Master Thesis in Sound and Music Computing

Universitat Pompeu Fabra

Musical Instrument Recognition in Multi-Instrument Audio Contexts

Venkatesh Shenoy Kadandale

Supervisor: Frederic Font

August 2018

Master Thesis in Sound and Music Computing

Universitat Pompeu Fabra

Musical Instrument Recognition in Multi-Instrument Audio Contexts

Venkatesh Shenoy Kadandale

Supervisor: Frederic Font

August 2018

Acknowledgement

Firstly, I would like to thank my thesis supervisor Frederic Font for agreeing to supervise this project and for helping me with the directions and feedback throughout the project time-line. A special thanks to Olga Slizovskaia and Siddharth Bhardwaj for patiently answering all my questions related to deep learning even when they were away on vacation.

I am very grateful to Jordi Pons for directing me to a very well written repository containing the deep learning source code. My hearty thanks to Eduardo Fonseca for his insights on deep learning. I would also like to thank Alastair Porter for helping me with the setup of computational resources and software. A special thanks to Xavier Serra and Perfecto Herrera for the insightful thesis related discussions.

I would like to acknowledge the support that I received from the most friendly and approachable faculty members, researchers and the masters students at Music Technology Group (MTG), UPF. Finally, I would like to thank my parents for supporting me throughout the master's journey.

Abstract

Automatic musical instrument recognition is an important aspect of machine listening. In this project, we deal with instrument recognition in the multi-instrument audio contexts. We evaluate the performance of a traditional machine learning method in juxtaposition with a deep learning method in a supervised multi-label multi-output machine learning approach. We also tune a set of analysis parameters: {analysis window size, hop size, binarization threshold} to improve the performance. We investigate the possibility of improving the instrument recognition performance by using alternative data representations along with the original data. We consider two such sets of alternative data representations: 1) LRMS (left, right, mid, side) channel audio data derived from the stereo audio, and 2) The harmonic and residual representations derived from the original audio. We propose two different strategies to combine the models built using each of the data representation sets and evaluate their performance. Finally, we use the best combination strategy to merge the capabilities of individual models to improve the overall instrument recognition performance. With the shortlisted set of analysis parameters and the best combination strategy, we achieve an improvement of 14.25% in the macro f-score and 24.17% in the exact match ratio with respect to the baseline performance reported for our dataset.

Keywords: Musical Instrument Recognition, Multi-Instrument Audio, Deep Learning, Alternative Data Representations

Contents

Chapter 1

Introduction

Humans have this incredible ability to identify a musical instrument just by listening to its sound. The listening activity gets more engaging when multiple instruments are played simultaneously. Despite the complexity of the task, the humans can still identify the instruments up to certain number of instruments and certain mixing configurations. The automatic instrument recognition task aims at imparting this human ability to machines.

Of late, the world of digital media is growing at a very fast pace. It is extremely difficult for humans to manage a database of such a size manually. In order to manage the audio content in them, it is important to empower the machines to identify specific information about the audio and generate tags. Of all these information, musical instrumentation is an important one. The knowledge of musical instrumentation could give insights regarding the genre and style of music and potentially help in segregating music content. Such tags with the instrumentation information could also enable the users to browse through specific sounds using search queries in text format. The knowledge about the instrumentation could also be used by the music streaming applications to automatically equalize the audio. This capability is also an important milestone on the way to machine listening. To this day, there is no one stop solution for automatic musical instrument recognition and the research continues. We would like to run a set of experiments that could possibly enhance

the performance of musical instrument recognition task.

The instrument recognition problem statement itself can be thought of in single instrument audio contexts and multi-instrument audio contexts. The methods for instrument recognition can be widely segregated into traditional machine learning methods and deep learning. Initially, the traditional machine learning methods were used to train the machines to identify solo-instrument sounds. These methods were investigated further in the context of multi-instrument sounds. The last decade saw a rise in application of deep learning methods in machine learning problems including instrument recognition. Along with these developments, new datasets have been created for specific music information retrieval (MIR) tasks including instrument recognition. Looking at all these factors, we can infer that now is a really good time for further research in musical instrument recognition. We would like to identify gaps and areas for improvement in the existing research works related to instrument recognition in multi-instrument audio contexts and experiment with methods to improve up on them.

1.1 Motivation

Socrates said "The perfect human being is all human beings put together, it is a collective, it is all of us together that make perfection." In our opinion, this ideology could be extended to the instrument recognition models as well - "The perfect instrument recognition system is all the instrument recognition models put together". This idea has been the main motivation behind our thesis. Even when we consider a single dataset, different classifiers (like support vector machines, decision trees, k-nearest neighbour classifiers) learn differently from the same dataset which is also reflected in the difference in the results obtained by each of them. Each classifier could possible gain specific instrument recognition capabilities that no other classifier could achieve. In fact, the ensemble machine learning methods make use of these unique capabilities of group of such classifiers and combine them into a hybrid classifier in an effort to improve the overall performance. Apart from using different classifiers on the same dataset, we could also consider deriving additional datasets

from the original dataset. For example, if we have stereo audio files, we could extract the left, right, mid and side channel data and build separate datasets for each of the channels. Such datasets could possibly project a different information on to the recognition models. In this thesis, we are interested in investigating the impact of using alternative representations of the dataset on instrument recognition with both traditional machine learning method and advanced deep learning method.

[Wiggins, 2009] points out that none of the datasets that we use for MIR tasks is music, but are merely representations of music, and are rather incomplete. The author supports the idea further by adding that the score, the sheet music, the midi files and even the audio files, taken alone or combined, do not define music completely. He further emphasizes that music is a phenomenon that occurs in the mind of a listener. With regard to this specific research problem of automatic instrument recognition, we are trying to impart the instrument recognition skills of a human to a machine. The difference here is that the input data for humans in music but the input data for the machines is something which is a representation of music and not the music itself. This difference makes the automatic instrument recognition task in machines a tough problem to solve. The concept of glass-ceiling effect comes into the discussion here since we are dealing with the representation of music rather than music itself and there is only limited information that one can extract from such an incomplete representation of music. A part of the MIR research community, believes that no matter how good the algorithm is, the performance in any machine listening task can not get better beyond a certain point because of this glass-ceiling effect.

At this juncture, we would like to introduce an analogy between automatic instrument recognition and a blind person trying to understand the figure of an elephant by feeling it though his hands in specific places (note that the elephant won't allow anyone to touch everywhere!). The blind person could touch a specific part of the elephant and believe that the whole elephant is the part that he/she touched. We can now understand incomplete nature of this blind person's understanding. Now, let us invite other blind people to touch the elephant. They touch different parts of

the elephant and develop their own limited understanding of the figure of elephant. We let them talk and share 'their versions of elephant'. After this discussion, the group of blind people have a better understanding of the elephant than they had before. This is exactly what we intend accomplish in this thesis. We use multiple instrument recognition models built out of different representations of the dataset and analyze results obtained using each of the models and their combinations. We would like to build up on the strengths of each model by combining them. These are merely our efforts to see if we can improve the automatic instrument recognition performance. However, we need to realize that, even with a million blind people discussing their experiences with the elephant, none of them will fully understand the figure of the elephant, let alone understanding the elephant as an individual beyond the body physique. Akin to how the blind people discuss to understand the elephant's physique better, we are interested in combining the intelligence of individual models and studying the performance of the resulting combinations.

1.2 Objectives

The following are the list of goals that we would like to accomplish through this thesis:

- To juxtapose and compare the performance of a traditional machine learning method and a deep learning method in the context of instrument recognition in multi-instrument audio contexts.

- To illustrate the advantages of using alternative representations of the data along with the original data in improving the performance of instrument recognition.

- To propose a combination strategy that could combine multiple models built from different data representations to improve the overall performance.

1.3 Structure of the Report

The rest of the thesis is organized as follows. We review the-state-of-the-art methods used in instrument recognition in Chapter 2. We segregate these approaches into traditional method and deep learning method and review them in chronological order of their application. In particular, we highlight the research work related to instrument recognition in multi-instrument audio contexts and identify gaps/ areas for improvement. In Chapter 3, we delineate the methodology used in our thesis with emphasis on the chosen dataset, experiment set-up and evaluation criteria. The experiments cover a wide range of configurations of dataset: analysis window size, hop size and binarization threshold along with merging strategy for LRMS channels and harmonic and residual component data representations with the application of traditional method as well as deep learning. In Chapter 4, we report the results obtained from each of our experiments and discuss our inferences. Chapter 5 is dedicated to conclusion and future work. In this chapter, we highlight the findings and contributions of our work and give directions to future work. We share the link to the repository containing the source code for reproducing our results in the Appendix.

Chapter 2

State of the Art

In this section, we review the state-of-the-art practices used in automatic musical instrument recognition so far. Some of the terminologies used in the literature - “polyphony”, “polytimbral” are a bit ambiguous and we would like to clear this up before going further. “Polyphony” can be interpreted as a phenomenon wherein multiple notes are being played simultaneously on the same instrument (for e.g. left and right hand parts in a piano). It can also represent a use case wherein the notes are being played simultaneously on multiple instruments. In most of these references, the word "polyphony" is used in the same context as "multi-instrument". Hence, the confusion. The word polytimbral, on the other hand, means multiple timbres. A single instrument can produce sounds of different timbres based on the way it is played. For example, a violin played in tremolo style and a violin played in pizzicato style have remarkably different timbres. [Agostini et al., 2003] reports that the instrument recognition performance improves when pizzicato-sustain discrimination is considered rather than instrument-wise or instrument-family-wise discrimination for categorizing data. Hence, it is more appropriate to segregate the instrument sounds based on timbre rather than the instrument source. The problem with the timbre based approach is that the dataset labels need to inherently include this information (e.g. ‘violin pizzicato’ rather than just ‘violin’ wherever appropriate). But, in most of the cases (including ours), the instrument sound datasets are labeled

only with instrument names, devoid of any timbre related information. Therefore, for this thesis, we consider all the sounds produced by a particular instrument to be belonging to one category, regardless of the timbre. Throughout the report, we use the word "multi-instrument" instead of "polyphony" or "polytimbral", since we deal with instrument recognition in multi-instrument audio contexts.

The earliest published research works in instrument recognition focus on identification of single instrument sounds in monophonic recordings (e.g. [Kaminsky and Materka, 1995], [Martin and Kim, 1998]). They report the use of traditional machine learning methods like Artificial Neural Network (ANN), Nearest Neighbor Classifier (NNC) and Support Vector Machines (SVM). To train the classifiers, these methods use handcrafted features that are extracted from the raw audio. During the earliest research efforts, only the time domain features like short-time energy and the zero crossing rate were considered, as in [Kaminsky and Materka, 1995]. Soon, the researchers started including features from spectral domain (e.g. spectral centroid, spectral roll-off, spectral flux, spectral flatness) and eventually the cepstral domain (MFCCs) features as reported in [Martin and Kim, 1998] and [Marques and Moreno, 1999], respectively. The feature selection strategies like Principal Component Analysis (PCA) were very commonly used to deal with the "curse of dimensionality". Some of the notable large scale datasets for the instrument recognition are MIS ([UIOWA, 1997]); RWC ([Goto et al., 2002]); IRMAS ([Bosch et al., 2012]) and MedleyDB ([Bittner et al., 2014]).

The deep learning architectures were already realized in 1990s, but their implementations were computationally expensive. In 2009, the Graphics Processing Units (GPUs) were incorporated in the deep-learning systems and [Raina et al., 2009] determined that this increased the computational speed by around 100 times. From this stage onward, the GPU based deep-learning systems got more popular. For the large scale datasets, it is advantageous to use deep-learning methods. From Fig. 1, it is clear that the performance of deep learning systems improve with more data, while the performance of traditional machine learning algorithms do not increase with increase in dataset size beyond a threshold size. There is a substantial increase

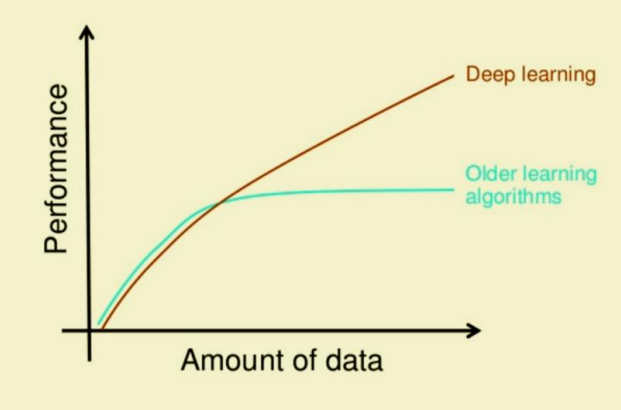

Figure 1: Performance of Deep Learning Vs. Traditional Machine Learning Algorithms. Image taken from [Mahapatra, 2018]

in number of deep learning articles in MIR publications since 2015 (6 in 2015 and 16 in 2016) as reported in [Choi et al., 2017]. One of the goals of our work is to juxtapose and compare the performance of traditional machine learning method and deep learning method for musical instrument recognition in multi-instrument audio contexts.

