FACULTY OF MATHEMATICS
AND COMPUTER SCIENCE
University of Lodz

Eldar Mukhtarov
Student ID: 412509

Advanced Acoustic Analysis and Classification of Instrumental Sounds in High-Dimensional Feature Spaces

Bachelor's Thesis
in the field of Computer Science

Thesis written under the supervision of
prof. dr hab. Stanisław Goldstein
Katedra Informatyki Stosowanej

Łódź, 2025

Contents

Abstract

This thesis provides an analysis of the classification of isolated instrumental sounds in acoustic data using a classical Machine Learning method — Support Vector Machine (SVM). The study uses the Philharmonia Sound Samples library and curates a dataset of 13,533 professionally recorded, monophonic clips from 19 instruments (average duration 1.91 s, sampling rate 44.1 kHz). A 32-dimensional feature vector composed of 13 MFCCs, 7 chroma features, and 12 spectral-contrast features represents each clip. A Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel was selected for its strong performance in high-dimensional spaces, computational efficiency, and good generalization properties. Other traditional classifiers (e.g., KNN) were considered but found less suitable for the chosen feature representation and problem scale. The dataset was split stratified 80:20 (10,826 training / 2,707 test samples). Hyperparameter optimization using Grid Search resulted in an optimal configuration that achieved approximately 98.96% accuracy on the held-out test set. The macro-averaged precision was around 98.5%, recall was about 97.2%, and the F1 score was approximately 97.6%. Error analysis revealed that most misclassifications occur between acoustically similar instruments, such as the violin and viola, as well as closely related plucked string instruments. These distinctions can also be challenging for human listeners. This thesis documents the feature-engineering decisions, methodological rationale, evaluation metrics, and a reproducible implementation using `librosa` and `scikit-learn` on Google Colab. We conclude that a compact and carefully tuned Support Vector Machine (SVM) pipeline, utilizing complementary timbral and harmonic features, can achieve near-state-of-the-art results on clean, monophonic datasets.

Keywords: Machine Learning · Instrumental Sound Classification · Support Vector Machine · Mel-frequency Cepstral Coefficients · Chroma Features · Spectral Contrast · Hyperparameter Tuning · Music Information Retrieval

List of Abbreviations

ADSR	Attack, Decay, Sustain, Release
AI	Artificial Intelligence
CENS	Chroma Energy Normalized Statistics
CNN	Convolutional Neural Network
CQT	Constant-Q Transform
DFT	Discrete Fourier Transform
DL	Deep Learning
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
FT	Fourier Transform
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MFCC	Mel-frequency Cepstral Coefficients
MIR	Music Information Retrieval
ML	Machine Learning
RBF	Radial Basis Function
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

Introduction

The classification of instrumental sounds is widely used in audio signal processing, with applications in music recommendation [1], transcription [2], and audio restoration [3]. It is a fundamental area of research in Music Information Retrieval (MIR), a field dedicated to extracting tangible information from audio signals. The objective of classifying acoustic sounds is to accurately categorize and analyze these sound types with more efficient processing capabilities and potentially improved user experience. The recognition of instruments is a nontrivial problem, mainly because of the variations in their playing style, dynamics, recording conditions, and similarity between instruments. Despite these variations, human listeners can often recognize instruments, but studies show that machines can perform on par with them [4] or even exceed human performance [5].

This surpassing of human performance has, in recent years, been primarily due to Machine Learning (ML). Although Deep Learning (DL) models can demonstrate remarkable results, they need large amounts of datasets, extensive computational power, and complex architectures [6], and this is a limitation for many researchers from fully leveraging their potential. The existence of such factors necessitates an exploration into the efficacy of more classical yet powerful Machine Learning (ML) algorithms, which, when combined with proper feature engineering, shows a more efficient and promising approach in instrument classification.

Hence, this thesis focuses on achieving high-accuracy classification of instrumental sounds while maintaining computational efficiency. Despite complex Deep Learning (DL) models being a viable solution, this paper analyzes the extent to which a simpler and a more interpretable model — Support Vector Machine (SVM) — can achieve equivalent or even superior results. The core hypothesis is that a meticulously engineered, high-dimensional feature space with rigorous hyperparameter optimization can prove Support Vector Machine (SVM) effective in the complexities of acoustic data, performing efficacious and efficient classification.

This study uses isolated and monophonic instrumental sounds acquired from a high-quality and professionally recorded dataset [7]. The scope is intentionally constrained to a specific set of 19 orchestral and common non-orchestral instruments. The research exclusively utilizes a Support Vector Machine (SVM), given the satisfactory performance achieved with this model alone, without exploring comparative

implementations of other Machine Learning (ML) models. The primary application of these findings is in scenarios involving clean audio signals, without interference from background noise or polyphonic textures.

The thesis consists of a literature review, theoretical foundations of audio feature extraction, classical machine learning models, and performance metrics, followed by practical implementations of feature engineering, model training and optimization, and a comprehensive analysis of the results. The work concludes with a discussion of the findings, their broader implications, and potential avenues for future research in instrumental sound classification.

1 Literature Review

Numerous studies have investigated the recognition of distinct sounds with the help of Artificial Intelligence (AI). While Music Information Retrieval (MIR) has broad usage and applications, instrumental sound recognition is sufficiently specialized that it has been relatively less researched. In particular, monophonic music is considered to be more tractable for recognition owing to the presence of a single instrument. When interference (e.g., a piano accompaniment) is introduced, the recognition accuracy and response times worsen, making the task especially degrading for human detectors, as demonstrated in [8]. Bürgel and Siedenburg (2024) ran a set of go/no-go recognition trials involving thirty participants using short vocal and instrumental excerpts presented in isolation or accompanied by interference. The researchers measured participants' response times and recognition accuracy. Ultimately, the study found that interference reduced recognition accuracy and increased response times: participants were able to detect isolated sounds in approximately 680 ms, while recognition under interference of a piano almost doubled to over 1100 ms. These findings corroborate psychophysical studies showing that, even in simple tasks, mean auditory reaction times are generally around 140–160 ms [9], suggesting that the human auditory system requires a considerable temporal window to process and identify sounds. By contrast, certain audio-processing hardware can perform compressive sensing and reconstruction instantly or with latencies as low as ~10 ms [10], underlining that machines can surpass human reaction times in speed of signal analysis.

Prior research on musical instrument recognition employing machines, such as deep learning (DL) models, has investigated a variety of approaches. In the literature, three complementary families of frame-level features are frequently used: timbral descriptors (most commonly Mel-Frequency Cepstral Coefficients, MFCCs), harmonic/pitch-class descriptors (chroma), and spectral-texture descriptors (spectral contrast). Empirical studies show that MFCCs provide a strong baseline for timbre-based discrimination [11], while combining timbral and harmonic/textural features generally improves classification accuracy compared with any single family alone [12]. These combined feature pipelines can be paired with classical classifiers such as Support Vector Machine (SVM) or K-Nearest Neighbor (KNN), forming a pragmatic foundation for many high-accuracy instrument recognition systems.

