Fine-tuning ImageNet Pre-trained Models To Be Transferred for Predominant Instrument Recognition in Polyphonic Music *

☆Lifan Zhong, Daisuke Saito, Nobuaki Minematsu (UTokyo)

1 Introduction

Predominant instrument recognition in polyphonic music is a task of determining what instruments are predominant in a music clip where one or multiple instruments may be played simultaneously. It has a potential to contribute to other musical tasks, including source separation, multi-track mixing, music recommendation, etc.

In recent years, with the development of deep learning, the computer vision community has achieved a great success and provided other research fields with various powerful models.

However, researchers were tackling predominant instrument recognition with domain-specific neural network architectures. Though good results were obtained, extra domain knowledge and many experiments were often required to achieve a good structure for musical content. We believe things will become much easier if we use the structure directly from computer vision. In the domain of music and audio, the spectrogram features act as a basic yet powerful representation. If we regard a spectrogram as a gray image, transfer learning from visual domain may be possible.

In this paper, we will demonstrate how the models with ImageNet weights are applied to predominant instrument recognition.

This paper is organized as follows. In Section 2, related work will be introduced. In Section 3, the basic method will be introduced. In Section 4, detailed settings of the experiments and the evaluation results will be discussed. Finally, we will conclude this paper in Section 5.

2 Related Work

2.1 Predominant Instrument Recognition

Fuhrmann et al. tackled predominant instrument recognition in polyphonic music using SVM (Support Vector Machines) in 2009 [1].

Fig. 1 Predominant Instrument Recognition

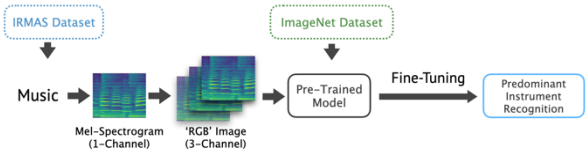

Fig. 2 Scheme of the Proposed Method

In the era of deep learning, Han et al. applied VGG-like convolutional neural networks (CNNs) to the task of predominant instrument recognition in polyphonic music, which outperformed the previous methods [2]. Slizovskaia et al. tried to modify the neural network structures based on domain knowledge, achieving similar scores to previous work with much fewer model parameters [3].

2.2 Instrument Recognition with ImageNet Pre-trained Models

Shukla et al. experimented with four-class instrument classification with ImageNet pre-train CNN models [4]. Shiroma et al. explored six methods to convert a 1-channel Mel-spectrogram to a 3-channel RGB image for the visual neural networks to process in the task of solo instrument classification [5].

3 Method

3.1 Transfer Learning

By transfer learning, the model trained on the source domain is then applied to the target domain. In this work, the source domain is image classification. We first train a model with images and then fine-tune this model with the instrument recognition corpus, as shown in Fig. 2.

---

* 主要楽器認識を目的とした ImageNet モデルの転移学習とその精緻化．鍾立帆，齋藤大輔，峯松信明（東大）

3.2 Channel Conversion

The visual models are trained with 3-channel RGB images, while our training samples are Mel-spectrograms, which can be regarded as 1-channel gray images. So, we have to convert the Mel-spectrograms to 3-channel inputs. In this work, we tried three different methods of channel conversion:

1) *Duplication:* the input Mel-spectrogram is duplicated and stacked to form a 3-channel gray image [6][7].
2) *Zero-padding:* the input Mel-spectrogram is stacked with two channels of zeros to form a 3-channel input [7].
3) *Stereo:* the three channels of the input are the Mel-spectrograms of the mono audio, the left channel audio, and the right channel audio.

4 Experiments

In this section, we demonstrated how our models were trained and evaluated. Generally, we adapted the workflow, the pre-processing settings and method based on 1-sec windows from [2].

4.1 Datasets

Two datasets were used in the experiments: the IRMAS dataset and the ImageNet dataset.

IRMAS (Instrument Recognition in Musical Audio Signals) [8] contains real-world professionally produced music clips that differ in years, genres and audio quality.

All the music excerpts in the IRMAS dataset are in the format of 44.1kHz 16-bit stereo. Label distribution and the annotated instrument labels are shown in Fig. 3. (Abbreviations: acoustic guitar - gac; electric guitar - gle; saxophone - sax; trumpet - tru) Other instruments like bass, drums, synthesizers, and percussion were not annotated.

The IRAMS dataset has an official split of the training set and testing set. The training set contains 6,705 audio files, of which each clip has a duration of 3 seconds and was labeled with one predominant instrument. The testing set contains 2,874 audio files, of which each clip has a variant duration from 5 seconds to 20 seconds and was labeled with one or multiple predominant instruments.