2.1 Solo-Instrument Audio Context

The human ability to recognize instruments in a mixture of sounds largely relies on our memory of the individual instrument sounds and the ability to recognize them individually. Instrument recognition research was first investigated in solo-instrument audio contexts. The studies [Kaminsky and Materka, 1995], [Fujinaga, 1998], [Martin and Kim, 1998], [Kostek, 1998], [Fraser and Fujinaga, 1999], [Fujinaga and MacMillan, 2000] and [Kaminskyj, 1998] address the instrument recognition using isolated notes of different pitches with one example from each instrument. The studies [Dubnov and Rodet, 1998], [Marques, 1999], [Brown, 1999], [Martin, 1999] and [Brown et al., 2001] address a relatively more realistic use case : instrument recognition with monophonic phrases. The Table 2 and Table 3 contain the results of instrument recognition with isolated notes and monophonic phrases, respectively. These tables clearly indicate that the performance numbers fell when we moved from the isolated note context to monophonic phrase context. The more closer we

Study	Percentage correct	Number of instruments	
[Kaminskyj95]	98	4	guitar, piano, marimba and accordion
[Kaminskyj00]	82	19	
[Fujinaga98]	50	23	
[Fraser99]	64	23	
[Fujinaga00]	68	23	
[Martin98]	72 (93)	14	
[Kostek99]	97	4	bass trombone, trombone, English horn and contra bassoon
	81	20	
[Kostek01]	93	4	oboe, trumpet, violin, cello
	90	18	

Figure 2: Performance of instrument recognition with isolated notes. [Table taken from [Eronen et al., 2001]]

Study	Percentage correct	Number of instruments	
[Dubnov98]	not given	18	
[Marques99] 0.2 seconds	70	8	
2 seconds	83	8	
[Brown99]	94	2	oboe, saxophone
[Brown01]	84	4	oboe, sax, flute and clarinet
[Martin99]	57 (75)	27	

Figure 3: Performance of instrument recognition with monophonic phrases. [Table taken from [Eronen et al., 2001]]

get to the real world use case, the harder the research problem gets. Going by this logic, instrument recognition in multi-instrument audio contexts will be even harder than in monophonic contexts. [Herrera-Boyer et al., 2003] extensively reviews the state-of-the-art methods used in automatic classification of isolated musical instrument sounds. MFCCs are reported to be one of the best set of features for automatic musical instrument recognition in [Martin, 1999] and [Eronen et al., 2001].

2.2 Multi-instrument Audio Context

In case of the multi-instrument audio contexts, the goal is to identify all the instruments present in the mix. [Fuhrmann et al., 2012] identifies three methodologies used for instrument recognition in multi-instrument contexts:

- **Pure Pattern Recognition** The recognition algorithms are run on multi-instrument sound samples directly, to look for individual instrument or group of instruments without much pre-processing as seen in [Essid et al., 2006].
- **Enhanced Pattern Recognition** The multi-instrument samples are pre-processed using signal processing based methods like source separation or multi-pitch estimation before running the instrument recognition algorithms on them as implemented in [Heittola et al., 2009].
- **Template Matching** The distances between the abstracted class representations and multi-instrument samples are evaluated to arrive at the class labels. [Cont et al., 2007] makes use of this approach.

[Fuhrmann et al., 2012] reviews most of the state-of-the-art approaches covering all the above three methodologies and reports the dataset description, recognition algorithm and the results. Fig. 4 contains this information.

Instrument recognition in the multi-instrument audio contexts is a multi-class multi-label machine learning problem : each sample contains multiple labels, with each label corresponding to one of the instrument categories. [Madjarov et al., 2012] points out the following two types of approaches to deal with the multi-class multi-label problems:

- **Algorithm Adaptation** This approach adapts and extends the single label classification algorithms to multi-label scenarios. Traditional machine learning algorithms like decision trees, adaboost and ranking ranking support vector machines have been adapted for use in multi-label classification scenarios.
- **Problem Transformation** The problem transformation approaches transform the multi-label classification problems into multiple single label multi-class problems. The single label classification algorithms like Support Vector Machines (SVM) could then be applied directly to this transformed problem.

The progress in MIR research, in general, critically depends on availability of good datasets. A good dataset is large in size (the larger the better), well annotated,

	Data & experimental settings					Algorithmic specifications				Evaluation		
Author	Poly.	Cat.	Type	Coll.	Genre	Class.	A priori	PreP.	PostP.	#Files	Metric	Score
Simmermacher et al. (2006)	4	4	real	pers.	C	SVM	×	×	×	10	Acc.	0.94
Essid et al. (2006a)*	4	12	real	pers.	J	SVM	×	×	✓	n.s.	Acc.	0.53
Little & Pardo (2008)	3	4	art. mix	IOWA	–	SVM	×	×	✓	20	Acc.	0.78
Kobayashi (2009)*	n.s.	10	real	pers.	P,R,J,W	LDA/RA	×	×	×	50	Acc.	0.88
Fuhrmann & Herrera (2010)*	10	12	real	pers.	C,P,R,J,W,E	SVM	×	×	✓	66	F	0.66
Eggink & Brown (2003)	2	5	real	pers.	C	GMM	×	✓	×	1	Acc.	1.0
Eggink & Brown (2004)	n.s.	5	real	pers.	C	GMM	×	✓	×	90	Acc.	0.86
Livshin & Rodet (2004)	2	7	real	pers.	C	LDA/kNN	×	✓	×	108	Acc.	n.s.
Kitahara et al. (2006)	3	4	syn. MIDI	RWC	C	HMM	×	✓	✓	n.s.	Acc.	0.83
Kitahara et al. (2007)	4	5	syn. MIDI	RWC	n.s.	Gauss.	✓	✓	✓	3	Acc.	0.71
Heittola et al. (2009)	6	19	art. mix	RWC	–	GMM	✓	✓	✓	100	F	0.59
Pei & Hsu (2009)	3	5	real	pers.	C	SVM	✓	✓	✓	200	Acc.	0.85
Barbedo & Tzanetakis (2011)*	7	25	real	pers.	C,P,R,J	DS	×	✓	✓	100	F	0.73
Cont et al. (2007)	2	2	real mix	pers.	n.s.	NMF	×	×	×	4	Acc.	n.s.
Leveau et al. (2007)	4	7	real mix	pers.	n.s.	MP	×	×	✓	100	Acc.	0.17
Burred et al. (2010)	4	5	art. mix	RWC	–	prob. dist.	×	✓	×	100	Acc.	0.56

Figure 4: Comparative view on the approaches for recognizing pitched instruments from polytimbral data. Asterisks indicate works which include percussive instruments in the recognition process. polyphonic density (Poly.), number of categories (Cat.), type of data used (Type), the name of the data collection (Coll.), the classification method (Class.),imposed a priori knowledge (Apriori), any form of preprocessing (PreP.) and post-processing (PostP.), and the number of entire tracks for evaluation (Files). Abbreviations for the evaluation metric refer to Accuracy (Acc.) and F-measure (F). Furthermore, the legend for musical genres include Classical (C), Pop (P), Rock (R), Jazz (J), World (W), and Electronic (E). The three blocks are pure, enhanced pattern recognition, and template matching with respect to the recognition approach. [Table taken from [Fuhrmann et al., 2012]]

contains truthful records obtained using the right practices and diverse enough for the intended purpose. For example, an ideal dataset for musical instrument recognition is the one which covers most of the musical instruments recorded in different environments, played in different styles and genres, perhaps by different artists. The lack of publicly available good quality datasets has always been a big challenge for the MIR research community. In the Fig. 4, we notice that large number of datasets (column 'Coll.') are 'pers.' which indicate that personal audio collections were used for the respective research activities. This is a set-back because the reported results on such datasets could not be validated or challenged by other researchers.

As discussed earlier in this section, the initial instrument recognition efforts were in isolated instrument contexts. The datasets like MIS ([UIOWA, 1997]),RWC ([Goto et al., 2002]) and MUMS ([Opolko and Wapnick, 1989]) were created for this

purpose. At that time, there were no publicly available datasets for instrument recognition in multi-instrument contexts. The researchers tried linearly combining the sounds from isolated instrument collection to create multi-instrument audio contexts as in [Heittola et al., 2009] and [Burred et al., 2009]. Using this approach, the instrument recognition accuracy was only around 59% for audio sample with six-note polyphony in [Heittola et al., 2009] and 61.4% with four voices in [Burred et al., 2009]. It is clear that there is lot of scope for improvement. Also, in these two studies, the training set and test set, both of them were generated by linearly mixing the isolated instrument samples from RWC dataset. The dataset created in this way is far too different from the real world music and we could expect even lower performance numbers if these methods are used for real world music. As pointed out in [Bosch et al., 2012], the real world music comprises of effects like reverbs and delays which make the mix more complex than the artificially generated mixes resulting from the random linear combination of sounds.

Of late, MIR community has been putting efforts in creating large-scale datasets which could be potentially used for musical instrument recognition in multi-instrument audio contexts, among several other tasks. One of the most recent efforts being the Freesound Datasets [Fonseca et al., 2017] which is "a platform for collaborative creation of open audio collections labeled by humans and based on Freesound content" [Font et al., 2013]. Though the Freesound Datasets do not primarily focus on musical instruments, but it has some samples with instrumentation information. IRMAS dataset, published by [Bosch et al., 2012], consists of real world music samples with annotations for predominant instrument in the mix. IRMAS dataset covers 11 pitched instruments. The training data comprises of 6705 audio files of around 3 second duration with a single predominant instrument. The testing data, spanning over 2874 excerpts, interestingly, comprises of more than one predominant instrument. MedleyDB, published by [Bittner et al., 2014], consists of around 122 full-length songs of duration ranging from 20 seconds to 600 seconds. Additionally, the annotations for instrument activation and stems [1] for individual instrument categories are provided. There is a dedicated stem for each instrument category.

[1] Stems are the constituent audio tracks of a mix, which, on combining, gives the entire mix.

MedleyDB contains music from 9 different genres making it diverse.

[Bosch et al., 2012] evaluates performance of instrument recognition using different segregation and classification algorithms on IRMAS dataset. Source separation is implemented as a pre-processing step. The novelty in this paper is the use of panning information to split the track into left (L), right (R), mid (M=L+R) and side (s=L-R) streams. Using the data from LRMS channels improves the performance of predominant instrument recognition by around 19 p.u. Data pre-processing using a source separation method called FASST (A Flexible Audio Source Separation Framework) is also investigated in this paper. FASST is a computationally complex algorithm which improves the performance score by around 32 p.u. The authors suggest using the LRMS channel data as one of the easiest ways to improve the performance of instrument recognition task.

2.3 Towards Deep Learning

One of the main challenges of using the traditional machine learning methods is feature engineering. The researchers need to have the domain knowledge to craft out a set of features that formed a very good representation of the raw data specific to the targeted application. But, here the possibilities are endless because one could think of infinitely many feature definitions relevant to the application context. Deep learning methods relieve us from this dilemma since these algorithms are designed to identify the patterns in any data without the need for feature engineering. Recurrent Neural Networks (RNN) and Convoluted Neural Networks (CNN) are the two most popular deep learning architectures. RNNs have the “memory" and can be used to model long time dependencies (e.g speech, text) suitable for sequential inputs while CNNs can be used for finding patterns in highly correlated local contexts (e.g images, video). A tutorial deep learning approaches for MIR applications presented in [Choi et al., 2017] is quite exhaustive and informative.

With regard to the application of deep learning architectures to musical instrument recognition, we mainly review two reference papers. [Li et al., 2015] presented an

Models	Accuracy	Exact Match	Precision	Recall	F-micro	F -macro
Audio + CNN	**82.74%**	**25.78%**	0.7560	**0.6888**	**0.7208**	**0.6433**
MFCC + Random Forest	82.13%	17.53%	**0.7908**	0.5400	0.6418	0.4471
MFCC + Logistic Regression	81.80%	18.17%	0.7457	0.5857	0.6561	0.4840
Predict Majority Class	70.37%	9.95%	0.5001	0.4602	0.4793	0.1801

Figure 5: Baseline performance of instrument recognition for MedleyDB. [Table taken from [Li et al., 2015]]

end-to-end CNN system for automatic musical instrument recognition by directly using the raw audio as the input. They compare the instrument recognition performance of CNN system and traditional machine learning algorithms built around MFCC features on MedleyDB dataset. Their performance numbers are reported in the Table 5. They report an exact match percentage of 25.78% with deep learning and 17.53% with traditional machine learning approaches on the test set evaluation. We note that even the f-measures get better in case of the deep learning approach. We consider the results reported by [Li et al., 2015] as our baseline.

