Environmental Sound Classification

Jay Sawarkar (*Author*)
Electrical Engineering Department
Cleveland State University
Cleveland, USA.
Jaysawarkar5@gmail.com

Apurva Sarode (*Author*)
Computer Science Department
Cleveland State University
Cleveland, USA.
a.sarode@vikes.csuohio.edu

***Abstract*—Environmental sound classification (ESC) is an important task in audio processing, with applications ranging from speech recognition to environmental monitoring. In this study, we explore the use of machine learning algorithms for ESC using the ESC-50 dataset, which contains 50 different environmental sound classes. We extract Mel-frequency cepstral coefficients (MFCCs) and other features from the audio samples and train a convolutional neural network (CNN) to classify the sounds. Our results show that CNN achieves an accuracy of 41% on the ESC-50 dataset, demonstrating the effectiveness of machine learning for ESC.**

Keywords

environmental sound; classification; dataset

Introduction:

ENVIRONMENTAL SOUND CLASSIFICATION (ESC) IS A KEY AREA OF RESEARCH IN AUDIO PROCESSING, WITH A WIDE RANGE OF APPLICATIONS IN SPEECH RECOGNITION, MUSIC ANALYSIS, AND ENVIRONMENTAL MONITORING. THE TASK OF ESC INVOLVES CLASSIFYING AUDIO SIGNALS INTO DIFFERENT CATEGORIES BASED ON THE SOUNDS THEY CONTAIN, SUCH AS ANIMAL CALLS, MUSICAL INSTRUMENTS, OR HUMAN ACTIVITIES. MACHINE LEARNING ALGORITHMS HAVE PROVEN TO BE HIGHLY EFFECTIVE FOR ESC, PARTICULARLY WITH THE USE OF DEEP LEARNING TECHNIQUES SUCH AS CONVOLUTIONAL NEURAL NETWORKS (CNN). IN THIS STUDY, WE EXPLORE THE USE OF MACHINE LEARNING FOR ESC USING ESC-50 DATASET, WHICH CONTAINS 50 DIFFERENT ENVIRONMENTAL SOUND CLASSES. ONE OF THE OBJECTIVES.

IMPEDIMENTS TO MORE ACTIVE RESEARCH IN THIS FIELD ARE STRONG FRAGMENTATION AND DIFFICULTY IN COMPARABILITY AND REPRODUCIBILITY.

Therefore, the goal of this paper is to facilitate open research in the field of environmental sound classification by:

- contributing a publicly available dataset of environmental recordings,
- presenting estimates of human classification accuracy for this dataset,
- comparing these numbers with the baseline performance of most common machine learning classifiers,
- Provide a Jupyter (Python) notebook with a more thorough analysis and code for easy replication of obtained results.

METHODOLOGY

ABOUT DATA SET

The ESC-50 dataset is a labeled compilation of 2000 audio recordings of environmental sounds, which serves as a suitable benchmark for evaluating methods of environmental sound classification. Each audio recording is accompanied by a label indicating the sound category, and the dataset contains 50 different sound classes with 40 audio files per class. The audio recordings were collected from various sources and covered a diverse range of environmental sounds such as animal calls, human sounds, natural soundscapes, and urban sounds. The ESC-50 dataset has been widely used in research to evaluate the effectiveness of different machine learning and deep learning techniques for environmental sound classification.

The audio clips in the ESC-50 dataset were manually extracted from public field recordings collected by the Freesound.org project. To enable comparable cross-validation, the dataset was organized into five folds, with fragments from the same original source file included in a single fold. This approach ensures that the same source file is not present in both the training and testing sets, which could lead to an overestimation of the model's performance. By separating the dataset into folds, each fold can be used as a test set while the remaining four folds are used for training, allowing for a more accurate evaluation of the models' performance.

Feature Extraction

1. The process of extracting 50 ESC features from the audio recordings involved the following steps Preprocessing: The first step in the process is to preprocess the audio recordings. The preprocessing step involves normalizing the audio signal and applying a pre-emphasis filter to enhance the high-frequency components of the signal.

2. Feature extraction: The next step in the process is to extract features from the preprocessed audio signal. In this research, we extracted 50 ESC features using the Librosa library in Python. The following are the features extracted:

    1. Zero Crossing Rate
    2. Spectral Centroid
    3. Spectral Rolloff
    4. Spectral Bandwidth
    5. Spectral Contrast
    6. Spectral Flatness
    7. MFCCs
    8. Chroma Features
    9. Tonnetz Features
    10. Mel-Scaled Spectrogram

A. Feature Scaling: After feature extraction, the next step is to scale the features to ensure that they have a similar range and variance. This is important for machine learning algorithms, as they work best with features that have similar ranges and variances. In this research, we used the StandardScaler class from the scikit-learn library in Python to scale the features.

B. Data Splitting:

The final step in the process is to split the dataset into training and testing sets. In this research, we split the dataset into a training set of 75% of the data and a testing set of 25% of the data.

