UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DETECÇÃO DE *ONSETS* EM NOTAS DE MÚSICAS INSTRUMENTAIS DE PIANO UTILIZANDO REPRESENTAÇÃO *PITCH* E APRENDIZADO DE MÁQUINA

Luciana Rolim Costa

Manaus
Julho de 2023

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DETECÇÃO DE *ONSETS* EM NOTAS DE MÚSICAS INSTRUMENTAIS DE PIANO UTILIZANDO REPRESENTAÇÃO *PITCH* E APRENDIZADO DE MÁQUINA

Luciana Rolim Costa

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Waldir Sabino da Silva Júnior

Manaus
Julho de 2023

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Costa, Luciana Rolim
C837d Detecção de onsets em notas de músicas instrumentais de piano utilizando representação pitch e aprendizado de máquina / Luciana Rolim Costa . 2023
82 f.: il. color; 31 cm.

Orientador: Waldir Sabino da Silva Júnior
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Detecção de onsets. 2. Representação pitch. 3. Aprendizado de máquina. 4. Recuperação da informação musical. 5. Notas de piano. I. Silva Júnior, Waldir Sabino da. II. Universidade Federal do Amazonas III. Título

LUCIANA ROLIM COSTA

DETECÇÃO DE ONSETS EM NOTAS DE MÚSICAS INSTRUMENTAIS DE PIANO UTILIZANDO REPRESENTAÇÃO PITCH E APRENDIZADO DE MÁQUINA.

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 03 de julho de 2023.

BANCA EXAMINADORA

Prof. Dr. Waldir Sabino da Silva Júnior, Presidente

Universidade Federal do Amazonas

gov.br Documento assinado digitalmente
GABRIEL MATOS ARAUJO
Data: 03/07/2023 20:33:30-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Gabriel Matos Araújo, Membro

Centro Federal de Educação Tecnológica -RJ

Profª. Dra. Eulanda Miranda dos Santos, Membro

Universidade Federal do Amazonas

À minha família.

Agradecimentos

- A Deus por renovar a minha fé, esperança e as minhas forças todos os dias.
- À minha mãe Ney e meu pai Eledi por sempre acreditarem na minha capacidade, por todo o apoio e sacrifícios que fizeram para me dar as oportunidades que tive. Vocês são as maiores inspirações da minha vida.
- Aos meus irmãos, e também engenheiros, Celso e Claudir, pelas conversas e distrações que me ajudaram nos momentos mais difíceis dessa jornada. Agradeço por todo apoio, incentivo e por me enxergarem de igual para igual. Vocês me inspiram demais.
- Às minhas amigas Kely, Adriana e Anne, pelas conversas, pelos momentos compartilhados, por todo apoio e incentivo.
- Por último, agradeço ao meu orientador prof. Waldir por me guiar ao longo de todo o processo, pela paciência, pelas conversas e por me incentivar a sempre buscar a excelência no que eu faço. Você é referência profissional para mim.
- O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo da Dissertação apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

DETECÇÃO DE *ONSETS* EM NOTAS DE MÚSICAS INSTRUMENTAIS DE PIANO UTILIZANDO REPRESENTAÇÃO *PITCH* E APRENDIZADO DE MÁQUINA

Luciana Rolim Costa

Orientador: Waldir Sabino da Silva Júnior

Programa: Pós-Graduação em Engenharia Elétrica

A análise de sinais de música e a extração de informações musicalmente relevantes para construir aplicações musicais fazem parte do campo de pesquisa de recuperação de informação de música (MIR, do inglês *music information retrieval*), dentro do qual a tarefa de detecção automática de *onsets* está inserida. Detectar *onsets* em sinais de música consiste em detectar os instantes de tempo de início dos eventos musicais contidos no sinal de música e esta tarefa, geralmente, serve de base para construção de aplicações como transcrição automática de música de um ou mais instrumentos musicais, alinhamento de áudio com *score*, estimação do tempo da música, dentre outros. Nesta dissertação, um sistema de detecção automática de *onsets* em sinais de música de piano usando aprendizado de máquina é apresentado. No *framework* proposto, a representação tempo-frequência *pitch* é utilizada e os classificadores investigados são máquina de vetor de suporte (SVM, do inglês, *support vector machine*), *gradient boosting* e rede neural convolucional de uma dimensão (CNN 1D, do inglês, *one dimensional convolutional neural network*). Os resultados dos experimentos realizados com as bases de dados BS1 e MAESTRO mostram que, na primeira abordagem, a SVM teve desempenho superior ao *gradient boosting* enquanto, na segunda abordagem, a métrica sensibilidade foi superior quando as características *pitch* foram utilizadas invés das características de espectrograma na base BS1.

Abstract of Dissertation presented to UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

NOTE ONSET DETECTION IN PIANO INSTRUMENTAL MUSIC USING PITCH REPRESENTATION AND MACHINE LEARNING

Luciana Rolim Costa

Advisor: Waldir Sabino da Silva Júnior

Department: Postgraduate in Electrical Engineering

The analysis of music signals and the extraction of musically meaningful information to build musical applications are part of the research field called music information retrieval (MIR), within which the task of automatic onset detection is inserted. The detection of onsets in musical signals consists in detecting the time instants of the beginning of the musical events contained in the musical signal, and this task, generally, serves as a base for building applications like automatic music transcription of one or more musical instruments, audio-score alignment, time estimation of music, among others. This dissertation presents a system for automatic onset detection in piano music signals using machine learning. In the proposed framework, the time-frequency representation pitch is used and the classifiers investigated are support vector machine (SVM), gradient boosting, and one-dimensional convolutional neural network (1D CNN). The experiments results made with the databases BS1 and MAESTRO show that, in the first approach, the SVM had superior performance compared with gradient boosting, while in the second approach, the sensibility metric was higher when using the pitch features instead of the spectrogram features on the BS1 database.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A audição, assim como cada um dos outros sentidos humanos, tem um papel crucial na vida do ser humano. Dependendo do tipo de som e da situação na qual estiver inserida, uma pessoa pode utilizar a informação sonora recebida para diferentes propósitos. Dentre os propósitos principais estão sobrevivência, comunicação ou lazer [1].

De forma similar, os sistemas de processamento digital de áudio são desenvolvidos com o intuito de fornecer serviços que supram uma ou mais necessidades dos indivíduos e essas necessidades, geralmente, estão relacionadas à facilitar a sobrevivência, estabelecer a comunicação e/ou obter uma forma de diversão.

Há vários tipos de sinais sonoros e as principais categorias nas quais os sinais de som podem ser organizados são sinais de eventos do dia-a-dia, sinais de voz e sinais de música [2]. A partir da análise desses sinais, inúmeras aplicações podem ser desenvolvidas.

E, dependendo do tipo de problema a ser resolvido, as informações extraídas do sinal de áudio podem conter todas as características necessárias para alcançar o objetivo da aplicação enquanto, em outros casos, essas informações podem ser usadas em conjunto com outros tipos de informações para construir um sistema mais abrangente.

O sistema desenvolvido por [3] é um exemplo de sistema de processamento digital de áudio que se encaixa dentro do propósito de sobrevivência, onde um sistema embarcado foi desenvolvido para medir o nível de som em uma linha de produção industrial com o objetivo de assegurar um ambiente que não causasse danos a audição

dos trabalhadores.

Ainda dentro do propósito de sobrevivência, podemos citar sistemas complementares às aplicações desenvolvidas por [4] e [5]. O primeiro trabalho propôs um sistema para identificar e rotular quatro classes de ações realizadas por operadores em uma linha de produção de uma fábrica através da análise de sequências de *frames* de um vídeo. Dentro desse contexto, sinais de áudio poderiam ser coletados e analisados com a finalidade de monitorar a condição de máquinas industriais [6], por exemplo, o que ajudaria na prevenção de acidentes.

No segundo trabalho, uma aplicação foi construída para monitorar os sinais vitais e movimentos de uma pessoa idosa de forma a fornecer o suporte necessário para garantir o seu bem-estar. A análise de sinais de áudio coletados no ambiente em que o idoso se encontra [7] poderia ser utilizada para complementar as informações fornecidas pelo sistema desenvolvido.

Para exemplificar um tipo de aplicação que se encaixa dentro do propósito de estabelecer a comunicação, utilizando o contexto do trabalho desenvolvido por [8], podemos citar um sistema que coleta sinais de áudio através de dispositivos móveis e os analisa para identificar o local em que a pessoa está situada, se é um restaurante, parque, estação de ônibus ou outro lugar [9], com o intuito de fornecer serviços personalizados.

Continuando com o propósito de estabelecer a comunicação, [10] propõe um sistema para detecção de sinal de áudio de teste em uma linha de produção de TVs para identificar o defeito de alto-falantes nos dispositivos produzidos. Esse sistema faz parte de um conjunto de sistemas desenvolvidos para automatizar processos dentro de uma linha de produção industrial, nos quais não somente áudio, mas imagens [11] e vídeos também são explorados.

Algumas aplicações utilizadas para fins de lazer que analisam sinais de música são o *SoundHound*[1] e o *Musixmatch*[2], através dos quais é possível recuperar informações de música, especificamente, o nome de uma determinada música, fornecendo ao sistema parte de sua melodia cantarolada, e a letra da música sincronizada com o seu respectivo áudio.

[1] https://www.soundhound.com/soundhound
[2] https://www.musixmatch.com/pt-br

Outro exemplo de aplicação é a plataforma *Spotify*[3] que, dentre outros serviços, é capaz de sugerir músicas baseado nas preferências do usuário. Por último, um sistema de transcrição de música chamado *ScoreCloud*[4], que tem a funcionalidade de transcrever um sinal de áudio monofônico, com voz ou com um único instrumento musical sendo tocado, em uma partitura digital. Essa aplicação pode ser utilizada tanto para lazer como para auxiliar no ensino de música.

A análise de sinais de música e a extração de informações musicalmente relevantes para construir aplicações musicais fazem parte do campo de pesquisa de recuperação de informação de música (MIR, do inglês *music information retrieval*). O MIR surgiu da necessidade de criar tecnologias que facilitassem o acesso, a recuperação e exploração de conteúdo digital musical [12], por conta da grande quantidade de arquivos digitais de músicas disponíveis, bem como seus vários formatos, áudio, MIDI (do inglês, *musical instrument digital interface*), partitura escaneada, dentre outros.

Nas últimas duas décadas, houve um aumento significativo no emprego de técnicas de aprendizado de máquina para tentar resolver problemas de processamento digital de sinais de música e de áudio, no geral [13]. Geralmente, as abordagens propostas são adaptadas para resolver um tipo específico de problema pois, modelos genéricos capazes de resolver mais de um tipo de problema tem um nível de complexidade maior, logo, alcançar desempenhos satisfatórios em cada tarefa realizada pelo sistema é mais trabalhoso.

