Musical Instrument Classification using Deep Convolutional Recurrent Neural Networks and Independent Component Analysis

Tse-Shao Chang
University of Michigan
tsechang@umich.edu

Wei-Chung (Leo) Lee
University of Michigan
leowcl@umich.edu

HaoTsung Lee
University of Michigan
haolee@umich.edu

Weichin Chien
University of Michigan
weichin@umich.edu

Jongwook Choi
University of Michigan
jwook@umich.edu

Honglak Lee
University of Michigan
honglak@eecs.umich.edu

Abstract

In the article we present a noise-robust architecture combining the Fractional Fourier transform-based Mel-frequency cepstral coefficient (FrFT-based MFCC) feature extraction and Convolutional Recurrent Neural Networks (C-RNNs) classifier to recognize musical instrument signals extracted by independent component analysis (ICA). This method has the ability to classify the non-perfect isolated musical instrument components separated by ICA. To demonstrate this technique, we apply FastICA to separate musical instrument components from a mixed musical data, and then extract the audio features by FrFT-based MFCC and classify them with a trained C-RNN model. We also show that this approach can be employed with real-world data.

1 Introduction

In the field of music information retrieval (MIR), it is a highly valuable task to know what instruments are used in music. For example, if an instrument information is included in the audio tags, it allows people to search for music with the specific instrument they want. Moreover, it can be used to enhance the performance of other MIR tasks. For instance, knowing the number and type of the instrument can greatly improve the performance of automatic music transcription, and genre classification [1].

After musical instrument recognition on solo performance [2] was implemented at 2004, tons of music instrument recognition research in the field of machine learning/deep learning has emerged. To elaborate, Heittola et al. used a non-negative matrix factorization (NMF)-based source-filter model with MFCCs and GMM for synthesized polyphonic sound [3]. Kitahara et al. used various spectral, temporal, and modulation features with PCA and linear discriminant analysis (LDA) for classification [4]. Duan et al. proposed the uniform discrete cepstrum (UDC) and mel-scale UDC (MUDC) as a spectral representation with a radial basis function (RBF) kernel support vector machine (SVM) to classify 13 types of Western instruments [5]. Nevertheless, most of the previous studies focus on identifying the instrument sounds in clean single tones or phases.

In the article, we present a method that can classify the musical instrument sounds separated by ICA. These independent components are separated from the mixture of plays of real world musics instruments. Furthermore, our method can potentially isolate multiple instruments from real world music like bands, concerts and orchestras. Then, we will design and implement a Convolutional Recurrent Neural Networks (C-RNN) classifier, a deep learning architecture combining convolutional neural network (CNN) and recurrent neural network (RNN), to recognize the instrument.

The paper was laid out as follows; in Section 2, we will discuss the related work and the state-of-the-art methods. In Section 3, we will present the pipeline of our work including the data used in our work, a novelty way of preprocessing, and describe in detail the chosen model architecture. Section 4 will outline the results. Finally, in Section 5, we will list our conclusions and discuss the future application.

2 Related Work

Music classification is a popular task that has been studied previously by many machine learning researchers. In 2004, musical instrument recognition on solo musical performance was proposed using MFCC and SVM [6]. The MFCCs were used as musical features to train the SVM classifier, resulting in a high accuracy recognition for instruments in different families. In 2008, Deng et al. compared the performance of different feature extraction techniques for different machine learning techniques for music instruments recognition [7]. The feature extraction techniques included perception based features, MPEG-7 timbral features, MFCC features; the machine learning techniques include k-NN, naive Bayes, multilayer perceptron (MLP) and RBFs, and SVM. MFCC features were found giving the best classification performance among them and suitable for musical instrument classification [8, 9]. Feature extractions for musical instrument classification using wavelet methods was proposed [10, 11]; however, these methods were found to be computationally expensive [12].

FrFT-based MFCC [12] was proposed in 2016 for automatic classification of musical instruments for Counter Propagation Neural Network (CPNN) classifier. Compared to MFCC, cepstrum analysis, temporal features and spectral features, FrFT-based MFCC were found dominating among the feature extraction techniques and showed a significant improvement in classification accuracy and robustness against additive white Gaussian noise.

