INSTRUDIVE: A MUSIC VISUALIZATION SYSTEM BASED ON AUTOMATICALLY RECOGNIZED INSTRUMENTATION

Takumi Takahashi[1,2] **Satoru Fukayama**[2] **Masataka Goto**[2]

[1] University of Tsukuba, Japan
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan
s1720822@s.tsukuba.ac.jp, {s.fukayama, m.goto}@aist.go.jp

ABSTRACT

A music visualization system called *Instrudive* is presented that enables users to interactively browse and listen to musical pieces by focusing on instrumentation. Instrumentation is a key factor in determining musical sound characteristics. For example, a musical piece performed with vocals, electric guitar, electric bass, and drums can generally be associated with *pop/rock* music but not with *classical* or *electronic*. Therefore, visualizing instrumentation can help listeners browse music more efficiently. Instrudive visualizes musical pieces by illustrating instrumentation with multi-colored pie charts and displays them on a map in accordance with the similarity in instrumentation. Users can utilize three functions. First, they can browse musical pieces on a map by referring to the visualized instrumentation. Second, they can interactively edit a playlist that showing the items to be played later. Finally, they can discern the temporal changes in instrumentation and skip to a preferable part of a piece with a multi-colored graph. The instruments are identified using a deep convolutional neural network that has four convolutional layers with different filter shapes. Evaluation of the proposed model against conventional and state-of-the-art methods showed that it has the best performance.

1 INTRODUCTION

Since multiple musical instruments having different timbres are generally used in musical pieces, *instrumentation* (combination or selection of musical instruments) is a key factor in determining musical sound characteristics. For example, a song consisting of vocals, electric guitar, electric bass, and drums may sound like *pop/rock* or *metal* but not *classical* or *electronic*. Consider, for example, a listener who appreciates *gypsy jazz* (featuring violin, acoustic guitar, clarinet, and double bass). How can he/she discover similar-sounding music? Searching by instrumentation can reveal musical pieces played with the same, slightly different, or completely different instrumentation, corresponding to his/her preferences.

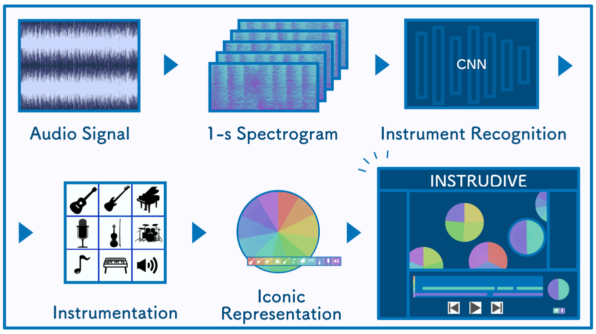

Figure 1: Overview of Instrudive music visualization system.

Instrumentation is strongly connected with musical sound and genres but is not restricted to a specific genre. For example, *pop/rock*, *funk*, and *fusion* are sometimes played with similar instrumentation. Therefore, it can be helpful for listeners to overcome the confinements of a genre by focusing on sound characteristics when searching for similar-sounding music.

To let users find musical pieces that they prefer, various methods and interfaces for retrieving and recommending music have been proposed. They are generally categorized into three approaches: bibliographic retrieval based on the metadata of musical pieces, such as artist, album, year of release, genres, and tags [2], music recommendation based on collaborative filtering using playlogs [5, 38], and music recommendation/retrieval based on content-based filtering using music analysis, such as genre classification [14, 30] and auto-tagging [4, 14, 20]. Music interfaces leveraging automatic instrument recognition [22] have received less attention from researchers.

