Master Thesis in Sound and Music Computing

Universitat Pompeu Fabra

Audio Data Augmentation with respect to Musical Instrument Recognition

Siddharth Bhardwaj

Supervisors: Olga Slizovskaia, Emilia Gómez and Gloria Haro

August 2017

upf. Universitat Pompeu Fabra Barcelona

Master Thesis in Sound and Music Computing
Universitat Pompeu Fabra

Audio Data Augmentation with respect to Musical Instrument Recognition

Siddharth Bhardwaj

Supervisors: Olga Slizovskaia, Emilia Gómez and Gloria Haro

August 2017

Acknowledgement

First, I would like to thank my supervisors Olga Slizovskaia, Emilia Gómez and Gloria Haro for allowing me the opportunity to work with them and letting me be a part of their project. I express my gratitude to them for making the thesis objectives clearer, guiding the thesis research, and providing constant support throughout, all the while being patient with me and making things easier.

I would also like to acknowledge the most insightful and approachable faculty members and researchers here at the Music Technology Group (MTG) who have been part of the learning explosion I have experienced during the last one year.

Finally, I would like to thank my friends and batch mates, specially Manaswi and Kushagra for loads of useful/useless discussions and keeping things fun.

Abstract

Identifying musical instruments in a polyphonic music recording is a difficult yet crucial problem in music information retrieval. It helps in auto-tagging of a musical piece by instrument, consequently enabling searching music databases by instrument. Other useful applications of instrument recognition are source separation, genre recognition, music transcription, and instrument specific equalizations. We review the state of the art methods for the task, including the recent Convolutional Neural Networks based approaches. These deep learning models require large quantities of annotated data, a problem which can be partly solved by synthetic data augmentation. We study different types of audio data transformations that can help in various audio related tasks, publishing an augmentation library in the process. We investigate the effect of using augmented data during the training process of three state of the art CNN based models. We achieved a performance improvement of 2% over the best performing model with almost half the number of trainable model parameters. We attained 6% performance improvement for the single-layer CNN architecture, and 4% for the multi-layer architecture . Also, we study the influence of each type of audio augmentation on each instrument class individually.

Keywords: Automatic Instrument Recognition; Data augmentation; Convolutional Neural Networks

Contents

Chapter 1

Introduction

Humans can identify the instruments present in a musical piece with ease, provided they have a concept of the sound of the instrument(s) being played. But for automatic algorithms to recognize the instruments from a polyphonic audio signal is still a challenging task. The variety of playing styles and timbres of the instruments in the real world makes this task more complex. This task of instrument recognition forms an important research problem for the music information retrieval community as it can be useful to a host of other MIR problems like source separation, auto-tagging by instrument, genre recognition, and music transcription. Research has been going on in this field for almost 20 years now ([Kaminsky and Materka, 1995]). In recent years, the trend is shifting from the earlier 'feature-extraction in conjunction with a simple classifier' based approaches ([Bosch et al., 2012]) to the more modern 'deep learning' based approaches ([Lostanlen and Cella, 2016], [Han et al., 2017]).

Deep learning models need to be trained on large amounts of data to create a more accurate statistical approximation of the real-world problem it is trying to solve. Having more data makes the system more robust to the different variations of the input, provided the data is varied and representative of the problem. Gathering and annotating the data requires a lot of human time and effort. Therefore, one of the most effective ways to tackle the problem of limited data is synthetic data augmentation i.e. making algorithms to manipulate the data in a known way, creating additional data relatively cheaply. Data augmentation has been quite successfully implemented in object and speech recognition ([Krizhevsky et al., 2012], [Jaitly and Hinton, 2013]). Different types of image data augmentation are affine transforms like rotation, shear, flip and shifts, grayscale, blur, and noise. Audio data augmentations are typically pitch scaling, time stretching, noise, and frequency filters but these are in no way an exhaustive list. These transformations help in representing the existing data from additional perspectives. For example, we can model reverberation as a data augmentation technique, which is a common effect in the music production pipeline, to make the data more symbolic of the real-world musical data, similarly for noisy recordings, or recordings in different keys etc. This, additional varied data, in turn helps to prevent the overfitting of the model. In this work, we attempt to study the different types of audio data augmentations (developing an augmentation library in the process) and their effects on the Convolutional Neural Networks architectures ([Pons et al., 2017]) in the context of musical instrument recognition.

1.1 Objectives

The main goals of this thesis are:

- To propose audio data augmentations for the task of instrument recognition from polyphonic music.
- To study the effect of the proposed augmentations on the performance of state of the art Convolutional Neural Networks based systems.
- To publish a flexible, generic, and efficient audio data augmentation library which can be plugged into training or testing with minimum effort.

1.2 Structure of the Report

The remainder of the thesis is organized as follows. In chapter 2, we review the different approaches for instrument classification, their shifting trend towards deep learning approaches, the need and use of data augmentation in other disciplines and then the use of data augmentation in the field of MIR. Chapter 3 details the dataset, evaluation metrics, and methodology decided for the experiments. Chapter 4 includes the evaluation of the instrument classification models with different configurations of data augmentation. It also includes our analysis of the results. In Chapter 5, we summarize the contributions of the work done, the limitations and some ideas for future work.

Chapter 2

Background

Researchers have been working on Instrument Classification since 1990s when the research focused more on monophonic recordings (e.g. [Cemgil and Gürgen, 1997]; [Kaminsky and Materka, 1995]). Since the 2000s ([Martin, 1999] and [Eronen, 2001]), the standard methodology has been to extract several features (mainly MFCCs in combination with other spectral features) and applying different simple classifiers to get the instrument classes. But with the success of deep learning in the field of image recognition and speech recognition ([Richardson et al., 2015] and [Krizhevsky et al., 2012] being a couple of examples), researchers in MIR are also trying to move forward and utilize these technology advances. With this usher into the era of deep learning, there has been an increasing need of huge amounts of labeled data which is crucial for the success of these models. Labeling this data is a resource intensive task and must be labeled by humans who are trained in the domain, which is not always possible and takes a lot of time. An alternative way is to create synthetic data for which we already know the labels for, but this data should be a good representative of the real-world problem we are trying to solve. This is called data augmentation, which is the focus of my research problem. Data augmentation is a common strategy adopted to increase the quantity of training data, avoid overfitting and improve robustness of the models. The aim of this section is to review the approaches for the task of instrument classification, the need and the advantages of data augmentation for building better models for this task.

2.1 Review of instrument classification

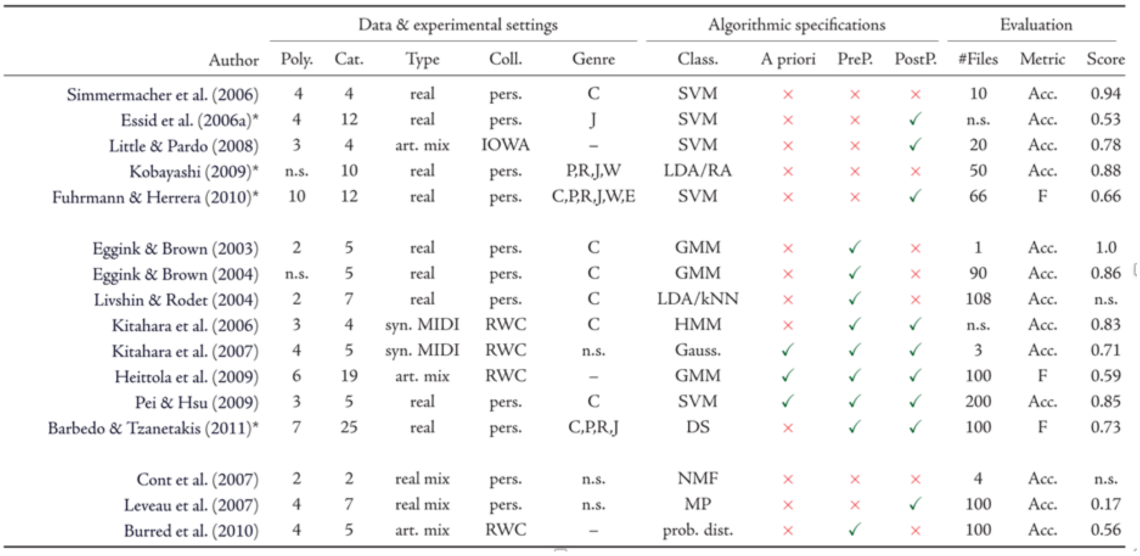
The first efforts in instrument classification were focused on monophonic recordings recorded in lab conditions. Monophonic data helps us to study features that remain invariant for a considered class of instrument/timbre, both for a human mind and for machines, and (perceptually) study these characteristic features for a category. For monophonic audio, [Martin, 1999] and [Eronen, 2001] used MFCCs in combination with different other spectral features, and concluded that MFCC are the best performing features for the recognition task. [Essid et al., 2006] evaluated SVM classifiers together with several feature selection algorithms and methods, plus a set of proposed low-level audio features. Their results emphasized the importance of context in musical instrument recognition. [Joder et al., 2009] studied the early integration of audio features and late integration of classifier decisions. Their conclusions were that early integration helps in removal of features' outlier values, while late integration should roughly capture temporal aspects of the music. Finally, [Yu and Slotine, 2009] used spectrograms of audio files as texture images and applied simple kNN to classify the different instruments. It gave 85.5% accuracy on seven instruments and drums, suggesting it could be an alternate method for source separation task.

