UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

FELIPE DE ALMEIDA LIMA

Exploração e Avaliação de Técnicas de Reconhecimento de Acordes em Gravações de Guitarra Elétrica

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo de Oliveira Johann
Coorientador: Prof. Dr. Rodrigo Schramm

Porto Alegre
2019

RESUMO

O presente trabalho tem como objetivo explorar diferentes técnicas na área de reconhecimento e classificação de acordes, tendo como foco gravações de áudio de guitarra elétrica. A partir de um algoritmo que utiliza um *pipeline* básico para reconhecimento de acordes, são feitos múltiplos experimentos em suas diferentes etapas, avaliando seus efeitos no resultado final do reconhecimento. O *pipeline* inicia com a segmentação do áudio em múltiplos quadros para processamento por FFT. São analisados diferentes tamanhos de janela e parâmetros de janelamento. Como segunda etapa, é feita a extração das notas musicais a partir do resultado da FFT. São comparadas diferentes técnicas para reduzir artefatos causados por sons harmônicos, principal fonte de erros da etapa. A terceira etapa do algoritmo efetua a acumulação das notas musicais num mapa conhecido como cromagrama, o qual contém 12 classes de semitons. Como etapa final, o cromagrama obtido é comparado com modelos de acorde para classificação. As diferentes técnicas são comparadas utilizando um conjunto de gravações de guitarra, considerando classificadores para os 24 acordes básicos (maiores e menores) e também com acordes estendidos (diferentes tipos de sétima) para um total de 60 acordes e a ausência de acorde, com acurácia de 68,9% para acordes do conjunto.

Palavras-chave: Detecção de acordes em guitarra. Recuperação de informação musical. Transcrição musical automática.

Exploration and Evaluation of Chord Recognition Methods on Electric Guitar Recordings

ABSTRACT

This work's objective is to explore different techniques on the chord recognition and classification field, with focus on audio recordings of electric guitar. From a basic chord recognition pipeline, multiple experiments are performed in its different stages, evaluating their effects on the final results of the recognition task. The pipeline begins with the segmentation of the audio into multiple frames for FFT processing. Different window sizes and windowing parameters are evaluated. As a second stage, the extraction of notes from the FFT's results is performed. Different overtone removal techniques are compared, as this is the main sources of errors of this stage. The third stage performs the accumulation of notes in a map known as chromagram, which contains 12 pitch classes. As the last stage, the resulting chromagram is compared to all chord templates for classification. The techniques are compared using a set of guitar recordings, considering classifiers for the 24 basic chords (major and minor) and also with an extended set of chords (different kinds of seventh chords) for a total of 60 chords and the absence of chords, resulting in an accuracy of 68,9% for chords in the set.

Keywords: Guitar chord detection. Music information retrieval. Automatic music transcription.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DFT	*Discrete Fourier transform* (transformada discreta de Fourier)
FFT	*Fast Fourier transform* (transformada rápida de Fourier)
RMS	*Root mean square* (valor eficaz)
maj	Acorde maior
min	Acorde menor
min7	Acorde de sétima menor
7	Acorde de sétima dominante
maj7	Acorde de sétima maior
NC	*No-Chord* (ausência de acorde)
f_0	Frequência fundamental de uma nota musical

SUMÁRIO

1 INTRODUÇÃO

1.1 Contexto e objetivos gerais

O reconhecimento de acordes é uma das mais importantes áreas no campo de recuperação de informação musical (*music information retrieval, MIR*) (MÜLLER, 2015, p. 293), sendo uma área de grande interesse da pesquisa internacional. Possui diversas aplicações práticas, como segmentação musical (BELLO; PICKENS, 2005), identificação de *covers* (LEE, 2006) e classificação de músicas de acordo com o humor (CHENG, 2008).

Além das aplicações já mencionadas, a detecção de acordes pode ser utilizada para fins educativos, como ferramenta de auxílio para estudantes de guitarra, conduzindo o estudante ao longo de uma música e permitindo que o estudante avalie se está tocando os acordes de forma correta.

1.2 Objetivos específicos

No contexto da detecção de acordes, este trabalho tem como propósito comparar e avaliar os efeitos de diferentes técnicas de reconhecimento de acordes já existentes, bem como explorar soluções alternativas, podendo servir de base para trabalhos futuros na área.

O trabalho é focado no reconhecimento de acordes no contexto de gravações de áudio de guitarra elétrica. A guitarra elétrica é um instrumento frequentemente utilizado como acompanhamento, conduzindo a progressão de acordes em diversos gêneros musicais populares como rock, country, reggae e jazz.

O foco instrumental específico em guitarra, embora torne a aplicação mais restrita, permite abstrair em grande parte certos problemas abordados por mecanismos de reconhecimento mais genéricos, como variações de timbre entre instrumentos e remoção de percussão, permitindo maior foco em outros desafios como a redução de artefatos causados por sobretons harmônicos.

1.3 Implementação

A aplicação foi implementada na linguagem C++, utilizando as bibliotecas *libsndfile*[1] para leitura de arquivos de áudio e *fftw*[2] para cálculo dos espectros FFT. Também foi utilizada durante o desenvolvimento a biblioteca *cairo*[3], para geração de imagens com gráficos para fins de análise. A linguagem C++ foi escolhida por sua eficiência e grande aplicabilidade prática.

O programa desenvolvido recebe como parâmetros obrigatórios um arquivo de áudio de entrada e o nome do arquivo de destino. O programa então gera como saída um arquivo texto onde cada linha contém o tempo do instante analisado e o nome do acorde reconhecido para aquele instante.

A aplicação possui também múltiplos parâmetros opcionais para as diferentes etapas do *pipeline* de detecção de acordes, permitindo controlar a forma de segmentação do áudio, a função de janelamento, o método de extração de notas, etc.

Como saída alternativa, o programa também gera imagens PNG com gráficos das diferentes etapas do *pipeline*. São gerados três gráficos ilustrativos para a *ativação de notas musicais*, *cromagrama* e *acordes detectados*, utilizando uma escala de cores para avaliar a intensidade da ativação de cada nota/classe de semitom/acorde, respectivamente, ao longo do tempo.

Foram implementados também scripts *Python* para o processamento de múltiplos arquivos, geração de banco de dados e avaliação dos resultados gerados pela aplicação e pelo sistema independente.

---

[1] Disponível em: http://www.mega-nerd.com/libsndfile/. Acessado em: 18 de julho de 2019
[2] Disponível em: http://www.fftw.org/ Acessado em: 18 de julho de 2019
[3] Disponível em: https://cairographics.org/ Acessado em: 18 de julho de 2019

1.4 Organização da monografia

O Capítulo 2, *Conceitos*, introduz conceitos da teoria musical e do processamento de áudio que são fundamentais para o entendimento do algoritmo implementado. O Capítulo 3, *Trabalhos Relacionados*, descreve trabalhos existentes na área de reconhecimento de acordes com propósito ou foco semelhante. O Capítulo 4, *Fluxo de Execução e Técnicas Avaliadas*, descreve as múltiplas etapas do *pipeline*, discutindo os desafios existentes em cada uma e as diferentes técnicas implementadas para comparação. O Capítulo 5, *Dados de Teste*, descreve o conjunto de dados usado para avaliação dos resultados. O Capítulo 6, *Análise dos Resultados*, compara o impacto das diferentes técnicas nos resultados, e compara o algoritmo como um todo a um sistema independente. Por fim, o Capitulo 7, *Conclusão*, resume as observações e traz sugestões para trabalhos futuros.

2 CONCEITOS

Certos conceitos da teoria musical e do processamento de áudio são fundamentais para o entendimento de um mecanismo de reconhecimento de acordes. Este capítulo serve como base para definir e exemplificar o uso desses conceitos no contexto do trabalho proposto.

2.1 Notas musicais

Uma vibração regular produz sons de altura definida, chamados notas musicais. Por altura, entende-se a frequência fundamental (f_0) de suas vibrações, sendo uma frequência maior responsável por um som mais agudo (MED, 1996).

No sistema ocidental temperado (*equal temperament*), o intervalo musical de uma oitava é organizado em uma escala de 12 classes de semitom (C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B).

O nome *equal temperament* (temperamento igual) é dado porque os semitons da escala temperada são distribuídos de forma equidistante em uma escala de frequências logarítmica. A partir de uma nota qualquer, como a nota A4 (com $f_0 = 440$ Hz), a f_0 da próxima nota, um semitom acima, pode ser obtida multiplicando essa frequência por $\sqrt[12]{2}$ ($\approx 1{,}0595$), e a f_0 da nota anterior, um semitom abaixo, dividindo-a pelo mesmo valor.

A escala temperada é cíclica, de forma que ao avançar 12 semitons a partir de uma nota de referência, será encontrada outra nota de mesma classe de semitom. A nota 12 semitons acima dessa referência terá, porém, o dobro da frequência da nota original, como exemplifica a Tabela 2.1. Musicalmente, o intervalo de 12 semitons é denominado uma oitava (HARTQUIST, 2012).

Em termos de notação, é adicionado um número para designar a oitava à qual a nota pertence, partindo do ponto de referência A4 = 440 Hz, que pertence à oitava 4. Doze semitons acima, ou uma oitava, existe a nota A5, e doze semitons abaixo, a nota A3. Notas consecutivas de C até B pertencem a uma mesma oitava.

Tabela 2.1 – Frequências das notas musicais entre A4 e A5

Classe de semitom	Nota	Semitons acima	Frequência (Hz)
A	A4	*Nota original*	440.0
Bb	Bb4	+1 st.	466.2
B	B4	+2 st.	493.9
C	C5	+3 st.	523.3
Db	Db5	+4 st.	554.4
D	D5	+5 st.	587.3
Eb	Eb5	+6 st.	622.3
E	E5	+7 st.	659.3
F	F5	+8 st.	698.5
Gb	Gb5	+9 st.	740.0
G	G5	+10 st.	784.0
Ab	Ab5	+11 st.	830.6
A	A5	+12 st.	880.0

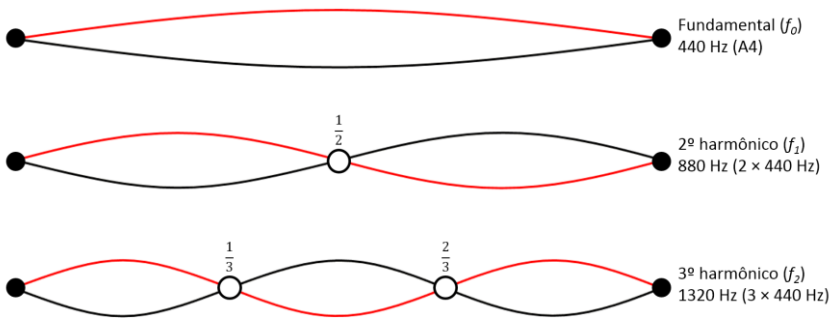
Fonte: O autor

2.2 Sobretons harmônicos

Uma nota musical produzida por um instrumento real não se constitui de apenas uma frequência isolada. Junto com o som principal, chamado som fundamental, soam outros sons secundários chamados sobretons harmônicos, que ocorrem em frequências que são múltiplos inteiros da frequência fundamental (HARTQUIST, 2012).

Ao soar, além de vibrar por inteiro, uma corda também vibra dividindo-se em duas metades, produzindo um som menos intenso uma oitava acima, ou seja, com o dobro da frequência da fundamental. Também se divide em terços, produzindo o triplo da frequência, em quartos, e assim sucessivamente, como mostra a Figura 2.1 (MED, 1996).

Figura 2.1 – Vibração de uma corda em sua frequência fundamental e seus dois primeiros sobretons harmônicos

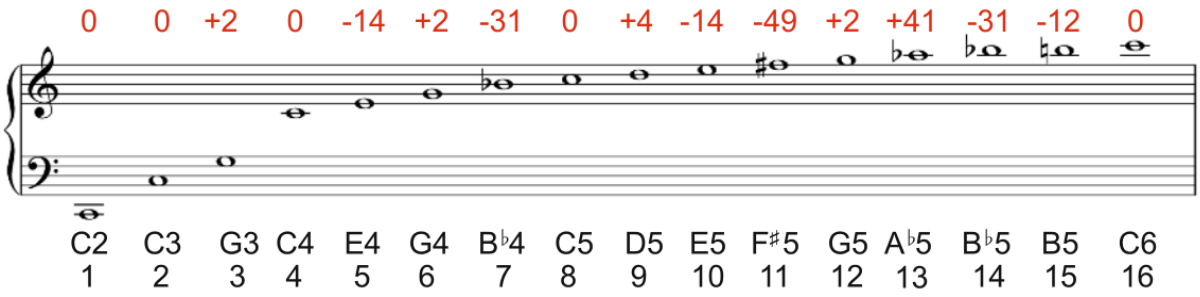


Fonte: O autor

Os sobretons harmônicos de uma nota coincidem aproximadamente com notas mais agudas no sistema temperado. A frequência fundamental é convencionada como primeiro harmônico de uma nota, ou f_0. O chamado segundo harmônico da nota A4 (440 Hz) pode ser calculado como 440 Hz × 2 = 880 Hz, coincidindo exatamente com a nota A5. O terceiro harmônico, 440 Hz × 3 = 1320 Hz, é muito próximo da nota E6 (1318,5 Hz). Assim, quando notas soam simultaneamente, há uma sobreposição altamente estruturada entre as notas fundamentais e seus sobretons harmônicos, sendo esse um dos principais desafios na transcrição de áudio polifônico (ALVARADO; DAN, 2018).

A Figura 2.2 representa, em notação musical, a nota C2 e seus 15 primeiros sobretons harmônicos, quando comparados a notas próximas do sistema temperado. Os valores em vermelho indicam a distância, em *cents*, de cada harmônico à nota mais próxima indicada. Um *cent* equivale a um centésimo de semitom.

Figura 2.2 –Primeiros 16 harmônicos de C2 e comparação com notas mais próximas do sistema temperado



Fonte: Müller (2013, p. 24)

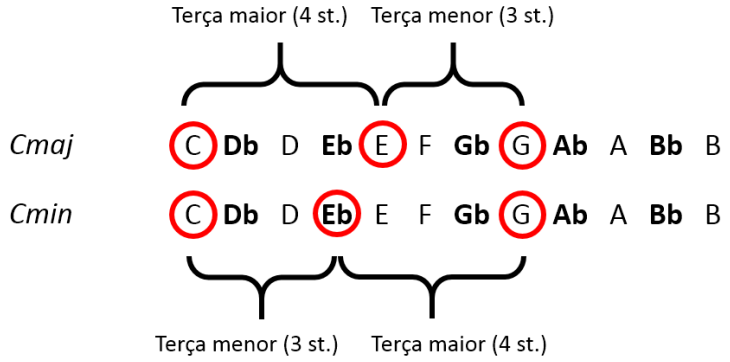
2.3 Acordes

Um acorde é um grupo de três ou mais notas que soam simultaneamente. Na música ocidental, acordes são em geral construídos por intervalos consecutivos de terças (3 ou 4 semitons) a partir de uma nota fundamental (DAY, 2007).

Acordes de três notas construídos com intervalos de terças são chamados de tríades. A nota inferior da tríade é chamada *nota fundamental* (não confundir com frequência fundamental), e dá nome ao acorde, por exemplo a nota C (dó) nos acordes Cmaj e Cmin. A segunda nota da tríade é chamada de *terça*, pois há o intervalo de uma terça entre ela e a nota fundamental. É a nota que define a qualidade do acorde, maior (terça maior, intervalo de 4 semitons) ou menor (terça menor, intervalo de 3 semitons). A terceira e última nota da tríade é chamada *quinta*, pois fica a uma quinta de distância da nota fundamental (7 semitons, no caso

da quinta perfeita). Fica também a uma terça de distância da segunda nota da tríade. A Figura 2.3 mostra exemplos de construção de tríades.

