Musical Instrument Recognition

Jiho Kang, 20213013

I. Algorithm Description

In this section, the methods of feature extraction to make feature scheme and the setting of the classification models are described sequentially.

A. Features Extraction

I consider four feature extraction methods, perception based features, delta of perception based features, delta of MFCC, and Codebook based features. These features, 78 in total, are listed in Table I. Specifically, the first 20 are perception based feature and delta of perception based feature, the next 26 are delta of MFCC, and the last 32 are Codebook based.

- Perception
  Perception based features consist of zero-crossing rate (zc rate), root mean squre (rms), spectral centroid, bandwidth, and flux as shown in [1].
- ΔPerception, ΔMFCC
  By using the difference between each frame, more dynamic characteristics are obtained.
- Codebook
  To capture the overall characteristics on data, {ZC rate, RMS, Centroid, BandWidth, Flux, MFCC, Derivative of MFCC} calculated for each frame were concatenated and considered as one word, and a codebook was created by collecting and compressing them. For compressing (vector quantization), k-means was used, and k was set to 32. The overall procedure depicted in [2] was followed.

The mean value and standard deviation were obtained in common for {Perception, ΔPerception, ΔMFCC}. Importantly, the hop size was fixed at 512 for all extraction methods in this assignment.

TABLE I
Feature Scheme Description

Method	Description	#
Perception	Mean & Std of ZC rate	1-2
	Mean & Std of RMS	3-4
	Mean & Std of Centroid	5-6
	Mean & Std of BandWidth	7-8
	Mean & Std of Flux	9-10
ΔPerception	Mean & Std of Delta of {ZC rate, RMS, Centroid, BandWidth, Flux}	11-20
ΔMFCC	Mean & Std of the first 13 Delta of a MFCC feature	21-46
Codebook	Codebook-based histrogram feature {ZC rate, RMS, Centroid, BandWidth, Flux MFCC, Derivative of MFCC} of each frame	47-78

B. Classification Model

To validate my feature scheme used in this assignment, various classification algorithms were used: Support Vector Machine (SVM) with radial basis function kernel, k-Nearest Neighbors (k-NN), Multilayer Perceptron (MLP) with $relu$ activation and 4 layers and $adam$ optimizer, and Random Forest (RF). Due to time constraints, the most important hyper-parameter for each algorithm given in Table II was selected as tuned variable using a my individual criterion and the rest were kept as a default setting provided by $scikit\text{-}learn$.

TABLE II
Classification Model Hyperparameter

Model \ Hyperparameter	Type	Value
SVM	Regularizer	$10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}$
MLP	Learning rate	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$
k-NN	k	$1, 3, 5, 7, 9$
RF	Estimators	$500, 1000, 1500, 2000$

II. Experiments and Results

To begin with, an experiment was conducted to see the change in accuracy about MFCC feature when the Mel-bin and DCT sizes were different. SVM with good overall performance was used as the classification model and only MFCC was used as the feature. For each setting, only the highest accuracy among the hyper-parameter values was reported in Table III. Afterward, the Mel-bin size was set at 128 and DCT size was set at 13, and subsequent experiments were carried out.

Next, experiments were conducted on various feature schemes including mine and models. For each setting, only the highest accuracy among the hyper-parameter values was reported in Table IV. As a result, among the experiments conducted, my feature scheme and SVM classification model had the highest performance on the validation set with an accuracy of 99.3% (Table IV (red) Fig. 2.). In the case of MLP and RF, the accuracy may be slightly different each time.

TABLE III
MFCC Parameter settings

Mel-bin size \ DCT size	50	20	17	13
200	0.947	0.953	0.960	0.963
128	0.957	0.953	0.963	0.960
64	0.957	0.950	0.947	0.963
40	-	0.953	0.950	0.960

III. Discussion

When mapping linear frequency to mel scale, all frequency content within the band of each filter is summarized into a single mel bin. Therefore, I initially thought that a better feature would be created if the size of the mel-bin was increased to collect as much spectral information as possible

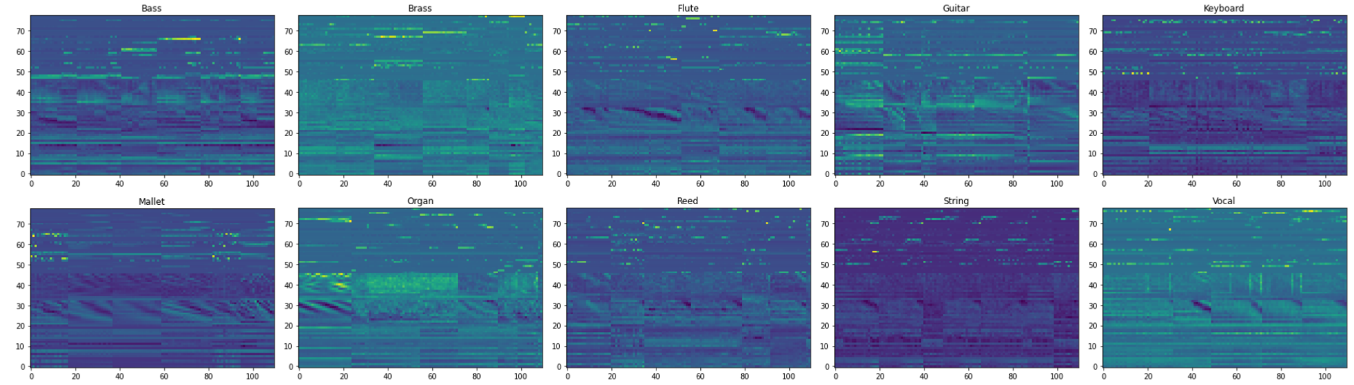

Fig. 1. Train data visualization after feature normalization. Feature scheme is {Perception, ΔPerception, ΔMFCC, Codebook}. Axis of y and x indicates feature and index of train data, respectively.