We have developed a music visualization system called *Instrudive* that automatically recognizes the instruments used in each musical piece of a music collection, visualizes the instrumentations of the collection, and enables users to browse for music that they prefer by using the visualized instrumentation as a guide (Figure 1). Instrudive visualizes each musical piece as a pie-chart icon representing the duration ratio of each instrument that appears. This enables a user to see which instruments are used and their relative amount of usage before listening. The icons of

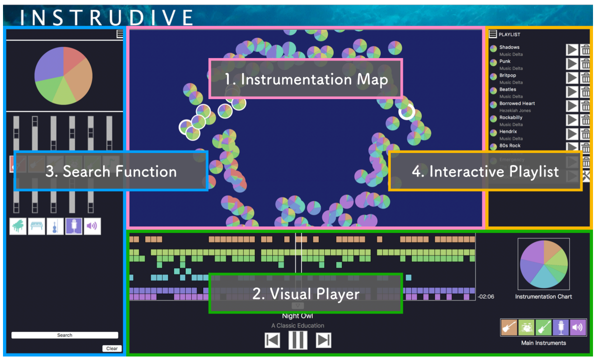

Figure 2: *Instrudive* interface consists of four parts.

all musical pieces in a collection are arranged in a two-dimensional space with similar-instrumentation pieces positioned in close proximity. This helps the user listen to pieces having similar instrumentation. Furthermore, the user can create a playlist by entering a pie-chart query to retrieve pieces having instrumentation similar to the query and listen to a musical piece while looking at a timeline interface representing when each instrument appears in the piece.

In the following section, we describe previous studies on music visualization and instrument recognition. We then introduce the usage and functions of Instrudive in Section 3 and explain its implementation in Section 4. Since the main contributions of this work are not only the Instrudive interface but also a method for automatically recognizing instruments on the basis of a deep convolutional neural network (CNN), we explain the recognition method and experimental results in Section 5. After discussing the usefulness of the system in Section 6, we summerize the key points and describe future work in Section 7.

2 RELATED WORK

2.1 Music Visualization

Visualization of music by using audio signal processing has been studied by many researchers.

Given a large collection of musical pieces, a commonly used approach is to visualize those pieces to make it easy to gain an overview of the collection [11, 13, 23, 24, 31, 32, 37, 40]. The collection is usually visualized so that similar pieces are closely arranged [13, 23, 24, 31, 32, 37]. The visualization helps listeners to find and listen to musical pieces they may prefer by browsing the collection. Instrumentation is not focused on in this approach, whereas Instrudive visualizes the instrumentations of the pieces in the collection by displaying pie-chart icons for the pieces in a two-dimensional space as shown in Figure 2.

Given a musical piece, a commonly used approach is to visualize the content of the piece by analyzing the musical elements [3, 9, 10, 12, 18, 29]. For example, a repetitive music structure is often visualized [3,9,10,12,29]. This enhances the listening experience by making listeners aware of the visualized musical elements. Our Instrudive interface also takes this approach. After a user selects a musical piece, Instrudive displays a timeline interface representing when each musical instrument appears in the piece. This helps the listener focus on the instrumentation while listening to music.

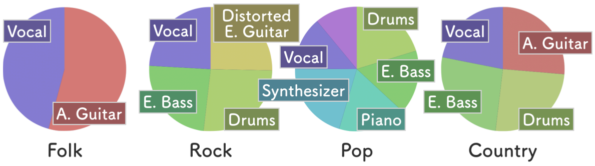

Figure 3: Multi-colored pie charts depict instrumentation.

2.2 Instrument Recognition

The difficulty in recognizing instruments depends on the number of instruments used in the piece. The greater the number of instruments, the greater the difficulty. When a single instrument is used in a monophonic recording, many methods achieve good performance [6, 8, 19, 41, 42].

On the other hand, when many instruments are used in a polyphonic recording, which is typical in popular music produced using multitrack recording, it is more difficult to recognize the instruments. Most previous studies [7, 15, 22, 26] used machine learning techniques to overcome this difficulty. In Section 5, we compare our proposed model of instrument recognition with one that uses a support vector machine (SVM).

A more recent approach to recognizing instruments is to use a deep learning method, especially a CNN [16, 27, 28, 34]. Methods using this approach have outperformed conventional and other state-of-the-art methods, but their performances cannot be easily compared due to the use of different databases and instrument labels. Despite their high performance, there is room for improvement in their accuracy. We aim to improve accuracy by proposing and implementing an improved CNN-based method.

3 INSTRUDIVE

Instrudive enables users to browse musical pieces by focusing on instrumentation. The key idea of visualizing the instrumentation is to use a multi-colored pie chart in which different colors denote the different instruments used in a musical piece. The ratios of the colors indicate relative durations in which the corresponding instruments appear. Figure 3 shows example charts created using ground truth annotations from the multitrack MedleyDB dataset [1]. The charts representing different genres have different appearances due to the differences in instrumentation among genres.