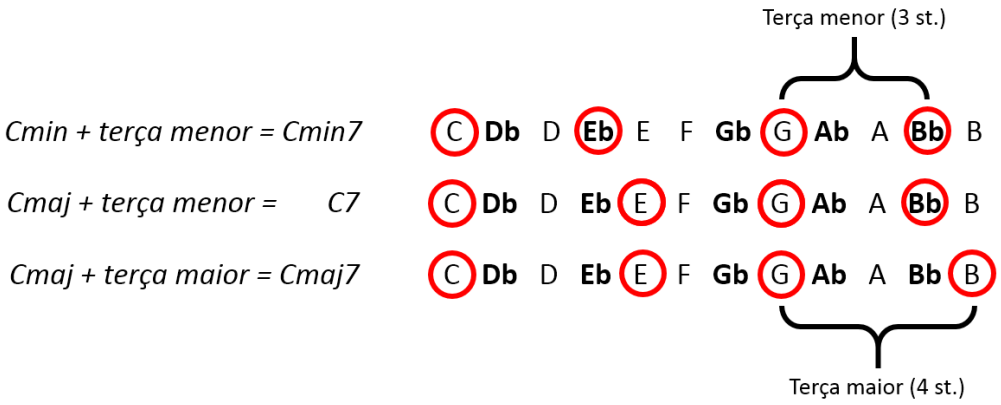
Figura 2.3 – Exemplos de construção de tríades maior e menor com fundamental C



Fonte: O autor

Se adicionarmos mais uma nota à tríade, a uma terça de distância de sua última nota, formamos outro tipo de acorde, os *acordes de sétima*, como mostra a Figura 2.4. A Tabela 2.2 mostra a formação das principais tríades e acordes de sétima na música ocidental.

Figura 2.4 – Exemplo de construção de acordes de sétima com fundamental C



Fonte: O autor

Tabela 2.2 – Distância em semitons (st.) das notas de diferentes tipos de acordes em relação à nota fundamental do acorde, para acordes comuns na música ocidental

Tipo de acorde	Fundamental	Terça	Quinta	Sétima
maj	0	+4 st. (maior)	+7 st.	
min	0	+3 st. (menor)	+7 st.	
min7	0	+3 st. (menor)	+7 st.	+10 st.
7	0	+4 st. (maior)	+7 st.	+10 st.
maj7	0	+4 st. (maior)	+7 st.	+11 st.

Fonte: O autor

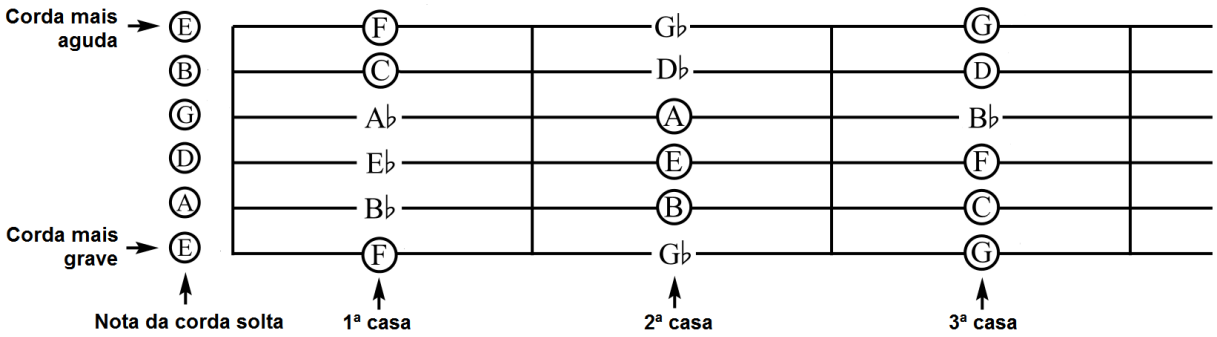
2.4 Acordes em uma guitarra

A guitarra elétrica moderna (no Brasil, simplesmente conhecida como guitarra) foi inventada em 1948 por Leo Fender (DAY, 2017), permitindo efeitos como sustentação e distorção que não existiam em sua versão acústica, o violão. Ao contrário do violão, que possui um corpo oco para ressonância da vibração de suas cordas, a guitarra utiliza captadores magnéticos para traduzir as vibrações das cordas em sinais elétricos, que são então transmitidos para um amplificador (HARQUIST, 2012).

Uma guitarra padrão possui seis cordas, que podem ser tocadas soltas ou pressionando certas casas. Todas as casas e cordas soltas correspondem a uma das 12 classes de semitom do sistema temperado, numa determinada oitava.

A Figura 2.5 mostra a disposição das notas ao longo das cordas e casas de uma guitarra. Cada corda emite uma nota, quando tocada solta. Ao longo da corda estão dispostas casas, separadas por trastes, que podem ser pressionadas para alterar a nota de uma corda. A primeira casa de cada corda é um semitom mais aguda que a própria corda, e cada casa subsequente é um semitom mais aguda que a casa anterior.

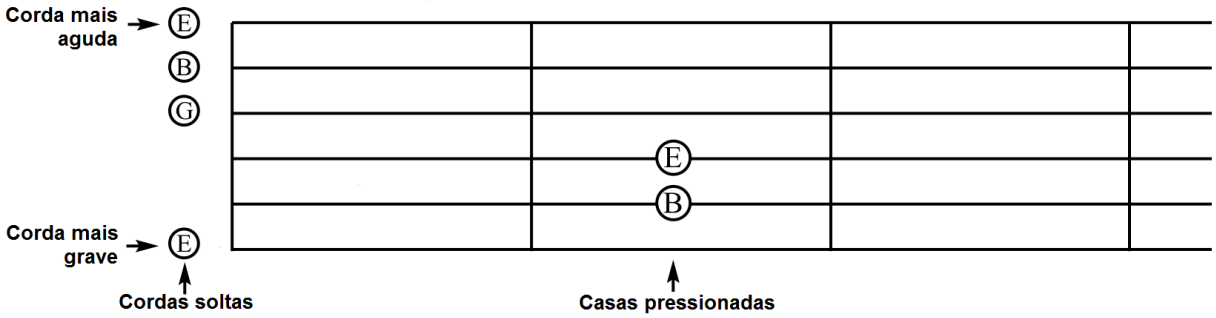
Figura 2.5 – Disposição das notas nas primeiras casas de uma guitarra



Fonte: Adaptado de Day (2007, p. 89)

Para tocar, por ex., o acorde *Emin* (mi menor), composto pelas notas E, G e B, basta o guitarrista pressionar duas casas específicas nas cordas D e A, como mostra a Figura 2.6, e manter as outras cordas soltas. Dessa forma, apenas as notas que pertencem à tríade irão soar quando a palheta passar por todas as cordas.

Figura 2.6 – Posição das notas do acorde Emin (mi menor)



Fonte: Adaptado de Day (2007, p. 89)

2.5 Progressões de acordes

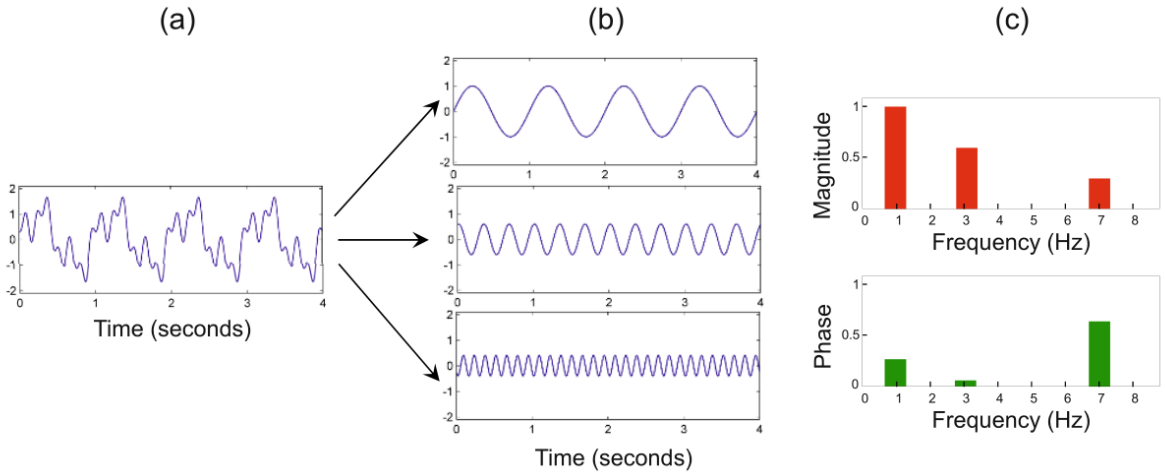
Assim como na construção de frases textuais, a construção de uma música é feita com base em uma série de regras e padrões, definidos pela linguagem musical. Embora seja tecnicamente possível que um acorde seja sucedido por qualquer outro, há uma forte tendência a serem seguidos padrões, que variam de acordo com o gênero musical (DAY, 2007).

Muitos algoritmos de reconhecimento de acordes fazem uso desse conhecimento para eliminar transições de acordes improváveis (CHO; BELLO, 2013). Como exemplo, na música popular e no jazz, progressões em que a fundamental do acorde sobe o intervalo de uma quarta (5 semitons) ou desce o intervalo de uma sétima (7 semitons) são as mais comuns e marcantes (LEVINE apud CHO; BELLO, 2013).

2.6 Transformada de Fourier

A transformada de Fourier é, segundo Müller (2015), a mais importante ferramenta matemática no processamento de sinais de áudio. Sua função é converter um sinal dependente de tempo em uma função dependente de frequência (MÜLLER, 2015). Dessa forma, podemos utilizá-la para converter um pequeno trecho de áudio do domínio de tempo para o domínio de frequências, decompondo-o em um espectro com as intensidades associadas a cada frequência observada, como mostra a Figura 2.7.

Figura 2.7 – (a) Sinal analógico periódico. (b) Decomposição do sinal em ondas seno. (c) Magnitude e fase resultantes da transformada de Fourier



Fonte: Müller (2015, p. 70)

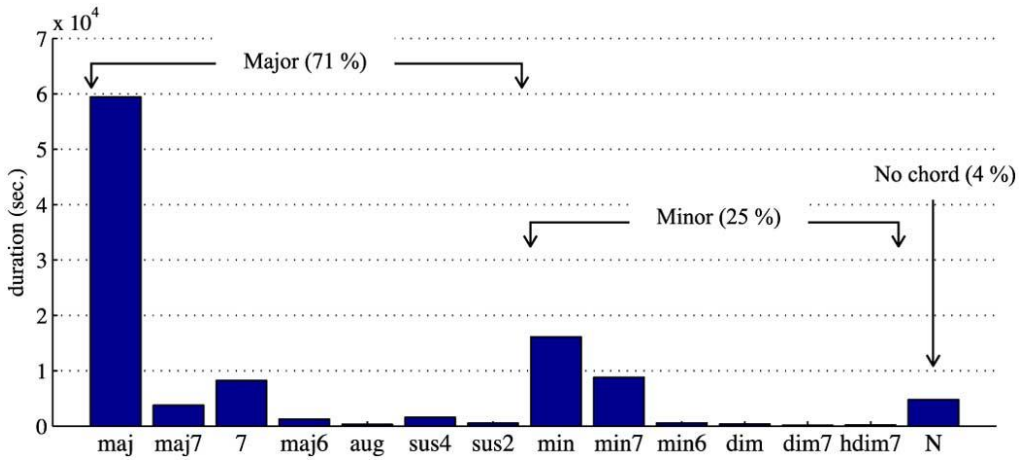
Na prática, a transformada de Fourier é aproximada através de somas finitas, a chamada *transformada discreta de Fourier (DFT)*, que agrupa o domínio de frequências, contínuo, em N *bins* discretos. Cada *bin* é centralizado em uma frequência, e sua magnitude reflete a magnitude observada nas frequências que compreendem seu intervalo.

O algoritmo mais utilizado para calcular a transformada discreta de Fourier é o chamado FFT (*Fast Fourier Transform*), devido à sua eficiência computacional, sendo considerado um dos mais importantes algoritmos com inúmeras aplicações em engenharia e matemática (MÜLLER, 2015).

3 TRABALHOS RELACIONADOS

O trabalho de Cho e Bello (2013) traz uma proposta semelhante, fazendo uma análise estruturada de diferentes técnicas em cada uma das etapas de um *pipeline* de reconhecimento de acordes. Possui, no entanto, escopo mais abrangente, sem o foco em um instrumento específico como a guitarra, utilizando *datasets* de música popular como Beatles e Queen. O trabalho também é limitado à detecção de 24 acordes (12 maiores e 12 menores), bem como a ausência de acorde, *NC* (*No Chord*). Acordes de 4 notas como *7*, *maj7* e *min7*, são mapeados para acordes maiores e menores, de acordo com suas notas básicas, como mostra a Figura 3.1. Trata-se de uma prática bastante comum por restringir o número de classes e a confusão entre as classes, limitando porém a utilidade prática de uma ferramenta de detecção de acordes. Outro motivo para a simplificação é a baixa ocorrência destes tipos de acordes na música popular, o que limita técnicas que envolvem aprendizado de máquina e requerem um grande volume de dados. No presente trabalho, será feita uma comparação do impacto da adição de mais classes de acordes além das classes maior e menor.

Figura 3.1 – Distribuição de tipos de acordes no dataset utilizado por Cho e Bello, bem como mapeamento para acordes maiores e menores



Fonte: CHO; BELLO, 2013, p. 484

Nadar et al (2019), num trabalho mais recente, focam em um vocabulário de acordes estendido, utilizando em vez de apenas 24 acordes maiores e menores, 84 diferentes tipos de acordes. O trabalho foi feito com base em um algoritmo proposto de redes neurais profundas convolucionais (*deep convolutional neural networks*). Devido à menor ocorrência de acordes estendidos na música, em geral, os autores consideraram necessária a criação de um conjunto de dados próprio, para fins de treino e avaliação. O conjunto de dados foi gerado sinteticamente

a partir de arquivos MIDI, utilizando programas como *Ableton Live* e *Garage Band* para síntese, permitindo uma grande autonomia e liberdade.

A tese de Mazhar (2012) tem foco instrumental semelhante, restringindo-se à detecção de acordes em violão. Baseia-se em uma técnica de *pattern-matching* aplicada a um banco de acordes, o qual contém não só versões corretas dos acordes como erros comuns de estudantes de violão. Utiliza o processo de *spectral whitening* para normalização do espectro, e compara as características extraídas através da distância cosseno em relação aos acordes de referência. Os resultados são então avaliados com um banco de dados próprio.

O trabalho de Mauch e Dixon (2010) trouxe um avanço significativo na tarefa de detecção de acordes. O trabalho obteve resultados de 80% de acurácia nas métricas utilizadas pela convenção MIREX, ultrapassando o resultado do ano anterior, de 74%. O foco do trabalho está na etapa de extração de notas musicais a partir do espectro de frequências. O algoritmo proposto pelos autores utiliza uma técnica posteriormente chamada de *Chroma NNLS*. A partir de um espectro de frequências e suas amplitudes, o algoritmo tenta deduzir qual a combinação de notas musicais e seus harmônicos que resulta nesse espectro. Para isso, utiliza o método de mínimos quadrados não negativos (*non-negative least squares*) para resolver um problema de otimização envolvendo perfis de notas e o espectro observado. O algoritmo está disponível online, em forma de programa e plug-in, e seus resultados serão utilizados de referência para comparação dos resultados do presente estudo.

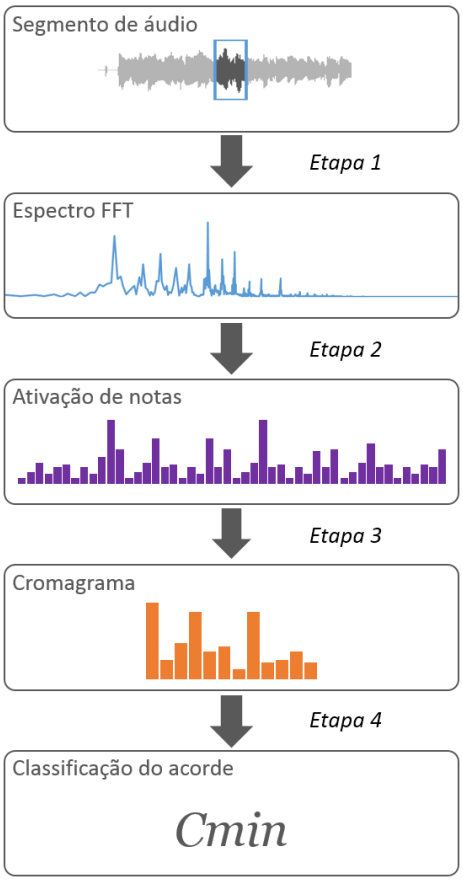
4 FLUXO DE EXECUÇÃO E TÉCNICAS AVALIADAS

Foi desenvolvido um sistema com base no *pipeline* descrito por Müller (2015). Ao longo deste capítulo, cada etapa será explicada em mais detalhes, e serão descritos os principais desafios e as técnicas implementadas para avaliação e comparação.