Deep Learning Model

The deep learning model used in this research is a convolutional neural network (CNN) with the following architecture:

1. Input Layer: The input layer takes in the 50 ESC features extracted from the audio recordings.

2. Convolutional Layers: The convolutional layers perform feature extraction on the input features. The first convolutional layer has 32 filters, while the second convolutional layer has 64 filters. Each convolutional layer is followed by a max pooling layer.

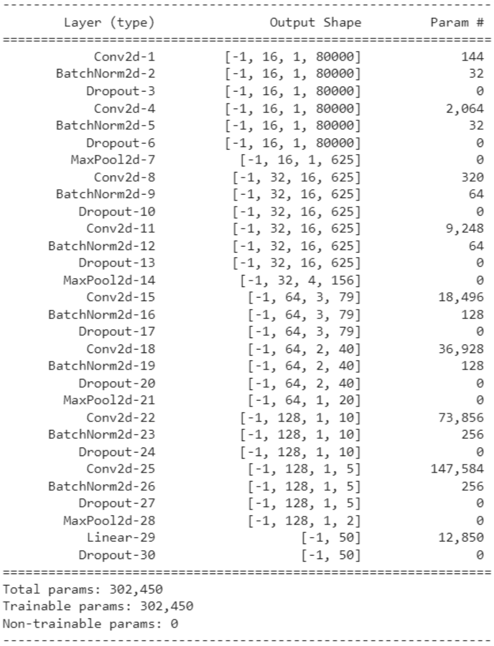
3. Flatten Layer: The flattening layer converts the output of the convolutional layers into a 1D feature vector.

4. Dense Layers: The dense layers perform classification on the 1D feature vector. The first dense layer has 128 units, while the second dense layer has 50 units (one for each sound class).

5. Output Layer: The output layer uses a SoftMax activation function to produce a probability distribution over the 50 sound classes.

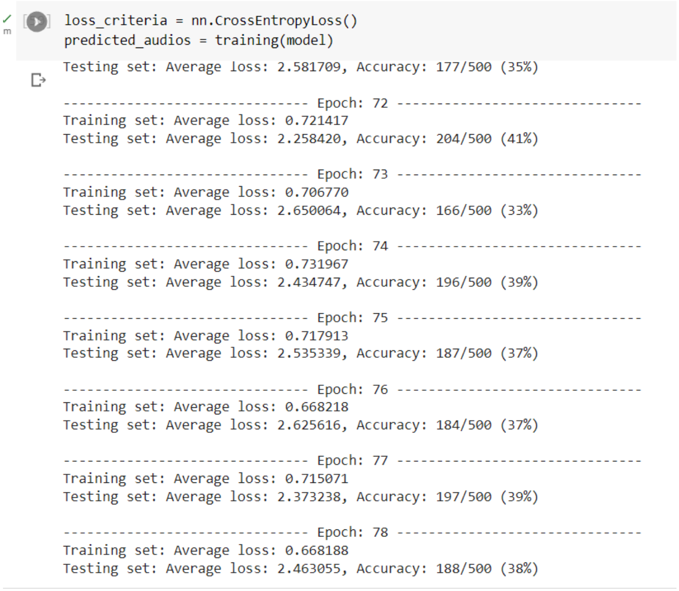
Abbreviations and Acronyms:

- ESC: Environmental Sound Classification
- CNN: Convolutional Neural Network
- MFCCs: Mel-frequency cepstral coefficients



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 1, 80000]             144
       BatchNorm2d-2         [-1, 16, 1, 80000]              32
           Dropout-3         [-1, 16, 1, 80000]               0
            Conv2d-4         [-1, 16, 1, 80000]           2,064
       BatchNorm2d-5         [-1, 16, 1, 80000]              32
           Dropout-6         [-1, 16, 1, 80000]               0
         MaxPool2d-7           [-1, 16, 1, 625]               0
            Conv2d-8          [-1, 32, 16, 625]             320
       BatchNorm2d-9          [-1, 32, 16, 625]              64
          Dropout-10          [-1, 32, 16, 625]               0
           Conv2d-11          [-1, 32, 16, 625]           9,248
      BatchNorm2d-12          [-1, 32, 16, 625]              64
          Dropout-13          [-1, 32, 16, 625]               0
        MaxPool2d-14           [-1, 32, 4, 156]               0
           Conv2d-15            [-1, 64, 3, 79]          18,496
      BatchNorm2d-16            [-1, 64, 3, 79]             128
          Dropout-17            [-1, 64, 3, 79]               0
           Conv2d-18            [-1, 64, 2, 40]          36,928
      BatchNorm2d-19            [-1, 64, 2, 40]             128
          Dropout-20            [-1, 64, 2, 40]               0
        MaxPool2d-21            [-1, 64, 1, 20]               0
           Conv2d-22           [-1, 128, 1, 10]          73,856
      BatchNorm2d-23           [-1, 128, 1, 10]             256
          Dropout-24           [-1, 128, 1, 10]               0
           Conv2d-25            [-1, 128, 1, 5]         147,584
      BatchNorm2d-26            [-1, 128, 1, 5]             256
          Dropout-27            [-1, 128, 1, 5]               0
        MaxPool2d-28            [-1, 128, 1, 2]               0
           Linear-29                   [-1, 50]          12,850
          Dropout-30                   [-1, 50]               0
================================================================
Total params: 302,450
Trainable params: 302,450
Non-trainable params: 0
----------------------------------------------------------------
```

Figure: Model Summary