The other important reference paper in deep learning topic with regard to this thesis is [Han et al., 2017]. [Han et al., 2017] proposes CNN for predominant instrument recognition with IRMAS dataset and uses the mel-spectrogram of the audio as input. They also experiment with the hyper-parameter tuning. They report that the deep CNNs perform as good as the traditional machine learning algorithms, if not better. Their CNN architecture is depicted in the Fig. 6. The main difference in architectures between [Han et al., 2017] and [Li et al., 2015] is that the former uses mel-spectrogram of the audio segment as the input to the CNNs while the latter uses the raw audio as input.

2.4 Conclusion

In this thesis, we consider the performance of automatic instrument recognition in [Li et al., 2015] as the baseline and explore ways of making it better. We would like to juxtapose and compare the performance of traditional machine learning method and deep learning method. We choose MedleyDB as our dataset. In our opinion,

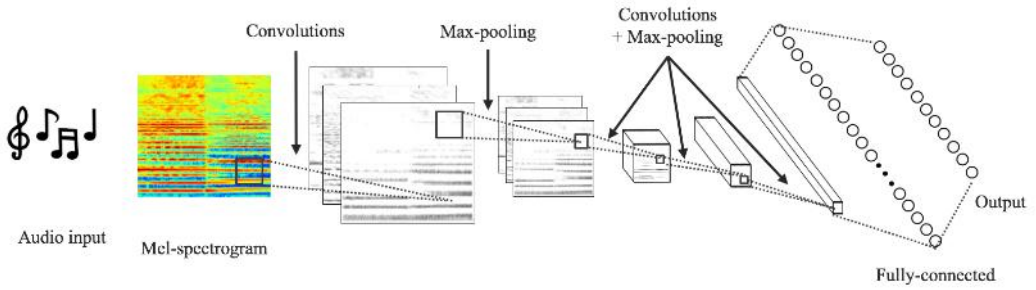

Figure 6: A deep CNN architecture used by [Han et al., 2017] for instrument recognition. [Figure taken from [Han et al., 2017]]

among the datasets available as of this day, MedleyDB is the most suitable dataset for instrument recognition task in multi-instrument audio contexts for two reasons: 1) The diversity of genres and instruments involved; 2) The instrumentation information is well annotated. Though the dataset by itself is small (as it contains only 122 songs of duration ranging from 20 seconds to 600 seconds), we could not find any larger datasets with this quality of data. Surprisingly, we could not find many projects in musical instrument recognition which make use of this dataset. The dataset labels are instrument-wise activation confidence scores which indicate the probability of the presence of the instrument in the audio clip at a given instant. Therefore, we have a multi-output multi-class regression problem here. We treat this problem as a pure pattern recognition task by taking in the audio data directly without any pre-processing like instrument source separation. Additionally, this dataset comes with stems of individual instrument tracks which we intended to make use of. However, this could not be done within the scope of this project, but we would like to share some insights in this regard for future work later.

With regard to the traditional machine learning approach, we transform the multi-output multi-class regression problem into multiple single output multi-class regression problems. On each of these single output regression problems, use the support vector regression (SVR) models because the support vector machines (SVM) are established as one of the best classifiers in the state-of-the-art instrument recognition literature discussed earlier. The time, frequency and cepstral domain features are extracted using Essentia's music extractor, thanks to [Bogdanov et al., 2013].

For the deep learning part, we investigate the performance with mel-spectrogram as input to CNNs using the architecture published by [Han et al., 2017]. Also, we study the effect of using different representations of audio data in improving the performance. In this regard, consider two different representation sets : 1) LRMS channel data; and 2) the harmonic and residual components of the audio. The motivation to use LRMS channel data came from [Bosch et al., 2012] which claims a performance improvement of 19 p.u in case of IRMAS dataset by making use of panning information. The motivation for using the harmonic and residual components came from experimental trials of music instrument recognition in single instrument sounds from Good-Sounds dataset [Romani Picas et al., 2017] conducted by [Shenoy Kadandale, 2018]. During these trials, we noticed that some instruments were recognized better in the harmonic model while some instruments were identified well in the residual model. Our idea is to build instrument recognition models for each of the data representations and combine them into a hybrid model which cherry-picks the strengths of each constituent model. The combination strategy is discussed in detail in the Section 3.5 of Chapter 3. As an extension to the experimentation in [Li et al., 2015], we investigate the effect of change in analysis window size and hop size on the exact match ratio and the f-measure metrics.

Chapter 3

Methodology

In this chapter, we describe the methodology applied in this project. To begin with, we discuss the data pre-processing, where we explain how the data is prepared for further analysis. In the next sections, we discuss our implementation of traditional machine learning algorithms and deep learning CNNs. Then, we describe the evaluation metrics which indicate the performance of a particular method in the instrument recognition task. Finally, we explain how we intend to make use of the two alternative representations of the data - 1) LRMS channel data and 2) harmonic and residual components of the audio, to improve the performance of both the traditional and deep learning methods.

3.1 Data Pre-processing

MedleyDB provides us the audio (in WAV format) files for each of the 122 songs along with their respective instrument annotations. The instrument annotations are provided in the form of activation confidences for each instrument present in a particular mix. The activation confidences indicate the probability of the instrument being active in the mix at a particular instant. The activation confidences were determined using a standard envelope tracking technique on each stem involving half-wave rectification, compression, smoothing and down-sampling. More information on this procedure can be found in [Bittner et al., 2014].

The dataset does not come with predefined splits for training data and testing data. However, to compare the results of our experiments with the baseline, it is necessary to use the same set of training and testing data that was used by [Li et al., 2015]. The full-length audio clips need to be sliced into smaller segments, each of which becomes a sample in our machine learning problem. Just as it was done by [Li et al., 2015], we assign 80% of these samples for training and 20% for testing. [Li et al., 2015] does not explicitly share the dataset splits, but directs us to an algorithm that splits the dataset into training and testing components in the best possible way. This splitting algorithm was published by [Sechidis et al., 2011]. It takes into account two main considerations : 1) There shouldn't be any labels in the test set that didn't occur in the training set 2) There shouldn't be segments belonging to the same audio clip in both the training set and the test set. [Li et al., 2015] considers one second long non-overlapping segments as samples. However, we investigate the performance of instrument recognition with the segment lengths (in seconds) : {1, 2, 3, 4, 5} with hop factors [1] of 25%, 50% and 100%. Also, unlike in [Li et al., 2015], we create a dataset for each of the LRMS channels by using the panning information in the original audio. L and R channel data are directly available from the stereo audio. M is obtained by averaging out the L and R. The difference between L and R gives us the side channel data.

For each of the segments, the global label for each instrument was determined by taking the maximum of moving average of the activation confidence values for that particular instrument. The annotations cover 82 instruments. However, all the instruments are not equally distributed. We group all the instruments appearing in less than 20 songs into 'OTHER' category. As a result, we have 11 classes for classification: acoustic guitar, clean electric guitar, distorted electric guitar, drum set, electric bass, fx/processed sound, OTHER, piano, synthesizer, violin and voice.

[1] We define hop factor as the percentage of number of samples between the starting points of consecutive window frames with respect to the window size.

3.2 Traditional Machine Learning

We start our instrument recognition task with the traditional machine learning approach. In this approach it is necessary to extract a set of features from the raw data relevant to instrument recognition. We use Essentia's music extractor to extract the low-level features pertaining to the temporal, spectral and cepstral domains, from each of the raw audio samples. More details regarding these features can be found in [Bogdanov et al., 2013]. The labels for each sample is a set of activation confidence scores, each one corresponding to respective instrument. We would like to treat this as a regression problem such that the predictions for the test set will also be a set of activation confidence scores. Next step is to use Support Vector Regressor (SVR) to fit the training data and then predict the labels of the test set. A threshold is required to binarize the predicted scores for each instrument into 'is present' (if the predicted score is greater than threshold) or 'not present' (if the predicted score is lesser than threshold) case. We investigate the performance of a traditional machine learning method with different configurations of data. By the word 'configurations', we are referring to the duration of the sample, the extent of overlap in samples when they were sliced from the parent audio clip and even the threshold to binarize the continuous labels. The idea is to pick the configuration which gives the best result and use this configuration to do further experiments which are discussed in the next sections.

3.3 Deep Learning

[Li et al., 2015] uses raw audio as the input to the CNNs. This makes the training stage time consuming since raw audio is bulky (44100 samples in a second). Instead, we use the mel-spectrogram of the audio clip as input, just like the implementation of deep learning method in [Han et al., 2017] and compare the performance with the baseline. We use the configuration of the data that gives best results for the experiments with traditional machine learning approach. We apply the threshold from the selected configuration to binarize the labels before feeding the data to

Input size	Description
1 × 43 × 128	mel-spectrogram
32 × 45 × 130	3 × 3 convolution, 32 filters
32 × 47 × 132	3 × 3 convolution, 32 filters
32 × 15 × 44	3 × 3 max-pooling
32 × 15 × 44	dropout (0.25)
64 × 17 × 46	3 × 3 convolution, 64 filters
64 × 19 × 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 (0.50)
11	sigmoid

Figure 7: ConvNet structure proposed by [Han et al., 2017]. [Table taken from [Han et al., 2017]]

CNNs.

We adapt the CNN architecture published by [Han et al., 2017] to our use case. The code implementing this architecture, provided by [Pons et al., 2017], is used. The Tab. 7 contains the details regarding the architecture proposed by [Han et al., 2017]. This architecture makes use of multiple fixed-length rectangular filters of size 3×3 with a stride size of 1. [Han et al., 2017] deals with single label output in the training phase since IRMAS dataset has only one predominant instrument for every sample in the training set. However, our training data samples contain multiple labels. Hence, we make two main changes with regard to the reference architecture : 1) we use the sigmoid layer instead of softmax layer as the final layer, 2) we use the binary cross entropy as the loss function instead of categorical cross entropy. The resulting predictions are continuous numbers between 0 and 1 for each label. We use the same threshold from the shortlisted data configuration to binarize the predicted labels.

3.4 Evaluation Metrics

The evaluation scheme is based on comparison between the predicted labels and the ground truth for the samples from the test set. To assess the performance of all our approaches in instrument recognition, we look at the four metrics : exact match ratio, instrument-wise f-score, the overall micro and macro f-score. The mathematical expressions used to define these metrics have been adapted from [Sorower, 2010] and [Bosch et al., 2012].

Let T be the multi-label dataset containing n multi-instrument sound samples $(x_i,\ Y_i),\ 1 \leq i \leq n,\ x_i \in \mathbb{X},\ Y_i \in \mathbb{Y} = \{Y_i\}$, with $Y_i = \{0,1\}^k = \{Y_{i_j}\}$ for $j = 1...k$ with an instrument-set $\mathbb{I}$ and $|\mathbb{I}| = k$. Let h be a multi-label classifier and $Z_i = h(x_i) = \{0,1\}^k = \{Z_{i_j}\}$ for $j = 1...k$ be the set of predictions for x_i determined by h.

The exact match ratio indicates the percentage of instances where all the labels were predicted correctly. In a multi-label multi-class problem this measure can be really harsh owing to the difficulty of getting all the labels right without accounting for partially correct predictions. In fact, the baseline exact match ratio for this dataset was 25.78% reported by [Li et al., 2015].

$$Exact\ Match\ Ratio = \frac{1}{n}\sum_{i=1}^{n} \hat{I}(Y_i = Z_i)$$

where $\hat{I}$ is the indicator function.

To understand f-score, we need to understand precision and recall. Precision is the ratio of number of true positives to the number of predicted condition positives. Recall is the ratio of number of true positives to the number of condition positives. The condition positives are the real positive cases in the data. The f-score is the harmonic mean of precision and recall. Let tp_j be true positives, tn_j be true negatives, fp_j be false positives and fn_j be false negatives for each instrument I_j in $\mathbb{I}$.