CNNs were introduced to the predominant instrument recognition [13]. The CNN generally uses in classifying images but in the system using CNN to identify sounds, the audio signal behaves as the image on transforming audio into Mel-spectrogram. The very deep CNNs were able to achieve a good performance by learning the MFCC features of audio data [13, 14].

Identifying sound is an inherently temporal task, and some previous work shows that classification of instrument may depend on temporal feature [13]. An effective way to model temporal processing is by using RNNs, which learn representations from sequential data [15]. In AI research, RNNs have made impressive progress in speech and action recognition [16], demonstrating the potential to use temporal feature for classification.

The latest papers show that C-RNNs architecture performs well on the multiple sound events [17] and different musical artists [18]. In our study, we proposed an architecture combining the FrFT-based MFCC feature extraction and C-RNNs classifier for real world musical instrument classification and recognition.

3 Proposed Method

This section discusses the dataset and process flow of proposed method: independent component analysis, features extraction, and model architecture and design.

3.1 Dataset

In this project, we selected five popular different instruments: piano, violin, flute, guitar and harp to train our model. All audios were collected from complete music shows on YouTube rather than a single tone. In other words, different music components such as chords were included in our dataset.

The audio files were clipped to 2 second length as the training data. All samples were provided as uncompressed .wav 16 bit, 44.1 kHz, mono audio files. 1400 data were collected for each instrument and totally 7000 data were prepared. Besides, we also took the mixed audio into consideration. Most of time, the audio signal we want to classify is not a single source, but a audio that contains several signals. In other words, one needs to extract every independent signal source first and then perform the classification. To achieve this goal, we applied ICA to separate different independent signals before extracting the features. We simulated these data by overlaying our five different audios, and added some noise into the mixed audio to mimic real world data.

3.2 Independent Component Analysis (ICA)

ICA is a technique that can separate the independent components from mixed sources, and is widely applied in audio stream separation [19]. In this project, we want to test whether the audios separated by ICA can be correctly classified by our model. We produced the sources of ICA by manually adjusting the volumes of five audios and overlaying them together to produce the mixed music. The audios were multiplied by different uniform random variables ranging from 0.3 to 1 respectively. Besides, we also created the noisy mixed music by adding some crowd noise into the mixed audio. The volume coefficient of the noise is a uniform random variable ranging from 0.1 to 0.3. Then we applied FastICA from sklearn library to extract the independent audio sources.

3.3 Features Extraction: FrFT-based MFCC

Fractional Fourier transform-based Mel-frequency cepstral coefficient (FrFT-based MFCC) has been shown a suitable acoustic feature for musical instrument classification [12]. In this study, we adopt the FrFT-based MFCC features as the acoustic features. MFCC is a short time power spectral representation of a signal, representing psychoacoustic properties of the human auditory system [7–9]. It has been widely used as the acoustic features for speech analysis and music analysis. Compared to MFCC, FrFT-based MFCC adopts FrFT to construct the acoustic features instead of fast Fourier transform (FFT), Figure 1. FrFT can decompose the signal into a set of orthonormal chirp signals, and thus can be used to capture the dynamic behavior of music sound signals. Thus, by using FrFT-based MFCC features, the classification accuracy and robustness against white noise can be improved [12]. The procedure is described below.

3.3.1 Pre-emphasis

The aim of pre-emphasis is to compensate for the high frequency components which usually have smaller magnitudes compared to lower frequency components, and improve the signal-to-noise (SNR) ratio. The pre-emphasis is given by Eq. 1.

$$M_p(n) = m(n) - 0.97m(n-1) \tag{1}$$

where $M_p(n)$ is the output pre-emphasis signal, $m(n)$ is the current value of music sound signal.

3.3.2 Frame Blocking

The input musical sounds are split into short-time frames (20 ms) and overlapped with 10 ms,

3.3.3 Windowing

To keep the continuity of the frame, we apply the a hamming window, $W(n)$, to each frame, $M_s(n)$, Eq. 2

$$M_w(n) = M_s(n) * W(n) \tag{2}$$

where N is the window length and $W(n) = 0.54 - 0.46(\frac{2n}{N})$.

