Deep Single Shot Musical Instrument Identification using Scalograms

Debdutta Chatterjee*, Arindam Dutta†, Dibakar Sil‡, Aniruddha Chandra*
*Department of Electronics and communication Engineering, National Institute Of Technology, Durgapur 713209, India
†Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India
‡Steradian Semiconductors Pvt. Ltd., Bangalore, India.

2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) | 978-1-6654-5645-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAIIC57133.2023.10066992

***Abstract*—Musical instrument identification has for long had a reputation of being one of the most ill-posed problems in the field of musical information retrieval. Despite several robust attempts made at solving the problem, a timeline spanning over the last five odd decades, the problem remains an open conundrum. In this work, we take on a further complex version of the traditional problem, we attempt to solve the problem with minimal data available - one audio excerpt per class. We propose to use a convolutional Siamese network and a residual variant of the same to identify musical instruments based on the corresponding scalograms of their audio excerpts. Results obtained for two publicly available datasets validate our algorithm, achieving over 80% accuracy with only 5 sets of training data. Moreover, our proposed architectures work for both spectrograms as well as scalograms, and exhibit improvements, albeit marginal ($\approx$ 3%), for the later input class.**

***Index Terms*—Audio excerpts, scalogram, one-shot learning, convolutional Siamese network.**

I. Introduction

MUSICAL signal processing applications like automatic music transcription, beat tracking and extraction of melody from music [1] are increasing at a brisk pace. Mixing of musical notes, instrument wise equalization, archiving and cataloging also require proper identification of instruments. The task can be quite challenging, for example, music generated in an orchestra is often a superposition of concurrent notes, echos, and other background noise. The problem of musical instrument classification is thus ill-posed and demands application of sophisticated algorithms.

Deep learning (DL) algorithms based on convolutional neural networks (CNNs) are progressively being used for musical instrument classification [2]. Recently, scalograms, instead of traditional spectrograms, have been proposed for such frameworks [3] to exploit the rich time-frequency localized features. However, none of the existing works are directed towards reducing the training data. This fact motivated the authors to explore a CNN based Siamese network as it has the potential to outperform several complex algorithms in single-shot classification [4]. Nevertheless, scalogram based one-shot classification task is unfrivolous as the scalogram of the same instrument at two different notes (e.g. clarinet notes G5 and F3) has a significant amount of difference while different instruments of similar class (e.g. violin and viola) have a striking similarity.

The problem of one-shot musical instrument classification is fairly immature and has received limited attention by far. In this paper, we present a deep convolution Siamese neural network, and the residual network variant of the same, to address this problem. Specifically, the major contributions of this work are:

- A robust Siamese network architecture has been proposed to classify the musical instrument in one-shot. Feasibility of the network ensures that it is possible to obtain appreciable accuracy despite the dearth of data.
- To reduce the number of parameters and thus the memory foot-print of these architectures, we also propose a residual version of the proposed Siamese architecture.
- We have tested the detection accuracy of both these architectures on scalograms obtained for two openly available datasets. Further, a comparative study of one-shot classification with scalograms and spectograms is performed to study how time-frequency representation affects accuracy of the identification task.

The rest of the paper is organized as follows. Section II gives a brief overview of the datasets used in our study and Section III introduces the architectures. Next, Section IV presents the experiments performed and their corresponding results. Finally, Section V concludes the paper.

II. Datasets

The music instrument datasets from Kaggle [5] and ISMIR [6] are used as they satisfy the basic conditions such as transparency, openness and proper annotations. Both the datasets contain enough musical instruments for testing and training our model.

We have taken the Morse analytic wavelet transform for generating scalograms. The short-time Fourier transform (STFT) suits better for non-stationary signals, while continuous wavelet transform (CWT) gives high time-frequency resolution and is better suited for analyzing signals that contain non-periodic and fast transients features.

A. Dataset from Kaggle

The dataset [5] has been procured by recording 14 different musical instruments for 1 s at a sampling rate of 44.1 kHz. Table I gives the figures for class and categories of all the instruments recorded.

Although the dataset contains 2500 audio signals, only 1540 audio excerpts were used as equal amount of data was required for each instrument. Scalograms of size $[224, 224, 3]$ produced

