### **Abstract**

Urban sound classification (USC) is an important and challenging problem. In contrast to speech, sound events have noiselike nature and may be produced by a wide variety of sources. In this paper, we propose to explore several deep convolutional neural networks for USC tasks. Our network architecture uses stacked convolutional and pooling layers to extract high-level feature representations from spectrogram-like features(MFCC). Furthermore, we also and some traditional machine learning methods such as SVM, KNN, random Forest, XGBoost and compare the accuracy and performance of those methods. Experiments were conducted on UrbanSound8K datasets. Our experimental results demonstrated that our deep learning module(ResNet) has achieved the best performance(85.7% in maximum) on UrbanSound8K, contrast to other model

**Index Terms**—Sound Classification, Deep Learning, MFCC

### **Introduction:**

Over the past five years, developments in artificial intelligence have moved into the medium of sound, whether it be in generating new forms of music (with varying degrees of success), or identifying specific instruments from a video. Sound recognition is a front and center topic in today’s pattern recognition theories, which covers a rich variety of fields. Some of sound recognition topics have made remarkable research progress, such as automatic speech recognition (ASR) [3,4] and music information retrieval (MIR) [2,7]. Urban sound

classification (USC) is an another important branch of sound recognition and is

widely applied in surveillance [6], home automation [8], scene analysis [1] and

machine hearing [5]. However, unlike speech and music, sound events are more

diverse with a wide range of frequencies and often less well defined, which makes

Urban Sound Classification tasks more difficult than ASR and MIR. Hence, USC still faces critical

design issues in performance and accuracy improvement.

Traditional machine learning methods such as Long-Short Term Memory neural networks (LSTMs), random forest, SVM are usually associated with audio-based deep learning projects, but elements of sound identification can also be tackled as a traditional image multi-class classification task using convolutional neural networks. In this paper, we will use some neutral network made in pytorch, together with some helpful audio analysis libraries, which can distinguish between 10 different sounds with high accuracy, using the UrbanSound dataset available on Kaggle.

Reference:

1. Barchiesi, D., Giannoulis, D., Dan, S., Plumbley, M.D.: Acoustic scene classification: Classifying environments from the sounds they produce. IEEE Signal Processing Magazine 32(3), 16{34 (2015)

2. Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Contentbased music information retrieval: Current directions and future challenges. Proceedings of the IEEE 96(4), 668{696 (2008)

3. 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)

4. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A.,

Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic

modeling in speech recognition: The shared views of four research groups. IEEE

Signal Processing Magazine 29(6), 82{97 (2012)

5. Lyon, R.F.: Machine hearing: An emerging field [exploratory dsp]. Signal Processing Magazine IEEE 27(5), 131{139 (2010)

6. Radhakrishnan, R., Divakaran, A., Smaragdis, P.: Audio analysis for surveillance

applications. In: Applications of Signal Processing to Audio and Acoustics, 2005.

IEEE Workshop on. pp. 158{161 (2005)

7. Typke, R., Wiering, F., Veltkamp, R.C.: A survey of music information retrieval

systems. In: Proc. 6th International Conference on Music Information Retrieval.

pp. 153{160. Queen Mary, University of London (2005)

8. Vacher, M., Serignat, J.F., Chaillol, S.: Sound classification in a smart room environment: an approach using gmm and hmm methods. Sped 1 (2014)

A residual neural network (ResNet) is an [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANN) of a kind that builds on constructs known from [pyramidal cells](https://en.wikipedia.org/wiki/Pyramidal_cell) in the [cerebral cortex](https://en.wikipedia.org/wiki/Cerebral_cortex). Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. Models with several parallel skips are referred to as DenseNets.

For single skips, the layers may be indexed either as   to  or as  to . Given a weight matrix {\textstyle W^{\ell -1,\ell }} for connection weights from layer {\textstyle \ell -1} to , the [forward propagation](https://en.wikipedia.org/wiki/Residual_neural_network#forward) through the activation function would be as follow:



Where  is the activations (outputs) of neurons in layer,  is  the activation function for layer ,  is the weight matrix for neurons between layer  and , and . {\textstyle \ell }

And the backward propagation shows as:

For normal path:



For skip path:



Skipping effectively simplifies the network, using fewer layers in the initial training stages, this method speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The Network shows as the figure xxx.  In this paper, we call the ResNet18, ResNet34, ResNet50 for neutral network training.

