# Classification of Background Audio

BCSE IV Semester II Project Report

Priti Shaw[1], Sourav Kumar[2] and Md Mobbasher Ansari[3]
*Under guidance of* ***Dr Subhadip Basu*** *and* ***Tapas Chakraborty***
*Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India*
[1]001710501076,[2]00171051004,[3]001710501039

## I. Introduction

Real-world audio recordings are a mix of different sound signals from multiple sources. These sounds encode different types of information. The speech signals can be used to decode the speech and speaker information. Whereas the sounds from secondary sources can help us get some clues about the environment of the speaker.

Minimal background interference is always desired in most audio processing applications. However, the background sounds can reveal some important information that can be useful at many times. The environment information can be useful in forensic applications to gather or reason about the evidence. Other applications of background sound classification include surveillance systems, searching through audio archives, or clustering audio recordings.

## II. Datasets

1. **UrbanSound8K** is a publicly available urban sounds dataset. It contains 8732 urban sound samples from 10 classes. Each sample has a length greater than or equal to 4 seconds. The ten classes are air conditioner, children playing, dog bark, car horn, drilling, siren, engine idling, gunshot, jackhammer, and street music.

2. The **ESC-10** dataset is a collection of short environmental recordings. It has been pre-arranged into five uniformly sized folds so that clips extracted from the same source recording are always contained in a single fold. It is a labeled set of 400 environmental recordings containing 10 classes, 40 clips per class. All clips have been extracted from public field recordings available through the Freesound.org project.

## III. Methods and Materials

The proposed method consists of four main stages which are pre-processing, model training, feature extraction and classification. The preprocessing step consists of filtering and signal to image conversion. Then a CNN model is trained on the images. Next features are extracted from the model. And classification consists of two results, first one is classifying using the CNN model itself, and second one is a SVM classifier in which features are fed. The same set of steps are followed for each of the training datasets i.e, UrbanSound8k and ESC-10. Each of the datasets have 10 classes and 10 folds.

**2.1. Preprocessing :** The given data sets consist of audio signals with wav extension. We need to convert each audio signal into a spectrogram image. First we use the load method in Python's librosa library to get the numpy array representation of the audio files. The array consists of float values. The method also returns the sample rate of the audio. Next we removed noises in the audio signals using the formula

$$\theta(n-1) = f(n+1) - \alpha * f(n)$$

Next, we divided each signal into some consecutive frames. For each frame, we calculated the energy of that frame. And we also calculated the average energy of the audio that is the total energy of the audio divided by the number of frames. Now for each frame, if the energy of that particular frame is greater than or equal to the average energy, we select that frame. So now for each signal, we have its corresponding numpy array having the energy values

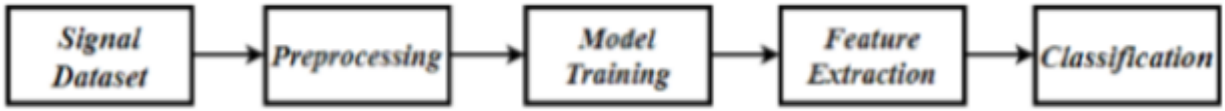


Fig. 1. Flowchart of Proposed Methodology

of the selected frames. Now we generate the mel-spectrogram images corresponding to the numpy arrays and store them foldwise in their respective class folder.

**2.2. Model Training :** In order to generate feature vectors and also for classification, we train a CNN model. In this study, we have used Densenet201 as the CNN model. First we converted the spectrograms into numpy arrays of size 3x224x224 (RGB Channel separation). Next we prepare our training, validation and test datasets by dividing all the spectrogram arrays into 8:1:1 ratio. Next we create a Densenet201 model and use SGD optimizer and categorical cross entropy loss. Now we train the model on the training dataset and validate it on the validation dataset. For UrbanSound8k , we train on 150 epochs and for ESC-10 , we train on 250 epochs. After each epoch, we store only the model having maximum validation accuracy.

**2.3. Feature Extraction :** `Next we generate feature vectors for the images. And to do that we use the trained model (on the previous step). The features are basically the outputs from the pre-final layer (the final layer is the fully connected layer ). So we pass the train and test data respectively and extract their corresponding feature vectors. Each feature vector has 1920 columns.

**2.4. Classification :** We classify both using the trained cnn model and an svm model. First, in the trained cnn model, we pass the original test data and find the accuracy. Next , in SVM, we pass the training features and labels for training. Then we pass the testing features and find the accuracy. In the SVM model, we use Linear and Sigmoid kernels.

