International Journal Of Multidisciplinary Research In Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com |

| Volume 2, Issue 9, September 2019 |

Musical Instrument Identification Using Machine Learning

Chinmay Relkar[1], Vishal Tejwani[1]

Department of IT, PVG's COET, Pune, India

ABSTRACT : Music is a combination of vocal and/or instrumental sounds which aims to express emotions and produce harmony. Identification of the instruments in polyphonic music is highly sought-after and can be beneficial in many fields such as music search based on instruments, genre classification and automatic music transcription. We used Mel Frequency Spectrograms along a Convolutional Neural Network to identify instruments present in the IRMAS dataset, which consists of over 10,000 audio clips of 11 instruments. We used an ensemble of 3 learners and aggregated time-distributed outputs by performing global class-wise max-pooling. We also used LRM source segregation as a preprocessing step to divide the original signal into 3 streams. Using this approach, we achieved F1 scores of 0.631 (micro) and 0.539 (macro), which are both better than the previous state of the art.

KEYWORDS : Convolutional Neural Network, Music Instrument Recognition, Mel Spectrogram

I. INTRODUCTION

Search engines work with text-based queries, which are ideal while searching for results stored in plaintext format, such as blogs and websites. However, music files are stored in binary format and do not contain any information about the type of music, such as the genre of the music or the instruments used in it. Automatically identifying the instruments will simplify audio tagging, which will pave the way for simple music search by instrument. Thus, we can say that identification of the instruments in polyphonic music is highly sought-after and can be beneficial in many fields such as searching for music based on the instruments in it, genre classification and automatic music transcription.

II. LITERATURE SURVEY

A variety of different approaches have been used for identifying musical instruments. Most initial works focused on studio-recorded isolated notes or solo phrases [1], [2]. More recent works have focused on polyphonic music synthesised from isolated notes [3], [4]. Deep Learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for decades [5]. Applications using DNNs have shown improved performance in a variety of applications, spread out over multiple domains, including computer vision [6], speech processing [7] and in medicine [8].

Park et al. [9] used a Convolutional Neural Network (CNN) along with feature fusion to identify musical instruments in monophonic audio. Han et al. [10] proposed a framework using Deep CNNs for predominant instrument recognition in real world polyphonic music.. They used 11 instruments from the IRMAS dataset [11]. A 4 layer ConvNet was used which was inspired by AlexNet [12] and VGGNet [13]. They reported that advances in deep learning in the image processing domain were transferable to the audio processing domain. They handled variable size music excerpts by performing local class-wise max-pooling and aggregating multiple outputs from sliding windows over test audio. They also found out that a window size of 1.0s and threshold of 0.55s gave the highest accuracy. Their proposed framework achieved 23.1% performance improvement compared to the previous state-of-the-art algorithm. However, their accuracy for some instruments was very low and they were only able to identify the predominant instrument, not all the instruments.

Performing musical information retrieval after audio segregation is beneficial. Bosch et al. [14], in their paper, presented approaches combining source separation and instrument recognition, where they learned that there is 32% improvement of the micro F1-measure over the original algorithm. Training the classification models with the split streams of separated audio and tuning the parameters of each of these models leads to better performance but their approach has a drawback of computational complexity. Alternatively, they proposed the decomposition of the stereophonic poly-timbral audio into the left, right, mid and side streams, and the combination of the labels identified by the instrument recognition algorithms in each of the streams, which led to 19.2% increase in the performance of the predominant instrument recognition.

III. PROPOSED WORK

The process of identifying the instruments present in it is split into two main stages: preprocessing and neural network. The process starts with preprocessing the sound wave(music), and is followed by passing the processed sound wave to a neural network.

In the preprocessing stage, the sound wave, if in stereo audio form is segregated into 3 different mono audio streams using LRM source segregation technique. Each mono audio stream of sound is then converted into a mel frequency spectrogram, which is represented in a form similar to an image. Since Convolutional Neural Networks (CNNs or ConvNets) are very good at extracting features from images, the spectrograms are fed to the ConvNet to learn patterns of how different musical instruments "look". This procedure is converted into 2 sections namely Preprocessing (Converting sound to spectrograms) and Neural Networks (learning instrument specific patterns).