For each instrument, the precision and recall can be determined as:

$$P_j = \frac{tp_j}{tp_j + fp_j} = \frac{\sum_{i=1}^{n} Y_{i_j} Z_{i_j}}{\sum_{i=1}^{n} Z_{i_j}}$$

$$R_j = \frac{tp_j}{tp_j + fn_j} = \frac{\sum_{i=1}^{n} Y_{i_j} Z_{i_j}}{\sum_{i=1}^{n} Y_{i_j}}$$

The instrument-wise f-score (F1 measure) can then be obtained using :

$$F_j = \frac{2P_j R_j}{P_j + R_j}$$

Now we can consider macro and micro averages of these metrics for the entire prediction set including all the instrument labels. The macro average is the average of precision and recall values determined separately for each instrument label.

$$P_{macro} = \frac{1}{|I|} \sum_{j=1}^{k} P_j, \; R_{macro} = \frac{1}{|I|} \sum_{j=1}^{k} R_j$$

In case of the micro average strategy, the average is taken directly over the instances, which results in giving more weight to the instruments which are found in larger number of instances.

$$P_{micro} = \frac{\sum_{j=1}^{k} tp_j}{\sum_{j=1}^{k} (tp_j + fp_j)} = \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} Y_{i_j} Z_{i_j}}{\sum_{j=1}^{k} \sum_{i=1}^{n} Z_{i_j}}$$

$$R_{micro} = \frac{\sum_{j=1}^{k} tp_j}{\sum_{j=1}^{k} (tp_j + fn_j)} = \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} Y_{i_j} Z_{i_j}}{\sum_{j=1}^{k} \sum_{i=1}^{n} Y_{i_j}}$$

Hence the macro and micro F1 are defined as :

$$F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}},\ F_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}}$$

3.5 Alternative Data Representations

Most of the related research work in this topic, work with the original raw audio in mono format. Mono audio is just one way of representing the stereo audio, but there can be other alternative representations that could be derived from the source. One of the main goals of this thesis is to see if these alternative representations can improve upon the instrument recognition performance achieved using mono raw audio representation alone. We consider two such sets of alternative representations - 1) LRMS components of the stereo audio, and 2) harmonic and residual components of original audio. The idea is to treat these alternative representations as separate datasets, train the classifiers on each dataset, evaluate over respective test sets and then merge the predicted labels into a final set of labels such that there is an improvement in the evaluation metrics. Before merging, we binarize the continuous predicted labels of each alternative representation in the set with a particular threshold. During the binarization, values above the threshold are set to 1 and those below the threshold are set to 0.

With regard to merging the predictions obtained from different representations, we investigate two such merging strategies : hybrid max combination strategy and hybrid weighted combination strategy. For predicting the label for a particular instrument, the hybrid max combination strategy only considers the decision of the classifier which got the highest f-score on the training set evaluation of a particular representation set (eg LRMS). Unlike the hybrid max combination strategy, the hybrid weighted combination strategy considers the predictions of all the classifiers and assigns the label supported by the majority. Let $\mathbb{R} = \{R_m\}$, for $1 \leq m \leq M$ be one of the alternative representation sets of length M i.e $|\mathbb{R}| = M$. Let $\mathbb{D} = \{D_{R_m}\}$ be the set of training set samples obtained from each of the representations in the

set $\mathbb{R}$. Let $\mathbb{T} = \{T_{R_m}\}$ be the set of test set samples obtained from each of the representations in the set $\mathbb{R}$. Let $\mathbb{C} = \{C_m\}$ be the set of classifiers such that C_m is trained on D_{R_m}. Let $\mathbb{I} = \{I_j\}$, for $1 \leq j \leq k$ be the instrument label set with $|\mathbb{I}| = k$. When the classifiers in $\mathbb{C}$ are evaluated on the respective training sets in $\mathbb{D}$, we compute macro f-scores for a particular instrument I_j in each of the $\{D_{R_m}\}$ and get $\mathbb{F}_j = \{F_{m_j}\}$. Let these scores be normalized to $\mathbb{W}_j = \{W_{m_j}\}$. Let $p_j = \arg\max_m(\mathbb{W}_j)$. Let $\mathbb{Z}_j = \{Z_{m_j}\}$ be the label for instrument $I_j \in \mathbb{I}$ as predicted by each of the classifiers in the set $\mathbb{C}$. In case of hybrid max combination strategy, the label for I_j for the samples in $\mathbb{T}$ is $Z_{p_j}\}$ without any consideration for the predictions by other classifiers in the representation set. However, in case of the hybrid weighted combination strategy, the labels for instrument I_j for the samples in $\mathbb{T}$ are predicted as $\left[\sum_{m=1}^{M} Z_{m_j}.W_{m_j}\right]$ where '[ ]' is nearest integer function. This procedure is extended for all $I_j \in \mathbb{I}$. Suppose the normalized scores are {0.3, 0.1, 0.2, 0.4} and the predictions for the instrument I_j in a sample by these 4 classifiers are {0, 0, 0, 1}. We notice that the label '0' gets support of 0.6 (= 0.3 + 0.1 + 0.2) and the label '1' gets the support of 0.4. Hence, the majority vote criterion would finally assign the label '0' for the instrument I_j in that sample. Note that the hybrid max combination strategy would have assigned the label '1' which was a prediction by the single highest weighted classifier C_4 in this set for instrument I_j in that sample.

3.5.1 Preliminary Experiments

The motivation to investigate the effect of incorporating LRMS components of the stereo audio came from [Bosch et al., 2012]. [Bosch et al., 2012] reports that using the panning information (LRMS data) alone improves the instrument recognition results by 19 p.u with regard to the IRMAS dataset. We would like to see if it holds true in case of MedleyDB.

The motivation to analyze instrument recognition in the harmonic and residual components of the dataset came from an experiment that we conducted on Good-Sounds dataset. This dataset was published by [Romani Picas et al., 2015]. It contains

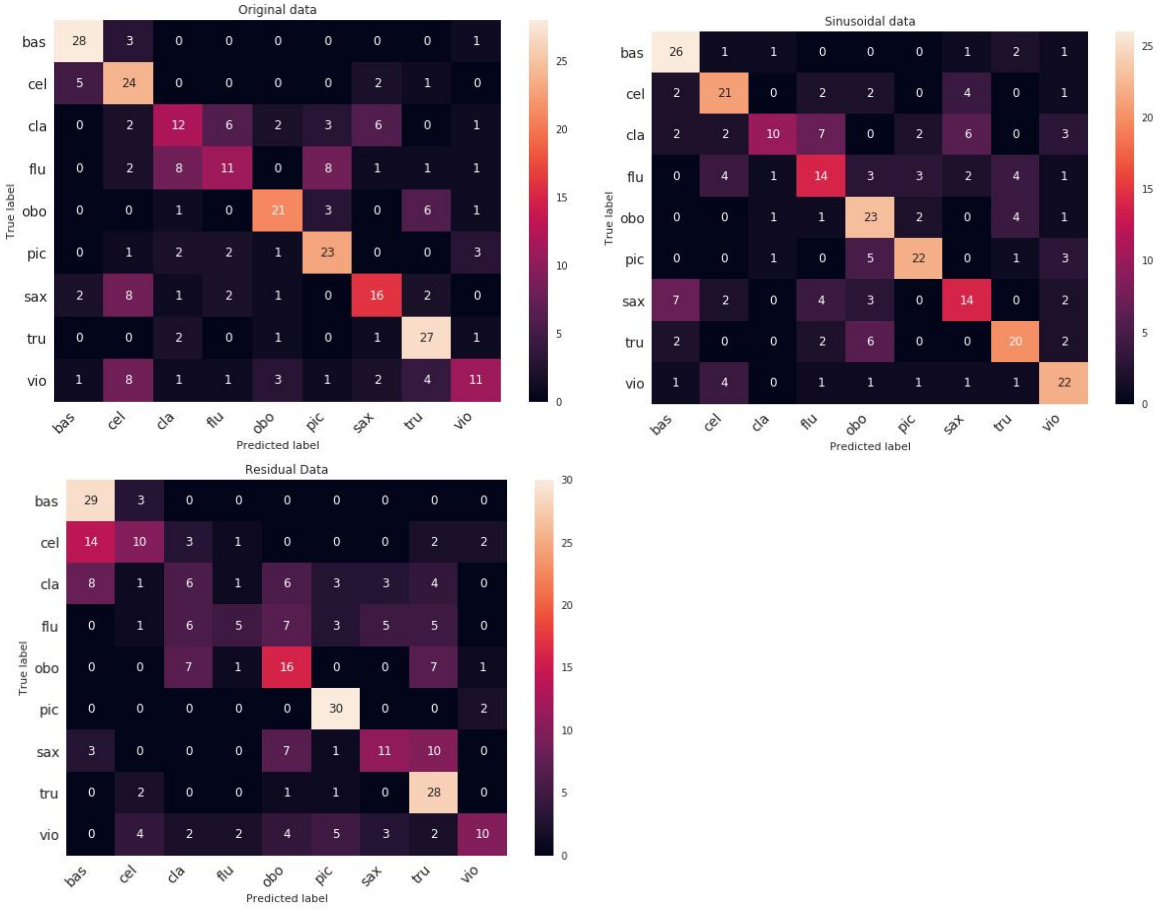

Figure 8: Confusion matrices for original (top left), sinusoidal (top right) and residual (bottom) datasets. Figures taken from [Shenoy Kadandale, 2018]

monophonic recordings of single notes and scales. The instrument set involved in this dataset are bass (bas), cello (cel), clarinet (cla), flute (flu), oboe (obo), piccolo (pic), saxophone (sax), trumpet (tru), violin (vio). We chose a subset of this data as this was supposed to be a simple experiment. This subset consisted of 159 samples from each instrument category which were randomly selected. The training set and test set were again randomly generated by splitting the chosen subset in 80%-20% ratio. Our experiment was to track the instrument recognition performance for each of the instrument in different representations of data. We had chosen original dataset and datasets formed by its sinusoidal and residual components as the three different representations of the dataset. The Python Notebook conducting the experiment and analyzing the results has been shared by [Shenoy Kadandale, 2018]. The Fig. 8 shows the confusion matrices obtained for the original, sinusoidal and

residual datasets. This experiment achieves an overall accuracy of around 60% for original and sinusoidal datasets and around 50% for the residual datasets. The confusion matrices show that the sinusoidal models perform better than the original models for flute (sinusoidal model accuracy=14/32, original model accuracy=11/32) and violin (sinusoidal model accuracy=22/32, original model accuracy=11/32, almost double!). Residual models perform better than the original models for bass (residual model accuracy=29/32, original model accuracy=28/32), piccolo (residual model accuracy=30/32, original model accuracy=23/32) and trumpet (residual model accuracy=28/32, original model accuracy=27/32). The experiment reveals that each of the representation helps in identifying certain instruments better. In our work, we investigate this with regard to the MedleyDB.

It is clear that the alternative representations have rich source of information that could potentially be used to improve the overall performance of instrument recognition. We use the hybrid weighted combination strategy that we applied to LRMS models discussed earlier in this chapter, to combine the models built using the alternative data representations. We then study the performance of such combinations.

Chapter 4

Experiments and Results

In this chapter, we present the results of all the experiments that we conducted with regard to automatic instrument recognition using MedleyDB. Firstly, we investigate the performance of a traditional machine learning method on original dataset in multiple configurations. We define the configuration of the dataset by four parameters : the analysis window size, the hop size during the segmentation of the full length audio and the threshold for binarizing labels along with the merging strategies for combining the results from LRMS datasets. The configuration which gives the best macro f-score is chosen for further experiments. In the next section, we investigate the variation of instrument-wise f-score for different analysis window sizes and hop sizes but with the threshold from the shortlisted configuration. Next, we report the performance numbers - instrument-wise f-score, exact match ratio, macro and micro f-score obtained using deep learning method with the chosen configuration. Further, we discuss the performance numbers that we obtained with regard to the harmonic and residual datasets with the traditional machine learning method as well as deep learning method with the chosen configuration. Further, we combine specific models and evaluate the performance of certain model combinations. Finally, we tabulate the best of our results and compare it with the baseline.

4.1 Experiment 1: Traditional machine learning method on original dataset

Multiple datasets are created from the original audio files of MedleyDB based on the analysis window size, hop size and channel type (LRMS). We randomly played samples from the original full-length audio and found that some instrument sounds lasted for about a second (mostly percussive) and some lasted for around 5 seconds (mostly the melodic instruments on sustain). For this reason, we consider the analysis window sizes of length (in seconds) {1, 2, 3, 4, 5}. We choose the hop sizes from {25%, 50% and 100%}. A 100% hop size means that the windows are non-overlapping. The channels Left (L), Right (R), Mid (M) and Side (S) are considered. From each of these datasets, all the low-level features belonging to the temporal, spectral and cepstral domains are extracted using Essentia. The support vector regressors (SVR) are used to fit the training data in each of these datasets and predict instrument activation confidences for the respective test data. The predictions as well as the ground truth are binarized using different threshold values between 0.3 and 0.7. Firstly, the macro f-score and exact match ratio are estimated for the mid (M) channel datasets for all these threshold values. The mid channel corresponds to the mono format version of the stereo audio. Next, each of the merging strategies - hybrid max and hybrid weighted, are applied separately to combine the labels of the LRMS datasets into a hybrid label set. The performance of the merging strategies is estimated in terms of macro f-score and exact match ratios for all the threshold values. The Fig. 9 represents the macro f-scores for the original dataset in different window configurations, thresholds and merging strategies. The Fig. 10 represents the exact match ratio for the original dataset in different window configurations, thresholds and merging strategies. The analysis window configuration is depicted as the title in each of the subplots in the format {window size}_h{hop%} in Fig. 9 and Fig. 10.