4.1 Visão geral

O arquivo de áudio é inicialmente segmentado em múltiplos quadros (*frames*), que são processados por um *pipeline*. A *Etapa 1* do processamento é a extração do espectro FFT a partir do áudio do segmento. O espectro revela as intensidades das frequências presentes no segmento. Essa informação é utilizada pela *Etapa 2* para estimar a intensidade das notas musicais presentes, sendo observadas 48 notas musicais. A *Etapa 3* é a acumulação das notas em um mapa das 12 classes de semitom, chamado cromagrama. Finalmente, na *Etapa 4*, os cromagramas são comparados a diferentes modelos de acorde, e o segmento é classificado.

Figura 4.1 – Etapas do *pipeline* de reconhecimento de acordes implementado



Fonte: O autor

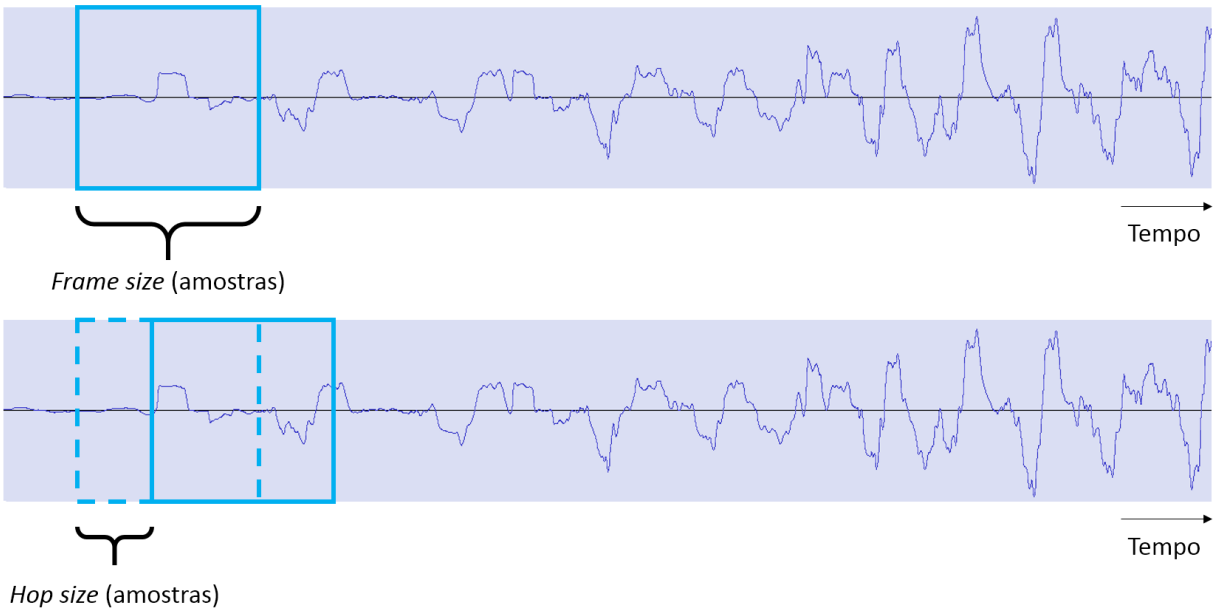
4.2 Etapa 1: Cálculo do espectro FFT a partir de um segmento de áudio

Para a primeira etapa, o arquivo de áudio é segmentado em múltiplos quadros (*frames*), pequenos trechos de áudio. É aplicada então uma função de janelamento ao quadro, e o espectro FFT é calculado.

O tamanho do quadro, ilustrado na Figura 4.2, é determinado pelo parâmetro *frame size* (tamanho do quadro), que especifica quantas amostras de áudio fazem parte de cada quadro. Por amostra, entende-se cada ponto discreto em um sinal de áudio.

Existe ainda o parâmetro *hop size* (tamanho do salto), também ilustrado na Figura 4.2, que determina quantas amostras devem ser avançadas para obter o quadro seguinte, podendo haver sobreposição com o quadro anterior. Para todos os testes, foi utilizado *hop size* de 4410 amostras (equivalente a 0,1 s de áudio em arquivos amostrados a 44100 Hz), valor também utilizado por Nadar et al (2019). Como os quadros são analisados de forma independente no presente algoritmo, sem processamento temporal, basta que o parâmetro tenha um valor suficientemente pequeno para capturar mudanças de acorde.

Figura 4.2 – Ilustração dos parâmetros tamanho do quadro (*frame size*) e tamanho do salto (*hop size*)



Fonte: O autor

Outro algoritmo frequentemente usado para o mesmo propósito é a transformada CQT (*Constant-Q Transform*) (CHO; BELLO, 2013). A CQT possui distribuição de frequências logarítmica, quando comparada à FFT, cuja distribuição é linear. Uma distribuição de frequências logarítmica permite um mapeamento mais simples para os semitons do sistema temperado, cujas frequências fundamentais também são distribuídas de forma logarítmica.

4.2.1 Tamanho do quadro

A escolha do tamanho do quadro tem efeito muito grande no reconhecimento de acordes. Isso acontece porque na transformada FFT, existe um compromisso entre a resolução temporal e resolução das frequências (MÜLLER, 2015). Um quadro muito pequeno tem alta resolução temporal, porém baixa resolução de frequências, o que compromete o reconhecimento de notas mais graves. As notas graves são mais afetadas porque os *bins* da FFT são distribuídos de forma linear, enquanto as notas musicais são distribuídas de forma logarítmica ao longo das frequências, resultando em uma baixa densidade de *bins* para as notas mais graves e alta densidade para as notas mais agudas. Por outro lado, se a janela for muito grande, a informação temporal da troca de acordes acaba sendo perdida, havendo uma mistura de múltiplos acordes em um mesmo quadro.

Uma ilustração do efeito descrito pode ser vista no *Experimento 1* (seção 6.2), bem como uma avaliação dos efeitos de diferentes tamanhos de quadro nos resultados do reconhecimento.

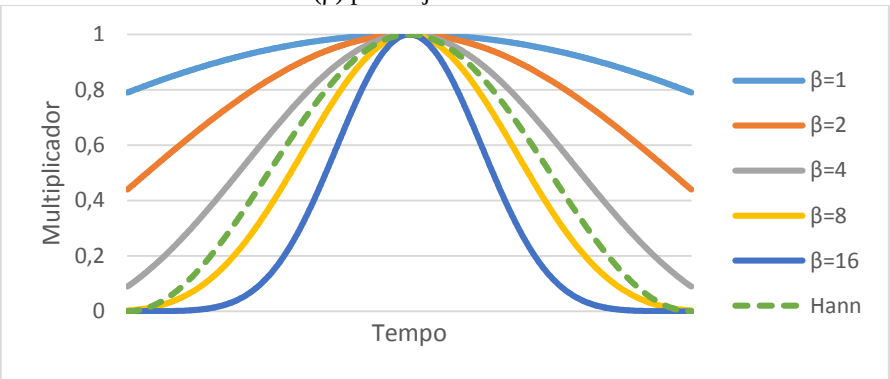
4.2.2 Função de janelamento

Após a segmentação, cada quadro é então processado por uma função de janelamento, de forma a suavizar o começo e fim da janela de tempo, como preparação para análise pela FFT. Sem o janelamento, as descontinuidades nas bordas geram interferências no espectro resultante (MÜLLER, 2015).

A função de janelamento escolhida inicialmente para uso na aplicação é a *janela de Kaiser* (KAISER; SCHAFER, 1980). Essa função possui um único parâmetro, *beta*, que determina o formato do janelamento ao longo do tempo, onde valores maiores geram um formato mais estreito, como demonstra a Figura 4.3.

Outra função de janelamento frequentemente utilizada no processamento de sinais é a *janela de Hann* (MÜLLER, 2015). Como exemplo, o trabalho de Nadar et al (2019) utiliza a janela de Hann aplicada a quadros de 8192 amostras. Uma comparação da janela de Hann com diferentes parâmetros da janela de Kaiser pode ser observada na Figura 4.3.

Figura 4.3 – Efeito temporal nas amostras do segmento de áudio do janelamento de Hann e de diferentes valores de *beta* (β) para o janelamento de Kaiser



Fonte: O autor

O ajuste fino do tamanho da janela dentro do quadro pode ser feito através do parâmetro opcional *window size* (tamanho da janela), que define quantas amostras a janela ocupará no centro do quadro, sendo o restante das amostras preenchido com valor zero.

Na seção 6.3, serão comparados os efeitos dos diferentes janelamentos e parâmetros *beta*, bem como os resultados obtidos quando nenhum janelamento é aplicado.

4.3 Etapa 2: Extração de notas musicais a partir do espectro FFT

Nesta etapa, o resultado da FFT do quadro é utilizado para determinar a ativação das diferentes notas musicais. Foram escolhidas 48 notas consecutivas (4 oitavas), por corresponder aproximadamente ao número de notas disponível em todo o braço de uma guitarra comum. A primeira nota detectada é a nota D2 (com $f_0 \approx 73{,}4$ Hz), dois semitons abaixo da nota mais grave de uma guitarra em afinação padrão, E2. A diferença permite certa robustez em relação a afinações alternativas que alteram a corda mais grave de E para D. A última nota detectada é a nota Db6 (com $f_0 \approx 1108{,}7$ Hz), nota mais aguda de uma guitarra de 21 casas.

A extração das notas a partir do espectro apresenta uma série de desafios, que serão discutidos na motivação e detalhamento de cada método.

4.3.1 Mapeamento direto da f_0

A primeira técnica a ser discutida é a extração direta das notas musicais a partir do espectro de um determinado quadro. A ativação de cada nota musical é determinada diretamente pela magnitude observada no espectro para a sua frequência fundamental, f_0. Por exemplo, a ativação da nota A4 é determinada pela magnitude do espectro observada em $f_0(A4)$, ou seja, em 440 Hz.

$$Ativação(nota) = Magnitude(f_0(nota))$$

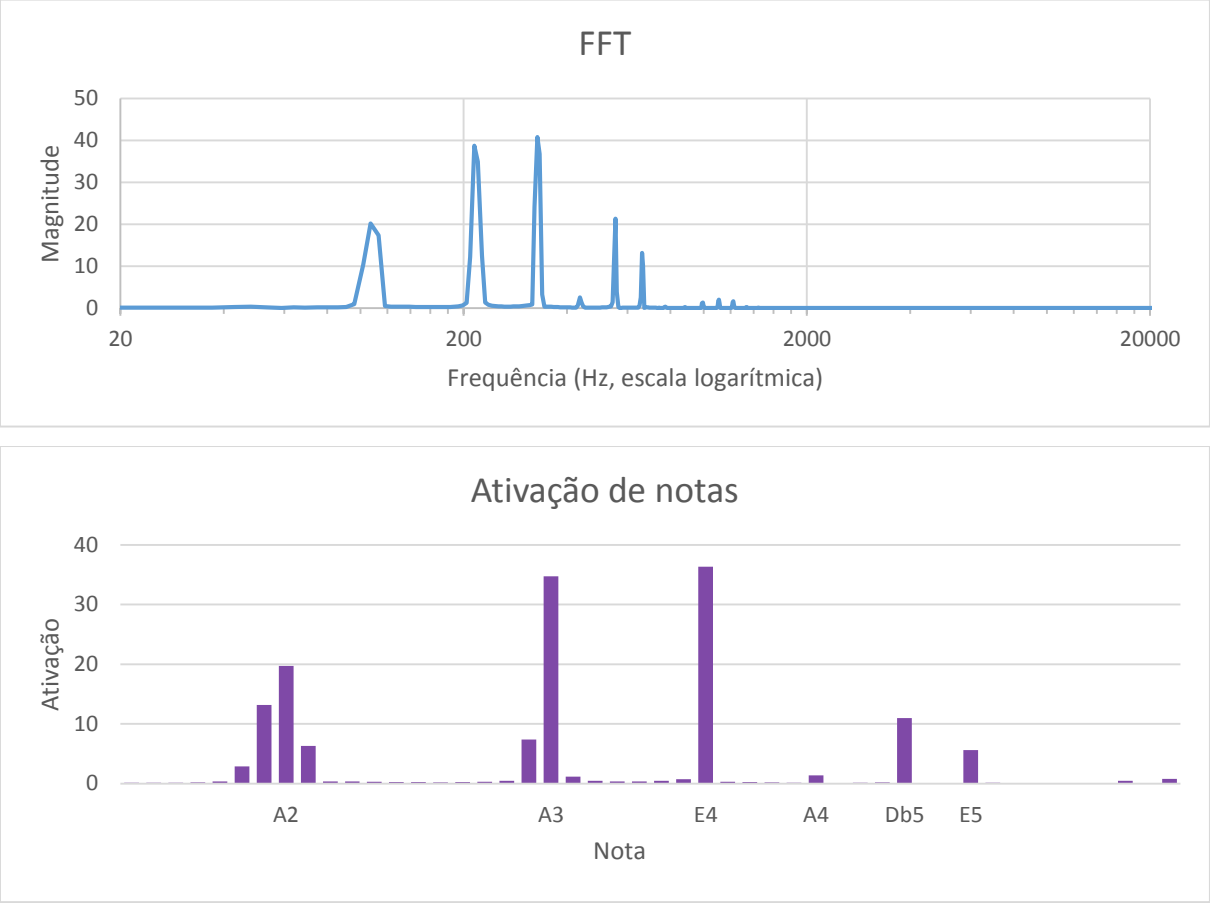
Como as frequências das notas podem não coincidir exatamente com as frequências dos *bins* da FFT, para calcular $Magnitude(frequência)$, é feita uma média ponderada das magnitudes dos *bins* mais próximos à frequência, limitando-se a meio semitom de distância. Nessa média ponderada, o peso da magnitude de cada *bin* é proporcional à distância que o *bin* está da frequência, sendo 0 a meio semitom de distância e 1 se coincide com a frequência. O resultado é então normalizado dividindo-se pela soma de todos os pesos.

Na ausência de pelo menos dois *bins* a essa distância da frequência desejada, é feita uma interpolação linear dos dois *bins* mais próximos sem restrição de distância. Para evitar o uso dessa interpolação nas notas mais graves observadas, é necessária uma janela de pelo menos 32768 amostras.

A principal limitação desse algoritmo está na existência de sobretons harmônicos, descritos no Capítulo 2, que poluem o espectro criando ativações incorretas para certas notas. Dessa forma, a ativação da nota G4 pode ter sido causada tanto pela existência de uma nota com fundamental G4 quanto pelo terceiro harmônico (segundo sobretom) de uma nota com fundamental C2, com frequência semelhante.

A Figura 4.4 mostra um caso do problema, a partir de uma gravação da nota avulsa A2, sendo tocada na segunda corda mais grave de uma guitarra. Idealmente, apenas a nota A2 deveria ser ativada. Porém, devido a fortes sobretons harmônicos, a nota original causa ativações mais intensas nas notas A3 e E4, que coincidem com seu segundo e terceiro harmônicos, respectivamente. Os demais harmônicos também podem ser visualizados no espectro como ativações de notas. Além disso, o uso de um tamanho de quadro pequeno (de 8192 amostras) causa um vazamento da magnitude para notas vizinhas.