Detecção de voz do cantor juntamente com estimação de *pitch* em um sinal de música polifônica [14], transcrição de notas do instrumento musical baixo em gravações de áudio de *Jazz* [15], detecção de emoções como tristeza, raiva e felicidade em sinais de fala [16] e um jogo de ritmos que utiliza a detecção de *onsets*, detecção do tempo da música e de gêneros musicais [17], são alguns exemplos de sistemas propostos que utilizam aprendizado de máquina.

A detecção automática de *onsets* é uma tarefa MIR que consiste em detectar, de forma automática, o tempo de início de cada evento musical contido em um sinal de música [18] e, geralmente, é uma das etapas iniciais na construção de outras aplicações.

[3] https://www.spotify.com/br/
[4] https://scorecloud.com/portugues/

A transcrição automática de música de um ou mais instrumentos musicais é um exemplo de aplicação na qual a detecção de tempos de *onsets* está contida pois, usualmente, os *pitches*, os tempos de *onsets* e os tempos de *offsets* de cada evento musical são necessários para fazer a transcrição.

O *pitch* é um atributo subjetivo que informa como uma determinada frequência ou um conjunto de frequências é percebido pelo ouvido humano. É através da estimação da frequência central do *pitch* que identificamos a nota musical presente em um sinal de áudio. O *offset*, por sua vez, corresponde ao instante de tempo que marca o término de um evento musical [19].

Diferentes algoritmos foram propostos ao longo dos anos para tratar o problema de detecção automática de *onsets* e as abordagens propostas podem ser divididas em quatro categorias [20]. A primeira categoria é composta por abordagens que usam somente técnicas de processamento digital de sinal para detectar os *onsets* e, geralmente, exploram características como a amplitude, fase e/ou frequência do sinal.

A segunda categoria é composta por abordagens que usam modelos ou dados estatísticos para resolver o problema. O modelo oculto de *Markov* (HMM, do inglês *hidden markov model*) e análise de componentes principais (PCA, do inglês *principal component analysis* [21–25]) são exemplos de técnicas utilizadas nessas abordagens.

Há também abordagens que usam aprendizado de máquina e podem pertencer às subcategorias supervisionado ou não-supervisionado e profundo ou não. Algoritmos de classificação como máquina de vetores de suporte (SVM, do inglês *support vector machines*, [22, 26]), algoritmos de agrupamento (do inglês *clustering*) e rede neural convolucional (CNN, do inglês *convolutional neural network*, [23, 27–30]) são alguns exemplos de técnicas utilizadas que fazem parte desta categoria.

Por último, há as abordagens que usam técnicas pertencentes a diferentes categorias como, por exemplo, técnicas de processamento digital de sinal juntamente com aprendizado de máquina ou outra combinação.

Um sistema de detecção automática de *onsets* em sinais de música de piano usando aprendizado de máquina é apresentado nesta dissertação. No *framework* proposto, a representação tempo-frequência utilizada é a *pitch* e três classificadores são investigados: SVM, *gradient boosting* e CNN 1D. Duas abordagens foram em-

pregadas, uma com os classificadores SVM e *gradient boosting* e outra com a CNN 1D.

A primeira abordagem foi dividida em três etapas onde, na primeira etapa, os sinais de áudio de piano são convertidos para a representação *pitch*, em seguida, os vetores de características *pitch* são extraídos e, posteriormente, fornecidos ao classificador que, os analisa, indentifica os padrões contidos nos vetores de características e gera um modelo para classificação.

Na segunda etapa, o sinal de música de entrada passa pelos mesmos estágios da primeira etapa até a extração de características, depois, as características *pitch* são fornecidas como entrada para o modelo gerado pelo classificador, e este, categoriza cada um dos vetores de características *pitch* como *onset* ou não-*onset*[5].

Por último, ocorre a etapa de pós-processamento, na qual os instantes de tempo dos *onsets* dos vetores de características *pitch* são obtidos e uma busca é feita para verificar quais instantes de tempo estão localizados dentro da janela de tolerância.

Na segunda abordagem, apenas as duas primeiras etapas da primeira abordagem são realizadas, ou seja, não há etapa de pós-processamento.

Experimentos foram feitos usando as bases de dados BS1 e MAESTRO e, no geral, na primeira abordagem, o classificador que apresentou os melhores resultados foi o SVM, pois o *gradient boosting* fez muitas detecções espúrias. Na segunda abordagem, a métrica sensibilidade foi superior quando as características *pitch* foram utilizadas invés das características de espectrograma na base BS1. Na base MAESTRO, o desempenho da CNN 1D foi superior usando as características de espectrograma.

1.1 Objetivos

Os objetivos principais da dissertação são:

- Construir um *framework* para a detecção automática de *onsets* no qual diferentes tipos de sinais de áudio possam ser utilizados e cuja estrutura possa

[5]O termo não-*onset* se refere a região que contém características que indicam a ausência de *onset*.

ser expandida. O objetivo é que o *framework* proposto possibilite a análise de outros tipos de sinais, como os sinais de voz e sinais sonoros de eventos do dia-a-dia, além dos sinais de música.

Dentro do contexto de análise de sinais de música, os tempos de *onsets* detectados podem ser utilizados, em conjunto com outras informações, para identificar as notas musicais e fazer a transcrição dessas notas para uma representação simbólica.

Em outro exemplo de aplicação, os tempos de *onsets* podem ser usados para descobrir a assinatura de tempo da peça musical e, assim, identificar quantos *beats* são tocados por minuto, informação essa que pode ser utilizada para, posteriormente, identificar o gênero da peça musical.

No contexto de análise de sinais de voz, geralmente, o objetivo ao analisar esses tipos de sinais consiste em detectar e identificar as palavras contidas na gravação de áudio e, neste sentido, os tempos dos *onsets* são informações necessárias para realizar essa tarefa. A identificação desses eventos pode ser utilizada na construção de sistemas de segurança que utilizam o reconhecimento automático de voz ou na construção de assistentes virtuais por voz, por exemplo.

Por último, em sinais sonoros de eventos do dia-a-dia, os tempos de *onsets* detectados podem ser utilizados para compor sistemas que identifiquem a presença de um sinal sonoro específico e, a partir desse reconhecimento, garantam a qualidade de um produto, que é o caso de alguns sistemas de automatização de processos industriais, por exemplo.

- Investigar uma representação diferente da representação do sinal de som no domínio do tempo, especificamente, a representação tempo-frequência *pitch*.

  A representação no domínio do tempo permite visualizar, em alguns casos, as fases de ataque, sustentação e liberação do sinal de música, porém, à medida que os sons se tornam mais complexos e, por consequência, há mais sobreposições de sons e a presença de ruídos, se torna mais difícil identificar características do sinal.

  Além disso, a representação no domínio do tempo do sinal sonoro não fornece

informações que nos ajudem a identificar os instrumentos tocados, as notas tocadas e outras características mais detalhadas do sinal de música. Por conta disso, neste trabalho investigamos uma representação alternativa à representação no domínio do tempo, a representação *pitch*.

- Propor uma solução com CNN 1D para a detecção automática de *onsets* em sinais de música de piano.

  A maioria das abordagens de estado da arte propostas para detectar os tempos de *onsets* de forma automática em sinais de música utiliza a CNN 2D ou 3D, por isso, neste trabalho propomos e investigamos uma solução utilizando a CNN 1D.

1.2 Contribuições

Quando se trata da representação do sinal de áudio, o espectrograma é a representação tempo-frequência comumente utilizada na construção de sistemas de detecção de *onset* [31–33]. O espectrograma mostra como as intensidades das componentes de frequências do sinal de música variam ao longo do tempo [34], logo, um coeficiente é associado a uma componente de frequência do sinal.

Uma representação mais compacta, no entanto, na qual um coeficiente é associado a uma banda de frequências do sinal, é uma opção alternativa que pode reduzir o custo computacional da tarefa de detecção automática de *onsets*. Por isso, propomos investigar a representação tempo-frequência *pitch*, que mede como a energia média quadrática em tempo curto em cada banda de frequência varia ao longo do tempo [35]. A representação *pitch* é composta por um número fixo de bandas de frequências, especificamente oitenta e oito.

Adicionalmente, outro fator observado na etapa de classificação dos sistemas de detecção automática de *onsets* do estado da arte, é o uso frequente da CNN como classificador, seja sozinha ou em conjunto com outros classificadores [27,28,31]. Porém, a estrutura de dados de entrada utilizada com mais frequência tem sido a 2D ou 3D. Com base nisso, propomos explorar a utilização da CNN 1D na tarefa de detecção automática de *onsets*.

Por último, apesar de o uso de métodos profundos ser predominante na cons-

trução de sistemas de detecção de *onsets* do estado da arte, propomos investigar também os métodos clássicos SVM e GB, tendo em vista ambientes nos quais a captura de áudios para formar a base de dados é mais difícil, resultando em bases pequenas disponíveis para avaliar os sistemas construídos. Um exemplo de tais ambientes é a linha de produção de uma indústria.

Tendo em vista os fatores mencionados, as contribuições dessa dissertação são:

- Investigar a representação tempo-frequência *pitch*, uma representação alternativa ao espectrograma, no contexto de detecção automática de *onsets*.
- Propor um *framework* para detecção automática de *onsets* no qual duas abordagens são investigadas:
  - Utilizando rotulagem manual de submatrizes *pitch* e os classificadores de aprendizado de máquina clássico SVM e GB.
  - Utilizando a CNN 1D, sem rotulagem manual das submatrizes *pitch*.
- Utilizar *cross-validation dataset* para avaliar as duas abordagens propostas.

1.3 Organização da Dissertação

Esta dissertação, como um todo, é composta por seis capítulos: introdução, fundamentos teóricos, trabalhos relacionados, metodologia para detecção de *onsets* em notas de piano, procedimento experimental e, por último, conclusão.

No capítulo atual, o de introdução, o contexto dentro do qual a tarefa de detecção automática de *onsets* está inserida foi fornecido, o sistema proposto foi introduzido e os objetivos principais desta dissertação foram abordados.

No segundo capítulo, os principais conceitos e técnicas empregados ao longo do desenvolvimento do sistema de detecção automática de *onsets* serão abordadas. Dentre as quais estão conceitos de teoria musical, características de sinais de música, *onsets* de notas musicais, as representações tempo-frequência espectrograma e *pitch*, conceitos de aprendizado de máquina relacionados ao sistema proposto e as características principais dos classificadores investigados, SVM, *gradient boosting* e CNN 1D.

No terceiro capítulo, como resultado de uma revisão bibliográfica, as características principais dos trabalhos relacionados ao sistema proposto serão descritas, uma tabela de comparações destes trabalhos será fornecida e, por último, uma análise desta tabela de comparação será realizada.

No quarto capítulo, a metodologia utilizada para desenvolver o sistema de detecção automática de *onsets* será descrita com detalhes. No geral, a metodologia foi dividida em três etapas principais: construção de modelo para classificação, classificação dos vetores de características *pitch* e pós-processamento. Neste capítulo, cada uma dessas etapas será explicada com detalhes.