We are using the IRMAS (Instrument Recognition in Musical Audio Signals) dataset for the task of training a classifier that recognises instruments in professionally produced polyphonic music. The dataset is made up approximately 10000 labelled files from 11 instruments. The dataset is split into a training and testing set. The training dataset is comprised of 6705 single labelled audio files which have a fixed length of 3s, whereas the testing dataset consists of 2874 multi-labelled files of variable length (between 5s and 20s). All the audio files are in 16 bit stereo wav format and are sampled at 44,100 Hz. The number of positive labels in the training set is equal to the number of samples in it. However, since the samples in the testing set may have multiple labels, the total number of positive labels in it is more than the number of files in it.

A. SOURCE SEGREGATION

Music can be categorised into two types, mono (having single audio stream) and stereo (having two audio stream). If the input is a stereo audio, its streams might contain some distinct features. To extract these features LRM source segregation technique is used. In this technique, audio is segregated into three streams with L = Left, R = Right, M = L+R (Mid)

Each stream is converted into mel-spectrogram representation and then passed to the neural network for instrument identification. A more commonly known source segregation technique is LRMS (Left-Right Mid-Side) source segregation, which splits stereo audio into 4 channels. Considering that the segregated stream Side (S) is noisy, and adversely affects the performance of the system, we removed that stream from the implementation. Figure 1 shows how each stream is processed independently.

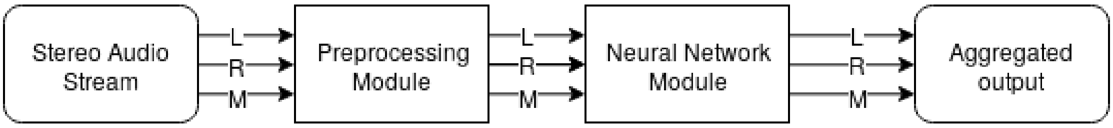


Figure 1: LRM Source Segregation Implementation

B. PREPROCESSING

While neural networks can take raw data as input, preprocessing the audio improves the performance. The raw audio representation is converted into an extracted feature representation, which is provided as input to the neural network.

To calculate the me spectrogram, the mono audio is normalised by dividing the amplitude by the maximum amplitude and the normalised audio is split into non-overlapping windows (frames) of 1 second each. Then, the Short-time Fourier Transform (STFT) is applied to the windowed audio to convert the audio into a spectrogram. The spectrogram obtained is mapped from the linear (hertz) scale to the mel scale and is compressed using a natural logarithm.

C. NEURAL NETWORK

A Convolutional Neural Network is essentially a combination of a feature extractor and a classifier. Our ConvNet has an architecture which is inspired from AlexNet and VGGNet. It has 4 sets of Convolutional and Max Pooling layers. The ConvNet is designed according to our input data. 1 x 1 zero padding has been applied to the input for each convolution layer. This is done to preserve the spatial resolution regardless of the input window size. Filters with a 3 x 3 receptive field and a fixed stride size of 1 have been used. Spatial abstraction has been done by max-pooling with a size of 3 x 3 and stride size of 1. The number of filters for the layers have been increased by a factor of 2 after every two convolution layers, starting from 32 up to 256. The last max-pooling layer prior to fully connected layer is a global max-pooling layer which takes a maximum value from all neurons in each channel.

The Adam optimizer has been used with the default parameters: $\square = 0.001, \square_1 = 0.9, \square_2 = 0.999, \square = 10^{-8}$. We used 15% of the training set as validation data and trained the model till the validation loss did not get better for 3 epochs. To regularize training, a dropout rate after each max-pooling layer has been set to 0.25 and after a fully connected layer to 0.5. Rectified linear unit (ReLU) has been used as the activation function to train all the convolution layers except the layer at the end of the network which uses sigmoid function. The sigmoid activation is used instead of softmax in the final fully connected layer since an arbitrary number of instruments are to be detected.