These multi-colored pie charts help a user browsing a collection of musical pieces to understand the instrumentations before listening to the pieces. Moreover, during the playing of a musical piece, Instrudive displays a multi-colored graph that indicates the temporal changes in instrumentation.

Instrudive can recognize 11 categories of instruments: acoustic guitar, clean electric guitar, distorted electric gui-

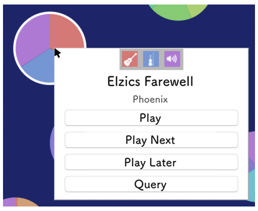

Figure 4: Menu appears after right-clicking chart.

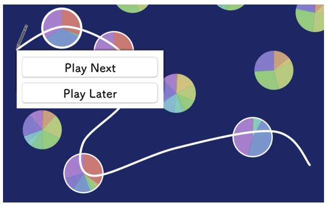

Figure 5: Scattering mode enables playlist to be created by drawing curve.

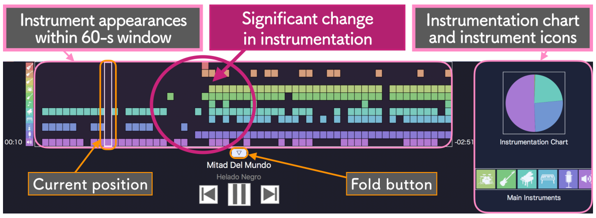
Instrument appearances within 60-s window
Significant change in instrumentation
Instrumentation chart and instrument icons
00:10
-02:51
Instrumentation Chart
Mitad Del Mundo
Helado Negro
Current position
Fold button
Main Instruments

Figure 6: Visual player helps listener understanding instrumentation and its temporal changes.

tar, drums, electric bass, fx/processed sound (sound with effects), piano, synthesizer, violin, voice, and *other* (instruments not included in the 10 categories). The categories depend on this dataset and are defined on the basis of [27].

As shown in Figure 2, the interface of Instrudive consists of four parts: an instrumentation map for browsing musical pieces, a visual player for enhancing the listening experience, a search function for finding musical pieces by using the pie-chart icons as queries, and an interactive playlist for controlling the order of play.

3.1 Instrumentation Map

The instrumentation map visualizes the musical pieces in a collection. Each piece is represented by a multi-colored pie chart. Similar pie charts are closely located in a two-dimensional space. As shown in Figure 9, this map supports visualization modes, *circular* and *scattering*.

When a user right-clicks on a pie chart, a menu appears as shown in Figure 4. The user can play the piece or use the piece as a query for the search function. By using the circular mode, which arranges the pie charts in a circular path, the user can automatically play the pieces with similar instrumentation one after another along the path. By switching to the scattering mode, the user can draw a curve to create a playlist consisting of pieces on the curve as shown in Figure 5.

3.2 Visual Player

The visual player (Figure 6) visualizes the temporal changes in instrumentation in the selected musical piece as it is played. It shows a graph along the timeline interface consisting of a number of colored rectangular tiles, each of which denotes activity (i.e., presence) of the corresponding instrument. As the musical piece is played, this activity graph (covering a 60-s window) is automatically scrolled to continue showing the current play position.

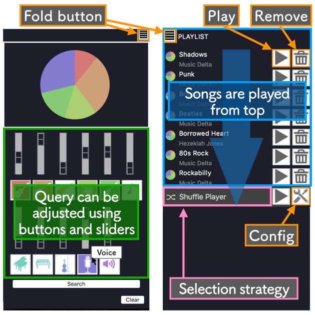

Figure 7: Interfaces for search menu and playlist.

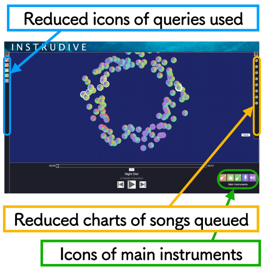
Reduced icons of queries used
INSTRUDIVE
Reduced charts of songs queued
Icons of main instruments

Figure 8: Simplified interface for novice users.