Give details about what (hyper)parameters you chose (e.g. why did you use X learning  
rate for gradient descent, what was your mini-batch size and why) and how you chose  
them.

#### Experiment

4.1 Dataset

The UrbanSound8K dataset is used for model training and performance evaluation of the proposed approach, The UrbanSound8K dataset is a collection of 8732 short (up to 4 seconds) audio clips of urban sound areas. And the audio clips are prearranged into 10 folds. The dataset is divided into 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music.

4.2 Preprocessing

sr=22050

We use a 22.05kHz sampling rate for UrbanSound8K datasets. All audio samples are normalized into a range from -1 to 1. We use two spectrogram-like representations, Mel spectrum(spec) and Mel-frequency cepstral coefficients(MFCC). spec features is extracted from all recordings with FFT window size of 2048, hop length of 512, and the MFCC feature return 20 MFCCs every record. Then we use the random Crop function to drop the degree of overfitting. Finally, the spectrograms are split into 224 frames (approximately 1:5s) length with mean value of [0.485, 0.456, 0.406] in three dimensions.

4.3 Training settings

All models are trained with batch size of 32 and shuffled sampler. We used a learning rate decrease schedule with an initial learning rate of 0.01, and then divided the learning rate by 10 every twice times epoch for UrbanSound8K. Every batch  
consists of more than 40 samples randomly selected from training set without repetition.  
The models are trained for 100 epochs for UrbanSound8K. We initialize all the weights to zero mean Gaussian noise with a standard deviation of 0.05. We use cross entropy as the loss function, which is typically used for multi classification task. In the test stage, feature extraction and audio cropping patterns are the same as those used in the training stage. Prediction probability of a test audio sample is the average of predicted class probability of each segment. The predicted label of the test audio sample is the class with the highest posterior possibility. The classification performance of the methods is evaluated by the K-fold cross validation.

All models are trained using Pytorch library with TensorFlow backend on an  
Nvidia GPU with a 32GB RAM.

#### Conclusion

In this paper, we use several convolutional neural network architectures for urban sound classification with the extracted Mel and MFCC features. We compared several neutral network and results showed that our proposed ResNet always performed best(85.7% at best), and the VGG was the second(82.05%), the traditional machine learning method came the last, which perform only 20%-45.1% accuracy. Based on our experiments, ResNet and VGG achieved much better performances in test data than LSMT and other traditional machine learning methods, which met our expectation. There are several reasons causing the result. The first reason is that UrbanSound8k is large enough to train the data and related parameters compared to some small datasets where the traditional are preferable. The second reason is that neutral network have high end infrastructure to train in reasonable time with 100 epoches. Furthermore, different features may also have different performance in the classification task. For instance, the Mel Spectrum behaves better in VGG and ResNet while mel\_mean feature behaves better in traditional machine learning method. Meanwhile, experiments show that VGGNet11 and ResNet18 have very excellent performance on sound classification tasks compared to other VGG and ResNet networks. That’s because less deep network means less training timing and less overfitting, which increase the performance of Neutral Network and Accuracy of result. What’s more, because ResNet18 alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters, it achieves the best performance among other types of network.

Due to limited time, there are still some ideas we cannot realize. The first thing is that we want to change some part of ResNet18 and add more dropout in the middle. We found the training data accuracy always goes over 98%, while the testing data accuracy only 50-60%, which has big difference. Our group discussion about it and think the overfitting is main reason causing this result and overfitting is a good solution to it. What’s more, we also want to explore other feature such as root-mean square (RMS) level, spectral centroid, bandwidth and so on. Furthermore, We could also add de-noising stage before we train the data to see if it could obtain better result. They are all the points which make the accuracy of machine learning testing result better.