Figura 4.4 – Espectro FFT e ativação de notas para gravação de uma única nota, A2



Fonte: O autor

4.3.2 Atenuação de harmônicos através de janela gaussiana

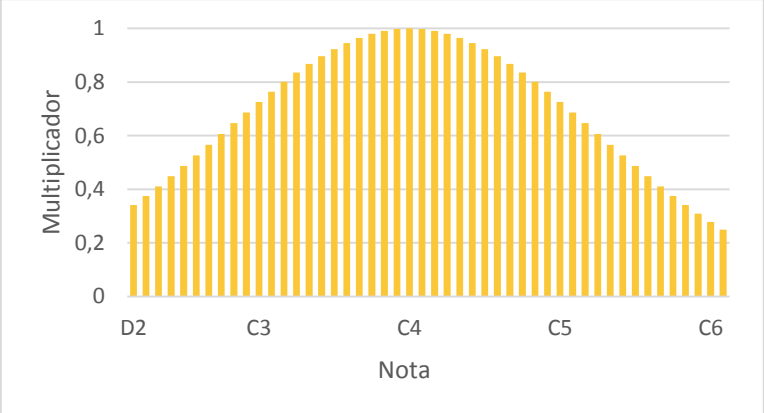
Cho e Bello (2013) descrevem como técnica mais comum e simples de lidar com os efeitos dos harmônicos, bem como a baixa resolução nas frequências graves, a atenuação das bordas do espectro. Isso porque a parte mais aguda do espectro é, segundo os autores, em grande parte composta de sobretons, e a parte mais grave, de ruídos e sons de bateria, em música popular.

Para tal, a ativação de cada nota musical é obtida a partir de sua f_0 como no método anterior, e é aplicado um janelamento nas notas detectadas. O janelamento é centralizado na nota C4, região onde se espera que os tons fundamentais sejam encontrados (BIASUTTI, 1997 apud CHO; BELLO, 2013), suprimindo sobretons e possíveis ruídos nas partes aguda e grave do espectro, através da seguinte fórmula:

$$\hat{P}(p) = \exp\left(-\frac{(p-60)^2}{2 \cdot 15^2}\right) \cdot P(p)$$

A fórmula é um janelamento gaussiano, centralizado na nota C4 ($p = 60$), de forma que para cada nota p, a sua ativação original $P(p)$ será alterada para $\widehat{P}(p)$. O efeito do janelamento nas 48 notas selecionadas pode ser observado na Figura 4.5.

Figura 4.5 – Formato do janelamento Gaussiano



Fonte: O autor

Embora a técnica seja capaz de reduzir os efeitos de sobretons causados por notas próximas a C4, a técnica não é robusta o suficiente para lidar com sobretons harmônicos de notas mais graves, os quais coincidem com o centro do janelamento. Um exemplo pode ser visto na Figura 4.6, que utiliza a mesma gravação da Figura 4.4. Na figura, embora certos sobretons harmônicos como os sobretons próximos a Db5 e E5 sejam reduzidos, a ativação da própria nota que está soando, A2, também é reduzida.

Figura 4.6 – Ativação de notas utilizando janelamento gaussiano

Fonte: O autor

4.3.3 Mapeamento da f_0 com base em componentes harmônicos

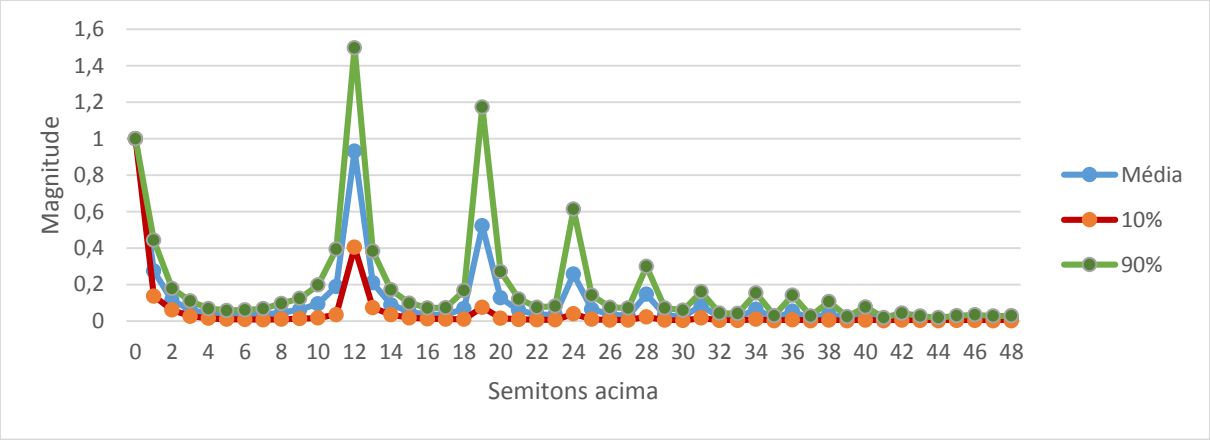
Este método é uma variação do método de mapeamento direto da f_0, descrito na seção 4.3.1, e leva em conta na fórmula de ativação de cada nota musical não só a presença de magnitude em sua frequência fundamental f_0, mas também em seus sobretons harmônicos $f_{1..n}$. A existência de harmônicos nas frequências esperadas *reforça* a ativação da nota, com pesos proporcionais a uma nota modelo, através da seguinte fórmula:

$$Ativação(nota) = \sum_{n=0}^{9} Magnitude(f_n(nota)) \times PesoHarm(n)$$

A ativação de cada nota é determinada pelo somatório das magnitudes obtidas na frequência de cada um de seus primeiros 10 harmônicos, de f_0 a f_9. O peso de cada harmônico n no somatório, $PesoHarm(n)$, é obtido a partir de uma nota modelo.

Para gerar a nota modelo, foram utilizadas 232 gravações de diferentes notas em diferentes tipos de guitarra elétrica, contidas no *Dataset 1*, obtendo-se o espectro médio ao alinhar o espectro normalizado de cada nota em semitons relativos à fundamental, como mostra a Figura 4.7. A figura ainda mostra os percentis 10% e 90% para cada semitom observado, entre os quais 80% dos dados observados para o semitom se situam.

Figura 4.7 – Nota modelo – média e percentis de múltiplas gravações alinhadas pela fundamental



Fonte: O autor

A partir da nota modelo, foram obtidos os pesos da Tabela 4.1.

Tabela 4.1 – Pesos de harmônicos obtidos a partir da nota modelo

n	f_n	$PesoHarm(n)$
0	f_0	1,00
1	$f_0 \times 2$	0,93
2	$f_0 \times 3$	0,52
3	$f_0 \times 4$	0,26
4	$f_0 \times 5$	0,15
5	$f_0 \times 6$	0,08
6	$f_0 \times 7$	0,06
7	$f_0 \times 8$	0,05
8	$f_0 \times 9$	0,04
9	$f_0 \times 10$	0,04

Fonte: O autor

Contudo, ao utilizar os pesos descritos, o algoritmo gerou ativações indesejadas em notas de classes de semitom diferentes. Por exemplo, uma nota C3, mesmo que não esteja soando, pode ser ativada pela presença de magnitude em sua f_2, causada por uma nota G4. Assim, foi feita um novo ajuste de pesos, descritos na Tabela 4.2, excluindo frequências que não coincidem com a classe de semitons da nota observada, mantendo apenas a frequência fundamental e sobretons em oitavas acima.

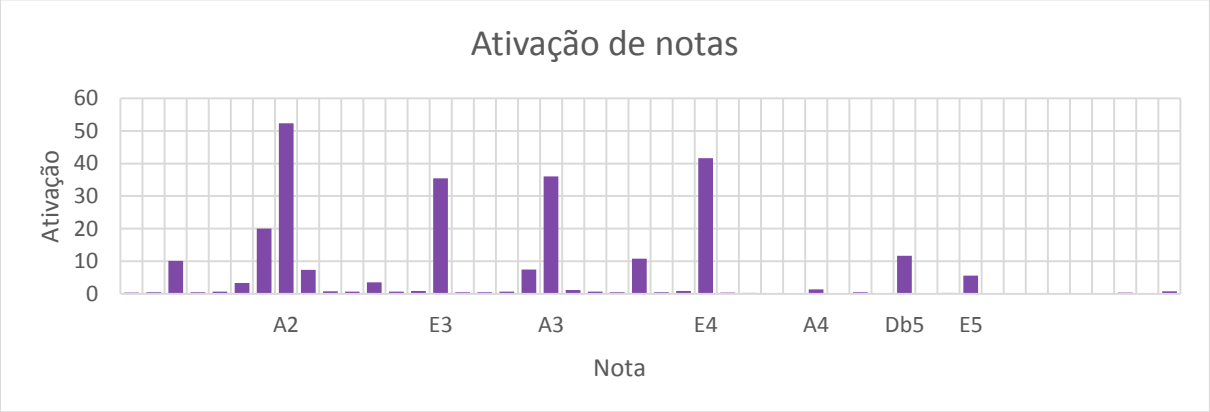
Tabela 4.2 – Pesos de harmônicos restritos a uma classe de semitom

n	f_n	$PesoHarm(n)$
0	f_0	**1,00**
1	$f_0 \times 2$	**0,93**
2	$f_0 \times 3$	0
3	$f_0 \times 4$	**0,26**
4	$f_0 \times 5$	0
5	$f_0 \times 6$	0
6	$f_0 \times 7$	0
7	$f_0 \times 8$	**0,05**
8	$f_0 \times 9$	0
9	$f_0 \times 10$	0

Fonte: O autor

O mesmo exemplo utilizado nos casos anteriores, a gravação da nota avulsa A2, pode ser visto na Figura 4.8. A ativação da nota A2 é reforçada, sendo a nota de maior ativação, como desejado. O algoritmo, porém, gera certos artefatos, ativando por exemplo a nota E3, que não estava presente nos métodos anteriores.

Figura 4.8 – Ativação de notas utilizando método baseado em componentes harmônicos



Fonte: O autor

4.4 Etapa 3: Extração do cromagrama a partir das notas musicais

As magnitudes das 48 notas, obtidas na *Etapa 2*, podem então ser simplificadas para o chamado *cromagrama*, um vetor de 12 posições representando as 12 classes de semitom. O objetivo desta etapa é obter um único coeficiente para cada classe de semitom. Isso porque a percepção de altura (*pitch*) humana é cíclica, de forma que duas notas de mesma classe de semitom em oitavas diferentes cumprem papéis semelhantes, harmonicamente (MÜLLER, 2015).

Para o cálculo, as 48 notas escolhidas são somadas, de acordo com a classe de semitom à qual pertencem. O cálculo da ativação da classe de semitom C no cromagrama, por exemplo, é determinado pela soma das ativações de C3, C4, C5 e C6.

A Figura 4.9 mostra as diferentes etapas do processamento de um segmento de áudio de um acorde *A* contendo as notas A2, E3, A3, Db4 e E4. A ativação das notas é efetuada utilizando o *mapeamento direto da* f_0, descrito na seção 4.3.1. Em destaque estão as notas das quais a ativação é esperada. As ativações das 48 notas são então agrupadas em 12 classes de semitom e somadas, resultando no cromagrama observado, com destaque para os semitons do acorde (A, Db e E).

Figura 4.9 – Espectro FFT, ativação de notas e cromagrama para um segmento do acorde *A* (lá maior)

Fonte: O autor

Alguns sistemas utilizam cromagramas de 24 notas (SHEH; ELLIS; 2003) para lidar com variações na afinação do instrumento. Existem ainda sistemas que utilizam um cromagrama adicional para as notas mais graves (MAUCH; DIXON, 2010), com a intenção de capturar a nota base do acorde para classificar inversões de acorde, nas quais a nota mais grave não é a nota fundamental do acorde.

4.5 Etapa 4: Extração de acorde a partir do cromagrama

Nesta etapa, o cromagrama é então comparado a uma série de modelos de acordes, para determinar qual o acorde presente no cromagrama. Os modelos de acordes, assim como o cromagrama, são vetores de 12 posições, representando cada classe de semitom. Essa comparação é feita através de uma função de similaridade. No presente trabalho, é utilizada como função de similaridade a multiplicação escalar do vetor do cromagrama pelo vetor de cada modelo de acorde, como descrita por Müller (2015). O acorde é determinado pelo modelo de acorde que maximiza a função de similaridade.

4.5.1 Modelagem de acordes com vetores binários

Uma das formas mais comuns para a criação de modelos de acordes, descritas por Müller (2015) e avaliadas por Cho e Bello (2013) é a utilização de vetores binários criados manualmente, representando cada um dos acordes, como mostra a Tabela 4.3.

Tabela 4.3 – Modelos de acordes maiores e menores

Modelo	C	Db	D	Eb	E	F	Gb	G	Ab	A	Bb	B
C	1	0	0	0	1	0	0	1	0	0	0	0
Db	0	1	0	0	0	1	0	0	1	0	0	0
D	0	0	1	0	0	0	1	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
Cmin	1	0	0	1	0	0	0	1	0	0	0	0
Dbmin	0	1	0	0	1	0	0	0	1	0	0	0
Dmin	0	0	1	0	0	1	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...

Fonte: Adaptado de Müller (2015, p. 255)

Em ambos os casos, os autores definem os modelos apenas para acordes maiores e menores. Os acordes estendidos analisados no presente estudo, contudo, possuem uma nota adicional, e um vetor binário com quatro valores 1 sempre terá multiplicação escalar com resultados maiores ou iguais aos resultados dos vetores binários com três valores 1 (considerando apenas valores positivos).

Uma solução simples para o problema é o uso de vetores normalizados. Para tal, basta dividir cada um dos elementos do vetor pela magnitude do vetor. A magnitude de um vetor de 12 posições pode ser obtida através da seguinte fórmula:

$$\|\boldsymbol{v}\| = \sqrt{{v_1}^2 + {v_2}^2 \ldots {v_{12}}^2}$$

Para um vetor binário com três valores 1, que representa os acordes maiores e menores, cada elemento deve ser dividido por $\sqrt{1^2 + 1^2 + 1^2} = \sqrt{3}$. Já para um vetor binário com quatro valores 1, que representa os valores estendidos, cada elemento deve ser dividido por $\sqrt{1^2 + 1^2 + 1^2 + 1^2} = \sqrt{4} = 2$.

A Tabela 4.4 descreve as 5 classes de acordes analisadas, com um exemplo de modelo de acorde para cada classe, com fundamental em C, após a normalização. Cada classe possui 12 acordes, totalizando 60 modelos. Os demais acordes de cada classe de acordes podem ser construídos através da rotação dos elementos do vetor de exemplo, como na Tabela 4.3.

Tabela 4.4 – Classes de acordes analisadas e exemplo de modelo de acorde de cada classe, com fundamental em C, após normalização do vetor

Classe	Exemplo	C	Db	D	Eb	E	F	Gb	G	Ab	A	Bb	B
maj	*C*	$\frac{1}{\sqrt{3}}$	0	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	0	0
min	*Cmin*	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	0	$\frac{1}{\sqrt{3}}$	0	0	0	0
min7	*Cmin7*	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0
7	*C7*	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0
maj7	*Cmaj7*	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$

Fonte: O autor

4.5.2 Modelagem da ausência de acorde *(NC)*

Finalmente, existe a anotação para a determinação da ausência de acorde, *NC (no chord)*. A anotação é encontrada em momentos de silêncio ou quando o acorde é indeterminado. Cho e Bello optam por não criar um modelo binário para a ausência de acorde, e definem empiricamente um limiar máximo de *-57 dB RMS* para classificar um quadro como *NC* (CHO; BELLO, 2013). Devido a diferenças no tamanho de quadro usado pelos autores e pelo presente trabalho, serão analisados e utilizados outros valores em dB para o limiar de ausência de acorde.

A medida RMS é uma das formas mais comuns de cálculo de intensidade de um som (LERCH, 2012), e pode ser definida a partir da seguinte fórmula:

$$RMS(quadro) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} mag(i)^2}$$

Na fórmula acima, o RMS de um quadro de tamanho n pode ser calculado como a raiz do valor correspondente à razão $\frac{1}{n}$ quando multiplicada pela soma dos quadrados das magnitudes de todas as suas amostras (de 0 a $n-1$).