No quinto capítulo, as bases de dados utilizadas para desenvolver o sistema serão descritas, os procedimentos experimentais realizados serão detalhados, dentro dos quais está a especificação da métrica utilizada e, por último, os resultados obtidos serão descritos e analisados.

Por fim, no capítulo de conclusão, as características principais do sistema proposto serão relembradas, considerações finais sobre o sistema desenvolvido e os resultados obtidos serão realizadas e sugestões para trabalhos futuros serão feitas.

Capítulo 2

Fundamentos Teóricos

Neste capítulo, os principais conceitos e técnicas utilizadas ao longo do desenvolvimento do sistema de detecção de *onsets* proposto serão abordados. Conceitos de teoria musical como melodia, harmonia, ritmo e oitava musical. As principais características de sinais de música também serão abordadas, incluindo os principais atributos *pitch*, *loudness* e timbre, e o conceito de *onset*. Por último, as representações tempo-frequência espectrograma e *pitch*, além dos algoritmos SVM, *gradient boosting* e CNN 1D, estão entre os assuntos que serão apresentados.

2.1 Conceitos de teoria musical

A música é composta por três elementos fundamentais: melodia, harmonia e ritmo [36]. A melodia é percebida quando um conjunto de notas musicais[1] é tocado em um instrumento musical sequencialmente. Quando uma pessoa cantarola uma música, por exemplo, a sua voz simula a melodia.

A harmonia, por sua vez, é percebida quando um conjunto de notas é tocado ao mesmo tempo. No piano, por exemplo, quando várias teclas, que representam as notas musicais, são pressionadas simultanemente, uma harmonia é gerada.

Por último, o ritmo está associado ao padrão dos intervalos existentes entre as notas musicais tocadas. Ao escutar uma música, geralmente, conseguimos identificar que o refrão tem um ritmo diferente do restante da música.

[1] Notas ou tons musicais são sons produzidos pela vibração regular gerada ao tocar instrumentos musicais.

Outro conceito musical relevante é o de oitava. Considerando dois tons produzidos por um instrumento musical, caso a frequência fundamental do segundo tom seja o dobro da frequência fundamental do primeiro tom, dizemos que o segundo tom está uma oitava acima em *pitch*[2] [37]. Quando isso acontece, se essses dois tons forem tocados simultaneamente, perceberemos sons bem similares. O conceito de *pitch* será explicado com mais detalhes na seção 2.2.

A escala musical é um conjunto de notas musicais ordenadas seguindo um padrão que define a sequência das notas e seus respectivos intervalos. Geralmente, as notas são ordenadas da nota mais grave para a mais aguda, ou seja, da nota com menor frequência fundamental para a nota com maior frequência fundamental. Além disso, o padrão empregado nas notas contidas em uma oitava se repete nas oitavas seguintes.

Uma escala comumente utilizada é a escala igualmente temperada de doze tons, ilustrada na Figura 2.1, cuja oitava é divida em doze passos de escala, onde os passos são os intervalos entre duas notas. Pelo fato de as frequências fundamentais das notas serem geradas usando a mesma taxa de frequência, as notas dessa escala são igualmente espaçadas.

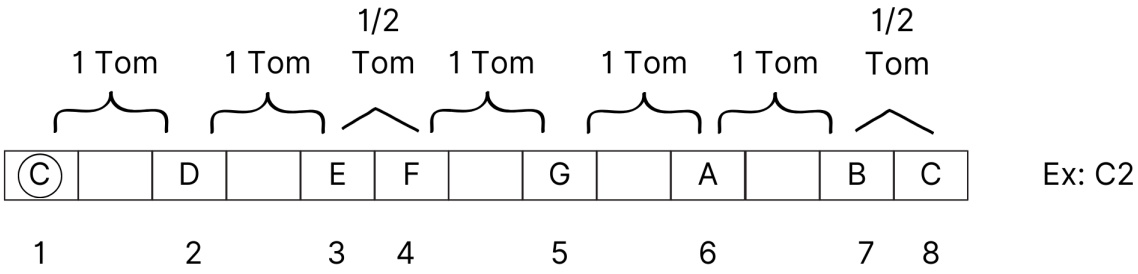

Figura 2.1: Uma oitava da escala musical igualmente temperada de doze tons.

As notas musicais contidas em uma oitava da escala igualmente temperada de doze tons seguem a sequência ilustrada na tabela 2.1 e quando a nota C se repete, isso indica que a segunda nota C está uma oitava acima em *pitch* da primeira nota C.

Cada nota é representada por dois componentes: o *chroma* e a altura do tom. O *chroma* ou cor da nota é especificado pelo conjunto de símbolos $\{C, C^{\#}, D, ..., B\}$,

[2] *Pitch* é um atributo subjetivo que informa como uma determinada frequência ou um conjunto de frequências é percebido pelo ouvido humano.

enquanto a altura da nota é representada por números situados após os *chromas* que indicam a oitava a qual a nota musical pertence. Por exemplo, $\{C1, C2, ..., C_N\}$, onde N indica o número máximo de oitavas contido na escala. Quanto maior for a oitava, maior será a altura do tom.

Símbolo da nota	**Nome da nota**	**Oitava da nota**
C	Dó	1
$C^{\#}$	Dó sustenido	1
D	Ré	1
$D^{\#}$	Ré sustenido	1
E	Mi	1
F	Fá	1
$F^{\#}$	Fá sustenido	1
G	Sol	1
$G^{\#}$	Sol sustenido	1
A	Lá	1
$A^{\#}$	Lá sustenido	1
B	Si	1
C	Dó	2

Tabela 2.1: Sequência de notas musicais contidas em uma oitava da escala igualmente temperada de doze tons.

2.2 Características de sinais de música

Sinais de música, geralmente, são compostos por um conjunto de sons ou notas musicais cujas frequências centrais se aproximam de uma estrutura de frequências definida pela escala musical utilizada. E essa é uma das características que diferencia os sinais de música de outros sinais de som, como os de fala e os eventos sonoros do dia-a-dia, por exemplo.

Os sinais de fala são modelados de acordo com os fonemas existentes em uma determinada língua, ou seja, pelos sons produzidos pelos falantes dessa língua. Já os sons do dia-a-dia, como o som da buzina de uma carro, de passos de uma pessoa ou de um apito, por exemplo, tem características variáveis e não seguem uma estrutura, como os sinais de música e os de fala.

Os sinais de música podem ser monofônicos ou polifônicos. Sinais monofônicos são caracterizados por possuir uma única melodia ao longo do tempo, isto é,

uma única nota musical é tocada por vez. Diferente dos sinais polifônicos, nos quais várias melodias são tocadas simultaneamente, ou seja, vários tons musicais são tocados ao mesmo tempo. Em termos de frequências fundamentais, o sinal monofônico tem uma única frequência fundamental enquanto o sinal polifônico é composto por várias frequências fundamentais mescladas [38].

Os principais atributos dos sinais de música são: *pitch*, *loudness* e timbre. Os três são propriedades perceptuais e, os dois primeiros, estão associados à propriedades físicas, como ilustrado na tabela 2.2.

Propriedade física	**Propriedade perceptual**
Frequência	*Pitch*
Intensidade do som	*Loudness*
—	Timbre

Tabela 2.2: Principais propriedades perceptuais dos sinais de música e as propriedades físicas as quais estão associados.

O *pitch* é um atributo subjetivo que informa como uma determinada frequência ou um conjunto de frequências é percebido pelo ouvido humano. O *pitch* não corresponde à frequência, pois o primeiro, é uma propriedade perceptual e, o segundo, uma propriedade física, mas a frequência é importante para a determinação do *pitch*.

Para exemplificar a diferença existente entre os dois, de acordo com [39], uma leve mudança de frequência, geralmente, não ocasiona uma mudança perceptível para o ouvido humano e, por conta disso, usualmente, um único *pitch* está associado a uma faixa de frequências .

Os tons produzidos por instrumentos musicais são complexos, pois são resultantes da combinação de formas de onda de puro tom, isto é, senóides. Essas formas de onda, chamadas de harmônicas, tem frequências que são múltiplas inteiras da frequência fundamental. A harmônica com a menor frequência, a primeira harmônica, tem a frequência denominada de fundamental enquanto as outras harmônicas são chamadas de *overtones*. É a frequência da harmônica fundamental que determina o *pitch* da nota musical [37].

A relação entre frequência e *pitch* é mais clara quando se trata de senóides, pois as suas frequências são fixas, porém, em tons ou notas musicais, essa relação fica mais complexa, pois estes tons são compostos por uma superposição de senóides

com frequências diferentes que mudam ao longo do tempo. Pelo fato de o *pitch*, geralmente, estar associado a um conjunto de frequências, há uma frequência que é considerada a central e a relação entre frequência central e o *pitch* pode ser descrita pela fórmula 2.1:

$$F_{pitch}(p) = 2^{(p-69)/12}.440 \tag{2.1}$$

Onde $F_{pitch}(p)$ é a frequência central do MIDI[3] *pitch* p, um número que representa o *pitch* de uma nota musical, formato esse adotado no protocolo MIDI (do inglês *musical instrument digital interface*), e que pode assumir os valores no intervalo $[0 : 127]$.

A frequência $440Hz$ é a frequência referência do *pitch* de número 69 que, geralmente, é adotada como padrão para afinar um conjunto de instrumentos musicais. O número doze, pelo o qual o expoente é dividido, diz respeito aos doze semitons contidos em uma oitava, onde um semitom representa o intervalo entre duas notas musicais subsequentes, cuja unidade de medida é *cents*.

Outro atributo subjetivo associado a um atributo físico é a *loudness*. A *loudness* está relacionada a como o ouvido humano percebe a intensidade do som e nos permite organizar os sons em uma escala que se estende do som mais suave até o som mais alto [40]. A intensidade sonora ou volume do som mede a quantidade de energia do som emitida pela fonte sonora por unidade de área e é medida em *Watts* por metro quadrado (W/m^2).

A escala usualmente utilizada para medir a *loudness* é a escala logarítmica decibel (dB). Ao tocar uma peça musical, uma propriedade do som frequentemente considerada é a dinâmica, que está associada ao volume do som e, por consequência, à *loudness*. Em um piano, por exemplo, quanto mais forte a tecla for pressionada, maior será o volume do som da nota musical, inversamente, quanto mais suave a tecla for pressionada, menor será o volume do som. Logo, a dinâmica altera a forma com a qual o volume da nota musical é percebido.

Há outros fatores que influenciam na *loudness* percebida, dentre esses fatores estão a idade da pessoa, a duração do som e a frequência. A idade influencia porque

[3] *MIDI* é um protocolo que possibilita a comunicação de instrumentos musicais eletrônicos com computadores e outros dispositivos eletrônicos similares.

pessoas mais velhas processam os sons de forma diferente de pessoas mais jovens pois, com o aumento da idade, o ouvido humano se torna mais sensível, ou seja, perde a habilidade de ignorar sons secundários e de focar nos sons principais [41].

