FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Computational Methods for Style Identification using Tonal Descriptions from Audio Recordings

Francisco Almeida

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Gilberto Bernardes

March 7, 2022

Computational Methods for Style Identification using Tonal Descriptions from Audio Recordings

Francisco Almeida

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Rui Rodrigues
Referee: Dr. Christof Weiß
Referee: Prof. Gilberto Bernardes

March 7, 2022

Abstract

Digital technologies have changed the way people consume music. Of note, the very large collections of musical audio available in online streaming services. Manually browsing these large collections would be infeasible due to the amount of time it requires. In this context, several solutions have been developed in the field of Music Information Retrieval to automatically organize collections of musical audio files according to different semantic categories. Musical style is a salient quality of such categories. Style refers to musical aspects such as historical periods, composers, performers, sonic texture, emotion, and genre.

Most of the work regarding style identification focuses on low-level and short-term attributes that ignore the horizontal dimension of the harmonic content and lack a perceptual basis. In this context, we adopt the perceptually-inspired Tonal Interval Space for computing descriptors of dissonance, chromaticity, dyadicity, triadicity, diminished-quality, diatonicity, and whole-toneness. Additionally, we propose a novel set of tonal audio features based on this pitch space that capture long-term structural harmonic relationships, namely, Euclidean and cosine distance between consecutive audio frames, Euclidean and cosine tonal dispersion, harmonic change peak interval and magnitude, and entropy. Furthermore, we present a new audio segmentation approach based on harmonic changes.

Using the above set of features, we developed a musical style identification model and performed classical style period and composer classification experiments by comparing them with state-of-the-art literature. We first evaluated the correlation between our features and those proposed in the state of the art and concluded the former capture complementary and meaningful tonal properties. In addition, we compared the proposed harmonic structural audio segmentation approach with a fixed-time segmentation strategy on multiple temporal resolutions in the context of musical style identification and concluded that the former performed best in most test cases, but is more computationally expensive. In a subsequent analysis, we performed style period and composer classification using a Support Vector Machine classifier on five different audio datasets by considering multiple combinations of features. By applying a filtering method that reduces the chance of model overfitting, our set of features improved the classification accuracy on four out of five datasets, with an increase of up to 4.74%. Without applying any filtering, we improved the accuracy score on two of the datasets by 1.98% and 1.64%. These results suggest that our set of features introduce performance benefits in real case classification scenarios.

Keywords: Music, Music Information Retrieval, Music Style Identification, Audio Descriptors, Tonal Interval Space, Classification

Resumo

As tecnologias digitais mudaram o modo como as pessoas ouvem música. De notar, as grandes coleções de áudio disponíveis em serviços de *streaming online*. Navegar estas grandes coleções manualmente seria inviável devido ao tempo que seria requerido. Neste contexto, foram desenvolvidas várias soluções no ramo de Recuperação de Informação Musical para automaticamente organizar coleções de ficheiros de áudio musical de acordo com diferentes categorias semânticas. O estilo musical é uma qualidade proeminente dessas categorias, podendo referir-se a aspetos como períodos históricos, compositores, artistas, textura sonora, sentimento e género.

Na sua maioria, o trabalho realizado em identificação de estilo foca-se em atributos de baixo nível e curto prazo que ignoram a dimensão horizontal do conteúdo harmónico e não possuem nenhuma base percetual. Neste contexto, adotamos o Espaço Tonal Intervalar para calcular descritores de dissonância, cromaticidade, diadicidade, triadicidade, qualidade diminuta, diatonicidade, e qualidade de tons inteiros. Adicionalmente, propomos um novo conjunto de atributos harmónicos baseado neste espaço que captam relações harmónicas a longo prazo, nomeadamente, distância Euclidiana e angular entre *frames* de áudio consecutivas, dispersão tonal Euclidiana e angular, intervalo entre picos de mudanças harmónicas e magnitude desses picos, e entropia. Além disso, apresentamos uma nova abordagem de segmentação de áudio baseada em mudanças harmónicas.

Utilizando o conjunto de atributos supramencionados, desenvolvemos um modelo de identificação de estilo musical e realizámos experiências de classificação por período estilístico e compositores clássicos, comparando-os com outros descritores em literatura de estado da arte. Primeiramente, avaliámos a correlação entre os nossos atributos e os propostos no estado da arte e concluímos que os nossos captam propriedades harmónicas complementares e significativas. Adicionalmente, comparámos a abordagem de segmentação harmónica estrutural com uma estratégia de segmentação temporal fixa com várias resoluções no contexto de identificação de estilo musical e concluímos que a primeira estratégia tem melhor desempenho na maioria dos casos de teste, mas é menos eficiente em termos computacionais. Numa análise seguinte, realizámos classificação por período estilístico e compositor utilizando o algoritmo *Support Vector Machine* em cinco conjuntos de dados de áudio considerando múltiplas combinações de atributos. Aplicando um método de filtragem que reduz a hipótese de sobreajuste do modelo, o nosso conjunto de atributos melhorou a precisão de classificação em quatro dos cinco conjuntos de dados, com melhorias até 4.74%. Sem aplicar qualquer tipo de filtragem, melhorámos a precisão em dois dos conjuntos de dados em 1.98% e 1.64%. Estes resultados sugerem que o nosso conjunto de atributos introduz benefícios de desempenho em cenários reais de classificação.

Keywords: Música, Recuperação de Informação Musical, Identificação de Estilo Musical, Descritores de Áudio, Espaço Tonal Intervalar, Classificação

Acknowledgements

Writing this dissertation was undoubtedly one of the most challenging undertakings I have ever attempted during my life. As I look back on the past year, I realize I could not have accomplished what I have (for what it's worth) without the help of those around me. It is for that reason that I now dedicate the remainder of my energy to writing this small section where I thank the people I owe the most to.

First of all, I express my sincerest gratitude to my supervisor, professor Gilberto Bernardes, for his infinite patience and knowledge, without which I could not have finished this document. No matter the question I had, he was always readily available to help and guide me through solving whichever problems I was facing.

I want to thank two researchers I had the opportunity to talk to who helped me solve various roadblocks. First, I thank Dr. Christof Weiß who I contacted multiple times, for explaining the finer details of his work, which was crucial for writing this dissertation. Second, I thank Pedro Ramoneda, who explained several of the more intricate processes in one of his research articles and made himself available to experiment with different approaches that could help improve our results.

I thank my colleagues of the Sound and Music Computing lab for their valuable input and feedback during meetings that helped steer the research direction of this dissertation.

As the years go by, I also feel progressively more fortunate for having the best friends I could have asked for. Even if unknowingly, you have helped me more than you could have ever imagined just by being around and spending time with me. I'm still not sure if I deserve you all, but do know that I am eternally grateful for having you as such a significant part of my life.

Finally, a special thanks to my parents, brother, and grandparents, for following me closely during this journey (and my entire life, really). You have always supported me and given me advice when I needed it the most, and I would not be here today without your help.

Francisco Almeida

"To achieve great things, two things are needed;
a plan, and not quite enough time."

Leonard Bernstein

Contents

List of Figures

List of Tables

Abbreviations

CQT	Constant-Q Transform
DCNN	Deep Convolutional Neural Network
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HCDF	Harmonic Change Detection Function
KNN	K-nearest Neighbors
LDA	Linear Discriminant Analysis
MFCC	Mel-frequency Cepstral Coefficient
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation Exchange
NNLS	Non-negative Least Squares
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RF	Random Forest
STFT	Short-time Fourier Transform
TIS	Tonal Interval Space
TIV	Tonal Interval Vector

Chapter 1

Introduction

1.1 Context and Motivation

The way people listen to music has changed markedly over the last decades. We have witnessed a shift from the consumption of music in physical media, such as vinyl and CDs, to digital streaming services that provide access to large online music collections. While physical formats are constrained by the availability of materials required to produce them, digital content is generally easier to create, making it more widely available [43]. Therefore, methods for organizing such large collections are fundamental to allow users fluid navigation and retrieval of musical contents. A common problem in the field of Music Information Retrieval (MIR) is the classification of songs or pieces according to specific categories so that they can later be retrieved accordingly. A typical approach is to perform such categorization according to style, one of the most salient qualities of music [4] which might refer to:

- The stylistic traits associated with historical periods, for example, the aspects that allow one to distinguish between a romantic and a baroque piece.
- The style of a certain composer.
- Styles associated with performers. Even when performing the same piece, each performer will interpret it in their own way. This is particularly true when improvisation is involved.
- Aspects of sonic texture such as melody, rhythm, harmony, and timbre.
- The emotions or mood music might transmit to the listener, e.g., scary, calm, happy, sad.
- Musical genre, e.g., rock, pop, jazz.

The above definitions imply the existence of characteristics in the musical structure that lead listeners to identify and associate a musical work with a given composer, historical period, or genre [17].

Most studies regarding the computational identification of musical style mainly focus on low-level and short-term attributes that do not account for the hierarchical structure of harmony. This approach captures mostly the vertical aggregates of harmony, such as the attributes of chords, and does not consider the transitions and long-term dependencies between them. However, harmonic progressions and phrase or formal structures are fundamental to musical composition and enforce stylistic traits [4]. Moreover, the audio segmentation strategies employed to extract these attributes are usually unaware of the harmonic structure. Lastly, a significant part of the audio features proposed in literature for style identification have a poor perceptual basis and do not account for the cognitive distances between harmonic elements, such as intervals, chords, or keys.

In this context, we adopt the Tonal Interval Space (TIS) proposed by Bernardes et al. [7] as the basis for perceptually-inspired tonal description. This pitch space maps individual pitches, intervals, and chords into a 12-dimensional vector space where distances between these elements in the space relate to their perceived proximity. The TIS provides perceptual descriptors of dissonance, chromaticity, dyadicity, triadicity, diminished-quality, diatonicity, and whole-toneness. Additionally, we propose further descriptors based on this pitch space and experiment with different segmentation strategies to capture the long-term harmonic relationships, aiming to improve the state of the art in style identification models.

1.2 Objectives and Methodology

By considering the current limitations of the style identification models described in literature, we define the dissertation's objectives as follows:

- To propose a novel set of higher-level perceptually-inspired tonal audio descriptors based on the TIS.
- To build a new musical style identification model that outperforms the current state of the art by using this set of descriptors.
- To verify if the use of the new audio descriptors improves style identification.
- To evaluate the performance of the proposed model when using different types of audio segmentation in the feature extraction process.

Using the Python programming language, we implement the computation of the newly proposed tonal features. Additionally, we implement the state-of-the-art tonal descriptors for later comparison, as their implementation is not publicly available. To conduct further analysis, we develop a system that can handle the calculation of these descriptors for large audio datasets. This process needs to be as efficient as possible in order to deal with large amounts of audio data. The system must also be able to combine different sets of features so that we might test their influence on style identification performance. Next, we implement the current state-of-the-art musical style identification model, which we adapt to the novel audio features. Finally, we add evaluation functionality to the system to calculate appropriate performance metrics for further analysis.

1.3 Dissertation Structure

This dissertation is structured as follows: Chapter 2 overviews current musical style identification models and presents relevant techniques for the tonal description of musical audio signals. Chapter 3 explores and proposes new tonal features in a perceptually-inspired tonal pitch space. Chapter 4 proposes and describes a system for musical style identification. Chapter 5 details several classical style period and composer classification experiments using harmonic audio features. Finally, Chapter 6 highlights the main contributions of this dissertation and suggests improvements and related future work.

Chapter 2

Modeling and Identifying Musical Style

In this chapter, we provide an overview of computational models for the identification of musical style. In Section 2.1, we refer to existing architectures of style identification and find examples for each one of them in current literature. Section 2.2 presents MIR techniques for the tonal description of audio signals frequently used in related topics. Finally, Section 2.3 concludes by highlighting the most significant limitations of current style identification models in literature.

2.1 Computational Models for Style Identification

Musical style identification models aim at identifying certain characteristics of style within pieces. The diagram in Figure 2.1 describes the typical structure of such models. Note that the Feature Extraction and Dimensionality Reduction processes are not always present.

Training requires either a labeled or unlabeled dataset, compiled directly from raw audio or symbolic data such as MIDI or from low-level features extracted from those sources of musical information. The data might then go through the process of Feature Extraction, which happens either through a Feature Engineering or Feature Learning approach.

The Feature Engineering approach involves manually modeling new higher-level features from those present in the initial dataset. Feature Engineering is defined as "the practice of constructing suitable features from given features that lead to improved predictive performance [...] usually conducted by a data scientist relying on their domain expertise and iterative trial and error and model evaluation" [41, p. 2529]. Therefore, it is entirely up to the scientist to design the new features they consider relevant. This process may also involve feature selection, meaning one may choose to discard certain features.

The Feature Learning approach relies on deep learning techniques to automatically create higher-level audio features from those in the initial dataset. Instead of creating new features manually, a deep neural network is typically fed with either audio or symbolic data directly or low-level

features and is then capable of identifying underlying patterns in the data to create new features automatically.

The audio features resulting from the Feature Extraction process may or may not go through a Dimensionality Reduction module and finally through the Machine Learning component, which represents the training process, typically, of either a classification or clustering model.

The following subsections in this section are organized according to the components depicted in Figure 2.1 to which we add a subsection concerning methods for evaluating style identification models.

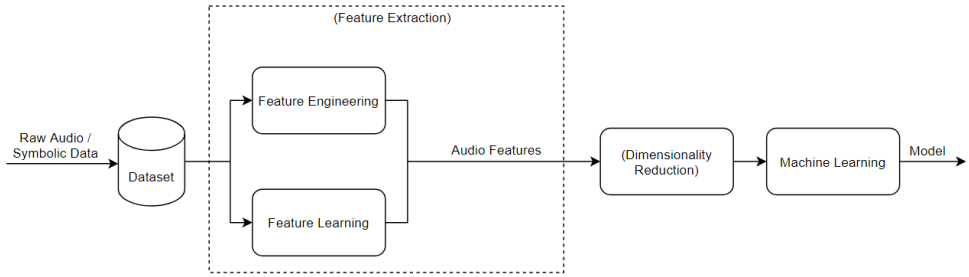


Figure 2.1: Typical architecture of a musical style identification model. The components that have their name enclosed in parentheses are not necessarily present.

2.1.1 Datasets

Music style identification models use a wide range of datasets for training and evaluation. These datasets typically consist of audio or MIDI files or a set of low-level features extracted from one of these formats. In literature, many works utilize previously compiled public datasets. However, some models are trained and tested with datasets built specifically in the context of those works. Table 2.1 compiles a list of publicly available datasets used in the context of musical style identification.

The majority of publicly available datasets[1] encompass a wide variety of genres and do not focus on a particular type of music. Comparatively, datasets focusing exclusively on Western classical music, in particular, are few and generally much smaller in size.

2.1.2 Machine Learning

The Machine Learning module involves training the model for a specific task. In musical style identification, this usually consists of either classification, a supervised learning task, or clustering, a type of unsupervised learning problem.

[1] https://ismir.net/resources/datasets/

Table 2.1: Relevant publicly available datasets used in training and evaluation of musical style identification models.

Dataset	**Classes**	**Contents**
ISMIR2004Genre [15]	Classical, Electronic, Jazz/Blues, Metal/Punk, Rock/Pop, World	729 excerpts (30s)
GTZAN [53]	Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, Rock	1000 excerpts (30s)
AcousticBrainz-Genre [12]	15-31 genres with 265-745 sub-genres across 4 datasets	Audio features for about 2 million songs
RWC [25, 35, 24]	12 genres with 40 sub-genres across 4 datasets	115 pop songs, 50 classical, 50 jazz, 100 various
FMA [18]	161 genres	106574 songs
Ballroom [26]	8 ballroom genres	698 excerpts (30s)
Extended Ballroom [36]	13 ballroom genres	4180 excerpts (30s)
uspop2002 [6]	400 pop artists across 251 styles	MFCCs of 8752 songs
Million Song Dataset [10]	Over 13 genres	Audio features for 1 million songs
SMD Western Music [40]	21 classical composers across 22 types of instrumentation	200 recordings
MusicNet [52]	10 classical composers across 21 types of instrumentation	330 annotated recordings
Cross-Era [56]	Baroque, Classical, Romantic and Modern periods	Chroma features and chords for 1600 pieces
Cross-Composer [56]	11 classical composers from the four style periods	Chroma features and chords for 1100 pieces

2.1.2.1 Classification

Classification is a type of supervised learning problem that consists of automatically assigning a label to unlabeled data or an object [3]. Solving a classification problem requires previously training a classifier algorithm with a labeled dataset. An example of this type of problem is the identification of musical genre. To build a model that could solve this problem, one would first train it with a dataset of songs, each of which is used to compute features and is annotated with a label specifying the genre they fit into. Then, we can make the model predict the genre of a different set of songs. This set needs to have the same features, but each song's genre would be unlabeled. The value of a label belongs to a set of classes. Classification problems are called binary if the set of classes has a size of two or multiclass if it has three or more classes. The example of genre classification is usually a multiclass classification problem.

2.1.2.2 Clustering

Clustering is a type of unsupervised learning problem that attempts to group entries in a dataset that are in some way similar by assigning them to a given cluster. These clusters do not have any predefined meaning, and their interpretation is therefore much more difficult than that of a label assigned in a classification problem. Unlike what happens in supervised learning, a clustering model is trained using an unlabeled dataset [3]. This type of problem is common in musicological research. For example, by training a model with pieces from composers from different countries and letting it cluster a different set of musical works, musicologists can study and try to understand the stylistic traits that set each country's music apart.

2.1.3 Dimensionality Reduction

In several types of problems, it is common for models to handle high-dimensional data, which results in a high number of features. In these cases, data is typically highly redundant [63] and its number of dimensions needs to be reduced in order to mitigate the effects of the "curse of dimensionality" [54]. Another problem is that it is challenging to visualize data beyond three dimensions, even though that might be useful in many situations. In order to mitigate these issues, dimensionality reduction techniques aim at transforming data to a lower-dimensional space.

In the literature discussed further ahead in sections 2.1.4 and 2.1.5, dimensionality reduction is often applied, either for easier visualization of features or before training classifiers to improve their performance. Two standard algorithms employed in this domain are Principal Component Analysis (PCA) [45] and Linear Discriminant Analysis (LDA) [21].

2.1.3.1 Principal Component Analysis

PCA is an unsupervised dimensionality reduction technique. Its goal is to project data of dimensionality D to a new space of dimensionality $M < D$ while maximizing the variance of the projected data [11]. To apply this technique to a dataset, we first compute the average standard deviation of each dimension and then iterate through all items, subtracting the mean of each dimension from each feature value and dividing it by the standard deviation, obtaining the matrix B. Next, from the resulting matrix, we compute the covariance matrix C and then its eigenvectors V and eigenvalues. The principal components T of matrix B are given by the expression:

$$T = BV \tag{2.1}$$

where the eigenvectors V represent the weights of each principal component [14]. The larger the weight of a principal component, the higher importance it has in explaining the variance of the data. This means that we can keep only the M principal components with the highest weights while still retaining most of the information in the original dataset.

2.1.3.2 Linear Discriminant Analysis

LDA is a supervised method that can be used for dimensionality reduction. It is considered supervised because, unlike PCA, it takes into account class labels. The goal of LDA is to project the original data onto a set of axes in a way that maximizes the distance between the means of each class' data while minimizing the variance within each class [14]. For a problem with two classes, this corresponds to maximizing the expression:

$$\frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} \tag{2.2}$$

where μ_n and s_n represent the mean and variance of class n's data, respectively. This approach can also be generalized to problems with three or more classes.

2.1.4 Feature Engineering

Several examples of the use of the Feature Learning approach can be found in literature.

Salamon et al. [50] train several classification models to identify musical genres. They use a method to extract pitch contours that describe the predominant melodic line in a given segment of a song [49]. They compute several features related to the pitch and duration of the contours, features related to the use of vibrato (a periodic variation in pitch), and the categorization of each contour according to the typology defined by Adams [1]. The authors compare the results when using only the high-level melodic features, Mel-frequency Cepstral Coefficients (MFCCs), and a combination of both. In one of the experiments using combined features, all classifier algorithms achieve accuracy higher than 95% (the exact values are not mentioned). Using the GTZAN dataset, they achieve an accuracy of 82% using an SVM (Support Vector Machine) classifier.

Weiß [56] builds classification models for subgenre and composer identification in classical music. To this end, the author experiments with several types of chroma features computed at different time resolutions to design higher-level tonal features to train these models. Further details on the set of Template-based and Tonal Complexity features proposed by the author are provided in sections 2.2.3 and 2.2.4, respectively.

For subgenre classification, the author builds a dataset of 1600 recordings containing 400 pieces for each style period: Baroque, Classical, Romantic, and Modern. For composer classification, he compiles a dataset of 1100 recordings of pieces from 11 different composers. He further divides these into several subsets according to instrumentation and the total number of composers, as shown in Table 2.2.

To evaluate classification performance, Weiß creates three types of models for each of the subgenre and composer classification tasks: one trained using only low-level standard audio features, one trained using the proposed chroma-based features, and the other one trained with a combination of both. Figure 2.2 depicts the general structure of these models.