Como técnica alternativa, o presente trabalho implementa um limiar dinâmico, também baseado no valor em dB RMS. A técnica mantém como referência o maior valor em dB RMS já encontrado em um quadro, considerando todos os quadros processados anteriormente na mesma gravação. São definidos então como *NC* todos os quadros que estão a N decibéis ou menos abaixo desse maior valor encontrado. A ideia do uso de um limiar dinâmico é aumentar a robustez para casos em que o volume de um sinal não é normalizado.

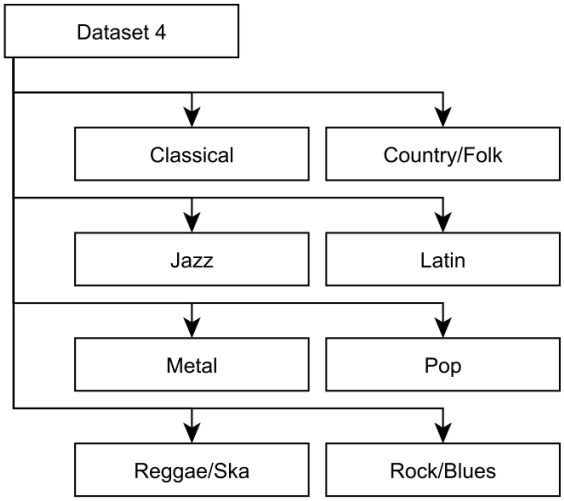
5 DADOS DE TESTE

Para avaliação dos resultados, foi utilizado o conjunto de dados de Eppler, Kehling e Männchen (2016), criado com o propósito específico de servir de referência a diferentes tipos de extração de informação musical[4].

O banco de dados é separado em 4 grupos com propósitos específicos. O grupo escolhido para avaliação do presente trabalho é o chamado *Dataset 4*, do qual um dos propósitos descritos é a tarefa de reconhecimento de acordes (EPPLER; KEHLING; MÄNNCHEN, 2016).

Este grupo é composto de 64 canções curtas, agrupadas em 8 gêneros, com diferentes tempos (devagar e rápido) e diferentes tipos de violão e guitarra elétrica. Cada canção é acompanhada de um arquivo de anotação, que contém, entre outras informações, marcações de tempo para cada uma das mudanças de acorde, bem como o nome do acorde correspondente.

Figura 5.1 – Gêneros musicais presentes no banco de dados

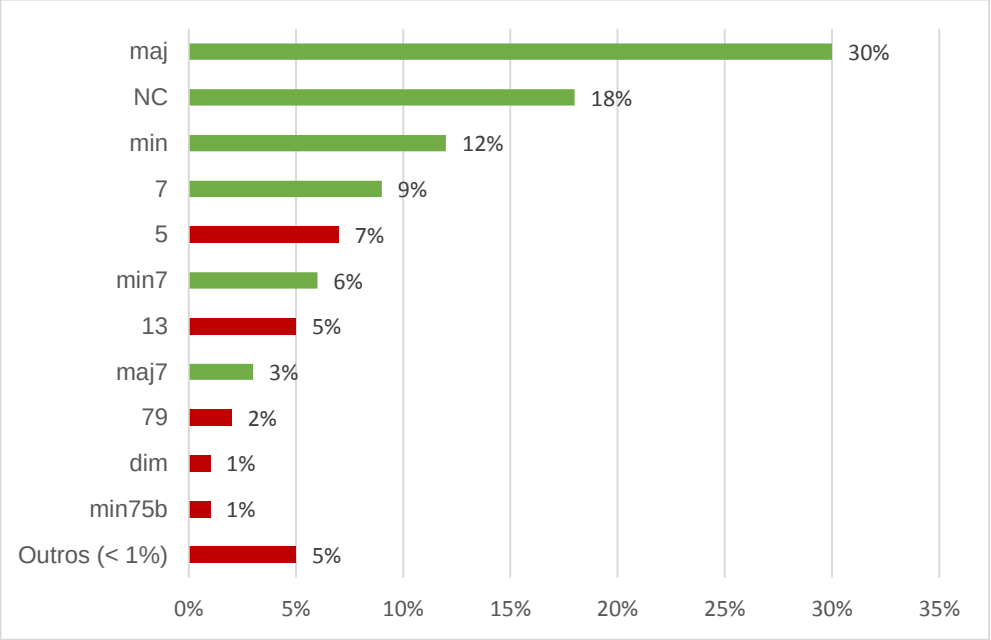


Fonte: Eppler, Kehling e Männchen (2016, p. 7)

O presente trabalho faz uso das 64 canções gravadas com a guitarra *Ibanez 2820* em velocidade rápida (subgrupo *fast*). Uma análise da proporção de cada classe de acorde em relação à duração total do conjunto de dados pode ser observada na Figura 5.2. Em verde estão as classes de acorde modeladas pela aplicação. Uma descrição detalhada das 64 canções e acordes presentes ser encontrada no *Apêndice B.*

[4] Disponível em: https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/guitar.html Acessado em: 18 de julho de 2019

Figura 5.2 - Distribuição de classes de acorde ao longo da duração do conjunto de dados



Fonte: O autor

Outro grupo utilizado em para certas técnicas é o *Dataset 1*, que contém gravações de notas avulsas de dois diferentes modelos de guitarra: *Fender Stratocaster* e *Ibanez RG2820*. Para cada guitarra, são disponibilizadas gravações de cada corda solta, bem como as 12 primeiras casas de cada corda. Entre os propósitos de sua criação está a estimação de múltiplas notas (*multi-pitch detection*) (EPPLER; KEHLING; MÄNNCHEN, 2016).

Ambos os *datasets* foram gravados com taxa de amostragem de 44100 e resolução de 16 bits por amostra (EPPLER; KEHLING; MÄNNCHEN, 2016).

6 ANÁLISE DE RESULTADOS

Neste capítulo, as diferentes técnicas e parâmetros de cada etapa serão comparados, com uma breve discussão dos resultados. Serão realizados diversos experimentos ao longo das etapas descritas no Capítulo 4, com ajustes de parâmetros em todo o conjunto de testes e, por fim, o sistema como um todo será comparado a um sistema independente.

6.1 Metodologia

As técnicas e parâmetros de cada etapa serão avaliados isoladamente, mantendo os parâmetros de outras etapas fixos. Ao final, a melhor combinação de parâmetros e técnicas será comparada com o resultado da aplicação de Mauch e Dixon (2010), utilizando os mesmos critérios, para validação dos resultados.

A avaliação será feita utilizando o conjunto de gravações anotadas do *Dataset 4*. Devido ao *hop size* escolhido, a saída do algoritmo será comparada com as anotações em intervalos de 0,1 segundo, a partir do primeiro instante anotado. Uma predição é considerada correta quando coincide com a anotação correspondente ao instante.

O principal critério a ser utilizado na avaliação será a acurácia do algoritmo em relação aos arquivos anotados, isto é, a razão entre o número de predições consideradas corretas e o número total de predições corretas e incorretas:

$$\mathrm{A} = \frac{Corretas}{Corretas + Incorretas}$$

A medida de acurácia será dividida em quatro subtipos, A_{24}, A_{60}, A_{61} e A_t, como descritos na Tabela 6.1. Para o subtipo A_{24}, serão analisadas somente as anotações do conjunto básico de 24 acordes (maiores e menores). Os demais acordes não serão modelados nessa versão do algoritmo. Quando a anotação se refere a um acorde fora do conjunto, a predição é descartada, não sendo considerada como correta ou incorreta. Da mesma forma, o subtipo A_{60} considera anotações em um conjunto de 60 acordes modelados (maiores, menores e diferentes tipos de sétima). O subtipo A_{61} considera e inclui na modelagem, além dos 60 acordes, a ausência de acorde, *NC*. Finalmente, o subtipo A_t considera todas as anotações, mesmo as que correspondem a acordes não modelados pelo algoritmo.

Tabela 6.1 – Acordes modelados e anotações consideradas para as diferentes medidas de acurácia

Medida	Acordes modelados	Anotações consideradas
A_{24}	24 (*maj*, *min*)	24 (*maj*, *min*)
A_{60}	60 (*maj*, *min*, *min7*, *7*, *maj7*)	60 (*maj*, *min*, *min7*, *7*, *maj7*)
A_{61}	60 (*maj*, *min*, *min7*, *7*, *maj7*) + *NC*	60 (*maj*, *min*, *min7*, *7*, *maj7*) + *NC*
A_t	60 (*maj*, *min*, *min7*, *7*, *maj7*) + *NC*	*Todas*

Fonte: O autor

A distinção entre A_{24}, A_{60} e A_{61} permite uma melhor contextualização dos resultados obtidos, revelando os efeitos para um conjunto menor de acordes, no caso de A_{24} e aumentando a abrangência, no caso de A_{60} e A_{61}. Ao contrário de A_{61}, a medida A_{60} não incluí a ausência de acorde, NC, pois sua modelagem é um caso especial que requer parâmetros. Já o subtipo A_t pode ser usado para avaliar e comparar o desempenho geral do algoritmo em relação a outras soluções.

No trecho de áudio analisado na Figura 6.1, cada linha representa um segmento de tempo da gravação. A coluna *Anotação* indica qual o acorde esperado para aquele instante de acordo com o arquivo de anotações correspondente à gravação de teste. A coluna *Predição* indica qual foi a saída do algoritmo para o mesmo instante. Finalmente, a coluna *Resultado* indica se a predição está correta ou incorreta, de acordo com a anotação, ou se foi descartada.

Figura 6.1 – Exemplo de saída do mecanismo de avaliação, considerando 61 acordes, para trecho inicial do arquivo *pop_4_120BPM.wav*

Tempo	Anotação	Predição	Resultado
6.00s	F	F	Correto
6.10s	F	F	Correto
6.20s	F	F	Correto
6.30s	F	F	Correto
6.40s	F	F	Correto
6.50s	Fsus9	F	Descartado
6.60s	Fsus9	F	Descartado
6.70s	Fsus9	F	Descartado
6.80s	Fsus9	F	Descartado
6.90s	Fsus9	F	Descartado
7.00s	F	F	Correto
7.10s	F	F	Correto
7.20s	F	F	Correto
7.30s	F	F	Correto
7.40s	F	F	Correto
7.50s	F	F	Correto
7.60s	NC	F	Incorreto
7.70s	NC	F	Incorreto
7.80s	NC	F	Incorreto
7.90s	NC	F	Incorreto
8.00s	NC	NC	Correto
8.10s	NC	NC	Correto

Fonte: O autor

Na imagem, o algoritmo considera o conjunto de 61 acordes para cálculo de A_{61}, de forma que o trecho com anotação *Fsus9* é descartado, mas o trecho *NC* é mantido. Considerando que há 13 predições corretas e 4 predições incorretas, a acurácia para o trecho pode ser calculada como $A_{61} = \frac{13}{13+4} \approx 76{,}4\%$. Se o parâmetro sendo calculado fosse A_t, as 5 predições feitas durante a anotação *Fsus9* seriam consideradas incorretas, resultando em $A_t = \frac{13}{13+9} \approx 59{,}0\%$.

6.2 Experimento 1: Diferentes tamanhos de quadro

Para o primeiro experimento, diferentes tamanhos de quadro (seção 4.2.1) foram testados. Em todos os casos, os outros parâmetros foram mantidos fixos. A função de janelamento escolhida foi o janelamento de Hann, por não exigir parâmetros e ser utilizada em outros trabalhos, como de Nadar et al (2019). Foi utilizada a técnica extração de notas musicais mais simples, via mapeamento direto da f_0. Para o cálculo de A_{24}, foram utilizados vetores binários para os 24 acordes básicos (maiores e menores). Para o cálculo de A_{60}, vetores para acordes estendidos (*min7, 7 e maj7*) foram incluídos.

Tabela 6.2 – Acurácia para diferentes tamanhos de quadro

Tam. quadro (amostras)	Tam. quadro (segundos)	A_{24}	A_{60}
4096	≈ 0,09	38,7%	9,5%
8192	≈ 0,19	60,1%	23,0%
16384	≈ 0,37	72,3%	42,2%
32768	≈ 0,74	78,8%	57,8%
65536	≈ 1,49	**82,5%**	**60,2%**
131072	≈ 2,97	81,5%	51,9%

Fonte: O autor

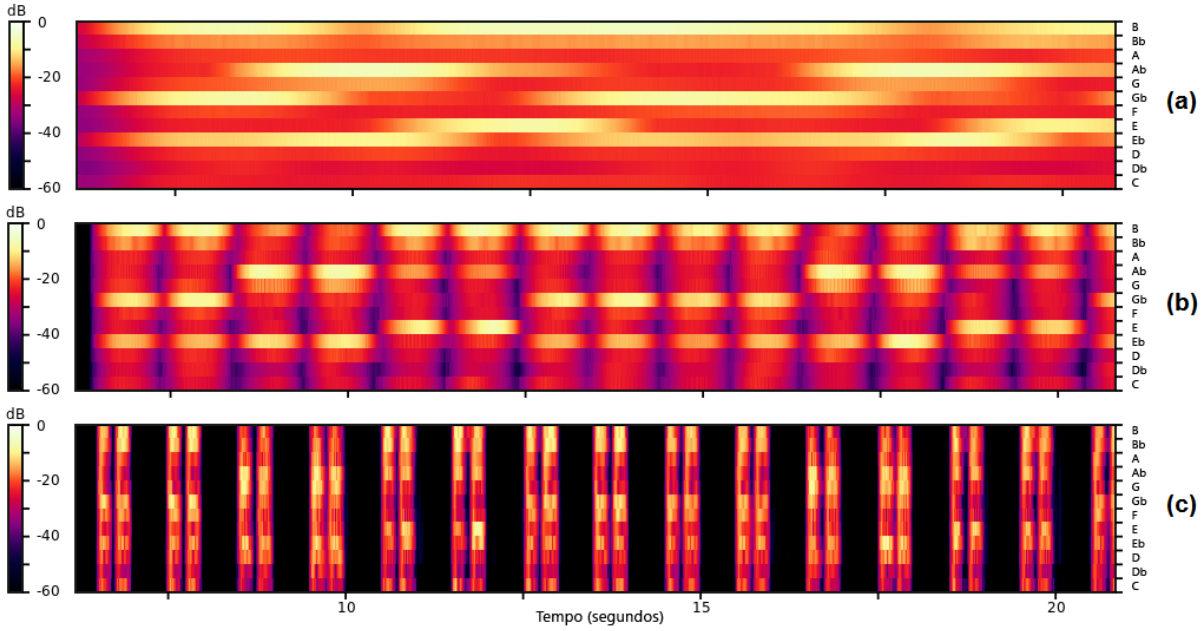
Um quadro muito estreito, como o quadro de 4096 amostras, é prejudicado pela baixa resolução nas frequências graves, onde as fundamentais das notas mais graves da guitarra se localizam. Em contrapartida, quadros muito largos, como o quadro de 131072 amostras, abrangem um espaço de tempo muito longo para capturar acordes individuais.

Embora o tempo em segundos da janela de 65536 amostras (~1,5 s) possa parecer relativamente longo para capturar mudanças de acordes, devemos levar em conta que a função de janelamento suaviza o começo e o fim do segmento de áudio, enfatizando apenas a sua região central.

A perda de resolução temporal causada por quadros muito grandes ainda tem uma vantagem bastante importante, pois em muitos casos de teste as notas do acorde são tocadas não simultaneamente, mas em sequência (arpejo), e um quadro mais longo combina essas notas que ocorrem em tempos diferentes em um mesmo espectro, causando um embaçamento do espectrograma ao longo do tempo.

A Figura 6.2 mostra os gráficos gerados pela aplicação para diferentes tamanhos de janela. O eixo vertical representa o cromagrama, isto é, a intensidade de cada classe de semitom para um dado instante.

Figura 6.2 – Gráficos gerados pela aplicação para diferentes tamanhos de quadro para o arquivo *reggae_2_120BPM.wav*, com os acordes *B* e *Abmin* separado por pausas *NC*.



Fonte: O autor

Em (a), é avaliado o tamanho de quadro de 131072 amostras (~3 segundos). As classes de semitom são bem definidas, sem vazamentos significativos para notas vizinhas, ou seja, há uma boa resolução de frequências. Porém, a resolução temporal sofre, misturando múltiplas tríades em um mesmo instante.