Input size (number of filters × time × freq)	**Layer Description**
1 × 44 × 128	Mel-spectrogram
32 × 46 × 130	3 × 3 Convolution, 32 filters
32 × 48 × 132	3 × 3 Convolution, 32 filters
32 × 16 × 44	3 × 3 Max-pooling
32 × 16 × 44	Dropout (0.25)
64 × 18 × 46	3 × 3 Convolution, 64 filters
64 × 20 × 48	3 × 3 Convolution, 64 filters
64 × 6 × 16	3 × 3 Max-pooling
64 × 6 × 16	Dropout (0.25)
128 × 8 × 18	3 × 3 Convolution, 128 filters
128 × 10 × 20	3 × 3 Convolution, 128 filters
128 × 3 × 6	3 × 3 Max-pooling
128 × 3 × 6	Dropout (0.25)
256 × 5 × 8	3 × 3 Convolution, 256 filters
256 × 7 × 10	3 × 3 Convolution, 256 filters
256 × 1 × 1	Global Max-pooling
1024	Flattened and fully connected
1024	Dropout (0.5)
11	Sigmoid

Table 1: CNN Architecture

IV. SOFTWARE IMPLEMENTATION

Every step mentioned above, namely Source Segregation, Preprocessing and Neural Network, is built as an independent module. Each has a definite input and provides a defined output and can also work in parallel.

The input to the Source Segregation module is the stereo audio stream. The module provides 3 different mono audio streams as output. These mono streams are passed to the preprocessing module as input, which then converts the stream to list of mel-spectrograms of 1 second time frame each.

The Neural Network module takes a single mel-spectrogram as input, predicts the list of instruments (out of 11 instruments) present in that 1 sec of the audio stream as a multi-hot encoded array of 11 in size. The module is an ensemble of 3 neural networks, built in a such a way that it aggregates the output for all the 1 seconds audio frames belonging to a single audio stream.

For every single mono audio stream we get a multi-hot encoded array of predicted output from the neural network module. So, for a stereo input we get 3 multi-hot encoded arrays, representing 3 streams output of the source segregation module. These 3 arrays are then aggregated to compute the final result.

V. RESULTS AND EVALUATION

The testing audios are of variable lengths between 5s and 20s. We used a fixed length window of 1.0s and hop length of 0.5s to obtain overlapping samples of the audio file. For preprocessing of testing data, we normalized the time domain signal and padded it to make it exactly a multiple of 0.5s. We then transformed the framed samples into mel-spectrograms and passed them all through the neural network. Performing class-wise averaging on the sigmoid outputs suppresses the activation of sporadically occurring instruments. We performed global class-wise max-pooling in order to reduce this effect. We used an identification threshold of 0.55.

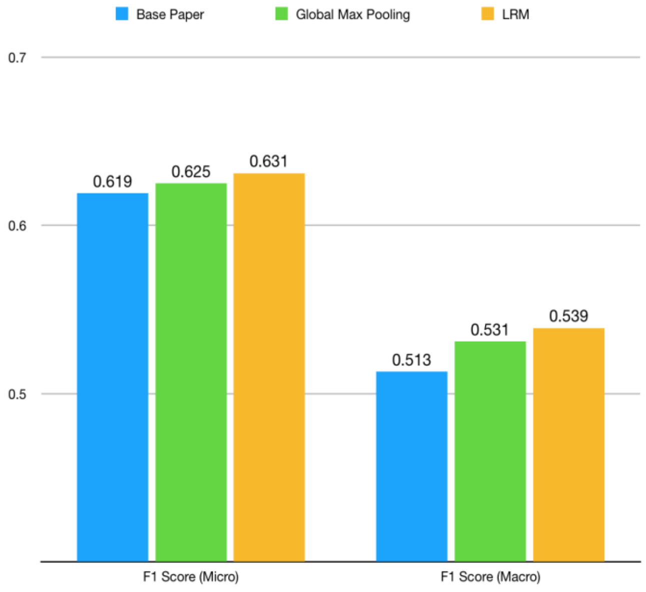
A. EXPERIMENTS CONDUCTED

We conducted experiments with single model(1M) and an ensemble of 3 models(3M). The aggregation strategy used was local class-wise max pooling followed by averaging(LMP) or global max pooling(GMP). The experiments were conducted without source segregation(n) and with source segregation using Left-Right-Mid-Side(LRMS) and Left-Right-Mid(LRM). The aggregations used for separated streams were Majority vote(MV) and Max pooling(MP).