3.3.4 Fractional Fourier Transform (FrFT)

FrFT of a signal $x(t)$ with order is represented by $F(u)$ which is represented by Eq. 3

$$F^{\alpha}(u) = \int_{-\infty}^{\infty} K_{\alpha}(t,u)\, dt \tag{3}$$

where

$$K_{\alpha}(u,t) = \begin{cases} \left[\frac{1-j\cot\alpha}{2\pi}\right] \exp j\left[\frac{t^2+u^2}{2}\cot\alpha - 2ut\csc\alpha\right], \text{if } \alpha \text{ is not a multiple of } \pi \\ \delta(u-t), \text{if } \alpha \text{ is a multiple of } 2\pi \\ \delta(u+t), \text{if } \alpha+\pi \text{ is a multiple of } \pi \end{cases} \tag{4}$$

The α makes an angle to the time axis and is called fractional order. It provides an additional degree of freedom and flexibility for processing of music signals [20, 21].

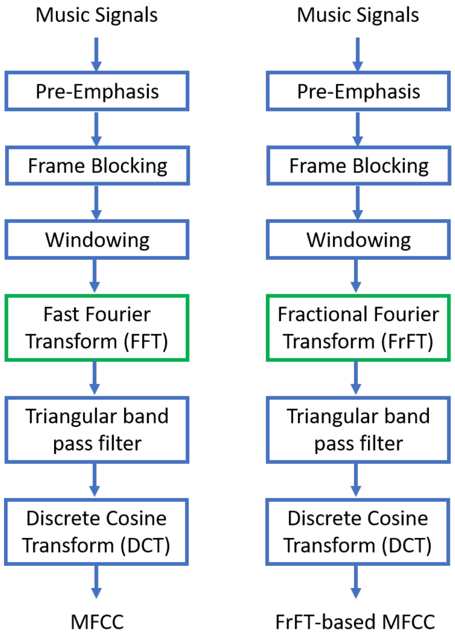

Music Signals
Pre-Emphasis
Frame Blocking
Windowing
Fast Fourier Transform (FFT)
Triangular band pass filter
Discrete Cosine Transform (DCT)
MFCC
Music Signals
Pre-Emphasis
Frame Blocking
Windowing
Fractional Fourier Transform (FrFT)
Triangular band pass filter
Discrete Cosine Transform (DCT)
FrFT-based MFCC

Figure 1: Process flow of MFCC and FrFFT-based MFCC.

3.3.5 Triangular Band Pass Filter

The power spectrum of the signals is multiplied by the magnitude response of a set of 26 triangular band pass filers. The positions of there filters are equally spaced along the mel frequency to mimic the non-linear human ear perception of sound, Eq. 5, 6

$$mel(f) = 2595\ \log_{10}\left(1 + \frac{f}{700}\right) \tag{5}$$

$$f = 700 \cdot (10^{\frac{mel}{2595}} - 1) \tag{6}$$

3.3.6 Discrete Cosine Transform (DCT)

DCT is used to decorrelate the filter bank coefficients and yield the mel-scale cepstral coefficients, Eq. 7

$$C_i(m) = \sum_{n=0}^{M-1} S_i(n) \cos\left[\pi m (\frac{n - 0.5}{M})\right] \tag{7}$$

where $0 \leq m \leq M$, M is the number of triangular band pass filters, $S_i(n)$ is cepstrum of the log filter bank energies and $C_i(m)$ is MFCC of i^{th} frame.

3.4 Model Architecture and Design: C-RNN

From different deep learning approaches, we focus on CNN-based models for instrument classification due to several reasons: (1) CNNs are more robust than conventional methods such as source separation with SVM, achieving 23.1% in performance improvement [1]. (2) Compared to one of the RNN structures, Long-Short Term Memory (LSTM), the CNN structure learns five times faster and is more accurate when doing instrument classification. (3) Arun Solanki and Sachin Pandey used eight layer CNN to train fixed-length music with a labeled predominant instrument and estimate an arbitrary number of instruments from an audio signal with variable length, resulting in 92.8% accuracy [22]. Moreover, we introduce RNN into the CNN model to extract temporal information from the dataset. In this work, we designed a C-RNN model and adopt a CNN model as baseline.