A duração do som também altera a forma com a qual o ouvido humano percebe a intensidade sonora, visto que um aumento na duração do som faz com o que esse som pareça mais alto em volume [42]. Por último, a frequência também influencia na *loudness* pois, ainda que dois sons tenham a mesma intensidade, se os mesmos tiverem frequências diferentes, a *loudness* percebida desses sons pode ser diferente [39].

O último atributo principal do sinal de música é o timbre, que também é um atributo perceptual. O timbre é uma propriedade que nos permite distinguir sons tocados por instrumentos musicais diferentes como, por exemplo, uma mesma nota musical tocada em um piano e em um violino. As diferenças nas características sonoras da nota musical são perceptíveis por conta do timbre.

Medir o timbre não é uma tarefa simples, no entanto, as diferenças de timbre nos tons musicais podem ficar mais evidentes quando analisamos as intensidades das frequências dominantes que compõem o tom musical, ou seja, as parciais, em uma representação tempo-frequência como o espectrograma, que será explicado com mais detalhes na seção 2.4.

As Figuras 2.2 e 2.3 mostram os espectrogramas da nota musical Si na oitava quatro (B4) tocada por dois instrumentos musicais diferentes, o piano e o violino, respectivamente. As frequências das parciais são representadas pelas linhas horizontais e a tonalidade da cor especifica a intensidade das frequências.

Particularmente, nestes espectrogramas, a cor branca representa a intensidade máxima e a cor cinza representa a intensidade mínima. Analisando os espectrogramas, observamos que na Figura 2.2, a maior energia acumulada ocorre, aproximadamente, nos primeiros $0,10$ segundos e depois decai ao longo do tempo.

Em constraste, na Figura 2.3, em algumas parciais, a energia máxima se prolonga por um intervalo de tempo maior e, além disso, as parciais se espalham ao longo de todo o espectro de frequências. Essas características demonstram algumas diferenças existentes entre o tom musical de piano e o de violino, e a análise das intensidades das frequências das parciais de tons produzidos por instrumentos

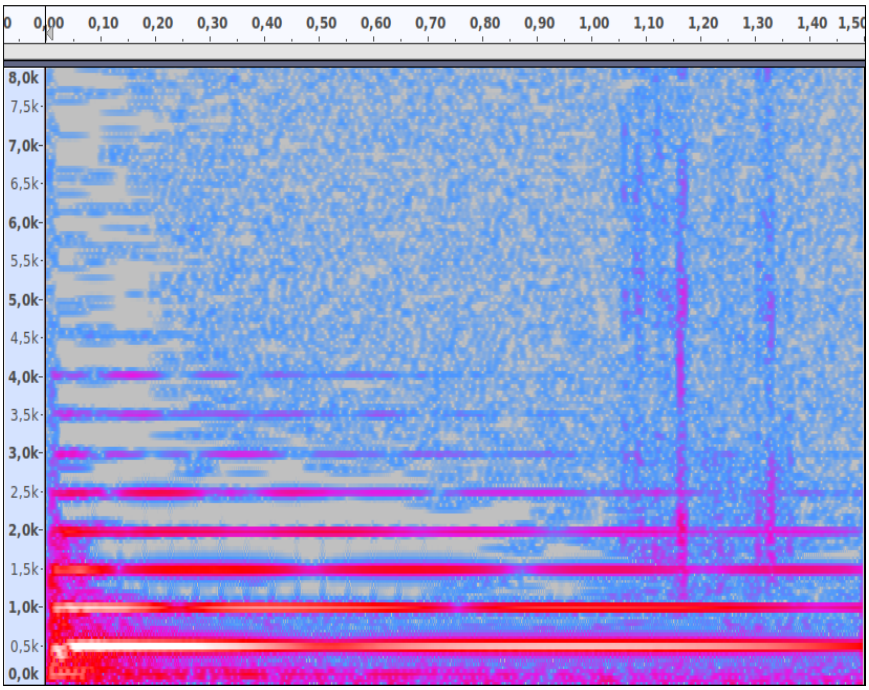

Figura 2.2: Espectrograma da nota musical B4 de piano.

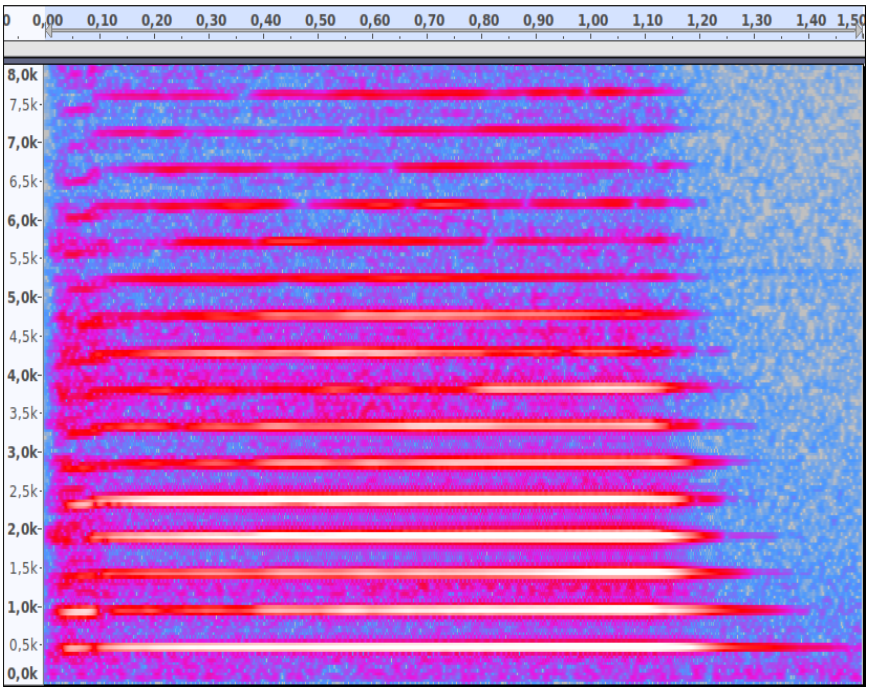

Figura 2.3: Espectrograma da nota musical B4 de violino.

musicais ajuda a caracterizar os seus respectivos timbres.

2.3 *Onsets* de notas musicais

O formato que a forma de onda de um tom musical pode assumir varia de acordo com o instrumento musical utilizado, porém, há um modelo simplificado que define as possíveis fases que o envelope do tom musical pode apresentar. Esse modelo, denominado de ADSR (do inglês, *attack*, *decay*, *sustain*, *release*), está ilustrado na Figura 2.4.

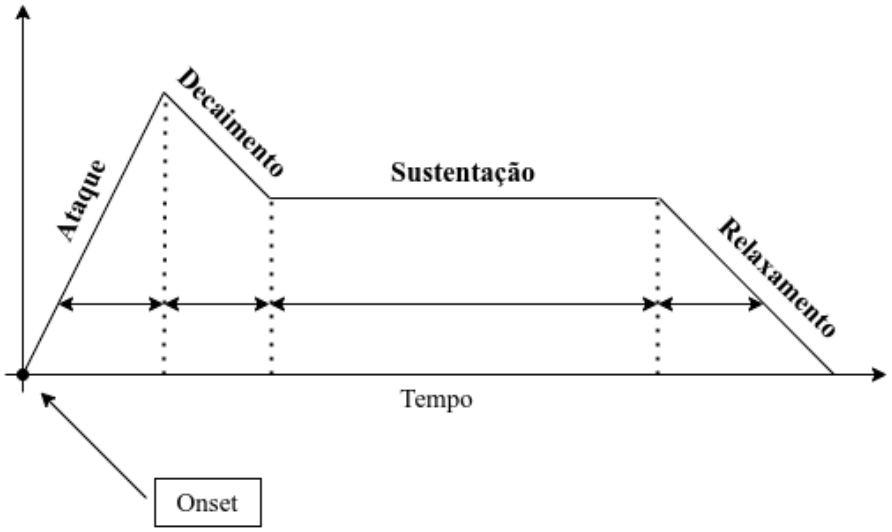

Figura 2.4: Envelope ADSR - (A) Fase de Ataque, (D) Fase de Decaimento, (S) Fase de Sustentação e (R) Fase de Relaxamento.

A primeira fase é a fase de ataque, caracterizada por um aumento abrupto de amplitude e, por consequência, um aumento de energia. Essa fase começa no momento em que a nota musical é tocada, gerando assim o som, e finaliza quando a amplitude máxima é alcançada. O intervalo de tempo correspondente a essa fase varia de acordo com o instrumento musical utilizado e, de acordo com [38], pode variar de cerca de 5 milissegundos a 200 milissegundos.

A segunda fase, a de decaimento, representa um decremento de amplitude que ocorre após a fase de ataque e precede a estabilização do som. Em seguida, acontece a fase de sustentação, na qual a energia do som permanece aproximadamente constante, ou seja, não ocorrem mudanças consideráveis em sua intensidade.

Por último, ocorre a fase de relaxamento. Nesta fase, o tocador do instrumento musical libera a tecla, no caso de um piano, ou para de causar a vibração da corda, no caso de um violão, por exemplo, interrompendo o fornecimento de energia

que mantém a nota soando e, por consequência, a energia do som decai até alcançar o estado de repouso.

É importante ressaltar que, por ser uma simplificação, na prática, as fases contidas no envelope ADSR geralmente não seguem à risca o que está no modelo. Na fase de sustentação, por exemplo, a amplitude não permanece estável durante todo o intervalo de tempo dessa fase e, dependendo do instrumento musical utilizado, alguma dessas fases pode até não existir.

Uma definição importante, dentro do contexto de detecção de *onsets*, é a definição de *onset*. O *onset* de uma nota musical é o instante de tempo que marca o início desse evento musical [38] [18]. A definição de *onset* também foi associada a transientes, como sendo a posição no tempo que marca o início do transiente [43], onde transiente é um intervalo curto do sinal caracterizado por apresentar um aumento súbito de energia e por evoluir de uma forma imprevisível.

Dessa forma, o *onset* é um único instante de tempo, não um intervalo de tempo e, no envelope ADSR da Figura 2.4, está especificado como o instante de tempo que marca o início da fase de ataque.

2.4 Espectrograma

Uma das formas de representar o som é através do espectrograma. Com o espectrograma é possível identificar características do som que não ficam visíveis na representação do sinal no domínio do tempo, por exemplo. O espectrograma é uma representação que mostra como as intensidades das frequências variam ao longo do tempo, logo, é uma representação tempo-frequência.

Exemplos de espectrogramas estão ilustrados nas Figuras 2.2 e 2.3. As frequências estão descritas no eixo y e o tempo, no eixo x. As linhas horizontais que aparecem nestes espectrogramas mostram as intensidades das componentes de frequência da nota musical tocada e, além das intensidades, podemos analisar o espalhamento das frequências ao longo do espectro.

