USING AUDIO LOOPS FOR INSTRUMENT FAMILY RECOGNITION IN MACHINE LEARNING TASKS

İsmet Emre YÜCEL[1]
Taylan ÖZDEMİR[2]

Abstract

This paper introduces an instrument recognition approach with the aid of audio loops. The aim is to show a basic instrument recognition recipe for music technology researchers by investigating whether the DAW-based audio loops can be an alternative to researched-based available libraries such as McGill University master samples, UIOWA samples, IRMAS audio libraries. For that purpose, audio loops from Apple Jam Pack were preferred to create instrument classes (Families). The loops were arranged according to their related instrument classes. The class names are Bass, Drums and Percussions, Guitars, Keyboards, Strings, Synthesizers, and Winds. After the extraction of temporal and spectral audio features from those classes, a 5736x105 dimensional dataset emerged. Then this dataset was examined with 19 different supervised machine learning algorithms. The SVM Cubic classification algorithm provided the best accuracy (90.2%). The result shows that the audio loops with mid-term feature extraction can be used for instrument recognition tasks.

Keywords: Instrument Recognition, Machine Learning, Audio Content Analysis, Music Information Retrieval, Music Technology, Usages of Audio Loops

Introduction

The Instrument Recognition studies have continued for more than three decades without losing importance. They are essential for many music information indexing and retrieval tasks. Sound recognition studies are examined in various disciplines such as biology, medical, surveillance, military, and multimedia, wherein speech, sound effects, and music-related retrieval tasks occur. Speech recognition studies were exclusive because of the industrial demand, and solutions were often used as guides while dealing with instrument recognition tasks. Over the years, the audio features, preference of the features, dimensionality reduction of features, and classification methods have been studied in instrument recognition literature. However, an instrument recognition scenario depends on the audio source types being sole-sourced, multiple-sourced, monophonic, or polyphonic. The aim of this study is to show how a loop-based audio dataset can be created and how the current machine learning algorithms perform with this dataset.

Instrument Recognition is a subordinate field of Music Information Retrieval (MIR) studies and based on Audio Content Analysis (ACA) theories. It seems there are similarities between the MIR and ACA fields, but the MIR concerns a wide variety of digital formats, such as midi, scores, audio, and even their representational or semantic relationships in a coordinative way. On the other hand, the ACA specifically deals with audio formats. Automatic

[1] This paper is based on the first author's Ph.D. thesis in progress, in ITU Institute of Graduate School, Music Doctoral Program,
eyucel@sakarya.edu.tr
ORCID: 0000-0001-7018-3349
Day of Application: 30.04.2021 Acceptance Date: 01.10.2021
[2] Assoc. Prof. Dr. Istanbul Technical University Turkish Music State Conservatory, Music Technology Department
ORCID: 0000-0001-8789-8893

audio alignment, organization of audio in a database, audio visualization, and intelligent audio processing are some subjects that the ACA involves. Lerch (2012:3) states that musical audio content has multi-faceted information that originates from the score (musical notation, form, structure), performance (musical expressions), and production (audio recording, effects, processing). The ACA has significant potential in terms of the audio industry, and Herrera et al. (1999) remarked that potential by suggesting to add the content-based audio descriptors scheme into the MPEG-7 standard. Also, the combination of music production with intelligent systems leads emergence of the Intelligent Music Production (IMP) field. The primary focus of this field is to find fully or semi-automatic solutions for music production stages. De Man et al. (2020:3) indicates that machine learning techniques are becoming an essential tool for IMP systems together with the knowledge of engineering, psychoacoustics, perceptual evaluation.

A general framework for an instrument recognition system comprises those efforts, choosing a sample library, audio feature extraction to create a dataset, data-processing, and classification. After obtaining a sample library, each audio file should be revised for the research purpose while solving a specific machine learning problem. In other words, the audio contents should correspond to their class to prevent any irregularities or duplicates on the feature vector. Further, the audio files may need pre-processing such as stereo-to-mono conversion, audio format conversion, downsampling, DC removal. Thus, audio pre-processing is a good routine before whenever the audio feature extraction step takes place. Next, the feature extraction process provides the creation of an audio dataset. In the dataset, because some features may be definite in different numerical ranges, data pre-processing (like scaling, normalization) is recommended. Then, the dataset is ready to be examined in supervised machine learning methods for instrument recognition. It should be kept in mind that the accuracy rate of a classification algorithm shows the feasibility of the dataset, but it does not mean the accuracy is always achievable when the instrument recognition system is tested with external audio files.