The user can interactively change the play position by left-clicking on another position on the graph. The graph enables the user to anticipate how the instrumentation will change. For example, a significant change in instrumentation can be anticipated, as shown in Figure 6

The pie chart on the right side of Figure 6 represents the instruments currently being played and changes in synchronization with the playing of the piece. The *instrument icons* shown below the chart are consistently shown in the same color, enabling the user to easily distinguish them. By hovering the mouse over an icon, the user can see the name of the instrument.

3.3 Search Function

The search function (left side of Figure 7) enables the user to retrieve pieces by entering a query. Pressing an instrument-icon button intensifies its color, so the selected button is clearly evident. The ratio of instruments in the query can be adjusted by moving the sliders.

When the *search* button is pressed, the system retrieves musical pieces with instrumentation similar to that of the query by using the search algorithm described in Section 4.3. The retrieved pieces are not only highlighted on the map as shown in Figure 10 but also instantly added to the playlist.

3.4 Interactive Playlist

The interactive playlist (right side of Figure 7) shows a list of the retrieved or selected musical pieces along with their pie charts, titles, and artist names. The user can change their order, add or delete a piece, and play a piece.

A musical piece disappears from the playlist after it has been played. If no piece is in the list, the next piece is selected automatically. In circular mode, the available play strategies are *clockwise* (pieces are played in clockwise order), and *shuffle* (pieces are played randomly). In scattering mode, the available play strategies are *shuffle* and *nearest* (pieces nearby are played). The user can thus play pieces having similar or different instrumentation.

3.5 Simplified Interface

We also prepared a simplified interface for novice users who are not familiar with music instrumentation. As shown in Figure 8, the visual player, the search function,

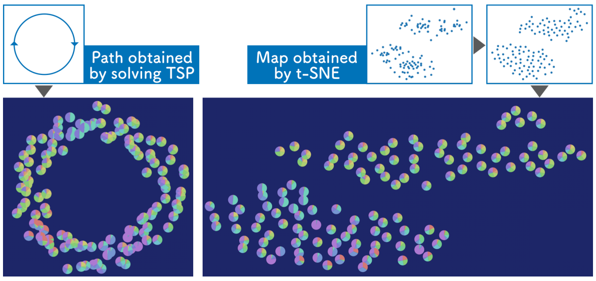

Figure 9: Two algorithms are used to create maps. Map on left is used in *circular* mode; map on right is used in *scattering* mode.

and the interactive playlist can be folded to the side to let the user concentrate on simple interaction using the instrumentation map.

4 IMPLEMENTATION OF INSTRUDIVE

The Instrudive interfaces were mainly programmed using a Python library *Tkinter* and executed on Mac OS X. After the instruments were recognized, as described in Section 5, the results were stored and used for the interfaces.

4.1 Iconic Representation

A multi-colored pie chart of a musical piece with length T s is displayed by computing the *absolute appearance ratio (AAR)* and the *relative appearance ratio (RAR)* for each instrument i ($\in \boldsymbol{I}$: recognized instrument categories).

The result of recognizing an instrument i is converted into AAR_i:

$$AAR_i = \frac{t_i}{T}, \tag{1}$$

where t_i ($\leq T$) s is the total of all durations in which instrument i is played. AAR represents the ratio of this total time against the length of the musical piece.

$$RAR_i = \frac{AAR_i}{\sum_i AAR_i} \tag{2}$$

represents the ratio of this total time against the total time of the appearances of all instruments. After RAR_i is computed for all instruments, an $|\boldsymbol{I}|$-dimensional vector (11-dimensional vector in the current implementation) summarizing the instrumentation of the piece is obtained. The pie chart is a visual representation of this vector: RAR_i is used as an area ratio in the circle for the corresponding instrument.

4.2 Mapping Algorithms

To visualize musical pieces in *circular* mode (Figure 9), we use an $|\boldsymbol{I}|$-dimensional vector (11-dimensional vector in the current implementation) of AAR. The AAR vectors for all the pieces are arranged on a circular path obtained by solving the traveling salesman problem (TSP) [25] to find the shortest route for visiting all pieces. After assigning all the pieces on the path, we scatter them randomly towards and away from the center of the circle so that the pie charts are not located too close together.