In instrument recognition specifically, Rajesh and Somaiya (2021) extracted MFCC,

Chroma, and CENS (a smoothed chroma) features and applied deep learning (bidirectional LSTM, CNN-LSTM) to achieve ~96–97% accuracy [13]. However, deep nets demand large datasets and computing power. In contrast, more classical models like Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) have also been applied. Prabavathy et al. (2020) compared SVM and KNN on MFCC (and sonogram) features, reporting SVM with MFCC achieved ~98% accuracy on 16 instruments [14]. This aligns with the general observation that SVMs tend to outperform KNN or decision-tree classifiers in high-dimensional audio tasks [15].

The dataset's quality is also a crucial factor in the performance of instrument recognition systems. A well-documented gap exists between the accuracy of models trained on clean, studio-quality recordings versus those applied to "in-the-wild" audio. The former are standard benchmarks precisely because their pristine recording conditions facilitate rigorous evaluation. In contrast, real-world audio introduces confounding variables like room acoustics and background noise, which are known to degrade system performance. For these reasons, using clean, short, professionally recorded samples is a deliberate choice.

Overall, the literature suggests that the efficacy of instrumental sound classification depends on these factors: proper feature engineering (e.g., extracting MFCCs), effective combinations of timbral and pitch-based features, the selection of an appropriate model—including traditional ones—and the use of professionally recorded datasets.

2 Theoretical Framework

This chapter discusses the theoretical foundation for the methods used in this thesis. It begins by examining the acoustic and perceptual properties that distinguish musical instruments, followed by an overview of main audio signal processing techniques for extracting meaningful features from sound. The chapter then introduces the machine learning algorithms employed for instrument classification and discusses the evaluation metrics used to assess their performance. Together, these concepts provide the necessary background for understanding the subsequent methodological choices and experimental results.

2.1 Instrumental Sound Characteristics

Musical instruments are traditionally organized into families according to their mode of sound production. There are several systems (e.g., the Hornbostel–Sachs taxonomy [16], Mahillon's earlier fourfold scheme [17]) widely used that differentiate instruments on the basis of whether sound arises from air, string, membrane, or the body of the instrument. Instruments in the same family often share similar acoustic traits, simplifying their classification in computational contexts. For example, string instruments generally produce rich harmonic spectra, while aerophones are characterized by prominent formant structures, thereby accentuating the acoustic distinction between the two families.

Timbre is the essential perceptual cue that differentiates instruments playing at the same pitch and loudness. This quality, also known as tone color, arises from a sound's unique temporal and spectral characteristics. The temporal envelope is modeled with an ADSR (Attack, Decay, Sustain, Release) curve and it defines how a note's amplitude evolves over time. The attack phase is particularly salient, as represented by the difference between a percussive piano strike and a smooth violin bow. Spectrally, timbre is a function of an instrument's harmonic structure—the specific arrangement and intensity of its fundamental frequency and overtones. This configuration of partials acts as a unique acoustic fingerprint for the instrument.

2.2 Audio Signal Processing

An audio signal is the time-domain representation of sound, essentially the fluctuating air pressure caused by musical vibrations, captured as amplitude vs. time. In

practice, it is digitized at a fixed sampling rate (e.g., 44.1 kHz). To analyze sound, we need to convert this time-domain signal into frequency-domain or time–frequency representations.

The most elementary frequency-domain representation is the spectrum, obtained by applying a Fourier Transform to the waveform. While the waveform directly conveys the temporal evolution of pressure variations, the spectrum reveals the distribution of energy across frequencies, thereby making periodicities, harmonics, and resonances explicit. This distinction is crucial as the waveform reflects how the sound unfolds in time, whereas the spectrum highlights its tonal and timbral makeup.

In this process, a spectrogram plays a crucial role, as it illustrates how the spectral content of an audio signal evolves over time. It is computed by applying a Short-Time Fourier Transform (STFT) to overlapping time frames of the signal, and then plotting the magnitude (or power) of each frequency bin as a heatmap [18]. As an example, Figure 2.1 displays a spectrogram of a bowed violin note, which displays a series of horizontal bands that rise and fall in intensity over time. Transient attacks and decays are also visible as changes in those bands. This time–frequency view makes timbral features more explicit.

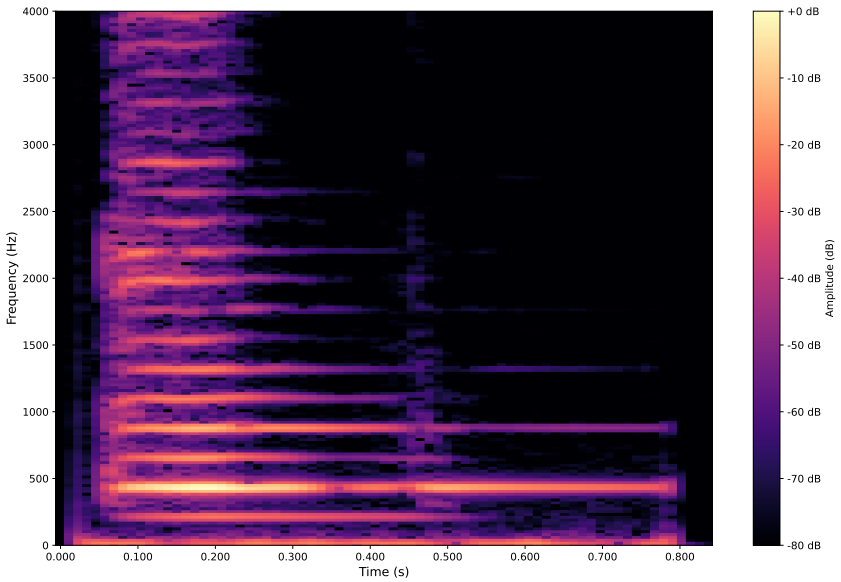

Figure 2.1: Example spectrogram of a violin tone, showing time (horizontal axis) vs. frequency (vertical axis) and intensity as color.

2.2.1 Fourier Transform

To enable effective sound processing, a time-domain signal is often converted into a more tractable representation—the frequency domain. The Fourier Transform (FT) plays a major role in this process by decomposing a complete time signal into its constituent frequency components. For digital applications, the formulation is ex-

pressed as the Discrete Fourier Transform (DFT). However, owing to its computational inefficiency and implementation caveats, a more refined algorithm—the Fast Fourier Transform (FFT)—was devised to compute the DFT efficiently in $O(n \log n)$ time. FFT is fine for stationary signals, but musical sounds are highly non-stationary (notes start and stop, change over time). Therefore, one uses the Short-Time Fourier Transform (STFT). STFT applies a sliding window to isolate short segments of the signal and computes the FT of each segment. The result is a time–frequency representation (the spectrogram) that shows how frequency content changes over time, as illustrated in Figure 2.2, which presents the average magnitude spectrum of a violin tone. [18]