Audio features are the low-level statistical representation of audio data that constitutes the core of the ACA. Low-level statistical information is a kind of metadata that only the computer-based systems interpret. Each audio feature (also called descriptor) is based on a specific mathematical theory and calculated in a predetermined sample range named a short-term window frame, and those frames overlap at a certain percentage. This overlapping of the successive windows is called “hop size.” Generally, the window frame length may range from 128-2048 samples and hop defined as a percentage (like 50%), but it is application and system dependent. The “quasi-stationary” structure of the digital audio each frame needs applying a window function. Schuller (2013:45) mentioned the rectangular, Hamming, and Hanning window functions and indicated that the most popular one is the Hamming window function. The audio feature calculation happens in the time and frequency domains. Zero-Crossing Rate (ZCR), RMS Energy, Energy Entropy are some time-domain features. Some frequency-domain features are Spectral Centroid, Spectral Spread, Spectral Entropy, Spectral Flux, Spectral Roll-off, MFCCs, Chroma Vector, and Harmonic Ratio.

Zero-Crossing Rate (ZCR), RMS Energy, and Energy Entropy are calculated directly from the signal. ZCR calculates how frequently an audio signal changes its position between the positive and the negative parts in the time domain. It is beneficial to distinguish the character of the signal, whether noise-like or not, so it gives a clue about the timbre of the source. Typically, the ZCR value is low during silence or lower frequency parts of the signal. Thus, it becomes effective for detecting silent and voiced parts of a signal.

Energy is the definition of the loudness of each successive frame of the signal. In speech signals, a high rate of alternation is observed between sequential frames. Herrera-Boyer et al. (2003) indicate that “one of the most

commonly used descriptors for musical, as well as non-musical, sound classification is energy." (p. 10). Energy Entropy indicates rapid energy changes in an audio file. This feature is advantageous in onset detection. Giannakopoulos & Pikrakis (2014:77) highlighted the high potential of the Energy Entropy in genre classification.

During the years, psycho-acoustic experiments on human sound perception led to discovering new audio timbral features. Those features are defined in the frequency domain, and they are based on the Fast Fourier Transform (FFT). Some frequency domain features are Spectral Centroid, Spectral Spread, Spectral Entropy, Spectral Flux, Spectral Roll-off, MFCCs, Chroma Vector, and Harmonic Ratio.

The Spectral Centroid and Spread are two closely related audio features. The definition of the spectral centroid "is the center of 'gravity' of the spectrum." (Giannakopoulos & Pikrakis, 2014:79). For the spectral centroid, a higher value means the audio file has a brighter character. On the other hand, the Spectral Spread defines how the spectrum of the signal propagated around the centroid. Lerch (2012:47) indicates that the higher spread values are observed at the transients for monophonic signals.

The Spectral Entropy defines spectral shape providing minimum value for a continuous signal and maximum value for a short signal (peak). "Spectral entropy is computed in a similar manner to the entropy of energy, although, this time, the computation takes place in the frequency domain."(Giannakopoulos & Pikrakis, 2014:81).

The Spectral Flux is used "to detect spectral changes in the signal, one basically computes the difference between subsequent spectral vectors using a suitable distance measure."(Müller, 2015:309). This feature describes the roughness of the signal. Schuller (2013) indicates that "Speech SF values are higher than music ones and the environment sound has the highest value. Also, the environmental sound changes dramatically between successive frames." (p. 146).

The Spectral Roll-off "is defined as the frequency below which a certain percentage (usually around 90%) of the magnitude distribution of the spectrum is concentrated."(Giannakopoulos & Pikrakis, 2014:85). Lerch highlights that the behaviour of the Spectral Roll-off "at pauses in the input signal may require special consideration. While the result will equal zero for absolute silence, it may be quite large for noise, including pauses with low-level noise."(Lerch, 2012:42). It makes the Spectral Roll-off functional for ACA applications.

Mel-Frequency Cepstrum Coefficients (MFCC) is widely used in the representation of both speech and music signals. The cepstrum stands for the logarithmic transformation of the spectrum for a signal. MFCC uses the Mel scale, which mimics the human perception of pitch. In Eronen's (2001:22) study, the results showed that "the mel-frequency cepstral coefficients gave the best accuracy in instrument family classification, and would be the selection also for the sake of computational complexity."