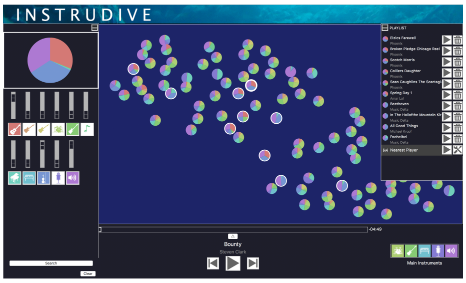

Figure 10: Top ten search results are highlighted and added to playlist. Users can check contents of results before listening.

Layer	Output size
Magnitude spectrogram	$1024 \times 87 \times 1$
Conv (4×1)	$1024 \times 87 \times 32$
Pool (5×3)	$204 \times 29 \times 32$
Conv (16×1)	$204 \times 29 \times 64$
Pool (4×3)	$51 \times 9 \times 64$
Conv (1×4)	$51 \times 9 \times 64$
Pool (3×3)	$17 \times 3 \times 64$
Conv (1×16)	$17 \times 3 \times 128$
Pool (2×2)	$8 \times 1 \times 128$
Dropout (0.5)	1024
Dense	1024
Dense	121
Dense	11

Table 1: Proposed CNN architecture.

To visualize musical pieces in *scattering* mode, the 11-dimensional AAR vectors are projected onto a two-dimensional space by using t-distributed stochastic neighbor embedding (t-SNE) [39], which is an algorithm for dimensionality reduction frequently used to visualize high-dimensional data. Since similar pie charts are often located too close together, we slightly adjust their positions one by one by randomly moving them until all the charts have a certain distance from each other.

4.3 Search Algorithms

Since both a query and a musical piece can be represented as 11-dimensional AAR vectors, we can simply compute the cosine similarity between the query and each musical piece in the collection. In Figure 10, for example, given a query containing acoustic guitar, violin, and others, the retrieved pieces ranked higher have similar pie charts. As the rank gets lower, the charts gradually becomes less similar.

5 INSTRUMENT RECOGNITION

5.1 Pre-processing

Each musical piece was converted into a monaural audio signal with a sampling rate of 44100 Hz and then divided into one-second fragments. To obtain a one-second magnitude spectrogram, we applied short-time Fourier transform (STFT) with a window length of 2048 and a hop size of 512. We then standardized each spectrogram to have zero mean and unit variance. As a result, each one-second spectrogram had 1024 frequency bins and 87 time frames.

5.2 CNN Architecture

We compared several CNN models; the one that showed the best performance is summerized in Table 1. The model mainly consists of four convolutional layers with max-pooling and ReLU activation. A spectrogram represents the structure of frequencies with one axis and its temporal changes against the other axis, which is unlike an image that represents spatial information with both axes. We set the shape of each layer to have length along only one axis (frequency or time). For convolutions, feature maps were padded with zeros so that dimensionality reduction was done only by using max-pooling layers. By doing this, we could use various shapes of layers and their combinations without modifying the shapes of other layers. After a 50% dropout was applied to prevent overfitting, two dense layers with ReLU and an output dense layer with a sigmoid function were used to output an 11-dimensional vector. Batch normalization [17] was applied to each of the convolutional and dense layers. In training, we used the Adam algorithm [21] as the optimizer and binary cross-entropy as the loss function. The mini-batch size was 128, and the number of epochs was 1000.

This proposed CNN model outputs 1-s instrument labels as a vector. By gathering the vectors corresponding to each musical piece, we can represent each musical piece as a sequence of 11-dimensional vectors (instrument labels/activations), which are used to calculate the instrumentation described in Section 4.

5.3 Dataset

To evaluate the proposed CNN model and apply it to *Instrudive*, we used the MedleyDB dataset [1]. This dataset has 122 multitrack recordings of various genres and instrument activations representing the sound energy for each stem (a group of audio sources mixed together), individually calculated along with time frames with a hop size of 46.4 ms.

We generated instrument labels and split the data on the basis of the source code published online [27]. We used the 11 categories listed in Section 3 based on the ground truth annotations from the multitrack MedleyDB dataset [1]. Since our system does not depend on these categories, it can be generalized to any set of categories given any dataset.