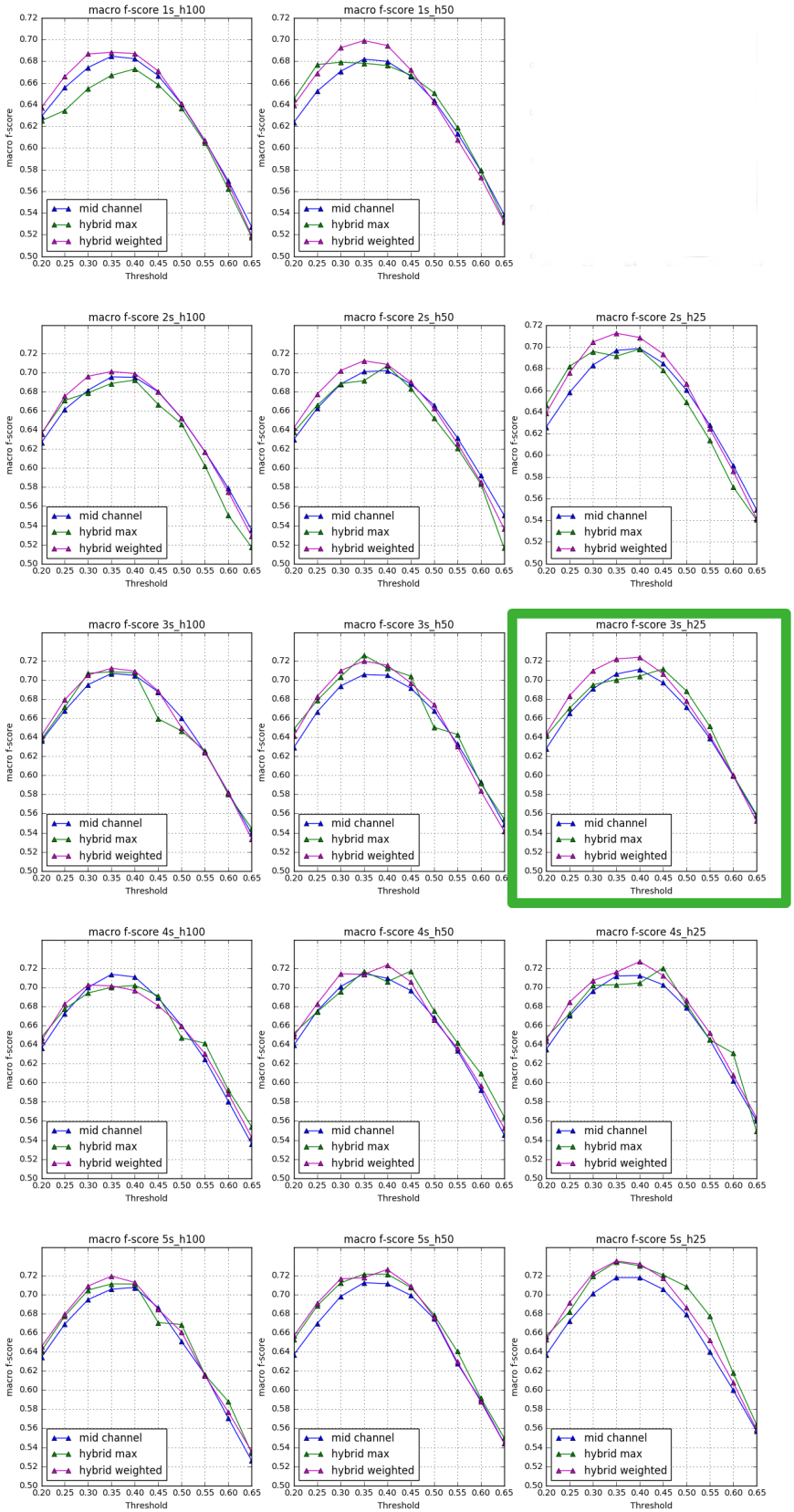

Figure 9: Macro f-scores obtained using traditional machine learning method on test set of the original dataset with different configurations. The shortlisted best configuration - 3s_h25 is highlighted in green.

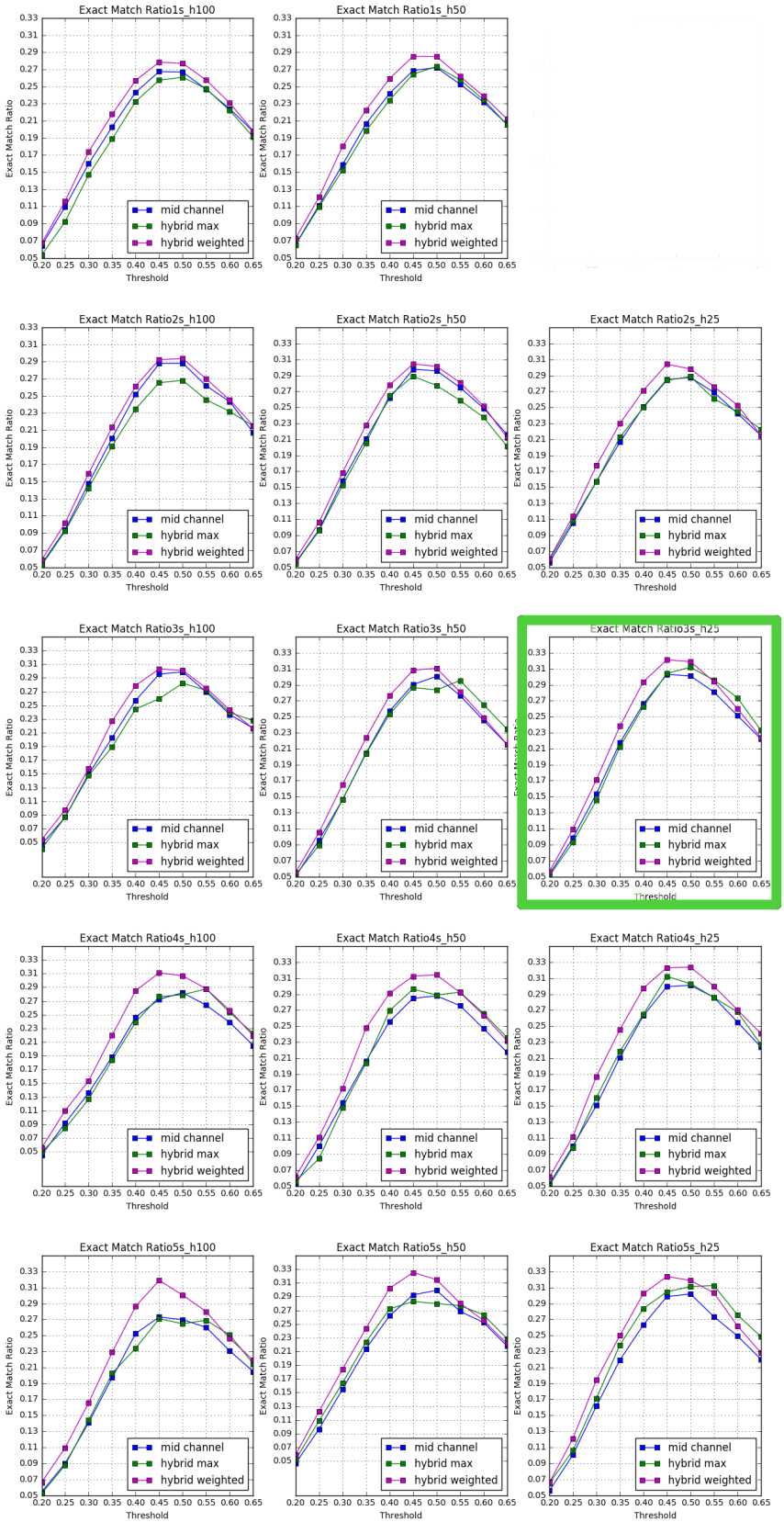

Figure 10: Exact match ratio obtained by traditional machine learning method on test set of the original dataset with different configurations. The shortlisted best configuration - 3s_h25 is highlighted in green.

4.1.1 Effect of analysis window size

From the Fig. 9 and Fig. 10, it can be seen that the macro f-score nd exact match ratio improves with increase in the analysis window size. The window size of length 5 seconds has the highest performance metrics. But, we need to keep in mind that larger analysis window results in smaller the number of samples in dataset. A better performance metric in case of larger windows can also indicate the possibility of over-fitting. The macro f-score numbers are increasing with window size in all the cases. But, Fig. 10 shows that the exact match ratio saturates around 0.32 from the window size of length 3 seconds onward for 25% hop ratio and hybrid weight combination strategy. Hence, we would like to consider 3s as the best window size.

4.1.2 Effect of hop size

The hop size in our thesis is expressed as a percentage of the window size that should always be the difference between the starting position of two consecutive windows. For example, if we have a window size of 6 seconds with hope size of 25%, then the first window will cover samples from 0 to 6 seconds and the next window starts from 1.5 seconds and ends at 7.5 seconds. Throughout all the subplots in Fig. 9 and Fig. 10, it is consistently seen that reducing the hop size improves the respective performance numbers. Lowering the hop size leads to increase in the number of samples. Since the performance numbers are improving with decrease in hop size, despite the increase in number of samples in the dataset, it is a clear indication that the system is learning better rather than over-fitting. This is also intuitively congruent because a lower hop size represents the signal in a greater detail. Therefore, we choose 25% hop size to be the best hop configuration.

4.1.3 Effect of threshold

The ground truth provided by the dataset is activation confidence scores which indicate the probability of an instrument being present in the mix at a particular instant. Our predictions are also continuous variables between 0 and 1. It is important to binarize these scores to interpret if the instrument is present or not. A threshold is

required to binarize these scores for both the ground truth as well as the predicted labels. We investigate the impact of threshold ranging from 0.2 to 0.7 in steps of 0.05 on the macro f-score and exact match ratio. Fig. 9 and Fig. 10 tell us that both macro f-score and exact match ratio curves increase till a certain limit and then decrease. The threshold values maximizing these two metrics are not same, thought they are close (0.4 and 0.45). Fig. 11 shows the impact of threshold on these two metrics for the dataset 3s_h25 with hybrid weighted combination strategy. In our opinion, f-score metric is more important than the exact match ratio for instrument recognition because of the generic nature of f-score and dataset specific nature of exact match ratio. Hence we choose 0.4 as the best threshold since it maximizes the f-score in 3s_h25.