As standard features, he considers MFCCs, octave spectral contrast, zero-crossing rate, audio spectral envelope, spectral flatness, spectral crest factor, spectral centroid, and loudness. With

Table 2.2: Classification subsets obtained from the *Cross-Era* and *Cross-Composer* datasets [56].

Dataset	**Classes**	**No. Classes**	**Items per class**	**Total items**
Cross-Era-Full	Baroque, Classical, Romantic, Modern	4	400	1600
Cross-Era-Piano	Baroque, Classical, Romantic, Modern	4	200	800
Cross-Era-Orchestra	Baroque, Classical, Romantic, Modern	4	200	800
Cross-Comp-11	Bach, Beethoven, Brahms, Dvořák, Handel, Haydn, Mendelssohn, Mozart, Rameau, Schubert, Shostakovich	11	100	1100
Cross-Comp-5	Bach, Beethoven, Brahms, Haydn, Shostakovich	5	100	500

these models, the author conducts several classification experiments using Gaussian Mixture Model (GMM), Random Forest (RF), and SVM classifiers. The RF classifier shows the worst performance for subgenre classification, while GMM and SVM perform similarly, even though SVM reports slightly higher accuracy (up to 92.2%) using combined features on the *Cross-Era-Full* dataset. For composer classification, the conclusions are similar, with SVM achieving up to 82.7% accuracy on the *Cross-Comp-11* dataset. Additionally, he performs another classification experiment using a GMM classifier, which involves applying a filter which prevents the same composer or performer from appearing both in the test and training folds of the cross-validation procedure. This approach results in worse accuracy and drops the previous values to 67.7% and 38.9%, respectively, but produces a less overfitted model.

In subsequent work, Weiß et al. [59] study the stylistic evolution of Western classical music by performing clustering experiments on pieces and composers according to style period, using the model described in Figure 2.3. They use the *Cross-Era-Full* dataset which they complement with 100 pieces from transitional composers of each style period, in a total of 2000 recordings. From this, they create two additional datasets, each containing Non-negative Least Squares (NNLS) chroma[2] and chord features, respectively. These chroma features are extracted for every 100 ms of audio. From the information in these datasets, they compute the same Template-based and Tonal Complexity features [56]. Additionally, they define a set of features related to chord transitions. The authors consider only the transitions between chord root notes and transitions between chord types, that is, for the chord progression $\{Dm, G, Am\}$, we would have the root note transitions $\{P4\uparrow/P5\downarrow,\ M2\uparrow/m7\downarrow\}$, and the chord type transitions $\{min \rightarrow maj,\ maj \rightarrow min\}$.

[2]Further details on this type of chroma features are provided in Section 2.2.2.

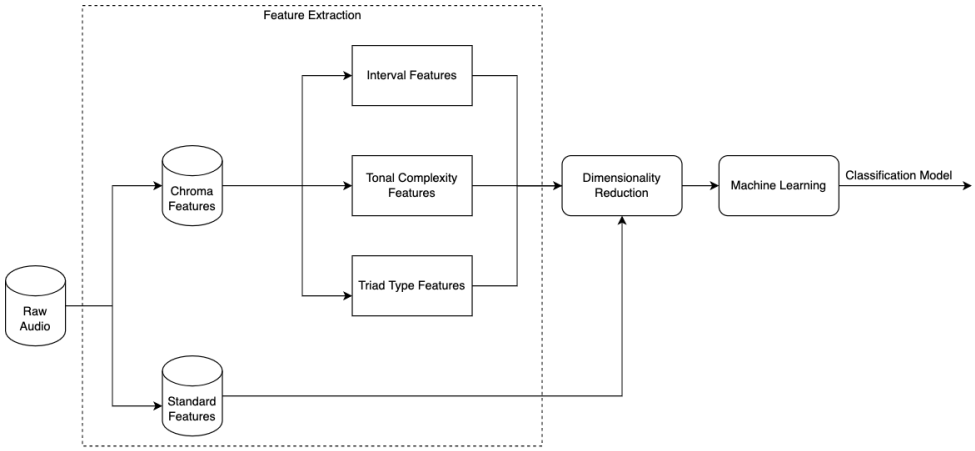


Figure 2.2: Representation of the classical subgenre/composer classification model proposed by Weiß [56] using a combination of chroma-based and standard audio features.

For clustering pieces, they perform PCA and use the first three principal components to create five clusters according to the K-means algorithm. The cluster assignments for each piece are mapped onto a timeline for visualization, allowing them to connect this information with musicological and music history knowledge regarding the several classical style periods.

For clustering composers, they first average the feature values for each piece (i.e., instead of multiple feature values per piece, each piece now contains a single value for each feature), perform PCA, and once again use the first three principal components for K-means clustering, with $K = 5$. They observe that, in general, composers with a similar lifetime are placed under the same cluster.

Li et al. [34] conduct several genre classification experiments using SVM, K-nearest Neighbors (KNN), GMM, and LDA classifiers. They extract audio features related to timbre (MFCCs, spectral centroid, spectral roll-off, spectral flux, zero crossings, and low energy), rhythm, and pitch. They identified SVM as the best performing classifier with an accuracy of up to 78.5% in one of the experiments.

Fu et al. [22] use a bag-of-words model to perform genre classification. They extract MFCC features from the audio and then use the K-means algorithm to cluster the feature vectors to build a bag of features. This data is used to train KNN and SVM classifiers, achieving 73.10% and 81.70% accuracy, respectively.

Weiß et al. [57] propose mid-level audio features that capture transitions between chords using Hidden Markov Models. The features consist on the probabilities of certain sequences of chord transitions happening at a given time in a piece. They test these features in style period classification tasks on the *Cross-Era-Piano*, *Cross-Era-Orchestra* and *Cross-Era-Full* datasets (Table 2.2) using a GMM classifier and applying composer filtering, an approach which prevents model overfitting (for a more detailed explanation on the filtering process, please refer to Section 5.4). They achieve 78.2% and 83.2% classification accuracy on the *Cross-Era-Full* and *Cross-Era-Orchestra* datasets, compared to previous results on the same datasets of 74.7% and 80.1%, respectively.

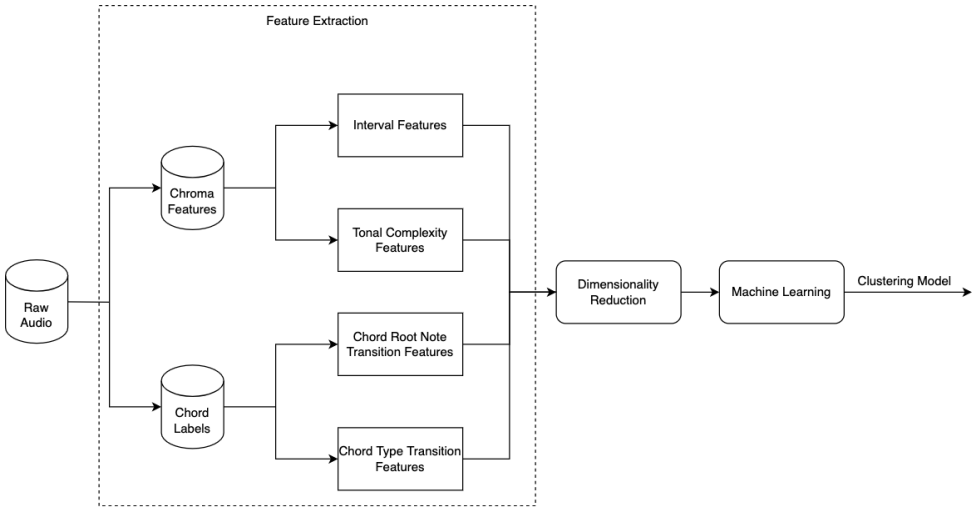


Figure 2.3: Representation of the style identification model proposed by Weiß et al. [59].

2.1.5 Feature Learning

Examples of Feature Learning can be found in several works in literature.

Lee et al. [32] perform unsupervised feature learning on a Convolutional Deep Belief Network (CDBN), which takes as input spectrograms processed using PCA and then use those features to perform genre and artist classification tasks. In the experiments that yielded the best results, they use an SVM classifier, achieving accuracy values of 73.1% and 81.9% for each task, respectively.

Sigtia and Dixon [51] present several methods of improving the performance of deep neural networks for learning music features from spectrograms extracted using the Fast Fourier Transform (FFT) and evaluate their experiments by performing genre classification using an RF classifier. Using the Rectified Linear Unit (ReLU) activation function, Stochastic Gradient Descent, and dropout (a regularization technique), they reach an accuracy of 83%. They conclude that using ReLU and Hessian-Free optimization significantly reduces training time.

Wang and Tzanetakis [55] use siamese convolutional neural networks fed with Constant-Q Transform (CQT) spectrogram and Mel-spectrogram features to identify singing style in voice recordings. They use a dataset containing recordings of 5429 singers, each singing a set of 14 songs, and perform clustering experiments that attempt to group songs and singers under the same cluster.

Park et al. [44] compare the performance of a Deep Convolutional Neural Network (DCNN) model with a siamese neural network model in music genre classification and song retrieval tasks. Each basic network is composed of 5 convolutional and max-pooling layers and is trained with MFCC features. The DCNN model contains only a single network, whereas the siamese model contains two connected networks. They train these models with MFCC features extracted from the Million Song Dataset [10] and then perform classification experiments using three other datasets. In both tasks, they conclude that the siamese model, in general, outperforms the basic model. However, they observe that the basic model appears to be more robust for classifying unseen

Table 2.3: Relevant Musical Style Classification Models in Literature. Features marked with an asterisk (*) are subsequently used to automatically produce new features using deep learning techniques, as per the feature learning approach previously described in Section 2.1.5.

Work	**Target**	**Train Dataset**	**Test Dataset**	**Features**	**Classifier**	**Acc.**
Li et al. [34]	Genre	10 genres, 100 songs each	10 genres, 100 songs each	Timbre, Rhythm and Pitch	SVM	78.5%
Lee et al. [32]	Genre	ISMIR2004	ISMIR2004	Spectrogram*	SVM	73.1%
Lee et al. [32]	Artist	ISMIR2004	ISMIR2004	Spectrogram*	SVM	81.9%
Sigtia & Dixon [51]	Genre	GTZAN	ISMIR2004	Spectrogram*	Random Forest	73.46%
Salamon et al. [50]	Genre	GTZAN	GTZAN	Pitch Contours, MFCCs	SVM	82%
Fu et al. [22]	Genre	GTZAN	GTZAN	MFCCs	SVM	81.70%
Weiß [56]	Style Period	Cross-Era	Cross-Era	Intervals, Triads, Tonal Complexity	SVM	92.2%
Weiß [56]	Composer	Cross-Composer	Cross-Composer	Intervals, Triads, Tonal Complexity	SVM	82.7%
Park et al. [44]	Genre	Million Song Dataset	GTZAN	MFCCs*	Linear Soft-max	69.93%

genres, i.e., genres which the model was not trained to identify. The highest classification accuracy values of 69.66% and 69.93% are reported on the GTZAN dataset [53] using KNN and Linear Softmax classifiers, respectively, in the genre identification task.

2.1.6 Evaluation of Musical Style Identification Models

Due to the large number of datasets used in training style identification models and their different properties (such as different genres, sizes, and diversity), comparing the performance of models is non-trivial. Nevertheless, Table 2.3 shows some relevant results from style classification tasks in literature. We present only the test scenario that shows the highest accuracy value for experiments that compare several classifiers.

Concerning classification algorithms, we observe the predominant use of KNN, RF, GMM, and SVM. In general, SVM outperforms the other classifiers in most experiments.

A commonly employed method for evaluating the performance of a model is cross-validation. This technique involves randomly splitting the training dataset into equally-sized subsets (called folds). Suppose we split it into five folds $F_k, k = 1, ..., 5$. Then, we train five models, each using four of the folds as training data and the last one as testing data for validation. Model 1 is trained

using folds F_2 to F_5 and uses F_1 as a testing set. Model 2 is trained using folds F_1, F_3, F_4 and F_5, and uses F_2 as a validation set. This process continues until all models have been trained. In the end, we compute the desired evaluation metric (e.g., accuracy) for each model and take the average of those values [3].

Regarding the comparison of models, we look at the method employed in the various MIR tasks proposed in the Music Information Retrieval Evaluation Exchange (MIREX) [20]. For a given task, for example, genre recognition, models are first ranked according to a metric such as the accuracy score, which is determined by the consensus of participants, and then pairwise comparison is performed between all of them using a statistical test. The tests used in past editions have included the Tukey-Kramer Honestly Significant Difference and Analysis of Variance. Essentially, rather than just ranking models according to a given score, a more detailed comparison is performed to identify whether statistically significant performance differences exist between them.

2.2 Tonal Description of Musical Signals

This section presents several concepts, methods, and algorithms for describing the tonal content of audio signals used in works related to musical style identification.

2.2.1 Spectral Representations

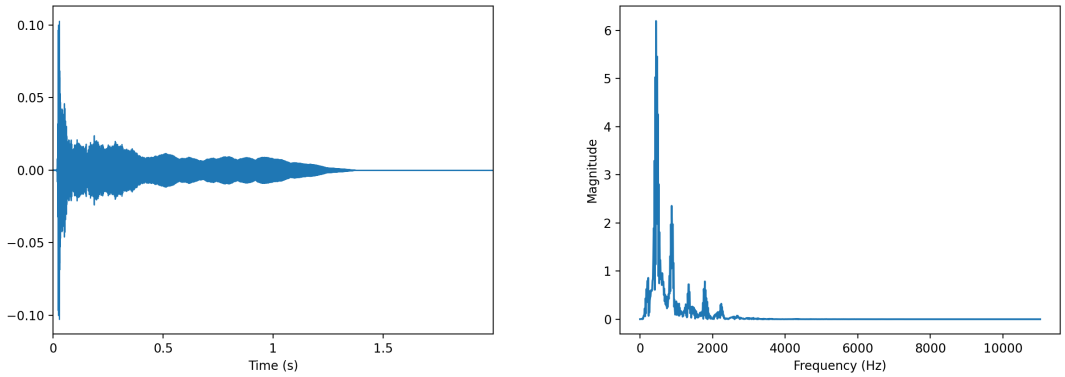
A pure tone is a sound that contains a single oscillating frequency. Most sounds one hears throughout their day are called complex sounds because they are actually comprised of many pure tones combined, [39], and this, of course, applies to music as well. With this in mind, a common approach for analyzing audio signals is to look at their frequency content. Two existing methods to conduct this analysis are the Fourier Transform and the CQT.

2.2.1.1 Fourier Transform

The Fourier Transform of a signal $x(t)$ is defined as:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt \tag{2.3}$$

The main idea is to compare the signal with sinusoidal waves of various frequencies ω. Suppose we apply this equation to a given signal using different frequency values. In that case, we obtain a magnitude coefficient representing how present each frequency is in the signal. If the value of that coefficient is high for a particular frequency, it means that the signal contains that oscillating frequency [39]. This means that we can extract all of the individual frequencies from a complex signal. A widespread application of the Fourier Transform in audio analysis is the extraction of the spectrum, which is a visual representation of the frequency content of an audio signal. An example of a spectrum can be seen in Figure 2.4.



(a) (b)

Figure 2.4: Waveform (a) and spectrum (b) of an A4 (440Hz) pitch played on a piano.

Equation 2.3 is adequate for continuous signals, meaning it could be applied to analog audio. However, only a finite number of values can be stored when dealing with digital audio, i.e., digital signals are not continuous but instead discrete [39]. To extract the individual frequency magnitudes from this type of signal, we would instead use the Discrete Fourier Transform (DFT). This version of the Fourier Transform is defined by a sum rather than an integral. The DFT of a signal $x(t)$ can be written as:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t)e^{-i\omega_k t}, \quad k = 0,1,2,...,N-1 \tag{2.4}$$

Where ω_k represents the kth frequency sample and N the total number of samples taken. The computation of the DFT can be further optimized by the FFT algorithm, which reduces its computational complexity from $O(N^2)$ to $O(Nlog_2N)$ and is therefore much more efficient [39]. Another useful version of the Fourier Transform is the Short-time Fourier Transform (STFT). The main idea of the STFT is to consider only a small section of the signal by multiplying it by a window function. To obtain the signal's frequency components at different time instances, the window function is shifted along the time axis, and the Fourier Transform (usually the FFT) is computed for each windowed signal [39]. A common application of the STFT is the extraction of an audio spectrogram (Figure 2.5), which plots the signal's frequency content throughout time.

2.2.1.2 Constant-Q Transform

The CQT is a time to frequency domain transformation that takes into account the way humans perceive sound by using a set of filters with logarithmically spaced center frequencies f_k given by:

$$f_k = f_{min}2^{\frac{k}{n}} \tag{2.5}$$

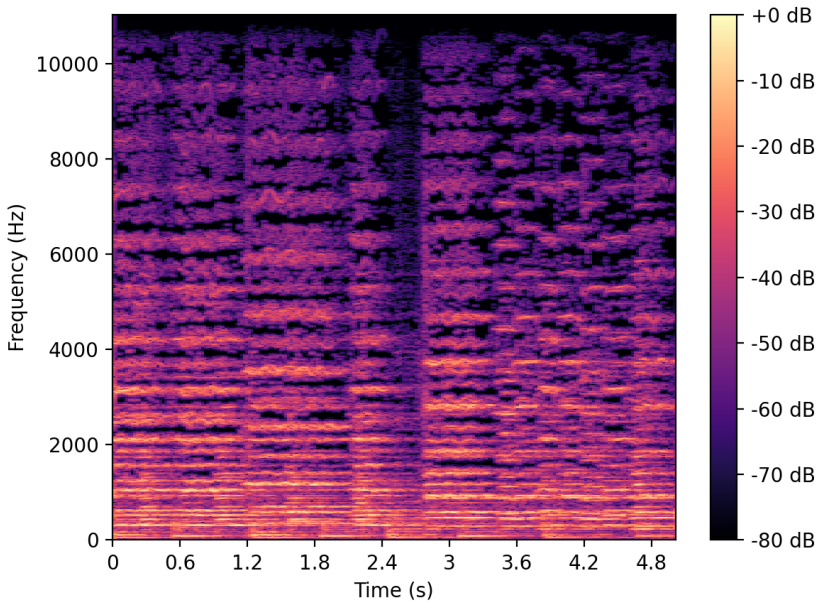


Figure 2.5: Spectrogram extracted from an excerpt of J. Brahms' Hungarian Dance no. 5.

where f_{min} is the center frequency of the lowest filter and n the number of filters per octave. The choice of n affects the resolution δ given by:

$$\delta = 2^{\frac{1}{n}} - 1 \tag{2.6}$$

For a given resolution δ, the quality factor Q is:

$$Q = \frac{f}{\delta f} = \frac{1}{\delta} \tag{2.7}$$

In order to keep the value of Q constant, the length of the kth analysis window N_k must be defined accordingly as follows:

$$N_k = \frac{S}{\delta f_k} = Q\frac{S}{f_k} \tag{2.8}$$

where S represents the sampling rate. Finally, the kth component of the CQT, $Y[k]$, can be determined from its corresponding component in the FFT:

$$Y[k] = \frac{1}{N_k}\sum_{i=0}^{N_k-1} y[i] \cdot w[k,i] \cdot e^{-\frac{j2\pi Qn}{N_k}} \tag{2.9}$$

where $y[i]$ is the sampled audio signal and $w[k,i]$ the window function used in the analysis [13].

2.2.2 Chroma Features

Chroma features are a way of aggregating all spectral information related to a given pitch into a single value [39]. The idea is to measure how predominant a given pitch is in a given frame of the audio signal according to twelve possible pitch classes. This means that the same note across different octaves will count towards the same pitch class, i.e., the pitches C1, C2, C3 all belong to the same pitch class C. Therefore, we can define the chroma vector as being a vector of dimension $D = 12$ which describes the energy of each of the pitch classes $q \in [0, D-1]$ [56]. From a log-frequency spectrogram (such as the CQT) Y, we obtain a chroma vector by summing up the coefficients of all pitches $\{p|p \bmod 12 = d\}$ belonging to each pitch class d. By repeating this process for all frames of the signal, a chromagram representation C is obtained as follows:

$$C(d,m) = \sum_{\{p|p \bmod 12 = d\}} Y(p,m) \tag{2.10}$$

where $m \in [0, M-1]$ is the frame's index out of M total frames. Figure 2.6 shows the chromagram extracted from a recording of an A4 pitch played on a piano. As expected, we observe that the most preeminent pitch class is A. The E pitch class is also very noticeable. This is because E6 is the third harmonic of A4 (the first being A4 itself and the second the octave, A5), which allows us to conclude that chroma features can still capture pitches that are not directly played by an instrument. Their musical intuitiveness makes them ideal features for representing note and chord events [23]. These features are used by various authors [56, 59, 7, 37] in works regarding the design of higher-level features and music style identification.