Para calcular o espectrograma, janelas do sinal de áudio com comprimento curto são selecionadas e, sobre essas janelas, um conjunto de transformações são realizadas. Há algumas formas de calcular o espectrograma e, uma das mais co-

nhecidas, é utilizando a transformada de *Fourier* de tempo curto (STFT, do inglês, *short-time Fourier transform*). A STFT é calculada usando a equação (2.2).

$$\mathcal{X}(m,k) = \sum_{n=0}^{N-1} x(n+mH)w(n)^{-2\pi ikn/N} \tag{2.2}$$

O número complexo $\mathcal{X}(m,k)$ representa o k-ésimo coeficiente de *Fourier* para o m-ésimo *frame* no tempo, onde um *frame* denota uma seção do sinal janelada. Com relação aos componentes da equação (2.2), x representa um sinal discreto e w é uma janela discreta com tamanho N cujos coeficientes $w(n)$ estão dentro do intervalo $n \in [0 : N/2]$. H é o tamanho do passo e o índice k corresponde à frequência física F_{coef}, cuja equação está descrita em (2.3), na qual F_s denota a taxa de amostragem, medida em *Hertz*.

$$F_{coef}(k) = \frac{kF_s}{N} \tag{2.3}$$

Por fim, o espectrograma é definido como o quadrado da magnitude da STFT, de acordo com a equação (2.4).

$$\mathcal{Y}(m,k) := |\mathcal{X}(m,k)|^2 \tag{2.4}$$

Há outras formas de calcular o espectrograma, bem como variações do espectrograma calculado usando a STFT. A escala do espectrograma descrito anteriormente é linear, no entanto, essa escala pode ser alterada para a escala logarítmica Mel [44]. Adicionalmente, bancos de filtros *Gammatone* podem ser usados em conjunto com a STFT para calcular o espectrograma [45]. Outra forma de calcular o espectrograma é usando a transformada Q-Constante, que procura simular o sistema de percepção humano [46], dentre outras abordagens.

2.5 Representação *pitch*

No espectrograma, conseguimos analisar a energia em tempo curto contida nas várias frequências presentes no sinal. A representação *pitch* também mede a energia em tempo curto, porém, não mais diretamente nas frequências do sinal e sim em bandas *pitch* ou faixas de frequências que representam o(s) *pitch*(*es*) contidos

no sinal.

A Figura 2.5 ilustra a representação *pitch*, onde o eixo das abscissas mostra o tempo e o eixo das ordenadas mostra as oitenta e oito bandas *pitch* que variam dos MIDI *pitches* $p = 21$ (nota musical $A0$) a $p = 108$ (nota musical $C8$). A barra de cores localizada ao lado direito do gráfico mostra a escala de cores utilizada para representar a quantidade de energia contida nas bandas *pitch*. A energia pode variar de zero, representado pela cor preta a, aproximadamente $1,4$, representado pela cor branca, neste caso.

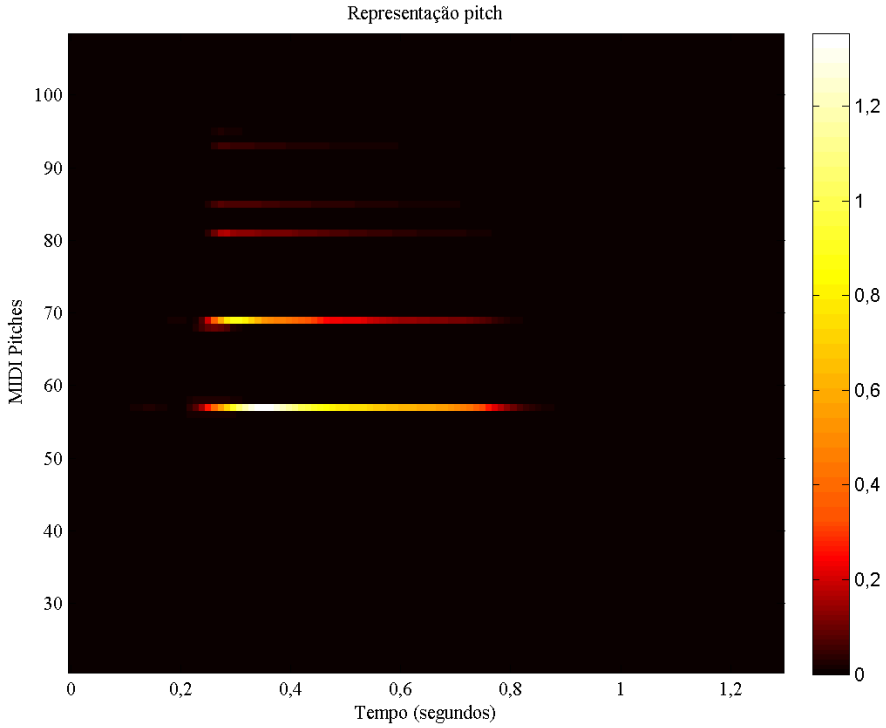

Figura 2.5: Representação *pitch* da nota de piano A3.

A representação *pitch* pode ser calculada de diversas formas e duas abordagens comumente utilizadas são a partir do espectrograma STFT, utilizando uma técnica de *pooling* [39], ou utilizando a técnica de banco de filtros para decompor o sinal [12].

No primeiro caso, após obter a STFT, são definidos intervalos baseados em frequências de corte definidas para cada um dos *pitches*. Todos os coeficientes $\mathcal{X}(m, k)$, cujas frequências físicas F_{coef} estiverem dentro de um desses intervalos, formam um conjunto $P(p)$ que irá referenciar um único *pitch* p. Em seguida, o

espectrograma na escala de frequências logarítmica é calculado usando a equação (2.5) e os coeficientes $\mathcal{Y}_{LF}$ resultantes são denominados de coeficientes *pitch*.

$$\mathcal{Y}_{LF}(m,p) := \sum_{k \in P(p)} |\mathcal{X}(m,k)|^2 \tag{2.5}$$

As equações (2.4) e (2.5) são similares, com a diferença de que, na segunda equação, é realizado o somatório do quadrado da magnitude de todos os coeficientes $\mathcal{X}(m,k)$ contidos em cada conjunto $P(p)$ referente a um dado *pitch* p.

Outra forma de calcular a representação *pitch* é utilizando banco de filtros. A primeira etapa dessa abordagem consiste em decompor o sinal de áudio em um conjunto de bandas de frequência *pitch* utilizando uma combinação de filtros passa-banda. Esses filtros mantêm faixas específicas de frequências do sinal enquanto rejeitam as outras frequências, isto é, um banco de filtros.

Na abordagem utilizada em [35], o sinal foi decomposto em oitenta e oito bandas de frequência utilizando um banco de filtros multitaxa, no qual diferentes taxas de amostragem foram empregadas. Os filtros utilizados foram filtros elípticos com três taxas de amostragem: 22050 Hz para *pitches* de alta frequência, 4410 Hz para *pitches* de média frequência e 882 Hz para *pitches* de baixa frequência.

Por último, para obter as cacterísticas *pitch*, a energia média quadrada em tempo curto (STMSP, do inglês, *short-time mean-square power*) foi calculada em cada banda de frequência *pitch* p. A equação utilizada para calcular as características *pitch* ou STMSP está descrita em (2.6).

$$\mathcal{C}_{stmsp}(m,k) := \sum_{k \in [m-\frac{w}{2}:m+\frac{w}{2}]} |b(k)|^2 \tag{2.6}$$

Onde m representa os *frames*, k é o índice de frequência, w é uma janela com tamanho fixo e b representa uma banda de frequência do sinal.

2.6 Aprendizado de máquina

O aprendizado de máquina é um subcampo da inteligência artifical caracterizado pelo auto-aprendizado, ou seja, um modelo é construído para tomar decisões sem a interferência humana. As decisões são tomadas com base em padrões aprendi-

dos no conjunto de dados fornecido como entrada para esse modelo, que vai refinando a tomada de decisões com a experiência obtida ao longo de processos de tentativa e erro. Logo, no aprendizado de máquina, um dos fatores importantes é o conjunto de dados utilizado, pois quanto maior for a qualidade dos dados aos quais o modelo for exposto, mais efetivo o modelo será.

O conjunto de dados $\mathbf{X}$ passado como entrada para o modelo f é composto por colunas de dados comumente denominadas de características ou atributos, enquanto as linhas de dados, geralmente, são chamadas de observações ou amostras. Adicionalmente, a variável de saída $\mathbf{Y}$ é referida como resposta ou variável dependente. A relação entre as variáveis de entrada $\mathbf{X}$ e saída $\mathbf{Y}$ do modelo f pode ser descrita de acordo com a equação (2.7).

$$\mathbf{Y} = f(\mathbf{X}) + e \tag{2.7}$$

Onde e representa o erro e o objetivo consiste em, a partir dos padrões aprendidos no conjunto de características de entrada $\mathbf{X}$ estimar uma função ou modelo f capaz de predizer o respectivo conjunto de saída $\mathbf{Y}$ de forma que o erro seja mínimo.

As técnicas de aprendizado de máquina podem ser divididas em quatro categorias principais: aprendizado supervisionado, aprendizado não-supervisionado, aprendizado semi-supervisionado e aprendizado por reforço.

Na categoria de aprendizado supervisionado, o modelo de aprendizado é construído analisando as relações existentes entre amostras de entrada cujas saídas são conhecidas para, dessa forma, aprender os padrões contidos nos dados. Essas combinações de entrada e saída conhecidas são referidas como amostras de dados rotuladas.

Na categoria de aprendizado não-supervisionado, os rótulos das amostras não são conhecidos e, portanto, não é possível analisar as relações existentes entre as amostras de entrada e suas respectivas saídas. Logo, a ênfase é dada no aprendizado das relações existentes entre as amostras de entrada com o objetivo de identificar padrões similares para separar o conjunto de dados em grupos com características similares e, dessa forma, classificar as amostras.

No aprendizado semi-supervisionado, geralmente, as amostras rotuladas são usadas para construir o modelo e o modelo construído é utilizado para rotular as amostras que não tem rótulos, ou seja, é uma junção dos aprendizados de máquina

supervisionado e não-supervisionado. O objetivo é usar todas as amostras disponíveis na base de dados, tanto rotuladas como não rotuladas, para tentar melhorar o desempenho do modelo.

Por último, o aprendizado por reforço é um tipo de aprendizado aplicado na construção de carros autônomos por exemplo, e é caracterizado por ser um processo contínuo de aprendizado. Várias combinações de entradas são fornecidas com o intuito de obter uma saída específica, isto é, executar uma ação, e a entrada mais adequada para executar uma dada ação é escolhida com base na pontuação que cada saída recebe.

2.6.1 Máquina de vetor de suporte

Máquina de vetor de suporte (SVM, do inglês, *support vector machine*) é uma técnica de classificação baseada no conceito de margens definidas a partir de um hiperplano de separação. Um hiperplano é uma fronteira de decisão utilizada para separar o conjunto de dados de entrada de forma que as amostras de dados sejam classificadas ou rotuladas de acordo com o lado da fronteira em que estiverem localizadas. Um exemplo de hiperplano de separação está ilustrado na Figura 2.6.

