International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Feature Extraction and Enhanced Classification of Urban Sounds

Asma Begum[1], Afshaan Kaleem[2]

[1]Masters Student, Department of ECE, Muffakham Jah College of Engineering and Technology, Hyderabad, India
Email: *syed.asma4474[at]gmail.com*

[2]Senior Assistant Professor, Department of ECE, Muffakham Jah College of Engineering and Technology, Hyderabad, India
Email: *afshan.kaleem[at]mjcollege.ac.in*

Abstract: ***Urban Sound Classification is an important but challenging problem. In this paper, we present a new deep convolutional neural network for classification tasks that combines MFCC with Mel spectrogram. In comparison to using a single feature, this feature combination can make the features richer. The network suggested extracts and derives high-level features using three convolutional blocks, each of which is made up of two convolutional layers and a pooling layer. We apply a filter with a limited receptive field in each convolutional layer to preserve the network's depth and lower the number of parameters. On ESC-50 and UrbanSound8K, where our technique was tested, classification accuracy was 45.60% and 91.0%, respectively. The experimental results show that the proposed method is suitable for Urban Sound classification***

Keywords: MFCC, Feature Extraction. Deep learning, Urban Sound classification

1. Introduction

This research project focuses on the application of convolutional neural networks (CNNs) and deep neural networks (DNNs) to investigate the classification of urban sounds. The primary dataset employed in this study is the UrbanSound8K dataset, a comprehensive collection of audio samples commonly encountered in urban settings. To facilitate comparative analysis and further bolster the research, the ESC-50 dataset has also been incorporated.

Traditional approaches for the Urban Sound Classification (USC) task have historically depended on manually crafted features, which are subsequently utilized by conventional classifiers like K-Nearest Neighbors (KNN) or Support Vector Machines (SVM) [1]. Nonetheless, these methods often fall short of our performance expectations, primarily because traditional classifiers are unable to perform additional feature extraction [2].

In contrast, Convolutional Neural Networks (CNNs) have emerged as a game-changing technology in numerous pattern recognition tasks, including but not limited to the classification of traffic signs, pedestrian detection, and facial recognition.

CNNs have traditionally found their primary application in the realm of visual recognition. However, their versatility has been notably demonstrated through their successful application in diverse domains, including speech [4], [5], music analysis [6], and the recognition and classification of everyday sounds [7], [8], [9]. In particular, within the field of sound detection and classification, there has been a growing trend in designing CNN-based methodologies, which have consistently delivered state-of-the-art performance. As an illustration, numerous CNN-based approaches have exhibited strong performance in sound classification and detection tasks within the DCASE (Detection and Classification of Acoustic Scenes and Events) community competition. You can readily access these valuable resources online for further reference.

Upon scrutinizing these highly effective approaches, we discern that the development of a CNN-based method for the Urban Sound Classification (USC) task can be dissected into two pivotal facets: the design of the network architecture and the selection of input features. It's worth noting that while data augmentation techniques often yield marginal enhancements in the ultimate performance, we have opted not to employ this strategy in our methodology. This decision is grounded in the recognition that data augmentation can potentially alter the original distribution space of the data.

Therefore, in this research paper, we introduce a novel deep CNN-based method tailored for the USC task. In this method, we leverage a fusion of MFCC (Mel-Frequency Cepstral Coefficients) and Mel spectrogram data as the input features for the CNN model.

2. Existing Work

In this paper [11], the research that employs a deep CNN architecture inspired by VGG for the Environmental Sound Classification (ESC) task has been examined. This approach utilizes concatenated spectrograms as input features and has demonstrated superior classification performance when compared to methods relying on LMS (Log-Mel Spectrogram) and LGS (Log-Gabor Spectrogram) features, particularly on both the ESC-50 and UrbanSound8K datasets.

3. Evaluation Metric

The "Classification Accuracy" evaluation metric, which is defined as the proportion of accurate predictions, will be used for this project.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} * 100$$

Number of correct predictions is the number of instances that the model classified properly. Total number of predictions is the overall number of instances for which the model provided predictions.