A particularly relevant type of chroma features in current literature is NNLS chroma [37]. This approach involves applying the NNLS algorithm on a log-frequency spectrum using a note dictionary with the goal of reducing the effect of overtones on the computation of the chroma features. In Section 2.2.6 we further detail how NNLS chroma outperforms other types of chroma features in automatic chord recognition tasks.

2.2.3 Template-based Features

The set of Template-based tonal features proposed by Weiß [58] measures the likelihood of a given interval or triad type being present in a chroma vector (Table 2.4).

For a chroma vector $\mathbf{c} = (c_0, c_1, ..., c_{11})$, this average likelihood value can be calculated as follows:

$$\Psi^T(\mathbf{c}) = \sum_{q=0}^{11} \left(\prod_{k=0}^{11} \left(c_{(q+k)\ mod\ 12} \right)^{T_k} \right) \tag{2.11}$$

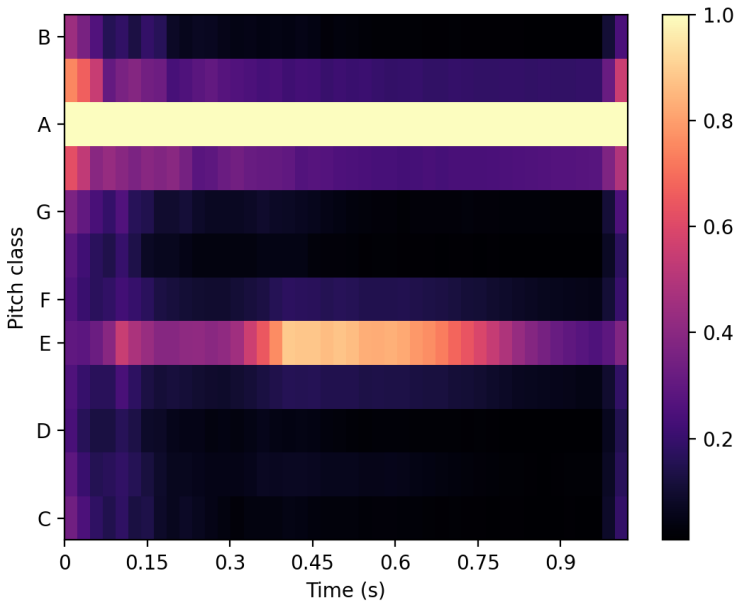


Figure 2.6: Chromagram of an A4 pitch played on a piano.

By adequately choosing a template T in Equation 2.11, one can determine the probability of a given interval category being present:

$$
\begin{aligned}
T^{IC1} &= (1,1,0,0,0,0,0,0,0,0,0,0)^T \\
T^{IC2} &= (1,0,1,0,0,0,0,0,0,0,0,0)^T \\
T^{IC3} &= (1,0,0,1,0,0,0,0,0,0,0,0)^T \\
T^{IC4} &= (1,0,0,0,1,0,0,0,0,0,0,0)^T \\
T^{IC5} &= (1,0,0,0,0,1,0,0,0,0,0,0)^T \\
T^{IC6} &= (1,0,0,0,0,0,1,0,0,0,0,0)^T
\end{aligned} \tag{2.12}
$$

Table 2.4: Template-based features proposed by Weiß [58].

Feature	**Definition**
IC1	m2 / M7
IC2	M2 / m7
IC3	m3 / M6
IC4	M3 / m6
IC5	P4 / P5
IC6	+4 / °5
M	Major
m	Minor
°	Diminished
+	Augmented

And similarly, for each triad type:

$$
\begin{aligned}
T^M &= (1,0,0,0,1,0,0,1,0,0,0,0)^T \\
T^m &= (1,0,0,1,0,0,0,1,0,0,0,0)^T \\
T^\circ &= (1,0,0,1,0,0,1,0,0,0,0,0)^T \\
T^+ &= (1,0,0,0,1,0,0,0,1,0,0,0)^T
\end{aligned}
\tag{2.13}
$$

Because the chroma vector representation collapses octave information, the six interval categories refer to the complementary intervals in Western tonal music. This implies that, for instance, these interval categories can not distinguish between a minor second or major seventh interval.

2.2.4 Tonal Complexity Features

Weiß [60] defines an additional set of features that attempt to quantify the tonal complexity of an audio signal, also based on chroma vector representations:

- $\Gamma_{\mathbf{Diff}}(\mathbf{c})$: absolute difference between all neighboring elements of the chroma vector ordered in fifths.
- $\Gamma_{\mathbf{Std}}(\mathbf{c})$: standard deviation of the chroma vector.
- $\Gamma_{\mathbf{Slope}}(\mathbf{c})$: negative slope of the chroma vector ordered in a descending series.
- $\Gamma_{\mathbf{Entr}}(\mathbf{c})$: Shannon entropy of the chroma vector.
- $\Gamma_{\mathbf{Sparse}}(\mathbf{c})$: non-sparseness of the chroma vector defined as the relationship between its l_1 and l_2 norms.
- $\Gamma_{\mathbf{Flat}}(\mathbf{c})$: flatness measure of the chroma vector defined as the relationship between its geometric and arithmetic means.
- $\Gamma_{\mathbf{Fifth}}(\mathbf{c})$: angular deviation of the chroma vector ordered in fifths.

According to the author, these features may capture musical aspects such as dissonance levels and harmonic change magnitude.

2.2.5 Tonal Interval Space

Tonal pitch spaces are models which typically aim to map the perceived proximity of pitches, chords, or regions [33]. Possibly the earliest pitch space to have been proposed is the well-known circle of fifths, whose origin can be traced back to the mathematician Pythagoras [30]. In this context, we focus on the TIS proposed by Bernardes et al. [7]. The TIS addresses some of the common limitations in existing tonal pitch spaces, namely the prior requirement of knowing the key when measuring the distance between chords or pitches, the fact that most models do not

account for how humans perceive the distance between sonorities, and the lack of representation of consonance and dissonance metrics.

The TIS has been applied in several research projects in fields such as MIR and generative music. Bernardes et al. develop Conchord [8], a real-time system that generates chord progressions by navigating through the TIS. Navarro-Cáceres et al. [42] propose a model for measuring musical tension related to melodic and harmonic motion by making use of the distance properties of the TIS. Bernardes et al. [9] present a hierarchical harmonic mixing method to help users create music mashups, which consists in using metrics from the TIS to compute the harmonic compatibility between audio tracks.

In Chapter 3 we look further into the TIS by describing its mathematical definition, exploring its spatial properties and proposing the addition of new descriptors.

2.2.6 Automatic Chord Recognition

The study of chord progressions throughout a piece is considered essential for composing and analyzing Western tonal music [39]. It is, therefore, no surprise that automatic chord recognition in music has been pursued computationally [37]. Most computational methods for the detection of chords operate in two steps. The first step is to extract audio features that capture harmony-related information. Chroma features, in particular, are suitable for this task. The second step involves pattern matching techniques to map the audio features to chord labels [39].

Mauch et al. [37] develop an automatic chord detection method that relies on a beat-synchronized NNLS chroma spectrogram. This method achieves an accuracy of up to 80% in the MIREX 2009 Chord Detection task dataset, distinguishing between 120 different chords and outperforming state-of-the-art approaches at the time, which achieved only up to 74% accuracy.

In a contrasting approach, Zhou et al. [64] use a deep neural network that learns high-level features from audio to detect up to 24 different chords. The training data consists of frequency domain data extracted by applying the CQT and then PCA for decorrelation. They test two different six-layer deep neural network architectures, one in which every layer has the same number of neurons and the other in which the middle layers have fewer neurons than the others (bottleneck architecture). After training the model, they experiment with different classifiers and obtain up to 91.9% accuracy with an optimal configuration.

Korzeniowski et al. [31] develop an end-to-end chord recognition system in which they train a convolutional neural network to infer audio features from spectral information for predicting chord labels. These features are fed to a conditional random field to classify each audio frame according to one of 24 possible chord labels (major and minor). The obtained results showed that this approach performed slightly better than state-of-the-art systems at the time.

McFee et al. [38] perform chord recognition over a large vocabulary of 170 possible chords. For each audio frame, they compute the CQT and extract the root note of the chord and the pitch classes it contains (relative to the root). This representation is then mapped to chord labels. For example, (1, (0, 4, 7)) represents a C#maj chord, since C# corresponds to pitch-class 1. The data is used to train a convolutional recurrent network for the task of chord identification. The results

showed that this approach resulted in improvements of up to 5% in prediction accuracy compared to the state of the art.

2.2.7 Harmonic Change Detection

In MIR tasks, features are commonly extracted from segments of the audio instead of considering the entirety of the information at once. Typically, these segments have a fixed time duration and are oblivious to the structure of a piece. However, feature extraction using structurally aware segmentation has proven more accurate [5]. A possible approach to partitioning audio into segments that take structure into account is by looking at changes in the harmony.

Harte et al. [28] propose the Harmonic Change Detection Function (HCDF) based on a 6-dimensional tonal space. The algorithm begins by applying a Constant-Q transform to the signal followed by the calculation of a chromagram. Using the proposed tonal space, a 6-dimensional tonal centroid vector ζ_n is calculated for each time frame n. This allows them to define the HCDF ε_n as the euclidean distance between the Gaussian-smoothed tonal centroid vectors $\hat{\zeta}_{n-1}$ and $\hat{\zeta}_{n+1}$ (Equation 2.15). Gaussian smoothing is applied in order to reduce the effects of transients and noise in the signal.

$$\varepsilon_n = \sqrt{\sum_{d=0}^{5} [\hat{\zeta}_{n+1}(d) - \hat{\zeta}_{n-1}(d)]^2} \tag{2.14}$$

In order to identify the instants of harmonic transitions, they apply peak detection to the HCDF values.

Compared to previous approaches, which obtained an average accuracy of up to 31% in chord boundary detection, the HCDF achieves up to 53% in this task, suggesting that it is better at detecting harmonic changes in the signal.

Degani et al. [19] suggest improvements to Harte et al.'s HCDF by experimenting with several types of chroma features and distance measurements. Overall, they conclude that chroma feature extraction methods that attempt to minimize the effects of certain musical traits such as timbre, transients, and noise improve the performance of the HCDF.

Ramoneda et al. [47] propose an improved version of Harte et al.'s HCDF by replacing the use of the proposed tonal space with the TIS. They define the new HCDF ξ_n for frame n as the distance between the Gaussian-smoothed TIVs $\widehat{\mathbf{T}}_{\mathbf{n-1}}$ and $\widehat{\mathbf{T}}_{\mathbf{n+1}}$. Equation 2.15 considers Euclidean and cosine distance metrics as follows:

$$\xi_n^{euc} = \|\widehat{\mathbf{T}}_{\mathbf{n+1}} - \widehat{\mathbf{T}}_{\mathbf{n-1}}\|, \quad \xi_n^{cos} = \frac{\langle \widehat{\mathbf{T}}_{\mathbf{n+1}}, \widehat{\mathbf{T}}_{\mathbf{n-1}} \rangle}{\|\widehat{\mathbf{T}}_{\mathbf{n+1}}\| \|\widehat{\mathbf{T}}_{\mathbf{n-1}}\|} \tag{2.15}$$

They experiment with different sets of parameters such as chroma features and distance measures and compare the new method with previous approaches. They observe that their algorithm shows an improvement of 5.57% and 6.28% in the *f-score* and *recall* measures, respectively, compared to the previous implementation of the HCDF.

2.3 Summary

In this chapter, we discussed techniques for the identification of musical style in current literature. Concerning machine learning approaches, we identified two typical architectures of style identification systems: Feature Engineering and Feature Learning. In addition, we presented several baseline concepts related to the tonal description of musical audio signals and discussed current state-of-the-art audio features in this context.

Most style identification models in literature have neglected the study of harmony in favor of aspects such as timbre or rhythm. Many of the studies that have taken harmony into account propose features that do not account for its long-term horizontal structure. Furthermore, fixed-time segmentation is typically employed instead of structurally-aware segmentation, despite the latter being proven to reduce extraction errors. Finally, most audio features proposed in literature do not possess a solid perceptual basis that accounts for the way humans perceive distances between individual pitches, intervals, or chords.

Chapter 3

Tonal Feature Design

This chapter discusses the design of new tonal features based on the TIS proposed by Bernardes et al. [7]. Section 3.1 details the mathematical formulation of the TIS and the musical interpretations of vector distances as harmonic qualities such as chromaticity, dyadicity, triadicity, diminished quality, diatonicity, whole-toneness, and dissonance. Furthermore, we propose a set of new harmonic descriptors derived from the TIS, namely Euclidean and cosine distance between audio frames, Euclidean and cosine tonal dispersion, TIV entropy, and harmonic rhythm. Section 3.2 presents a statistical analysis of the descriptors based on the TIS, which explores their potential for discriminating between style periods. Finally, Section 3.3 summarizes the proposed contributions for the tonal description of musical audio signals.

3.1 Tonal Interval Space

The perceptually inspired TIS, previously mentioned in Section 2.2.5, explores the properties of the Fourier coefficients of a chroma vector. The space maps a chroma vector $\mathbf{c} = (c_0, c_1, ..., c_{11})$ to a 12-dimensional complex Tonal Interval Vector (TIV) using the DFT as follows:

$$T(k) = w_a(k) \sum_{n=0}^{N-1} \bar{c}_n e^{-\frac{j2\pi kn}{N}}, \quad 1 \leq k \leq 6 \tag{3.1}$$

where $N = 12$ is the dimension of the chroma vector, $\mathbf{w_a} = \{3, 8, 11.5, 15, 14.5, 7.5\}$ the set of weights used to adjust the contribution of each dimension of the space according to dyad's dissonance ratings[1], and $\bar{\mathbf{c}}$ is the normalized chroma vector.

The $T(k)$ components $7 \leq k \leq 12$ can be discarded since they are symmetrical to the $T(k)$ components $1 \leq k \leq 6$. A coefficient k can be represented as a point within a circle by plotting the real and imaginary numbers as x and y coordinates, respectively. Figure 3.1 shows the plot representation of the C major triad, including the notes C, E, and G (pitch classes 0, 4, and 7). In each circle, the distance of the red dot from the center represents the prevalence of each complementary

[1] The $\mathbf{w_a}$ set of weights is optimized for audio input. For symbolic data, there is a different set of weights $\mathbf{w_s} = \{2, 11, 17, 16, 19, 7\}$.

interval in this pitch class configuration, i.e., the further away it is from the center, the higher the likelihood of that interval being present in the chroma vector. A tonal quality is associated with each complementary interval category (further details on this topic are provided in section 3.1.1).

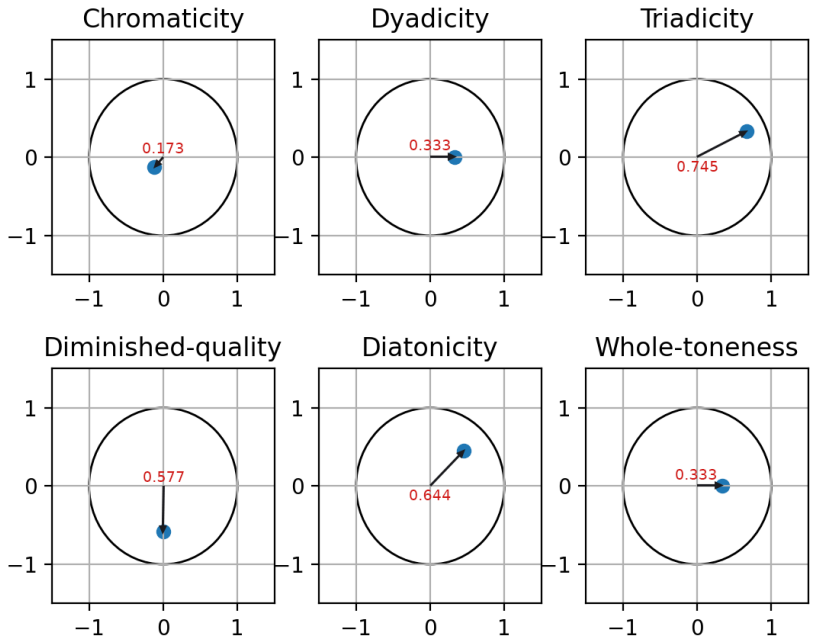


Figure 3.1: TIV for the C major chord which contains pitch classes 0, 4, and 7, and the M3/m6, m3/M6, and P4/P5 intervals, which translate to the tonal qualities of triadicity, diminished-quality, and diatonicity, respectively. Each circle represents one of the $T(k)$ components projected as a 2-dimensional vector in a complex plane. The value in red corresponds to the norm of each vector, which is higher for $T(3)$, $T(4)$, and $T(5)$, the coefficients associated with each interval in the C major chord.

The magnitude and phase coefficients of each value $T(k)$ in the TIV can be computed from the complex number values $\Re\{T(k)\}$ and $\Im\{T(k)\}$ using Equations 3.2 and 3.3, respectively.

$$\|T(k)\| = \sqrt{\Re\{T(k)\}^2 + \Im\{T(k)\}^2}, \quad 1 \leq k \leq 6 \tag{3.2}$$

$$\varphi(k) = \tan^{-1} \frac{\Im\{T(k)\}}{\Re\{T(k)\}}, \quad 1 \leq k \leq 6 \tag{3.3}$$

In the following sections, we detail how the magnitude and phase of TIVs can be used to compute perceptually-inspired audio features.

3.1.1 TIV Coefficients

For each value of k, the corresponding TIV coefficient is interpreted as being associated with one of the six complementary intervals in Western tonal music. Additionally, following Yust's interpretation of the Fourier coefficients of a chroma vector [62], a harmonic quality can be attributed to the magnitude of each TIV coefficient (Table 3.1).

Table 3.1: Correspondence between the coefficients of the TIV and the six complementary interval categories in Western tonal music.

T(k)	IC	Interval	Harmonic Quality
T(1)	IC1	m2 / M7	chromaticity
T(2)	IC6	Tritone	dyadicity or 'quartal-quality'
T(3)	IC4	M3 / m6	triadicity or 'hexatonicity'
T(4)	IC3	m3 / M6	octatonicity or 'diminished quality'
T(5)	IC5	P4 / P5	diatonicity
T(6)	IC2	M2 / m7	whole-tone

Each magnitude $\|T(k)\|$ is normalized between the values 0 and 1 by dividing it by its respective weight $w_a(k)$.

$\frac{\|T(1)\|}{w_a(1)}$ indicates the degree of chromaticity of a given sonority. Its value is minimal for pitch-class sets with an even spacing such as tonal chords and scales and maximal for chromatic distributions.

$\frac{\|T(2)\|}{w_a(2)}$ is defined as dyadicity, which relates to the presence of tritones and fifths in tonal contexts. In non-tonal contexts, it is also referred to as 'quartal-quality' because it is maximal for chords comprised of stacked perfect and augmented fourths.

$\frac{\|T(3)\|}{w_a(3)}$ is referred to as triadicity, which in tonal contexts evaluates the presence of stacked major and minor thirds and weights the position of a pitch-class distribution towards the dominant or subdominant side of a given key. In non-tonal contexts, its value is maximal for the augmented triads or hexatonic scale.

$\frac{\|T(4)\|}{w_a(4)}$ is associated with diminished quality and has a maximal value for pitch-class sets that represent diminished seventh chords.

$\frac{\|T(5)\|}{w_a(5)}$ evaluates diatonicity, that is, the concentration of a sonority in the circle of fifths. This value is higher for pitch-class distributions that represent diatonic aggregates such as a major or minor scale or triad.

Lastly, $\frac{\|T(6)\|}{w_a(6)}$ is defined as whole-toneness, which indicates the proximity of a sonority to one of the two existing whole tone sets. Its value approaches 1 for the whole-tone scale.

3.1.2 Consonance and Dissonance

The TIS provides an indicator of consonance (or dissonance) of a given sonority. Consonance is computed as the normalized magnitude of a TIV, $\frac{\|\mathbf{T}\|}{\|\mathbf{w_a}\|}$, while dissonance corresponds to this value

subtracted from unity, $1 - \frac{\|\mathbf{T}\|}{\|\mathbf{w_a}\|}$. The magnitude (or consonance indicator) equals the distance from the center of the space; therefore, consonant TIVs exist further from the center. Table 3.2 shows the dissonance value for different types of chords. To represent each chord in the TIS, we derive its pitch-class set and convert it into a chroma vector using a method that simulates the spectral content based on the average spectrum of 1338 recorded tones played by 23 Western orchestral instruments [9]. This way, the set of audio weights $\mathbf{w_a}$ may be applied.

Table 3.2: Different chords ordered by increasing dissonance value.

	maj/min	**dim**	**aug**	**m7**	**M7**	**7**
Dissonance	0.784	0.805	0.807	0.814	0.825	0.832

The values for each chord line up with their perceived dissonance. Major and minor chords sound the least dissonant, while major and dominant 7th chords sound the most dissonant, as they contain a minor 2nd and a tritone interval, respectively.

