fig. 5. Waveplot of children playing

data-frame has been split into train and test set using sklearn's train_test_split. Test size of 20 % and train set of 80 % size have been created. Layers are added as follows:

- **Input Layer:** The first layer is an input layer, which has

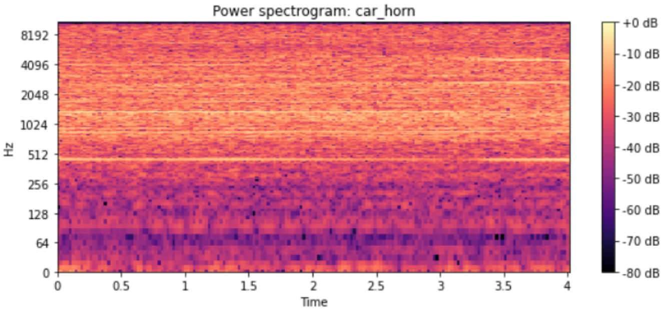


Fig. 6. Spectrogram of car horn

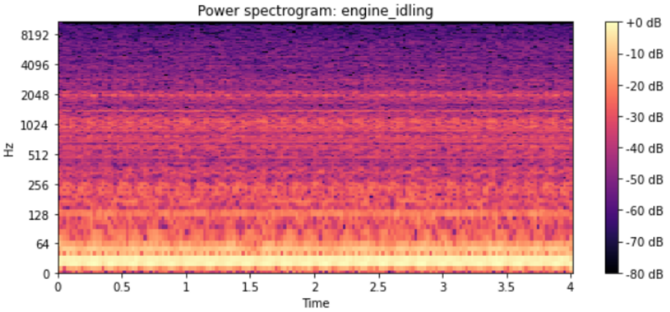


Fig. 7. Spectrogram of engine idling

a filter = 16, the size of the kernel is 2, and the activation function is 'ReLu'. MaxPooling2D has been added with pool_size of (2,2) and dropout has been set as 0.2. The negative part of the argument will be removed by using the 'ReLu' activation function by tackling with vanishing gradient problem.

- **Three Hidden Layers:** Hidden layers contain a total of 3 layers of filters 32, 64, and 128. All have the same activation function 'ReLu' with dropout as 0.2 and MaxPooling2D as (2,2). MaxPooling2D will reduce the dimension of the image by retaining important information. GlobalAveragePooling2D will be used as the next sub-layer, this will be used to apply pooling on a spatial dimension, which is average pooling. A flatten sub-layer is used after the last hidden layer, which will convert the image into the one-dimension image, which will be used as input for the next layer.

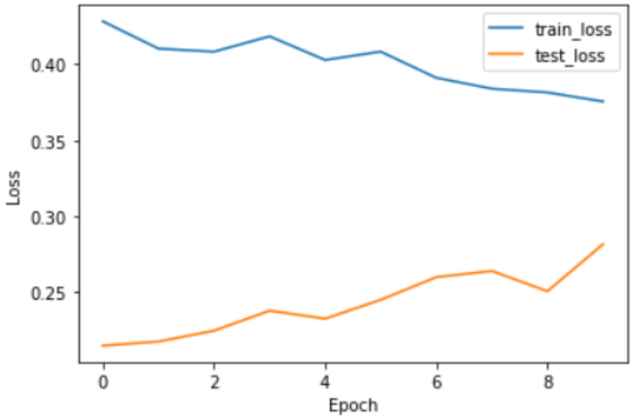


Fig. 8. ANN Model - Loss

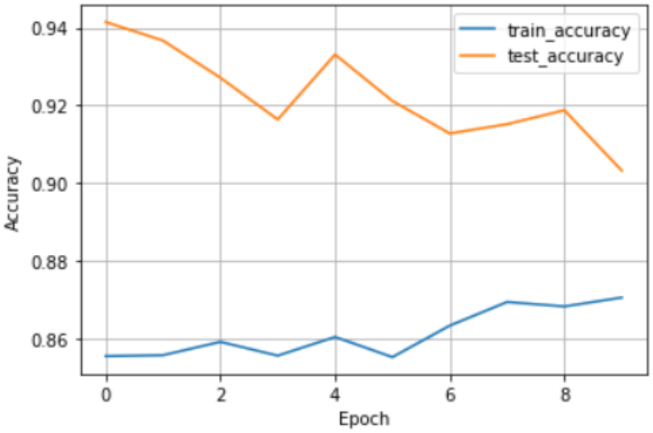


Fig. 9. ANN Model - Accuracy

- **Output Layer:** The final layer is the output layer. The activation function used is 'SoftMax'. The output layer will be used to generate the output based on the number of labels, i.e., 10. 'SoftMax' will give the probability of each class that will sum up equal to 1.