The Harmonic Ration (HR) is an indicator that whether the audio file is periodic or aperiodic (noise-like). The periodicity does not fully explain the character of the regular signals such as voice and music. This situation leads to the usage of the term "quasi-periodic" for them. HR feature is used fundamental frequency estimation of a signal.

The Chroma Vector is a representation of audio spectral energy in 12 bins as a matrix. "Each bin represents one of the 12 equal-tempered pitch classes of Western-type music (semitone spacing). Each bin produces the mean of log-magnitudes of the respective DFT coefficients." (Giannakopoulos & Pikrakis, 2014:91). For musical signals, the bins are prominent in a short-term frame in chromagram.

Mid-term windowing (also called texture window) is a common technique to get a meaningful representation of the audio features, particularly for audio files longer than 1 second. It is especially preferred in the genre, musical similarity, and mood classifications. While the duration of short-term window frames is typically ranging between 10-40 msec, for mid-term window frames, the duration is between 1-20 s. Accordingly, each short-term feature is extracted individually, then some statistical calculations are applied to the features at the mid-term frame. For an analogy, the short-term features are similar to letters, and the mid-term statistics stand for words or sentences. Arithmetic mean, median, standard deviation, and standard deviation by mean are some of the statistical methods which can be used while utilizing mid-term windowing.

In most cases, a vast audio vector composes after a feature extraction process. Dimensionality reduction is necessary to acquire a more meaningful representation with a smaller dimension of the audio vector. Feature extraction and feature selection are two categories in dimensionality reduction. One of the most popular reduction methods is PCA (Principal Component Analysis), "In PCA data is projected into abstract dimensions that are contributed with different –but partially related- variables. Then PCA calculates which projections, amongst all possible, are the best for representing the structure of data." (Herrera-Boyer et al., 2003:8). The Singular Value Decomposition Method, Fisher's Linear Discriminant Analysis, The Kernel PCA, Laplacian Eigenmap, Independent component analysis (ICA), and Non-negative matrix factorization (NMF or NNMF) are other dimensionality reduction methods. For feature selection, there are three primary methods, which are filter, wrapper, and embedded. Agostini et al. (2003) indicated the performance of the identification system depends on feature choice. Essid et al. (2006a) worked on pairwise strategies for classification to find the most relevant features.

Generally, unsupervised and supervised approaches are two main categories in machine learning. The supervised approach is suitable for instrument recognition applications because "supervised learning consists of understanding the relationship between a given set of features and a target value, also known as a label or class." (Saleh, 2020:42). The term supervised means each class (or target) name known by the algorithm before starting the classification task. Each target description coincides with its feature row in a dataset. Conversely, an unsupervised method does not expect any target name but groups the samples according to their similarities. Bishop (2006) has sorted out the classification algorithms under those main topics: Linear models, Neural networks, Kernel Methods, Sparse Kernel Machines, Graphical Models, and Mixture Models. Over the years, various machine learning methods have been examined for instrument recognition tasks. K-Nearest Neighbours, Bayesian Classifiers, Discriminant Analysis, and Decision Tree are some of them.

Method

Building Dataset

In this work, Apple Jam Packs were used to create an audio dataset. The available libraries which the researchers have are Jam Pack 1, Jam Pack Remix Tools, Jam Pack Rhythm Section, Jam Pack Symphony Orchestra, Jam Pack Voices, Jam Pack World Music folders, respectively. For the scope of this study, the Voices and Word Music folders were excluded. Each folder comprises more than 2000 pieces of audio loops with different types of instruments. Due to the irregular content, the folders are not suitable to use in a classification task directly. It was required to reorganize their content according to their instrument family. Therefore, the audio files from the loop library were arranged into seven instrument groups. The names of family groups are Bass, Drums and Percussions, Guitars, Keyboard, Strings, Synthesizers, and Winds.

The format of the audio files was stereo CAF[3], which is not a standard audio file format. Thus, they were converted into mono wave files without editing the lengths. The information about those audio files and the instrument families is given in **Table-1**.

Table 1: Instrument families (classes) in seven instrument groups and their properties.