While consonant textures are predominant in works from the Baroque and Classical periods, music becomes increasingly more dissonant during the Romantic period and even more so during the Modern period [61]. Motivated by these facts and the previous observations, we compute the average dissonance value per piece (Figure 3.2) on a large dataset of 1600 musical audio tracks per Western classical music style (the dataset has a uniform split of 400 musical audio tracks per style). Four era labels are adopted: Baroque, Classical, Romantic, and Modern (for a comprehensive description of the dataset, please refer to Section 2.1.1).

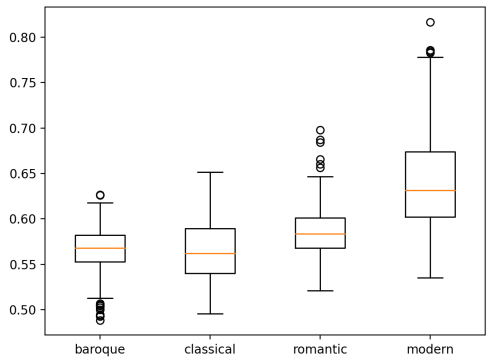


Figure 3.2: Dissonance values for each style period.

Although subtle, we observe an increase in dissonance for romantic pieces when compared to previous periods. It is for the Modern period that we see noticeably higher dissonance values as expected. Another observation is that the Baroque period generally shows slightly higher values when compared to the Classical period. A possible explanation for this could be the simplicity and restricted use of ornaments in pieces of the Classical period [27].

3.1.3 Distance Between Consecutive Audio Frames

One of the properties of the TIS is its ability to capture perceptually similar pitch-class distributions as distances in space. For example, the minor second interval resulting from C and C#, which are close together on a piano keyboard and in the chroma space, are relatively distant in the TIS due to being perceived as less related than remaining intervals. On the other hand, intervals of perfect fifth, such as C and G, exist at a very small distance in space. Furthermore, TIVs of chords are placed in the space such that the closer they are to each other, the smoother the voice leading from one to the other.

Euclidean and cosine distance metrics between TIVs have been considered in the literature [7, 46], which measure different perceptual characteristics of the signal. The angular or cosine distance θ between two TIVs $\mathbf{T_1}$ and $\mathbf{T_2}$ is defined as:

$$\theta\{\mathbf{T_1},\mathbf{T_2}\} = \frac{\langle \mathbf{T_1},\mathbf{T_2}\rangle}{\|\mathbf{T_1}\|\|\mathbf{T_2}\|} \tag{3.4}$$

This distance metric measures the difference between the phases $\varphi(k)$ of each TIV and captures the number of shared pitch classes between two TIVs, which relates to parsimony voice leading. The smaller the distance, the greater the shared pitch-class content between the two.

The Euclidean distance d between two TIVs $\mathbf{T_1}$ and $\mathbf{T_2}$ is defined as:

$$d\{\mathbf{T_1},\mathbf{T_2}\} = \sqrt{\sum_{k=1}^{6} |T_1(k) - T_2(k)|^2} \tag{3.5}$$

The Euclidean distance considers both the phase $\varphi(k)$ and magnitude $\|T(k)\|$ of each component and not only measures shared pitch classes between TIVs but also to what degree they share similar interval content.

Table 3.3 presents the Euclidean and cosine distance values between the C major triad and each triad belonging to the C major tonality. Excluding the C major triad itself, the smallest distances are attributed to the A and E minor triads, as they both share two pitch classes and the same type of intervals with the C major triad, having the smoothest voice leading. In contrast, the b° triad only shares one interval and no pitch classes and is, therefore, the furthest from the C major triad.

Table 3.3: Euclidean (d) and angular (θ) distance between the C major triad and each of the diatonic triads in the C major tonality, ordered by increasing distance. For each triad, their constituent pitch classes and intervals are displayed.

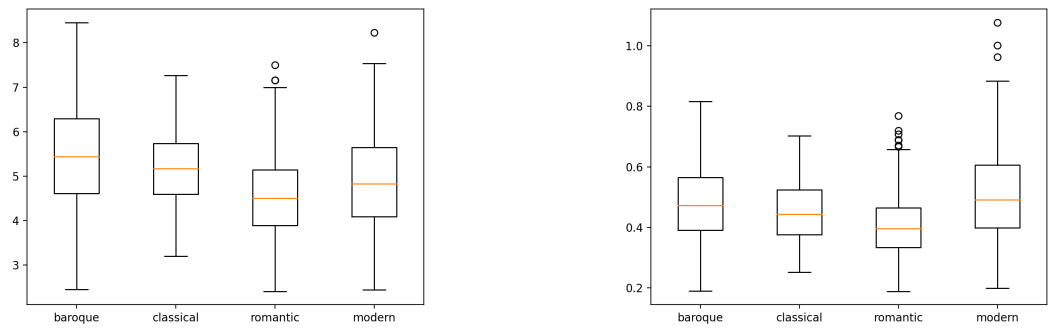
	C {C, E, G} {M3, m3}	**a** {A, C, E} {m3, M3}	**e** {E, G, B} {m3, M3}	**F** {F, A, C} {M3, m3}	**G** {G, B, D} {M3, m3}	**d** {D, F, A} {m3, M3}	**b°** {B, D, F} {m3, m3}
d	0	13.40	14.97	20.70	20.70	24.86	26.99
θ	0	0.88	0.99	1.43	1.43	1.81	1.97

To take advantage of the properties of these two distance metrics, we now intend to build new features that evaluate the distances between audio frames. By doing this, we can study the

magnitude of the changes in tonal content throughout a given piece. To accomplish this, we compute the TIV of each frame and define the set of inter-frame Euclidean and cosine distances as follows:

$$s_n^{euc} = d\{\mathbf{T_n}, \mathbf{T_{n+1}}\}, \quad s_n^{cos} = \theta\{\mathbf{T_n}, \mathbf{T_{n+1}}\} \tag{3.6}$$

Following the same approach as Section 3.1.2, we compute the mean Euclidean and cosine inter-frame distance for each piece and display it in Figure 3.3.



(a) Euclidean inter-frame distance

(b) Cosine inter-frame distance

Figure 3.3: Euclidean and cosine distance between consecutive audio frames grouped by style period.

For both distance metrics, the Romantic period shows lower values. Considering the Euclidean distance, this suggests a tighter relationship between the interval content of neighboring frames in pieces from this period. For the cosine distance, it could indicate smoother voice leading.

3.1.4 Tonal Dispersion

Tonal dispersion refers to how much the harmony of a given segment deviates from the tonal center of the piece. This measure makes it possible, for instance, to identify moments in which notes outside the key of the piece (in tonal contexts) are being played. The tonal center TIV $\overline{\mathbf{T}}$ corresponds to the TIV of the average pitch-class distribution calculated for the entire piece. Mathematically, the tonal dispersion of an audio frame n is defined as:

$$\sigma_n^{euc} = d\{\mathbf{T_n}, \overline{\mathbf{T}}\}, \quad \sigma_n^{cos} = \theta\{\mathbf{T_n}, \overline{\mathbf{T}}\} \tag{3.7}$$

We adopt the Euclidean and cosine distance metrics in order to capture the several harmonic relationships mentioned previously in Section 3.1.3.

Table 3.4 presents the Euclidean and cosine tonal dispersion values for several triads relatively to the C major scale. The tonal dispersion values for the triads that belong to this tonality (presented in bold) are clearly lower than the remaining triads, which contain notes that do not belong to C major.

Table 3.4: Euclidean (d) and cosine (θ) tonal dispersion of several triads relatively to the C major scale. The C major diatonic triads are presented in bold.

d	**b°**	**C**	**a**	**G**	c#°	D	A	B
	4.56	4.56	4.56	4.71	5.60	5.79	6.13	7.54
θ	**C**	**a**	**G**	**b°**	D	c#°	A	B
	0.92	0.92	0.97	1.07	1.33	1.43	1.44	2.01

To study how measuring tonal dispersion can be useful in discriminating the style of Western classical music, we first conduct a brief analysis on a set of four pieces from each style period: 1st Movement from A. Vivaldi's *Concerto No. 3 in F major Op. 8, RV 293, "Autumn"*, 3rd Movement from W.A. Mozart's *Piano Sonata No. 11 in A major, K. 331*, J. Brahms' *Rhapsody in G minor Op. 79 No. 2* and A. Schoenberg's *Drei Klavierstücke Op. 11 No. 2*. Figure 3.4 relates the diatonicity of each piece to its mean Euclidean and cosine tonal dispersion.

Vivaldi's and Mozart's works are the most diatonic, with the piece diatonicity vector further away from the center of the circle. The concentration of the by-frame diatonicity around a particular region of the circle suggests that these pieces tend more towards a specific tonality. This is further confirmed by the fact that these two pieces have the lowest cosine tonal dispersion values, which indicates that the set of pitch classes played throughout deviates less from the tonal center.

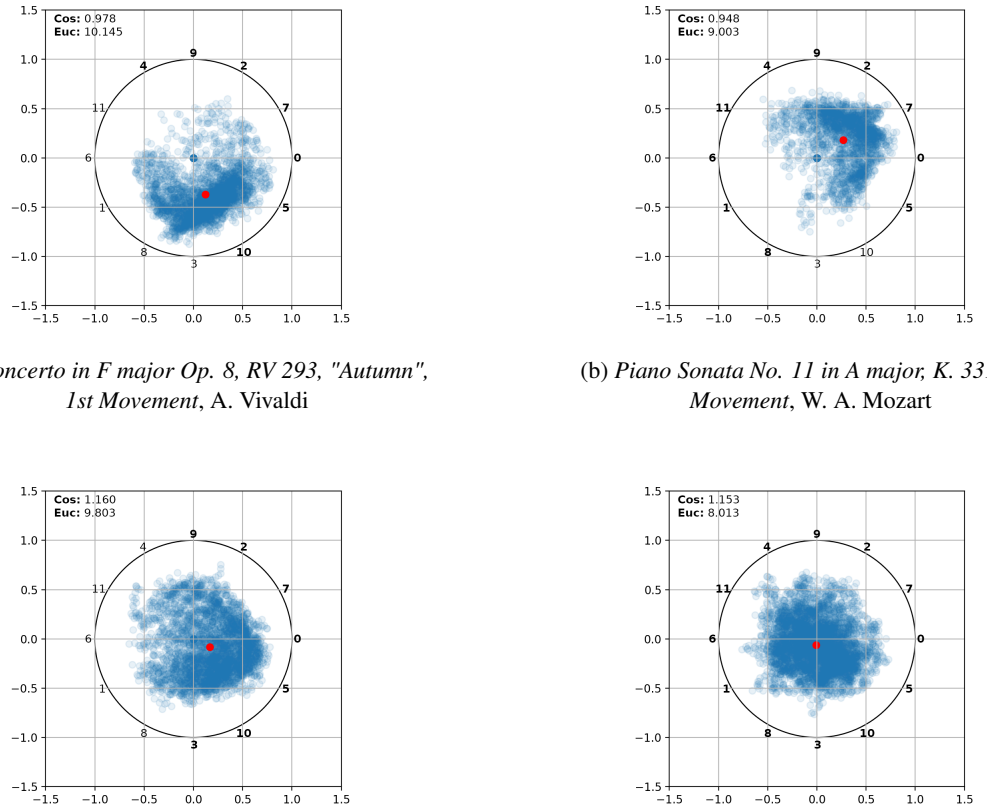
In contrast, the works by Brahms and Schoenberg appear much less diatonic, as their piece diatonicity vectors are much closer to the center. Their by-frame diatonicity is not particularly concentrated towards any part of the circle, which is even more evident in Schoenberg's piece. Moreover, these two pieces show higher cosine tonal dispersion values.

The Euclidean tonal dispersion is more difficult to interpret in these results. The lowest value is attributed to Schoenberg's piece, which could possibly be caused by its atonal nature: since it lacks a tonal center that leans towards a certain tonality, any pitch configuration might be seen as being closer to the tonal center in terms of shared pitch classes and interval content.

As an additional analysis, we compute the average tonal dispersion value per piece, using an analogous approach to the one adopted in Section 3.1.2. Figure 3.5 plots the Euclidean and cosine tonal dispersion values for each style period.

In general, pieces from the Modern period seem to exhibit lower Euclidean tonal dispersion values than the rest, which may be attributed to the common use of atonality during this period. Moreover, the higher spread of the Modern period's Euclidean tonal dispersion values could indicate increased stylistic freedom.

In contrast, the cosine tonal dispersion follows a nearly inverse trend. The lower values are located in the Baroque and Classical periods, which could be explained by the more tonal nature of



(a) *Concerto in F major Op. 8, RV 293, "Autumn", 1st Movement*, A. Vivaldi

(b) *Piano Sonata No. 11 in A major, K. 331, 3rd Movement*, W. A. Mozart

(c) *Rhapsody in G Minor Op. 79 No. 2, J. Brahms*

(d) *Drei Klavierstücke Op. 11 No. 2*, A. Schoenberg

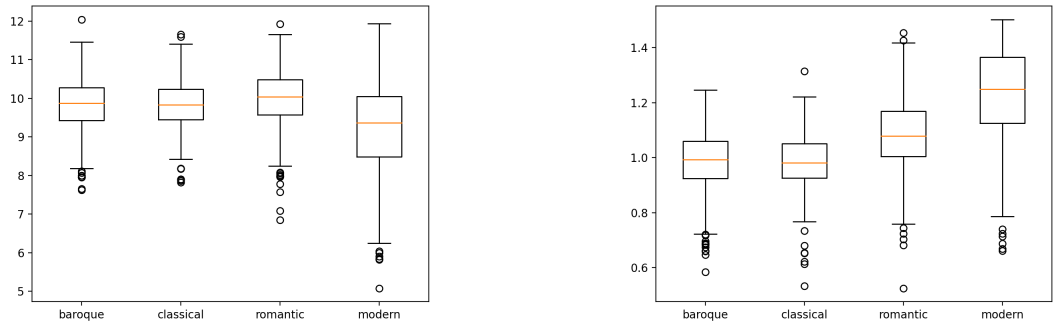
Figure 3.4: Diatonicity of each frame (in blue) and of the overall piece (in red) for four pieces from different style periods. The numbers around the circles represent each of the twelve pitch classes and those that belong to the tonality of the piece are presented in bold. The piece by Mozart (b) starts in the C major tonality and later changes to A major, therefore the pitch classes belonging to both tonalities are shown in bold in this case. In the case of Schoenberg's piece (d), all twelve pitch classes are shown in bold due to its dodecaphonic nature. The mean Euclidean and cosine tonal dispersion for each piece is shown in the top left corner.

the pieces. The higher values from the Romantic period onward can be attributed to the increased presence of modulations, which were simultaneously larger in general as well.

Overall, the Euclidean tonal dispersion may be useful to distinguish the Modern period from the rest. In contrast, the cosine tonal dispersion might help discriminate the Baroque and Classical periods from the rest and the Romantic from the Modern period. However, neither metric appears to distinguish the Baroque from the Classical period very well.

3.1.5 TIV Entropy

Amiot [2] proposes to study complexity in musical manifestations through the concept of information entropy. In detail, he explores the entropy of pitch-class set Fourier magnitudes as a



(a) Euclidean tonal dispersion

(b) Cosine tonal dispersion

Figure 3.5: Euclidean and cosine tonal dispersion for each style period.

measure of musical complexity. He defines Fourier entropy $H(X)$ as the Shannon entropy of the L_1-normalized Fourier coefficient magnitudes of a pitch-class set X:

$$H(X) = \sum_{k=1}^{n-1} -p_k \log p_k, \quad p_k = \frac{|a_k|^2}{\sum_{j=1}^{n-1} |a_j|^2} \tag{3.8}$$

Where p_k is the set of normalized Fourier coefficient magnitudes of X (sums to unity). The Fourier entropy value is maximal when X contains only a single unique pitch class or all twelve pitch classes and minimal when it represents a whole-tone scale 3.5.

With this notion in mind, we introduce the concept of TIV entropy. The TIV coefficient magnitudes $\|T(k)\|, 1 \leq k \leq 6$, come from the magnitudes that result from applying the DFT. Therefore, we define the entropy of a TIV as the entropy of its set Z of coefficient magnitudes $\|T(k)\|$:

$$H(Z) = \sum_{k=1}^{6} -p_k \log p_k, \quad p_k = \frac{\|T(k)\|}{\sum_{j=1}^{6} \|T(j)\|} \tag{3.9}$$

Table 3.5 shows a comparison between the Fourier and TIV entropy values for different pitch-class sets. To calculate the TIV entropy, we represent the pitch class sets as binary chroma vectors, where pitch classes that are present have the value 1 or 0 otherwise.

A few key observations can be drawn here: the TIV entropy of the whole-tone set is higher than the octatonic and diminished seventh sets, which does not verify for their Fourier entropy. A possible explanation for this might be the symmetric nature of the octatonic scale and the diminished seventh chord seemingly being given more importance in the TIS. The Fourier entropy value of the chromatic chunk is equal to that of the pentatonic and diatonic scales. However, the last two exhibit a more evident structural organization from a musical perspective. This aspect is better captured by the TIV entropy, which is higher for the chromatic chunk. Lastly, the maximal Fourier entropy corresponds to the chromatic scale, whereas the highest TIV entropy is assigned

Table 3.5: Fourier and TIV entropy of several pitch class sets, sorted by increasing order of the latter.

Pitch class set	Elements	Fourier Entropy	TIV Entropy
Octatonic	{0, 1, 3, 4, 6, 7, 9, 10}	0.6931	0.3551
Diminished seventh	{0, 3, 6, 9}	0.6931	0.3552
Whole tone	{0, 2, 4, 6, 8, 10}	0	0.5268
Diatonic	{0, 2, 4, 5, 7, 9, 11}	1.4698	1.2444
Pentatonic	{0, 2, 4, 7, 9}	1.4698	1.2972
Balanced chord	{0, 1, 4, 7, 8}	1.7916	1.3005
Harmonic minor	{0, 2, 3, 5, 7, 8, 11}	2.0565	1.4927
Minor seventh	{2, 5, 8, 12}	2.0693	1.5542
Single note	{3}	2.3979	1.6767
Redoubled note	{5, 5}	2.3979	1.6767
All-interval chord	{0, 1, 4, 6}	2.3394	1.6856
Chromatic - 1	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}	2.3979	1.6906
Chromatic chunk	{0, 1, 2, 3, 4}	1.4698	1.7204

to the chromatic chunk set.

In general, the TIV entropy feature can capture meaningful properties related to the organization of pitch-class configurations.

3.1.6 Harmonic Rhythm

In a musical piece, harmonic rhythm refers to the rate at which chord changes happen. Harmonic rhythm is a relevant aspect of music that can help differentiate between classical style periods. For example, pieces written during the baroque period tend to have a fast harmonic rhythm [29]. With this in mind, we employ the HCDF proposed by Ramoneda et al. [47] to capture information pertaining to harmonic rhythm in musical works by extracting the peaks at which changes in the harmony occur according to a peak picking function W_n defined as follows:

$$W_n = \xi_n : (\xi_{n-1} < \xi_n) \wedge (\xi_n > \xi_{n+1}) \tag{3.10}$$

The peak picking function picks all values of the HCDF ξ_n where the previous and the next value are lower. Figure 3.6 plots the HCDF of a baroque piece by G. Handel against a romantic piece by R. Wagner.

The most noticeable difference is in the magnitude of the harmonic changes, which is, in general, much higher in the piece by G. Handel. Additionally, the HCDF for this piece contains more peaks which are also closer together, strongly suggesting it has a faster harmonic rhythm than the piece by R. Wagner.

We now propose a set of features that rely on the HCDF. Let Ξ_m denote the set of peak frame indexes that result from applying a peak picking function to the HCDF ξ_n. We define the inter-peak interval Δ, a descriptor that provides an indicator of harmonic rhythm, as the difference in

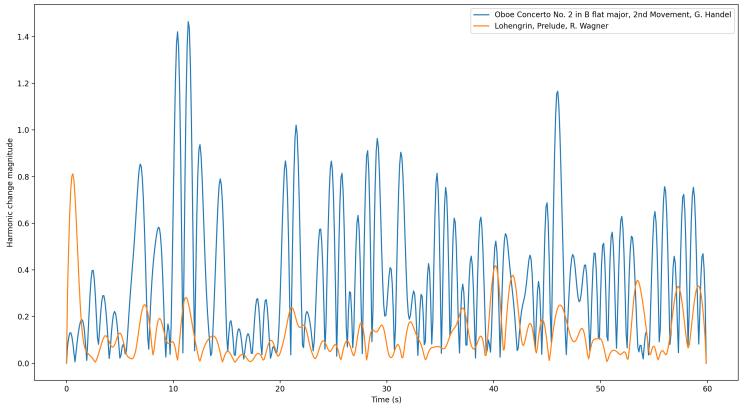


Figure 3.6: Comparison between the HCDF of two pieces from the Baroque and Romantic periods: 2nd Movement from the Oboe Concerto No. 2 in B flat major by G. Handel (in blue), and Prelude from Lohengrin by R. Wagner (in orange), respectively.

frames between consecutive peaks of the HCDF:

$$\Delta = \Xi_{m+1} - \Xi_m \tag{3.11}$$

In addition, we consider the magnitude of these peaks ξ_{Ξ_m} as an additional relevant descriptor, as it captures the degree of harmonic change between peaks.