[Fuhrmann et al., 2012] divides the approaches for the instrument classification task for polyphonic audio in the following three categories:

- **Pure pattern recognition algorithms**: Systems that try to apply the knowledge directly from monophonic audio related approaches to the more complex input mostly by releasing the constraints on either the data or the categories itself.

- **Enhanced pattern recognition algorithms**: These types of algorithms consider additional knowledge about the source signals in the recognition process. Some authors even introduce some pre-processing in the form of source separation or multi-pitch estimation. Eight studies as shown in the comparative table below lies in this broad category.

- **Template matching algorithms**: These systems derive class memberships by evaluating distances to abstracted representations of the categories. Here, global optimization methods are often applied to avoid erroneous pre-processing resulting from, for instance, source separation.



	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 1: 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 arepure, enhanced pattern recognition, and template matching with respect to the recognition approach. [Table taken from [Fuhrmann et al., 2012]]

In this thesis, the model by [Fuhrmann et al., 2012] is used as the basis for conducting some preliminary data augmentation experiments. The model uses 92 low level features - Local energies, spectral envelope, spectral distribution, pitch based features extracted frame-wise. After extraction of these features, using framesize of 46 ms and hopsize of 24 ms using a Blackman Harris windowing function, L2 normalization and χ^2 feature selection are performed. SVM is then used for classification task. The code used in this thesis for the experiments is the one extended by [Slizovskaia et al., 2016], usage of a simple classifier and the evaluation metrics for this task.

2.2 Advent of deep learning approaches

With higher computation power, there has been a steady rise in deep learning approaches in wide variety of domains like vision, speech and language. Deep learning approaches are getting increasingly more focus in the MIR community due to their high performance and because of the reduction in need for feature engineering, which consumes a lot of time and resources. [Humphrey et al., 2012] critiques the traditional approaches, which seemed to be adopting a two-stage architecture of feature extraction and getting some higher level semantic information from them using classification, regression, clustering, similarity ranking, etc. They argue that this trend has forced the researchers to look towards better model selection for optimizing the results as they have converged towards a small set of features for their respective tasks like MFCC's or chroma. They claim deep signal processing architectures and automatic feature learning is a good replacement to hand-crafted feature design in audio-based MIR. [Pons et al., 2016] points that since deep learning architectures are inherently hierarchical, owing to their depth, they can well represent the hierarchy in music - notes, chords, onset, or rhythms (frequency and time). Deep learning architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Network (CNNs) enable us to model long time dependencies which maybe long-term as in RNNs or local context as in CNNs.

Deep learning has already been successfully applied to many different MIR tasks. Many researchers have hsuccessfully used deep learning for several tasks: onset detection [[Schlüter and Böck, 2014]], genre classification [[Dieleman et al., 2011]], chord estimation [[Lerch, 2015]], auto-tagging [[Dieleman and Schrauwen, 2014]] or source separation [[Huang et al., 2015]].

2.3 Data Augmentation

MIR researchers have always lacked well-annotated training data and this is an even bigger limitation when training deep learning models. In the larger context of deep learning, this problem has partly been solved by creating synthetic data, carefully designing the transformations on the original annotated training data and augmenting it in the training pipeline. The ground truth labels may be preserved in some transformations, or may change in a known way for some other kind of transformations. This idea is known as data augmentation. Synthetic data augmentation is cheaper than manually annotating data both in terms of time and human resource. For many years, dataset augmentation has been a standard regularization technique used to reduce overfitting while training supervised learning models. One of the earliest examples of data augmentations can be found in visual recognition tasks where [LeCun et al., 1998] trained LeNet5, one of the most early and well-known convolutional neural network architectures, and applied a series of transformations to the input images to improve the robustness of the model. [Krizhevsky et al., 2012] was a breakthrough paper which introduced AlexNet, reviving the interest in deep convolutional networks achieving a top 5 test error rate of 15.3% on the test data in ILSVRC-2012 competition which was unarguably the best at the time and still convolutional neural networks are the de-facto standard for image recognition task. AlexNet used translation, horizontal reflection and PCA (altering the intensities of RGB pixel values) as the data augmentation techniques. Nowadays, data augmentation for computer vision tasks is a given while training CNNs as they make the system more robust to the different cameras, angles, lighting, focus etc.. The typical augmentations used for images are affine transforms like rotation, shear, flip and shifts, grayscale, PCA, blur, and additive noise. Motivated by the positive results of augmenting training data in object recognition, [Jaitly and Hinton, 2013] proposed audio augmentations for phoneme recognition task in speech signal processing. They pioneered a data augmentation technique called Vocal Tract Length Perturbation (VTLP), in which the frequency of each utterance by the speaker is mapped to a random new frequency (with some reasonable warp factor), thereby warping the input frequencies. They further use this random warp factors during the test time and finds that averaging over multiple VTLP warp factors for an utterance leads to better results. This was possibly the first example of data augmentation in the audio field. In the same year, [Kanda et al., 2013] also proposed vocal tract length distortion along with speech rate distortion, and frequency-axis random distortion as their choice of data augmentation for speaker recognition. They found the performance results of these augmentations to be nearly additive meaning the individual accuracy improvements summed up equal to the approximate accuracy improvements by combining all three transformations. VTLP Vocal Tract Length Normalization(VTLN) was previously used to normalize the speaker variability in the samples but these two researches established it as a data augmentation techniques for speaker recognition. Other examples of successful implementations of data augmentation are [Cui et al., 2014], [Ragni et al., 2014], and [Ko et al., 2015]. [DeVries and Taylor, 2017] proposes an interesting method for domain independent dataset augmentations. They train a sequence autoencoder to build a learned feature space in which they perform interpolation, extrapolation between samples adding random noise to the samples. Their conclusion is that this type of augmentation can be combined with domain-specific augmentations. They find that extrapolating the samples in feature space is the most useful of the three types of transformations tried.