Due to the presumption that the dataset will have a balanced distribution (as further detailed in the following section), classification accuracy was chosen as the best statistic.

3.1. Audio Data Overview and Analysis

The field of sound classification and analysis frequently makes use of the "UrbanSound8K" dataset, which is a large collection of audio recordings.8732 tagged sounds are present in the data, divided into 10 different classes, including:

- Siren
- Street Music
- Drilling
- Engine Idling
- Air Conditioner
- Car Horn
- Dog Bark
- Jackhammer
- Drilling
- Gun Shot

All audio samples are in the.wav file type and are taken at regular intervals at the common sampling rate of 44.1 kHz, or 44,100 samples per second.

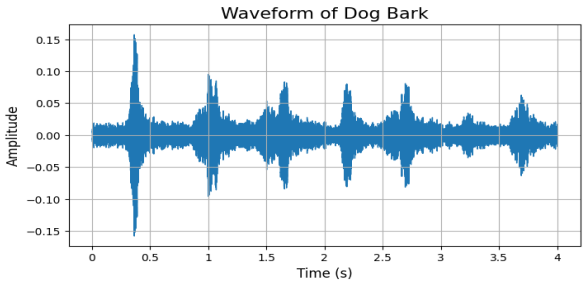

Figure 1: Waveform representing dog bark

4. Methodology

4.1. Algorithm

Our proposed solution to address this problem leverages the remarkable success of Deep Learning techniques, particularly in the realm of image classification.

To initiate the process, we commence by extracting Mel Frequency Cepstral Coefficients (MFCCs) and Mel Spectogram from the audio samples. These coefficients encapsulate the relationship between perceived audio frequencies and their actual measured values, enabling us to analyze both temporal and frequency characteristics within the samples. These representations provide us with the necessary discriminative features essential for classification.

Convolutional Neural Networks (CNNs), a neural network architecture that builds upon the foundations of Multi-Layer Perceptron with notable modifications, constitute a pivotal component of our methodology. In CNNs, the dimensions—height, width, and depth—are structured into layers, and connections between nodes in one layer are not established with all nodes in the subsequent layer. The architecture of CNNs facilitates a two-fold process:

Firstly, the feature extraction phase, in which a filter window traverses the input data, accumulating convolutions at each location, and the feature map stores these extracted features from each window. Intermediate pooling layers are interspersed within the CNN architecture. Typically, the maximum value within each window is retained during pooling, preserving essential data while reducing the feature map's dimensions. Pooling plays a pivotal role as it reduces network dimensionality, mitigating the risk of overfitting and expediting training.

Subsequently, we transition to the classification phase. After this process, the 3-D data representation within the interface is flattened and transformed into a 1-D vector.

The choice of CNNs is motivated by their inherent capabilities in feature extraction and classification, making them well-suited for tasks like image classification. This combination of robust feature extraction and classification capabilities positions CNNs as superior classifiers for our purpose.

4.2. Data Preprocessing and Splitting

Figure 2 illustrates a log-scaled mel spectrogram that was generated from an audio sample within the dataset. To process the data uniformly, we employed Mel-frequency cepstral coefficients (MFCC), which involve applying a linear cosine transform to the log power spectrum, all within a nonlinear mel scale of frequency. Essentially, this preprocessing step transformed the audio files into spectrograms, representing audio signals as images.

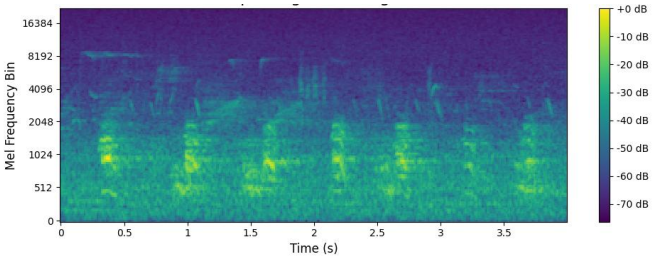

Figure 2. Log scaled Mel Spectogram

However, this translation necessitated the use of various parameters such as Fourier-Transform, Hop-length, and Melcoefficients. It's important to note that while this transformation to an image enriches the feature set, it does come at the cost of reducing some information inherent in the original audio files. Specifically, the choice of parameters affects the log-frequency (y-axis) and time domain (x-axis) representation.