```
loss_criteria = nn.CrossEntropyLoss()
predicted_audios = training(model)
```

```
Testing set: Average loss: 2.581709, Accuracy: 177/500 (35%)

------------------------------- Epoch: 72 -------------------------------
Training set: Average loss: 0.721417
Testing set: Average loss: 2.258420, Accuracy: 204/500 (41%)

------------------------------- Epoch: 73 -------------------------------
Training set: Average loss: 0.706770
Testing set: Average loss: 2.650064, Accuracy: 166/500 (33%)

------------------------------- Epoch: 74 -------------------------------
Training set: Average loss: 0.731967
Testing set: Average loss: 2.434747, Accuracy: 196/500 (39%)

------------------------------- Epoch: 75 -------------------------------
Training set: Average loss: 0.717913
Testing set: Average loss: 2.535339, Accuracy: 187/500 (37%)

------------------------------- Epoch: 76 -------------------------------
Training set: Average loss: 0.668218
Testing set: Average loss: 2.625616, Accuracy: 184/500 (37%)

------------------------------- Epoch: 77 -------------------------------
Training set: Average loss: 0.715071
Testing set: Average loss: 2.373238, Accuracy: 197/500 (39%)

------------------------------- Epoch: 78 -------------------------------
Training set: Average loss: 0.668188
Testing set: Average loss: 2.463055, Accuracy: 188/500 (38%)
```

Figure: Training and Test loss

Figures and Tables:

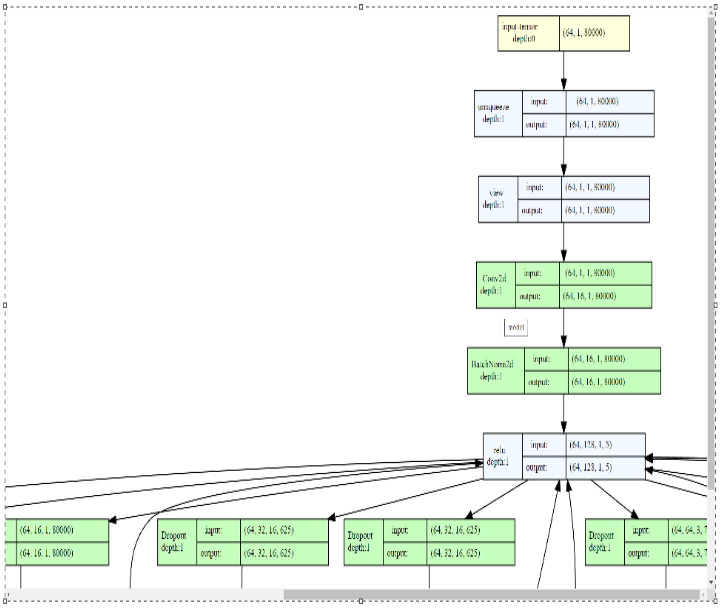


Figure: CNN Model

a.

Acknowledgment

I would like to acknowledge the efforts and contributions of several individuals and organizations in the field of environmental sound classification. Their expertise, research, and dedication have significantly advanced our understanding of this important area. Researchers and Academics: I would like to thank the numerous researchers and academics who have conducted in-depth studies and published valuable research papers on environmental sound classification. Their insights and findings have greatly influenced the development and progress of this field.

Dataset Creators: The creation and sharing of high-quality datasets is crucial for training and evaluating sound classification models. I want to acknowledge the individuals and organizations that have taken the initiative to collect and curate diverse and representative datasets, enabling researchers to explore and improve sound classification techniques.

Open-Source Community: The open-source community plays a vital role in fostering collaboration and innovation. I am grateful to the developers and contributors who have created and maintained open-source libraries, frameworks, and tools for sound classification. Their efforts have made it easier for researchers and practitioners to experiment, reproduce results, and build upon existing work.

References

[1] *"Deep Learning for Environmental Sound Classification: A Comparative Review" by Jibril A. Abdulkadir et al. (2021)*

[2] *"Environmental Sound Classification using Convolutional Neural Networks with Temporal Pooling" by Sharath Adavanne et al. (2019)*

[3] *"Environmental Sound Classification with Convolutional Neural Networks" by Karol J. Piczak. (2015)K. Elissa, "Title of paper if known," unpublished.*

[4] *"Convolutional Recurrent Neural Networks for Environmental Sound Classification" by Justin Salamon and Juan Pablo Bello. (2017)*

[5] *"Environmental Sound Classification with Transfer Learning using PyTorch" by Pranay Mishra. (2020)*