2.4 Data augmentation in MIR

Encouraged by the positive results of deep learning with data augmentations in other disciplines, researchers began to test the possibilities of these newer technologies in Music Information Retrieval. [Li and Chan, 2011] proposed some musical augmentations for the genre classification task. They use Mel Frequency Cepstral Coefficients (MFCCs) for the task (GMM as the classification model) and found that MFCCs are responsive to key and tempo changes, so these are selected as the augmentations for the experiments. Their hypothesis is that these transformations will help to make the system invariant to key and tempo changes. But this is a problem for some genres as they are more sensitive to these changes. Using the augmented data only in training phase did not give better results, which may be due to the response of the GMM to the augmented or noisy data or recognizing a genre might be more sensitive to pitch and key changes. [Humphrey et al., 2012] proposed a CNN based approach for an established task of chord recognition. They use the time-frequency representation (filter bank applied) as the first layer of the CNN architecture instead of the raw time-domain audio. They use pitch shifting as an augmentation technique but instead of doing it on the audio signal itself, they perform this in the time-frequency representation. Since it also affects the chord labels, they change the labels according to the pitch shift. Though both these researches did not find improvement in the performance results when trained with additional transformed data, they observed that the system is more robust to augmented (degraded) audio files. [Han and Lee, 2016] applied ConvNets to the task of acoustic scene classification and proposed a multiple-width frequency-delta (MWFD) data augmentation method that used static mel-spectrogram and frequency-delta features as individual input examples and achieved a significant accuracy improvement. [Schlüter and Grill, 2015] investigated a variety of data augmentation techniques for application to singing voice detection. They use two data-independent augmentations: dropout i.e. setting zero values for a percentage of input points and additive Gaussian noise with a given standard deviation. They apply these to the mel spectrograms directly which is then fed into the CNN. They also performed pitch shifting and time stretching in the spectrogram phase, scaling the spectrogram vertically for pitch and horizontally for time. They also include increasing the loudness level, and mixing two excerpts together. As in speech recognition, they found pitch shift combined with time stretching and random frequency filtering to be the most useful augmentation methods. Individually, pitch shift was found to be the most successful method. Loudness alteration, Gaussian noise and mixing excerpts did not give good results, but they might be helpful in making the system more robust to these characteristics in the music. There has been one recent attempt at making a musical data augmentation (MUDA) framework by [Mcfee et al., 2015]. They present a framework to perform annotation aware data augmentations on annotated musical data, changing data and labels accordingly.

Their framework requires the song and its annotations, metadata to be in JAMS format ([Humphrey et al., 2014]). Getting the data in JAMS format might need some additional pre-processing of the data. This framework is based on Audio Degradation Toolbox ([Mauch and Ewert, 2013]), and has been implemented in Python. It currently supports the following augmentations - pitch shift, time stretch, background noise, and dynamic range compression, which provides a good starting point but is not an exhaustive list. [Salamon and Bello, 2017] used MUDA for data augmentation in the context of Environment Sound classification and they found that each sound class is influenced differently by each augmentation set, which emphasizes that the augmentations should be done keeping in mind the perceptual changes in the data. Upon reviewing all these sources, I feel the researchers are rapidly moving towards more deep learning techniques which removes the need for hand-picked features for giving as input to the model, which inevitably loses some information in the process. Deep learning architectures just need labeled data to work on and directly learn the weights and some high level understandable/non-understandable features needed to perform the task. Gathering enough labeled data can be a bottleneck for performance sometimes, so researchers started creating synthetic data which started with the field of vision where the image was transformed in many ways like rotate, scale, shift etc. so that the model is robust to these changes in data without the change in label (or in some cases, a known change) and consequently, overfitting of the model is prevented. There have been only a handful number of experiments which focus on the study different types of audio augmentations and their effects on timbre and other properties. There has been one attempt at making an audio data augmentation framework which does not provide much flexibility and they also miss some of the useful augmentations. This leaves an opportunity to develop another library that can be plugged with any kind of models, does augmentation processing in an efficient way, and provides most of the useful audio data transformations. Also, there is a need to do an organized study studying these different types of augmentations for the task of instrument classification.

Chapter 3

Methodology

In this chapter, we discuss the methodology applied in this work. We start with the overview of the major datasets for instrument classification, proceeding to the main evaluation metrics for the task, and then, detailing our augmentation library and the methodology selected for our research.

3.1 Dataset

For creating human annotated datasets, the taxonomy of the instruments needs to be decided first. A version of the instrument taxonomy, taken from Hornbostel and Sachs (1961) is shown in figure 2. The datasets can have the instruments as the annotated categories and the classification results can then be reported hierarchically to see whether the confusion between the instruments classes is also propagating between their parent classes like woodwind vs brass, or the confusion more internal to a broader class which is more understandable.

The dataset used for the experiments is IRMAS dataset. It is a dataset from the Music Technology Group(MTG), Barcelona and includes musical audio excerpts with annotations of the predominant instrument(s) present. This dataset is divided into training: 6705 audio files, and testing: 2874 audio files in 16-bit stereo wav format sampled at 44.1kHz. Each file is a 3 second excerpt taken from more than

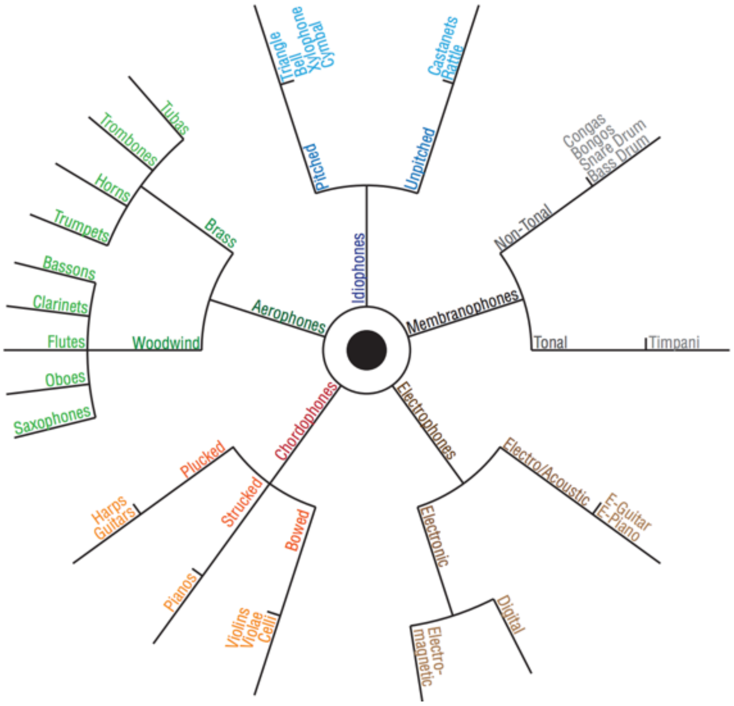


Figure 2: The enhanced scheme of taxonomy as given by Hornbostel and Sachs (1961)

2000 audio recordings under 11 instrument categories. All the categories are pitched instruments. Overview of the main datasets for the task of instrument classification have been included in table 1.

3.2 Evaluation metrics

The evaluation method is based on comparing the output labels against the manually annotated ground-truth labels. The algorithms can be evaluated using precision,recall and F1-measure.

$$\textbf{Precision} \quad \frac{\text{True Positives}}{\text{Predicted Positives}} \quad \textit{or} \quad \frac{\text{TP}}{\text{TP}+\text{FP}} \text{ (\% of selected items that are correct)}$$

$$\textbf{Recall} \quad \frac{\text{True Positives}}{\text{Actual Positives}} \quad \textit{or} \quad \frac{\text{TP}}{\text{TP}+\text{FN}} \text{ (\% of correct items that are selected)}$$

where,

Table 1: Datasets

Dataset	**Description of Data**	**Limitations/Comments**
Dataset for Instrument Recognition in Musical Audio Signals(IRMAS)	Training: 6705 audio files, and Testing: 2874 audio files in 16 bit stereo format sampled at 44.1kHz. 11 pitched instrument categories	Polyphonic audio with predominant instrument labeled. Limited number of instrument categories.
University of Iowa Musical Instrument Sampled(UIOWA MIS)	2182 samples of 20 instruments	Monophonic audio files
MedleyDB	122 multitrack recordings(mix + processed stems + raw audio for music pieces and excerpts) annotated by instrument. Contains instrument activations in sections of songs.	Limited data - 52 instrumental tracks and 70 containing vocals.
Real World Computing (RWC)	3544 audio excerpts labeled in 50 pitched and percussion instruments, including human voice	The frequency distribution of the instrument categories is not very uniform and a few categories is not very uniform and a few categories have even less than 20 samples
Good sounds	12 instrument categories. For all the instruments the whole set of playable semitones in the instrument is recorded several times with different tonal characteristics.	Developed mainly for checking quality of sounds.

