An Experiment of Sound Recognition using Machine Learning

Young-Jin Park, Hui-Sup Cho
Division of Electronics and Information System, DGIST
yjpark@dgist.ac.kr

Abstract

In this study, we conduct a machine learning experiment using the UrbanSound8k dataset to recognize urban sound. It can be used in cases where sound cannot be recognized due to a variety of noises through machine learning, and in areas where hearing must be protected, such as worker safety of workers. We extract mel-spectrogram from a sound file dataset and convert it to an image format for learning and validation and then determine the accuracy of the experiment. In the future, the results of our experiment will be used with non-contact sensor information from various devices such as electromagnetic sensors to safeguard workers operating in dangerous areas.

Keywords: Sound recognition, Machine Learning, MFCC, UrbanSound8k

1. Introduction

Extensive research, prompted by requirements in industry, has been conducted on machine learning using contactless sensor information. Automatic urban sound classification is a growing area of research with applications in multimedia retrieval and urban informatics [1]. Audio data are ubiquitous. Sounds outline the context of our daily activities, ranging from conversations, music, and environmental sounds, such as sounds from moving vehicles, the patter of rain, and ambient noise. The human brain is continuously processing and interpreting these audio data, either consciously or subconsciously to present information about the environment. Automatic environmental sound classification is an advancing area of research with several real-world applications [2]. In particular, several studies in the field of worker safety use machine learning. The application of technology in the field of sound recognition, particularly in factories where various noises are mixed, is likely to be of considerable help to workers To prevent hearing damage and noise. Furthermore, it is thought that the simultaneous use of various contactless sensors, including electromagnetic sensors such as radars, can be beneficial for the safety of workers. This paper shows the results of image conversion and machine learning using UrbanSound8k sound signals.

Section 2 explains how sound datasets are constructed; it also describes related studies that use sound signals. Section 3 presents the experiments conducted in this study and the subsequent results. Finally, we conclude the paper in Section 4.

2. Relate works

Using the mel-spectrogram, an algorithm that forms sound data feature vectors in the sound registration field can be designed. To use each sound as a feature, the lengths need to be equal. To obtain the mel-frequency cepstral coefficients (MFCC), the sound is split into certain units and the mel extracted and used as a feature, to yield same-size features. These features are used to compute a mel-scaled power spectrogram. These feature vectors can be used in the field of machine learning, to train appropriate models. The extracted characteristics can vary depending on the parameter settings in Python using the Librosa library, a Python package for music and audio processing. The audio is loaded in the program as a NumPy array for analysis and manipulation. The models in [1-2] that that inspired these studies use MFCC information; the results and classification accuracy of these models are shown in Table 1.

Table 1: Results of benchmark research

Result from [1]		Result from [2]	
Algorithm	Accuracy	Algorithm	Accuracy
SVM_rbf	68%	CNN	92%
IBK5	55%	MLP	88%
RandomForest500	66%	Benchmark SVM_rbf	68%
J48	48%	-	-
ZeroR	10%	-	-

The class names were drawn from the urban sound taxonomy. The files were pre-sorted into ten folds (folders named fold1–fold10) to facilitate the reproduction and comparison of the automatic classification results reported in [1].

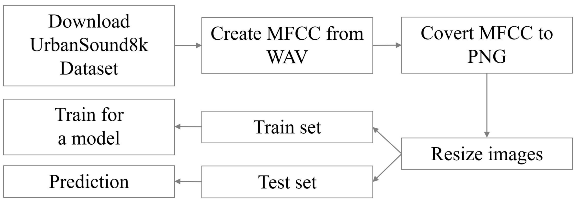
3. Experiment and result

An important aspect in machine learning is character extraction using accurate datasets. In this experiment, after extracting the characteristics of the sound files in WAV format, machine learning was performed using images converted to Portable Network Graphic (PNG) format to show recognition accuracy. The sound waves were digitized by sampling at discrete intervals known as the sampling rate (typically 44.1 kHz for CD audio quality indicating that samples were taken 44,100 times per second). First, we downloaded the UrbanSound8k dataset to acquire and generate the training set and test set used in the machine learning algorithm. This dataset contains 8732 labeled sound excerpts (<= 4 s) of urban sounds from 10 classes as shown in Table 2.