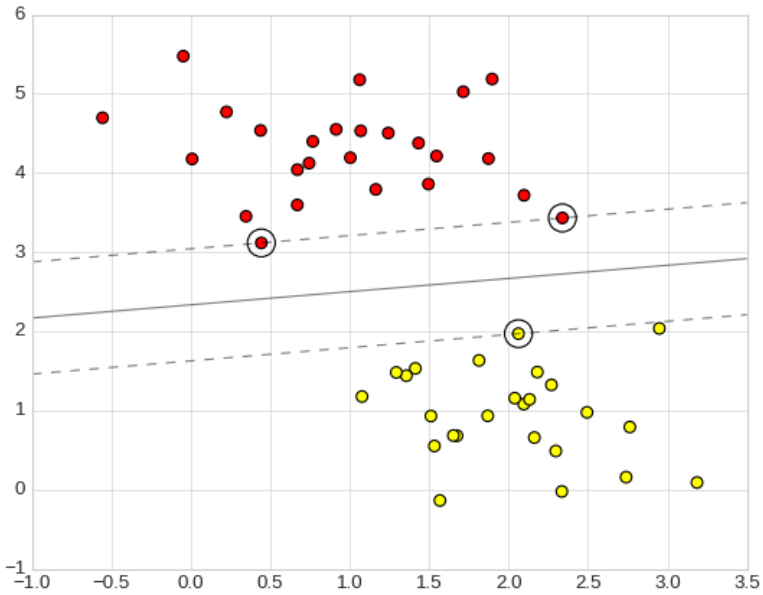

Figura 2.6: Exemplo de um conjunto de dados linear categorizado utilizando o algoritmo SVM.

O hiperplano de separação é composto por margens, uma de cada lado do

hiperplano, cuja largura é definida pela distância entre a fronteira de decisão e a amostra de dados mais próxima. A margem do hiperplano pode indicar o grau de confiança da classificação de uma dada amostra de dados e, em SVM, a princípio, o objetivo é que é essa margem seja máxima. A equação de um hiperplano é descrita na equação 2.8.

$$\mathbf{wx} - b = 0 \tag{2.8}$$

Onde $\mathbf{x}$ é o vetor de características de entrada, $\mathbf{w}$ representa um vetor com valores reais com a mesma dimensão de $\mathbf{x}$ e b é um número real. A expressão $\mathbf{wx}$ pode ser expandida de acordo com o número de dimensões D do vetor de características de entrada e possui o formato $w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + ... + w^{(D)}x^{(D)}$. O modelo da SVM é definido pela equação 2.9.

$$f(\mathbf{x}) = sign(\mathbf{wx} - b) \tag{2.9}$$

O algoritmo SVM, primeiramente, encontra valores ótimos para os parâmetros $\mathbf{w}$ e b e, em seguida, aplica a função $sign$ que retorna $+1$ caso a entrada seja positiva e -1 caso a entrada seja negativa, definindo assim a classe a qual a amostra de dados pertence.

Na prática, nem sempre é possível encontrar um hiperplano que separe todas as amostras de dados de forma que cada uma das amostras seja atribuída a classe correta. Logo, em alguns casos, é melhor assegurar um maior grau de confiabilidade na classificação de um subconjunto de amostras do que no conjunto todo.

Por isso, a SVM permite que uma certa quantidade de amostras viole a margem, o que é conhecido como margem suave. Quanto maior for a largura da margem, maior é a quantidade de amostras que podem violar a margem e, quanto mais estreita for a margem, menor é a quantidade de amostras que podem adentrar na região da margem. Esse grau de tolerância é definido alterando o valor do hiperparâmetro C, que é proporcional à largura da margem.

Outro tipo de situação recorrente é quando o conjunto de dados se comporta de forma não-linear e, neste caso, um hiperplano de separação linear, geralmente, não consegue realizar a tarefa de classificação das amostras de dados. Para resolver

esse problema, a SVM utiliza uma função chamada *kernel*.

O *kernel* é uma função que quantifica a similaridade entre duas amostras de dados [47] e, através da utilização dessa função, o espaço de características original pode ser mapeado em um espaço de características de maior dimensão. Quando isso é feito, o hiperplano não-linear do espaço de características original se torna linear no espaço de características de maior dimensão, permitindo assim, a classificação das amostras de dados.

Há vários tipos de *kernel* como, por exemplo, linear, polinomial, radial, dentre outros. A escolha do *kernel* a ser utilizado depende das características da aplicação. No sistema desenvolvido nessa dissertação, o *kernel* utilizado foi o linear, cuja equação está descrita em 2.10.

$$k(x, y) = x.y \tag{2.10}$$

Onde x e y representam vetores de características de entrada e a função *kernel* $k(x, y)$ é dada pelo produto interno entre esses dois vetores no espaço de características original. Nesse caso, o espaço de características original não é mapeado em um espaço de características de maior dimensão e o cálculo do produto interno é utilizado para medir a similaridade existente entre x e y.

Além do hiperparâmetro C e da definição da função *kernel*, outro hiperparâmetro que pode ser ajustado é o *gamma*. O *gamma* define o alcance da influência de uma dada amostra de dados. Quando o valor de *gamma* for alto, isso significa que as amostras de dados mais próximas da fronteira de decisão têm maior peso e isso afeta nas sinuosidades que a fronteira irá apresentar. Por outro lado, se o valor de *gamma* for baixo, as amostras de dados situadas mais longe da fronteira de decisão terão mais peso e, portanto, a fronteira de decisão será mais suave.

2.6.2 *Gradient boosting*

O *gradient boosting* é uma técnica de aprendizado de máquina caracterizada por utilizar um algoritmo de otimização similar ao gradiente descendente para minimizar os erros produzidos por modelos ou aprendizes fracos criados ao longo do processo de aprendizado.

A característica principal dos métodos de *boosting* é a utilização de vários aprendizes fracos de forma sequencial para armazenar os padrões aprendidos [48] e auxiliar nos ajustes necessários para reduzir o erro do sistema. A ideia é que um conjunto de aprendizes fracos formem um sistema que forneça resultados melhores se comparado ao uso de um único modelo.

Além disso, os pesos das amostras de dados são alterados à medida que a classificação é realizada, isto é, os pesos das amostras classificadas incorretamente são incrementados enquanto os pesos das amostras classificadas corretamente são decrementados. Isso é feito para que, na iteração seguinte, a probabilidade de seleção das amostras classificadas de forma errada aumente e os erros sejam tratados ao longo do processo.

A primeira etapa do algoritmo *gradient boosting* consiste em inicializar o modelo com um valor constante e isso é feito construindo um modelo, com o conjunto de dados de treinamento, cujo valor da predição alvo garante que a função de perda seja mínima. Em seguida, os resíduos são computados calculando a diferença entre o valor referência e o valor predito.

Após isso, um novo modelo é criado tendo como base os resíduos, criando assim, uma árvore de decisão. A próxima etapa consiste em calcular os valores de saída de cada nó folha da árvore e, por último, as predições são atualizadas utilizando as predições anteriores, a taxa de aprendizado e a árvore de decisão feita a partir dos resíduos.

A taxa de aprendizado determina o tamanho do passo de aprendizado e, no caso do *gradient boosting*, reduz a contribuição que cada árvore de decisão ou modelo criado tem no resultado final. Além disso, o número de nós que cada árvore pode assumir é um fator a ser levado em consideração durante a construção do classificador.

2.6.3 Rede neural convolucional de uma dimensão

A rede neural convolucional (CNN, do inglês, *convolutional neural network*) é um algoritmo de aprendizado profundo cujo elemento principal é a camada de convolução que, como o próprio nome indica, realiza a operação de convolução em um conjunto de dados.

A convolução é uma operação linear capaz de identificar padrões nos dados e seus componentes principais são o sinal de entrada x e o *kernel* w, como descrito na equação 2.11. As dimensões de x e de w dependem do tipo dos dados fornecidos como entrada à rede que podem ser, por exemplo, séries temporais, imagens, dentre outros.

$$y_j = \sum_{k=-p}^{p} x_{j-k} w_k \tag{2.11}$$

Na prática, o sinal de entrada é um vetor multidimensional de dados enquanto o *kernel* é um vetor multidimensional de parâmetros e esses vetores, geralmente, são denominados de tensores. No sistema proposto, os dados de entrada são unidimensionais e, portanto, a CNN utilizada é a 1D.

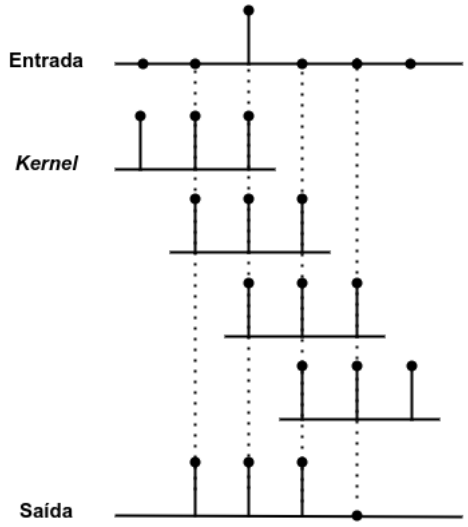

Figura 2.7: Operação de convolução.

O processo de convolução, descrito na equação 2.11 e ilustrado na Figura 2.7, consiste em, primeiramente, refletir o *kernel* e, em seguida, passar o *kernel* w por cada elemento do sinal de entrada x. Geralmente, o tamanho do *kernel* é ímpar, restrição essa contida na equação 2.11, e seu elemento central é alinhado com cada elemento do sinal de entrada.

Em cada posição de w com relação a x, os elementos do *kernel* que estiverem alinhados com elementos do sinal de entrada são multiplicados e, depois, os resultados das multiplicações são somados, ou seja, o produto interno é calculado. Logo, para cada posição de alinhamento, um valor é calculado e o conjunto de valores

formado ao final de todo o processo é o resultado da convolução, também conhecido como mapa de características.

A CNN possui três características principais, utiliza campos receptivos locais, compartilha parâmetros e faz uso da operação de *pooling* [49]. O campo receptivo local corresponde a região do conjunto de dados de entrada que é alinhada ou sobreposta pelo *kernel* em cada etapa do processo de convolução. Geralmente, os dados de entrada são conectados a uma camada de neurônios escondidos. A vantagem de usar o campo receptivo local é que invés de conectar cada dado de entrada a um neurônio escondido, o conjunto de dados correspondente ao campo receptivo é concectado, o que reduz o número de conexões necessárias.

A segunda característica da CNN é que o mesmo conjunto de pesos, definido pelo *kernel*, é utilizado ao longo de cada elemento do conjunto de dados de entrada. Dessa forma, há compartilhamento de parâmetros e essa característica, além de reduzir os requisitos de armazenamento da rede, faz com que a rede adquira uma propriedade chamada equivariância à translação [50].