The 122 musical pieces were divided into five groups by using the algorithm in [35] so that the instrument labels were evenly distributed among the five groups. Four of the groups were used for training, and the fifth was used for evaluation. All the musical pieces that appear in Instrudive were included in the data used for evaluation, and their instrumentations were predicted using cross validation.

5.4 Baseline

For comparison with our model, we used a conventional bag-of-features method, a state-of-the-art deep learning method with mel-spectrogram input, and a state-of-the-art deep learning method with raw wave input.

Layer	Output size
Mel-spectrogram	128 × 43 × 1
Conv (3 × 3)	130 × 45 × 32
Conv (3 × 3)	132 × 47 × 32
Pool (2 × 2)	44 × 15 × 32
Dropout (0.25)	44 × 15 × 32
Conv (3 × 3)	46 × 17 × 64
Conv (3 × 3)	48 × 19 × 64
Pool (2 × 2)	16 × 6 × 64
Dropout (0.25)	16 × 6 × 64
Conv (3 × 3)	18 × 8 × 128
Conv (3 × 3)	20 × 10 × 128
Pool (2 × 2)	6 × 3 × 128
Dropout (0.25)	6 × 3 × 128
Conv (3 × 3)	8 × 5 × 256
Conv (3 × 3)	10 × 7 × 256
Global pool	1 × 1 × 256
Dense	1024
Dropout (0.5)	1024
Dense	11

Table 2: Han's architecture.

Layer	Output size
Raw wave	44100 × 1
Conv (3101)	41000 × 256
Pool (40)	2049 × 256
Conv (300)	1750 × 384
Pool (30)	87 × 384
Conv (20)	68 × 384
Pool (8)	16 × 384
Dropout (0.5)	16 × 384
Dense	400
Dense	11

Table 3: Li's architecture.

5.4.1 Bag-of-features

For the bag-of-features method, we used the features described by [15], consisting of 120 features obtained by computing the mel-frequency cepstral coefficients and 16 spectral features [33]. We trained an SVM with a radial basis function (RBF) kernel by feeding it these 136 features.

5.4.2 Mel-spectrogram (Han's CNN model)

For the deep learning method with mel-spectrogram input, we used Han's CNN architecture [16] (Table 2). This architecture is based on VGGNet [36], a commonly used model in the image processing field. Each one-second fragment of the audio signal was resampled into 22050 Hz, converted into a mel-spectrogram, and standardized. Every activation function was LReLU ($\alpha = 0.33$) except the output sigmoid.

In preliminary experiments, training this model failed in almost 700 epochs due to a gradient loss. Therefore, we applied batch normalization to each of the convolutional and dense layers, enabling us to successfully complete 1000 epochs of training. We also used 500 epochs, but the performance was worse than for 1000.

5.4.3 Raw Waveform (Li's CNN model)

For the deep learning method with raw wave input, we used Li's CNN model in [27] (Table 3). This model performs end-to-end learning using a raw waveform. We standardized each one-second fragment of the monaural audio signal obtained in pre-processing. Every activation function was ReLU except the output sigmoid. Batch normalization was again applied to each layer. We trained the model with 1000 epochs.

5.5 Metrics

We evaluated each model using four metrics: *accuracy*, *F-micro*, *F-macro*, and *AUC*.

Accuracy was defined as the ratio of predicted labels that exactly matched the ground truth. Each label predicted by the CNN at every one-second fragment in all pieces was

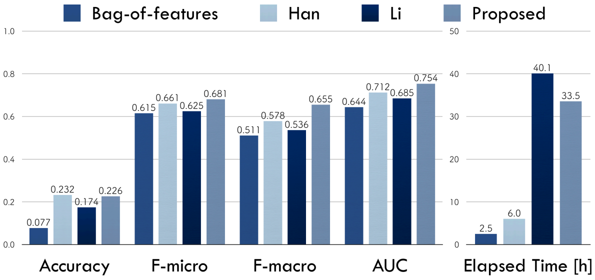

Figure 11: Proposed model showed best performance for F-micro, F-macro, and AUC but took five times longer to complete training than Han's model, which showed second-best performance.

an 11-dimensional vector of likelihoods. Since each likelihood ranged between 0 and 1, we rounded it to an integer (0 or 1) before matching.