3.2 Analysis of Descriptors in the Tonal Interval Space

Along this chapter, we have described and proposed a set of tonal audio features based on the properties of the TIS, which we list in Table 3.6.

To study TIS tonal features in discriminating the style of Western classical music, in this section, we present different analyses of their inter-relationship. Moreover, we compare the newly proposed TIS features with the tonal descriptors by Weiß [56] and how they relate to each other in terms of relevancy in classification tasks. Using the *Crossera-Full* dataset, we compute all features with a 100ms resolution and calculate their mean and standard deviation per piece.

First, we evaluate the correlation distances between descriptors by performing hierarchical clustering. We include both sets of descriptors in order to understand how they might describe different aspects of tonality.

Figure 3.7 reports a hierarchical clustering model which displays the Spearman correlation distances between descriptors using Ward's linkage. This model divides the descriptors into two main clusters: the green one appears to predominantly contain descriptors in the TIS, with only 3 out of 13 (23%) belonging to Weiß' model; the yellow cluster mainly contains descriptors proposed by Weiß, with 4 out of 18 descriptors (22%) belonging to the TIS. This observation suggests that the two sets of descriptors are indeed complementary and evaluate different characteristics of

Table 3.6: Summary of audio features based on the TIS.

Feature	Description
Chromaticity	Measures the presence of chromatic pitch distributions
Dyadicity	Measures the presence of tritones, fifths and stacked perfect fourths
Triadicity	Evaluates the presence of stacked thirds and augmented triads
Diminished Quality	Indicates the presence of diminished seventh chords
Diatonicity	Measures the presence of diatonic aggregates
Whole-toneness	Evaluates the presence of the whole-tone scale
Dissonance	Indicates the perceived dissonance of a given sonority
Inter-frame Euclidean distance	Measures shared pitch classes and interval content between consecutive audio frames
Inter-frame cosine distance	Measures the number of shared pitch classes between consecutive audio frames
Euclidean tonal dispersion	Measures the degree of deviation from the tonal center in terms of shared pitch classes and interval content
Cosine tonal dispersion	Measures the degree of deviation from the tonal center in terms of the number of shared pitch classes
TIV entropy	Describes the level of organization of a given sonority
HCDF inter-peak interval	Measures the harmonic rhythm
HCDF peak magnitude	Measures the magnitude of the harmonic changes

tonality. An additional observation is that the distances between descriptors in the green cluster generally seem larger, which points to a lower internal correlation between descriptors in the TIS. Looking at the leaf nodes of the tree. The proximity between the IC5 and diatonicity descriptors, although not necessarily desirable, does make sense, as both evaluate the presence of a perfect fifth/fourth interval in a pitch class distribution. The inter-frame Euclidean and cosine distances also appear highly correlated, and the HCDF peak interval descriptor also shows some correlation to these two distance metrics. This might suggest a possible relationship between the harmonic rhythm and the smoothness of the voice leading.

In order to understand the influence of different feature resolutions in the correlation between descriptors, Figure 3.8 repeats the same experiment but with a 10s resolution instead (further details on feature resolution and audio segmentation strategies are provided in Section 4.1). Similarly to Figure 3.7, we observe two clusters. In this case, however, the yellow cluster is quite smaller and appears to contain mostly features in the TIS (6 out of 7). The coupling between some descriptors also appears to have increased. For example, Tonal Complexity Features appear even closer together in this experiment. Distances between descriptors in the TIS and those proposed by Weiß still remain quite large, which reinforces the conclusion that the two sets of descriptors are complementary.

Another relevant experiment is to include different descriptive statistics to understand how these might affect the inter-descriptor distances. Therefore, in addition to the piece-wise mean of each descriptor, we also include its standard deviation (Figure 3.9). The first observation is that

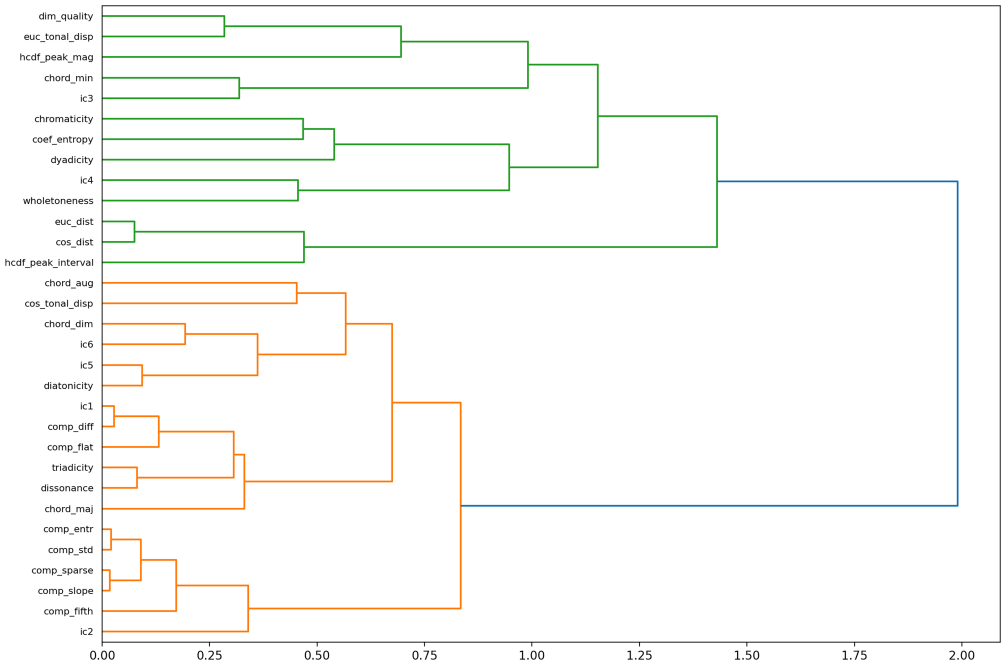


Figure 3.7: Hierarchical clustering of descriptors. Each descriptor is computed at a 100ms resolution and its value is averaged for each piece.

each statistic tends to stay close together, which indicates that they are complementary. The red cluster mainly contains standard deviation values (58%) and descriptors from the TIS (70%); the green cluster is almost entirely comprised of standard deviation values (90%) but shows an even distribution of descriptors from each set (50% for the TIS and for Weiß's; the yellow cluster is, for the most part, made up of means (70%) and descriptors proposed by Weiß (78%).

Additional resolutions for the previous clustering experiments are available under Section A.1 of Appendix A.

Finally, we perform a 10-fold cross-validation using a logistic regression classifier on the piece-wise mean and standard deviation of all descriptors computed at a 100ms resolution to rank them according to their relative importance in a style period classification task (Figure 3.10).

Out of the five highest-ranked descriptors, three belong to the TIS. However, the sum of the importance of the descriptors proposed by Weiß is 6.9% higher than that of the TIS. This could indicate that this set of descriptors overall contributes slightly more to the prediction of style period.

3.3 Summary

This chapter explored the potential of the TIS for computing descriptors capable of discriminating between style periods of classical music. We proposed the addition of the following descriptors based on the TIS: inter-frame Euclidean/cosine distance, Euclidean/cosine tonal dispersion, TIV

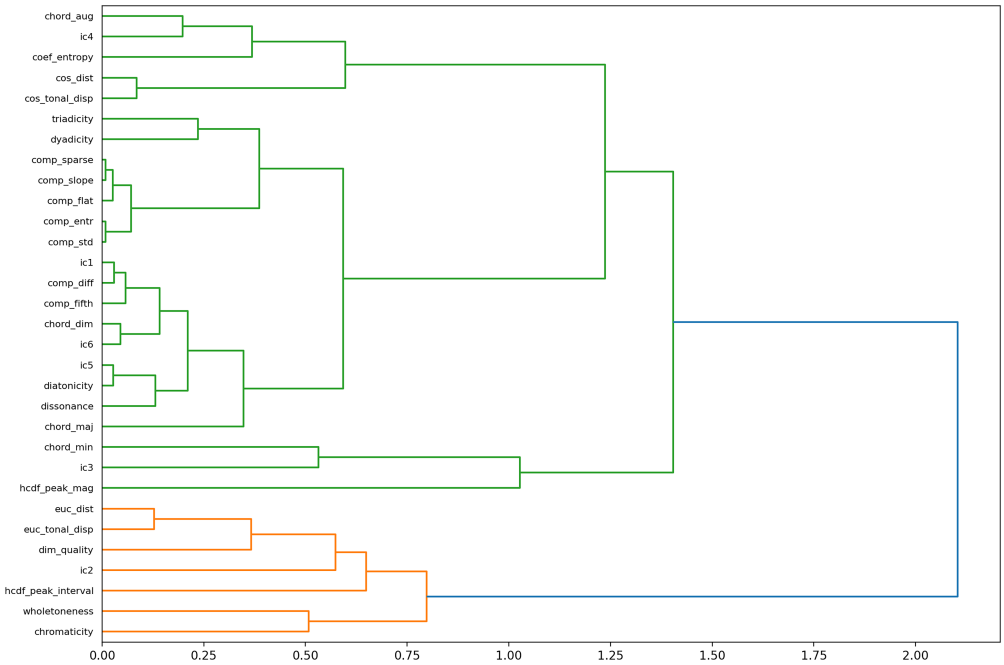


Figure 3.8: Hierarchical clustering of descriptors. Each descriptor is computed at a 10s resolution and its value is averaged for each piece.

entropy, HCDF inter-peak distance, and HCDF peak magnitude. Descriptors in the TIS show some degree of robustness for distinguishing between pieces and style periods. We concluded that the base TIS descriptors, as well as the additional ones, are, for the most part, complementary to those proposed by Weiß, which constitutes a possible indicator that they can contribute to improving the performance of the current state-of-the-art model for musical style identification.

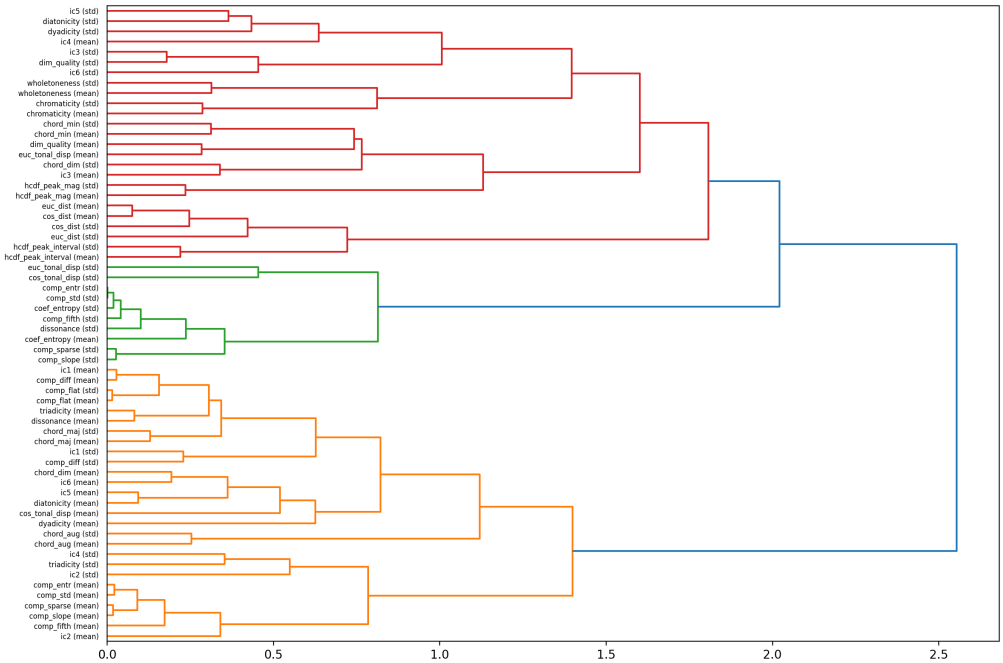


Figure 3.9: Hierarchical clustering of descriptors. Each descriptor is computed at a 100ms resolution, and we display its mean and standard deviation per piece.

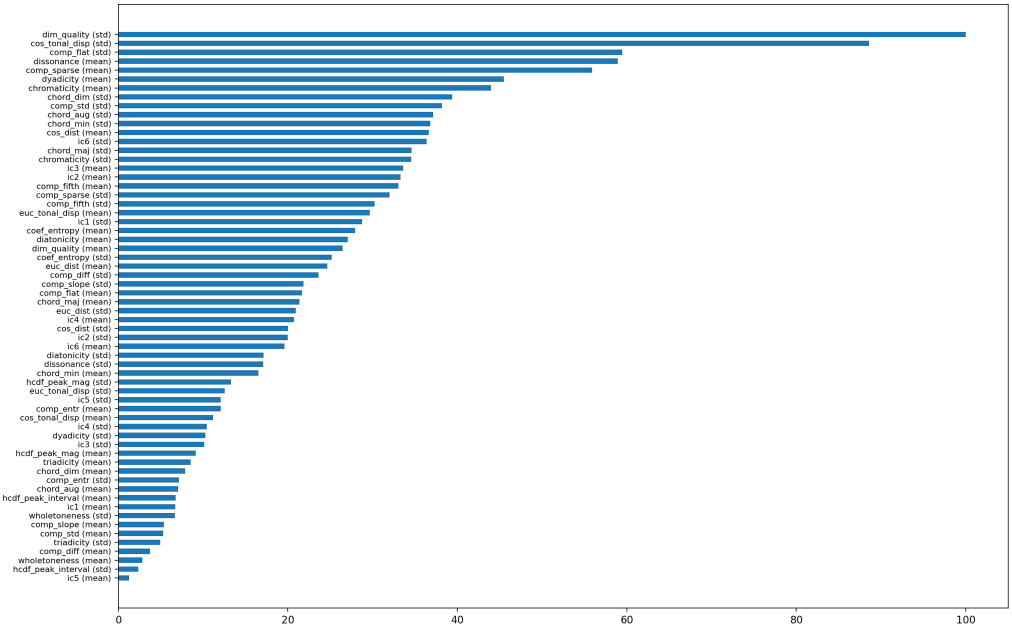


Figure 3.10: Relative importance of TIV, Template-based and Tonal Complexity features computed at a 100ms resolution.

Chapter 4

A Classification Model for Musical Style Identification

In this chapter, we present the implementation of a computational system for identifying musical style, specifically classical style period (Baroque, Classical, Romantic and Modern) and composers, using tonal audio features. Figure 4.1 defines the architecture of our system for musical style identification.

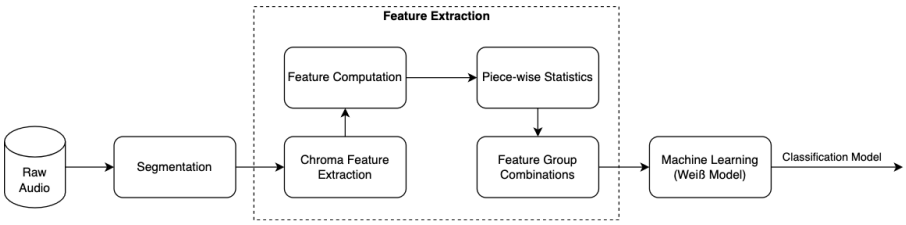


Figure 4.1: Architecture diagram of the proposed system for musical style identification.

The system uses raw audio files as input and focuses on discriminating musical style based on harmonic structure. It is comprised of three modules: the Segmentation module, which is responsible for splitting the audio into smaller segments that can be analyzed; the Feature Extraction module, which extracts audio features from each audio segment; the Machine Learning module, which trains models for predicting classical style period and composers. The system is implemented using the Python programming language and is available as open-source software[1].

In the following sections, we describe each of the modules in detail. Section 4.1 discusses the two audio segmentation strategies supported by the system: fixed-time segmentation on multiple temporal resolutions and harmonic structural segmentation. Section 4.2 details the system's harmonic audio feature extraction process and proposes groupings of features according to the musical properties they capture. In Section 4.3, we describe the state-of-the-art musical harmonic-driven style identification model by Weiß [56] on which we establish a baseline system for the task.

[1] https://github.com/fcfalmeida/style-ident

Lastly, Section 4.4 concludes the chapter by summarizing the main contributions of the proposed system.

4.1 Audio Segmentation

The segmentation of the musical audio input is a preprocessing step prior to the extraction of audio features. The size of the analysis segments from which harmonic features can be extracted ultimately exposes the multiple levels of the hierarchical structure of the musical audio. Therefore, the choice of time scale in the feature extraction process influences the type of musical properties that are captured. As the time resolution increases, these range from the pulse to tone, and finally texture [48]. Harmonically, this translates to aspects such as individual notes and chords, chord progressions, and modulations. We adopt two different segmentation strategies for the computation of audio features. Section 4.1.1 discusses the use of multiple fixed-time resolutions and the type of musical structures they capture. Section 4.1.2 presents a segmentation strategy in which the analysis window size is determined according to harmonic changes. This approach is motivated by the fact that the use of structurally-aware segmentation in the feature extraction process of the computational analysis of musical audio has been discussed in previous literature and shown to improve the accuracy of such systems when compared to using fixed window sizes [5].

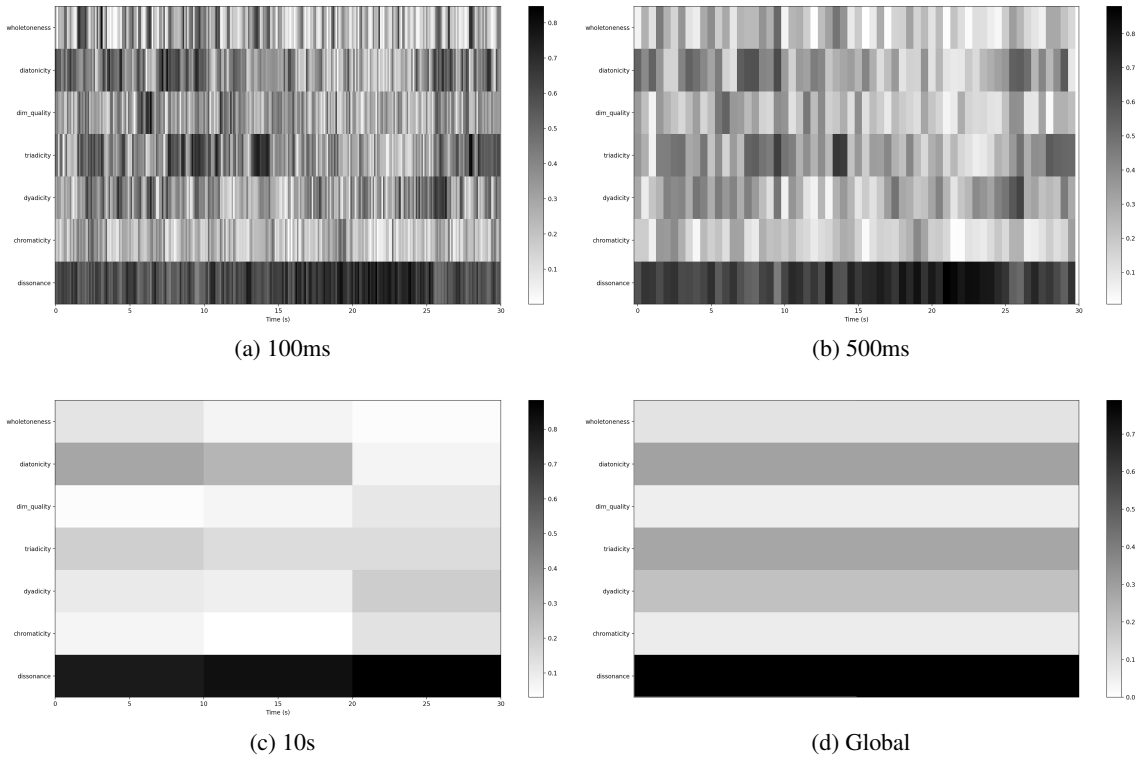
4.1.1 Fixed-time Segmentation with Multiple Resolutions

This strategy consists in segmenting the musical audio at multiple temporal resolutions with equal segment duration. Following [56], we capture different harmonic hierarchical features by adopting four time resolutions: 100ms, 500ms, 10s, and global (the entire duration of the piece). While smaller windows isolate finer musical details such as the notes, intervals, or chords, larger windows capture coarser structures such as chord progressions, tonality, and modulations. The hierarchical nature of tonality makes the analysis of music at different scales pivotal for style identification [56]. Figure 4.2 illustrates the extraction of TIV features at each of the four fixed-time resolutions.

4.1.2 Harmonic Structural Segmentation

Harmonic structural segmentation is a strategy that involves computing the Ramoneda et al. [47] HCDF on the musical audio signal to infer the points where function peaks denote harmonic changes (i.e., notes, chords, or key changes). To this end, we adopt the publicly available implementation of the HCDF[2] and use the frame indexes of the peaks to segment the musical audio. Conversely to the fixed-time segmentation approach, this strategy accounts for the musical audio structure and will undoubtedly result in segments with different durations. Figure 4.3 shows the extraction of TIV features using harmonic structural segmentation.

