iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VI **Month of publication:** June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6272

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

Urban Sound Classification using Neural Networks

By Mridul Aggarwal[1], Ayushi Rai[1], Mrs. Parul Yadav[2]
[1, 2]*Student,* [3]*Assistant Professor, Department of Information and Technology, Bharati Vidyapeeth's College of Engineering, New Delhi*

Abstract: ***We are surrounded by sounds, we hear various types of sounds on a day to day basis whether it is music sound, different noises, etc. The urban life is filled with such sounds, which makes it important and highly useful for us to work on these sounds and get some useful information from it so that we can use it efficiently. These sounds are continuously processed by human minds to decipher information about the environment. The same can be done by a machine learning model. It has been seen that convolutional neural networks have been really successful in classifying images, so it becomes a question of interest that how good do they work with different sounds. In this paper, we have worked upon using different deep learning models to see which can be used for the purpose of sound classification. We have used the Urbansound8K dataset which contains 8732 sound excerpts of urban sounds from 10 classes.***
Index Terms: ***[Deep Learning, Convolutional Neural Network (CNN), Sound Classification, MFCC, Artificial Neural Network]***

I. INTRODUCTION

Sound recognition is one topic which can be found everywhere. It covers various variety of fields, including but not limited to Automatic Speech Recognition (ASR) [1] and music information retrieval (MIR) [2]. A lot of work has been done on these topics. Environment Sound Classification is relatively less worked upon. It has various applications like home security surveillance, content-based multimedia indexing and retrieval, helping deaf people in carrying out their day to day tasks, maintenance of machines in industries. Environment Sound Classification is more varying and it is not defined as well as ASR or MIR, so it is a bit complex to work on environment sounds. MFCC, LPC, LPP are different ASR techniques that have been applied to environment sound classification. But the best performance was achieved by using Mel lter bank features, Gammatone features and wavelet-based features. Traditional machine learning approaches are used to model this data like SVM, GMM and KNN. But still the performance gain is not satisfactory, one main reason for this is these algorithms lack feature extraction abilities. We have used the Urbansound8K [5] dataset which contains 8732 sound excerpts of urban sounds from 10 classes.

During the past few years, DNN (Deep Neural Network) has been highly successful in ASR and MIR fields. In our approach we will be using ANN (Artificial Neural Network) and Convolutional Neural Network (CNN), the important feature of deep learning is it can extract features to give more accurate predictions. Here, we will find the performance of convolutional neural networks in classifying short audio clips. Convolutional Neural Network are as old as 1980s but they have been coming to use very recently, the prime reason for this they require high computation power which was not available before and became available with the recent invention of multi-core processors. Now the CNNs are prominently used for various classifications such as traffic signs, house numbers, and handwritten digits [4], pedestrian detection [6], and electron microscopy image processing [10]. It is used majorly in images classification but it has also been used in audio classification for speech and music. But for environment sound classification the use of convolution neural network has been rarely used and it is mostly used on highly pre-processed acoustic features. So, it poses an interesting question that can these be used to classify environmental sounds and we try to find the answer through this paper.

II. RELATED WORK

We will look at different recent deep learning methods to classify environment sounds. Piczak made a feature of two-channel by using log mel spectogram and its delta information, he gave this data to the CNN model as input. Dharmesh et al. [8] made use of gammatone spectrogram with the CNN architecture like that of Piczak. They claimed accuracy levels quite high. But we will not compare our results with theirs as they did not used the right official cross validation method and also the results they achieved were on different training and validation datasets than mentioned in the paper. Some researchers worked on raw data and used 1-D convolution layer with 34 layers on 1-D raw data, Dai et al. used this method and was able to achieve comparable accuracies with the CNN used on mel frequency spectrum. Tokuzome et al. [9] also used raw 1-D on an end-to-end network, EnvNet they were able to find discriminative feature which helped in achieving good accuracy. Some researchers preferred to use external data for training the model. They made use of transfer learning by making the model for sounds on the web and then adapting it to the environmental sounds. Some others like Aytar et al. have used models that are trained for visual recognition and tried to adapt it for the use of environment sound classification.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

III. CONVOLUTION NEURAL NETWORKS

Convolution Neural Network is very similar to Artificial Neural Network i.e., Multi-Layer Perceptron, it is extended version which is most prominently used for its unique feature extraction for image classification.