TP: number of samples correctly classified in an instrument category (e.g. violin as violin)

FP: number of samples wrongly classified in that instrument category (e.g. cello as violin)

FN: number of samples of the instrument category which are predicted as some other category (e.g. violin classified as any other instrument)

F - Measure: It is a measure of a test's accuracy. F1 measure is defined as the harmonic mean of precision and recall.

$$\mathrm{F1} \quad = \quad 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

3.3 Data augmentation library

One of the main contributions of this thesis is the audio data augmentation library, which is written in python. There was a clear need for a general-purpose audio data augmentation library which gives an option for large number of transformations to be applied on the audio. These augmentations may be label-preserving (as in this case) for a certain range of parameters, or may change the labels in a known way (for instance, pitch scaling for a chord recognition task). The applications of this library include testing the system for robustness on degraded test inputs. Another application is to create augmented data to include in the training pipeline of a system. The transformations currently supported by the library are explained in the following subsections. The library written is based on the work by [Mauch and Ewert, 2013] who created an Audio Degradation Toolbox (ADT) in MATLAB. It also takes inspiration from Musical Data Augmentation framework ([Mcfee et al., 2015]) and an Audio Degrader Library ([Molina, 2016]), both of which are themselves based on the ADT. The library currently takes audio file (16 bit .wav) as the input. All the sounds used for augmentation are taken from the ADT toolbox and manipulated further as per the requirement. The code has been published at `https://github.com/sid0710/audio_data_augmentation`.

3.3.1 Random Cropping

This transformation takes the input audio and outputs a randomly cropped sample of the same, where the user should specify the minimum duration of the output. This is done using the slice function from the Essentia ([Bogdanov et al., 2013]) library.

3.3.2 Background noise

This perturbation adds noise to the input audio signal at a required signal to noise ratio (SNR). Currently, only white-noise is present but the code can easily be modified to add more types of noise. SNR is the ratio of root mean square(RMS) amplitude of the signal to RMS amplitude of noise, squared, which formed the basis of the implementation.

3.3.3 Convolution (smartphone and classroom microphone)

This augmentation convolves the input audio signal with an impulse response recorded from a smartphone microphone (Google Nexus One) and another from a classroom and the user can specify the level of convolution. The smartphone impulse response adds a kind of distortion to the sound while the classroom impulse response creates a reverberation. This can be easily extended to more types of transformations by adding the desired(may be different for each problem) impulse responses.

3.3.4 Simple gain

This augmentation increases the loudness of an input signal by a specified gain (in dB). It may be helpful for data, where the object of interest (bass, rhythm etc.) are not loud enough. It is implemented by reading the audio file and carrying out simple array manipulations.

3.3.5 Pitch scaling

This transformation changes the pitch of the input audio without changing the duration. This has been implemented using Rubberband ([rub, 2012]), which is a command line utility for pitch-shifting and time-stretching audio. The user can give the ratio by which the pitch needs to be scaled. The code can be restricted to scale by only semitone(s) differences, but in its current version, it scales to a ratio between -9.99 to $+9.99$.

3.3.6 Time stretching

This augmentation changes the time duration of the input audio signal. Depending upon the requirement, it can stretch or compress without changing the pitch of the audio. This is also implemented using Rubberband ([rub, 2012]), and ratios to be inputted can vary between -9.99 to +9.99.

3.3.7 Dynamic range compression

This applies a signal dependent normalization to the input audio signal, reducing the energy difference between soft and loud parts of the signal. It has been implemented using the compand function of sox ([sox, 2013]). Most of the produced music today have already been through dynamic range compression but this is not necessarily the case for research datasets, hence, it can be a useful augmentation to apply on the data before predicting some label for it (on the test data) as a kind of a normalization. Three degrees of dynamic range compression can be stated by the user.

3.3.8 Equalization

It applies an equalization to the input audio signal given a center frequency, bandwidth and gain. With this transformation, the signal-level at and around a selected frequency can be increased or decreased, whilst (unlike band-pass and band-reject filters) that at all other frequencies is unchanged. This is implemented using the equalizer function of sox ([sox, 2013]).

3.4 Preliminary experiments

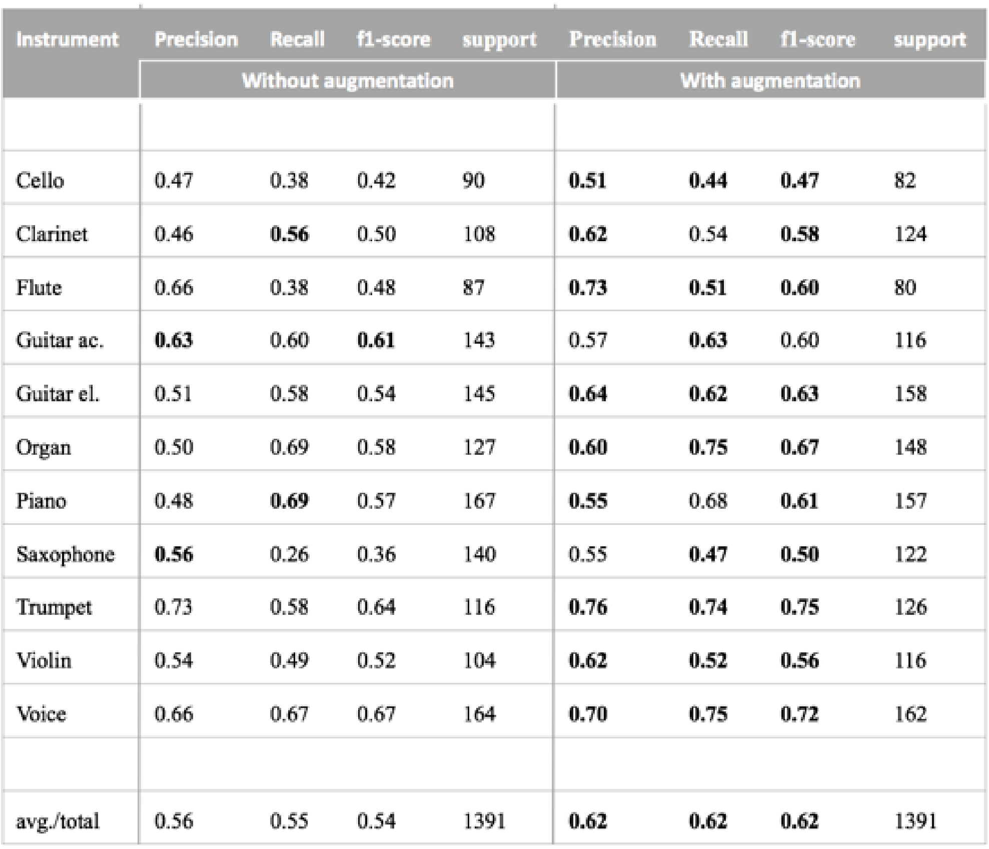
For the purposes of testing/debugging of the data augmentation library and to study the conventional methods of instrument classification, the method proposed by [Fuhrmann et al., 2012] is selected and the code used is as extended by [Slizovskaia et al., 2016]. In this experiment, an audio file is taken as input, which are then split with a fixed framesize of 46 ms and hopsize of 24 ms using a Blackman-Harris windowing function, a large number of spectral, cepstral and tonal descriptors from the audio are extracted and statistical measures like mean, variance and standard deviation are calculated. These are then used as features for the classifier. Then, normalization of the features is done. In the paper, feature selection is performed using χ^2(chi square) but we use mutual information instead. Support Vector Machine (SVM) is used as the classifier for this task. For evaluation, the dataset (IRMAS) is divided into 10 subsets for stratified 10-fold cross-validation. Multidimensional grid search is performed (GridSearchCV from scikit-learn) to tune the parameters, the predictions are made and evaluated for each subset, then for overall accuracy of the model, the accuracies are averaged across all partitions. The model also reports the best parameters found for the current system. Reported in table 2 are precision, recall, f-score and support for each instrument category, in two configurations: without augmentation and with augmentation. Also, the confusion matrix is made for the eleven instrument labels - 'cello', 'clarinet', 'flute', 'guitar (acoustic)', 'guitar (electric)', 'organ', 'piano', 'saxophone', 'trumpet', 'violin', 'voice' as in the IRMAS dataset on which the system is trained and evaluated.