The optimizer used was 'Adam'. Adam optimizing algorithm is an extension of stochastic gradient descent, which is used to update the weights of the network in the training data.

Loss has been selected as 'categorical_crossentropy'and metrics used is 'accuracy'. The batch size used is 256 and epochs is 100 at the time of fitting the model. CNN 2D model provided very good results when compared to the ANN model.

Training Accuracy: 97.26 %

Testing Accuracy: 91.12 %

(Figure 10) shows the loss of CNN 2D ReLu model and it can be observed that loss of both test and train sets are decreasing after each epochs. (Figure 11) shows the accuracy of CNN 2D ReLu model and it can be observed that accuracy of both test and train sets are increasing after each epochs.

D. Convolutional Neural Network (CNN 2D) - ELU

The third model is Convolutional Neural Network (CNN 2D) using the 'ELU' activation function. Data-frame will be created from the extracted feature, which will be split into

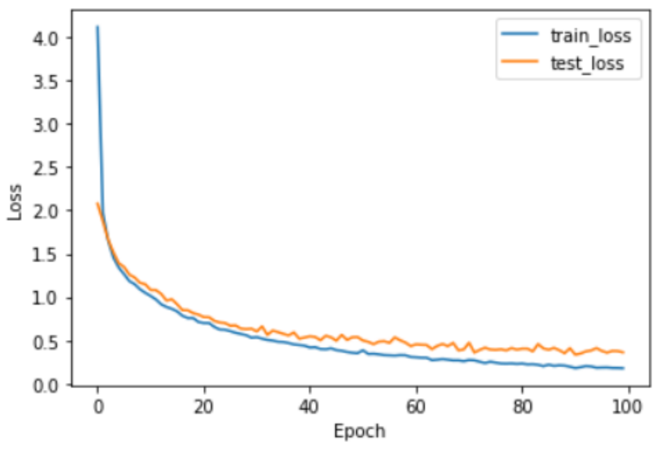


Fig. 10. CNN ReLu Model - Loss

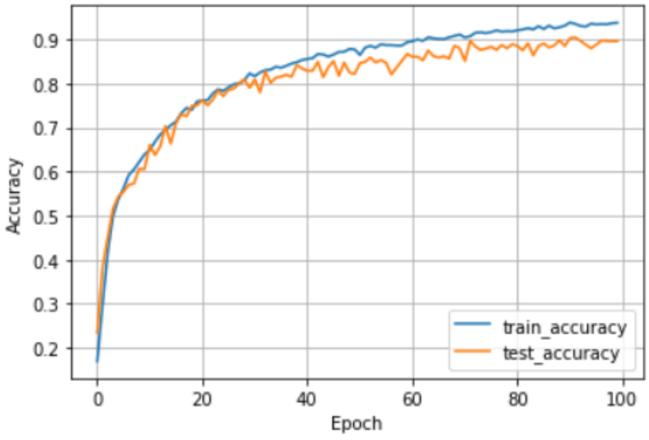


Fig. 11. CNN ReLu Model - Accuracy

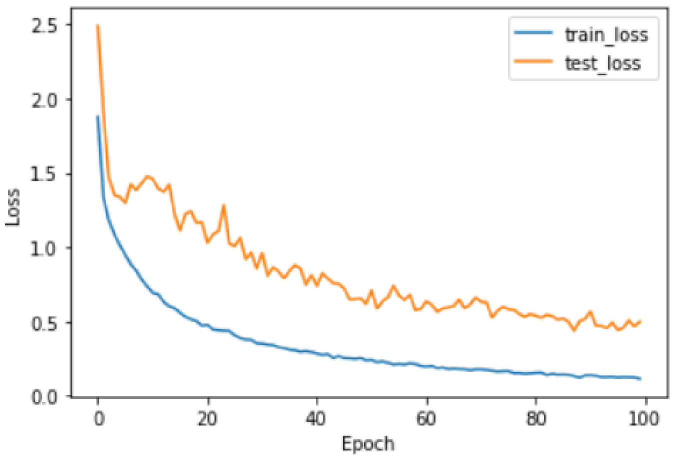


Fig. 12. CNN ELU Model - Loss

train and test sets of 80 % and 20 % respectively. Layers are added as follows:

- **Input Layer:** The first layer is an input layer, which has a filter = 16, the size of the kernel is 2, and the activation function is 'ELU'. 'ELU' has negative values, which mean unit activation moves close to zero, and computational complexity is very low. A sub-layer of Batch Normalization (BN) has been implemented, which will help in improving training time and makes the model more stable by reducing interdependence between hidden layers. BN helps in reducing the gradient vanishing problems as well. BN can help in utilizing larger learning rates. MaxPooling2D has been added with pool_size of (2,2) and dropout has been set as 0.2. Dropout will ignore few neurons which are chosen randomly at the training phase.
- **Three Hidden Layers:** Hidden layers contain a total of 3 layers of filters 32, 64, and 128. All have the same activation function 'ELU' with dropout as 0.2 and MaxPooling2D as (2,2). MaxPooling2D will reduce the dimension of the image by retaining important information. Batch Normalization (BN) has been applied after two hidden layers to reduce the training time. GlobalAveragePooling2D will be used after the third hidden layer as a sub-layer, this will be used to apply pooling on a spatial dimension, which is average pooling. A flatten sub-layer is used after the last hidden layer, which will convert the image into a one-dimension image, which will be used as input for the next layer.
- **Output Layer:** The final layer is the output layer. The activation function used is 'Softmax'. The last layer of the model will take the number of labels to generate output. 'SoftMax' will give the probability ofeach class that will sum up equal to 1.

Again, (Indolia et al. 2018) 'adam' optimizer has been used which will help in updating the network's weight in the training phase. Loss has been selected as 'categorical_crossentropy'and metrics used is 'accuracy'. The number of epochs used is 100 and the batch size is set to 256. This CNN ELU model provides a better result than the CNN ReLu model, mostly because of the use of Batch Normalization and the 'SoftMax' activation function. Results are as follows:

Training Accuracy: 94.45 %

Testing Accuracy: 87.75 %

(Figure 12) shows the loss of CNN 2D_ELU model. Plots demonstrates that loss of both test and train sets are decreasing after each epochs. (Figure 13) shows the accuracy of CNN 2D_ELU model and it demonstrates that accuracy of both test and train sets are increasing after each epochs.

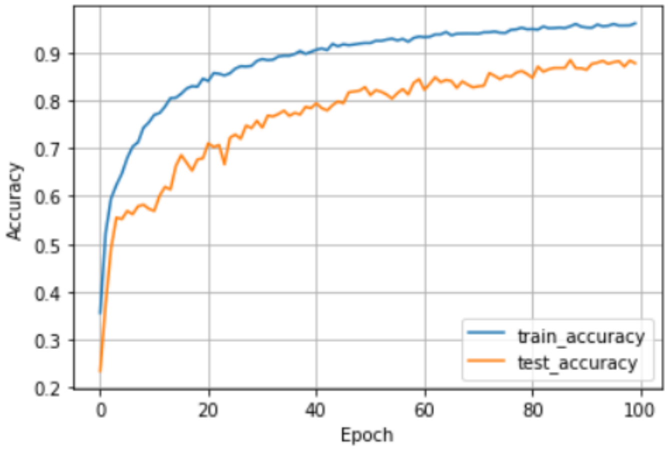


Fig. 13. CNN ELU Model - Accuracy

E. Long Short Term Memory (LSTM)

The final model used for this research paper is the Long short-term memory (LSTM) model. LSTM is a unique type of Recurrent Neural Network (RNN). Unlike CNN, LSTM does not suffer from the problem of vanishing gradient. LSTM has subnets, which is also known as a memory block. Every block has a memory cell with three units, such as input, output, and forget gates. A dataset is created with an extracted feature, n_mfcc was set to 40 during feature extraction. This dataset was split into test and train sets of 15 % and 85 % respectively. The final LSTM model will be used to compare the optimizers. 8 optimizers with different learning rates have been used for this LSTM model, which is as follows:

1) Adam - learning_rate = 0.001
2) SGD - learning_rate = 0.01
3) Nadam - learning_rate = 0.001
4) RMSprop - learning_rate = 0.001
5) AdaBelief - learning_rate = 0.001
6) RAdam - learning_rate = 0.001
7) AdaBound - learning_rate = 0.001, final_lr = 0.1
8) Yogi - learning_rate = 0.001

The model is implemented with input, hidden, and output layers, which are as follows:

- **Input Layer:** The first layer is the input layer. The node is 40 and the input shape is (1,40). This is because, at the time of feature extraction, n_mfcc = 40. The return sequence has been set to True. For every input time step, the return sequence will return the hidden state output.
- **Hidden Layers:** There are two hidden layers. The activation function used is 'ReLu', which will help in removing negative parts of the argument. The dropout is set as 0.2. This will ignore few neurons which are chosen randomly at the training phase.

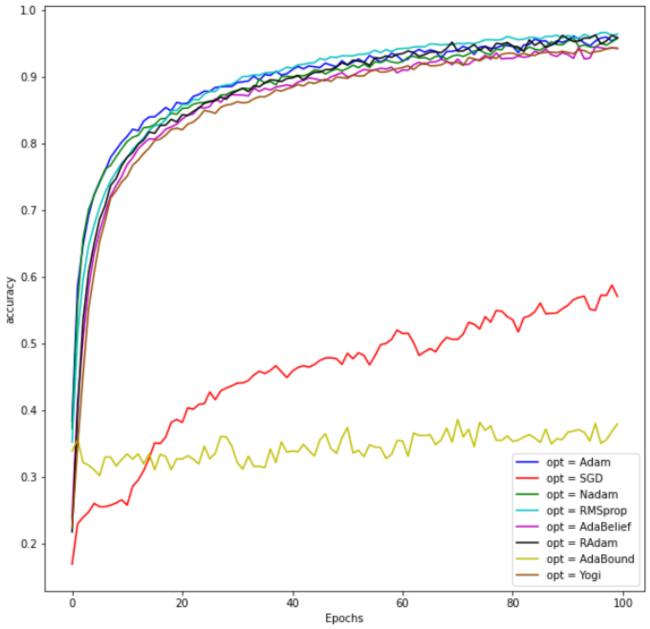


Fig. 14. LSTM Training - Accuracy

- **Output Layer:** The last dense layer is added with 10 as num_labesls, which will help in generating the final output. The activation function used here is 'SoftMax'. 'SoftMax' will give the probability of each class that will sum up equal to 1.

The batch size is set to 32 and epochs = 100. (Figures 14, 15, 16, and 17) demonstrates that 'SGD' and 'AdaBound' are not showing good results and rest 6 optimizers are performing good for this dataset. Overall, LSTM produced: **Train Accuracy: 94 %** and **Testing Accuracy: 86 %**.

V. Conclusion and Future Work

Conclusively, deep learning models are not able to understand the audio files directly. This problem was solved by extracting features from all the audio files. Librosa library has been used to extract the features and a new data-frame was created with those extracted features. This data-frame was used by 4 deep learning models. The train and test sets are created using 2 ways, ANN model is using 1 fold for test and the rest 9 folds for training, the model will be fit multiple times so that all folds will come under the test set once and the remaining 9 sets will be used for training sets. ANN didn't achieve a good result on this data-set and was able to get 73 to 80 % accuracy. Next, the normal train_test_split has been used to split the data into train and test sets. These sets are used by 3 models, which are CNN 2D - ReLu model, CNN 2D - ELU model,