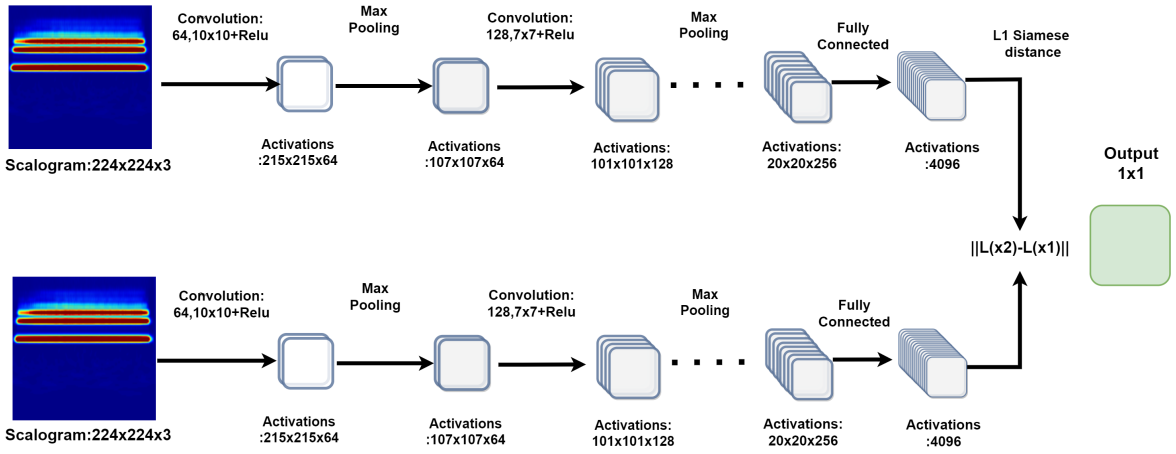

Fig. 1. Schematic of the Siamese neural network architecture for musical instrument identification using scalograms.

TABLE I
INSTRUMENT SAMPLES IN KAGGLE DATASET

String	Brass	Woodwind
Bass[Double] (153)	French Horns (166)	Clarinet (481)
Cello (227)	Saxophone [Alto] (480)	Flute (454)
Guitar (420)	Saxophone [Soprano] (284)	Oboe (360)
Viola (220)	Tuba (560)	Trombone (312)
Violin (560)		Trumpet (246)

by CWT are fed to the proposed networks for training and validation of one shot learning.

1) Dataset from ISMIR: The second dataset used for our work is the open-source ISMIR dataset [6] which consists of 3 s long clips of various musical instruments sampled at 44.1 kHz. As shown in Table II, the dataset consists 6715 number of audio data of 10 musical instruments. In order to maintain consistency, 388 samples of each category of instruments (total of 3880) were considered for training.

TABLE II
INSTRUMENT SAMPLES IN ISMIR DATASET

String	Brass	Woodwind
Piano (721)	Saxophone (626)	Clarinet (505)
Cello (388)	Trumpet (577)	Flute (451)
Electric Guitar (760)		Organ (682)
Violin (580)		
Acoustic Guitar (637)		

A major difference of the second dataset from the first one is, apart from the signal of the primary instrument each audio clip also contains notes from other instruments or human voice in the background.

Similar to the previous dataset, scalograms are obtained for each audio data of size $[656, 875, 3]$. We further resize these images to size $[224, 224, 3]$ to obtain the final dataset that was fed to the model.

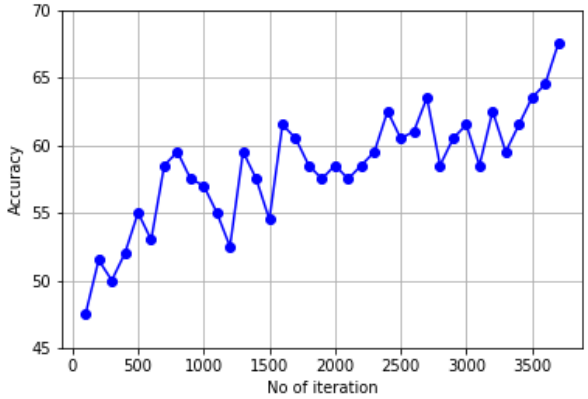

Fig. 2. Accuracy of training set with respect to epochs.

III. METHODOLOGY

A. Convolutional Siamese Network Architecture

Fig. 1 demonstrates the basic VGG type convolutional network architecture in detail. We base our architecture on deep Siamese (twin) networks which are congruent networks tied by the same weights. When two networks are sharing weights, it is expected that when the same image is passed through both the networks, the corresponding feature maps and hence the single dimensional feature vectors obtained at the penultimate layer will be similar. We have used the difference between the feature vectors as a weighted L1 distance in the last layer, so that the last layer is sparse enough for easy processing. Further, we use the sigmoid function to squash the values of the elements of the last layer to $[0, 1]$ and use it as a probabilistic measure. The binary cross-entropy objective has been used for training the model and the loss-epoch training plot has been shown in Fig. 2.