The following six type of augmentations are used for the experiments

- Adding Background noise (SNR value: between 10 and 20)
- Convolving with the impulse response of smartphone mic (level: between 0.1 and 0.5)
- Convolving with the impulse response of classroom mic (level: between 0.1 and 0.5)
- Pitch scaling (Ratio of pitch scaling: between 0.7 and 1.3)
- Time stretching (Ratio of time stretching: between 0.7 and 1.3)
- Dynamic range compression (level: 1,2 or 3)

Random value generator is used between the above specified values for each type of transformation applied. These parameter values have been arrived at by first creating a large set of files for a range of values, then listening to the created files. Under these limits, the class label does not change i.e. the perception of the predominant instrument remains the same. There were two main challenges during augmentation, one was the time it took to create the 33,361 augmented audio files, and the other was the space that these files would take. The space constraint is handled in the augmentations script by creating the augmented files, extracting their features, and deleting them once their job is done. Feature extraction script is modified to append all the features from the augmented files and extract their ground truth labels properly. The different transformations can be parallelized(using multi-threading) while applying to the data making the library more time-efficient.

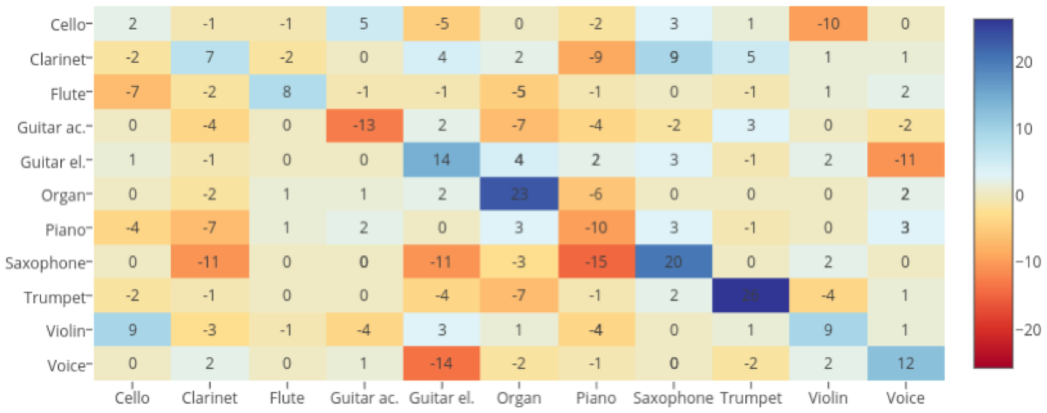
Table 2: Evaluation of the svm model without and with augmentation



Instrument	Precision	Recall	f1-score	support	Precision	Recall	f1-score	support
	Without augmentation				With augmentation			
Cello	0.47	0.38	0.42	90	**0.51**	**0.44**	**0.47**	82
Clarinet	0.46	**0.56**	0.50	108	**0.62**	0.54	**0.58**	124
Flute	0.66	0.38	0.48	87	**0.73**	**0.51**	**0.60**	80
Guitar ac.	**0.63**	0.60	**0.61**	143	0.57	**0.63**	0.60	116
Guitar el.	0.51	0.58	0.54	145	**0.64**	**0.62**	**0.63**	158
Organ	0.50	0.69	0.58	127	**0.60**	**0.75**	**0.67**	148
Piano	0.48	**0.69**	0.57	167	**0.55**	0.68	**0.61**	157
Saxophone	**0.56**	0.26	0.36	140	0.55	**0.47**	**0.50**	122
Trumpet	0.73	0.58	0.64	116	**0.76**	**0.74**	**0.75**	126
Violin	0.54	0.49	0.52	104	**0.62**	**0.52**	**0.56**	116
Voice	0.66	0.67	0.67	164	**0.70**	**0.75**	**0.72**	162
avg./total	0.56	0.55	0.54	1391	**0.62**	**0.62**	**0.62**	1391

For setting up the experiment with data augmentation, the training data is divided into training and test folds during the feature preprocessing. The augmented files are created for the training data, their features are extracted in the same way and stored along with the original feature set of samples. This combined set of features (original training data + augmented data) is used for training the SVM. The performance of the model improves significantly by using additional augmented data during the training process as seen in table 2.

The average precision went from 0.56 to 0.62, average recall from 0.55 to 0.62 and f1-score from 0.54 to 0.62 which is quite a significant improvement. The difference in confusion



	Cello	Clarinet	Flute	Guitar ac.	Guitar el.	Organ	Piano	Saxophone	Trumpet	Violin	Voice
Cello	2	-1	-1	5	-5	0	-2	3	1	-10	0
Clarinet	-2	7	-2	0	4	2	-9	9	5	1	1
Flute	-7	-2	8	-1	-1	-5	-1	0	-1	1	2
Guitar ac.	0	-4	0	-13	2	-7	-4	-2	3	0	-2
Guitar el.	1	-1	0	0	14	4	2	3	-1	2	-11
Organ	0	-2	1	1	2	23	-6	0	0	0	2
Piano	-4	-7	1	2	0	3	-10	3	-1	0	3
Saxophone	0	-11	0	0	-11	-3	-15	20	0	2	0
Trumpet	-2	-1	0	0	-4	-7	-1	2	26	-4	1
Violin	9	-3	-1	-4	3	1	-4	0	1	9	1
Voice	0	2	0	1	-14	-2	-1	0	-2	2	12

Figure 3: Difference of confusion matrices with and without augmentation, positive values(blue) signifies a positive change in confusion and negative values(red) signifies a negative change in confusion

matrices with and without augmentation is shown in figure 3. We can see that for most of the classes, the confusion with themselves (the diagonal values) have increased indicating an improvement in model performance. 'Acoustic guitar' and 'piano' are the only two worse performing instrument classes. Also, observed from the results, is a confusion between cello and violin (bow instruments), clarinet, saxophone, trumpet, and flute (wind instruments),electric guitar and voice, organ and voice, piano with a lot of other instruments, violin with cello. These classes can be mislabeled even by humans sometimes, so these mistakes are understandable.

3.5 Experiment set-up

This thesis builds upon the work on designing musically motivated Convolutional Neural Networks(CNNs) by [Pons et al., 2017]. They discuss about the different strategies for designing the CNN architecture incorporating the domain knowledge. They propose a design strategy for efficiently learning timbre representations using different musically motivated filter shapes in the first layer of the CNN. Fixed-length log-mel based spectrograms are given as the input to the first layer. They design different CNN architecture based on their proposed strategy for three timbre related tasks: singing voice phoneme classification, musical instrument recognition and music auto-tagging. We are specifically interested in the task of musical instrument recognition and will study that in a little more detail.

For instrument classification task, they use the IRMAS dataset, which have 11 instrument categories as mentioned in section 3.1. Two state of the art approaches are used as baselines, one deep CNN based ([Han et al., 2017]) and another feature extraction + SVM classifier based approach. Then, they propose the following two architectures

Single-layer: It has single wide convolutional layer. The input is a log-mel spectrogram of size 96 × 128. 128 filters of sizes 5 × 1 and 80 × 1, 64 filters of sizes 5 × 3 and 80 × 3, and 32 filters of sizes 5 × 5 and 80 × 5 are used in accordance with the discussion that different filter shapes should be used in the first layer to capture different characteristics of timbre. Max pooling is done in the vertical direction (or the frequency axis) to learn pitch invariant representations. A softmax output layer is used with a 50% dropout to finally predict one of the 11 instrument classes.