To convert audio files into these log spectrograms, we employed the Python library librosa. Our data processing workflow began with the loading of a comma-separated (.csv) file containing the titles of the audio files along with their corresponding labels. We defined a function to iterate through each row of this data frame, extracting the features by reading the file paths.

The result was an array consisting of 193 features, each paired with its respective label. This array served as the basis for defining our training, validation, and testing datasets.

Following this, we performed dataset scaling and selected 6985 samples for our training dataset, while allocating 1747 samples each for our validation and testing datasets. This meticulous preparation laid the foundation for our subsequent model training and evaluation efforts.

4.3. Model Implementation

The model is using Keras for audio classification task. This model architecture follows the VGG pattern of stacking multiple convolutional and max-pooling layers, followed by dense (fully connected) layers for classification.

It consist of three pairs of convolutional layers, each with a rectified linear unit (ReLU) activation function.

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

These layers are responsible for extracting features from the input data. The number 64, 128, and 256 represents the number of filters (also known as kernels) in each layer. The (3, 3) specifies the size of the convolutional kernels, and padding='same' ensures that the spatial dimensions of the output feature maps remain the same as the input.

After each pair of convolutional layers, max-pooling layer is added with a (2, 2) pool size. Max-pooling helps reduce the spatial dimensions of the feature maps, which can help in reducing computational complexity and over fitting. Following the convolutional and max-pooling layers, a Flatten layer is included. This layer reshapes the 2D feature maps into a 1D vector, which can be fed into the dense layers for classification.

Finally two dense layers with 512 units each and ReLU activation functions are added. These layers are responsible for high-level feature aggregation and making final classification decisions.