Classes	Some Properties of the Audio Classes	Min. Max. Length	Number of Audio Files
Bass	Bass guitars, some distorted, Double basses (played with different techniques)	1-23 sec.	834
Drums & Percussions	Drum-sets (fills, different styles with many playing techniques), Electronic-Dance Drums, Drum-set with percussion(s), Drums Machines, Percussions (such as congas, Tambourine, and similar.)	1-20 sec.	1965
Guitars	Acoustic, Nylon, Electric, with various styles (rock, metal, jazz and similar) and playing technique (fingers, plucked, strummed, slides, and similar), different effects (like distortion, wah-wah, reverb)	1-32 sec.	1114
Keyboards	Organs, Clavinets, Wurlitzers, Rhodes, Acoustic and Electronic Pianos.	1-32 sec.	602
Strings	Mostly orchestral, sample-based, or real. Various playing techniques, some recordings are solo, some ensemble. The predominant effect is reverb.	1-32 sec.	398
Synthesizers	Various Funk Synths, Synth basses, pads, some arpeggiated, mostly chords, some have effects (like delay, reverb).	1-42 sec.	464
Winds	Brasses, Harmonica, Flutes, French and English Horns, Clarinets, Oboes. Some performances are solo but mostly played as groups.	1-40 sec.	358

Audio Feature Extraction

The audio feature extraction is an inevitable process before conducting a machine learning task. Giannakopoulos and Pikrakis (2014) describe the audio feature extraction as “representing the properties of the original signals while reducing the volume of data.” (p. 59). Some frame-based information can be derived from the processed audio file and filtered for research.

Generally, the audio feature extraction is a two-step process, short-term and mid-term. The short-term features, also called low-level features, are calculated at a particular window (FFT) length with a step size. Step size (also called hop size) is a length that determines the number of samples between each successive FFT window; for example, %25 refers to a quarter of the window size. The purpose of using consecutive steps (hop) is to achieve more detailed calculation results from an audio file. On the other hand, mid-term feature extraction allows obtaining a general perspective about an audio file. Giannakopoulos and Pikrakis (2014) explain the mid-term feature extraction as a process that “can be employed in a longer time-scale scenario, in order to capture salient features of the audio signal.” (p. 65).

[3] Core Audio Format by Apple

There are a lot of Python libraries for MIR works. In this research, pyAudioAnalysis (Giannakopoulos, 2015) is preferred for the audio feature extraction task. Additionally, some modifications were applied to the library code to acquire median statistics of the features and saving the feature vector as a spreadsheet file to examine other machine learning platforms such as Matlab. As a result, the audio dataset had 5736x105 dimensions.

Properties of Short-Term Features

In this part, the window size of FFT is 0.04 seconds, the step of each window is 0.02 (%50 hop) seconds. For each audio file, the calculated audio features are Zero Crossing Rate, Energy, Energy Entropy, Spectral Centroid, Spectral Spread, Spectral Entropy, Spectral Flux, Spectral Roll-off, Harmonic Ratio, MFCC, and Chroma Vector. In the dataset, MFCC holds 13 columns, Chroma Vector contains 12 columns, and the others hold a single column. Therefore, audio features have been calculated at each short-term frame as 35 columns in total **(Table-2)**.

Table 2: Audio Features and Number of Columns

Features	**Number of Columns**
Zero-Crossing Rate	Single
Energy (Power)	Single
Entropy of Energy	Single
Spectral Centroid	Single
Spectral Spread	Single
Spectral Entropy	Single
Spectral Flux	Single
Spectral Roll-off	Single
Harmonic Ratio	Single
MFCC	13
Chroma Vector	12+1 (mean of the bins)

Properties of Mid-Term Features

In the dataset, each audio file has various lengths, and because most of them exceed 1 second, the mid-term statistical approach is used to get quick results at the stages of feature extraction. The mid-term window length is 1 second, and the step size is 500 msec (%50 hop). The applied statistics on each mid-term frame were arithmetic mean, median, and standard deviation.

Data Pre-processing

Generally, data are represented in different ranges and types in a dataset. Nevertheless, in most situations, the diversity of data types and ranges cannot be interpreted by classification algorithms successfully. Because of that, datasets should be reviewed, cleaned, and prepared before the machine learning algorithm runs. Zheng and Casari (2018) indicate that some classifications that use smoothing functions at inputs (like regression-based models) are affected by scaling. Although in this dataset, all feature columns are represented as floating-point types, some columns range differently. Therefore, standard scaling was applied to the data in order to centralize every column around zero.