A. Layer Architecture

Usually a Convolutional Neural Network consists of an input layer, some number of convolution layers coupled with pooling layers, then some hidden layers and finally an output layer. Each unit of a convolution layer works on a small piece of input space instead of whole input space. Then it creates kernel with weights which is used all over the input data to create feature maps. The use of doing this is to make use of the hidden internal connections in the 2-D data which is mostly images but not necessarily. These feature maps are then forwarded to the next layer. Pooling layers like max pooling, average pooling is used to reduce the dimensionality of the data for faster processing, it extracts the essential information of the data while reducing its size.

B. Rectified Linear Units

Nowadays ReLUs are used as non-linear activation function and it has replaced the earlier ones of logistic sigmoid and hyperbolic tangent functions. It is due to ReLUs simplicity that it is preferred over these functions, ReLUs always gives a non-negative value. ReLUs still maintain its discriminatory quality.

$$f(x) = \max(0, x)$$

There is one disadvantage of ReLUs that when the weights are randomly initialized, it may make many of the units as dead as their output would be zero. For this reason, there is Leaky Rectified Linear Units [7].

C. Dropout Learning

Usually Deep Neural Networks tend to overfit to the training data, same is the case with Convolution Neural Networks. To overcome this overfitting on training data we use dropout learning [3]. During each iteration of training process, every unit is deactivated with some predefined probability which is 50% by default. This makes the units more independent of other units. It is a very simple method and yet highly effective.

IV. METHODS

A. Artificial Neural Networks (ANN)

For the process of feature extraction, we used Mel-Frequency Cepstral Coefficients (MFCC), for each audio sample, 40 MFCCs were calculated on a per frame basis with a window size of few milliseconds. The MFCCs signifies both the frequency characteristics and also the time characteristics of the sound. This allows us to extract features of the sound. Then we have to train a Deep Neural Network (DNN) model, we will start by working on Multi-Layer Perceptron (MLP) before working with more complex Convolution Neural Network. Multi-Layer Perceptron consists of multiple layers of perceptron units. It uses a non-linear activation function making it more efficient than linear perceptrons. It consists of an input layer and an output layer and in between there are an arbitrary number of hidden layers.

The Artificial Neural Network model is given a labelled training dataset to train, it consists of both inputs and outputs (as it is form of supervised learning). Initially the weights are randomly assigned, and are optimized through iterations. Back Propagation process is used to optimize the weights, first the data is passed through random weights and then the output is compared with the truth ground values. Then an optimizer function like stochastic gradient descent is used to optimize the weight values. The weight values are optimized until it can go no lower which is known as convergence.

We have used a simple sequential model which consists of three layers, the input layer, 1 hidden layer and 1 output layer. All the 3 layers are of dense type which is a common layer in neural networks. The first two layers have 256 nodes each and the output layer has 10 nodes. We have used ReLU function as the activation function for the first two layers. ReLU function has proven to work well in Neural Networks. We have calculated 40 MFCCs per audio file so the shape of the input layer will be (1*40). We have used a dropout value of 50% for our first two layers. This will randomly exclude 50% of the units in each iteration while training the model so that the model does not get overfitted and enhances its generalization.

Our output layer has 10 nodes which is same as the number of labels. We used softmax function as the activation function for the output layer because the softmax function makes the sum of outputs to be 1, so that we can treat the outputs as probabilities. Then it selects the label corresponding to the highest probability as the prediction.

©IJRASET: All Rights are Reserved

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_10 (Dense)             (None, 256)               10496
_________________________________________________________________
activation_10 (Activation)   (None, 256)               0
_________________________________________________________________
dropout_7 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_11 (Dense)             (None, 256)               65792
_________________________________________________________________
activation_11 (Activation)   (None, 256)               0
_________________________________________________________________
dropout_8 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_12 (Dense)             (None, 10)                2570
_________________________________________________________________
activation_12 (Activation)   (None, 10)                0
=================================================================
Total params: 78,858
Trainable params: 78,858
Non-trainable params: 0
```

Figure 1: ANN Model

B. Convolution Neural Networks (CNN)

We again used a sequential model, where we had 4 Conv2D layers followed by an output layer which is of type dense. In CNN, the filter parameter determines the number of units in each layer, and the kernel size represents the filter size. The filter is moved all over the input data and multiplication process is done, the result of which is stored in feature map, this process is known as convolution. We had 16, 32, 64, and 128 units respectively in the 4 Conv2D layers. The kernel size is kept to be 2 that means the size of filter will be 2*2.

The input to the first Conv2D layer will be of dimension (40, 174, 1) as we have 40 MFCCs and 174 frames including the 0 paddings. Here, 1 signifies the mono channel. We have used a dropout value of 20% in the conv2D layer to minimize overfitting. We have used ReLU as the activation function for the 4 convolution layers.

Every convolution has a pooling layer after it, to reduce the dimensionality and increase the training and testing time of the model. For the first three convolution layers MaxPooling2D type pooling layer is used. And for the last convolution layer, GlobalAveragePooling2D layer is used. MaxPooling2D layer chooses the maximum value from the kernel in the feature map. And GlobalAveragePooling2D layer chooses the average value of the kernel. The GlobalAveragePooling2D layer is ideal for the data to be sent to hidden layer. Same as ANN model, the output layer had 10 output units which is same as the number of output labels. For the output layer, we have used softmax activation function, the softmax activation function makes the sum of the outputs to be 1, so they can be treated as probabilities. Then the label with the highest probability is predicted.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 39, 173, 16)       80
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 19, 86, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 19, 86, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 18, 85, 32)        2080
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 9, 42, 32)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 9, 42, 32)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 41, 64)         8256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 20, 64)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 20, 64)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 19, 128)        32896
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 1, 9, 128)         0
_________________________________________________________________
dropout_4 (Dropout)          (None, 1, 9, 128)         0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 44,602
Trainable params: 44,602
Non-trainable params: 0
```