The final dense layer has a number of units equal to the number of classes (num_classes) in your classification task. It uses a softmax activation function to produce class probabilities. The following is the model summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 64)      1792
conv2d_1 (Conv2D)            (None, 224, 224, 64)      36928
max_pool2d (MaxPooling2D)    (None, 112, 112, 64)      0
conv2d_2 (Conv2D)            (None, 112, 112, 128)     73856
conv2d_3 (Conv2D)            (None, 112, 112, 128)     147584
max_poo2d_12D) (MaxPooling2D (None, 56, 56, 128)       0
conv2d_4 (Conv2D)            (None, 56, 56, 256)       295168
conv2d_5 (Conv2D)            (None, 56, 56, 256)       590080
conv2d_6 (Conv2D)            (None, 56, 56, 256)       590080
max_pool2d_22D) (MaxPooling2D (None, 28, 28, 256)      0
flatten (Flatten)            (None, 200704)            0
dense (Dense)                (None, 512)               102760960
dropout (Dropout)            (None, 512)               0
dense_1 (Dense)              (None, 512)               262656
dropout_1 (Dropout)          (None, 512)               0
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 104,764,234
Trainable params: 104,764,234
Non-trainable params: 0
_________________________________________________________________
```

4.4. Training and Testing

We conducted training over approximately 30 epochs, with an epoch step size of 64, and each epoch consisted of 219 validation steps per batch. Our test dataset comprises 1747 audio samples, which encompass a diverse and random distribution of various sounds.

4.5. Results

We have compared our approach with recent related studies to provide a more comprehensive evaluation of our method's performance. The results from these comparisons are presented in the table. Notably, our method achieves the highest performance, scoring 91.06%, on the Urban Sound 8k dataset. This represents a substantial 18.36% improvement over Piczak's method and a 10.76% improvement compared to the baseline method.The plot of training and validation accuracy is shown in figure 3. And model summary in figure 4

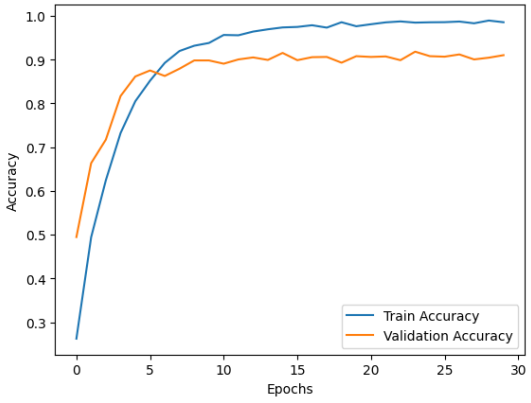

Figure 3: Model Accuracy

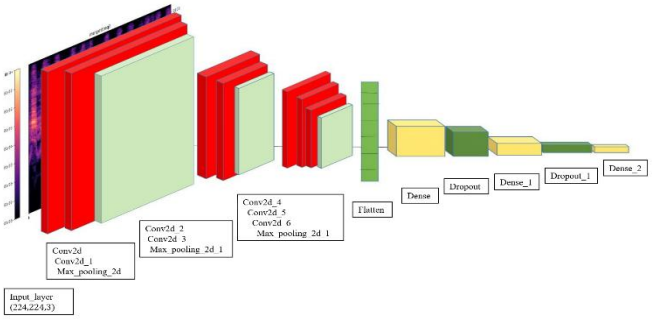

Figure 4: Model Summary

Table 1: Classification accuracy between different methods

Methods	ESC-50	Urban Sound 8K
Piczak	64.9 [3]	72.7
D-CNN	68.1 [10]	81.9
Zhang	76.8 [2]	74.7
Baseline	**83.8**[11]	80.3
Proposed	45.06	**91.06**

While our method didn't outperform all others on the ESC-50 dataset, these findings demonstrate that increasing the depth of the Convolutional Neural Network (CNN) and employing filters with smaller receptive fields can significantly enhance the network's recognition capabilities.

5. Conclusions

In our paper, we introduce a deep Convolutional Neural Network (CNN) architecture inspired by VGG, specifically designed for the Urban Sound Classification (USC) task. This architecture takes advantage of MFCC and mel spectrogram features as input, and it has achieved superior performance in classifying the UrbanSound8K dataset.

Moreover, we conducted experiments where we evaluated our proposed CNN alongside a Deep Neural Network (DNN) on both datasets. The results clearly demonstrate that our VGG-inspired architecture not only exhibits better performance but also boasts a reduced number of parameters compared to the alternative models. Furthermore, our proposed approach showcases superior urban sound classification capabilities when contrasted with various recent CNN-based methods.

References

[1] C. Wang, J. Wang, A. Santoso, C. Chiang and C. Wu, "Sound Event Recognition Using Auditory-Receptive-Field Binary Pattern and Hierarchical-Diving Deep Belief Network," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 8, pp. 1336-1351, Aug. 2018.

[2] Z. Zhang, S. Xu, S. Cao, S. Zhang. "Deep convolutional neuralnetwork with mixup for environmental sound classification." Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, Cham, 2018.

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

[4] T. N. Sainath, A. Mohamed, B. Kingsbury and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing,Vancouver, BC, 2013, pp. 8614-8618.

[5] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn and D.Yu, "Convolutional Neural Networks for Speech Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1533-1545, Oct. 2014.

[6] Van den Oord, S. Dieleman, B. Schrauwen, "Deep content-based music recommendation," Advances in Neural Information Processing Systems, 2013, pp. 2643-2651.

[7] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," in IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279-283, March 2017.

[8] McLoughlin, H. Zhang, Z. Xie, Y. Song and W. Xiao, "Robust Sound Event Classification Using Deep Neural Networks," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 3, pp. 540-552, March 2015.

[9] X. Zhang, Y. Zou, and S. Wei, "Dilated convolution neural network with Leaky ReLU for environmental sound classification." 2017 22nd International Conference on Digital Signal Processing (DSP). IEEE, 2017

[10] Kopparapu, S. Kumar, and M. Laxminarayana. "Choice of Mel filterbank in computing MFCC of a resampled speech." 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010). IEEE, 2010.

[11] Zhejian Chi, Ying Li, Cheng Chen "Deep Convolutional Neural Network Combined with Concatenated Spectrogram for Environmental Sound Classification" In 2019 IEEE 7th International Conference on Computer Science and Network Technology