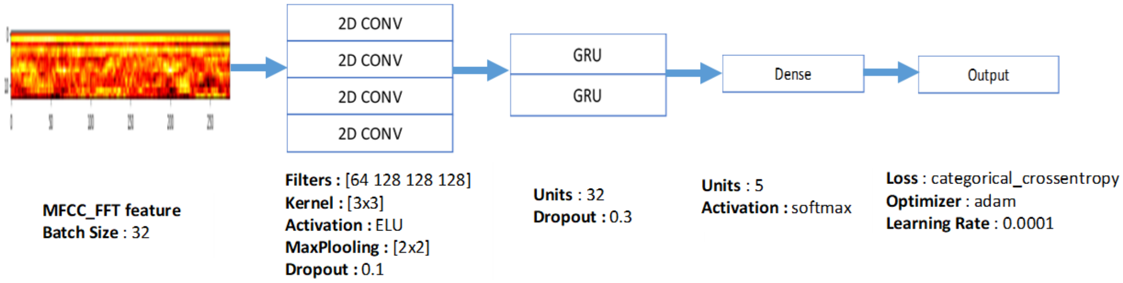


Figure 2: CRNN Architecture

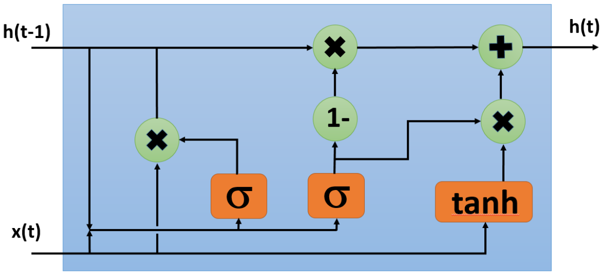


Figure 3: GRU unit

The pipeline of our model is shown in Fig. 2. The C-RNN consists of three parts: convolutional layers, recurrent layers, and fully-connected layers. The input are FrFT-based MFCC features with a batch size of 32.

3.4.1 Convolutional Layer

Four convolutional layers with batch normalization, max pooling, dropout and ELU layers are used in our model. Adding more convolutional layers makes performance better, and thus we implement four two-dimentional conv cell to reach a balance of performance and efficiency. The Exponential Linear Unit (ELU) activation function Eq. 8

$$R(z) = \begin{Bmatrix} z & z > 0 \\ \alpha(e^z - 1) & z \leq 0 \end{Bmatrix} \tag{8}$$

is used as a smooth alternative of the Rectified Linear Unit (ReLU) because of the better generalizing performance[23]. Batch normalization (BN) and dropout regularization are also included to improve generalization.

3.4.2 Recurrent Layer

Two recurrent layers are used to summarize temporal components. We chose Gated Recurrent Units (GRUs) rather than LSTM layers since they require fewer parameters to train and have similar performance. GRUs are improved version of standard recurrent neural network. 3 is a plot of a typical GRU cell. To solve the vanishing gradient problem of a standard RNN, GRU uses update gate and reset gate. It can keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction. The update gate helps the model to determine how much of the past information needs to be passed along to the future. On the other hand, the reset gate can let the model decide how much of the past information to forget. Additionally, we also add a dropout layer for regularization.

3.4.3 Fully-Connected Layer

Fully-connected layers are responsible for getting input from all the layer combining them for the classification. These layers with softmax activation provide the class score of each musical instrument.

3.4.4 Training Considerations

Musical instrument can be treated as a multi-class classification problem, and thus categorical cross-entropy is selected as the loss function in our work. Adam is chosen as the optimizer because it has shown the state of the art performance in convolution-based classification. We optimize the learning rate to 0.0001 to avoid the oscillation on the validation accuracy curve and improve training stability. Besides, BN layer could help improve stability as well. Early stopping callback is also used, with a patience of 10, to prevent overfitting. In addition, the chosen of batch size 32 is due to the result that it made the training converge very fast.

4 Results

4.1 ICA

We first evaluated the separation performance of ICA. The performance of both clean mixed audio (Clean dataset) and noisy mixed audio (Noisy dataset) were great. The different instruments music were separated perfectly. The only difference between them was that one can still hear a little noise in the results of noisy mixed data. Fig. 4 is the waveform of the noisy mixed music. The five subfigures in Fig. 5 are the waveforms of separated audios from ICA.