4.1.4 Effect of LRMS merging strategies

We investigate the impact of the two merging strategies : hybrid max combination strategy and hybrid weighted combination strategy, on the performance of traditional machine learning method and compare them with a use case where only the mid channel (which is effectively mono) of the audio is picked and other channel datasets are discarded. These combination strategies are explained in detail in the section 3.5 of the Chapter 3. From the Fig. 9, except for 4s_h100, we can see that hybrid weighted combination strategy results in better performance numbers in all the dataset configurations than the hybrid max strategy and the use case without any combination strategy. We select the hybrid weighted combination strategy as the best combination strategy to incorporate the predictions from LRMS datasets. This confirms that claims made by [Bosch et al., 2012] that panning information can be used to improve instrument recognition performance. This means that left, right, mid and side models contain mutually exclusive important information that can improve instrument recognition performance when considered together. Another important point here is to note that the hybrid max combination strategy is not effective and it even performs worse than no-combination strategy (when only the mid channel data is considered). The hybrid max combination strategy assigns the role of determining the presence of a particular instrument to a particular chan-

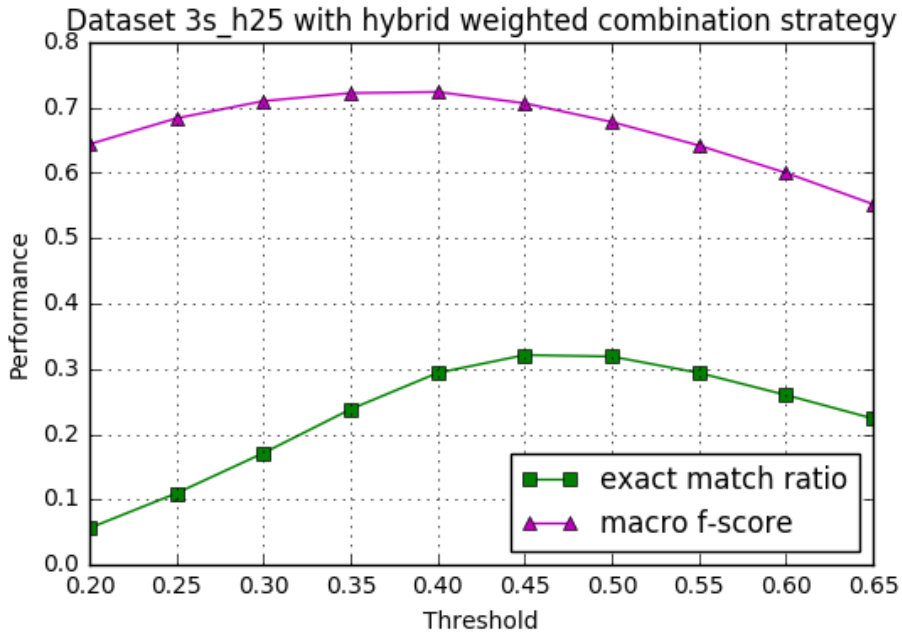

Figure 11: Comparing impact of threshold on exact match ratio and macro f-score for the dataset 3s_h25 with hybrid weight combination strategy.

nel. This can be disadvantageous because a particular instrument may be present in different channels in test data and training data. The hybrid weighted combination strategy considers the predictions from all the channels and performs better than no-combination strategy and hybrid max combination strategy.

4.1.5 Instrument-wise Performance

We compute the f-score for each instrument using traditional machine learning method in different window configurations with 0.4 threshold using hybrid weighted combination strategy with regard to the LRMS datasets. The standard deviation of instrument-wise f-scores over all the window configurations are not significant (standard deviation < 0.1). Hence, we plot their mean values across all the window configurations and display the standard deviations as error bars to indicate how small they are. This can be seen in both Fig. 12. The key observations here are that the sounds which are generated by human activity like singing or hitting or plucking are easier to learn than those which are produced electronically. The fx/processed sound and synthesizers have the lowest performance numbers as compared to the

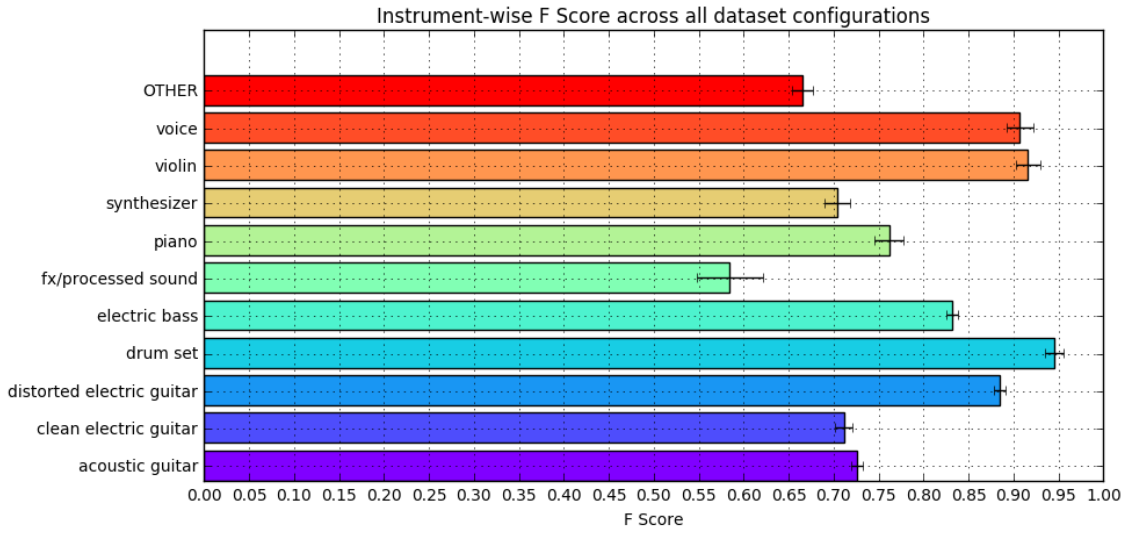

Figure 12: Distribution of instrument-wise f-score across all the window configurations.

rest. The instruments which involve the highest human activity such as drum set, violin and voice have the highest performance numbers (f-score $>$ 0.9).

At the end of experiment 1, we choose the dataset configuration : 3s_h25, threshold of 0.4 with hybrid weighted combination strategy as the configuration that gives best results. This is highlighted using green boxes in Fig. 9 and Fig . 10. Hence, we would like to continue all our further experiments on this dataset configuration. The performance numbers obtained using the traditional machine learning method on the original dataset with this configuration is {exact match ratio= 0.2938, macro f-score= 0.7241, micro f-score= 0.7784}.

4.2 Experiment 2: Deep learning method on original dataset

Unlike in [Li et al., 2015] which uses raw audio as the input, we use the mel-spectrogram as the input to the CNNs. For each of the LRMS datasets, we train our models for at least 20 epochs. We retained the parameters from the best configuration that we determined at the end of Experiment 1. For this configuration, we obtain the performance numbers {exact match ratio= 0.1376, macro f-score=

0.6578, micro f-score= 0.7253}. To compare with the baseline performance, we also get the performance numbers for the window configuration used in [Li et al., 2015] : window of length 1s with 100% hopping (non-overlapping windows). For the window configuration from the baseline paper, we get {exact match ratio= 0.1540, macro f-score= 0.6571, micro f-score= 0.7182}. Surprisingly, there is an increase in exact match ratio despite the decrease in the f-scores when the window configuration is changed from 3s_h25 to 1s_h100. It would be interesting to study the impact of all the configuration parameters on the performance numbers for deep learning method just as we did in Experiment 1. Unfortunately, due to time constraints, we could not include it within the scope of this project. Fig. 13 shows the instrument-wise f-score obtained by the traditional machine learning method and the deep learning method for the best configuration. We notice that the f-scores obtained by deep learning method is lesser than those obtained by traditional machine learning method for all the instruments. A possible reason for the relatively low performance of deep learning method in comparison with the traditional method could be the presence of samples that are almost silent in the former method. The almost silent samples are automatically filtered by the Essentia's music extractor in case of the traditional method. We wanted to redo the experiment with deep learning method after filtering out these almost silent samples, but could not do so due to the time constraints. Such a filtering was not done even in the baseline paper [Li et al., 2015]. Also, we haven't done hyper-parameter tuning for the CNNs. Moreover, we use mel-spectrogram as input instead of raw audio which was used in [Li et al., 2015]. We also need to note that we have not investigated the deep learning method with raw audio as input. Mel-spectrogram samples (each of size 43×128 for 1 second of audio) are compressed representations of raw audio samples (each of size 44100 for 1 second of audio). Since we lose information during compression, using mel-spectrogram as input instead of raw audio could possibly lower the instrument recognition performance. Further, the "best" configuration parameters determined for traditional method need not be the best configuration parameters for deep learning method. We have not investigated the deep learning results for all the combinations of the dataset configuration parameters within the scope of this thesis. Despite all these drawbacks, our deep

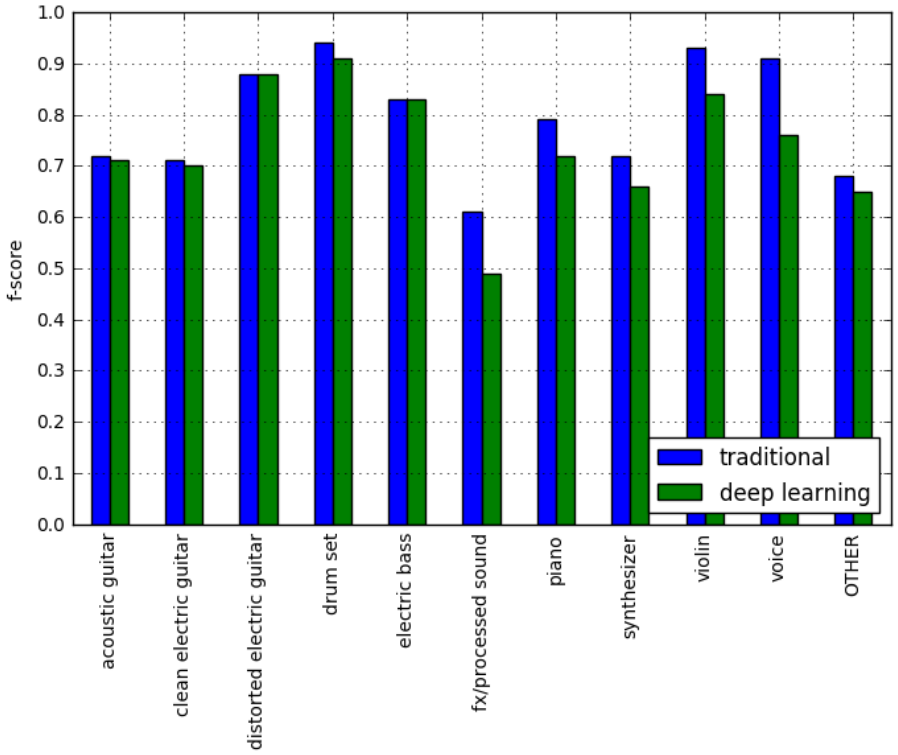

Figure 13: Instrument-wise f-score obtained using deep learning and traditional machine learning methods on original dataset for the best dataset configuration.

learning method performed better than the baseline with regard to the macro and micro f-scores at a lower computational cost.

4.3 Experiment 3: Using the harmonic and residual component datasets

It is interesting to compare the instrument-wise f-scores obtained for harmonic dataset and residual dataset. These f-scores have been determined for both traditional and deep learning methods and plotted in the Fig. 14. For the sake of convenience, we represent the model where traditional machine learning method is used on original dataset as 'trad_original' and likewise we extend this naming convention to other models. Though the trad_original model gives the best over all results, in the Fig. 14, it is interesting to note that this model doesn't perform the best for all the instruments considered one at a time. Acoustic guitar and synthesizer labels are identified best by the deep_residual model. Electric bass is

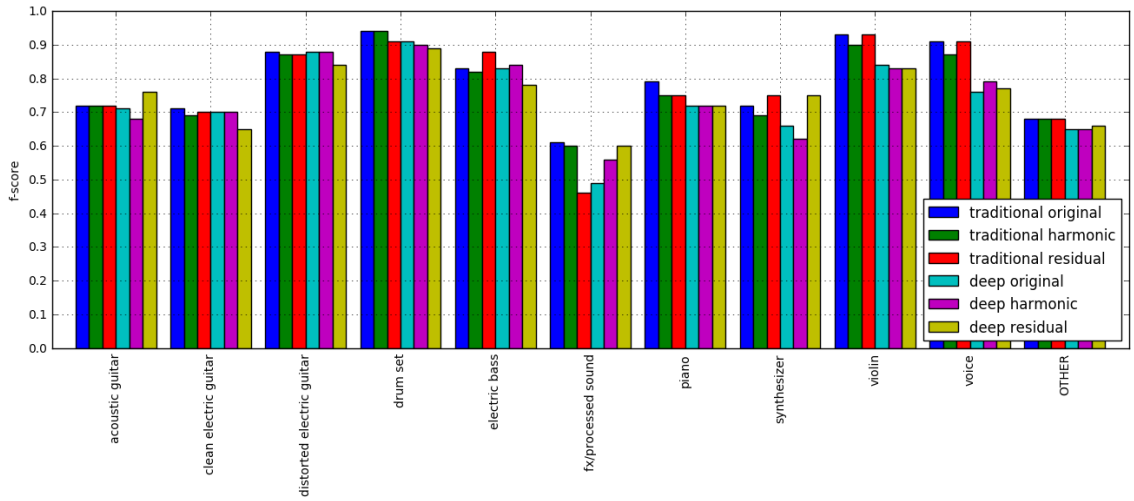

Figure 14: Instrument-wise f-score obtained using deep learning and traditional machine learning methods for the best dataset configuration on respective test sets.

identified best by the trad_residual model. Another important observation here is that the instrument-wise performance does not alter much across all the models, as illustrated in Fig. 15. This indicates that, despite the different ways in which the information is learned by each of the models, the intelligence gained with regard to each instrument is similar across all of them.