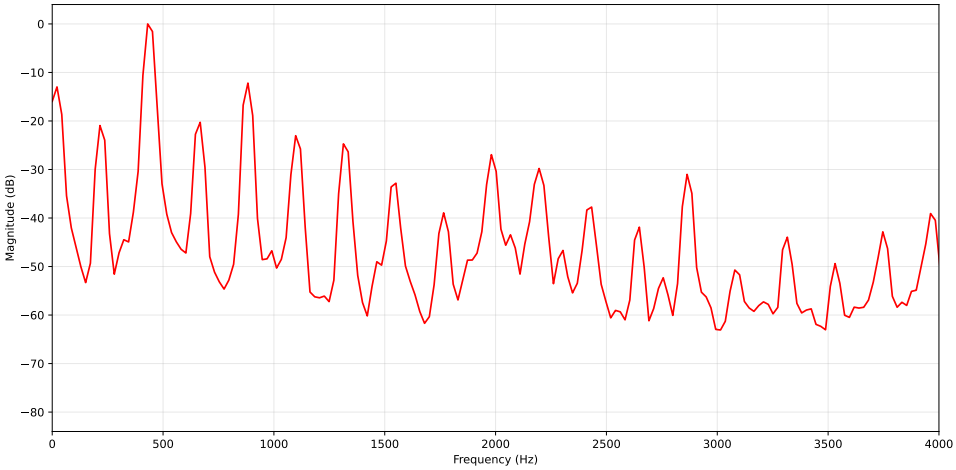

Figure 2.2: Average magnitude spectrum across frequency for the STFT of a violin tone.

2.2.2 Mel Scale

The human ear does not perceive frequency on a linear basis; lower frequencies are perceived with greater sensitivity than higher ones. In their seminal work, Stevens et al. [19] introduced a perceptual unit of pitch in which equal steps correspond to equal perceptual distances as judged by listeners. This unit forms the basis of the mel scale, a perceptual frequency scale that is approximately linear at low frequencies and logarithmic at high frequencies. By definition, a 1000 Hz tone corresponds to 1000 mels, while higher frequencies increase more gradually in mel values. For example, one commonly used formula is:

$$m = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.1}$$

where m is the frequency in mels and f is the frequency in Hz.

The mel scale is frequently used in the form of a mel-spectrogram, where the frequency axis of the spectrogram is warped to the mel scale. Compared with a linear-frequency spectrogram, it provides finer resolution at low frequencies and a compressed view at higher ones, thereby aligning more closely with human auditory perception. Figure 2.3 illustrates this contrast.

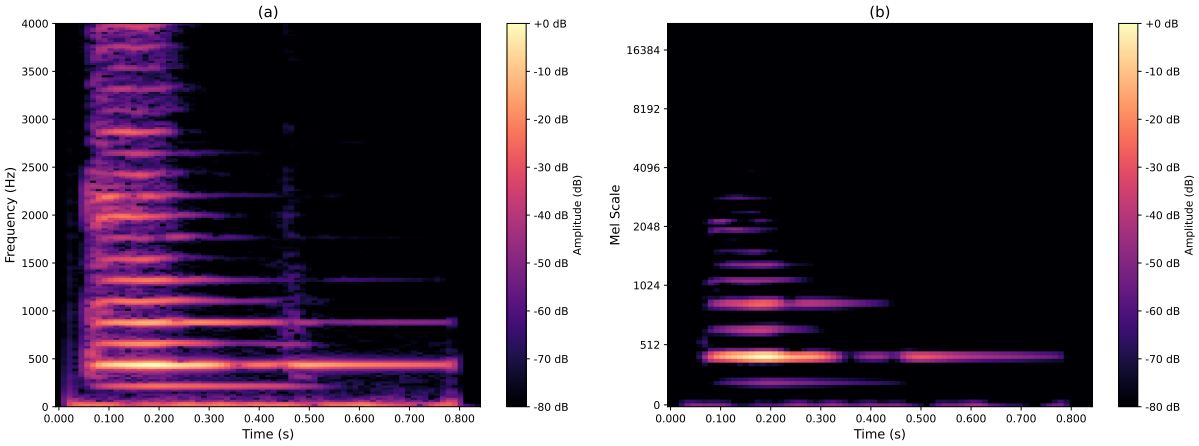

Figure 2.3: (a) Linear-frequency spectrogram in 2.1. (b) Mel-scale spectrogram.

2.2.3 Mel-Frequency Cepstral Coefficients (MFCCs)

Mel-frequency Cepstral Coefficients are a compact representation of an audio spectrum that operates on the mel scale. The process involves taking the Fourier transform of a signal, mapping the powers of the spectrum onto the mel scale, and then taking the discrete cosine transform of the mel log powers. In other words, MFCCs capture the overall shape of the spectrum using the mel scale. The resulting coefficients provide an adequate representation of the short-term power spectrum of a sound which captures its timbral characteristics. Commonly, the first 13-20 coefficients are used as they contain the most perceptually relevant information. Figure 2.4 illustrates this process

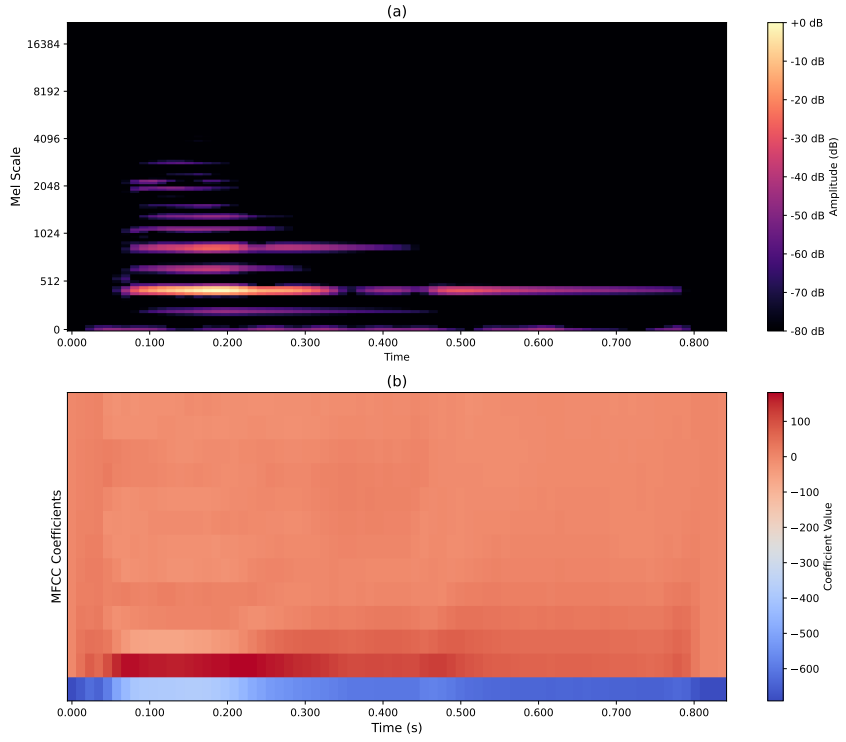

Figure 2.4: (a) Mel-spectrogram of a violin tone. (b) First 13 MFCCs over time derived from the mel-spectrogram.

by showing a mel-spectrogram of a violin tone alongside the first 13 MFCCs extracted over time.