Classification

For machine learning and its applications, the Matlab environment provides a versatile tool called Classification Learner, which comes within the "Statistical and Machine Learning Toolbox." The tool allows working on supervised classification methods with various classifiers.

Before starting classification, holdout validation is applied to the dataset to prevent overfitting. The held-out percentage is % 25, which means a quarter of each instrument group is divided for testing and the rest for the training. In this research, the deployed classification methods are Decision Trees, Discriminant Analysis, Bayesians, Support Vector Machines (SVM), K-Nearest Neighbour (K-NN). Thus, with several types of these classifiers, 19 numbers of algorithms were compared in this study.

The Decision Tree is a famous classification and regression method. In this method, input data is tested with many basic conditional operators (if questions). The testing process resumes from top to bottom over different decision nodes. The term "decision node" is used for every conditional step, and each one can branch out to a new decision node. A "leaf" symbolizes a prediction result. One decision node can have multiple branches towards the other nodes and reaches the final leaf (prediction). Saleh (2020) highlighted that "Decision trees can handle both quantitative and qualitative features, considering that continuous features will be handled in ranges. Additionally, leaf nodes can handle categorical or continuous class labels; for categorical class labels, a classification is made, while for continuous class labels, the task to be handled is regression." (p. 141). A fundamental decision tree diagram is given in **Figure-1.**

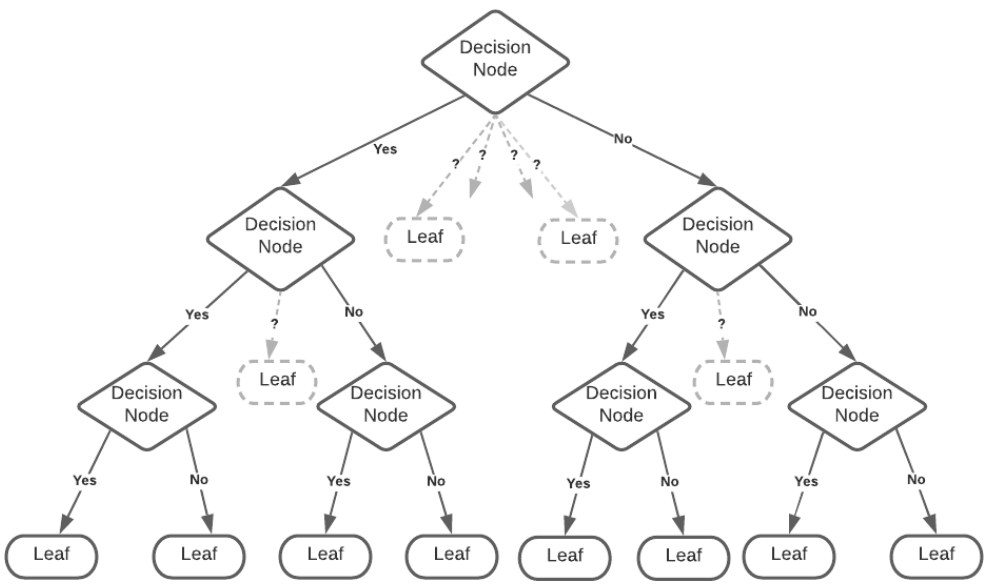

Figure 1: A basic diagram for Decision Tree algorithm. Each decision nodes can have more than two branches (as yes, no and others). Leaves stand for the results or predictions.

Discriminant analysis is also known as Fisher discriminant analysis. It is based on binary classification (defined as K=2) but can be used for multivariate and multiclass (K>2) problems. Simply, a discriminant function assigns the input vector to a class (K), and the classes are represented on hyperplane surfaces. The hyperplane surface is divided by the boundaries that define classes. A basic demonstration of discriminant analysis is given in **figure 2.** D1, D2, and D3 represent the classes. Xa and Xb points are classified in the D3 class. If x is a point between Xa and Xb, then the algorithm decides that x also belongs to the D3 class.

In machine learning, some classification methods are based on Bayesian probabilistic functions. In Bayesian classification theory, "the optimal classification decision can be achieved based on the knowledge of the distributions of feature vectors and the prior probabilities of the classes." (Tulyakov & Govindaraju, 2013). According to Nisbet et al. (2018:184), the advantages of this algorithm is fast during the training and classification, and more not sensitive unimportant variables. But disadvantages the algorithm assumes all variables are independent, which means interaction between them is excluded.