For the shortlisted configuration in Experiment 1, we determine the performance numbers for instrument recognition using the models - {trad_original, trad_harmonic, trad_residual, deep_original, deep_harmonic, deep_residual}. This has been tabulated in the Tab. 1. Surprisingly, in our experiments, the deep learning method worked best with the residual component and not the original dataset itself with regard to the f-score. Even, in case of the traditional machine learning method, the residual component dataset gave the best exact match ratio. This could be possibly because the residual component instrument sounds could be the characteristic sound of an instrument. For example, in case of guitar, the plucking has a distinct sound which is not found in a bowed instrument or a percussion instrument. These residual sounds could therefore help in recognizing the instruments better. In the next section, we investigate the performance of original, harmonic and residual models more closely.

So far, our best results are from the trad_original model. Now, we find the number

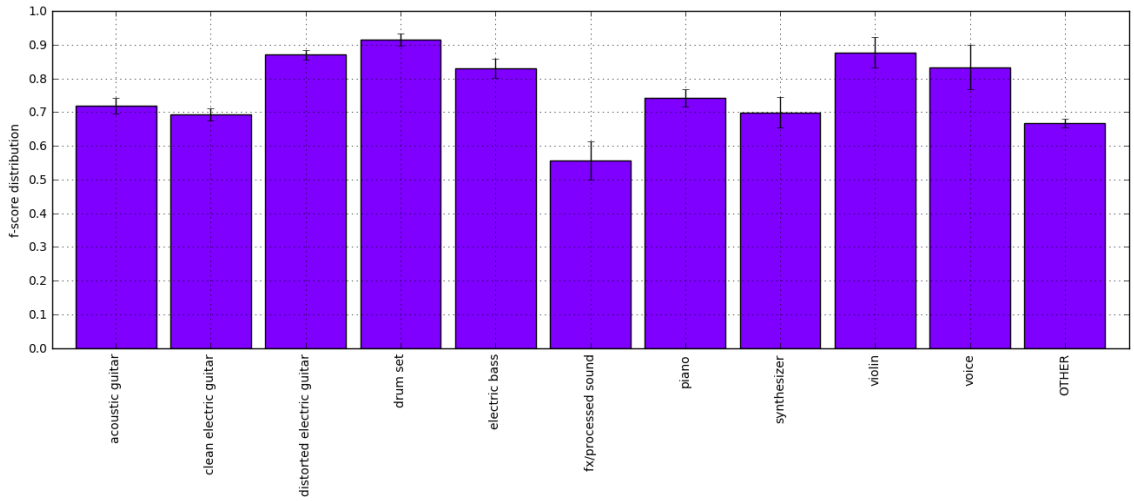

Figure 15: Distribution of instrument-wise f-scores across all the chosen models for the best dataset configuration on respective test sets.

Use Case	Exact Match Ratio	Macro f-score	Micro f-score
trad_original	0.2938	**0.7241**	**0.7784**
trad_harmonic	0.2604	0.7012	0.7590
trad_residual	**0.3163**	0.7008	0.7773
deep_original	0.1377	0.6578	0.7253
deep_harmonic	0.1404	0.6639	0.7224
deep_residual	0.1550	0.6782	0.7353

Table 1: Performance numbers for original, harmonic and residual component datasets for the dataset configuration shortlisted from Experiment 1.

of wrongly predicted samples for each instrument in trad_original that are correctly predicted by trad_residual, trad_harmonic, deep_original, deep_harmonic and deep_harmonic models for the same dataset configuration. These numbers indicate the potential of predictions of all the other models (other than trad_original) in improving the overall performance of trad_original model if appropriately combined. Also, not all the correct predictions of trad_original are rendered true in the other models. An important thing to note here is that we could design a merging strategy that will pick the best capabilities of trad_original predictions while incorporating the best capabilities of predictions obtained in other use cases. The information for merging strategy should come from the performance on training set as we saw in Experiment 1 in case of LRMS merging strategies. The Fig. 16 shows the percentage of wrongly predicted samples in trad_original case from the test set

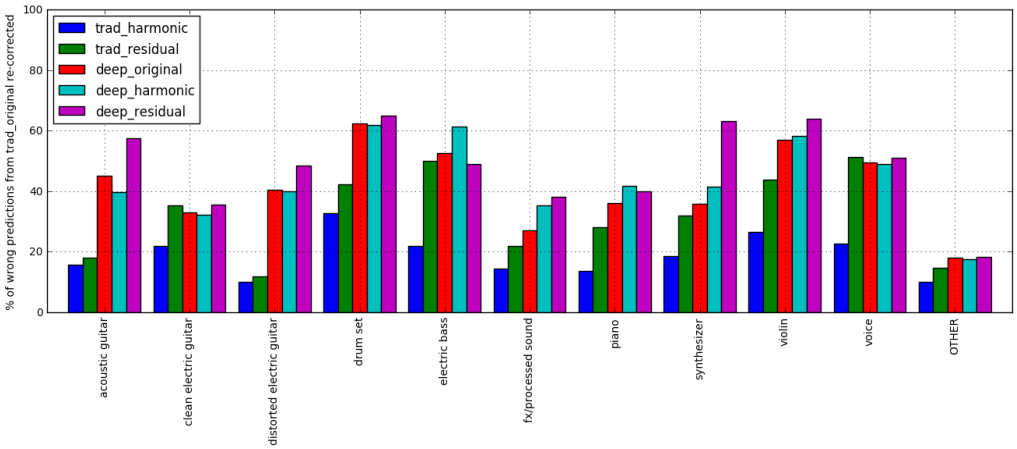

Figure 16: The percentage of wrongly predicted samples in trad_original case that are predicted correctly in other use cases for each instrument.

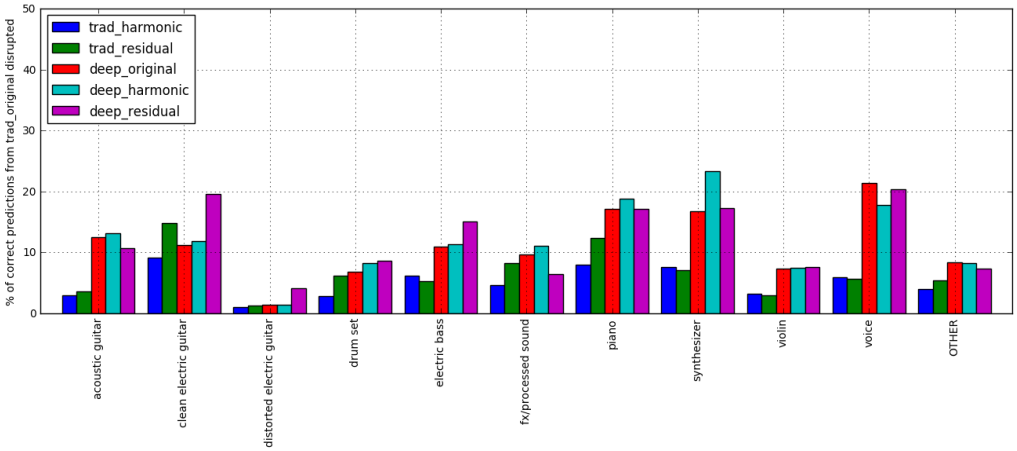

Figure 17: The percentage of correctly predicted samples in trad_original case that are predicted incorrectly in other use cases for each instrument.

that are predicted correctly in other use cases for each instrument. The Fig. 17 shows the percentage of correctly predicted samples in trad_original case from the test set that are predicted incorrectly in other use cases for each instrument. The Fig. 16 and Fig. 17 considers the pairs of trad_original model with all other models taken one at a time. Likewise, plots could be generated for all other possible model pairs. These plots could be used to design a sophisticated strategy which could help us to intelligently select specific models for combining such that the combination maximizes the overall f-score rather than using the brute force approach of trying out all the possible model combinations.

Now, we put together the best of the results that we achieved and compare it with the

Use Case	Window Configuration	Exact Match Ratio	Macro f-score	Micro f-score
trad_original	3s_h25	0.2986	**0.7241**	**0.7784**
trad_harmonic	3s_h25	0.2604	0.7012	0.7590
trad_residual	3s_h25	**0.3163**	0.7008	0.7773
deep_original	3s_h25	0.1376	0.6578	0.7253
deep_harmonic	3s_h25	0.1404	0.6639	0.7224
deep_residual	3s_h25	0.1550	0.6782	0.7353
deep_original	1s_h100	0.1540	0.6571	0.7182
trad_original	1s_h100	0.2572	0.6871	0.7539
baseline	1s_h100	0.2578	0.6433	0.7208

Table 2: Comparison of performance of our methods and baseline.

baseline performance ([Li et al., 2015]) in the Tab. 2. The baseline performance is obtained by using deep learning method with raw audio in mono format as input and a window configuration of 1s_h100. We would like to point out that even with the same window configuration 1s_h100 as in baseline, we achieve a better performance with our traditional machine learning method with regard to the f-scores and almost same exact match ratio. Merely, changing the window configuration to 3s_h25 with the traditional machine learning method results in performance improvement of 15.83% in exact match ratio and 12.56% in macro f-score with respect to the baseline. Our deep learning methods have not performed well with regard to exact match ratio, but they too achieved a better f-score than the baseline f-score in almost all the entries listed in Fig. 2. We need to understand that our deep learning method is not as computationally complex as the baseline method since we do not use raw audio as input. Yet, it performs better than the baseline with regard to the f-scores. It is interesting to see how the deep_residual model outperforms the deep_original and deep_harmonic models with regard to the f-score. Also, the trad_residual model gives the highest exact match ratio of 0.3163 among all our use cases, so far.

4.4 Experiment 4: Combinations of instrument recognition models

In this final experiment, we investigate the instrument recognition performance with different combinations of the instrument recognition models that we built so far. We use the hybrid weighted combination strategy to merge the individual models just as we did in case of LRMS channels. We consider the models from the set {trad, deep} × {original, harmonic, residual} for the shortlisted configuration from Experiment 1. The performance numbers are tabulated in Tab. 3. We retain the best performing model from Tab. 2 and the baseline for comparison. The best performing model is {trad_ori, trad_har, trad_res, deep_res} with regard to f-score and {trad_ori, trad_har, trad_res} with regard to exact match ratio. With respect to the baseline performance numbers, these two models result in an improvement of 14.25% in the f-score and 24.17%, respectively.

It is interesting to note how deep_residual model when combined with the three trad models give the highest f-score ever. We also need to note that trad_original alone performs better than the combination {trad_original, trad_harmonic, trad_residual, deep_original, deep_harmonic, deep_residual}. The reason why some model combinations are more effective in improving the performance than others could be intuitively justified to some extent by studying the plots like Fig. 16. For example, Fig. 16 shows that the deep_residual model predicts the highest number of wrongly predicted samples in trad_original than compared to others. So, we could expect an improvement in performance by combining deep_residual and trad_original models. This method of intuitively guessing a model combination gets really complex when more than two models are considered. For this thesis, we ended up combining the individual models randomly. Since, we can not possibly understand what exactly each of the models are learning, a more sophisticated approach needs to be designed for identifying and combining specific models which is out of scope of our project. We could only suggest that such an approach could be based on an extended set of plots like Fig. 16 and Fig. 17. These plots involved

Model Combination	Exact Match Ratio	Macro f-score	Micro f-score
{trad_ori, trad_har, trad_res, deep_res}	0.2834	**0.7350**	**0.7875**
{trad_ori, trad_har, trad_res}	**0.3201**	0.7282	0.7863
{trad_ori, trad_har, trad_res, deep_ori, deep_har, deep_res}	0.2300	0.7221	0.7833
{trad_ori, trad_res}	0.2682	0.7255	0.7764
{trad_ori, trad_har}	0.2515	0.7224	0.7700
trad_ori	0.2986	0.7241	0.7783
baseline	0.2578	0.6433	0.7208

Table 3: Comparison of performance of our hybrid models and baseline.

the combination pairs of all other models with trad_original. Similarly, more plots could be obtained by considering all other possible pairs of models. Based on these plots, a more sophisticated approach could then be designed to identify the models for combining such that the combination improves the overall f-score. In general, we find that not all the model combinations lead to a performance improvement. However, the fact that some model combinations result in the better performances make this line of research worthwhile.

Chapter 5

Conclusion and Future Work

We investigated the task of instrument recognition in multi-instrument audio contexts with MedleyDB. In doing so, we employed both the traditional machine learning method and the deep learning approach and compared their performances. We experimented with different dataset configurations (analysis window size, hop size and label binarizing threshold) and alternative data representations. We have considered two such alternate data representation sets : 1) LRMS channels; 2)harmonic and residual components of the original audio. After an exhaustive analysis of the results in the previous chapter, we present the conclusion and pointers for future work in this chapter. The following points are the highlights of our thesis:

- Efforts in dataset pre-processing could improve the overall instrument recognition performance.