Table 2: Classes of urban sounds

No	Class name	No	Class name
1	Air_conditioner	6	Engine_idling
2	Car_horn	7	Gun_shot
3	Children_playing	8	Jackhammer
4	Dog_bark	9	Siren
5	Drilling	10	Street_music

Figure 1 below shows the overall flow of the experiment in a flow chart. The process of creating images used as training sets and test sets using the UrbanSound8k dataset is as follows. The sound files in the folders with the 10 classes are read in WAV format; the MFCC characteristics are extracted using Python's Librosa library. Next, to extract the features for machine learning, MFCC is converted into PNG format image files, a commonly used format of images. The PNG format files were converted using Python from a size of 480 by 640 to a size of 224 by 214 for GoogLeNet and Resnet50 (converted to a size of 227 by 227 for SqueezeNet) with the shape height, width and 3 channels, and partitioned into training and test sets.



Download UrbanSound8k Dataset

Create MFCC from WAV

Covert MFCC to PNG

Resize images

Train set

Test set

Train for a model

Prediction

Figure 1. Flow chart

Benchmark studies have shown how to extract features from a sound dataset and train a feedforward neural network model in Keras (Tensorflow) to categorize sound clips. In this study, in contrast to the benchmark studies shown in Table 1, machine learning was performed using images; GoogLeNet, Resnet50 and SqueezeNet were used as algorithms in MATLAB. Training and validation using the neural networks was performed like below Table 3.

Table 3: Experiment systems

	Single GPU	Dual GPU
CPU	i7-8700K	i9-10900KF
Memory	32GB	32GB
GPU	Single Nvidia Titan xp	Dual Nvidia RTX 2080Ti
Operating system	Ubuntu 18.04LTS	Microsoft Windows 10
Development tool	MATLAB (Mathworks, 2019b)	MATLAB (Mathworks, 2020a)

During learning, accurate prediction and training repetition times vary depending on the hardware used and the size of the deployment, so experiments should be conducted using multiple systems and deep learning neural networks. Figure 2, Figure 3 and Figure 4 show the results of the training and validation of machine learning from MATLAB to GoogLeNet, SqueezeNet and Resnet50, respectively. In other words, the python is used up to the previous stage of machine learning, the train set and test set generated by the python are used to machine learning in the MATLAB.

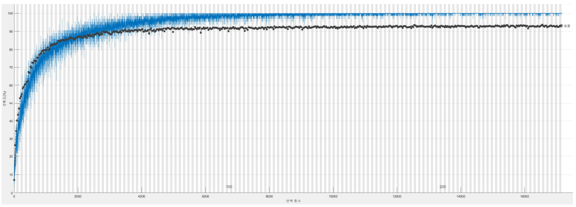


Figure 2. Train accuracy - GoogLeNet

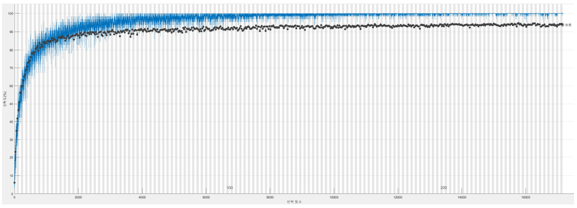


Figure 3. Train accuracy – SqueezeNet

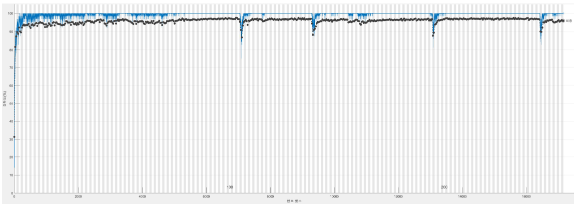


Figure 4. Train accuracy - Resnet50

Figure 5 shows some images of the results using the test set. A class name and prediction accuracy are respectively displayed on the upper area and the lower area of each image, and the image is represented by selecting one from six classes from the prediction result images using GoogLeNet.