The network takes input in the form of a pair of scalograms (or spectrograms), each of which have dimensions $[224 \times 224 \times$

TABLE III
ACCURACY WITH KAGGLE DATASET

Training data sets	Testing data sets	convolutional Siamese				residual Siamese			
		Scalogram		Spectogram		Scalogram		Spectogram	
		max	mean	max	mean	max	mean	max	mean
2	12	82%	65%	74%	61%	74%	60%	71%	52%
5	9	82%	69%	76%	63%	84%	70%	79%	64%
8	6	86%	74%	78%	66%	90%	73%	83%	68%
10	4	86%	75%	80%	69%	94%	81%	89%	78%
12	2	92%	78%	82%	73%	96%	94%	89%	84%

3]. The number of parameters of the network stands out at ≈ 420 million $(420,646,209$, to be exact). Thus, the network can overfit to a large extent. This necessitates use of pairwise training and dropout [7] to make the network robust. The use of max-pooling layers after convolutional layers ensure dimensionality reduction. This helps the network to focus on a smaller subspace of the input data, and thus aiding in increased classification accuracy and lower memory requirements. The use of ReLU non-linearity [8] ensures that the activations do not die out in deeper layers of the network. A popular optimizer, Adam [9], has been used with a constant learning rate of 6×10^{-4} and with other default hyper-parameter values to obtain an easy convergence.

B. One Shot Learning

Our problem can be mathematically expressed as follows. Given the set of audio excerpts $\mathbf{X}$, such that $\mathbf{x_1}$, $\mathbf{x_2}$, ... , $\mathbf{x_k}$ with corresponding elements x_{11}, x_{12}, ... , x_{1a}; x_{21}, x_{22}, ... , x_{2b}; x_{k1}, x_{k2}, ... , x_{kn}, are disjoint subsets of $\mathbf{X}$, with respective correspondence to the labels y_1, y_2, ... , y_k, we wish to find a many-to-one mapping $\mathcal{F}(.)$ such that,

$$y_i \mathbf{1}(i = j) = \mathcal{F}(x_{ip}, x_{jq}; \Theta), \tag{1}$$

where, $i, j \leq k$, y_i is the true label corresponding to some x_{ip} $\in \mathbf{x_i} \in \mathbf{X}$, Θ represents the parameters of the network $\mathcal{F}(.)$, and $\mathbf{1}(.)$ is the indicator function.

The prediction vector (P) is given by,

$$P = \sigma \left(\sum_j \gamma_j \left| M^j_{1,L-1} - M^j_{2,L-1} \right| \right), \tag{2}$$

where, $\sigma(.)$ is the sigmoid function, γ_j are additional weights (parameters) of the network which are duly learned during training, and $M_1(.)$ and $M_2(.)$ are the two component networks of the Siamese network.

The procedure of learning adequate features from small datasets is daunting and, at the same time, computationally expensive. One shot learning is one such problem in which predictions are made based on a single example. However, once the network has been optimally trained, we are all set to test and demonstrate the discriminative potential of the network not just on new data but to data from unknown distributions. Given a query scalogram x_q and corresponding scalograms $X^{k=C}_{k=1}$ belonging to one of the $\mathbf{C}$ classes, we predict the class C^* in accordance to,

$$C^* = \arg\max_{X^k} \mathbf{P}_k, \tag{3}$$

where, $\mathbf{P}_k$ is the prediction vector from (2).

C. Residual Siamese Network Architecture

Mathematically, for some part of a traditional deep CNN, say x is the input and $\mathcal{H}(x) = \mathcal{F}(x)$ is the output. Then, incorporating skip connection, the input-output relation can be expressed as,

$$\mathcal{H}(x) = \mathcal{F}(x) + x. \tag{4}$$

Making networks go deeper, often makes them overfit on the training set, resulting in poor performance on the test set. Residual networks, powered by skip connections as shown in (4), is able to realize the idea of deeper networks with a lesser number of parameters. These networks also mitigate the problem of vanishing gradients, a problem extremely common to deep CNNs.

Taking an inspiration from [10], we modify our initially presented architecture to contain residual connections. We find that the number of parameters drop by 18 times to $\approx$ 23 million $(23,553,025$, to be exact), without any sacrifice in accuracy. Like the basic convolutional Siamese network, we use Adam optimizer but with a learning rate of 5×10^{-4} while all other hyper-parameters are set to their default values.

IV. EXPERIMENTS

The codes for the project were executed on two separate systems for two separate tasks. Codes pertaining to the network training and testing were written in the TensorFlow 2.0 environment on the Google Colaboratory and were executed on a 12 GB Tesla K80 GPU. The codes have been made available on a open-source repository [11]. Codes pertaining to CWT of the audio excerpts for obtaining scalograms were written and executed in MATLAB 2019b on a system with 64GB RAM and Intel-i7 core processor.

A. Dataset from Kaggle

To access and hence justify the potential of our algorithm, we train the network on randomly chosen training sets which are essentially subsets of the dataset. We randomly choose 2, 5, 8, 10, and 12 sets of training examples and test it the network on the rest of the unknown audio examples.