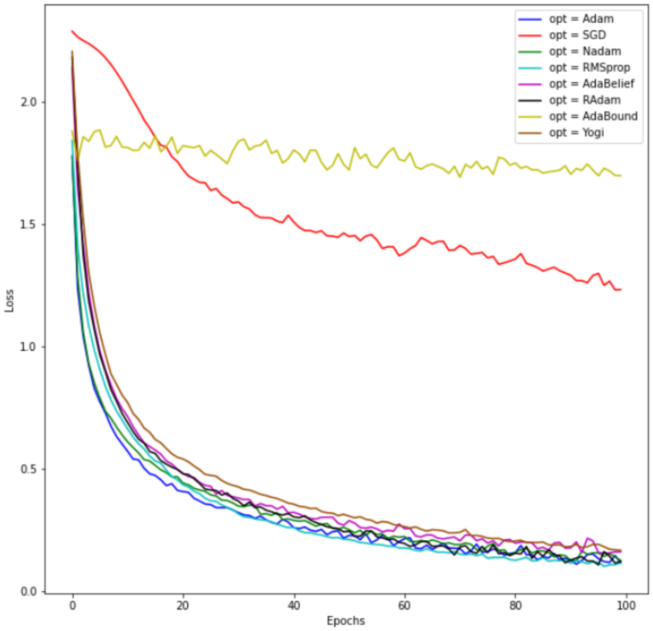


Fig. 15. LSTM Training - Loss

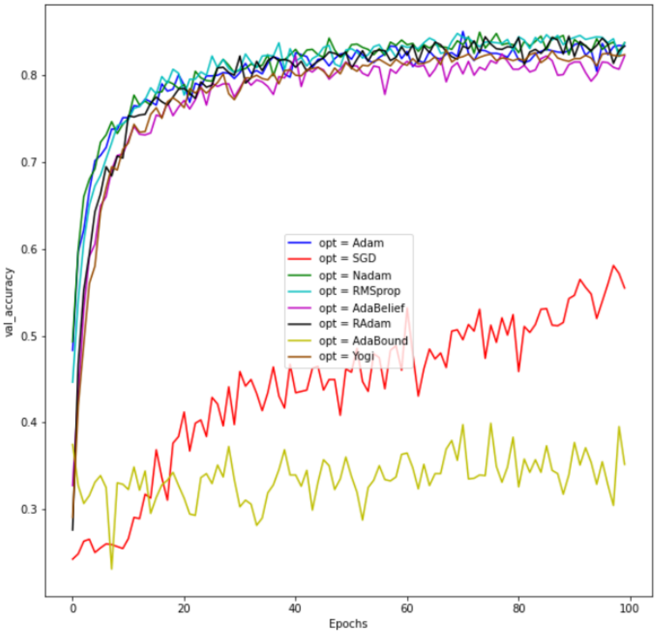


Fig. 16. LSTM Testing - Accuracy

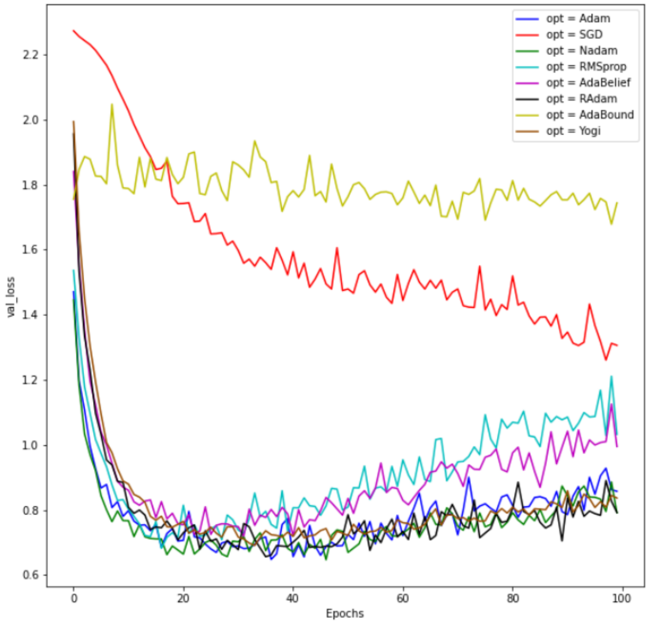


Fig. 17. LSTM Testing - Loss

and LSTM model. Both CNN models generate good results, CNN 2D - ReLu model produced a slightly better result with 97 % accuracy on training and 91 % accuracy on the test set, which is 3-4 % more than CNN 2D - ReLu model. The LSTM model produced on average 94 % accuracy for train and 85 % accuracy for test sets. Overall, CNN 2D models perform best for the audio dataset, which produced better accuracy with less error rate.

In future, multiple layers can be implemented with different parameters and to find the high, medium, and low loudness levels. Other methods of feature extraction can also be used.

References

Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi (2017). “Understanding of a convolutional neural network”. In: *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

Demir, Fatih, Daban Abdulsalam Abdullah, and Abdulkadir Sengur (2020). “A New Deep CNN Model for Environmental Sound Classification”. In: *IEEE Access* 8, pp. 66529–66537. DOI: 10.1109/ACCESS.2020.2984903.