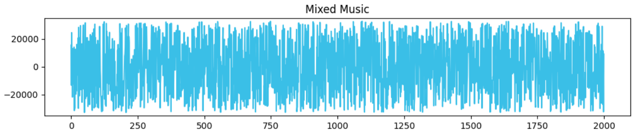
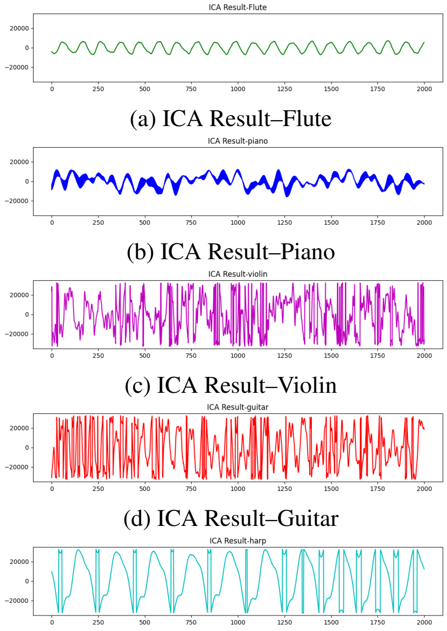


Figure 4: Noisy Mixed Music Waveform



(a) ICA Result–Flute

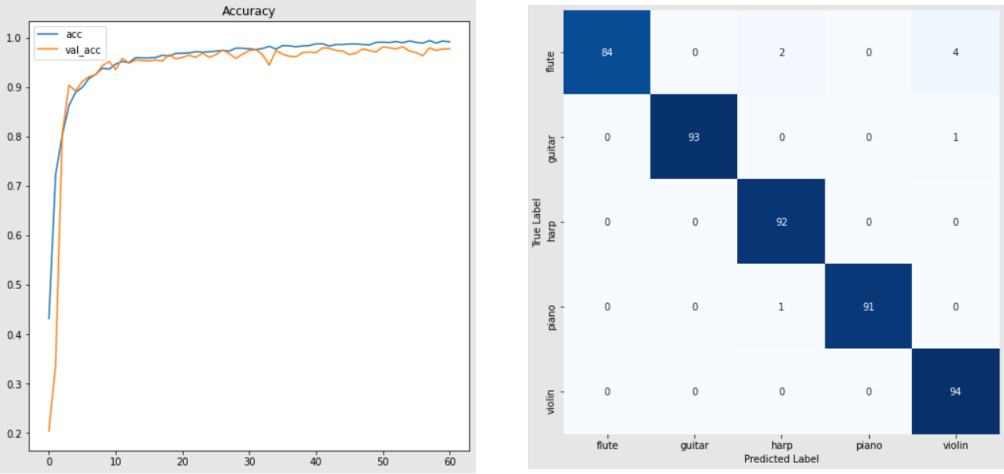
(b) ICA Result–Piano

(c) ICA Result–Violin

(d) ICA Result–Guitar

(e) ICA Result–Harp

Figure 5: The separation result of ICA



(a) training curve

(b) Confusion matrix

Figure 6: Accuracy history and confusion matrix of our model

Table 1: Optimal value of α and acc, val_acc

α	0.93	0.95	0.96	0.97	0.98	0.99
acc	0.9736	0.9773	0.9796	0.9804	0.9808	0.9854
val_acc	0.9488	0.9606	0.9627	0.9723	0.9712	0.9712

4.2 C-RNN

4.2.1 Model Training and Evaluation

The model was trained by using based on our Noisy dataset. The dataset were split into 70% training, 20% validation and 10% test data. The training result, Fig. 6a, shows that the model converged well and fast, and the training process early stopped at around 60 epochs, indicating that the overfitting didn't occur. The model performs very well, with a validation accuracy of 0.978. Fig. 6b shows our testing result. Almost all predicted labels are correct, with only a few trivial errors, such as the between violins and flutes. In addition, the training and testing results of the Clean dataset reach an accuracy of 0.996, which is better than that of the Noisy dataset. These results also show that our C-RNN model is robust to noise.