The ImageNet dataset [9] contains more than 1.2 million real-world images and 1,000 classes in total. It often serves as a benchmark for image classification and a powerful dataset for transfer learning.

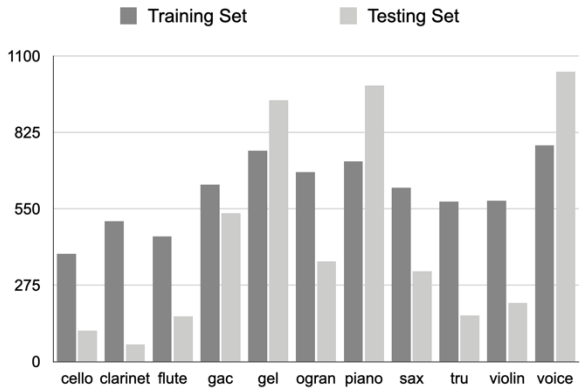

Fig. 3 Label Distribution of the IRMAS dataset

4.2 Pre-processing

Pre-processing here means extracting the log Mel-spectrogram from the audio. Pre-processing is often regarded as a crucial step to achieving a better performance of a predominant instrument recognition system.

First, all the audio files were loaded and normalized by Torchaudio [10]. In the case of duplication and zero-padding, the input audio files were then converted into mono by calculating the mean of the two channels. While in stereo channel conversion, apart from the mono waveform, the left and the right channel were separated. After that, all the audio files were resampled to 22,050 Hz.

We then extracted Mel-spectrogram from the re-sampled audio files, with 128 Mel bins, a window size of 2048, and a hop length of 512. Finally, the magnitude of the Mel-spectrogram is compressed with a natural logarithm.

We divided the obtained log Mel-spectrogram into a duration of 1 second, giving input of size 1×43×128 for the neural networks.

4.3 Model Training

The standard ResNet18 and ResNet50 were used [11], and the ImageNet pre-trained weights were loaded from Torchvision. For the final activation function, we used sigmoid instead softmax, because in the evaluation, the determination of

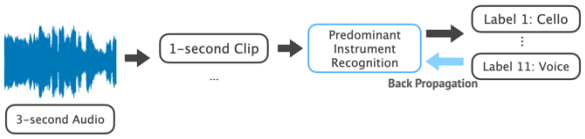

Fig. 4 Model Training

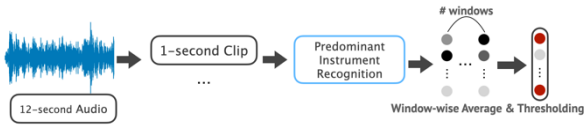

Fig. 5 Model Evaluation

the predominant instruments could be a multi-label prediction.

During training, the batch size was set to 64, and the initial learning rate was set to 0.0005. We used a cosine annealing scheduler and a linear warm-up strategy for the first 5 epochs to adjust the learning rate. In order to make the training process more stable, we applied a weight decay of 0.0005. We set the maximum number of epochs to 300 and applied an early stop strategy with patience of 25 epochs.

Fifteen percent of the training set of IRMAS was randomly selected to form a validation set. We monitored the performance of our models on the validation set in training. After training, the model with the best validation loss would be saved and used for testing.

4.4 Model Evaluation

To handle the various-length audio in the testing data, we adapted an aggregation strategy, which is the 's1' proposed in [2]. As indicated in the scheme in Figure 5, we first calculated the sigmoid scores for each of the 1-sec windows with 50% overlap. Then we used the average window-wise sigmoid scores as the aggregated output. Thresholds were applied to the aggregated output to get final predictions.

Note that in [2], it was reported that the other strategy, 's2', which is to add up all the window-wise scores and normalize the final 11-d output by dividing it with the maximum value, had a better performance. However, during our reproduction and the following experiments with pre-trained models, we found that the difference between the two strategies was slight, and most of the time, 's1' had better performance. Hence, we used 's1' for our experiments. We recommend readers refer to the original paper for a detailed explanation of the two aggregation strategies.

We performed each of our experiments three times with different random seeds and reported the average scores of the three experiments.

4.5 Evaluation Metrics

We used precision, recall, and F1-score as the evaluation metrics of our system, following the previous work. Their definitions are as follows:

$$P_{micro} = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FP_l} \quad (1)$$

$$R_{micro} = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FN_l} \quad (2)$$

$$F1_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \quad (3)$$

Where the capital L denotes the number of the instrument classes, and the lower l denotes the instrument index.