2.2.4 Chroma Features

While MFCCs capture timbre, Chroma features are designed to represent the harmonic and melodic content of an audio signal. A chroma vector is a 12-element feature vector where each element represents the intensity of one of the 12 pitch classes of the musical octave (C, C#, D, etc.), regardless of the octave in which they occur. This makes chroma features particularly useful for tasks such as chord recognition and genre classification. Figure 2.5 illustrates two common approaches: an STFT-based chromagram and a CQT-based chromagram, both of which map pitch class information over time but with slightly different resolutions.

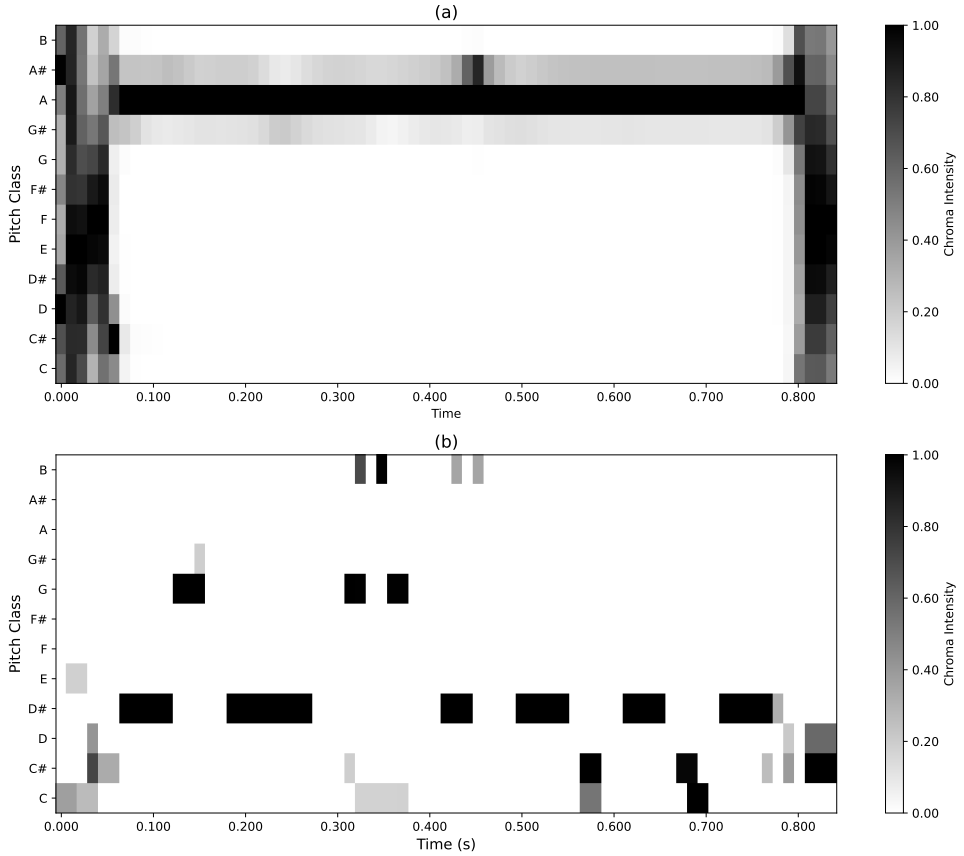

Figure 2.5: (a) STFT-based Chromagram. (b) CQT-based Chromagram.

2.2.5 Spectral Contrast

Spectral contrast measures the difference between peaks and valleys in the spectrum across different frequency bands. It captures the relative distribution of energy in the spectrum, which is important for distinguishing between different types of sounds. High spectral contrast indicates a sound with pronounced harmonic content, while low contrast suggests a more noise-like sound. An example is shown in Figure 2.6,

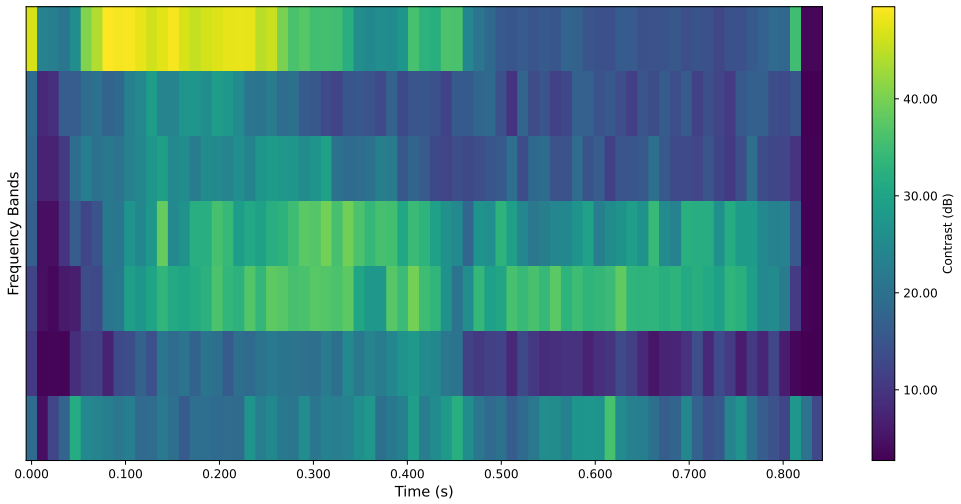

Figure 2.6: Spectral contrast of a violin tone, showing the difference between peaks and valleys in the spectrum across frequency bands.

which depicts the spectral contrast of a violin tone across different frequency regions.

2.3 Classification Algorithms

There is a myriad of ML algorithms available for classification tasks, each with its own strengths and weaknesses. In this section, we discuss a few of the commonly used supervised learning algorithms that are particularly relevant for audio classification tasks.

Supervised learning is a paradigm of machine learning where an algorithm learns from a labeled dataset. In the context of instrument classification, the algorithm is "trained" on a collection of audio samples, each tagged with the name of the instrument that produced it. The goal is for the model to learn a mapping function that can correctly predict the instrument for new, unseen audio samples.

2.3.1 k-Nearest Neighbors (KNN)

K-Nearest Neighbors is a simple yet effective non-parametric algorithm being one of the simplest supervised learning algorithms. It operates on the principle that similar data points are likely to belong to the same class. Given a new data point, KNN identifies the k closest points in the training set (using a distance metric like Euclidean distance) and assigns the most common class among those neighbors to the new point. Despite intuitivity and easy implementation, KNN suffers significantly in high-dimensional spaces, a phenomenon known as the "curse of dimensionality." As the number of features (dimensions) increases, the distance between data points becomes less meaningful, and the volume of the feature space grows exponentially, requiring an infeasibly large amount of data to maintain density.

2.3.2 Support Vector Machines (SVM)

Support Vector Machines are one of the most powerful supervised learning models used for both classification and regression tasks. It is exceptionally well-suited for high-dimensional classification tasks. SVMs work by finding the hyperplane that best separates data points of different classes in a high-dimensional feature space. The optimal hyperplane maximizes the margin between the closest points of each class, known as support vectors.