4.2.2 Comparison

We compared our C-RNN model with a CNN model using both dataset: Clean and Noisy. The CNN model adopted the same CNN layers and regularization hyperparameters as the C-RNN model; and the recurrent layers were replaced by dense layers. Fig. 7 shows that CNN model performs well, with an accuracy of over 0.95, but our C-RNN do even better by and improvement of 2 percentages. These results were collected by an average over dozen times of training.

For the feature processing, we tested several value of α to generate the best features for the model. It was observed that the maximum accuracy is obtained for $\alpha = 0.93 \sim 0.99$. Table 1 reveals the accuracy based on different α, from 0.93 to 0.99. The performance of α over 0.97 are similar; while α lower than 0.95 could lead to worse accuracy. Thus, we choose the optimal value 0.97 in our work since it provided the best validation accuracy.

5 Conclusion

In the article, we present a noise-robust architecture which combines ICA and C-RNN for musical instrument classification. We prepared two mixed musical dataset (Clean and Noisy) of five different musical instruments for training and testing our architecture. ICA was first adopted to extract the

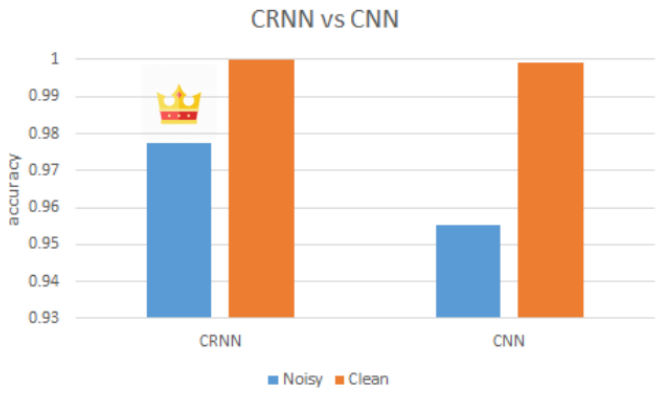


Figure 7: Comparison between CNN

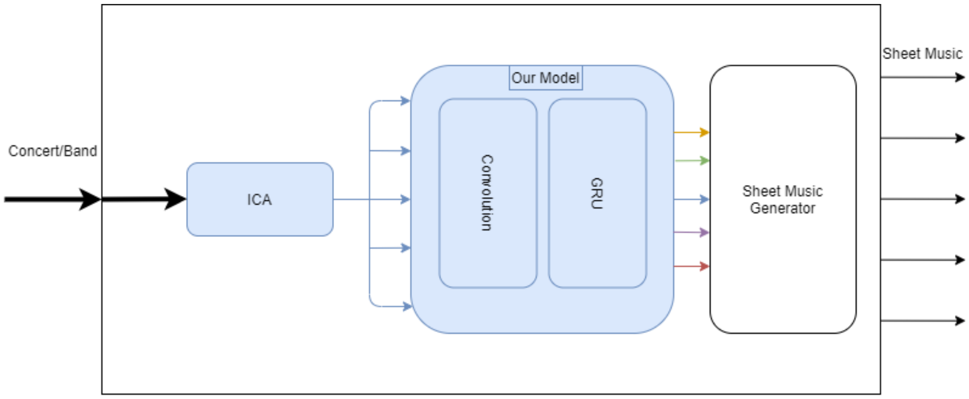


Figure 8: Pipeline of the sheet music generator application

independent audio sources from the dataset. Then, the features of each independent audio source were extracted by FrFT-based MFCC. A C-RNN classifier was designed and implemented for classifying the musical instrument of each independent audio source. The performance of our model can reach a validation accuracy of 0.978 and 0.996 for our Noisy and Clean dataset, respectively. The aim of the work is to evaluate the performance of our architecture, and, therefore, only five representative instruments were used. The result shows that our method works well in this classification task. In the future, to make the model more practicable and reliable, more kinds of instruments will be adopted to this architecture.