TABLE IV
CLASSIFICATION PERFORMANCE ON VALIDATION DATA

Feature Scheme \ Model	SVM	MLP	k-NN	RF
Perception ($\in \mathbb{R}^{10}$)	0.957	0.937	0.953	**0.960**
MFCC ($\in \mathbb{R}^{26}$)	0.960	0.947	0.940	**0.963**
Codebook ($\in \mathbb{R}^{32}$)	0.883	0.893	0.830	**0.923**
MFCC, ΔMFCC, Δ^2MFCC ($\in \mathbb{R}^{26\times 3=78}$)	**0.980**	0.973	0.970	0.970
Perception, ΔPerception ($\in \mathbb{R}^{10\times 2=20}$)	**0.973**	0.963	0.963	0.967
ΔPerception, ΔMFCC ($\in \mathbb{R}^{10+20=30}$)	**0.983**	0.970	0.980	0.980
Perception, ΔPerception, ΔMFCC ($\in \mathbb{R}^{10\times 2+26=46}$)	**0.987**	0.980	0.983	0.980
Perception, ΔPerception, ΔMFCC, Codebook ($\in \mathbb{R}^{10\times 2+26+32=78}$)	**0.993**	0.973	0.967	0.980

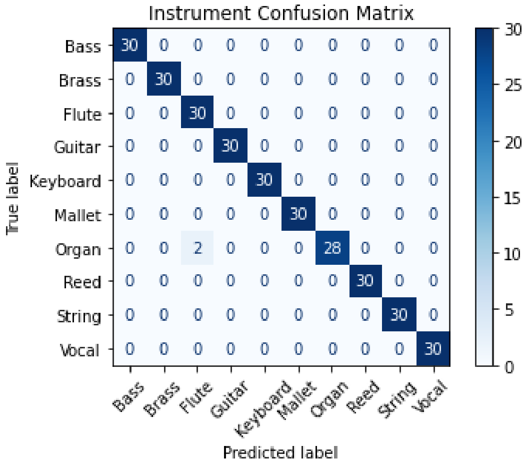

Fig. 2. Instrument confusion matrix on validation data. Feature scheme is {Perception, ΔPerception, ΔMFCC, Codebook}, and classification model is SVM (radial basis function kernel)

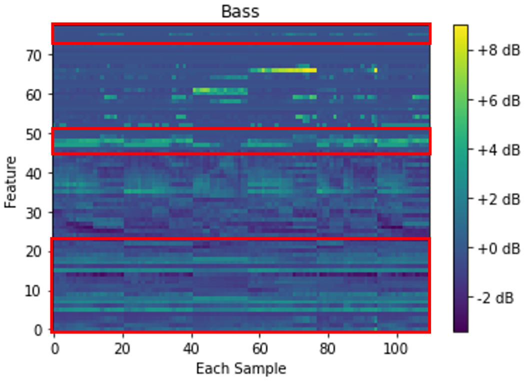

Fig. 3. Train data visualization (Bass)

into each mel-bin before DCT was performed. However, when classification was performed using only MFCC as a feature, there was no significant change in performance as shown in Table III. Since most of the information is focused on low frequency, it is estimated that performance is not significantly affected if it exceeds a certain mel-bin size. Furthermore, if the DCT size is set within 13-20, it does not seem to affect the performance much because it keeps the lowest coefficients and discards the remainder to make the timbre feature invariant with respect to the pitch information appearing in higher coefficients [3].

Train data visualization of which my feature scheme is {Perception, ΔPerception, ΔMFCC, Codebook} after feature normalization is shown in Fig. 1. This shows a horizontal pattern for each instrument. Also, this pattern is distinguished to some extent for each instrument. More specifically, certain patterns of Bass were prominent in perception based features and some of Codebook based features as shown in Fig. 3. Additionally, when principal component analysis (PCA) is applied to train data and visualized, Reed, Keyboard, Brass, and Organ can be linearly discriminated as illustrated by Fig. 4. and Fig. 5. It can be seen that the feature scheme represents well musical characteristics of each instrument in the dataset.

In this assignment, Chroma features were not used. Although Chroma features can be useful in music synchronization, chord recognition, music genre classification, it tends to remove timbre information. Thus, it was thought that Chroma-oriented features were not appropriate in this task because each data contains each independent instrument sound and different pitch.

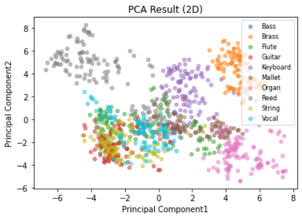

Fig. 4. Principle Component Analysis on Train Data (2 dimension)

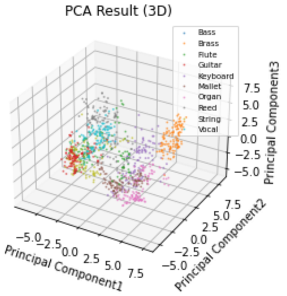

Fig. 5. Principle Component Analysis on Train Data (3 dimension)

References

[1] J. D. Deng, C. Simmermacher, and S. Cranefield, “A study on feature analysis for musical instrument classification,” *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, vol. 38, no. 2, pp. 429–438, 2008.

[2] MathWorks, “Image classification with bag of visual words,” Available at https://kr.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html.

[3] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard, “Signal processing for music analysis,” *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, vol. 5, no. 6, pp. 1088–1110, 2011.