Multi-layer: Its first layer is similar to single-layer but it is deepened by two convolutional layers of 128 filters of size 3 × 3, one fully-connected layer of size 256 and a softmax output layer with 11 output classes. 25% dropout is used for all convolutional layers and 50% for dense layers. There is a max pooling layer after every convolutional layer. Each convolutional layer is also followed by batch normalization.

These models were set up (code provided in [Pons et al., 2017]) for evaluation without and with data augmentation. Each network is trained optimizing the cross-entropy with Standard Gradient Descent (SGD) from random initialization. The best model in the validation set is kept for testing. For the data augmentation setup, first, the training data of 6705 files was augmented using the below mentioned 12 transformations.

- Adding Background noise (SNR value: between 10 and 20)
- Convolving with the impulse response of smartphone mic (level: between 0.1 and 0.5)
- Convolving with the impulse response of classroom mic (level: between 0.1 and 0.5)
- Pitch scaling (Ratios of pitch scaling: 0.7, 1.3 and 1.5)
- Time stretching (Ratios of time stretching: 0.7, 1.3 and 1.5)
- Dynamic range compression (level: 1,2 and 3)

This created quite a large dataset of original training + the augmented training data (6705×12 data samples). Then, the log-mel spectrograms were computed for the required model and fed to the training pipeline. These experiments are done for single-layer and the multi-layer architecture models. The results for these experiments are analyzed in the next section.

To further study how each type of transformation affects each instrument class, we carried out a series of experiments using the single-layer architecture. We used only one type of audio transformation for each experiment, making it five experiments (noise, convolution, pitch scaling, time-stretching, and dynamic range compression). The results for these experiments have been elaborated upon in the next section. For this set-up, we used similar parameter values for all the augmentations except the pitch scaling, where we scaled the audio pieces from -4 to +4 semitone difference, where in the earlier experiments, we used simple ratio based scaling rather than the semitone difference.

Chapter 4

Evaluation

In this chapter, we present our evaluation strategy and the results for the experiments conducted. The preliminary experiments on the feature extraction + SVM model gave very encouraging results, enabling us to test the data augmentation on the spectrogram + CNN based classification.

4.1 Evaluation Strategy

The performance of the different models is evaluated using standard metrics like micro and macro precision,recall and F1-measure which are explained in section 3.2. The micro-metrics are calculated globally by counting the total true positives, false negatives and false positives while the macro-metrics are computed label-wise and their unweighted mean is reported without taking label imbalance into account.

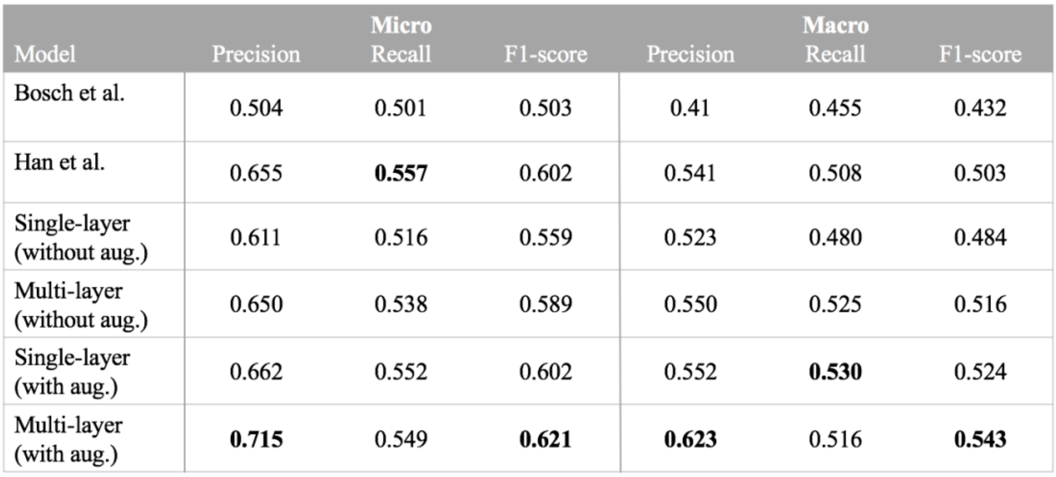
There is an option for two evaluation strategies s1 and s2. The s1 strategy computes a mean activation through whole audio excerpt and apply identification threshold. The s2 strategy computes sum of activations, normalize it by dividing by maximum activation. The weights file with the best results from validation set is used for the evaluation of the test set, same as the original work. For this thesis, we use the strategy s1, unless otherwise stated.

4.2 Results

The experiments were conducted for both single-layer and multi-layer models, in accordance with the experiment set-up detailed in section 3.5. The table 3 shows the comparison of the performance for single-layer and multi-layer set-up without and with augmentation, alongside the other baseline approaches.

As is clear from the results, data augmentation improves the performance of the model significantly. The results for single-layer (with augmentation) were obtained with the evaluation strategy s1 and a threshold of 0.2, same as for the other three CNN architectures. Single-layer architecture (with augmentation) not only improves upon itself, but also outperforms other deeper architectures with just 79k parameters as opposed to 743k parameters for multi-layer architecture and 1446k parameters for the [Han et al., 2017] architecture. Multi-layer with augmentation is found to be the best performing architecture with an overall classification accuracy improvement of around 2% over the previous best performance.

Table 3: Performance results in comparison with other architectures



		Micro			**Macro**	
Model	Precision	Recall	F1-score	Precision	Recall	F1-score
Bosch et al.	0.504	0.501	0.503	0.41	0.455	0.432
Han et al.	0.655	**0.557**	0.602	0.541	0.508	0.503
Single-layer (without aug.)	0.611	0.516	0.559	0.523	0.480	0.484
Multi-layer (without aug.)	0.650	0.538	0.589	0.550	0.525	0.516
Single-layer (with aug.)	0.662	0.552	0.602	0.552	**0.530**	0.524
Multi-layer (with aug.)	**0.715**	0.549	**0.621**	**0.623**	0.516	**0.543**

The results in table 3 clearly demonstrates the advantage of using synthetic data augmentation during the training process of a CNN for the task of instrument classification. Further, we visualized the results as a heat map of the difference of confusion matrices with and without augmentation. This gives us a clearer idea about the changes in predictions before and after augmentation. The plot for the single-layer architecture is shown in figure 4. From the plot, it is evident that most of the diagonal values i.e. the confusion of the correct classes with themselves, are positive indicating better classification accuracy for those classes. A few classes like 'cello' and 'electric guitar', and 'flute' performed worse than previously.

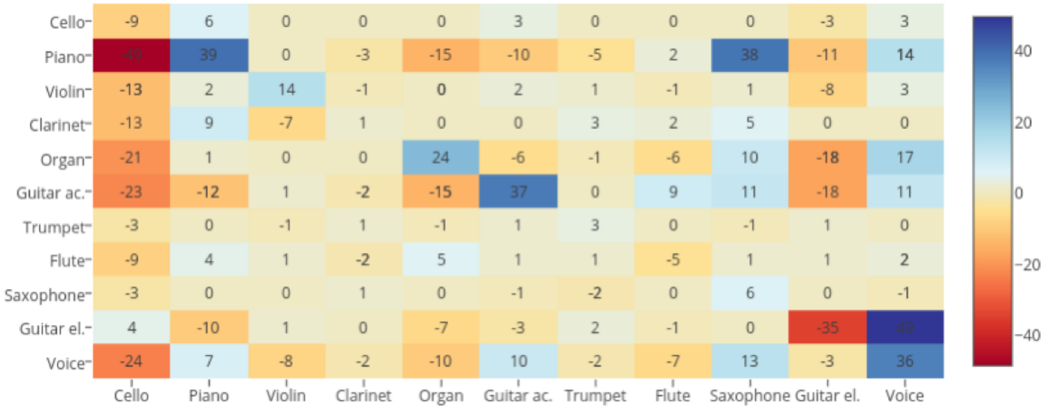


Figure 4: Difference of confusion matrices with and without augmentation for the single-layer architecture, positive values(blue) signifies a positive change in confusion and negative values(red) signifies a negative change in confusion