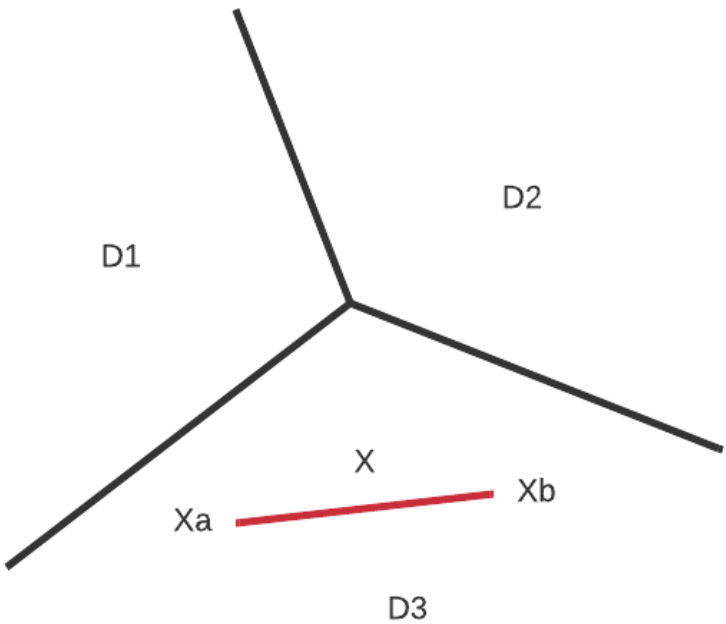

Figure 2 Discriminant analysis demonstration it a hyperplane surface. D1, D2, and D3 represents the classes. Xa, Xb, and x are definite in the D3.

SVM (Support Vector Machine) algorithm has been used to solve different classification problems since the 90s. Izenman (2008) highlights the extensive usage of the algorithm as "SVMs have been successfully applied to classification problems as diverse as handwritten digit recognition, text categorization, cancer classification using

microarray expression data, protein secondary-structure prediction, and cloud classification using satellite-radiance profiles." (p. 369) SVM method depends on a kernel function to map the feature vectors to kernel space. The kernel type can be linear, quadratic, cubic, or gaussian based. There are two strategies for the solution of multiclass problems, one-versus-rest, and one-versus-one. These strategies depend on the application.

The k-NN (K-Nearest Neighbourhood) is one of the simplest machine learning algorithms. Albon (2016) describes the k-NN as a "lazy learner" and states that "it does not technically train a model to make predictions. Instead, an observation is predicted to be the class of that of the largest proportion of the k nearest observations." (p. 251). This technique strongly depends on a distance (dissimilarity) measure function such as Euclidean. The parameter k stands for the number of neighbours, and it changes according to the dataset and the number target (class) in an application.

The solution of instrument recognition problems changes according to the source types being single or multiple sources. While the term "single-sourced" denotes that only one instrument performs in the audio file, "multiple-sourced" means that more than one instrument plays simultaneously. Approaches used for single-sourced instrument recognition tasks also provide a foundation for solving multiple-sourced instrument identification problems, but audio source separation has an essential role in that kind of task. In instrument recognition for polyphonic sources, Essid et al. (2006b) proposed hierarchical taxonomy. Heittola et al. (2009) provide a solution, using a source-filter model and an augmented non-negative matrix factorization algorithm for sound separation, and attained 59% accuracy for six-note polyphony.

Moreover, recent works in instrument recognition for polyphonic sources have concentrated on deep learning methods since they outperform other available state-of-the-art machine learning approaches. They also provide more accurate results in different fields such as image speech recognition and source recognition. Han et al. (2017) state the importance of identifying musical instruments in polyphonic recordings, musical genre classification, and music transcription and proposed the (CovnNet) a Deep Convolutional Neural Networks (DNN) instrument recognition system. On the other hand, Yu (2020) proposed a system that combines DNN with the principal classification with the assistance of auxiliary classification. This system aims to find the predominant instrument which plays simultaneously in polyphonic music. According to the researchers, the proposed system performs 10.7% and 16.4% better than CovnNet.

As mentioned above, before conducting an instrument recognition task, the first objective is to find an appropriate audio sample library. In previous studies, researchers have utilized free audio libraries such as McGill University master samples, UIOWA samples, and IRMAS dataset or custom-made audio libraries for instrument recognition studies. However, audio loops bundled with Digital Audio Workstations (DAW) or other commercially available libraries can also be an excellent alternative to the free libraries mentioned above.