For data that is not linearly separable, SVMs use the kernel functions. A kernel function implicitly maps the data into a higher-dimensional space where a linear separation becomes possible. The Radial Basis Function (RBF) kernel is a popular choice, as it can handle complex, non-linear relationships between class labels and features. This capability makes SVMs particularly effective for complex audio classification tasks.

2.3.3 Performance Metrics

This subsection presents the standard evaluation metrics for supervised learning, explicates their derivation from the confusion matrix, and assesses their suitability for multiclass instrumental sound classification.

Confusion matrix and basic terms. For a single (binary) class, the confusion matrix entries are defined as follows:

- **True Positive (TP):** correctly predicted positive examples
- **False Positive (FP):** examples incorrectly predicted as positive
- **False Negative (FN):** positive examples incorrectly predicted as negative
- **True Negative (TN):** correctly predicted negative examples

In a multiclass problem, we obtain per-class TP/FP/FN/TN by treating each class in a one-vs-rest way. The confusion matrix is then an $N \times N$ table whose diagonal entries are the correctly classified counts for each class.

Accuracy. Accuracy measures the overall proportion of correctly classified examples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

In multiclass settings this is simply the total number of correct predictions (sum of diagonal entries of the confusion matrix) divided by the total number of examples. Accuracy is intuitive and widely reported, but it can be misleading when classes are imbalanced: a model that always predicts the majority class can achieve high accuracy while failing on minority classes.

Precision and recall provide class-sensitive information and are therefore essential for instrument classification, where some instruments may have fewer samples.

Precision. Precision (also called positive predictive value or confidence) answers: of the examples predicted as a class, what fraction truly belong to that class?

$$Precision = \frac{TP}{TP + FP}$$

Recall. Recall (also called sensitivity or true positive rate) answers: of the true examples of a given class, what fraction were correctly detected?

$$Recall = \frac{TP}{TP + FN}$$

Precision penalizes false alarms (confusing other instruments as another class), while recall penalizes missed detections (failing to detect the given class).

F_1 Score. The F1-score is the harmonic mean of precision and recall and balances the two:

$$F_1\text{-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2\,TP}{2\,TP + FP + FN}$$

These formulas apply directly for binary evaluation and are computed per class in a one-vs-rest setting for multiclass problems. Accuracy is a convenient global summary, but precision, recall, and F_1 provide class-sensitive diagnostics that are essential when class supports differ or when different error types have different costs.

Averaging in multiclass settings. When reporting aggregated precision, recall, and F_1 across classes, three common schemes are used:

- *Macro average:* unweighted mean of per-class metrics (treats all classes equally).
- *Micro average:* compute global TP/FP/FN by summing over classes and then apply the metric (weights classes by support; for multiclass single-label, micro-precision = micro-recall = global accuracy).
- *Weighted average:* per-class metrics averaged with weights equal to each class's support (useful when accounting for class imbalance).

3 Practical Application

This chapter details the practical implementation of the theoretical concepts discussed earlier. It outlines the steps taken to preprocess the audio data, extract relevant features, and train machine learning models for instrument classification. The section begins with a description of the dataset used, followed by the specific preprocessing techniques applied to enhance data quality. Next, it delves into the feature extraction methods employed to capture the distinctive characteristics of different instruments. The chapter then discusses the selection and configuration of various machine learning algorithms, including hyperparameter tuning and model evaluation strategies. Finally, it presents the results of the experiments conducted, highlighting key findings and insights gained from the practical application of these methods.

3.1 Dataset

This project utilizes the Philharmonia Sound Samples Library [7], a comprehensive

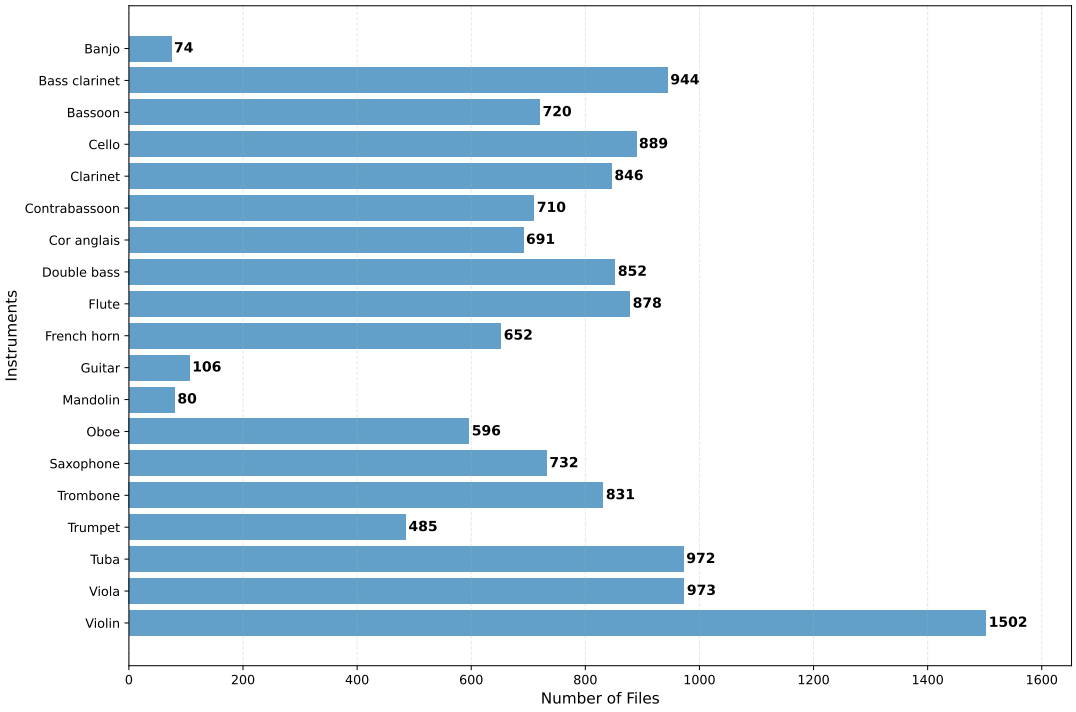

Figure 3.1: Distribution of audio samples per instrument class in the dataset.

collection of instrumental sound samples recorded by members of the Philharmonia Orchestra in the UK. The dataset represents audio recordings from 20 different instruments, covering standard orchestral instruments such as strings, brass, and woodwinds, as well as non-traditional instruments like guitar, mandolin, and banjo. These samples are versatile and suitable for creating music in a variety of styles. The dataset comprises 13,683 labeled audio samples, making it structured and well-organized for machine learning purposes. Each sample is labeled with its corresponding instrument class, facilitating supervised learning tasks. After initial cleaning and removal of percussion samples (since they contained limited number of samples lacking enough training information) we were left with 13,533 samples from 19 instruments. Each sample had an average duration of 1.91 seconds, with a sampling rate of 44.1 kHz. The distribution of audio samples across instrument classes is illustrated in Figure 3.1, which highlights the inherent class imbalance that the classification models must address.

3.2 Feature Extraction

Feature extraction was conducted using the `librosa` (version 0.11.0) library [20], a widely used Python toolkit for music and audio analysis, to create a 32-dimensional feature matrix, which includes only the most crucial sound characteristics. This matrix consists of 13 Mel-Frequency Cepstral Coefficients (MFCCs), 7 Chroma features, and 12 Spectral Contrast features, combining to form the 32 features in total.