Broader Impact

Our ultimate goal is to create a mobile application that can generate the sheet music for every instrument once the audio source containing that corresponding instrument is given. Fig. 8 describes the pipeline of the application we want to build. The blue blocks in the diagram are implemented in our submitted project and it performs the core task of classification of instruments according the input audio stream. In the future, we plan to build a complete front-end which can produce the sources ICA requires once the single audio source (e.g., concert music) is given, so that users do not need to manually create five independent sources from different microphones, which introduces non-ideal characteristics and could potentially hurt model performance. In addition, we aim to train a music sheet generator (i.e., back-end of our project) which is able to generate comprehensive sheet music for instruments making an appearance in the audio stream. This application allows music enthusiasts to obtain music sheet of the instruments they are particularly interested in without too much hassle. Sheet music for popular tunes are available online but at a price and our extension of our final project aims to make these sheet music more accessible to the general public.

Acknowledgments and Disclosure of Funding

Tse-Shao Chang worked on data collection, data (pre)processing and feature extraction. Weichin Chien worked on data collection, ICA, testing data generation, and part of model design. Leo and HaoTsung developed classification pipeline and performed hyperparameter search. All co-authors were involved in fine tuning and writing this report. All co-authors equally contributed to this project.

Reference

[1] Y. Han, J. Kim and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[2] S. E. et al, "Musical instrument recognition on solo performances," *2004 12th European signal processing conference*, p. 1284–1286, 2004.

[3] T. Heittola, A. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," *Proc. Int. Soc. Music Inf. Retrieval Conf.*, p. 327–332, 2009.

[4] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, p. 155–155, 2007.

[5] Z. Duan, B. Pardo, and L. Daudet, "A novel cepstral representation for timbre modeling of sound sources in polyphonic mixtures," *Proc. 2014 IEEE Int. Conf. Acoust., Speech Signal Process.*, p. 7495–7499, 2014.

[6] B. David and G. Richard, "efficient musical instrument recognition on solo performance music using basic features," *journal of the audio engineering society*, june 2004.

[7] J. D. Deng, C. Simmermacher, and S. Cranefield, "A study on feature analysis for musical instrument classification," *Trans. Sys. Man Cyber. Part B*, vol. 38, p. 429–438, Apr. 2008.

[8] M. Müller, D. P. Ellis, A. Klapuri, and G. Richard, "Signal Processing for Music Analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, Oct. 2011.

[9] P. S. Jadhav, "Classification of musical instruments sounds by using mfcc and timbral audio descriptors," *International Journal on Recent and Innovation Trends in Computing and Communication*, 2015.

[10] M. E. Özbek, N. Özkurt, and F. A. Savacı, "Wavelet ridges for musical instrument classification," *Journal of Intelligent Information Systems*, vol. 38, 2012.

[11] T. Ramalingam and P. Dhanalakshmi, "Speech/music classification using wavelet based feature extraction techniques," *J. Comput. Sci.*, vol. 10, pp. 34–44, 2014.

[12] D. G. Bhalke, C. B. R. Rao, and D. S. Bormane, "Automatic musical instrument classification using fractional fourier transform based- mfcc features and counter propagation neural network," *Journal of Intelligent Information Systems*, 2016.

[13] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *CoRR*, vol. abs/1605.09507, 2016.

[14] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *International Journal of Information Technology*, vol. abs/1605.09507, 2019.

[15] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning Adaptive Computation and Machine Learning series*. MIT Press, 2016, 2016.

[16] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio., "Generating sentences from a continuous space," *SIGNLL Conference on Computational Natural Language Learning (CONLL), 2016*, 2015.

[17] Z. Nasrullah and Y. Zhao., "Music artist classification with convolutional recurrent neural networks," *IJCNN*, 2019.

[18] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[19] J. Wellhausen and V. Gnann., “Independent component analysis for audio signal separation,” *SPIE*, 2005.

[20] D. H. Bailey and P. N. Swarztrauber, “The fractional fourier transform and applications,” *SIAM Review*, 1990.

[21] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform: with Applications in Optics and Signal Processing*. New York: John Wiley and Sons, 2001.

[22] S. P. A. Solanki, “Music instrument recognition using deep convolutional neural networks.,” *Int. j. inf. tecnol.*, 2019.

[23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *ICLR*, 2016.