Por outro lado, em (c) temos o outro extremo, com 4096 amostras (~0,1 segundo), onde as notas musicais se espalham ao longo de múltiplos semitons, em função da baixa resolução das frequências. Já a resolução temporal é extremamente precisa, o que nem sempre é desejável.

O tamanho de quadro de 4096 amostras é capaz de capturar as pausas na gravação, características do ritmo do reggae, que seriam avaliadas corretamente como *NC*. Porém, em outros casos o embaçamento é desejado, pois permite que notas tocadas sequencialmente (como

arpejos) sejam consideradas um acorde. Algumas técnicas utilizam filtros para alcançar o embaçamento *após* a captura do espectro (CHO; BELLO, 2013).

Por fim, em (b), temos um meio termo entre os dois casos, com 32768 amostras. Embora exista um pequeno vazamento vertical das notas B e Ab (notas mais graves da gravação) para as suas notas vizinhas, o vazamento não é o suficiente para causar uma detecção incorreta. Não há sobreposição dos acordes, ao contrário de (a), porém a detecção das pausas entre acordes não é tão nítida quanto em (c).

6.3 Experimento 2: Diferentes funções e parâmetros de janelamento

No segundo experimento, são comparadas diferentes alternativas de janelamento (seção 4.2.2). A janela de Hann é comparada a diferentes parâmetros da janela de Kaiser. Também é comparado o efeito obtido ao não aplicar nenhuma função de janelamento ao segmento.

Como existe uma forte relação entre tamanho do quadro e efeito do janelamento, optou-se por avaliar cada janela em combinação com diferentes tamanhos de quadro. Os demais parâmetros foram mantidos fixos, com extração de notas via mapeamento da f_0 e vetores binários para 60 acordes.

Tabela 6.3 – Acurácia A_{60} para diferentes combinações de função de janelamento e tamanho do quadro

Tam. quadro (amostras)	Função de janelamento						
	Nenhuma	*Kaiser (β = 1)*	*Kaiser (β = 2)*	*Kaiser (β = 4)*	*Hann*	*Kaiser (β = 8)*	*Kaiser (β = 16)*
4096	8,2%	8,8%	9,8%	**10,3%**	9,5%	9,0%	6,9%
8192	23,0%	24,0%	25,3%	**25,4%**	23,0%	21,9%	34,7%
16384	42,6%	43,6%	**44,8%**	44,7%	42,2%	40,6%	34,9%
32768	55,4%	56,2%	57,6%	**58,5%**	57,8%	57,0%	53,3%
65536	49,1%	50,3%	53,4%	58,5%	60,2%	**60,9%**	**60,9%**
131072	33,0%	35,1%	39,8%	47,8%	51,9%	54,7%	**59,5%**

Fonte: O autor

As janelas na Tabela 6.3 estão ordenadas de acordo com o volume sob suas respectivas curvas. Com exceção dos quadros de 4096 e 8192 amostras, é interessante notar que para os quadros menores, janelas com formato mais largo são apropriadas, e para quadros menores, janelas mais estreitas, o que pode sugerir que o efeito está menos relacionado ao formato curvo específico da janela, e sim ao número de amostras significantes que captura.

Já para os tamanhos 4096 e 8192, as janelas de Kaiser com β = 2, β = 1 e ausência de janela possuem valores piores do que a janela de Kaiser β = 4. Uma possível explicação é a maior existência de artefatos nessas janelas, em função de uma suavização insuficiente das bordas do espectro.

6.4 Experimento 3: Diferentes deslocamentos de quadro

Ao realizar os experimentos iniciais, uma das primeiras características observadas foi que o algoritmo muitas vezes antecipa os acordes que estão por vir. A Figura 6.3 mostra um exemplo de predições antecipadas.

Figura 6.3 – Exemplo de antecipação de anotações pelo algoritmo de predição

Tempo	Anotação	Predição	Resultado
9.00s	A	A	Correto
9.10s	A	A	Correto
9.20s	A	A	Correto
9.30s	A	A	Correto
9.40s	A	E	Incorreto
9.50s	A	E	Incorreto
9.60s	E	E	Correto
9.70s	E	E	Correto
9.80s	E	E	Correto
9.90s	E	E	Correto
10.00s	E	E	Correto
10.10s	E	E	Correto
10.20s	E	E	Correto
10.30s	E	E	Correto
10.40s	E	E	Correto
10.50s	E	E	Correto
10.60s	E	E	Correto
10.70s	E	E	Correto
10.80s	E	E	Correto
10.90s	E	E	Correto
11.00s	E	Dbmin	Incorreto
11.10s	E	Dbmin	Incorreto
11.20s	Dbmin	Dbmin	Correto
11.30s	Dbmin	Dbmin	Correto
11.40s	Dbmin	Dbmin	Correto

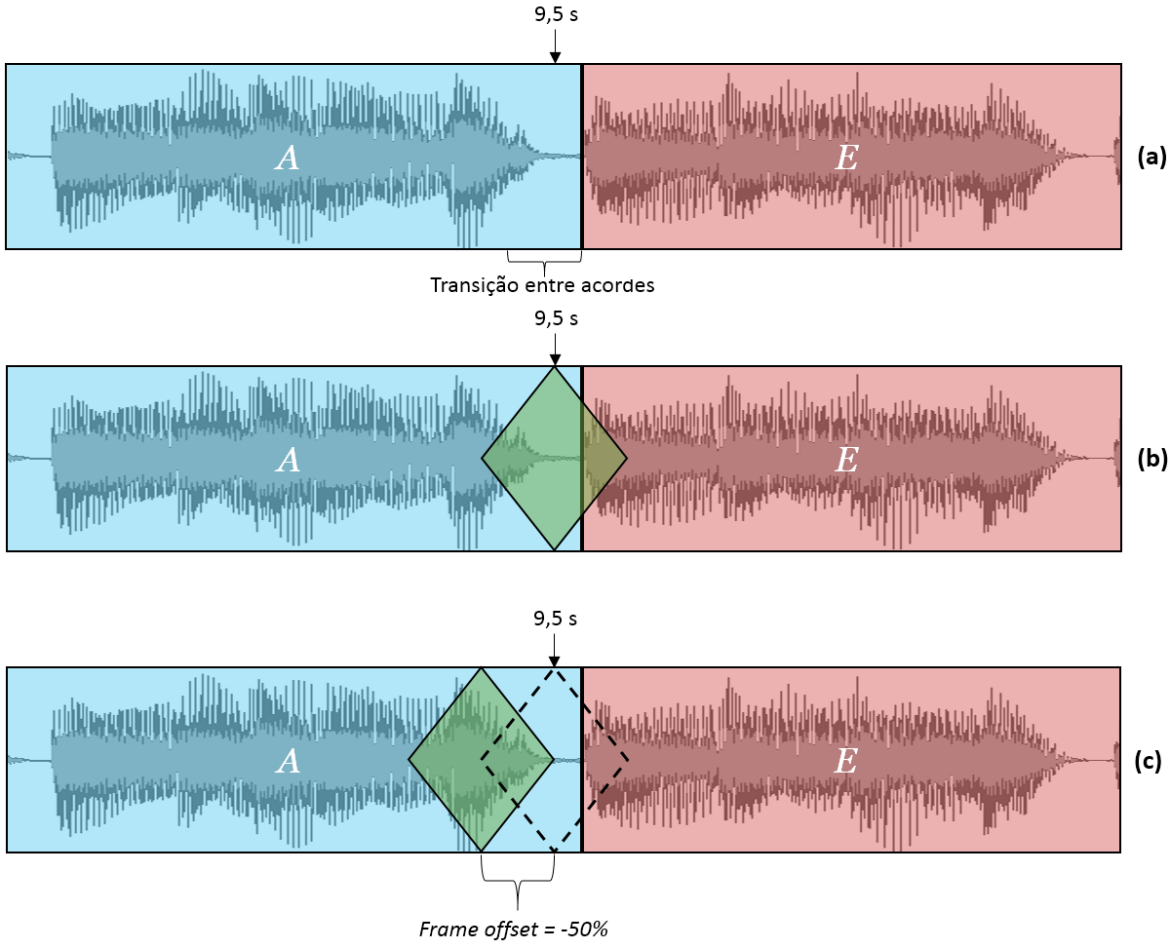
Fonte: O autor

Uma possível explicação para o fenômeno pode ser visualizada na Figura 6.4. Para determinar qual acorde está soando em 9,5 s de gravação, o algoritmo obtém um quadro centralizado nessa posição. Em (a) está marcado o período de transição entre os acordes, onde há uma batida nas cordas e um breve período de silêncio enquanto o guitarrista se posiciona para o próximo acorde. O losango verde em (b) representa aproximadamente o efeito do

janelamento do quadro, que captura a batida e o silêncio, e um breve trecho do acorde E, não capturando o acorde esperado, A.

A solução encontrada para o problema foi a criação de um parâmetro de deslocamento de quadro (*frame offset*), que indica a porcentagem de deslocamento em relação ao tamanho do quadro, variando de -50% a 50%. Em (c), é demonstrado um deslocamento de quadro de -50%, com o tracejado representando a leitura original para a posição 9,5 segundos, e o losango verde indicando a leitura após o deslocamento.

Figura 6.4 – Caso problemático de alinhamento de quadro para trecho de gravação anotado com os acordes *A* e *E*, e exemplo de deslocamento de quadro (*frame offset*)



Fonte: O autor

Para este experimento foram, portanto, avaliados diferentes valores para o deslocamento de quadro quando combinados com os diferentes tamanhos de quadro. Assim como no *Experimento 1*, foram mantidos como parâmetros fixos o janelamento de Hann e a modelagem para 24 e para 60 acordes, para cálculos de A_{24} e A_{60}, respectivamente.

Tabela 6.4 – Acurácia A_{24} para diferentes combinações de deslocamento de quadro e tamanho de quadro

Tam. quadro (amostras)	Deslocamento de quadro					
	-25%	*-20%*	*-15%*	*-10%*	*-5%*	*0%*
4096	**39,3%**	39,1%	38,9%	38,4%	38,5%	38,7%
8192	60,6%	60,8%	**60,9%**	60,5%	60,2%	60,1%
16384	73,5%	**73,9%**	73,7%	73,6%	72,7%	72,3%
32768	80,4%	80,6%	**80,9%**	80,6%	79,6%	78,8%
65536	80,3%	82,5%	84,2%	**84,8%**	84,1%	82,6%
131072	70,4%	75,0%	79,2%	82,1%	**83,0%**	81,5%

Fonte: O autor

Tabela 6.5 – Acurácia A_{60} para diferentes combinações de deslocamento de quadro e tamanho de quadro

Tam. quadro (amostras)	Deslocamento de quadro					
	-25%	*-20%*	*-15%*	*-10%*	*-5%*	*0%*
4096	9,8%	9,9%	**10,0%**	9,7%	9,4%	9,5%
8192	**23,9%**	23,8%	23,5%	23,5%	23,1%	23,0%
16384	**43,3%**	43,1%	43,0%	**43,3%**	42,7%	42,2%
32768	60,0%	**60,2%**	**60,2%**	59,2%	58,6%	57,8%
65536	61,1%	62,2%	**62,8%**	62,5%	61,7%	60,2%
131072	48,8%	51,7%	54,0%	**54,8%**	54,2%	51,9%

Fonte: O autor

Em todos os casos o deslocamento se mostrou benéfico em relação ao posicionamento original (0%). O deslocamento de quadro necessário para quadros maiores tende a ser menor, possivelmente pela relação direta entre o deslocamento absoluto em amostras e o tamanho do quadro. Há ainda um efeito positivo/negativo maior na acurácia para os quadros maiores analisados.

6.5 Experimento 4: Diferentes técnicas de extração de notas musicais

O quarto experimento compara as diferentes técnicas de extração de nota musical implementadas, como descritas na seção 4.3. Para o experimento, foi escolhido o tamanho de quadro de 65536 amostras, por apresentar melhores resultados nos experimentos anteriores. O janelamento de Hann foi utilizado, assim como em demais experimentos.

Tabela 6.6 – Acurácia para diferentes técnicas de extração de notas musicais

Técnica	A_{24}	A_{60}
Mapeamento direto da f_0	84,0%	63,6%
Janelamento gaussiano	83,1%	65,7%
Componentes harmônicos	**84,5%**	65,4%
Componentes harmônicos + Janelamento gaussiano	84,2%	**66,5%**

Fonte: O autor

Em relação ao mapeamento direto, a técnica de janelamento gaussiano, descrita por Cho e Bello (2013), causou uma pequena redução na acurácia para modelagem de 24 acordes, mas uma melhoria para a modelagem de 60 acordes. O mapeamento com componentes harmônicos, proposto na seção 4.3.3, trouxe melhoria em ambos os casos, embora limitada para o conjunto de 24 acordes.

Uma possível interpretação dos resultados é que o mapeamento direto é uma técnica adequada para um conjunto pequeno de acordes, tendo, porém, desvantagem para acordes que vão além do conjunto básico.

Como experiência adicional, foram combinadas as técnicas de componentes harmônicos e janelamento gaussiano, através da aplicação do janelamento às ativações resultantes da técnica de componentes harmônicos, que obteve os melhores resultados para a modelagem de 60 acordes.

6.6 Experimento 5: Diferentes formas de detecção da ausência de acorde

Na seção 4.5.2, foram discutidas duas formas de detecção da ausência de acordes. A primeira se baseia em um limiar fixo em dB, utilizada por Cho e Bello (2013), e a segunda em um limiar dinâmico, baseado na diferença relativa ao maior valor em dB encontrado na gravação até um dado instante.

Assim como nos demais experimentos, será utilizado o tamanho de janela de 65536 amostras, janelamento de Hann e extração de notas via mapeamento direto da f_0.

Tabela 6.7 – Acurácia para diferentes formas de detecção da ausência de acorde

Limiar fixo		Limiar dinâmico	
dB	A_{61}	*dB*	A_{61}
-23,0	54,9%	-5,0	58,0%
-24,0	58,0%	-6,0	59,1%
-25,0	59,8%	-7,0	**59,5%**
-26,0	**60,6%**	-8,0	**59,5%**
-27,0	60,4%	-9,0	59,2%
-28,0	59,7%	-10,0	58,8%
-29,0	59,2%	-11,0	58,5%

Fonte: O autor

O limiar fixo se mostrou vantajoso em relação ao limiar dinâmico. Embora tenha obtido resultados inferiores neste experimento, como nem todos os sinais de guitarra encontrados na prática serão normalizados para o mesmo volume em dB, o autor acredita que um limiar dinâmico possa ter maior aplicabilidade prática para o contexto proposto.

6.7 Experimento 6: Impacto da adição de modelos de acorde

Para este experimento, serão analisados os efeitos da inclusão de classes de acorde ao conjunto de acordes básicos, maiores e menores. Como descrito na Figura 5.2, os acordes maiores e menores compõem 42% do conjunto de acordes. O experimento avaliará os efeitos da adição sucessiva das classes de acorde *7*, *5*, *min7* e *maj7*. A classe de acorde *13* não foi inclusa porque sua modelagem é mais complexa, geralmente envolvendo omissão de certas notas.

Assim como nos demais experimentos, foram utilizados a janela de Hann e a extração de notas via mapeamento direto. O tamanho de quadro escolhido foi de 65536 amostras, por apresentar melhores resultados em experimentos anteriores.

Tabela 6.8 – Acurácia para diferentes conjuntos de acordes

Acordes modelados	A_{24}	A_m	A_t	Abrangência
24 (*maj, min*)	82,6%	82,6%	34,7%	42,0%
36 (*maj, min, 7*)	79,1%	73,5%	37,6%	51,1%
48 (*maj, min, 7, 5*)	49,7%	52,5%	30,5%	58,1%
48 (*maj, min, 7, min7*)	73,6%	67,5%	38,3%	56,7%
60 (*maj, min, 7, min7, maj7*)	64,3%	60,2%	36,2%	60,1%

Fonte: O autor

Na Tabela 6.8, A_{24} representa a acurácia observando apenas as anotações referentes aos 24 acordes maiores e menores, descartando as demais anotações. *Am* é a modelagem considerando anotações para todos os acordes do conjunto modelado. *At* é a acurácia total. *Abrangência* é a porcentagem de todas as anotações de teste que é coberta por cada escolha de acordes.