The 13 MFCCs are a standard and effective choice because they are designed to mimic the way humans perceive sound [21]. The first coefficient represents the overall energy or loudness of the sound, while the next 12 capture the essential shape of the spectral envelope, which helps define its unique timbre or tonal quality. Using more than this often just adds noise without providing significant new information.

The 7 Chroma features relate to the musical notes being played. While a full musical octave contains 12 distinct notes (e.g., C, C#, D, etc.), using 7 features simplifies this to the seven notes of a standard major or minor scale (the familiar "do-re-mi" scale). This is a common simplification for analyzing music that is primarily tonal.

The final feature extraction method, Spectral Contrast, uses 12 features to measure the texture of the sound, done by splitting the sound's frequency spectrum into 6 different bands, from low to high. This contrast between peaks and valleys across 6 bands provides the 12 features (6 bands $\times$ 2 values), providing a good summary of the sound's harmonic and noise characteristics.

The mean of each feature across the time series of the audio sample was computed, resulting in a single value for each of the 32 features. These 32 mean values were concatenated to form a single feature vector for each audio sample. The entire feature matrix was then normalized using `StandardScaler` from `scikit-learn`. This process

scales the features to have a mean of 0 and a standard deviation of 1, which assures that no single feature dominates the learning process due to its scale.

3.3 Model Selection and Initial Training

As outlined in the literature review, Support Vector Machines (SVM) are robust algorithms due to their ability to efficiently handle high-dimensional data, which justifies their selection for this experiment. The 32-dimensional feature space constructed in this study is both dense and complex. SVMs, particularly with the Radial Basis Function (RBF) kernel, are well-suited for identifying non-linear decision boundaries in such spaces, making them a more theoretically sound choice compared to algorithms like k-Nearest Neighbors (KNN), which can struggle with the curse of dimensionality.

The primary hyperparameters of the RBF kernel are the regularization constant C, which controls the trade-off between maximizing the margin and minimizing training error, and the RBF kernel width γ, which determines the influence range of each training example. Proper tuning of these parameters is essential: excessively large values of C and γ can lead to overfitting, while values that are too small may result in underfitting.

To train and evaluate the model effectively, the dataset of 13,533 labeled audio samples was split into training and testing sets using an 80:20 ratio. We employed `StratifiedShuffleSplit` to make sure that the class distribution of instruments is maintained in both sets. Specifically, the training set (80%) consists of 10,826 samples to train the model, while the testing set (20%) comprises 2,707 samples reserved for evaluating the model's performance. The latter set was not seen during training and provides an unbiased measure of the model's accuracy.

An initial SVM model was trained on the combined 32-dimensional feature set using default `scikit-learn` parameters. This served as a baseline to quantify the impact of subsequent optimization steps. This initial, unoptimized model achieved a respectable accuracy of 95%.

3.4 Hyperparameter Fine-Tuning

With such a strong baseline, the next step was hyperparameter tuning to further enhance the model's performance. To this end, an extensive grid search was conducted to identify the optimal configuration of SVM parameters. Grid search performs an exhaustive evaluation over a predefined parameter space, ensuring that the best-performing combination within the specified ranges is identified, albeit at a considerable computational cost.

The search space included candidate values for the regularization parameter C (e.g., 0.1, 1, 10, 100), the kernel coefficient γ (e.g., 0.001, 0.01, 0.1, 1), the polynomial degree for

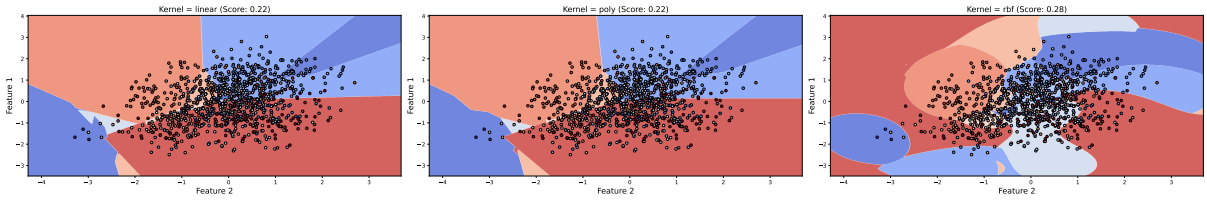

Figure 3.2: SVM decision boundaries and scores for different kernel types on 1000 samples.

non-linear kernels, and multiple kernel types (linear, radial basis function (RBF), and polynomial). To provide intuition about the role of these hyperparameters, Figure 3.2 visualizes the decision boundaries and corresponding accuracy scores obtained with different kernel types on a simplified two-dimensional dataset of 1,000 samples. As the figure illustrates, linear and polynomial kernels struggle to separate the classes in this reduced space, whereas the RBF kernel captures the non-linear boundary more effectively, yielding a higher score. This visualization underscores the sensitivity of SVM performance to kernel choice and parameter scaling, thereby justifying a systematic search for optimal hyperparameter settings.

3.5 Results and Analysis

The optimized SVM achieved 98.9% accuracy on the held-out test set (2,707 clips). This means only about 1.1%, or approximately 30 clips, were misclassified overall. Precision, recall, and F1-score averaged across classes were similarly high, around 98.9%, due to the low number of errors that were evenly distributed. Table 3.1 below summarizes the metrics for each feature set, comparing the use of only MFCC to the inclusion of additional features. Notably, using MFCC alone resulted in an accuracy of about 91.8%, and the addition of chroma and contrast features increased the accuracy to approximately 98.9%.

Metrics (%)	**MFCC**	**Chroma**	**Contrast**	**All**
Accuracy	91.80	28.41	64.76	98.89
Precision	91.81	30.53	64.57	98.91
Recall	91.80	28.41	64.76	98.89
F1-score	91.71	25.63	64.20	98.89

Table 3.1: Comparison of classification metrics for different feature sets.

When comparing results by feature, we observe that MFCCs contribute the most to overall accuracy. On its own, the chroma feature performed poorly, which is not surprising since pitch-class information does not effectively differentiate between instruments. The spectral contrast feature demonstrated moderate performance with an accuracy of 64%, indicating that it captures some differences but lacks sufficient

effectiveness on its own.

The best results come from combining these features: MFCCs provide timbral discrimination [15], chroma adds complementary harmonic cues [12], and spectral contrast introduces textural variance. The incremental accuracy gain of approximately 7% from including chroma and contrast alongside MFCCs reflects the concept of diminishing returns [22]. Once the primary variance is captured through MFCCs, adding more related features results in smaller improvements in accuracy.

A confusion matrix (see Figure 3.3) illustrates the points of confusion among different instruments. The majority of the diagonal entries are nearly perfect (indicated by dark shading), which shows that nearly every instrument was recognized correctly. The few off-diagonal entries represent confusions among acoustically similar instruments. For example, some clips of the double bass were misidentified as viola or contrabassoon clips, and contrabassoon clips were confused with those of the cor anglais or flute.