A equivariância à translação significa que, se as amostras na entrada da rede forem transladadas, a mesma translação ocorrerá na saída. Sendo assim, a posição do padrão de características na entrada não precisa ser fixa para que o padrão seja detectado.

Por último, a rede neural convolucional realiza a operação de *pooling*, que consiste em calcular uma estatística resumida dos elementos de saída da rede. Essa estatística pode ser o valor máximo do conjunto, também chamado de *max-pooling*, o valor médio, denominado de *average-pooling*, dentre outras opções. Uma invariância aproximada à suaves deslocamentos aplicados na entrada pode ser alcançada usando a operação de *pooling*.

Neste capítulo foram abordados conceitos de teoria musical relevantes para o melhor entendimento do sistema proposto, como melodia, harmonia, ritmo, oitava e escala musical. Adicionalmente, algumas características de sinais de música foram destacadas bem como os atributos dos sinais de música *pitch*, *loudness* e timbre foram descritos.

Além disso, o conceito de *onset* foi definido e a representação tempo-frequência *pitch* foi descrita juntamente com características principais dos classi-

ficadores SVM, *gradient boosting* e CNN 1D. No próximo capítulo, como resultado de uma revisão bibliográfica realizada, os principais trabalhos relacionados ao sistema de detecção automática de *onsets* proposto serão abordados.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, informações relevantes de trabalhos relacionados ao sistema de detecção de *onsets* proposto, encontrados ao longo do processo de revisão bibliográfica, serão apresentadas, assim como uma tabela de comparação com as principais características dos trabalhos relacionados.

O artigo [27] investiga o uso de CNNs no contexto de transcrição automática de sons de bateria. Duas abordagens são propostas para fazer a transcrição. Na primeira, a CNN realiza o papel de detector de *onsets* dos sons dos instrumentos de bateria chimbal, bumbo e tarola.

Os tempos dos *onsets* obtidos são fornecidos como entrada para o algoritmo de transcrição de sons de bateria, denominado de deconvolução de matriz não-negativa (NMD, do inglês *non-negative matrix deconvolution).* Esse algoritmo analisa os padrões situados ao redor dos tempos dos *onsets* gerados, usando um dicionário de padrões e uma matriz de ativações, e identifica a qual instrumento de bateria cada um dos tempos pertence.

Na segunda abordagem proposta, somente a CNN é usada para fazer a transcrição dos instrumentos de bateria. Como são três instrumentos, chimbal, bumbo e tarola, três redes são utilizadas, uma para cada instrumento. Como saída, os eventos referentes ao instrumento principal da rede são detectados e identificados.

Nas duas abordagens, foram avaliados os desempenhos do detector de *onsets* e da tarefa de transcrição. Na primeira abordagem, o melhor desempenho obtido pelo método de detecção de *onsets* atingiu um *F-measure* de 93,50%, enquanto o desempenho obtido na transcrição não foi competitivo com os métodos *baseline*.

Na segunda abordagem, o melhor valor de *F-measure* obtido na detecção de *onsets* foi de 93,40%, enquanto na tarefa de transcrição dos sons dos instrumentos de bateria, o desempenho obtido na detecção dos eventos de som do bumbo foi suavemente melhor e, na detecção do instrumento tarola, foi significativamente melhor. Além disso, os *F-measures* de cada um dos três instrumentos ultrapassou os *F-measures* dos mesmos instrumentos nos métodos *baseline*.

Em [28], um modelo que usa informações da fase de transição de ataque e de transição completa de notas de sinais de piano é proposto para realizar a transcrição automática. O modelo proposto é composto pelos ramos de transição de ataque, de transição completa e pela etapa de seleção de picos.

O ramo de transição de ataque é responsável por fazer a modelagem da transição de ataque da nota, anotando vários *frames* situados ao redor do tempo do *onset* detectado, enquanto o ramo de transição completa realiza a modelagem do processo de transição completa da nota. O segundo ramo anota todos os *frames* que se estendem desde o tempo do *onset* até o tempo do *offset* e, com esses dados, faz a predição de ativação das notas.

Cada um dos ramos usam, na sequência, uma rede neural convolucional (CNN), seguida por uma rede BiLSTM (do inglês, *bidirectional long short-term memory*) e, por último, por camadas totalmente conectadas. Após a aplicação da transformada Q constante (CQT, do inglês *constant Q transform*) no sinal de entrada, a CNN extrai as características do espectrograma CQT. Em seguida, a BiLSTM faz a modelagem dependente do tempo e, por último, as camadas totalmente conectadas são usadas como classificador para completar a detecção dos tempos dos *onsets* e da estimação de *frames*.

As probabilidades geradas no ramo de transição de ataque são fundidas no ramo de transição completa para melhorar a estabilidade do modelo. Além disso, o ramo de transição de ataque possui, após as camadas totalmente conectadas, uma etapa de seleção de picos e fornece como saída os tempos dos *onsets* e os *pitches*, completando assim a tarefa de transcrição.

O modelo *transition-aware* proposto obteve os melhores desempenhos nos conjuntos de teste das bases de dados MAPS e OMAPS, com *F-measures* de 87,52% e 88,21%, respectivamente, ultrapassando o desempenho do modelo *baseline high-*

resolution por 4,35% e 10,31% nas mesmas bases. No conjunto de teste das bases MAESTRO-v1 e MAESTRO-v2, as *F-measures* obtidas foram 1,93% e 1,30% abaixo das obtidas pelo modelo *high-resolution*. Além disso, o modelo *transition-aware* ultrapassou as *F-measures* do segundo modelo *baseline onsets and frames* em todas as quatro bases.

A detecção automática de *onsets* usando redes de estado de eco (ESNs, do inglês *echo state networks*) foi proposta em [51]. A ESN é um tipo de rede neural recorrente (RNN, do inglês *recurrent neural networs*) cujo elemento principal é o reservatório, formado por um conjunto não-ordenado de neurônios, que pode reter informações de entradas anteriores por um período de tempo limitado.

Na abordagem proposta, o sinal de entrada mono é dividido em segmentos de áudio sobrepostos, denominados de *frames*, usando uma janela *Hann*. Em seguida, a extração de características é realizada aplicando, sobre cada *frame*, a transformada de *Fourier* de curto tempo.

Após isso, um banco de filtros triangular com frequência logarítmica é aplicado e, posteriormente, uma versão adaptada do algoritmo *spectral flux* é executada. O vetor de características resultante alimenta a ESN, que processa os dados e fornece como saída a função de detecção de *onset*. Por último, um algoritmo de seleção de picos baseado em limiar é aplicado sobre a ODF.

Adicionalmente, os autores introduziram uma nova forma de empilhar reservatórios, na qual a ODF resultante do primeiro reservatório é fornecida como *bias* para o segundo reservatório, que recebe como entrada o vetor de características original. A saída do sistema é binária, *onset* ou não-*onset* e os tempos dos *onsets* detectados são comparados com os tempos dos *onsets* referência.

O desempenho do sistema proposto foi comparado com dois algoritmos *baseline*, denominados de rede bidirecional LSTM e CNN. Os valores de precisão e *F-measure* obtidos na abordagem proposta foram de $92,00\%$ e $88,60\%$, respectivamente. Ultrapassando o desempenho do método *baseline* rede bidirecional LSTM e ficando um pouco abaixo do desempenho do método *baseline* CNN, cujo valor de *F-measure* obtido foi $90,30\%$.

Uma arquitetura de rede neural unificada que prediz múltiplos estados de notas é proposta em [31] para fazer a transcrição automática de sinais polifônicos

de piano, ou seja, estimar as notas a partir das características acústicas. Dentre os estados de notas investigados estão *onset*, *off*, sustentação, *re-onset* e *offset*, dos quais cinco representações são definidas através da combinação desses estados.

A arquitetura apresentada integra um modelo acústico com um modelo de linguagem musical (MLM, do inglês *musical language model*). O modelo acústico utiliza a CNN e tem o papel de extrair as características acústicas locais e armazená-las em um vetor.

O modelo MLM utiliza a RNN para predizer os estados das notas usando uma conexão autoregressiva. A conexão autoregressiva aprende a dependência existente entre os estados internos e, adicionalmente, é possível aumentar o número de estados sem ter que aumentar a complexidade da arquitetura.

Após a estimação dos estados das notas, ocorre a inferência desses estados por um algoritmo de decodificação e, nesta etapa, dois algoritmos foram testados, um algoritmo guloso e um algoritmo que realiza pesquisa por feixe. Por último, uma regra simples é aplicada para determinar as notas.

O sistema proposto foi avaliado na base de dados MAESTRO v1.0.0 que contém áudios polifônicos de piano e, além disso, um limiar de 50 milissegundos foi usado para detectar os tempos dos *onsets*. Dentre os modelos propostos, o que obteve melhor desempenho na detecção de *onsets* foi o *five*, que alcançou um *F-measure* de $94,67\%$, ficando um pouco atrás do desempenho obtido pelo método *baseline non-saturating GAN*, que obteve um *F-measure* de $95,60\%$.

Um transformador codificador-decodificador com decodificação padrão é empregado em [32] para fazer a transcrição automática de sons polifônicos de piano. Na abordagem proposta, o áudio de entrada é dividido em segmentos, sobre os quais o espectrograma mel na escala logarítmica é calculado. O *frame* do espectrograma juntamente com a descrição simbólica correspondente é fornecido como entrada para a rede na etapa de treinamento.

A descrição simbólica se refere a eventos codificados semelhantes a mensagens do protocolo MIDI (do inglês *musical instrument digital interface*). Em seguida, o codificador, que tem o papel de extrair as características de áudio musicalmente significativas, processa as entradas usando uma pilha de camadas de autoatenção e fornece, como resultado, uma sequência de associações. Essa sequência então é

passada para a pilha de camadas do decodificador autoregressivo que aplica dois tipos de autoatenção, uma sobre a saída do codificador e outra sobre a saída do decodificador.

A decodificação da saída do modelo é feita por um algoritmo guloso autoregressivo e a saída do modelo é uma distribuição *softmax* com um vocabulário de eventos discretos similares ao MIDI, composto pelos eventos *onset*, *offset*, *pitch* e velocidade.

A arquitetura do transformador é treinada do zero e com transcrições rotuladas, ou seja, de forma supervisionada comum. A avaliação do modelo proposto foi realizada na base de dados MAESTRO e o valor de *F-measure* obtido para a detecção de *onsets* foi de $95,95\%$, ficando $0,77\%$ abaixo do modelo *baseline* identificado como modelo de *Kong*, que obteve um *F-measure* de $96,72\%$.

No artigo [33], três métodos de detecção automática de *onsets* são avaliados usando uma base de dados de sons de clarinete. O primeiro método, proposto por *Bock* [52], é baseado em CNN e tem como entrada uma pilha de espectrogramas mel na escala logarítmica, calculados com três tamanhos de janelas diferentes.