Since the labels in the IRMAS dataset are not equally distributed, we report the micro-weighted score of the above metrics. By doing so, the large classes will influence the final results more than the small ones. Also, the F1 score would be emphasized because we obtained the final results through thresholds, in which a higher threshold results in a higher precision score and a lower recall and vice versa.

4.6 Results

The results of channel conversion are shown in Table 1. The experiments on channel conversion were done using the ResNet50 model with ImageNet pre-trained weights.

As can be seen in Table 1, among the three methods, duplication, the most straightforward one, achieved the best F1 score of 0.649. Besides, the additional information introduced by the stereo channel separation did not bring any improvement to our system as we expected.

To study the effects of ImageNet pre-trained weights on our task, we compared the results of the baseline model, ImageNet pre-trained ResNet models, and randomly initialized ResNet models.

Table 1 Evaluation Results of Different Channel Conversion Methods

Method	Precision	Recall	F1-micro
Duplication	0.704	0.601	**0.649**
Zero-padding	0.697	0.588	0.638
Stereo	0.717	0.559	0.628

Table 2 Evaluation Results of Different Models (P: Pre-trained with ImageNet)

Model	P	Precision	Recall	F1
Baseline [2]		0.655	0.557	0.602
ResNet18	N	0.722	0.553	0.626
ResNet18	Y	0.732	0.574	0.643
ResNet50	N	0.734	0.547	0.627
ResNet50	Y	0.704	0.601	**0.649**

In the experiments, channel conversion was done by duplication. The results are shown in Table 2. In this table, the baseline is the best model reported in [2].

As can be seen in Table 2, the best F1-micro score achieved is 0.649 from the ResNet50 with ImageNet pre-trained weights. Also, whether pre-trained with ImageNet or not, the ResNet models outperformed the baseline model. This indicates that the residual visual model itself can better model this task than the baseline does.

From this table, it can also be seen that the ImageNet pre-trained models outperformed the ones with random weights. This indicates that our task of predominant instrument recognition can benefit from pre-training with large image corpus, even if the training samples in the corpus have no apparent relationship with the Mel-spectrograms.

5 Conclusions

In this paper, we demonstrated how we could improve the predominant instrument recognition system simply by ImageNet transfer learning without further modification of the structures of the standard visual models. In our experiments, three ways of channel conversion and two visual models were tested, of which ResNet50 with duplication channel conversion achieved the best evaluation results.

In the future work, we would like to explore better representations of the music domain data for the image domain transfer learning. Also, even though the experiments have shown that such a cross-domain transfer learning can boost the performance of the predominant instrument recognition system, there is no convincible explanation of the underlying mechanism yet. Hence, a proper explanation of the connections between visual domain data and music domain data is also needed.

References

[1] F. Fuhrmann *et al*., "Scalability, Generality and Temporal Aspects in Automatic Recognition of Predominant Musical Instruments in Polyphonic Music." In *ISMIR*, pp. 321-326. 2009.

[2] Y. Han, *et al*., "Deep convolutional neural networks for predominant instrument recognition in polyphonic music." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25, no. 1 (2016): 208-221.

[3] J. Pons *et al*., "Timbre analysis of music audio signals with convolutional neural networks." In *EUSIPCO*, pp. 2744-2748. IEEE, 2017.

[4] U. Shukla *et al*., "Instrument classification using image based transfer learning." In *ICCCS*, pp. 1-5. IEEE, 2020.

[5] Y. Shiroma *et al*., "Investigation on fine-tuning with image classification networks for deep neural network-based musical instrument classification." *IEICE Tech. Rep.* 121, no. 66 (2021): 75-79.

[6] K. Palanisamy *et al*., "Rethinking CNN models for audio classification." *arXiv preprint arXiv:2007.11154* (2020).

[7] A. Guzhov *et al*., "Esresnet: Environmental sound classification based on visual domain models." In *ICPR*, pp. 4933-4940. IEEE, 2021.

[8] J. J. Bosch *et al*., "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals." In *ISMIR*, pp. 559-564. 2012.

[9] J. Deng *et al*., "Imagenet: A large-scale hierarchical image database." In *CVPR*, pp. 248-255. 2009.

[10] Y. Y. Yang *et al*., "Torchaudio: Building blocks for audio and speech processing." In *ICASSP*, pp. 6982-6986. IEEE, 2022.

[11] K. He *et al*., "Deep residual learning for image recognition." In *CVPR*, pp. 770-778. 2016.