Figure 2: CNN Model

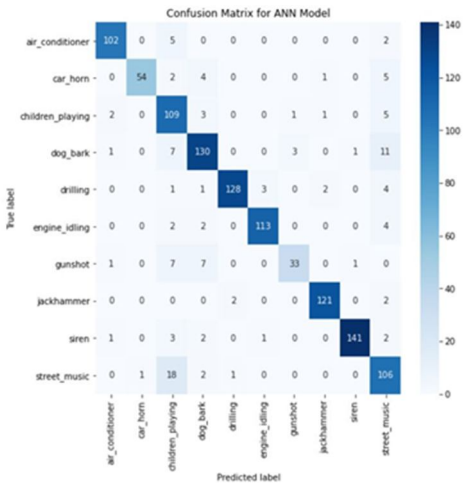


Figure 3: Confusion Matrix for ANN Model

©IJRASET: All Rights are Reserved

V. RESULTS AND ANALYSIS

Results of both the models are mentioned in this section and analysis is the inferences obtained from the results. We got the following accuracies for the ANN model; training accuracy came out to be 94.67% and testing accuracy is 88.37%. We can see the difference between their accuracies is ~6%, which means that the model has not overfitted to the training data. With the CNN model, we got the training accuracy to be 99.31% and testing accuracy to be 93.36%. Again, here we see that the model is not overfitted. The testing accuracy of both models show that the CNN model is better than the ANN model.

Figure 3 is the graphical representation of confusion matrix of ANN model. The different number of cases are represented by different shade of blue in the graph, the darker the shade of blue, the higher the number. The bright color in the diagonal represent the match cases i.e., true positives and false negatives. The lighter colors represent the wrong predictions i.e., true negatives and false positives.

Figure 4 is the graphical representation of confusion matrix of CNN model. The depiction of different cases is exactly similar to the previous model.