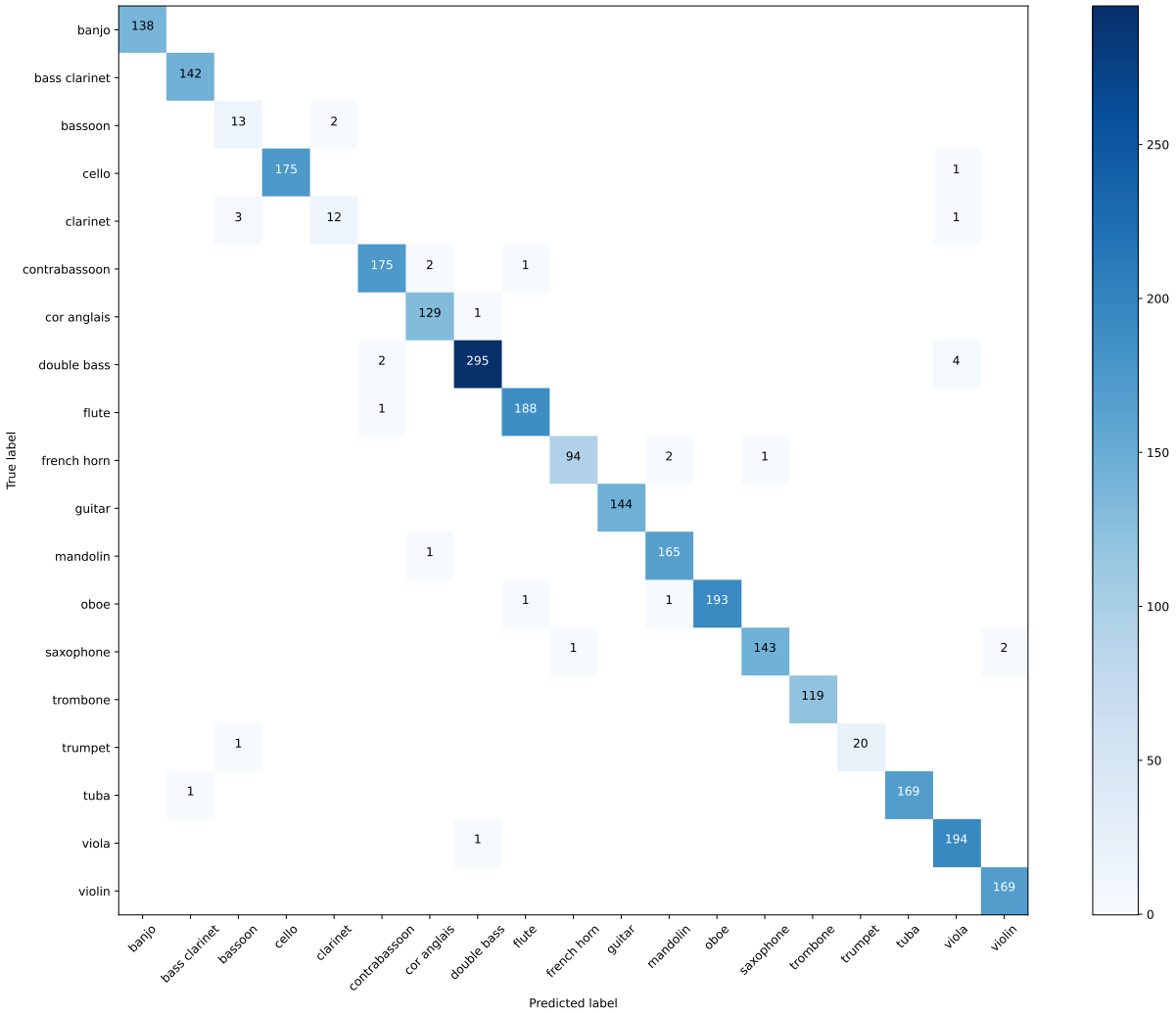

Figure 3.3: Confusion matrix for the SVM model on the test set.

Table 3.2 reports the per-class evaluation metrics. Overall, most instruments

achieved precision, recall, and F1-scores close to or above 0.98, demonstrating consistently high classification accuracy across the dataset. Several instruments, such as banjo, guitar, trombone, and tuba, reached perfect or near-perfect scores. Likewise, strings (e.g., violin, viola, cello) and woodwinds such as flute and oboe performed exceptionally well, with F1-scores above 0.98.

The most challenging cases were instruments with relatively few training samples, such as bassoon (15 clips, F1 = 0.81) and clarinet (16 clips, F1 = 0.80), where the limited data likely constrained performance. Despite these exceptions, even the weakest results remained reasonably high, and all other instruments achieved F1-scores above 0.95, underscoring the robustness of the optimized SVM classifier.

Instrument	**Precision**	**Recall**	**F1-score**	**Support**
banjo	1.00	1.00	1.00	138
bass clarinet	0.99	1.00	1.00	142
bassoon	0.76	0.87	0.81	15
cello	1.00	0.99	1.00	176
clarinet	0.86	0.75	0.80	16
contrabassoon	0.98	0.98	0.98	178
cor anglais	0.98	0.99	0.98	130
double bass	0.99	0.98	0.99	301
flute	0.99	0.99	0.99	189
french horn	0.99	0.97	0.98	97
guitar	1.00	1.00	1.00	144
mandolin	0.98	0.99	0.99	166
oboe	1.00	0.99	0.99	195
saxophone	0.99	0.98	0.99	146
trombone	1.00	1.00	1.00	119
trumpet	1.00	0.95	0.98	21
tuba	1.00	0.99	1.00	170
viola	0.97	0.99	0.98	195
violin	0.99	1.00	0.99	169

Table 3.2: Precision, recall, F1-score, and support (number of samples) for each instrument class.

A qualitative analysis of the misclassified samples provides insight into the model's limitations and the inherent difficulty of the task. The majority of errors occurred between instruments within the same Hornbostel–Sachs family or with broadly similar timbral characteristics. For instance, woodwinds such as bassoon and clarinet were occasionally confused, as were contrabassoon and cor anglais. At the same time, a few unexpected cross-family confusions appeared, such as french horn being misclassified as mandolin or saxophone as violin. These rare cases suggest that the model can sometimes be misled by overlapping spectral or dynamic features rather than structural similarities. Overall, however, the error patterns indicate that the classifier successfully

captured both family-level distinctions and finer-grained timbral cues, with residual confusion limited to cases that are inherently challenging even for human listeners.

3.6 Implementation and Outputs

The implementation of the experiment, including the outputs, can be found in the following Google Colab notebook:

`https://colab.research.google.com/drive/1AE-WRIjaDCYyTjUCmf6tsvTOfwBA2Tmx?usp=sharing`

Discussion

The results compellingly demonstrate that a carefully constructed feature space, paired with a well-tuned SVM, is a highly effective strategy for instrumental sound classification. The synergy between MFCC, Chroma, and Spectral Contrast features was the key to efficacy. Each modality captures a different facet of the sound—timbre, harmony, and texture, respectively. By combining them, we provide the SVM with a rich, multi-faceted representation that enables it to discern the subtle differences between instruments with very high precision.