Experiment Title	**Micro F1 Score**	**Macro F1 Score**
1M-LMP-n (Last layer softmax)	0.575	0.480
1M-GMP-n	0.622	0.512
3M-GMP-n	0.625	0.531
3M-GMP-LRMS-MV	0.621	0.532
3M-GMP-LRMS-MP	0.620	0.526
3M-GMP-LRM-MV	0.627	0.533
3M-GMP-LRM-MP	**0.631**	**0.539**

Table 2: Table of Experiments

By applying LRM source segregation as a preprocessing step to the stereo audio before converting it into mel-spectrograms. Using this approach, we achieved F1 scores of 0.631 (micro) and 0.539 (macro), which is new state of the art. Figure 2 shows the comparison between our approach and state of the art algorithm. Figure 3 shows the shows the Micro and Macro average precision, recall and F1 scores obtained.



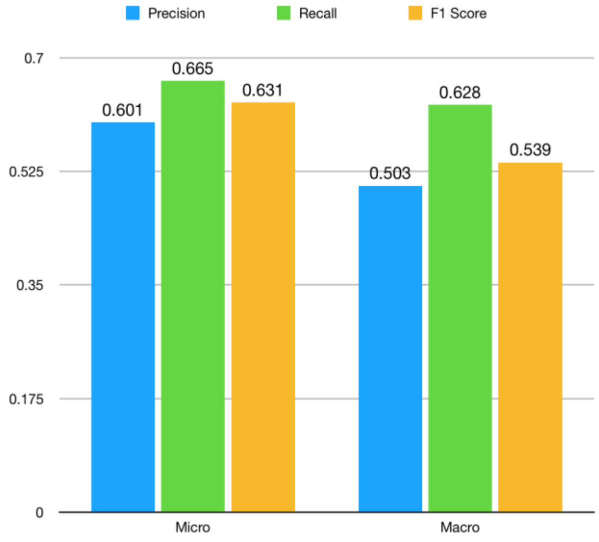


Figure 2 : Comparison between our approach and state of the art algorithm

Figure 3 : Micro and Macro Measures

VI. CONCLUSION

We have implemented a four layer ConvNet that achieves state of the art results (as of 2016) on the task of instrument recognition in polyphonic music. Changing the local max pool of output to a global max pool helped in achieving micro and macro average F1 measures of 0.625 and 0.531 respectively, which are better than the previous state of the art. Using LRM source segregation as a preprocessing step further increased the F1 scores to 0.631 (micro) and 0.539 (micro), which are better than the previous state of the start by 0.012 (micro) and 0.026 (micro).

REFERENCES

[1]A. Eronen and A. Klapuri, “Musical instrument recognition using cepstral coefficients and temporal features,” in Proc. 2000 IEEE Int. Conf. Acoust., Speech Signal Process., 2000, vol. 2, pp. II753-II756.
[2]A. Krishna and T. V. Sreenivas, “Music instrument recognition: From isolated notes to solo phrases,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2004, vol. 4, pp. iv-265-iv-268.
[3]T. Heittola, A. Klapuri, and T. Virtanen, “Musical instrument recognition in polyphonic audio using source-filter model for sound separation,” in Proc. Int. Soc. Music Inf. Retrieval Conf., 2009, pp. 327-332.
[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, pp. 155-155, 2007.
[5]Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436-444, 2015.
[6]A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks”, in Proc. Advances in Neural Information Processing Systems 25, 2012, pp. 1090-1098.
[7]L. Deng and D. Yu, “Deep learning: Methods and applications,” Found. Trends Signal Process., vol. 7, no. 3-4, pp. 197-387, 2014.
[8]V. Gulshan, L. Peng, M. Coram, M. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. Nelson, J. Mega and D. Webster, "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs", JAMA, vol. 316, no. 22, p. 2402, 2016.
[9]T. Park and T. Lee, “Musical instrument sound classification with deep convolutional neural network using feature fusion approach,” arXiv:1512.07370, 2015.
[10]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, 2016.
[11]Juan J. Bosch, Ferdinand Fuhrmann, & Perfecto Herrera. (2014). IRMAS: a dataset for instrument recognition in musical audio signals (Version 1.0).
[12]Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[13]Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
[14]J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, “A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals,” in Proc. Int. Soc. Music Inf. Retrieval Conf., 2012, pp. 559-564.