[2] https://github.com/PRamoneda/HCDF



(a) 100ms

(b) 500ms

(c) 10s

(d) Global

Figure 4.2: TIV audio features computed for a 30s excerpt of F. Liszt's *La Campanella* at four different temporal resolutions: 100ms, 500ms, 10s, and global.

In a more detailed analysis, Section 5.2 explores how this strategy compares to the fixed-time segmentation approach described in Section 4.1.1 when applied to classical style period and composer recognition.

4.2 Feature Extraction

In this section, we explain how the proposed system uses raw audio segments to extract the features presented in Chapter 3, as well as the Template-based and Tonal Complexity features proposed by Weiß [56]. We further organize these features into seven feature groups according to the type of musical information they capture. Table 4.1 lists these feature groups, presenting their constituent features and the types of segmentation that can be used to compute them.

The TIV Basic group comprises the six TIV coefficients and the dissonance descriptor. The TIV Complexity group contains the features that rely on distance metrics as well as the TIV entropy. The Harmonic Rhythm group contains the features that capture the rhythm and magnitude of harmonic changes. Some feature groups (TIV Basic, TIV Complexity) can be computed both in multiple fixed-time resolutions as well as using harmonic structural segmentation, others (Template Based, Tonal Complexity) are only computed using multiple fixed-time resolutions, and others (Harmonic Rhythm) exclusively using harmonic structural segmentation.

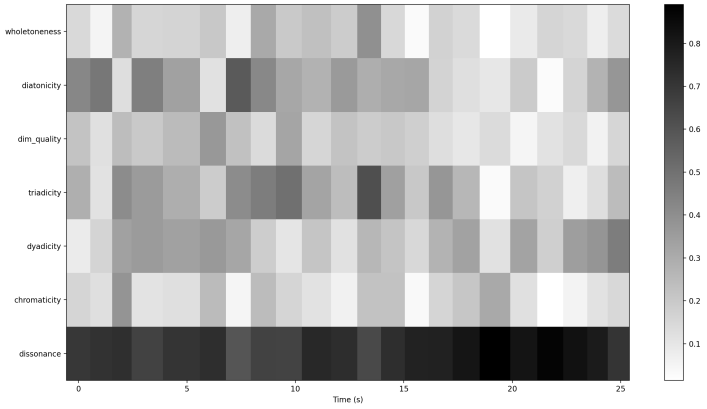


Figure 4.3: TIV audio features computed for an excerpt of F. Liszt's *La Campanella* using harmonic structural segmentation. Unlike frame-based segmentation, this approach produces segments of different sizes.

The system follows a four-step process to compute audio features: the first step in this module is the extraction of NNLS chroma features [39] from each audio segment. For the fixed-time segmentation approach, the system utilizes the NNLS chroma Vamp plugin[3]. The plugin takes the window length (w) and hop size (H) as input parameters, both expressed in samples. To extract NNLS chroma features at a 100ms resolution considering a sampling rate of 44.1KHz, we use $w = 8192$ and $H = 4410$. However, to avoid recomputing these features, we use a precomputed set of 100ms NNLS chroma features and downsample them to match the remaining temporal resolutions. For the 500ms resolution, this implies summing every five chroma vectors; for a 10s resolution, we sum every 100 chroma vectors; finally, for a global resolution, we sum all of the chroma vectors of a piece. The result from this step is a dataframe, a tabular data structure defined by the *pandas*[4] Python library, where each row contains a chroma vector for a given timestamp. The second step consists in using these chroma vectors to compute all remaining tonal features presented in Table 4.1. Third, the system computes mean and standard deviation descriptive statistic metrics for each feature per piece. These descriptions are then used as musical audio representations in the system for both training and testing. Finally, we combine multiple features per group. The optimal way of taking advantage of this capability is to first let the system compute all feature groups described in Table 4.1 and then either manually create new combinations or let the system exhaustively generate all possible ones, which for a total of 7 groups gives us 127 different combinations. This step is computationally efficient and makes any combination readily available for later testing.

[3] http://www.isophonics.net/nnls-chroma
[4] https://pandas.pydata.org/

Table 4.1: Feature groups with their constituent features and supported segmentation types.

Group	Features	Segmentation
TIV Basic	- Chromaticity - Dyadicity - Triadicity - Diminished-quality - Diatonicity - Whole-toneness - Dissonance	Harmonic and Multiple Resolutions
TIV Complexity	- Euclidean/cosine inter-frame distance - Euclidean/cosine tonal dispersion - TIV entropy	Harmonic and Multiple Resolutions
Harmonic Rhythm	- HCDF peak interval - HCDF peak magnitude	Harmonic
Template-based	- IC1 to IC6 - Triad types (maj, min, dim, aug)	Multiple Resolutions
Tonal Complexity	- Sum of chroma differences - Chroma standard dev. - Negative slope - Chroma entropy - Non-sparseness - Flatness - Angular dev.	Multiple Resolutions

Weiß's Template-based and Tonal Complexity features were implemented following their description in [56]. TIV features adopt the TIV.lib [46], an open-source library[5] for the baseline description of musical audio signals in the TIS. This library is a Python implementation of the TIS and allows the computation of several of the descriptors previously described in Chapter 3, namely diatonicity, chromaticity, whole-toneness, and dissonance. Additionally, it allows the calculation of the Euclidean or cosine distance between two TIVs.

One of the main contributions of our work is the implementation of novel tonal descriptors which were included in the TIV.lib library. We expanded the library with additional harmonic quality indicators such as dyadicity, triadicity, and diminished-quality. Moreover, we implement the newly proposed TIV features, specifically inter-frame Euclidean and cosine distance, Euclidean and cosine tonal dispersion, and TIV entropy.

The feature computation relies heavily on the capabilities of the *numpy*[6] library to define vectorized operations between and within arrays. This allows simultaneous operations between elements of two or more arrays or between elements of the same array and eliminates the need to directly use loops, thus optimizing the efficiency of the computation, which is highly desirable in the current context of large musical audio data processing.

[5] https://github.com/aframires/TIVlib
[6] https://numpy.org/

4.3 Weiß Model

In this section, we briefly detail the style period and composer classification model proposed by Weiß [56], which we use as a baseline model to expand and assess novel tonal features. In particular, we outline the system used for the training and testing process of an SVM classifier, which the following three steps can briefly describe. First, we perform dimensionality reduction on the set of extracted features using the LDA algorithm. Next, a grid search optimizes the hyper-parameters of the SVM classifier. Finally, using the optimal parameters, we train another SVM classifier which is used for predicting classical style period and composer. We repeat the entire procedure multiple times for each feature group to evaluate the stability of the results.

Because the source code of the model is not publicly available, we re-implement it as closely as possible to this description using the *scikit-learn*[7] library. Despite the fact that additional classifiers are used in the original work [56], we only implement this procedure using SVM, as it is, in general, the classifier that performs best.

4.4 Summary

Throughout this chapter, we covered the architecture and implementation of the proposed system for musical style identification. The system can essentially be divided into three modules: audio segmentation, feature extraction, and the machine learning model. Concerning audio segmentation, we discussed and implemented two approaches, a fixed-time segmentation strategy using multiple temporal resolutions, and harmonic structural segmentation. The feature extraction module contains all of the logic pertaining to the computation of the proposed audio features, as well as those proposed by Weiß [56]. We grouped audio features into sets according to the musical aspects they evaluate, which will later be used for training and testing the classification model. Finally, we described the state-of-the-art classical style period and composer classification model [56] and provided insights on its implementation. The main contributions highlighted during this chapter were the classical style period and composer identification system using Weiß's [56] implementation as a baseline, a new structural audio segmentation approach based on harmonic change peaks and the addition of novel tonal descriptors to an open-source implementation of the TIS.

[7] https://scikit-learn.org/

Chapter 5

Evaluation

In this chapter, we detail the evaluation of the proposed model, whose novelty in relation with the baseline system in [56] relies on the adoption of novel high-level tonal features. In greater detail, we aim to assess how the higher-level TIV features presented in Chapter 3 compare to the Template-based and Tonal Complexity features proposed by Weiß [56] in the context of musical style classification. To this end, we adopt the state-of-the-art model by Weiß and design several experiment trials that inspect the impact of segmentation strategies and feature groups in style classification accuracy. Section 5.1 details the general classification procedure followed throughout this chapter. Section 5.2 evaluates different types of musical audio segmentation. Section 5.3 discusses the results of style period and composer classification tasks. Section 5.4 pursues the latter experiment while applying an artist filter in order to prevent model overfitting. Finally, Section 5.5 concludes the chapter by summarizing the results from the classification experiments.

5.1 Experimental Setup

The classification procedure used to conduct the experiments uses the balanced (same number of entries for each class) subsets created from the NNLS chroma *Cross-Era* and *Cross-Composer* datasets (presented in Table 2.2) for era and composer classification, respectively. From the chroma vector data at multiple temporal resolutions extracted from these subsets, we compute features grouped according to Table 4.1. Then, we calculate their piece-wise mean and standard deviation and aggregate the data in the multiple feature groups. Finally, this data is fed into the baseline model proposed by Weiß, which can be split into the following seven steps:

1. Stratified cross-validation split into three folds, one for testing and two for training.

2. Using the two training folds, train an LDA classifier and use it to transform all three folds, reducing the number of feature dimensions.

3. Perform a five-fold cross-validation grid search on the two training folds in order to optimize SVM parameters C and γ.

4. Train the SVM classifier using the two training folds and the optimal parameters determined during step three.

5. Classify the test fold instances.

6. Repeat steps two to five another two times, each time using a different fold as the test fold.

7. Repeat steps one to six another nine times, each time with re-initialized cross-validation folds.

First, the data is split into three folds, from which one fold is used for testing and the remaining two folds for training. The split is stratified, meaning that, if possible, each fold is guaranteed to have the same number of elements from each class. Next, we adopt the LDA algorithm to reduce the feature space of all three folds, reducing the dimensionality of the data to $L = Z - 1$, where Z represents the number of unique classes. For the *Cross-Era-Piano*, *Cross-Era-Orchestra* and *Cross-Era-Full*, this implies $L = 3$, for the *Cross-Comp-11* subset $L = 10$, and for the *Cross-Comp-5* subset $L = 4$.

Having prepared the data, we consider an SVM classifier and perform a five-fold cross-validation grid search on the two train folds. This step attempts to optimize two of the SVM parameters to the classification task. The C parameter controls the error penalty of the algorithm, while γ is a parameter specific to certain kernel functions that can be applied in the context of this classifier. Following the approach described in [16], we use $C = 2^{-5}, 2^{-3}, ..., 2^{15}$ and $\gamma = 2^{-15}, 2^{-13}, ..., 2^{3}$ as the sets of values that are tested for each parameter during the grid-search procedure. Next, an SVM classifier is trained on the two training folds using the optimal parameters determined during the previous step. This step is then used to classify the instances in the test fold. For the SVM classifier, we use an RBF kernel when only the Template-based and Tonal Complexity features are considered in order to replicate Weiß's test cases. For the cases, where TIS features are adopted, we employ a linear kernel instead. This decision stems from a previous grid search on some of the feature groups, for which this type of kernel performed best in most cases. This additional test was conducted to avoid a larger number of optimization parameters during the grid search, thus enhancing computational efficiency in the training phase.

In order to reduce the effect the cross-validation split might have on the classification accuracy, and to ensure more consistent results, steps two to five are repeated twice more with a different test fold, and steps one to six are repeated another nine times with re-initialized cross-validation folds.

To gauge the classification performance of the model, we calculate the following metrics:

- **Mean classification accuracy:** We calculate the mean classification accuracy over the three folds and average it over the 10 runs.

- **Inter-run deviation:** Standard deviation of the mean accuracy values for each run, which measures the stability of the results over different cross-validation partitionings.

- **Inter-fold deviation:** Standard deviation of the accuracy values across cross-validation folds. It measures the stability of the classification results within a cross-validation run.
- **Inter-class deviation:** Standard deviation of the individual accuracy values of each class. A high value in this metric indicates a bias towards certain classes, which is unwanted.

5.2 Influence of Different Types of Segmentation

Before performing the classification on a larger set of feature group combinations, we performed an evaluation to assess the accuracy of the model for the classification tasks when using TIV features. As segmentation approaches, we consider frame-based segmentation (100ms), fixed-time segmentation with multiple resolutions (100ms, 500ms, 10s, and global), and harmonic structural segmentation, as well as the last two approaches combined. Table 5.1 presents the classification accuracy for the TIV Basic, TIV Complexity, and Harmonic Rhythm feature groups, as well as the combination of these. In total, the experiment includes twenty classification runs. Six out of twenty had the best result using MR segmentation and nine the MR + HS strategy. The features in the Harmonic Rhythm group are only calculated using harmonic structural segmentation because the timestamps of the values of these features will necessarily correspond to the time boundaries of the harmonic structural audio segments, that is, the segments are determined based on the timestamps of the harmonic change peaks.

Combining the fixed-time segmentation with multiple resolutions and the harmonic structural segmentation approaches only results in a very slight improvement to the classification accuracy in some cases. Considering that the harmonic structural segmentation process is computationally expensive, we opt for the MR segmentation strategy in subsequent experiments, using harmonic structural segmentation only for the Harmonic Rhythm feature group.

5.3 Style Period and Composer Classification

This section details the evaluation of classification experiments on an extended set of feature group combinations. For Weiß's features, we consider the Template-based and Tonal Complexity groups in addition to their combination. This makes it possible to compare each group individually to those in the TIS. We also consider all TIV feature groups and their combination. Due to the poor performance of Harmonic Rhythm features, observed in Table 5.1, we additionally test a case with only the TIV Basic and TIV Complexity feature groups, without Harmonic Rhythm features. Finally, we test the combination of Weiß's and TIV features and display these results in Table 5.2.

In the *Cross-Era-Full* dataset the highest classification accuracy is obtained when using a combination of Weiß's and TIV features excluding the Harmonic Rhythm group, with a nearly 2% improvement when compared to using only Weiß's combined features. In the *Cross-Era-Piano* and *Cross-Era-Orchestra* subsets, using only Template-based and Tonal Complexity features leads to slightly better accuracy. Results for the subsets of the *Cross-Composer* dataset are generally worse

Table 5.1: Classification accuracy for different types of segmentation.

	Frame-based	**Multiple Resolutions (MR)**	**Harmonic (HS)**	**MR + HS**
		Cross-Era-Piano		
TIV Basic	71.49%	**80.10%**	64.99%	79.05%
TIV Comp.	69.97%	73.24%	62.16%	**73.65%**
H. Rhythm	-	-	37.95%	-
Combined	77.25%	**81.79%**	71.03%	81.40%
		Cross-Era-Orchestra		
TIV Basic	76.64%	83.56%	71.88%	**84.63%**
TIV Comp.	77.03%	79.02%	72.53%	**79.51%**
H. Rhythm	-	-	39.95%	-
Combined	83.30%	85.41%	77.83%	**85.79%**
		Cross-Era-Full		
TIV Basic	70.54%	79.84%	66.87%	**79.98%**
TIV Comp.	68.64%	72.57%	67.20%	**73.45%**
H. Rhythm	-	-	37.97%	-
Combined	76.35%	81.77%	72.84%	**81.78%**
		Cross-Comp-11		
TIV Basic	49.54%	**59.55%**	40.27%	59.45%
TIV Comp.	46.52%	50.44%	36.39%	**51.02%**
H. Rhythm	-	-	15.56%	-
Combined	57.89%	**63.67%**	48.40%	61.77%
		Cross-Comp-5		
TIV Basic	65.48%	**74.78%**	57.50%	74.52%
TIV Comp.	60.94%	67.18%	56.74%	**67.76%**
H. Rhythm	-	-	31.28%	-
Combined	71.92%	**75.98%**	68.70%	74.86%

for all feature groups when compared to the *Cross-Era* dataset. Several possible factors contribute to this observation. First, the *Cross-Composer* dataset contains less items per class (100) when compared to the *Cross-Era* dataset (400). Second, the *Cross-Era* dataset contains more diverse pieces, essentially helping the model adapt to more varied characteristics of each class. Third, the higher number of classes creates a more difficult classification problem [56]. For the *Cross-Comp-11* dataset, the highest accuracy is attributed to the combination of TIV and Weiß's features, with a 1.64% improvement relative to Weiß's combined features. The higher-level TIV features complement Weiß's in the sense that they capture further musical dimensions such as horizontal structure and harmonic qualities. Overall, the addition of TIV features appears more beneficial in the larger, more diverse datasets when considering this metric, which may suggest they are more suitable for real classification scenarios. For that reason, we conclude that the additional dimensions they capture allow for a better identification of musical style.

Considering now the three deviation metrics, we find the results to be more varied. The inter-run deviation remains consistently low in general and is slightly higher in the *Cross-Composer* datasets. In four out of the five subsets, this metric is the lowest for TIV features. The inter-fold

deviation values appear more consistent across datasets. In the *Cross-Era-Full* and *Cross-Comp-11* datasets it is the lowest when using solely TIV features. For the *Cross-Era-Piano* and *Cross-Era-Orchestra*, the lowest value is obtained when combining Weiß's and TIV features without Harmonic Rhythm. Similar to what happens for the inter-run deviation, the inter-fold deviation in the *Cross-Comp-5* dataset is the lowest when using the Template-based feature group. The inter-class deviation value is mostly the lowest when only Weiß's features are used, although the difference is only slight when compared to the cases where Weiß's and TIV features are combined. Due to the nature of the dataset, the inter-class deviation values in the *Cross-Comp-11* and *Cross-Comp-5* subsets are quite higher in general.

We now compare the Weiß feature group that obtained the highest mean classification accuracy with the best feature group that includes TIV features in each dataset by looking at the percentage of correctly and incorrectly classified instances per class in order to establish a more detailed comparison between test cases where TIV features are included and cases in which they are not. To this end, Figure 5.1 shows the confusion matrices of several feature group combinations per dataset.

For each combination, the confusion matrix was obtained from a single cross-validation run, by performing a grid search and then training an SVM classifier with the optimal parameters, and not by taking into account the accuracy values over all runs of the classification procedure, which is why the mean accuracy values presented in Table 5.2 differ from the mean of the values in the diagonal of each matrix.

A consistent observation across all *Cross-Era* subsets is that there is almost always over 5% of instances from the Baroque period classified as Classical and over 10% of pieces from the Classical period classified as belonging to the Baroque era. In the *Cross-Era-Piano* and *Cross-Era-Full* subsets, the percentage of instances from the Modern period classified as Romantic is higher than 8% in all cases, and the same is true for the percentage of Romantic pieces classified as Modern. These observations are somewhat expected since they are relative to neighboring periods that share more stylistic traits. In terms of correctly classified instances, in the *Cross-Era-Piano* subset, the model that includes TIV features shows higher precision for the Baroque and Classical periods. Conversely, in the *Cross-Era-Orchestra* subset, TIV features seem to improve the number of correctly classified pieces from the Romantic and Modern periods. For the *Cross-Era-Full* dataset, despite showing higher mean classification accuracy (please refer to Table 5.2), the model including TIV features generally exhibits a lower accuracy for most classes except for the Classical period for which over 90% of the pieces are classified correctly.

In the *Cross-Comp-5* subset, we find some differences between the two feature group combinations with respect to misclassified instances. For example, with Weiß's combined features, 18.18% of pieces written by Beethoven are classified as having been written by Brahms, whereas using TIV features, this value drops to 6.06%. Conversely, 21.21% of pieces written by Beethoven are wrongly attributed to Haydn when TIV features are included, a value that drops to 15.15%, when adopting Weiß's combined features only. Concerning the percentage of correctly classified pieces for each composer, the model that includes TIV features is only worse at classifying pieces

by Brahms.

Finally, due to its larger number of classes, the *Cross-Comp-11* represents the most difficult classification scenario. In this case, using a combination of Weiß's and TIV features seems to improve the classification accuracy per class. Regarding misclassified instances, the model using Weiß's combined features only shows lower accuracy in classifying neighboring composers, i.e., composers whose lifetimes are closer to each other (e.g., Haydn and Mozart, Mendelssohn and Brahms, Dvorăk and Brahms). While the same also verifies, albeit to a somewhat lesser degree, when TIV features are included, with this combination of features the model appears to mostly confuse composers which are further apart from each other and even from different style periods (e.g., Mozart and Handel, Beethoven and Mendelssohn, Mozart and Mendelssohn).

In sum, the use of TIV features provides an improvement in the *Cross-Era-Full* and *Cross-Comp-11* datasets. However, when the classification accuracy per class is considered, it performs better in identifying certain style periods and composers.