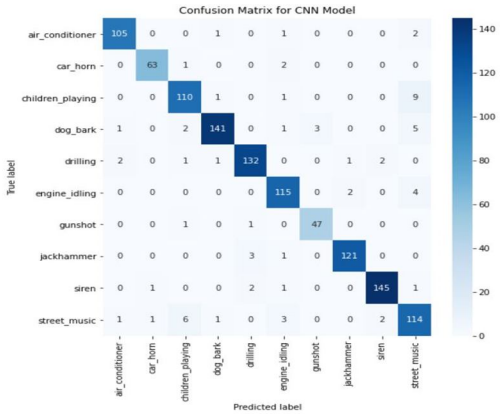


Figure 4: Confusion Matrix for CNN Model

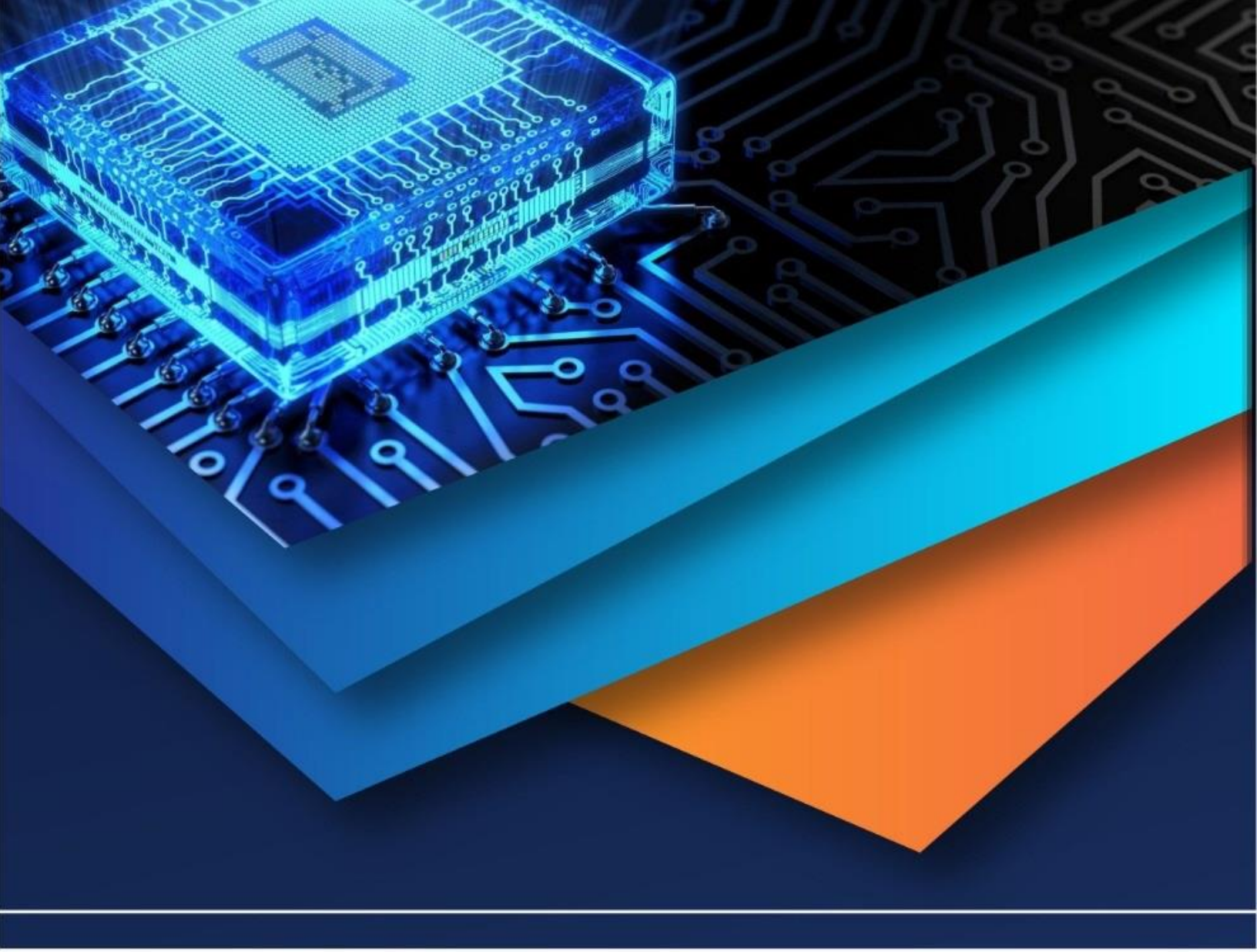
IV. CONCLUSION AND FUTURE WORK

We were able to check the working of different neural network models on the environment sound classification. In the future we can work upon real-time sounds. We can augment the data to add noises as the real-world sounds are not clear and have noises. We can experiment with other techniques of feature extraction like different forms of spectrograms.

REFERENCES

[1] Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. pp. 6645{6649. IEEE (2013).
[2] Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-based music information retrieval: Current directions and future challenges. Proceedings of the IEEE 96(4), 668{696 (2008)
[3] G. E. Hinton et al., “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv:1207.0580, 2012.
[4] D. Cires¸an, U. Meier, and J. Schmidhuber, “Multicolumn deep neural networks for image classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012, pp. 3642–3649.
[5] J. Salamon, C. Jacoby, and J. P. Bello, “A dataset and taxonomy for urban sound research,” in Proceedings of the ACM International Conference on Multimedia. ACM, 2014, pp. 1041–1044.
[6] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2013, pp. 3626–3633.
[7] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in ICMLWorkshop on Deep Learning for Audio, Speech, and Language Processing, 2013.
[8] Agrawal, D.M., Sailor, H.B., Soni, M.H., Patil, H.A.: Novel teo-based gammatone features for environmental sound classication. In: Signal Processing Conference (EUSIPCO), 2017 25th European. pp. 1809{1813. IEEE (2017)
[9] Tokozume, Y., Harada, T.: Learning environmental sounds with end-to-end convolutional neural network. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. pp. 2721{2725. IEEE (2017)
[10] D. Cires¸an, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” in Advances in Neural Information Processing Systems, 2012, pp. 2843–2851.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)