## IV. Experiment Results

The model was evaluated on UrbanSound8K and ESC-10 datasets. ESC-10 is a 10-fold dataset. UrbanSound8K being a 5-fold dataset was converted to a 10-fold by dividing each fold into two. Our proposed method is compared with other methods in Table I for ESC-10 and Table II for UrbanSound8K.

We have used the same model structure for all the datasets. The parameters for the spectrogram were determined heuristically by doing experimental results. The initial size of the spectrogram was 400 x 400 but it was resized to 224 x 224 to fit with DenseNet. Results show that our model achieves better performance on all two datasets 97.8% and 78.2% for ESC-10 and UrbanSound8K respectively.

**TABLE I. Comparison to the Proposed Method with Other Methods for ESC-10**

| Method | ESC-10 |
|---|---|
| Pyramid-Combined CNN [2] | 94.8% |
| PiczakCNN [1] | 80.5% |
| SoundNet [3] | 92.1% |
| WaveMsNet [4] | 93.7% |
| Multi-Stream CNN [5] | 93.7% |

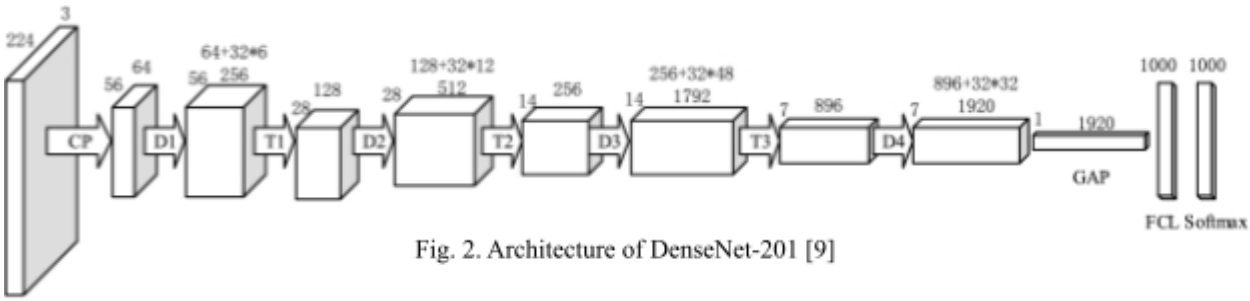


Fig. 2. Architecture of DenseNet-201 [9]

**TABLE II. Comparison to the Proposed Method with Other Methods for UrbanSound8K**

| Method | UrbanSound8K |
|---|---|
| Pyramid-Combined CNN [2] | 78.14% |
| Dilated convolution [6] | 78% |
| Convolutional layers with max-pooling [1] | 74% |
| Deep CNN [7] | 74% |
| Long segments/majority voting [8] | 71.8% |

## V. Conclusions:

In this paper, we propose a CNN method that is composed of various stages such as pre-processing, model training, feature extraction, and classification. Our system was evaluated on two commonly used benchmark datasets and achieved better performance or state-of-art with single network architecture.

In future works, we are planning to work on larger datasets such as ESC-50, and to use other pre-trained CNN models and classifiers.

## References:


1. Piczak KJ. Environmental sound classification with convolutional neural networks. In 25th International Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, USA; 2015. p. 1–6.
2. Demir, Fatih, et al. "A new pyramidal concatenated CNN approach for environmental sound classification." Applied Acoustics 170 (2020): 107520.
3. Aytar Y, Vondrick C, Torralb A. Soundnet: Learning sound representations from unlabeled video. In Advances in neural information processing systems p. 892–900.
4. Zhu B, Wang C, Liu F, Lei J, Lu Z, Peng Y. Learning environmental sounds with multi-scale convolutional neural network. arXiv preprint arXiv:1803.10219.
5. Li X, Chebiyyam V, Kirchhoff K. Multi-stream network with temporal attention for environmental sound classification. arXiv preprint arXiv:1901.08608
6. Chen Y, Guo Q, Liang X, Wang J, Qian Y. Environmental sound classification with dilated convolutions. Appl Acoust 2019;148:123–32.
7. Salamon J, Bello JP. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process Lett (SPL) 2017;24(3):279–83.
8. Maxudov N, Özcan B, Kıraç MF. Scene recognition with majority voting among sub-section levels. 2016 24th signal processing and communication application conference (SIU) IEEE; 2016. p. 1637–1640.
9. Wang, Shui-Hua, and Yu-Dong Zhang. "DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16.2s (2020): 1-19.