Table III shows the accuracy obtained on the Kaggle dataset upon using the two proposed networks, namely the VGG type convolutional Siamese network and residual-convolutional Siamese network (2 way one shot accuracy achieved after 4000 epochs on). Even with just 2 training sets, we have a mean

TABLE IV
ACCURACY WITH ISMIR DATASET

Training data sets	Testing data sets	convolutional Siamese				residual Siamese			
		Scalogram		Spectogram		Scalogram		Spectogram	
		max	mean	max	mean	max	mean	max	mean
3	8	78%	55%	73%	53%	74%	60%	71%	52%
5	6	82%	60%	76%	59%	78%	61%	73%	54%
7	4	74%	63%	72%	61%	80%	61%	72%	61%
9	2	78%	61%	76%	64%	82%	64%	78%	63%

accuracy of around 65 %, which increases to more than 90 %, when trained on 12 training sets.

B. Dataset from ISMIR

The ISMIR dataset is one of the few standard datasets in this field for musical instrument classification and hence has been studied thoroughly. Table IV shows the performance of the networks under varied conditions of training and testing. While best accuracy hovers around 80%, mean accuracy, more often than comes out to be around 65%. Although our model do not predict instrument classes more accurately than convention DL models [12], these numbers are justifiably appreciable.

C. Discussion

From both datasets it was found that our proposed networks can quite efficiently categorize a musical instrument from its audio excerpt, even from a noisy version. It is important to note that traditional DL based classification algorithms [13] would achieve such prediction levels only when the network has access to scalograms pertaining to all instrument classes. Also, since the datasets were not manually screened for ill-audio excerpts, which if done would have resulted in higher accuracy, the difference in best accuracy and mean accuracy is obvious. Fig. 3 sums up a comparative study against synonymous baselines.

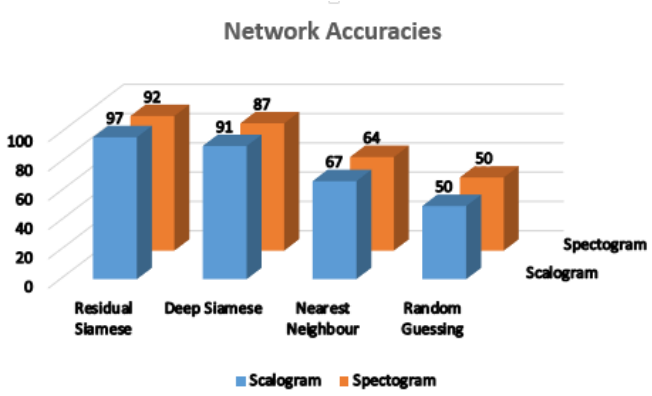

Fig. 3. Comparing best one-shot accuracy from each type of network against baselines.

V. CONCLUSION

In this work, we propose a novel single-shot musical instrument recognition algorithm. Given the fact that properly annotated data for musical instruments is not cornucopious, our proposed algorithm with its abundant pragmatism fits right into the gap. We base our algorithm on convolutional Siamese networks which in-effect study the similarities of two given scalograms rather than memorizing the feature spaces corresponding to scalograms of one particular musical instrument. Our experiments show that we can achieve state-of-the-art results even with just one audio excerpt example per class. However, our network suffers a major drawback in terms of network parameters, which although are low in terms of memory utilization on GPUs but are not suited for portable device applications. In our future work, we will try to develop an online-learning method based on light-weight convolutional or recurrent neural architectures which would make this algorithm a perfect match with low-power portable devices.

REFERENCES

[1] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F. R. Stoter, "Musical source separation: An introduction," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 31–40, Jan. 2019.

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

[3] Z. Ren, K. Qian, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, "Deep scalogram representations for acoustic scene classification," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 3, pp. 662–669, May 2018.

[4] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, Jul. 2015, pp. 1–8.

[5] D. Sil, "Music Instruments Dataset," Nov. 2018, Kaggle. [Online]. Available: https://www.kaggle.com/dibakarsil/music-instruments-and-2d-figures

[6] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack, "ISMIR 2004 Audio Description Contest," Apr. 2006, Tech. Rep., Music Technology Group, Univ. Pompeu Fabra. [Online]. Available: http://mtg.upf.edu/node/461

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[8] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, Jun. 2010, pp. 1–8.

[9] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[11] D. Sil, "Deep single shot musical instrument identification using time-frequency localized features," Aug. 2020, GitHub. [Online]. Available: https://github.com/Dibakar1/Deep-Single-Shot-Musical-Instrument-IdentificationUsing-Time-Frequency-Localized-Features

[12] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *Int. J. Inf. Tecnol.*, pp. 1–10, Jan. 2019.

[13] A. Dutta, D. Sil, A. Chandra, and S. Palit, "CNN based musical instrument identification using time-frequency localized features," *Internet Technol. Lett.*, vol. 2020, no. e191, pp. 1–6, May 2020.