- One of the difficulties with regard to the data pre-processing stems from the fact that the labels are continuous variables between 0 and 1 which indicate the probability of an instrument's presence. This necessitates an analysis with different threshold values to binarize these continuous variables into 'instrument present' and 'instrument absent' decisions. In congruence with intuition, we found that the best threshold value in case of the traditional machine learning method was 0.4, which is close to 0.5.

- It is helpful to have overlapping segments while splitting the full-length audio into smaller samples in improving the performance. It is intuitively convincing since overlapping segments provide richer contextual information.

- We expected that the instrument-wise f-scores will significantly vary with regard to the window size. However, from the plots in Fig. 12, we notice that the standard deviation for each of the instrument-wise f-scores is low (< 0.1). This shows that the window size does not critically effect the instrument-wise performance.

- The traditional machine learning method learns differently than the deep learning method owing to their fundamentally different ways of operation. Some of the information learned by these two approaches are mutually exclusive and their capabilities could be potentially combined to improve the overall performance.

- Despite the different information that is presented to the learning algorithms in each of the models, the intelligence (f-score) gained with regard to each instrument is similar across all of them.

- The sounds produced by human activity like singing or hitting or plucking are easier to identify than those which are produced electronically.

- Panning information is very helpful in the instrument recognition task. Just by using the LRMS channel information, which is already present in the stereo audio, we show that we can improve the overall performance.

- The machines learn new information from the alternative representations of the dataset that could potentially be used to improve the overall performance. This is already verified in case of the LRMS component datasets. Mutually exclusive information is learned by traditional and deep learning methods in original, harmonic and residual component datasets. If these models are combined appropriately, we could achieve a higher performance.

- We find that certain combinations of instrument recognition models improve the performance. But, we are not able to conclude whether the statement "the perfect model is all models put together" is true or not. We can imagine an infinitude of such models. For example, just like harmonic and residual decomposition, we can formulate an infinitude of decompositions and build different models using each of them. But we will never be able to combine all of them as they are infinite in number. Hence, we are limited by the number of models that we choose to work with. We notice in Experiment 4 of previous chapter that the combination containing all these finite set of models do not give the best performance. In general, not all the model combinations improve the performance. However, we illustrate how a very simple approach of combining specific models can improve the overall performance.

All the highlights mentioned above are a result of a limited set of experiments run on a single parent dataset. Though these findings are intuitively congruent, at this point, we can not generalize these inferences. Further in-depth experimentation with different datasets and perhaps even the combination of other datasets is required to validate them. Across all our experiments, the highest number of samples that we got from MedleyDB was 52,277 for the window configuration 1s_h50. This number is very small as compared to the millions of distinct audio files in the world. In this context, MedleyDB, as a dataset, is too small to make any general deductions out of our experiments.

There is definitely a lot more that can be done from this point in improving the instrument recognition performance. We would like to point out certain aspects related to this research work that we would have loved to try out, but couldn't do so due to the time and resource constraints.

- We used the default configuration provided by Librosa [McFee et al., 2015] to split the audio into harmonic and residual parts. This split function assigns energy to each time-frequency bin based on whether a horizontal (harmonic) or vertical (residual) filter responds higher at that position. A margin parameter

determines the percentage excess of the energy responded by the horizontal filter with respect to the vertical filter to generate the split. The default configuration for the split function is unit margin. It will be interesting to analyze the results obtained for other margin values.

- One of the main advantages of MedleyDB dataset is the availability of the instrument-wise stems. This undoubtedly is a very rich source of information. We have not made use of it in this thesis. However, it could be helpful to train the systems with these stems so that they can learn every instrument sound better and hence improve the overall instrument recognition performance. One could also think of involving these stems as a data augmentation strategy.

- The experiments using deep learning method contained samples which are almost silent. We should have discarded these low energy samples. By the time we realized this, it was too late. In case of the traditional machine learning method, this was automatically handled by the Essentia's music extractor which skipped these low energy samples. This could also explain the low performance of our deep learning method. The deep learning method needs to be tested after filtering out the samples having energy below a particular threshold.

- We chose the 'best dataset configuration' obtained with the traditional machine learning method and used this configuration even for the deep learning method. This configuration need not be the best configuration for the deep learning method since the way these methods learn are fundamentally different from each other. It could be worthwhile to repeat the Experiment 1 in the previous chapter for the deep learning method. Also, hyper-parameters of the CNN need to be tuned to improve the performance of the deep learning method.

- Experiment 4 revealed that the combination of individual instrument recognition models with a hybrid weighted strategy gives the best performance. It will be interesting to investigate all such combinations (even including those

with different window configurations) and then find the combination which gives the best results. It will be worthwhile to design a sophisticated method to identify specific models for combining such that the combination maximizes the overall f-score rather than using the brute force approach of trying out all the possible combinations.

- In our deep learning implementation, we used fixed length filters. [Pons et al., 2017] reported an improvement in performance by using a set of filters with different length. This could also possibly improve our performance numbers and might be worthwhile to try.

List of Figures

List of Tables

Appendix A

Reproducibility

The repository containing the source code for this research work is made available on GitHub. All the results that we reported in Chapter 4 could be reproduced and validated by implementing this code. The code also contains a readme file with all the information that may be helpful for executing the code. The performance numbers in case of the deep learning method, published in this work are obtained by using the weights that are learned after a minimum of 20 epochs [1].

`www.github.com/kvsphantom/instrument-recognition-polyphonic`

[1] epochs are defined as the number of times the learning algorithm runs through the complete training dataset

Bibliography

[Agostini et al., 2003] Agostini, G., Longari, M., and Pollastri, E. (2003). Musical instrument timbres classification with spectral features. *EURASIP Journal on Advances in Signal Processing*, 2003(1):943279.

[Bittner et al., 2014] Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., and Bello, J. P. (2014). Medleydb: A multitrack dataset for annotation-intensive mir research. In *ISMIR*, volume 14, pages 155–160.

[Bogdanov et al., 2013] Bogdanov, D., Wack, N., Gómez Gutiérrez, E., Gulati, S., Herrera Boyer, P., Mayor, O., Roma Trepat, G., Salamon, J., Zapata González, J. R., and Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.* International Society for Music Information Retrieval (ISMIR).

[Bosch et al., 2012] Bosch, J. J., Janer, J., Fuhrmann, F., and Herrera, P. (2012). A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *ISMIR*, pages 559–564.

[Brown, 1999] Brown, J. C. (1999). Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. *The Journal of the Acoustical Society of America*, 105(3):1933–1941.

[Brown et al., 2001] Brown, J. C., Houix, O., and McAdams, S. (2001). Feature dependence in the automatic identification of musical woodwind instruments. *The Journal of the Acoustical Society of America*, 109(3):1064–1072.

[Burred et al., 2009] Burred, J. J., Robel, A., and Sikora, T. (2009). Polyphonic musical instrument recognition based on a dynamic model of the spectral envelope. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 173–176. IEEE.

[Choi et al., 2017] Choi, K., Fazekas, G., Cho, K., and Sandler, M. (2017). A tutorial on deep learning for music information retrieval. *arXiv preprint arXiv:1709.04396*.

[Cont et al., 2007] Cont, A., Dubnov, S., and Wessel, D. (2007). Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints. In *Proceedings of Digital Audio Effects Conference (DAFx)*. Bordeaux.

[Dubnov and Rodet, 1998] Dubnov, S. and Rodet, X. (1998). Timbre recognition with combined stationary and temporal features. In *ICMC*. Citeseer.

[Eronen et al., 2001] Eronen, A. et al. (2001). Automatic musical instrument recognition. *Mémoire de DEA, Tempere University of Technology*, page 178.

[Essid et al., 2006] Essid, S., Richard, G., and David, B. (2006). Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):68–80.

[Fonseca et al., 2017] Fonseca, E., Pons Puig, J., Favory, X., Font Corbera, F., Bogdanov, D., Ferraro, A., Oramas, S., Porter, A., and Serra, X. (2017). Freesound datasets: a platform for the creation of open audio datasets. In *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR).

[Font et al., 2013] Font, F., Roma, G., and Serra, X. (2013). Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 411–412. ACM.

[Fraser and Fujinaga, 1999] Fraser, A. and Fujinaga, I. (1999). Toward real-time recognition of acoustic musical instruments. In *ICMC*. Citeseer.

[Fuhrmann et al., 2012] Fuhrmann, F. et al. (2012). *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, Universitat Pompeu Fabra.

[Fujinaga, 1998] Fujinaga, I. (1998). Machine recognition of timbre using steady-state tone of acoustic musical instruments. In *ICMC*.

[Fujinaga and MacMillan, 2000] Fujinaga, I. and MacMillan, K. (2000). Realtime recognition of orchestral instruments. In *ICMC*.

[Goto et al., 2002] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2002). Rwc music database: Popular, classical and jazz music databases. In *ISMIR*, volume 2, pages 287–288.

[Han et al., 2017] Han, Y., Kim, J., Lee, K., Han, Y., Kim, J., and Lee, K. (2017). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(1):208–221.

[Heittola et al., 2009] Heittola, T., Klapuri, A., and Virtanen, T. (2009). Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *ISMIR*, pages 327–332.

[Herrera-Boyer et al., 2003] Herrera-Boyer, P., Peeters, G., and Dubnov, S. (2003). Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21.

[Kaminsky and Materka, 1995] Kaminsky, I. and Materka, A. (1995). Automatic source identification of monophonic musical instrument sounds. In *Neural Net-*

works, 1995. Proceedings., IEEE International Conference on, volume 1, pages 189–194. IEEE.

[Kaminskyj, 1998] Kaminskyj, I. (1998). Multi-feature musical instrument sound classifier. mikropolyphonie www journal (6)(2001).

[Kostek, 1998] Kostek, B. (1998). Soft computing-based recognition of musical sounds. In *Rough Sets in Knowledge Discovery 2*, pages 193–213. Springer.

[Li et al., 2015] Li, P., Qian, J., and Wang, T. (2015). Automatic instrument recognition in polyphonic music using convolutional neural networks. *arXiv preprint arXiv:1511.05520*.

[Madjarov et al., 2012] Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9):3084–3104.

[Mahapatra, 2018] Mahapatra, S. (2018). Why deep learning over traditional machine learning? link.

[Marques, 1999] Marques, J. (1999). *An automatic annotation system for audio data containing music*. PhD thesis, Massachusetts Institute of Technology.

[Marques and Moreno, 1999] Marques, J. and Moreno, P. J. (1999). A study of musical instrument classification using gaussian mixture models and support vector machines. *Cambridge Research Laboratory Technical Report Series CRL*, 4.

[Martin, 1999] Martin, K. D. (1999). *Sound-source recognition: A theory and computational model*. PhD thesis, Massachusetts Institute of Technology.

[Martin and Kim, 1998] Martin, K. D. and Kim, Y. E. (1998). Musical instrument identification: A pattern-recognition approach. *The Journal of the Acoustical Society of America*, 104(3):1768–1768.

[McFee et al., 2015] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25.

[Opolko and Wapnick, 1989] Opolko, F. and Wapnick, J. (1989). Mcgill university master samples (mums). 11 cd-rom set. *Faculty of Music, McGill University, Montreal, Canada*.

[Pons et al., 2017] Pons, J., Slizovskaia, O., Gong, R., Gómez, E., and Serra, X. (2017). Timbre analysis of music audio signals with convolutional neural networks. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2744–2748. IEEE.

[Raina et al., 2009] Raina, R., Madhavan, A., and Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pages 873–880. ACM.

[Romani Picas et al., 2017] Romani Picas, O., Parra Rodriguez, H., Dabiri, D., and Serra, X. (2017). Good-sounds dataset.

[Romani Picas et al., 2015] Romani Picas, O., Parra Rodriguez, H., Dabiri, D., Tokuda, H., Hariya, W., Oishi, K., and Serra, X. (2015). A real-time system for measuring sound goodness in instrumental sounds. In *Audio Engineering Society Convention 138*. Audio Engineering Society.

[Sechidis et al., 2011] Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer.

[Shenoy Kadandale, 2018] Shenoy Kadandale, V. (2018). Instrument classification in solo-instrument sounds from good-sounds dataset. `https://github.com/kvsphantom/instrument-recognition`.

[Sorower, 2010] Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning.

[UIOWA, 1997] UIOWA, M. (1997). Mis uiowa. `http://theremin.music.uiowa.edu/`.

[Wiggins, 2009] Wiggins, G. A. (2009). Semantic gap?? schemantic schmap!! methodological considerations in the scientific study of music. In *Multimedia, 2009. ISM'09. 11th IEEE International Symposium on*, pages 477–482. IEEE.