Comparando os resultados da coluna A_{24}, podemos perceber que a introdução de acordes adicionais à modelagem tem efeito negativo nos acordes já existentes (maiores e menores). Como já antecipa Müller (2015), o aumento do número de classes leva a um aumento na probabilidade de confusão durante a etapa de classificação.

Embora a escolha de 24 acordes básicos tenha uma grande acurácia para os acordes modelados (A_m), a abrangência desses acordes é de apenas 42% do conjunto de testes, limitando a máxima acurácia total possível a esse valor.

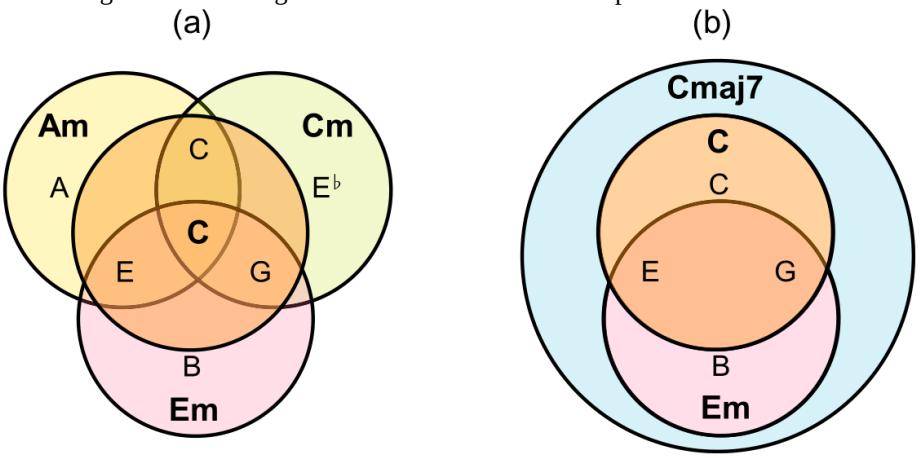
Ao incluir acordes da classe *7*, o algoritmo se torna mais abrangente, sendo limitado a 51,1% do conjunto de dados. Embora a acurácia para os 24 acordes básicos (A_{24}) seja prejudicada com essa adição, esse fato é compensado pelo aumento na acurácia total.

Ao adicionar acordes *5*, conhecidos como acordes de quinta ou *power chords*, embora o algoritmo tenha uma abrangência ainda maior do conjunto de dados, há uma redução da acurácia total. Isso porque a confusão criada por acordes de quinta, compostos apenas da fundamental e sua quinta, 7 semitons acima, é muito grande. Por exemplo, os semitons do acorde *C5* (semitons C e G) são subconjuntos não só dos acordes *C* e *Cmin* como de todas as suas variações com sétima.

Ao contrário da adição de acordes *5*, a adição da classe *min7* ao conjunto de 36 acordes, embora tenha abrangência menor do que no caso anterior, causa um aumento na acurácia total, pois as confusões criadas por essa adição são menores, sendo compensadas pelo aumento da abrangência.

Finalmente, a inclusão de acordes *maj7* causa uma redução na acurácia total, pois o ganho em abrangência é pequeno, não sendo suficiente para compensar a confusão criada por sua modelagem.

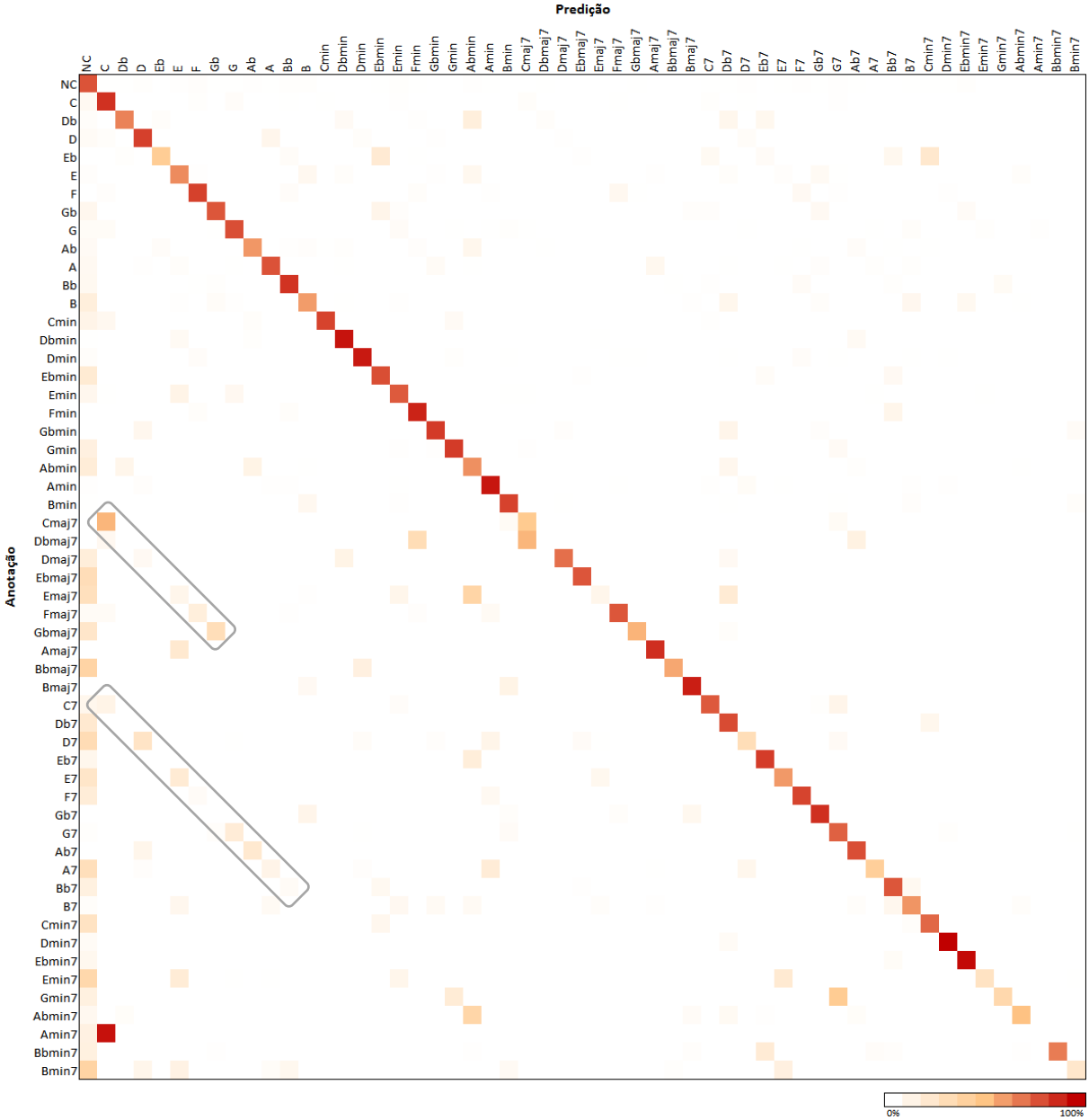
Figura 6.5 – Ambiguidade entre acordes causadas por notas em comum



Fonte: Müller (2015, p. 61)

A Figura 6.5 ilustra a ambiguidade causada por notas em comum entre diferentes acordes. Os símbolos em negrito representam acordes, e os demais símbolos representam classes de semitom. O sufixo *m* é utilizado para simbolizar acordes menores. Em (a), considerando apenas acordes básicos, o acorde *C* possui diferença de apenas uma nota para outras três tríades, *Amin*, *Cmin* e *Emin*. Já no caso (b), os acordes *C* e *Emin* estão completamente contidos no acorde *Cmaj7*. Considerando notas de intensidades diferentes e existência de sobretons harmônicos, a ambiguidade rapidamente se torna um problema.

Figura 6.6 – Matriz de confusão para modelagem com 61 acordes



Fonte: O autor

A Figura 6.6 apresenta a matriz de confusão observada na modelagem com 61 acordes. São omitidos 6 acordes que não ocorrem no conjunto de testes (*Bbmin*, *Gmaj7*, *Abmaj7*, *Dbmin7*, *Fmin7*, *Gbmin7*). Uma situação real do caso descrito por (b) no diagrama anterior pode ser vista na matriz de confusão, onde a predição *C* é utilizada para a anotação *Cmaj7* e também para a anotação *Amin7*. Esse tipo de ambiguidade pode ser observado visualmente através de diagonais que se formam na matriz de confusão, como a diagonal que parte da predição *C* com anotação *Cmaj7* e a diagonal que parte da predição *C* com anotação *C7* (em destaque).

6.8 Comparação com sistema independente

Para avaliação geral do algoritmo implementado, os resultados serão comparados com o sistema proposto e implementado por Mauch e Dixon (2010). A implementação[5], conhecida como *Chordino*, combina a técnica de NNLS Chroma, foco do estudo, com uma técnica de pós-filtro baseada em HMM (*Hidden Markov Models*), utilizando o algoritmo de *Viterbi*. Os autores alertam, contudo, que a técnica de pós-filtro não é considerada estado-da-arte.

O sistema independente foi escolhido por sua disponibilidade e facilidade de uso, sendo utilizado por diversos trabalhos, incluindo Cho e Bello (2013), como técnica padrão para comparação.

De acordo com Cho e Bello (2013), a técnica de pós-filtro baseada em HMM é utilizada pela maior parte dos trabalhos em reconhecimento de acordes, com poucas exceções. Os autores ressaltam o seu grande impacto positivo nos resultados da classificação. O presente trabalho, contudo, não implementa uma técnica de pós-filtro, sendo esse um ponto a ser explorado em trabalhos futuros.

Para comparação, as 64 músicas foram divididas em dois conjuntos de 32 músicas, sendo um conjunto dedicado exclusivamente para treino e outro conjunto para teste. Para cada conjunto, foram selecionadas de forma aleatória 4 músicas de cada gênero.

Para determinar os parâmetros a serem usados pelo sistema proposto na comparação, foram realizadas duas buscas extensivas de parâmetros através de *random search* no conjunto de treino, utilizando os resultados da primeira busca para então restringir os resultados. A partir das buscas foram escolhidos os parâmetros da Tabela 6.9.

Tabela 6.9 – Parâmetros escolhidos para o sistema proposto após busca de parâmetros

Parâmetro	Valor
Frame offset	-9%
Acordes modelados	61 (*maj, min, min7, 7, maj7 + NC*)
Tamanho do quadro	65536
Tamanho da janela	40553
Função de janelamento	Kaiser ($\beta = 5{,}6$)
Extração de notas	Comp. harmônicos + Janela gaussiana
Detecção de NC	-10,0 dB relativo

Fonte: O autor

[5] Disponível em: http://www.isophonics.net/nnls-chroma Acessado em: 18 de julho de 2019

Para o sistema independente, *Chordino*, foram utilizados os parâmetros padrão, os mesmos utilizados na submissão MIREX de 2010. Foram obtidos os seguintes resultados:

Tabela 6.10 – Resultados do sistema proposto comparados ao sistema independente (*Chordino*)

Sistema	A_{61}	A_t
Proposto	68,9%	**53,7%**
Independente	59,3%	47,0%

Fonte: O autor

Considerando apenas as anotações dos 61 acordes modelados, há uma grande vantagem do sistema proposto em relação ao sistema independente. A vantagem é um pouco menor considerando todas as anotações. Isso porque o sistema independente considera sempre um conjunto maior de acordes, além dos 61, como acordes *dim*, *dim7* e *b5*, não sendo a medida A_{61} apropriada para a comparação, sendo exibida apenas para referência.

Uma avaliação mais cuidadosa dos resultados, como mostra a Tabela 6.11, revela que por padrão o sistema independente possui muito baixa sensibilidade à ausência de acorde, *NC*, o que prejudica a performance do algoritmo. A anotação NC compõe 18,4% do conjunto de testes, sendo essencial para um bom desempenho.

Tabela 6.11 –*Precisão*, *Sensibilidade* e *F-measure* para diferentes classes de acorde no sistema independente (*Chordino*)

Classe	VP	FN	FP	*Precisão*	*Sensibilidade*	*F-measure*
maj	3485	735	3303	51,3%	82,6%	63,3%
min	1425	218	912	61,0%	86,7%	71,6%
min7	488	288	567	46,3%	62,9%	53,3%
7	573	700	573	50,0%	45,0%	47,4%
maj7	308	167	442	41,1%	64,8%	50,3%
NC	**219**	**2354**	**0**	**100,0%**	**8,5%**	**15,7%**
Total	6498	4462	5797	52,9%	59,3%	55,9%

Fonte: O autor

Na tabela, as colunas VP, FN e FP informam o número de verdadeiros positivos, falsos negativos e falsos positivos, respectivamente. A coluna *Precisão (precision)* indica a fração dos resultados retornados que é relevante. Para NC, todos os resultados retornados são relevantes, ou verdadeiros positivos (VP). Já a coluna *Sensibilidade (recall)* indica a fração de instâncias relevantes que é retornada. Como a grande maioria das anotações NC não é reconhecida, o valor é baixo. O valor *F-measure* é uma pontuação estatística que combina os valores de *Precisão* e *Sensibilidade*.

Para o entendimento da tabela e das medidas, é importante notar que os números em VP, FN e FP de cada classe de acorde são as somas desses valores para os 12 acordes de cada conjunto. Confusões entre acordes de mesma classe não são, portanto, consideradas corretas. Ainda, a linha Total é contabilizada considerando os conjuntos de acordes observados, através das somas de VP, FN e FP presentes na tabela.

Entre os parâmetros do sistema independente, existe um valor chamado *boostn*, que controla a sensibilidade à ausência de acorde, tendo originalmente valor 0,1. Foi então realizada uma busca utilizando o conjunto de treino, com granularidade 0,01, que resultou no valor 0,72 para *boostn*. Os resultados entre ambos os sistemas se tornam muito mais próximos, como mostra a Tabela 6.12.

Tabela 6.12 – Resultados do sistema proposto comparados ao sistema independente (*Chordino*), após ajuste de parâmetro para *NC*

Sistema	A_{61}	A_t
Proposto	68,9%	53,7%
Independente (*boostn = 0.72*)	68,7%	**54,3%**

Fonte: O autor

É interessante notar que o sistema proposto, embora tenha uma pequena vantagem para os 61 acordes modelados, perde em acurácia total, em função da maior abrangência do sistema independente.

Utilizando o novo valor para o parâmetro, pode também ser obtida também uma nova tabela para fins de comparação dos sistemas. A Tabela 6.13 compara os valores de *Precisão*, *Sensibilidade* e *F-measure* para o sistema proposto e o sistema independente, após ajuste do parâmetro *boostn*.

Tabela 6.13 – Medidas comparativas entre os sistemas proposto e sistema independente, após ajuste de sensibilidade a *NC* (*Chordino*)

Classe	*Precisão*		*Sensibilidade*		*F-measure*	
	Proposto	Indep.	Proposto	Indep.	Proposto	Indep.
maj	**64,7%**	62,5%	68,7%	**79,6%**	66,6%	**70,0%**
min	56,4%	**67,9%**	**77,2%**	73,2%	65,2%	**70,5%**
min7	28,5%	**58,5%**	**54,9%**	46,3%	37,6%	**51,7%**
7	34,8%	**64,8%**	61,5%	**73,5%**	44,5%	**68,9%**
maj7	35,0%	**74,5%**	**58,4%**	56,2%	43,7%	**64,0%**
NC	**77,1%**	42,7%	**78,0%**	58,7%	**77,6%**	49,4%
Total	53,7%	**58,5%**	**68,9%**	68,7%	60,3%	**63,2%**

Fonte: O autor

Embora a acurácia total dos sistemas seja semelhante, o sistema independente apresenta *F-measures* consistentemente superiores para as diferentes classes de acorde, com exceção da classe *NC* que, mesmo com o ajuste do parâmetro, continua com *F-measure* inferior ao sistema proposto.