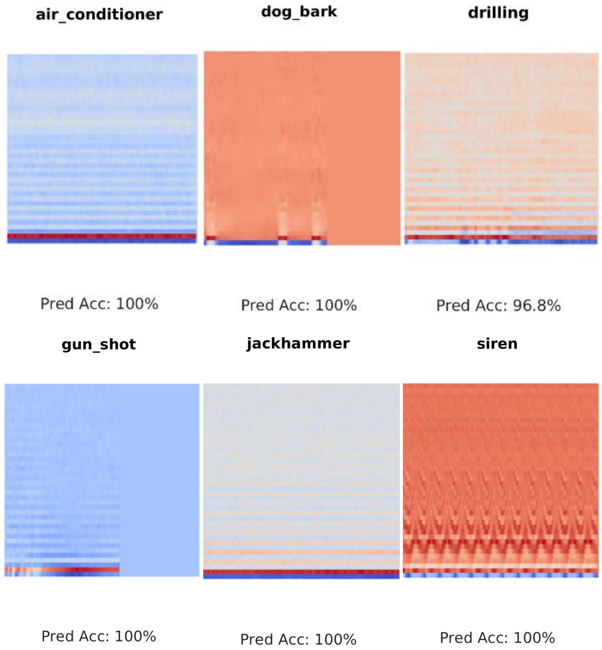


Figure 5. Prediction results using test set

Table 4 shows the number of iterations in the two neural networks used and their accuracies using single or dual GPU system. As already known, GoogLeNet and SqueezeNet were mainly used in the early stages of the experiment, but the Resnet50 was slower but more accurate than both models.

Table 4: Results using models

Epoch	Accuracy (Elapsed time)			The Number of GPU
	GoogLeNet	SqueezeNet	Resnet50	
32	90.44% (52’ 24”)	85.62% (11’ 14”)	96.29% (122Z’ 57”)	Single
	89.33% (34’ 50”)	92.3% (17’ 35”)	96.38% (81’ 10”)	Dual
128	89.7% (107’ 7”)	95.17% (31’ 40”)	97.22% (231’ 32”)	Single
	92.95% (67’ 13”)	93.97% (35’13”)	95.73% (169’ 40”)	Dual
256	92.49% (157’ 15 ”)	94.25% (48’ 11”)	96.38% (320’ 58”)	Single
	93.14% (74’ 47”)	93.78% (40’ 08”)	96.01 (206’ 28”)	Dual

4. Conclusion

Following recent advancements in the field of image classification using machine learning, particularly, using convolutional neural networks to classify images with high accuracy and at scale, the applicability of these techniques to other domains, such as sound classification, where discrete sounds occur over time is receiving considerable research attention. In this study, using machine learning, we investigated sound recognition. From our results that only shows the accuracies of sound classification using neural networks, we recommend using contactless sensors simultaneously, such as electromagnetic sensors, for the safety of workers. The simultaneous use of contactless sensors and sound signals can prevent possible accidents in remote places far from the sight and hearing range of others. In our future work, the results of this experiment will be used with non-contact sensor information from various electromagnetic sensors to safeguard workers operating in dangerous areas.

Acknowledgment

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT, and Technology of Korea (20-IT-09).

References

[1] Salamon, J., Jacoby, C., and Bello, J. P., “A dataset and taxonomy for urban sound research”, *In Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1041-1044, Nov. 2014.

[2] Mike S., “Classifying urban sounds using deep learning”, *Udacity-ML-Capstone*, Dec. 12, 2018, https://github.com/mikesmales/Udacity-ML-Capstone.