5.4 Classification with Filtering

Following Weiß's approach, in this section, we evaluate the impact of the partitioning of folds in the cross-validation procedure. In *Cross-Era* and *Cross-Composer* datasets, many pieces are taken from the same CD recording, which may lead to overfitting during the cross-validation procedure in cases where the train and test folds both contain tracks performed by the same interpreters [56]. To prevent the model from possibly adapting to the timbre, recording settings, or stylistic traits of the performers, we repeat the experiments in the previous section and apply a filter that forces the pieces from the same composer (in the case of the *Cross-Era* dataset) or interpreter (in the case of the *Cross-Composer* dataset) to be placed in the same fold [56]. Table 5.3 shows the classification results with filtering.

In general, the classification accuracy values with filtering drop substantially when compared to the previous experiment. This observation is expected as the filtering produces more homogeneous folds, thus creating a much more difficult classification problem. Test cases that include TIV features show an improvement in classification accuracy in four of the five datasets. For the *Cross-Era-Orchestra* dataset, the combination of TIV Basic, TIV Complexity, and Harmonic Rhythm feature groups improves the accuracy by almost 2% when compared to the highest accuracy obtained using Weiß's features (77.19% compared to 75.25% for the Template-based features). In the *Cross-Era-Full* dataset, by combining Weiß's and TIV features (excluding harmonic rhythm), we obtain 74.04% accuracy when compared to the 71.16% when using only Weiß's combined features. For the *Cross-Comp-11* dataset, combining Weiß's and TIV features also slightly improves the accuracy. Finally, in the *Cross-Comp-5* we obtain a 4.74% improvement with TIV Basic features compared to Template-based features.

Similar to the previous section, we now analyze the confusion matrices considering the best performing model that includes TIV features and the best model that uses only Weiß's features (Figure 5.2).

For the *Cross-Era* dataset, the highest confusion is clearly between the Baroque and Classical periods. In the *Cross-Era-Piano* subset, over 50% of Classical pieces are misclassified as belonging to the Baroque period when using Template-based features, a value which drops to 14.93% when using the TIV Basic feature group. In terms of correctly classified pieces, the models including TIV features perform better for the *Cross-Era-Orchestra* and *Cross-Era-Full* subsets, which may be explained by the robustness of the TIS to timbral changes.

Similar to the confusion matrices presented earlier for the classification experiments without filtering in the previous section (Figure 5.1), in the *Cross-Composer* dataset the highest confusion is observed for composers with neighboring lifetimes. This is quite normal, as it is expected for such composers to share more stylistic traits than those whose lifetimes are more distant. Despite resulting in lower mean classification accuracy in both the *Cross-Comp-11* and *Cross-Comp-5* subsets, the percentage of correctly classified pieces per composer is almost always higher when using Template-based features.

Overall, the use of TIV features for the classification experiments with filtering has proven beneficial in terms of classification accuracy in most datasets, with an improvement of 1.94%, 2.88%, 0.84% and 4.74% in the *Cross-Era-Orchestra*, *Cross-Era-Full*, *Cross-Comp-11* and *Cross-Comp-5* subsets, respectively, thus suggesting they may be more suitable for musical style classification in real case scenarios.

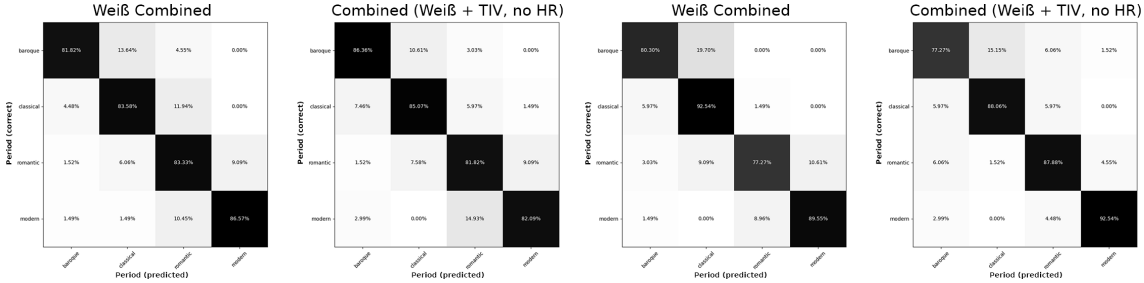
5.5 Summary

In this chapter, we compared the TIV features presented in Chapter 3 with the Template-based and Tonal Complexity features proposed by Weiß [56]. Several classification experiments using the system proposed in Chapter 4 were conducted. First, we compared the performance of harmonic and fixed-time audio segmentation strategies previously discussed in Section 4.1 in style classification tasks. We concluded that using fixed-time segmentation with multiple resolutions overall offers a compromised balance between computational performance and classification accuracy. Then, we conducted style period and composer classification experiments and observed that the use of TIV features results in an improvement to the classification accuracy in two out of the five datasets used for training and testing, the most significant improvement happening on the *Cross-Era-Full* subset (1.98%). These two datasets were the largest and the most heterogeneous, which may suggest that TIV features are more suitable for real classification scenarios, where there is typically a higher variability in the data. Finally, we repeated the style period and composer classification experiments with a filter strategy that forces pieces from the same composer or performer to be placed under the same fold during the cross-validation step, reducing the chance of overfitting and resulting in a harder classification problem. In this case, models trained with TIV features performed better in four out of the five datasets, the biggest improvement being on the *Cross-Comp-5* dataset (4.74%), which allowed us to further validate our conclusion that this type of features may be capable of performing better in real musical style classification scenarios. A possible explanation for these improvements may be the additional musical dimensions captured

by TIV features when compared to Weiß's, namely the harmonic structure captured by distance and harmonic rhythm features.

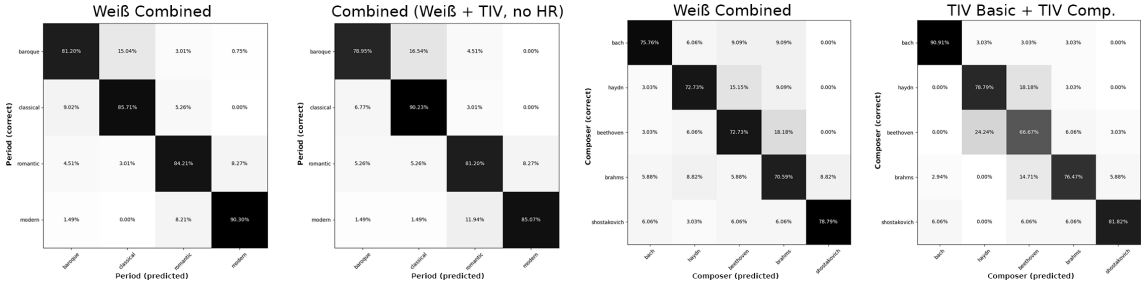
Table 5.2: Style period and composer classification results on the subsets of the *Cross-Era* and *Cross-Composer* datasets, respectively. We test several feature group combinations and present the mean accuracy, inter-run, inter-fold, and inter-class deviation metrics.

		Mean Accuracy	Inter-run dev.	Inter-fold dev.	Inter-class dev.
Cross-Era-Piano ($L = 3$)					
Weiß	**Template-based**	81.08%	0.93%	2.43%	**4.65%**
	Tonal Complexity	81.39%	0.74%	2.15%	5.25%
	Combined	**84.60%**	0.70%	2.12%	4.71%
TIV	**TIV Basic**	80.10%	0.69%	2.09%	5.13%
	TIV Complexity	73.24%	0.65%	1.83%	5.83%
	H. Rhythm	37.95%	**0.51%**	1.95%	13.30%
	Basic + Comp.	81.89%	1.16%	1.78%	4.83%
	Combined	81.79%	0.90%	1.79%	4.87%
Combined (Weiß + TIV)		83.55%	0.96%	1.89%	5.48%
Combined (Weiß + TIV, no HR)		83.94%	0.87%	**1.39%**	5.23%
Cross-Era-Orchestra ($L = 3$)					
Weiß	**Template-based**	84.81%	0.80%	1.68%	5.01%
	Tonal Complexity	83.30%	0.80%	1.98%	6.23%
	Combined	**86.50%**	0.87%	1.55%	**4.08%**
TIV	**TIV Basic**	83.56%	**0.52%**	1.85%	5.74%
	TIV Complexity	79.02%	0.91%	1.54%	5.99%
	H. Rhythm	39.95%	0.61%	2.13%	8.22%
	Basic + Comp.	85.71%	1.05%	1.84%	4.40%
	Combined	85.41%	0.95%	1.67%	4.60%
Combined (Weiß + TIV)		85.64%	0.80%	1.62%	4.60%
Combined (Weiß + TIV, no HR)		85.96%	0.59%	**1.33%**	4.68%
Cross-Era-Full ($L = 3$)					
Weiß	**Template-based**	81.36%	0.77%	1.32%	4.53%
	Tonal Complexity	78.41%	0.91%	1.25%	5.26%
	Combined	83.83%	0.32%	0.88%	3.82%
TIV	**TIV Basic**	79.84%	**0.28%**	**0.80%**	4.92%
	TIV Complexity	72.57%	**0.28%**	1.25%	4.48%
	H. Rhythm	37.97%	0.60%	1.64%	11.38%
	Basic + Comp.	81.91%	0.36%	1.09%	4.73%
	Combined	81.77%	0.31%	1.03%	4.83%
Combined (Weiß + TIV)		85.64%	0.44%	0.93%	3.53%
Combined (Weiß + TIV, no HR)		**85.81%**	0.46%	1.07%	**3.41%**
Cross-Comp-11 ($L = 10$)					
Weiß	**Template-based**	60.54%	0.81%	1.43%	12.38%
	Tonal Complexity	59.83%	0.84%	2.14%	13.96%
	Combined	67.32%	0.64%	1.19%	**10.16%**
TIV	**TIV Basic**	59.55%	1.01%	2.39%	13.17%
	TIV Complexity	50.44%	0.67%	1.67%	12.94%
	H. Rhythm	15.56%	**0.62%**	**0.94%**	14.40%
	Basic + Comp.	63.19%	1.31%	2.06%	12.02%
	Combined	63.67%	1.24%	2.03%	11.56%
Combined (Weiß + TIV)		**68.96%**	1.11%	1.83%	10.27%
Combined (Weiß + TIV, no HR)		68.40%	0.75%	1.89%	10.19%
Cross-Comp-5 ($L = 4$)					
Weiß	**Template-based**	74.64%	**0.60%**	**1.81%**	11.05%
	Tonal Complexity	74.42%	1.31%	2.37%	9.78%
	Combined	**77.38%**	1.66%	2.66%	7.66%
TIV	**TIV Basic**	74.78%	1.01%	2.01%	9.65%
	TIV Complexity	67.18%	1.66%	2.36%	9.81%
	H. Rhythm	31.28%	1.57%	2.44%	10.27%
	Basic + Comp.	76.34%	1.01%	1.98%	6.83%
	Combined	75.98%	1.03%	2.38%	7.25%
Combined (Weiß + TIV)		73.04%	1.85%	1.86%	**6.46%**
Combined (Weiß + TIV, no HR)		73.58%	2.28%	2.02%	6.60%



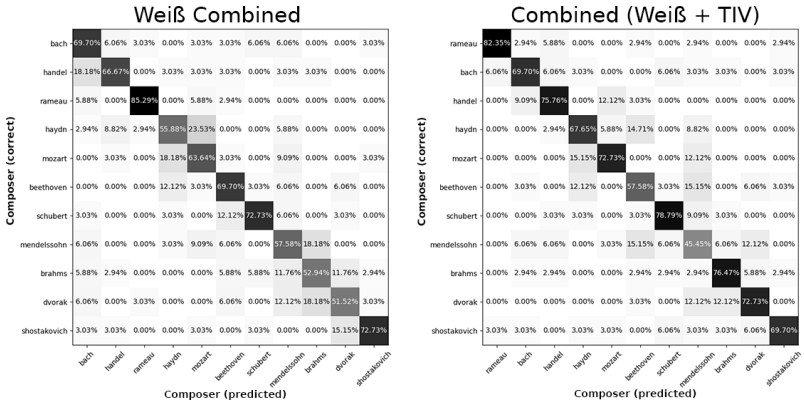
(a) Cross-Era-Piano

(b) Cross-Era-Orchestra



(c) Cross-Era-Full

(d) Cross-Comp-5

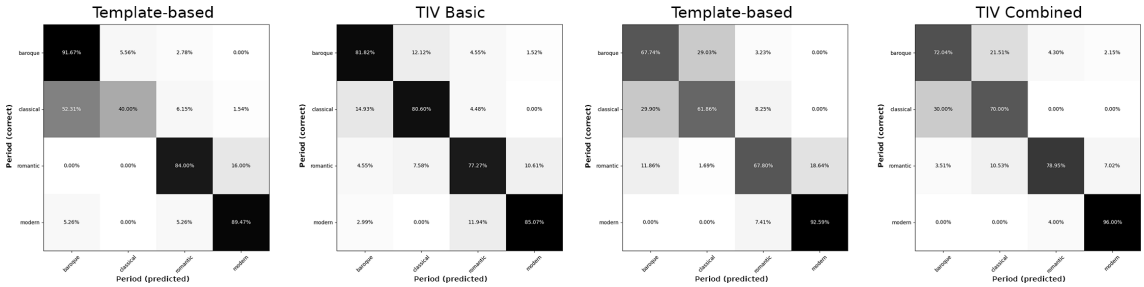


(e) Cross-Comp-11

Figure 5.1: Confusion matrices for the various *Cross-Era* and *Cross-Comp* subsets. For the *Cross-Era* subsets, style periods are sorted chronologically. Similarly, for the *Cross-Comp* subsets, the names of the composers are displayed in order according to their lifetime. The color of each matrix cell varies according to its respective percentage value. Higher percentages are displayed in a darker shade.

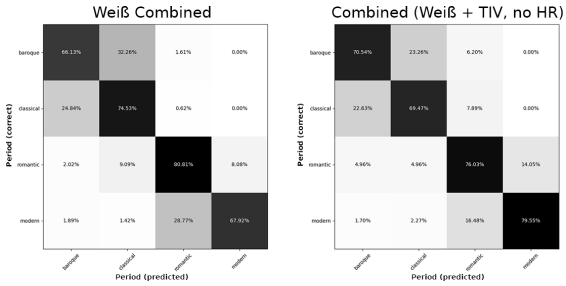
Table 5.3: Style period and composer classification results with composer and artist filtering on the subsets of the *Cross-Era* and *Cross-Composer* datasets, respectively.

		Mean Accuracy	Inter-run dev.	Inter-fold dev.	Inter-class dev.
Cross-Era-Piano ($L = 3$)					
Weiß	**Template-based**	**69.82%**	3.39%	6.28%	13.45%
	Tonal Complexity	65.84%	4.19%	5.08%	**12.80%**
	Combined	67.77%	3.56%	5.06%	15.36%
TIV	**TIV Basic**	66.84%	**1.69%**	4.57%	15.31%
	TIV Complexity	57.99%	3.22%	5.38%	17.67%
	H. Rhythm	21.26%	4.60%	6.41%	28.06%
	Basic + Comp.	65.59%	2.04%	4.66%	16.77%
	Combined	65.47%	2.33%	4.49%	16.63%
Combined (Weiß + TIV)		64.39%	1.86%	4.10%	16.27%
Combined (Weiß + TIV, no HR)		64.78%	1.82%	**4.02%**	15.80%
Cross-Era-Orchestra ($L = 3$)					
Weiß	**Template-based**	75.25%	3.15%	6.56%	12.16%
	Tonal Complexity	71.68%	4.64%	5.85%	14.09%
	Combined	75.23%	3.16%	5.55%	12.17%
TIV	**TIV Basic**	74.80%	3.10%	3.70%	11.22%
	TIV Complexity	69.87%	3.80%	4.84%	12.87%
	H. Rhythm	28.31%	4.60%	**2.97%**	21.51%
	Basic + Comp.	76.68%	**2.76%**	3.74%	10.35%
	Combined	**77.19%**	**2.76%**	3.59%	**9.89%**
Combined (Weiß + TIV)		76.70%	3.14%	3.68%	11.41%
Combined (Weiß + TIV, no HR)		76.56%	3.67%	3.55%	11.29%
Cross-Era-Full ($L = 3$)					
Weiß	**Template-based**	70.51%	**1.80%**	4.21%	11.57%
	Tonal Complexity	65.51%	2.00%	4.97%	13.26%
	Combined	71.16%	2.04%	**3.78%**	11.42%
TIV	**TIV Basic**	70.13%	2.53%	4.00%	12.52%
	TIV Complexity	62.18%	2.22%	4.56%	13.71%
	H. Rhythm	21.65%	4.80%	6.11%	26.44%
	Basic + Comp.	71.50%	2.29%	4.19%	12.03%
	Combined	71.63%	2.58%	3.91%	12.03%
Combined (Weiß + TIV)		73.78%	2.33%	3.84%	10.80%
Combined (Weiß + TIV, no HR)		**74.04%**	2.38%	3.90%	**10.79%**
Cross-Comp-11 ($L = 10$)					
Weiß	**Template-based**	37.41%	2.50%	2.85%	19.96%
	Tonal Complexity	29.74%	1.95%	3.26%	21.33%
	Combined	36.97%	2.39%	2.98%	21.59%
TIV	**TIV Basic**	37.84%	1.81%	3.51%	20.66%
	TIV Complexity	29.61%	1.35%	2.85%	19.77%
	H. Rhythm	7.77%	**1.10%**	**1.19%**	**17.64%**
	Basic + Comp.	37.82%	1.35%	3.60%	20.30%
	Combined	37.83%	1.13%	3.35%	20.01%
Combined (Weiß + TIV)		**38.25%**	1.92%	4.31%	21.01%
Combined (Weiß + TIV, no HR)		37.89%	2.42%	4.22%	21.57%
Cross-Comp-5 ($L = 4$)					
Weiß	**Template-based**	50.01%	3.73%	10.85%	19.77%
	Tonal Complexity	43.32%	4.20%	8.08%	25.68%
	Combined	48.86%	3.72%	10.07%	22.47%
TIV	**TIV Basic**	**54.75%**	4.08%	10.00%	17.96%
	TIV Complexity	43.16%	2.81%	7.19%	19.06%
	H. Rhythm	16.47%	3.57%	**5.42%**	19.35%
	Basic + Comp.	53.40%	3.10%	9.86%	16.85%
	Combined	53.75%	3.13%	9.78%	**16.65%**
Combined (Weiß + TIV)		49.72%	**2.63%**	12.12%	19.25%
Combined (Weiß + TIV, no HR)		50.44%	3.11%	11.35%	18.72%

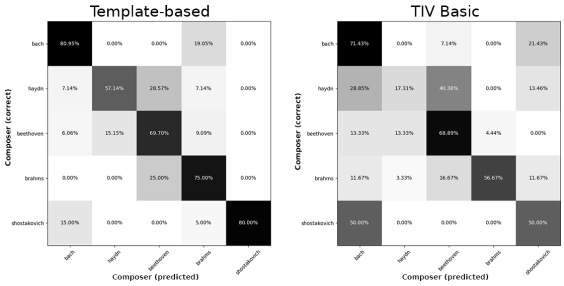


(a) Cross-Era-Piano

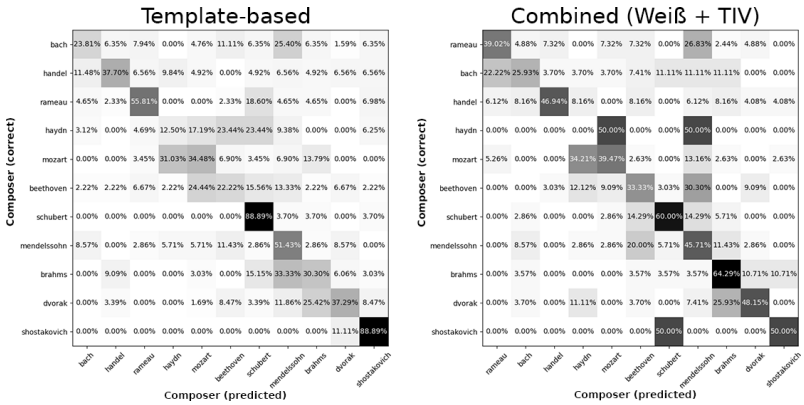
(b) Cross-Era-Orchestra



(c) Cross-Era-Full



(d) Cross-Comp-5



(e) Cross-Comp-11

Figure 5.2: Confusion matrices for the various *Cross-Era* and *Cross-Comp* subsets with artist filtering. For the *Cross-Era* subsets, style periods are sorted chronologically. Similarly, for the *Cross-Comp* subsets, the names of the composers are displayed in order according to their lifetime.

Chapter 6

Conclusions and Future Work

The increased availability of digital music content over the last decades has promoted the development of methods for organizing large online collections. Musical style is an important aspect when categorizing musical audio files. Existing models for style identification have prioritized timbre and rhythmic features over harmony or tonal features, which are nonetheless fundamental stylistic traits of music composition practice. A vast majority of computational models that consider harmony in style recognition, do not consider long-term horizontal structure and rely mostly on low-level features driven from the standard chroma vector representation. Moreover, most audio features in literature lack a perceptual basis that accounts for the way humans perceive distances between sonorities. A further limitation in current studies is that only a few focus specifically on style identification within classical music, where harmony is known to have markedly changed across eras (i.e., stylistic periods, such as Baroque or Classical).