The F-micro was defined as the micro average of the F1 measure for all predicted labels over the 11 categories. The F1 measure is defined as the harmonic mean of recall and precision and is widely used in multi-label classification tasks. Since it is calculated immediately without considering the categories, if some instruments frequently appear, their predicted labels considerably affect the F-micro.

The F-macro was defined as the macro average with each instrument equally considered. For each of the 11 categories, the F1 measure of the predicted labels was first calculated. Then, the average of the resulting 11 values was calculated as the F-macro.

The area under the curve (AUC) of the receiver operating characteristic was first calculated for each category. Then, the macro average of the resulting 11 values was used as the AUC in our multi-label task.

5.6 Results

As shown in Figure 11, the proposed model outperformed the other models in terms of AUC, F-micro, and especially F-macro, which was about 8% better than the next-best model (Han's model). This indicates that our model has higher generic performance and is more powerful in dealing with various kinds of instruments.

Interestingly, all of the deep learning methods showed significantly higher accuracy than the bag-of-features method. Since the accuracy cannot be increased with predictions made through guesswork, such as predicting classes that frequently appear, the deep learning methods are more capable of capturing the sound characteristics of instruments in sound mixtures.

The proposed model took five times longer to complete training than Han's model. This is because Han's model took advantage of using a more compact mel-spectrogram (128 × 87) than the raw spectrogram (1024 × 87) used for the proposed model. Since using a mel-spectrogram results in losing more information, the performance was worse.

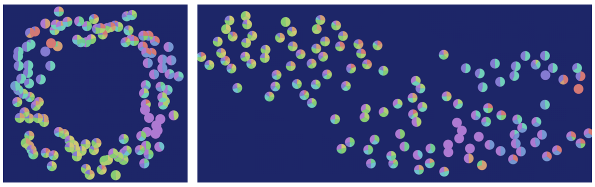

Figure 12: Maps created using ground truth data.

6 DISCUSSION

6.1 Smoothing Transitions Between Listening States

Our observations during testing showed that the use of *Instrudive* helped smooth the transition between listening states. Although the music was often passively listened to, the listeners sometimes suddenly became active when the time came to choose the next piece. In the *circular* mode of Instrudive, for example, the *clockwise player* played a piece that had instrumentation similar to the previous one. Since the sound characteristics were changing gradually, a user was able to listen to various genres in a passive state. If non-preferred music started playing, the user skipped to a different type of music by using the *shuffle player*. In addition, the user actively used the search function to access pieces with similar instrumentation and enjoyed looking at the temporal changes in the activity graph.

6.2 Studies from Ground Truth Data

We compared maps created using the automatically recognized (predicted) data (Figure 9) with maps created using the ground truth data (Figure 12). Although they are similar to some extent, the contrast of the color distributions is much more vivid for the ground truth data, suggesting that the performance of our CNN model still has room for improvement. Since the proposed Instrudive interface is independent of the method used for instrument recognition, we can simply incorporate an improved model in the future.

7 CONCLUSION

Our Instrudive system visualizes the instrumentations of the musical pieces in a collection for music discovery and active music listening. The first main contribution of this work is showing how instrumentation can be effectively used in browsing musical pieces and in enhancing the listening experience during playing of a musical piece. The second main contribution is proposing a CNN model for recognizing instruments appearing in polyphonic sound mixtures that achieves better performance than other state-of-the-art models.

We plan to conduct user studies of Instrudive to analyze its nature in more detail and to test different shapes of filters to analyze the reasons for the superior performance of our CNN model. We are also interested in investigating the scalability of our approach by increasing the number of musical pieces and allowing a greater variety of instruments.

8 ACKNOWLEDGMENTS

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

9 REFERENCES

[1] Rachel M. Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 155–160, 2014.

[2] Dmitry Bogdanov and Perfecto Herrera. How much metadata do we need in music recommendation? A subjective evaluation using preference sets. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pages 97–102, 2011.

[3] Mattew Cooper and Jonathan Foote. Automatic music summarization via similarity analysis. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, 2002.

[4] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2014)*, pages 6964–6968, 2014.

[5] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2010.

[6] Antti Eronen and Aussi Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2000)*, volume 2, pages 753–756, 2000.

[7] Slim Essid, Gaël Richard, and Bertrand David. Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):68–80, 2006.