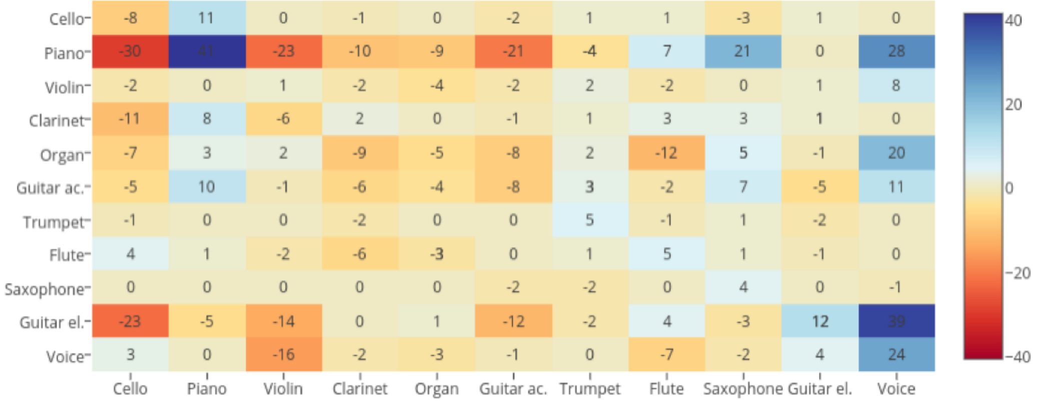


Figure 5: Difference of confusion matrices with and without augmentation for the multi-layer architecture, positive values(blue) signifies a positive change in confusion and negative values(red) signifies a negative change in confusion

The plot for the multi-layer architecture is shown in figure 5. With data augmentation, the classes 'piano', 'electric guitar', and 'voice' performed extremely well; most other classes performed reasonably well. The classes 'cello', 'organ', and 'acoustic guitar' performed worse than previously.

From figure 4 and 5, it is seen that the sound samples from all the classes are being less confused with 'Cello' and more with 'Saxophone' and 'Voice'. The confusion with the 'voice' class could be due to the random pitch scaling rather than shifting it in semitones which is a more musical way of doing the augmentation. Therefore, while doing the experiment with only pitch shifting, we chose to use the semitone based pitch shifting.

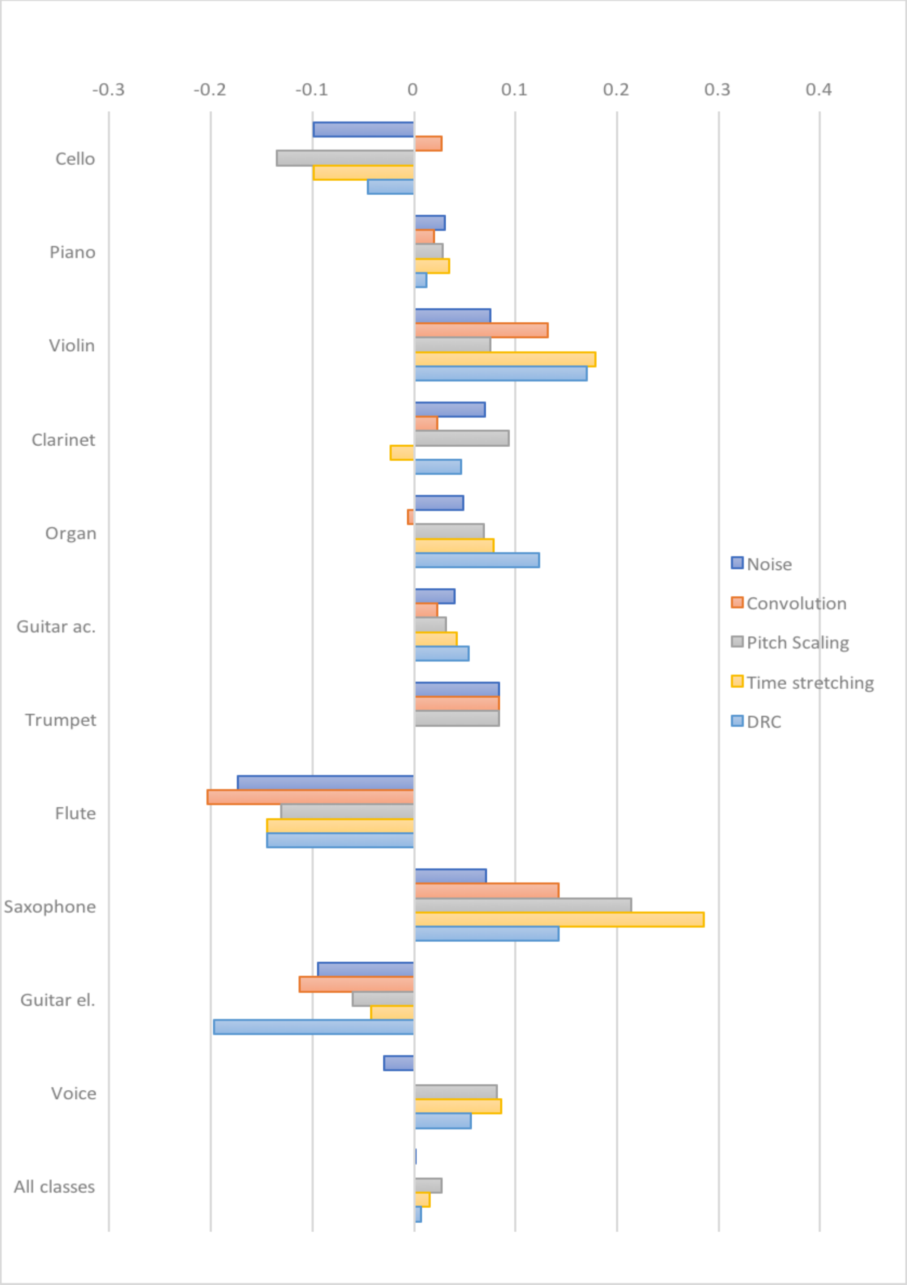


Figure 6: Accuracy changes for each class-augmentation pair

There was a need to further investigate as in which type of augmentations help the model most and which ones not so much or perhaps even degrade the performance. We therefore studied how the effect of these different augmentations separately add up in the combined augmentations model i.e. linearly or non-linearly. For getting more insight, we visualized the results to see how the class accuracies change as described in section 3.5. We studied the changes in accuracies for each type of augmentation for each instrument class. Overall accuracy increased for all augmentations, though only marginally for 'noise' and 'convolution'. Pitch scaling was found to be the best performing augmentation, while time-stretching was the second best augmentation type. In figure 6, we plot the change in classification accuracy for each class with each type of augmentation type, zero axis being the original classification accuracy for the model without data augmentation. It is evident that most classes react positively to the data augmentation, with the exceptions of 'cello', 'flute' and electric guitar'. It remains an open question as to why these categories react negatively to data augmentation. We put forth our idea of approaching this in the next section.

Chapter 5

Conclusions and Future Work

Concluding this thesis, we would like to reiterate the contributions of the work. We reviewed the MIR literature and found scarce research exploiting the benefits of data augmentation, a technique which had been successfully applied in other domains previously. Thus, we started with the development of a general purpose audio data augmentation library. We tested our library on a simple classifier, applying data augmentation for the task of instrument classification and got very promising results. We then replicated the results of these preliminary experiments, using data augmentation with single layered and deeper Convolutional Neural Networks based models for the same task of instrument classification. We experimented with the current state of the art models, improving the performance by almost 6% for the single-layer architecture and by 4% for the multi-layer architecture. Using data augmentation with even the basic single layer model (explained in section 3.5) achieved results comparable to the deeper state of the art models. Then, we also investigated the effects of augmentation on the deeper model architectures, using the multi-layer model explained in section 3.5. With this, we achieved even better results with a f1-score of 0.621 which is a 2% increase over the best performing model using approximately half the number of parameters (1446k for Han et. al. model and 743k for the multi-layer model).