Haoye Lu Haolong Zhang, Amit Nayak (2020). “A Deep Neural Network for Audio Classification with a Classifier Attention Mechanism”. In: *Expert Systems with Applications*.

Huang, Zilong et al. (2020). “Urban sound classification based on 2-order dense convolutional network using dual features”. In: *Applied Acoustics* 164, p. 107243. ISSN: 0003-682X. DOI: https://doi.org/10.1016/j.apacoust.2020.107243.

Indolia, Sakshi et al. (2018). “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach”. In: *Procedia Computer Science* 132. International Conference on Computational Intelligence and Data Science, pp. 679–688. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.069. URL: https://www.sciencedirect.com/science/article/pii/S1877050918308019.

Jaiswal, Kaustumbh and Dhairya Kalpeshbhai Patel (2018). “Sound Classification Using Convolutional Neural Networks”. In: *2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pp. 81–84. DOI: 10.1109/CCEM.2018.00021.

Khamparia, Aditya et al. (2019). “Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network”. In: *IEEE Access* 7, pp. 7717–7727. DOI: 10.1109/ACCESS.2018.2888882.

Nanni, Loris et al. (2020). “Ensemble of convolutional neural networks to improve animal audio classification”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2020, pp. 1–14.

Patil, Nilesh and Milind Nemade (May 2019). “Content-Based Audio Classification and Retrieval Using Segmentation, Feature Extraction and Neural Network Approach”. In: pp. 263–281. ISBN: 978-981-13-6860-8. DOI: 10.1007/978-981-13-6861-5_23.

Piczak, Karol J. (2015). “Environmental sound classification with convolutional neural networks”. In: *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. DOI: 10.1109/MLSP.2015.7324337.

Raguraman, Preeth, R Mohan, and Midhula Vijayan (2019). “Librosa based assessment tool for music information retrieval systems”. In: *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, pp. 109–114.

Rahmandani, Muhammad, Hanung Adi Nugroho, and Noor Akhmad Setiawan (2018). “Cardiac Sound Classification Using Mel-Frequency Cepstral Coefficients (MFCC) and Artificial Neural Network (ANN)”. In: *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, pp. 22–26. DOI: 10.1109/ICITISEE.2018.8721007.

Sang, Jonghee, Soomyung Park, and Junwoo Lee (2018). “Convolutional Recurrent Neural Networks for Urban Sound Classification Using Raw Waveforms”. In: *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2444–2448. DOI: 10.23919/EUSIPCO.2018.8553247.

Scarpiniti, Michele et al. (2021). “Deep Belief Network based audio classification for construction sites monitoring”. In: *Expert Systems with Applications* 177, p. 114839. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.114839.

Shamsaldin, Ahmed et al. (Dec. 2019). “The Study of The Convolutional Neural Networks Applications”. In: *UKH Journal of Science and Engineering* 3, pp. 31–40. DOI: 10.25079/ukhjse.v3n2y2019.pp31-40.

Shu, Haiyan, Ying Song, and Huan Zhou (2018). “Time-frequency Performance Study on Urban Sound Classification with Convolutional Neural Network”. In: *TENCON 2018 - 2018 IEEE Region 10 Conference*, pp. 1713–1717. DOI: 10.1109/TENCON.2018.8650428.

Shuiping, Wang, Tang Zhenming, and Li Shiqiang (2011). “Design and Implementation of an Audio Classification System Based on SVM”. In: *Procedia Engineering* 15. CEIS 2011, pp. 4031–4035. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2011.08.756.

Vatolkin, Igor, Philipp Ginsel, and Günter Rudolph (2021). “Advancements in the Music Information Retrieval Framework AMUSE over the Last Decade”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2383–2389.

Wu, Yu, Hua Mao, and Zhang Yi (2018). “Audio classification using attention-augmented convolutional neural network”. In: *Knowledge-Based Systems* 161, pp. 90–100. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2018.07.033.

Zhang, Zhichao et al. (2018). “Deep Convolutional Neural Network with Mixup for Environmental Sound Classification”. In: *Pattern Recognition and Computer Vision*. Springer International Publishing, pp. 356–367. DOI: 10.1007/978-3-030-03335-4_31.