This study illustrates the economic principle of diminishing returns in the context of machine learning [22]. The transition from the baseline model to the final optimized version reveals two distinct phases of improvement.

- **Feature Engineering:** In the first phase, a significant increase in performance was achieved during feature engineering. By expanding the feature space from a 13-dimensional MFCC vector to a 32-dimensional combined matrix, the model's accuracy improved by 3.2%.
- **Hyperparameter Optimization:** In the second phase, meticulous hyperparameter optimization proved to be even more beneficial, resulting in an additional 4% increase in accuracy.

While both steps were crucial, they suggest that after a certain point of feature complexity, detailed optimization can yield even greater returns than simply adding more features.

The main strength of this work is its impressive classification accuracy of 98.89%, highlighting the effectiveness of the chosen methodology. Additionally, the study emphasizes the efficiency of classical machine learning, demonstrating that a less computationally intensive model, such as an SVM, can achieve results comparable to more complex deep learning architectures. This conclusion is backed by a robust evaluation framework that used StratifiedShuffleSplit cross-validation to ensure the reported metrics are both fair and reliable.

Nonetheless, it is crucial to acknowledge the study's limitations to properly contextualize the findings. The model was trained and evaluated exclusively on high-quality, monophonic studio recordings; its performance would likely degrade when confronted with real-world challenges such as background noise, reverberation, or polyphonic

textures. Finally, despite the use of stratification, the inherent class imbalance in the dataset could introduce a subtle bias toward more frequently represented instruments.

A logical next step is to investigate dimensionality reduction techniques, such as Principal Component Analysis (PCA), on the 32-dimensional feature set. This could help reduce computational complexity without significantly sacrificing accuracy. To advance towards practical application, it is essential to test the model's robustness on noisy, real-world recordings and complex polyphonic music. Lastly, using more advanced feature selection algorithms could help in identifying the most impactful subset of the 32 features, potentially leading to a more streamlined and efficient classification model.

Conclusion

This thesis presents a comprehensive study on the classification of instrumental sounds using a Support Vector Machine (SVM). By extracting a 32-dimensional feature vector that combines Mel-frequency cepstral coefficients (MFCC), Chroma, and Spectral Contrast features from a dataset of 13,533 samples, the research achieved a remarkable classification accuracy of 98.89% through meticulous hyperparameter tuning of the SVM.

The analysis highlights the synergistic benefits of combining diverse feature types and emphasizes the critical importance of hyperparameter optimization. The central conclusion of this work is that a well-thought-out, feature-rich approach utilizing a classic machine learning algorithm can serve as an exceptionally powerful tool for audio classification. This challenges the notion that state-of-the-art results are solely the domain of deep learning.

For problems with well-defined characteristics and moderately sized datasets, the combination of sophisticated feature engineering and a robust, well-tuned model like the SVM offers an effective and computationally accessible path to high performance.

This study underscores a fundamental principle in machine learning: the quality of data representation is as crucial as the complexity of the model itself. By focusing on creating a feature space that accurately captures the essence of instrumental sounds, we enabled a relatively simple algorithm to achieve extraordinary results. This work contributes to the field of Music Information Retrieval by providing a strong benchmark and a clear demonstration of the enduring power of Support Vector Machines in the acoustic domain.

Bibliography

[1] J. M. Alyza, F. S. Utomo, Y. Purwati, B. A. Kusuma, and M. S. Azmi, "Music recommendation system based on cosine similarity and supervised genre classification," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, p. 77–80, Aug. 2023.

[2] A. Klapuri, *Introduction to Music Transcription*, pp. 3–20. Boston, MA: Springer US, 2006.

[3] B. Kostek, "Musical instrument classification and duet analysis employing music information retrieval techniques," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 712–729, 2004.

[4] J. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, pp. 74–99, 2015.

[5] S. K. Zieliński, H. Lee, P. Antoniuk, and O. Dadan, "A comparison of human against machine-classification of spatial audio scenes in binaural recordings of music," *Applied Sciences*, vol. 10, no. 17, 2020.

[6] J. Hestness, N. Ardalani, and G. Diamos, "Beyond human-level accuracy: Computational challenges in deep learning," 2019.

[7] "Sound samples | philharmonia." Online. Accessed: Jan. 20, 2025.

[8] M. Bürgel and K. Siedenburg, "Impact of interference on vocal and instrument recognition," *The Journal of the Acoustical Society of America*, vol. 156, no. 2, pp. 922–938, 2024.

[9] R. Elliott, "Simple visual and simple auditory reaction time: A comparison," *Psychonomic Science*, vol. 10, no. 10, pp. 335–336, 1968.

[10] G. K. K, B. Chatterjee, and S. Sen, "Cs-audio: A 16 pj/b 0.1–15 mbps compressive sensing ic with dwt sparsifier for audio-ar," *IEEE Journal of Solid-State Circuits*, vol. 57, pp. 2220–2235, 2022.

[11] G. Giri and M. Radhitya, "Musical instrument classification using audio features and convolutional neural network," *Journal of Applied Informatics and Computing*, vol. 8, pp. 226–234, 07 2024.

[12] D. P. Ellis, "Classifying music audio with timbral and chroma features.," pp. 339–340, 01 2007.

[13] S. Rajesh and N. N. J., "Recognition of musical instrument using deep learning techniques," *International Journal of Information Retrieval Research (IJIRR)*, vol. 11, pp. 41–60, October 2021.

[14] S. Prabavathy, V. Rathikarani, and P. Dhanalakshmi, "Classification of musical instruments using svm and knn," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 7, pp. 1186–1190, 2020.

[15] J. Chulev, "Improving musical instrument classification with advanced machine learning techniques," *arXiv preprint arXiv:2411.00275*, 2024.

[16] E. M. v. Hornbostel and C. Sachs, "Systematik der musikinstrumente: Ein versuch," *Zeitschrift für Ethnologie*, vol. 46, no. 4–5, pp. 553–590, 1914.

[17] V.-C. Mahillon, "Catalogue descriptif et analytique du musée instrumental du conservatoire royal de musique de bruxelles, précédé d'un essai de classification méthodique de tous les instruments anciens et modernes," 1880. Instrument catalog and classification scheme that inspired Hornbostel–Sachs.

[18] E. Mukhtarov, "Time–frequency fourier methods for live audio noise reduction," in *Matematyka – od zastosowań oczywistych do zdumiewających* (G. Gwóźdź-Łukawska, ed.), (Stefanowskiego 18/22 90-924 Łódź), Politechnika Łódzka, Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki, Centrum Nauczania Matematyki i Fizyki, 2025. Forthcoming.

[19] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.

[20] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," pp. 18–24, 01 2015.

[21] R. Ahuja, V. Solanki, V. Khullar, and L. Kumar, "Classification of non-speech sound signals: An approach of machine learning with mfcc feature extraction," *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, vol. 1, pp. 1–5, 2024.

[22] O. Sadeghi, *The Diminishing Returns (DR) Property and Its Applications in Machine Learning*. University of Washington, 2023.