[8] Slim Essid, Gaël Richard, and Bertrand David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006.

[9] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the Seventh ACM International Conference on Multimedia (ACM Multimedia 1999)*, pages 77–80, 1999.

[10] Masataka Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1783–1794, 2006.

[11] Masataka Goto and Takayuki Goto. Musicream: Integrated music-listening interface for active, flexible, and unexpected encounters with musical pieces. *IPSJ Journal*, 50(12):2923–2936, 2009.

[12] Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. Songle: A web service for active music listening improved by user contributions. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pages 311–316, 2011.

[13] Masahiro Hamasaki and Masataka Goto. Songrium: A music browsing assistance service based on visualization of massive open collaboration within music content creation community. In *Proceedings of the 9th International Symposium on Open Collaboration (ACM WikiSym + OpenSym 2013)*, pages 1–10, 2013.

[14] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 339–344, 2010.

[15] Philippe Hamel, Sean Wood, and Douglas Eck. Automatic identification of instrument classes in polyphonic and poly-instrument audio. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 399–404, 2009.

[16] Yoonchang Han, Jaehun Kim, and Kyogu Lee. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):208–221, 2017.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[18] Dasaem Jeong and Juhan Nam. Visualizing music in its entirety using acoustic features: Music flowgram. In *Proceedings of the International Conference on Technologies for Music Notation and Representation*, pages 25–32, 2016.

[19] Ian Kaminskyj and Tadeusz Czaszejko. Automatic recognition of isolated monophonic musical instrument sounds using kNNC. *Journal of Intelligent Information Systems*, 24(2):199–221, 2005.

[20] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *Processings of the 14th Sound and Music Computing Conference (SMC 2017)*, 2017.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Instrogram: Probabilistic representation of instrument existence for polyphonic music. *IPSJ Journal*, 2(1):279–291, 2007.

[23] Peter Knees, Markus Schedl, Tim Pohle, and Gerhard Widmer. An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web. In *Proceedings of the 14th ACM International Conference on Multimedia (ACM Multimedia 2006)*, pages 17–24, 2006.

[24] Paul Lamere and Douglas Eck. Using 3D visualizations to explore and discover music. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pages 173–174, 2007.

[25] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

[26] Pierre Leveau, David Sodoyer, and Laurent Daudet. Automatic instrument recognition in a polyphonic mixture using sparse representations. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pages 233–236, 2007.

[27] Peter Li, Jiyuan Qian, and Tian Wang. Automatic instrument recognition in polyphonic music using convolutional neural networks. *arXiv preprint arXiv:1511.05520*, 2015.

[28] Vincent Lostanlen and Carmine-Emanuele Cella. Deep convolutional networks on the pitch spiral for music instrument recognition. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, pages 612–618, 2016.

[29] Meinard Müller and Nanzhu Jiang. A scape plot representation for visualizing repetitive structures of music recordings. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, pages 97–102, 2012.

[30] Sergio Oramas, Oriol Nieto, Francesco Barbieri, and Xavier Serra. Multi-label music genre classification from audio, text and images using deep features. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, pages 23–30, 2017.

[31] Elias Pampalk, Simon Dixon, and Gerhard Widmer. Exploring music collections by browsing different views. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 2003.

[32] Elias Pampalk and Masataka Goto. MusicRainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 367–370, 2006.

[33] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, 2004.

[34] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO 2017)*, pages 2744–2748, 2017.

[35] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahava. On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–158, 2011.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Marc Torrens, Patrick Hertzog, and Josep-Lluis Arcos. Visualizing and exploring personal music libraries. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, 2004.

[38] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*, pages 2643–2651, 2013.

[39] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[40] Kazuyoshi Yoshii and Masataka Goto. Music Thumbnailer: Visualizing musical pieces in thumbnail images based on acoustic features. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, pages 211–216, 2008.

[41] Guoshen Yu and Jean-Jacques Slotine. Audio classification from time-frequency texture. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2014)*, pages 1677–1680, 2009.

[42] Xin Zhang and Zbigniew W. Ras. Differentiated harmonic feature analysis on music information retrieval for instrument recognition. In *Proceedings of the IEEE International Conference on Granular Computing*, pages 578–581, 2006.