This work aims to provide a basic instrument recognition recipe for music technology researchers and investigate whether the DAW-based audio loops are usable or not. The research objectives are to find an audio loop library and organize them as instrument families, building a dataset from those instrument families by audio features extraction methods, and estimating the best-supervised machine learning algorithm for this dataset.

Results

Accuracies of the classification algorithms are given in **Table-3**. According to the table, while the SVM Cubic algorithm provides the best classification result, % 90.2, SVM Fine-Gaussian gives the worst classification result, % 48.6.

Table 3: The overall success rates of classification algorithms.

Classification	Model Type	Accuracy
Tree	Fine	% 74.8
	Medium	% 70.7
	Coarse	% 64.7
Discriminant	Linear	% 78.2
	Quadratic	% 82.2
Bayesian	Naive	% 67.8
	Kernel Naive	% 70.2
SVM	Linear	% 83.2
	Quadratic	% 89.5
	Cubic	% 90.2
	Fine Gaussian	% 48.6
	Medium Gaussian	% 88.2
	Course Gaussian	% 76
KNN	Fine	% 83.8
	Medium	% 79
	Coarse	% 70.0
	Cosine	% 80.8
	Cubic	% 76.7
	Weighted	% 81.6

The confusion matrix of the best classification (SVM-Cubic) result is given in **Figure-3.** In the confusion matrix, True Positive Rates (TPR) represents overall accuracy for each instrument family, and False Negative Rates (FNR) shows the classification mistakes. According to the matrix, the algorithm attained a 99% achievement result for the Bass class, and the misclassification rate is 33.3% for the Synths class.

Figure-3: Confusion matrix for SVM-Cubic. In left side percentages of success and misclassification rates for each instrument family are shown in a comparative way. In the right side, True Positive Rates (TPR) and False Negative Rates (FNR) gives a general clue about misclassifications.

Conclusion

A loop-based audio dataset has been created for instrument family recognition tasks with the mid-term statistical approach. Seven instrument families were created: Bass, Drums/Percussions, Guitars, Keyboards, Strings, Synths, and Winds. For this dataset, the SVM-cubic classification algorithm scored high precision results; for Bass 99%, Drums/Percussions 96.1%, Guitars 94.6%, Keyboards 83.3%, Strings 79%, Winds 77.5%, and Synthesizers 66.7% accuracy attained. The overall classification accuracy is 90.2%.

These results show that the approach in this paper can be useful when classifying a wide range of audio files with various durations from different audio classes (instrument families). The size of each class and lengths of individual audio files does not affect the classification result drastically. So, the matter is not quantity but the quality of each audio file's content and how they represent the audio instrument family. Due to the higher classification accuracy results, the audio features seem sufficient to distinguish for given instrument families, at

least for Drums, Bass, and Guitar samples. Therefore, mid-term audio features for the classification of audio files excerpted from audio loops work pretty well. However, the dataset obtained in this study may not provide the same accuracy when tested with external audio files because of the diversity of musical styles, playing, recording techniques. Of course, large-scale datasets with various classes may be preferred to solve different instrument recognition problems, but one must keep in mind that extensive datasets cause a significant drop in the classification speed and may also severely affect the accuracy. In that kind of scenario, the deployment of dimensionality reduction techniques may become inevitable to determine the best result with fewer audio features.

In terms of quality, given in **Table-1**, the audio files in instrument groups have various properties. In the Synth class, the audio files have some audio effects, very distinct characters, and, recorded as chords. There are two concerns here. The first one is that the audio effects change the timbral character of the audio files. Thus, an instrument family class with that kind of audio file will be affected in that situation. The second one is that a wide variety of instrument types cause inconsistent feature structure for a class, especially created by the different synthesizing techniques such as additive, subtractive, granular, wave-shaping, physical modelling. These concerns may explain why the synth family classification provides the worst result (TPR 66.7%) in this experiment. Of course, these are upfront subjects for further research.

The Strings family consists of solo, ensemble, sample base, or real recordings in various reverberant environments. As previously mentioned, the reverberation, especially in the ensemble recordings, drastically changes the timbral character of the instrument family. Similarly, the winds family samples coincide with many distinctive instrument recordings in terms of timbres, and some of them are recorded as groups. This situation explains the low accuracy rate of those audio families, given in **Figure-3**.