We further studied the influence of each type of augmentation separately on the performance of the single-layer model, showing an improvement in overall accuracy for all the augmentations. With these experiments, we also found that not all augmentations are helpful for all instrument classes, so the augmentations should be designed keeping these conclusions in mind. For our task, we found pitch shifting to be the most useful augmentation overall, followed by time-stretching. We would like to further study the classes/augmentation pairs for which the performance accuracies decreased, so as to develop a better understanding of data augmentation and its effects on the deep learning models, and how it is influenced by the domain and the problem it is being employed in. A more detailed study could also be conducted on the changes in the spectrograms and the learned filters of the CNN, before and after augmentation for each instrument class-augmentation type pair. This could tell us precisely why the model is confusing a particular class with another class, with the additional understanding of the influence of the augmentation on the spectrogram or the learned filter.

We believe that with the advent of deep learning in MIR and the paucity of well-annotated data, incorporating synthetic data augmentation during the training phase could be a really helpful technique to achieve better results. It helps in making the model more robust, ensuring that it learns the typical characteristics of the class rather than the data itself.

We would have liked to test the performance of our audio augmentation library against other alternatives like the Audio Degradation Toolbox and the MUDA but could not do so because of the time constraint. All the experiments were performed on the IRMAS dataset, leaving room for the testing of our methodology on other datasets, which hopefully are also more representative of the real world data. Trying out different combinations of augmentations and their parameters might also be helpful.

We would like to test data augmentation for some other MIR tasks in the future. Keeping in mind that each augmentation type influences each class differently, performance of the models can be further improved by doing class specific augmentations.

List of Figures

Bibliography

[rub, 2012] (2012). Rubberband library. `http://rubberbandaudio.com/`. [Online; accessed 19-July-2017].

[sox, 2013] (2013). sox library. `http://sox.sourceforge.net/`. [Online; accessed 19-July-2017].

[Bogdanov et al., 2013] Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). ESSENTIA: An audio analysis library for music information retrieval. *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 493–498.

[Bosch et al., 2012] Bosch, J., Janer, J., Fuhrmann, F., and Herrera, P. (2012). A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals. *13th International Society for Music Information Retrieval Conference*, (Ismir):559–564.

[Cemgil and Gürgen, 1997] Cemgil, A. T. and Gürgen, F. (1997). Classification of musical instrument sounds using neural networks. In *Proc. of SIU97*.

[Cui et al., 2014] Cui, X., Goel, V., and Kingsbury, B. (2014). Data Augmentation for Deep Neural Network Acoustic Modeling. *ICASSP*, pages 5582–5586.

[DeVries and Taylor, 2017] DeVries, T. and Taylor, G. W. (2017). Dataset Augmentation in Feature Space. pages 1–12.

[Dieleman et al., 2011] Dieleman, S., Brakel, P., and Schrauwen, B. (2011). Audio-based Music Classification with a Pretrained Convolutional Network. *International Society for Music Information Retrieval Conference (ISMIR)*, (Ismir):669–674.

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

[Eronen, 2001] Eronen, A. (2001). Automatic musical instrument recognition. *Mémoire de DEA, Tempere University of Technology*, (April):69.

[Essid et al., 2006] Essid, S., Richard, G., and David, B. (2006). Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4):1401–1412.

[Fuhrmann et al., 2012] Fuhrmann, F. et al. (2012). Automatic musical instrument recognition from polyphonic music audio signals.

[Han et al., 2017] 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*, 25(1):208–221.

[Han and Lee, 2016] Han, Y. and Lee, K. (2016). Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation. 14(8):1–11.

[Huang et al., 2015] Huang, P. S., Kim, M., Hasegawa-Johnson, M., and Smaragdis, P. (2015). Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation. *IEEE/ACM Transactions on Speech and Language Processing*, 23(12):2136–2147.

[Humphrey et al., 2012] Humphrey, E. J., Bello, J. P., and LeCun, Y. (2012). Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics. *International Society for Music Information Retrieval Conference (ISMIR)*, (Ismir):403–408.

[Humphrey et al., 2014] Humphrey, E. J., Salamon, J., Nieto, O., Forsyth, J., Bittner, R. M., and Bello, J. P. (2014). 15th International Society for Music Information Retrieval Conference (ISMIR 2014) JAMS : A JSON ANNOTATED MUSIC SPECIFICATION FOR REPRODUCIBLE MIR RESEARCH. *ISMIR 2014:Proceedings of the 15th International Society for Music Information Retrieval Conference*, (Ismir):591–596.

[Jaitly and Hinton, 2013] Jaitly, N. and Hinton, G. E. (2013). Vocal Tract Length Perturbation (VTLP) improves speech recognition. *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language.*

[Joder et al., 2009] Joder, C., Essid, S., and Richard, G. (2009). Temporal Integration for Audio Classification With Application to Musical Instrument Classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(1):174–186.

[Kaminsky and Materka, 1995] Kaminsky, I. and Materka, A. (1995). Automatic source identification of monophonic musical instrument sounds. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 1, pages 189–194 vol.1.

[Kanda et al., 2013] Kanda, N., Takeda, R., and Obuchi, Y. (2013). Elastic spectral distortion for low resource speech recognition with deep neural networks. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, pages 309–314.

[Ko et al., 2015] Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (2015). Audio augmentation for speech recognition. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015-January:3586–3589.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323.

[Lerch, 2015] Lerch, A. (2015). Chord detection using deep learning. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 52–58.

[Li and Chan, 2011] Li, T. L. H. and Chan, A. B. (2011). Genre classification and the invariance of MFCC features to key and tempo. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6523 LNCS, pages 317–327.

[Lostanlen and Cella, 2016] Lostanlen, V. and Cella, C.-E. (2016). Deep convolutional networks on the pitch spiral for musical instrument recognition. *arXiv preprint arXiv:1605.06644*, pages 612–618.

[Martin, 1999] Martin, K. D. (1999). Sound-Source Recognition : A Theory and Computational Model. *Electrical Engineering*, Doctor of(1993):172.

[Mauch and Ewert, 2013] Mauch, M. and Ewert, S. (2013). The Audio Degradation Toolbox and Its Application To Robustness Evaluation. In *14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 2–7.

[Mcfee et al., 2015] Mcfee, B., Humphrey, E. J., and Bello, J. P. (2015). A software framework for musical data augmentation. *Ismir*, pages 248–254.

[Molina, 2016] Molina, E. (2016). Audio degrader library. `https://github.com/EliosMolina/audio_degrader`. [Online; accessed 19-July-2017].

[Pons et al., 2016] Pons, J., Lidy, T., and Serra, X. (2016). Experimenting with musically motivated convolutional neural networks. In *Proceedings - International Workshop on Content-Based Multimedia Indexing*, volume 2016-June.

[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. *arXiv*.

[Ragni et al., 2014] Ragni, A., Knill, K. M., Rath, S. P., and Gales, M. J. (2014). Data augmentation for low resource languages. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 810–814.

[Richardson et al., 2015] Richardson, F., Reynolds, D., and Dehak, N. (2015). Deep Neural Network Approaches to Speaker and Language Recognition. *IEEE Signal Processing Letters*, 22(10):1671–1675.

[Salamon and Bello, 2017] Salamon, J. and Bello, J. P. (2017). Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Processing Letters*, 24(3):279–283.

[Schlüter and Böck, 2014] Schlüter, J. and Böck, S. (2014). Improved musical onset detection with Convolutional Neural Networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6979–6983.

[Schlüter and Grill, 2015] Schlüter, J. and Grill, T. (2015). Exploring data augmentation for improved singing voice detection with neural networks. *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 121–126.

[Slizovskaia et al., 2016] Slizovskaia, O., Gómez, E., and Haro, G. (2016). Automatic musical instrument recognition in audiovisual recordings by combining image and audio classification strategies. *13th Sound and Music Computing Conference (SMC 2016)*, pages 0–5.

[Yu and Slotine, 2009] Yu, G. and Slotine, J. J. (2009). Audio classification from time-frequency texture. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 1677–1680.