With these considerations in mind, we proposed a style identification model to improve the current state-of-the-art by using a set of perceptually-inspired audio descriptors, based on the TIS, a perceptual pitch space proposed by Bernardes et al. [7]. We used the existing audio features of this pitch space, namely chromaticity, dyadicity, triadicity, diminished-quality, diatonicity, wholetoneness, and dissonance, and proposed new tonal features based on the properties of the TIS: Euclidean and cosine inter-frame distance, Euclidean and cosine tonal dispersion, TIV entropy, harmonic change peak interval, and harmonic change peak magnitude. As an additional contribution, the newly proposed audio features were added to an open-source implementation of the TIV [46].

Furthermore, we proposed a new harmonic structural audio segmentation approach based on harmonic change peaks and compared it with an existing fixed-time segmentation strategy with multiple temporal resolutions in classical style period and composer classification tasks. We observed that, in most cases, harmonic structural segmentation improved the classification accuracy slightly. However, the benefits do not outweigh the computational cost of this process.

At the core of our contribution lies the comparison of the Template-based and Tonal Complexity features by Weiß [56] with those proposed in the context of this dissertation, which adopt the TIS for computing descriptors that evaluate hierarchical harmonic structure, voice leading, tonal dispersion, dissonance, and entropy. We performed clustering experiments and evaluated the importance of these features for style period classification and concluded that TIV features capture musical aspects complementary to those captured by the features proposed by Weiß. The tonal features have also been assessed in style period and composer classification tasks adopting the *Cross-Era* and *Cross-Composer* datasets, from which five balanced subsets were derived. Using an SVM classifier, our novel feature set has shown that in classification experiments with filtering, which come close to real case scenarios, the classification accuracy improves in four of the five subsets, the most significant improvement being on the *Cross-Comp-5* dataset, with an increase of 4.74%. In classification experiments without the filtering procedure, the use of TIV features improves the accuracy on the larger and more diverse *Cross-Era-Full* and *Cross-Comp-11* datasets by 1.98% and 1.64%, respectively. From these results, we conclude that TIV features introduce performance benefits in more realistic classification scenarios where the available data is much more diverse and exists in larger amounts.

In future work, we highlight several aspects that should be addressed. First, the proposed model can be used for style classification of other musical genres other than classical music. We believe that gauging the performance of TIV features in other classification tasks may yield interesting results from the perspective of MIR research. Second, to evaluate the influence of the classification procedure and to better measure the potential of TIV features, it could be beneficial to experiment with different machine learning approaches, such as other classification algorithms or deep learning models. Third, the current implementation of the HCDF proposed by Ramoneda et al. [47] can be further optimized by exploiting the vectorization of operations provided by the *numpy* library. This optimization can greatly increase the efficiency of our system's harmonic structural segmentation process. Finally, the proposed system could eventually be integrated into online music streaming or hosting services to improve browsing and recommendations.

References

[1] Charles R. Adams. Melodic Contour Typology. *Ethnomusicology*, 20(2):179, 1976.

[2] Emmanuel Amiot. Entropy of fourier coefficients of periodic musical objects. *Journal of Mathematics and Music*, 15(3):235–246, 2021.

[3] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[4] Shlomo Argamon, Kevin Burns, and Shlomo Dubnov. *The structure of style: Algorithmic approaches to understanding manner and meaning*. Springer Berlin Heidelberg, 2010.

[5] Juan P. Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *ISMIR 2005 - 6th International Conference on Music Information Retrieval*, pages 304–311, 2005.

[6] Adam Berenzweig, Beth Logan, Daniel P.W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[7] Gilberto Bernardes, Diogo Cocharro, Marcelo Caetano, Carlos Guedes, and Matthew E.P. Davies. A multi-level tonal interval space for modelling pitch relatedness and musical consonance. *Journal of New Music Research*, 45(4):281–294, 2016.

[8] Gilberto Bernardes, Diogo Cocharro, Carlos Guedes, and Matthew E.P. Davies. Conchord: An application for generating musical harmony by navigating in the tonal interval space. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9617 LNCS, pages 243–260. Springer, Cham, 2016.

[9] Gilberto Bernardes, Matthew E.P. Davies, and Carlos Guedes. A Hierarchical Harmonic Mixing Method. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11265 LNCS, pages 151–170. Springer, Cham, 2018.

[10] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. Number January 2011, pages 591–596, 2011.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[12] Dmitry Bogdanov, Alastair Porter, Hendrik Schreiber, Julián Urbano, and Sergio Oramas. The acousticbrainz genre dataset: Multi-source, multi-level, multi-label, and large-scale. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, pages 360–367, 2019.

[13] Judith C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[14] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering*. Cambridge University Press, 2019.

[15] Pedro Cano, Emilia Gómez, Fabien Gouyon, and Perfecto Herrera. ISMIR 2004 Audio Description Contest. Technical report, Music Technology Group, Universitat Pompeu Fabra, 2006.

[16] Chih-Jen Lin Chih-Wei Hsu, Chih-Chung Chang. A Practical Guide to Support Vector Classification. *BJU international*, 101(1):1396–1400, 2008.

[17] Roger B Dannenberg. Style in music. In *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*, pages 45–57. Springer Berlin Heidelberg, 2010.

[18] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 316–323. International Society for Music Information Retrieval, 2017.

[19] Alessio Degani, Marco Dalai, Riccardo Leonardi, and Pierangelo Migliorati. Harmonic Change Detection for musical chords segmentation. In *Proceedings - IEEE International Conference on Multimedia and Expo*, volume 2015-Augus. IEEE Computer Society, 2015.

[20] J. Stephen Downie. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research, 2008.

[21] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[22] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. Music classification via the bag-of-features approach. *Pattern Recognition Letters*, 32(14):1768–1777, 2011.

[23] Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.

[24] Masataka Goto. Development of the RWC Music Database. In *Proceedings of International Congress on Acoustics*, pages I–553–556, 2004.

[25] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC Music Database: Popular, Classical, and Jazz Music Database. In *Proceedings on the 3rd International Conference on Information Music Retrieval (ISMIR 2002)*, pages 287–288, 2002.

[26] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. *AES 25th International Conference*, pages 1–9, 2004.

[27] Barbara Russano Hanning. *Concise History of Western Music*. W. W. Norton & Company, 1997.

[28] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the ACM International Multimedia Conference and Exhibition*, pages 21–26. ACM Press, 2006.

[29] Glen Haydon and Manfred F. Bukofzer. Music in the Baroque Era from Monteverdi to Bach. *The Journal of Aesthetics and Art Criticism*, 7(3):262, 1949.

[30] Steven R. Holtzman. The Circle of Fifths. In *Digital Mantras: The Languages of Abstract and Virtual Worlds*, pages 15–33. The MIT Press, 1994.

[31] Filip Korzeniowski and Gerhard Widmer. A fully convolutional deep auditory model for musical chord recognition. *CoRR*, abs/1612.05082, 2016.

[32] Honglak Lee, Largman Yan, Peter Pham, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pages 1096–1104, 2009.

[33] Fred Lerdahl. Tonal pitch space. *Music Perception: An Interdisciplinary Journal*, 5(3):315–349, 1988.

[34] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. page 282. Association for Computing Machinery (ACM), 2003.

[35] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Music genre database and musical instrument sound database. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, pages 229–230, 2003.

[36] Ugo Marchand and Geoffroy Peeters. The Extended Ballroom Dataset. *ISMIR 2016 Late-Breaking Session*, pages 1–3, 2016.

[37] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, pages 135–140, 2010.

[38] Brian McFee and Juan Pablo Bello. Structured training for large-vocabulary chord recognition. In Sally Jo Cunningham, Zhiyao Duan, Xiao Hu, and Douglas Turnbull, editors, *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 188–194, 2017.

[39] Meinard Müller. *Fundamentals of Music Processing*. Springer International Publishing, 2015.

[40] Meinard Müller, Wolfgang Bogler, and Vlora Arifi-Müller. Saarland Music Data (SMD). In *Proceedings of the Late-Breaking and Demo Session of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.

[41] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B Khalil, and Deepak Turaga. Learning feature engineering for classification. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 2529–2535, 2017.

[42] María Navarro-Cáceres, Marcelo Caetano, Gilberto Bernardes, Mercedes Sánchez-Barba, and Javier Merchán Sánchez-Jara. A computational model of tonal tension profile of chord progressions in the tonal interval space. *Entropy*, 22(11):1–30, 2020.

[43] Keith Negus. From creator to data: the post-record music industry and the digital conglomerates. *Media, Culture and Society*, 41(3):367–384, 2019.

[44] Jiyoung Park, Jongpil Lee, Jangyeon Park, Jung Woo Ha, and Juhan Nam. Representation learning of music using artist labels. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pages 717–724. International Society for Music Information Retrieval, 2018.

[45] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[46] António Ramires, Gilberto Bernardes, Mathew Davies, and Xavier Serra. TIV.lib: an open-source library for the tonal description of musical audio. *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-20), Vienna, Austria*, (September):304–309, 2020.

[47] Pedro Ramoneda and Gilberto Bernardes. Revisiting harmonic change detection. In *149th Audio Engineering Society Convention 2020, AES 2020*, 2020.

[48] Curtis Roads. *Microsound*. MIT Press, 2004.

[49] Justin Salamon and Emilia Gomez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759–1770, 2012.

[50] Justin Salamon, Bruno Rocha, and Emilia Gomez. Musical genre classification using melody features extracted from polyphonic music signals. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 81–84, 2012.

[51] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6959–6963. Institute of Electrical and Electronics Engineers Inc., 2014.

[52] John Thickstun, Zaid Harchaoui, and Sham M Kakade. Learning features of music from scratch. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[53] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[54] Laurens Van Der Maaten, Eric Postma, and Jaap Van Den Herik. Dimensionality Reduction: A Comparative Review. Technical report, Tilburg Centre for Creative Computing, 2009.

[55] Cheng I. Wang and George Tzanetakis. Singing Style Investigation by Residual Siamese Convolutional Neural Networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 2018-April, pages 116–120. Institute of Electrical and Electronics Engineers Inc., 2018.

[56] Christof Weiß. *Computational Methods for Tonality-Based Style Analysis of Classical Music Audio Recordings*. PhD thesis, Ilmenau University of Technology, 2017.

[57] Christof Weiß, Fabian Brand, and Meinard Müller. Mid-level chord transition features for musical style analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 341–345. IEEE, 2019.

[58] Christof Weiß, Matthias Mauch, and Simon Dixon. Timbre-invariant audio features for style analysis of classical music. In *Music Technology meets Philosophy - From Digital Echos to Virtual Ethos: Joint Proceedings of the 40th International Computer Music Conference, ICMC 2014, and the 11th Sound and Music Computing Conference, SMC 2014, Athens, Greece, September 14-20, 2014*. Michigan Publishing, 2014.

[59] Christof Weiß, Matthias Mauch, Simon Dixon, and Meinard Müller. Investigating style evolution of Western classical music: A computational approach. *Musicae Scientiae*, 23(4):486–507, 2019.

[60] Christof Weiß and Meinard Müller. Tonal complexity features for style classification of classical music. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 688–692. IEEE, 2015.

[61] Erling Wold and James Tenney. A History of Consonance and Dissonance. *Computer Music Journal*, 13(3):94, 1989.

[62] Jason Yust. Stylistic information in pitch-class distributions. *Journal of New Music Research*, 48(3):217–231, 2019.

[63] Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, 1(2):56–70, 2020.

[64] Xinquan Zhou and Alexander Lerch. Chord detection using deep learning. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015*, pages 52–58, 2015.

Appendix A

Additional Graphs and Tables

A.1 Hierarchical Clustering of Descriptors - Additional Time Resolutions

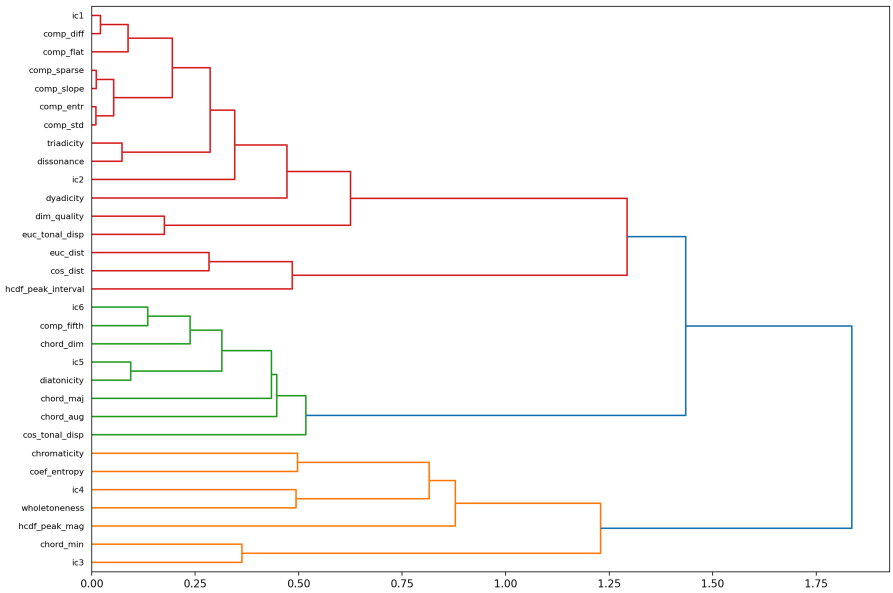


Figure A.1: Hierarchical clustering of descriptors. Each descriptor is computed at a 500ms resolution and its value averaged for each piece.

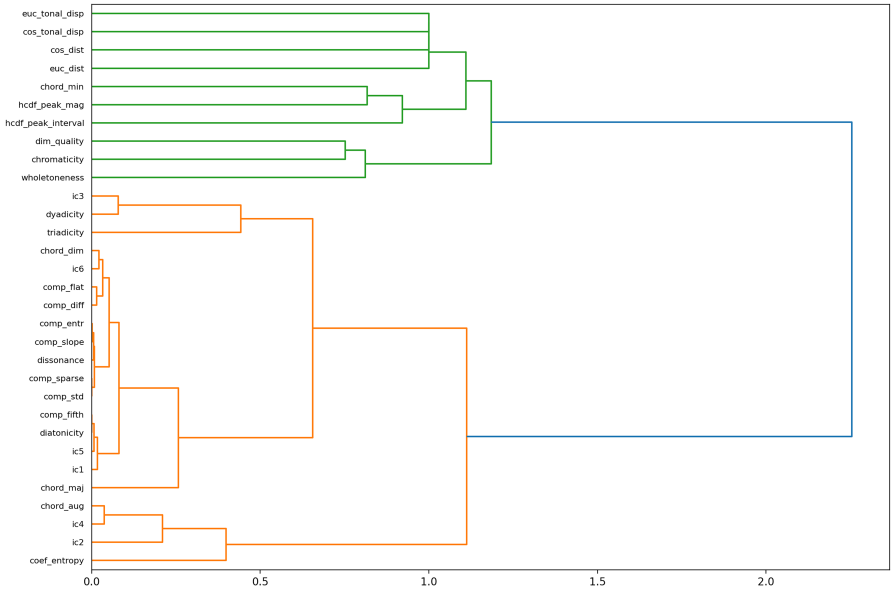


Figure A.2: Hierarchical clustering of descriptors. Each descriptor is computed at a global resolution (entire piece collapsed into a single chroma vector) and its value averaged for each piece.

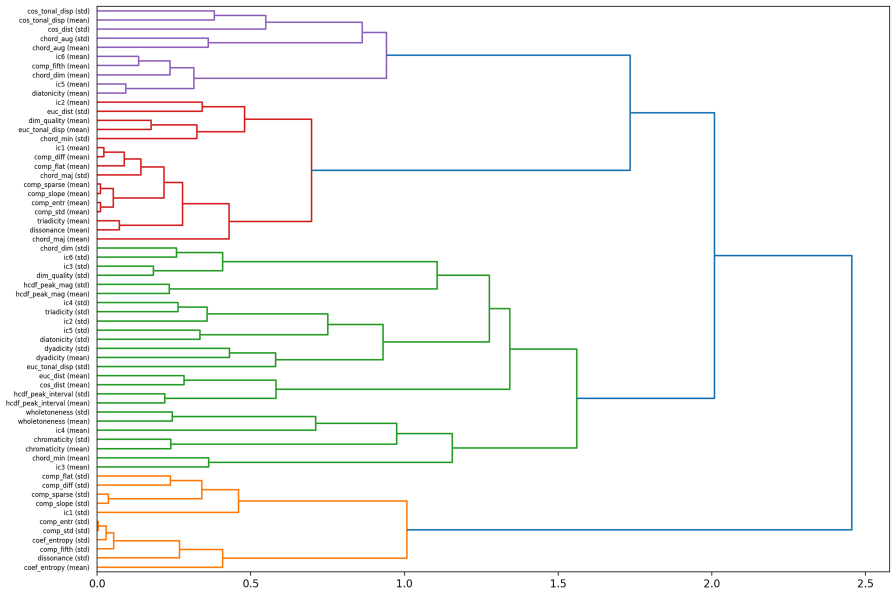


Figure A.3: Hierarchical clustering of descriptors. Each descriptor is computed at a 500ms resolution and we display its mean and standard deviation per piece.

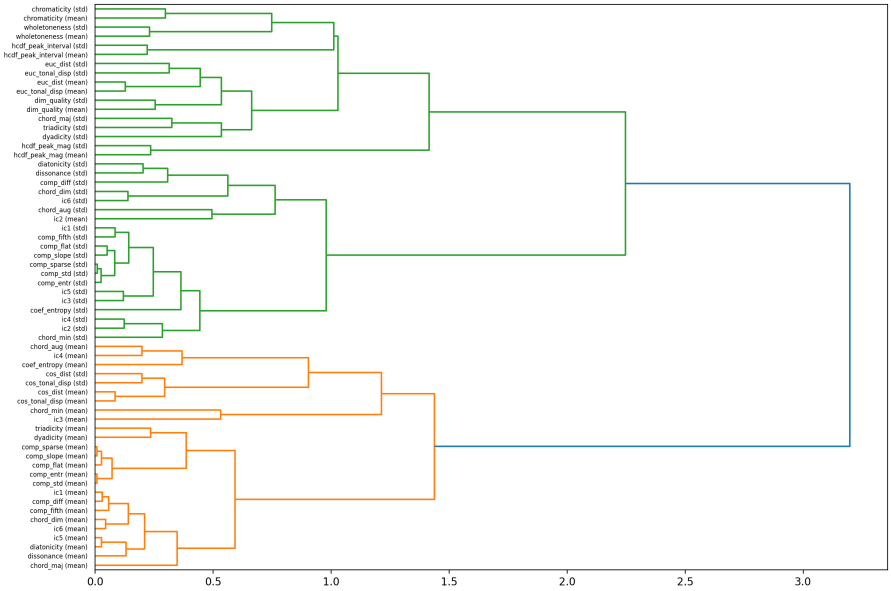


Figure A.4: Hierarchical clustering of descriptors. Each descriptor is computed at a 10s resolution and we display its mean and standard deviation per piece.

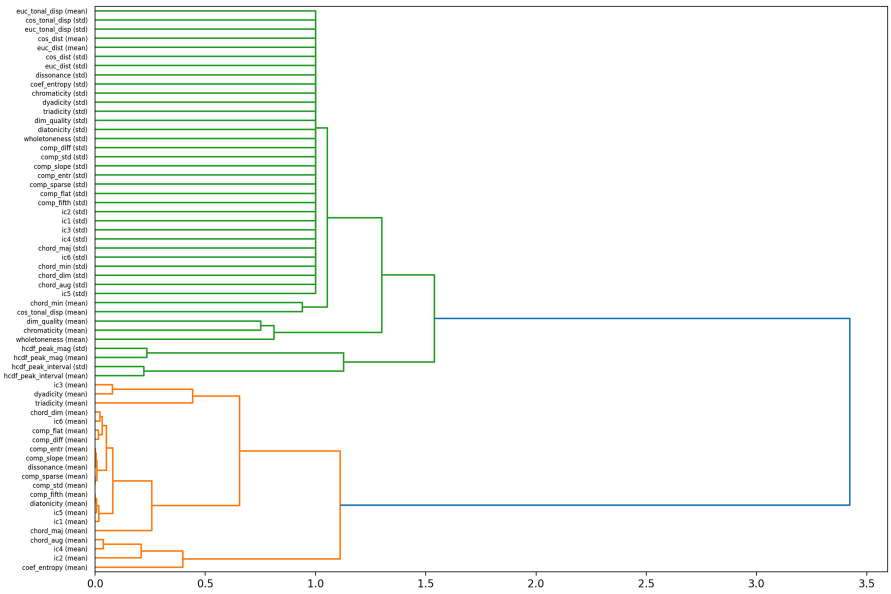


Figure A.5: Hierarchical clustering of descriptors. Each descriptor is computed at a global resolution and we display its mean and standard deviation per piece.