As a result, the DAW-based audio loops are a practical choice for dataset creation in instrument recognition tasks. However, the instrument family categorization must be considered carefully, and the audio files from a loop library placed into the related family group in that sense.

References

Agostini, Giulio, Maurizio Longari and Emanuele Pollastri. 2003. “Musical Instrument Timbres Classification with Spectral Features.” *Eurasip Journal on Applied Signal Processing,* 2003(1): 5-14. Springer Open (Accessed April 4th 2021).

Albon, Chris. 2018. *Machine Learning with Python Cookbook*. Sebastopol, CA:O'Reilly Media Inc.

Alice, Zheng and Amanda Casari. 2018. *Feature Engineering for Machine Learning, Principles and Techniques for Data Scientist (1st edn.).* USA:O’Reilly Media.

Bishop, Christopher M. 2006. *Machine Learning and Pattern Recognition. In Information Science and Statistics.* Verlag, NY: Springer Science+Business Media, LLC.

De Man, Brecht, Ryan Stables, Joshua D. Reiss. 2020. *Intelligent Music Production* (1st end.). NY: Routledge.

Eronen, Antti. 2001. “Comparison of Features for Musical Instrument Recognition.” *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics.* 2001:19-22. IEEE Xplore (Accessed April 4th 2021).

Essid, Slim, Gaël Richard, Bertrand David. 2006a. “Musical Instrument Recognition by Pairwise Classification Strategies.” *IEEE Transactions on Audio, Speech, and Language Processing,* 14(4):1401–1412. IEEE

Xplore (Accessed April 5th 2021).

Essid, Slim, Gaël Richard, Bertrand David. 2006b. "Instrument Recognition in Polyphonic Music Based on Automatic Taxonomies." *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):68–80. IEEE Xplore (Accessed April 5th 2021).

Giannakopoulos, Theodoros. 2015. "pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis." *PLOS ONE*, 10(12). <https://www.journals.plos.org/> (Accessed April 6th 2021).

Giannakopoulos, Theodoros and Aggelos Pikrakis. 2014. *Introduction to Audio Analysis: A MATLAB Approach (1st edn.).* Oxford, UK: Academic Press.

Han, Yoonchang, Jaehun Kim, and Kyogu Lee. 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. IEEE Xplore (Accessed April 6th 2021).

Heittola, Toni, Anssi Klapuri and Tuomas Virtanen. 2009. "Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation*". Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR* 2009, 327–332. Kobe:Japan, October 26-30.

Herrera-Boyer, Perfecto, Geoffroy Peeters and Shlomo Dubnov. 2003. "Automatic Classification Of Musical Instrument Sounds." *Journal of New Music Research*, 21(1):3–21. Routledge.

Herrera, Perfecto, Xavier Serra, Geoffroy Peeters. 1999. "Audio Descriptors and Descriptor Schemes in the Context of MPEG-7." *International Computer Music Conference*. Beijin:China, (22-27 October 1999). Michigan Publishing.

Izenman, Alan Julian. 2008. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning (1st edn.).* NY:Springer.

Lerch, Alexander. 2012. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. NJ: Wiley-IEEE Press.

Müller, Meinard. 2015. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.* Switzerland:Springer.

Nisbet, Robert, Gary Miner and John Elder. 2018. *Handbook of Statistical Analysis and Data Mining Applications (2nd edn.).* London,UK:Elsevier.

Saleh, Hyatt. 2020. *The Machine Learning Workshop (2nd edn.).* Birmingham,UK:Packt Publishing.

Schuller, Björn W. 2013*. Intelligent Audio Analysis (1st edn.).* Berlin, Germany:Springer.

Tulyakov, Sergey and Venu Govindaraju. 2013. "Matching Score Fusion Methods." *Handbook of Statistics: Machine Learning: Theory and Applications*. Vol (31):151–175. ScienceDirect. (Accessed April 8th 2021).

Yu, Dongyan, Huiping Duan, Jun Fang, Bing Zeng. 2020. "Predominant Instrument Recognition Based on Deep Neural Network with Auxiliary Classification." *IEEE/ACM Transactions on Audio Speech and Language Processing*, Vol. 28:852–861, 2020. IEEE Xplore (Accessed April 9th 2021).