Vale notar que grande parte dos exemplos de acorde do conjunto de testes são de classes maior e menor (42%), onde há uma diferença pequena de acurácia, e *NC* (18%), onde o algoritmo proposto tem melhores resultados. Acordes com sétima, nos quais o sistema independente tem maior vantagem sobre o proposto, compõem aproximadamente 18% do conjunto de testes.

Nenhum dos algoritmos, contudo, considera acordes onde a terça é omitida, os chamados *power chords*, com notação *5*, descritos no *Experimento 6*. Esses acordes são frequentemente utilizados em guitarras elétricas para os gêneros rock e metal, como é o caso do conjunto de testes utilizado. Da duração total do conjunto de testes, 7% dos acordes são *power chords*. Apesar disso, como mencionado no experimento, a confusão criada por esse acorde não torna benéfica sua inclusão no presente algoritmo.

A matriz de confusão do sistema independente, utilizando os parâmetros padrão, isto é, sem o ajuste de sensibilidade de ausência de acordes, pode ser encontrada no *Apêndice A*.

7 CONCLUSÃO

O presente trabalho explorou e propôs diversas técnicas e alternativas para o reconhecimento de acordes em sinal de áudio gerado por guitarra elétrica. Um conjunto de gravações de áudio de guitarra foi então utilizado para comparar o impacto do uso das diferentes técnicas em cada etapa do processamento. Por fim, a combinação dos melhores resultados de cada etapa foi comparada com um sistema independente, para referência.

A análise dos resultados revela que existem oportunidades para melhorias significativas em todas as etapas do processamento, com um o tamanho do quadro sendo de maior impacto, bem como o conjunto de acordes observado. Como esperado, o estudo também revelou sensibilidade do algoritmo em relação ao tamanho da janela de análise e ao número de classes (número de acordes) utilizados no processo de classificação.

É interessante notar que o sistema proposto, mesmo implementando um *pipeline* básico com o uso de técnicas simples, foi capaz de alcançar resultados bastante competitivos em relação ao sistema de referência. O sistema proposto obteve acurácia total de 53,7%, levemente superior à acurácia do sistema independente, que, após ajuste para detecção da ausência de acorde, obteve acurácia de 54,3%. O valor de *F-measure* dos sistemas também teve resultados comparáveis, com 60,3% para o sistema proposto e 63,2% para o sistema independente. Vale observar que grande parte do sucesso se deve à escolha de parâmetros ideais, e o mesmo processo aplicado a outros parâmetros do sistema independente poderia alcançar resultados superiores.

O uso de técnicas e algoritmos simples ainda permite uma grande aplicabilidade prática. Embora o foco do sistema não fosse a performance, o sistema se mostrou bastante eficiente, atingindo execuções 45 vezes mais rápidas que o tempo real de um arquivo de áudio em um computador desktop *Intel Core i5*, o que sugere que o algoritmo também é apropriado para uso em dispositivos móveis ou até mesmo sistemas embarcados.

Uma etapa não explorada pelo presente trabalho é a filtragem temporal, feita após a classificação dos acordes, que permite eliminar transições de acordes improváveis, sendo de grande impacto na detecção de acordes (CHO; BELLO, 2013), tornando-a um ponto de partida interessante para trabalhos futuros.

Ainda em trabalhos futuros, será interessante observar diferentes alternativas para modelagem de acordes, além de vetores binários, como a utilização de modelos de acordes obtidos através de médias de múltiplas amostras do acorde. Os modelos de acordes podem ser

aprendidos a partir de uma base de dados gerada artificialmente, como a base de dados gerada por Nadar et al (2019). Ainda, um grande volume de dados de teste possbilitaria o uso de técnicas mais avançadas, como *deep learning*, para treino e aprendizado das características que definem o acorde em um cromagrama.

O aprendizado de máquina também pode ser aplicado à etapa de extração de notas, utilizando as características e limitações do instrumento a seu favor, como o tempo de transição entre acordes, o limite de notas simultâneas e a restrição física humana com relação a diferentes combinações de casas.

Também pode ser explorada a fatoração das notas observadas no espectrograma a partir de timbres aprendidos em tempo real, de forma que o algoritmo se adapte a diferentes tipos de guitarra, formas de gravação e até mesmo outros instrumentos.

REFERÊNCIAS

ALVARADO, P., STOWELL, D. **Efficient Learning of Harmonic Priors for Pitch Detection in Polyphonic Music.** 2017.

BELLO, J.; PICKENS, J. A robust mid-level representation for harmonic content in music signals. **Proc. ISMIR,** p. 304–311, 2005.

BIASUTTI , M. Sharp low-and high-frequency limits on musical chord recognition. **Hear. Res.,** vol. 105, no. 1, p. 77–84, 1997.

CHENG, H. et al. Automatic chord recognition for music classification and retrieval. **Proc. ICME**, p. 1505-1508, 2008.

CHO, T., BELLO, J. On the Relative Importance of Individual Components of Chord Recognition Systems. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, Vol. 22, No. 2, 2014.

DAY, H.; PILHOFER, M. **Music Theory For Dummies.** Indianapolis, Indiana: Wiley Publishing, Inc.

EPPLER A.; KEHLING, K.; MÄNNCHEN, A. **IDM-SMT-GUITAR:** Dataset Description. 2016.

HARTQUIST, J. **Real-Time Musical Analysis Of Polyphonic Guitar Audio.** Faculty of California Polytechnic State University, 2012.

KAISER, J.; SCHAFER,R. On the use of the I0-sinh window for spectrum analysis. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, vol. 28, no. 1, pp. 105-107, 1980.

LEE, K. Identifying cover songs from audio using harmonic representation. **Proc. MIREX Audio Cover Song ID**, 2006.

LERCH, A. **An Introduction to Audio Content Analysis.** Wiley-IEEE Press, 2012.

LEVINE, M. **The jazz theory book.** Petaluna, CA, USA: Sher Music, 1995.

MAUCH, M.; DIXON, S. Approximate Note Transcription for the Improved Identification of Difficult Chords. **Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)**, 2010.

MAZHAR, F. **Automatic Guitar Chord Detection.** Tampere University of Technology, 2012.

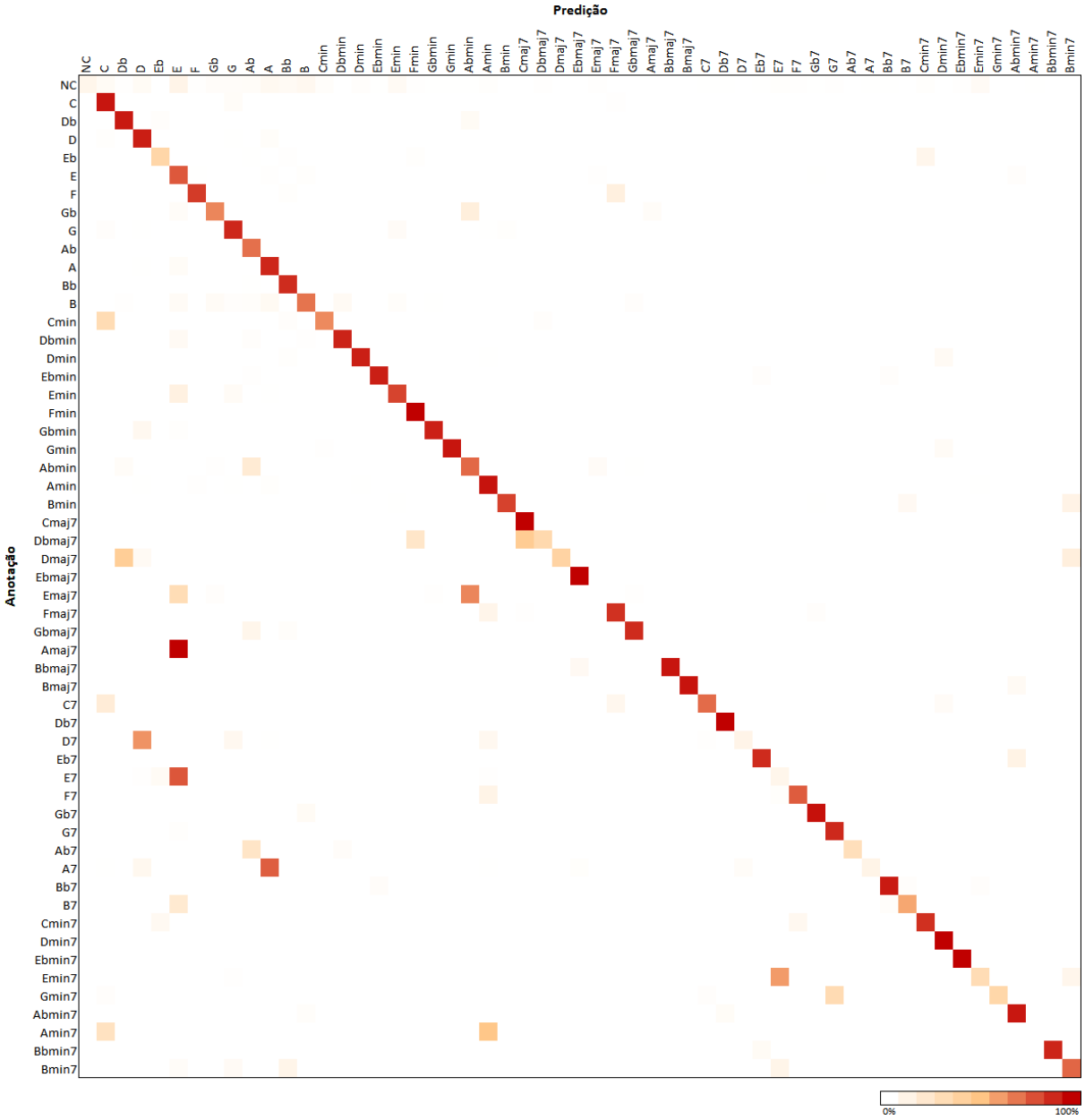
MED, B. **Teoria da Música.** 4. ed. Brasília, DF: Musimed, 1996.

MÜLLER, M. **Fundamentals of Music Processing:** Audio, Analysis, Algorithms, Applications. 1st ed. Springer Publishing Company, Incorporated, 2015.

NADAR, C. R.; ABEβER J.; GROLLMISCH S. Towards CNN-based Acoustic Modeling of Seventh Chords for Automatic Chord Recognition. **16th Sound & Music Computing Conference (SMC)**, 2019.

SHEH, A; ELLIS, D. Chord Segmentation and Recognition using EM-Trained Hidden Markov Models. **International Society for Music Information Retrieval Conference (ISMIR 2003)**, 2003.

APÊNDICE A – MATRIZ DE CONFUSÃO – SISTEMA INDEPENDENTE



APÊNDICE B – DATASET 4

Arquivo	Acordes presentes (ordem alfabética)
classical_1_80BPM.csv	A5, Amin, B7, B79b, Bmin, D, D11, D1113, D13, D7, Dmin, E7, Emin, F7, G, G5, Gbmin75b, NC
classical_2_60BPM.csv	A, Ab7, Absus4, D, Dbmin, NC
classical_3_60BPM.csv	C, F13, G7, NC
classical_4_80BPM.csv	B, B7, Bmin, E, G7, Gb7, Gbsus4, NC
classical_5_100BPM.csv	A, Bmin, D, Dbdim, Dbmin75b, E13, Fdim, G, Gb, NC
classical_6_100BPM.csv	Amin, B, B7, C, D5, Emin, G, NC
classical_7_60BPM.csv	A, Bmin, Bmin7, D, Dmaj7, E, G, NC
classical_8_120BPM.csv	A, A13, Amaj7, B, B7, E, Esus4, NC
country_1_150BPM.wav	C, G, NC
country_2_150BPM.wav	Bb, F, NC
country_3_120BPM.wav	C, F, G5, NC
country_4_100BPM.wav	E, NC
country_5_100BPM.wav	F5, G5, NC
folk_1_110BPM.wav	Emin7, G, G5, NC
folk_2_100BPM.wav	Bmin, C, D, D5, Dmin, Dsus4, Dsus9, G, NC
folk_3_180BPM.wav	Amin, Cmaj7, Edim, Emin75b, Fmaj7, G7, G713, NC
jazz_1_180BPM.csv	Bb, C7, C9, D7, F79, G7, NC
jazz_2_200BPM.csv	A7, Bb13, Bbmaj7, Bdim7, C7, D79b, F13, F7, Gmin7, NC
jazz_3_140BPM.csv	Bb7913, C713, NC
jazz_4_88BPM.csv	Abmin13, Abmin7, B7, Bb7, Db79, Eb7, Ebmin, Gb13, Gdim7, NC
jazz_5_100BPM.csv	Bbmin7, Eb79, NC
jazz_6_180BPM.csv	Ab713, Bb713b, Dbmaj7, Ebmin7, NC
jazz_7_180BPM.csv	Abmin7, Bbmin7, Bmaj7, Db79, Ebmin79, Gbmaj7, NC
jazz_8_120BPM.csv	Amin75b, Bbmaj7, Cmin7, D79b, Ebmaj7, F7, Gmin, NC
latin_1_200BPM.wav	C, Db, Eb, Fmin, NC
latin_2_120BPM.wav	Amin7, D79, NC
latin_3_120BPM.wav	Amin, E, E7, F, NC
latin_4_160BPM.wav	C, C7, Dmin, F, Gbdim, Gmin, NC
latin_5_180BPM.wav	Abmin, Db, Gb, NC
latin_6_140BPM.wav	Amin, Bb75b, Bmin75b, NC
latin_7_150BPM.wav	C7, F913, Fmaj7, G79, Gb7, Gmin7, NC
latin_8_180BPM.wav	B75b, Cmin7, Db7, Dmin7, NC
metal_1_180BPM.wav	E5, F5, G5, NC
metal_2_180BPM.wav	D5, E5, NC

(continua)

Arquivo	Acordes presentes (ordem alfabética)
metal_3_135BPM.wav	Emin, Eminmaj79, Gbmin75b, NC
metal_4_180BPM.wav	Bb5, Edim, F5, NC
metal_5_180BPM.wav	Cmin, E5, F5, Gmin, NC
metal_6_130BPM.wav	E5, Emin, Fdim, NC
metal_7_100BPM.wav	Bmin7, Emin7, NC
metal_8_125BPM.wav	Ab, Cmin, Dbmaj79, NC
pop_1_160BPM.wav	C, D, Emin, G, NC
pop_2_140BPM.wav	A, Dsus9, E, Gbmin, NC
pop_3_150BPM.wav	A, Ab, Dbmin, E, NC
pop_4_120BPM.wav	Bb, C, Dmin, F, Fsus9, NC
pop_5_90BPM.wav	Amin, Bb, Dmin, F, NC
pop_6_110BPM.wav	C, D, G, NC
pop_7_70BPM.wav	Abmin, B, Emaj7, NC
pop_8_100BPM.wav	Ab, B, Bb, Gbmaj7, NC
reggae_1_120BPM.wav	Bb, Cmin, NC
reggae_2_120BPM.wav	Abmin, B, E, NC
reggae_3_120BPM.wav	D, Emin, NC
reggae_4_120BPM.wav	Ab, Db, Ebmin, Gb, NC
rock_1_100BPM.wav	B5, E5, G, G5, Gb5, NC
rock_2_115BPM.wav	A, B, Dbmin, NC
rock_3_115BPM.wav	A, D, NC
rock_4_125BPM.wav	Eb, Eb5, F, NC
rock_5_120BPM.wav	Ab5, Bb5, Db5, F5, NC
rock_6_120BPM.wav	A7, D7, E7, NC
rock_7_120BPM.wav	Ab, Ab13, Bb, Bb13, Eb, Eb13, NC
rock_8_120BPM.wav	A, A13, B, B13, E, E13, NC
ska_1_180BPM.wav	Ab, Db, Eb, NC
ska_2_180BPM.wav[6]	Bb, C, F, NC
ska_3_180BPM.wav	A, E, Fdim, Gbmin, NC
ska_4_180BPM.wav	Bb, Eb, Fmin, NC

[6] Os acordes originais da anotação do arquivo *ska_2_180BPM.wav* (B, Db, Gb) estavam incorretos, descritos um semitom acima do valor correto. A anotação foi corrigida.