O segundo método, proposto por *Eyben* [53], é baseado em RNN e vetores de características empilhados alimentam a rede. O conjunto de características de entrada é formado por três trechos de espectrogramas, calculados usando três tamanhos de janelas diferentes. E pelo resultado da diferença positiva de primeira ordem entre dois *frames* de espectrogramas sucessivos, calculada usando dois tamanhos de janelas diferentes. O método de processamento digital de sinal, denominado de *superflux*, é o terceiro método investigado.

O *superflux* é uma versão melhorada de um método baseado no cálculo de fluxo espectral, que usa informações de fase para calcular o atraso de grupo local e, adicionalmente, contém um estágio de rastreamento de notas para evitar valores altos na função de detecção de *onsets*.

Esses três métodos, pré-treinados com uma base de dados contendo sons de vários instrumentos musicais, foram avaliados usando a base de dados proposta, a *clari-onsets-50*, composta exclusivamente por sons de clarinete. Além disso, o método baseado em CNN foi reimplementado, treinado e testado com a base *clari-onsets-50*, com o objetivo de verificar o impacto da presença das características

específicas dos sons de clarinete nos dados de treinamento do detector de *onsets*.

O desempenho obtido pela *CNN-Clari* foi superior a todos os três métodos, obtendo uma *F-measure* de $95,40\%$, enquanto o segundo melhor desempenho obtido foi do método *CNN-Bock*, com *F-measure* de $86,10\%$. O método *superflux* obteve o pior desempenho.

Podemos destacar também os sistemas de detecção automática de *onsets* descritos a seguir. O sistema proposto por [54] utiliza características de classes de *pitch* e de energia para detectar *onsets* de notas de vários instrumentos musicais. Os elementos principais do sistema são dois módulos que atuam paralelamente. O primeiro módulo estima as características das notas e categoriza os *onsets*, enquanto o segundo módulo realiza processamentos baseados em informações de energia e informações de classes de *pitch*, calculando o cromagrama.

A proposta de [52] consiste em utilizar a CNN 2D para processar trechos de uma pilha de espectrogramas, dispostos de forma similar as imagens RGB. A CNN fornece como saída a função de ativação de *onsets* sobre a qual, posteriormente, é aplicada uma operação de convolução, usando uma janela *Hann* para suavizá-la. Os valores de pico situados acima de um dado valor de limiar são identificados como *onsets*.

O artigo [55] aprensenta uma abordagem de aprendizado de máquina estatístico para detectar *onsets* de notas. O vetor de características extraído de um espectrograma, calculado utilizando a transformada de *Gabor*, alimenta uma cascata de classificadores *AdaBoost* discretos, no qual cada classificador é dedicado ao processamento de uma nota musical. A saída do sistema é binária, positivo para o caso de o *onset* estar presente ao longo do vetor de características selecionado, e negativo para o caso oposto.

Em [26] é apresentada uma abordagem na qual características de intensidade e energia são fornecidas como atributos de áudio de entrada para avaliar três algoritmos de aprendizado de máquina, a SVM, redes neurais e *Naïve Bayes* com *Adaboosting*.

Os arquivos de áudio são sintetizados a partir de arquivos MIDI de peças musicais e as características são obtidas aplicando a transformada rápida de *Fourier* (FFT, do inglês *fast fourier transform*). O resultado do sistema é comparado com

os tempos dos *onsets* de referência dos arquivos MIDI e os tempos situados dentro da janela de tolerância de 50 milissegundos são identificados como *onsets*.

Por fim, [56] propõe um algoritmo de aprendizado semi-supervisionado no qual um alinhamento de áudio com *score* é realizado para localizar os *onsets*. Para fazer o alinhamento, uma matriz de distância entre o vetor de características *chroma* do áudio e o vetor MIDI correspondente é calculada.

Em seguida, um conjunto diverso de características de áudio baseadas em informações de frequência, energia, fase, dentre outras, são extraídas. Por último, o conjunto de dados de treinamento rotulado, resultante do alinhamento, é passado como entrada para o classificador SVM com *kernels* de função de base radial (RBF, do inglês *radial basis function*), que detecta os *onsets* de forma mais precisa.

A Tabela 3.1 contém características relevantes de seis trabalhos principais relacionados ao sistema de detecção automática de *onsets* proposto. Dentre eles estão o problema abordado, o tipo de sinal de música investigado, a estrutura de dados de entrada do sistema, as principais técnicas empregadas, o tamanho da janela ou margem de tolerância utilizada, as métricas utilizadas e o resultado.

No geral, os problemas abordados nos artigos foram detecção automática de *onsets* ou transcrição automática, um problema mais amplo no qual a detecção de *onsets* está inclusa. Os sinais de música analisados em cada trabalho foram sinais monofônicos e/ou polifônicos de bateria, piano, clarinete e um conjunto formado por sons de instrumentos musicais variados.

Com relação a estrutura de dados de entrada, foram investigadas estruturas de dados unidimensional (vetores de características), bidimensional (um trecho de espectrograma) e tridimensional (dois e três trechos de espectrogramas empilhados). Além disso, a representação do sinal de música mais utilizada foi o espectrograma.

A maioria das técnicas utilizadas para classificação foram técnicas de aprendizado profundo onde, dos seis trabalhos, quatro investigaram a CNN. Em [27], somente a CNN foi utilizada para classificação enquanto nos trabalhos [28], [31] e [33], a CNN foi utilizada em conjunto com outras técnicas.

No primeiro caso, a CNN foi usada com a BiLSTM e a técnica utilizada para calcular o espectrograma foi a CQT. No segundo caso, a CNN foi utilizada em conjunto com o MLM e RNN e, no terceiro caso, a CNN foi usada com a BiLSTM e

com a técnica *SuperFlux*. Adicionalmente, em [27], a transcrição automática também foi feita utilizando a técnica NMD para fins de comparação e, além disso, fitros logarítmicos foram utilizados em [51] e [33].

Com relação às bases de dados, a base MAESTRO foi explorada em três trabalhos do seis e as métricas utilizadas para avaliar os sistemas, na maioria dos trabalhos, foram precisão, *recall* ou sensibilidade e *F-measure*. Por último, há o resultado, correspondente aos valores de *F-measure* obtidos ao avaliar cada sistema.

Analisando a Tabela 3.1, observamos quatro fatores principais. O primeiro é que, dos seis trabalhos relacionados, cinco utilizam o espectrograma como representação do sinal de música, onde um ou mais trechos de espectrograma formam a estrutura de dados fornecida como entrada para os sistemas propostos, o que abre espaço para explorar outro tipo de representação do sinal de áudio.

O segundo fator observado é que a maioria das técnicas utilizadas para extrair e/ou classificar as observações são de aprendizado profundo, com estruturas de dados de entrada 2D ou 3D, apenas um trabalho possui uma estrutura de dados de entrada 1D.

O terceiro fator é que o tamanho da janela de tolerância utilizada em todos os trabalhos é fixo e, por último, sinais de música de instrumentos musicais variados são explorados nos trabalhos, incluindo sinais de instrumentos percussivos, de instrumento de sopro e até de sinais de música compostos por instrumentos musicais juntamente com voz. Esses foram os fatores considerados na definição dos principais objetivos desta dissertação.

Neste capítulo, as principais características de trabalhos relacionados ao sistema de detecção automática de *onsets* proposto foram descritas e uma tabela contendo informações relevantes de seis trabalhos principais foi fornecida.

Nessa tabela, as abordagens propostas puderam ser comparadas analisando o tipo de problema, o tipo de sinal de música utilizado, a estrutura de entrada, as técnicas e métricas usadas, o tamanho da janela de tolerância e os valores de *F-measure* obtidos. Adicionalmente, uma análise desta tabela foi realizada. No próximo capítulo, a metodologia empregada no desenvolvimento do sistema proposto será detalhada.

Autor principal	**Problema**	**Sinal de música**	**Entrada**	**Técnicas**	**Janela (ms)**	**Base de dados**	**Métricas utilizadas**	**Resultado**
(C. Jacques, 2018) [27]	Transcrição automática	Sinais polifônicos de três instrumentos de bateria	Uma trecho de espectrograma e três trechos de espectrogramas empilhados	CNN e NMD	30	*ENST drum* e RWC	Precisão, *Recall* e *F-measure*	$93,50\%$
(X. Wang, 2021) [28]	Transcrição automática	Sinais polifônicos de piano	Dois trechos de espectrogramas	CQT, CNN e BiLSTM	50	MAPS, MAESTRO e OMAPS	Precisão, *Recall* e *F-measure*	$88,21\%$
(P. Steiner, 2021) [51]	Detecção de *onsets*	Sinais monfônicos e polifônicos de vários instrumentos musicais	Vetor de características	STFT, filtro triangular com frequência logarítmica e *echo state networks* (ESNs)	25	*Bock*	Precisão, *Recall* e *F-measure*	$88,60\%$
(T. Kwon, 2020) [31]	Transcrição automática	Sinais polifônicos de piano	Uma trecho de espectrograma	*Autoregressive musical language model* (MLM), CNN e RNN	50	MAESTRO	Precisão, *Recall* e *F-measure*	$94,67\%$
(C. Hawthorne, 2021) [32]	Transcrição automática	Sinais polifônicos de piano	Um trecho de espectrograma	FFT e *encoder-decorder transformer*	50	MAESTRO	*F-measure*	$95,95\%$
(T. N. Magalhães, 2021) [33]	Detecção de *onsets*	Sinais monofônicos de clarinete	Três trechos de espectrogramas empilhadas	Bancos de filtros logarítmicos, CNN, BiLSTM e *SuperFlux*	20 e 50	Criada pelos autores	Precisão, *Recall* e *F-measure*	$95,40\%$

Tabela 3.1: Comparação das principais características dos trabalhos relacionados ao sistema proposto.

Capítulo 4

Metodologia para Detecção de *Onsets* em Notas de Piano

Neste capítulo, a metodologia utilizada para desenvolver o sistema de detecção de *onsets* proposto será descrita. No geral, a metodologia foi dividida em três etapas principais: construção de modelo para classificação, classificação dos vetores de características *pitch* e pós-processamento.

Na primeira abordagem, empregada com os classificadores máquina de vetor de suporte (SVM, do inglês, *support vector machine*) e *gradient boosting*, as três etapas foram implementadas. Na segunda abordagem, empregada com a rede neural convolucional de uma dimensão (CNN 1D, do inglês, *one dimensional convolutional neural network*), as duas primeiras etapas foram implementadas.

4.1 Construção de modelo para classificação

A primeira etapa do sistema proposto, a construção de modelo para classificação, tem por objetivo gerar um modelo para classificar um conjunto de características *pitch*. O diagrama da Figura 4.1 ilustra as subetapas da construção de modelo para classificação.

Primeiramente, cada áudio de entrada é convertido para a representação *pitch*, sobre a qual as rotulações das coordenadas de referência são realizadas na etapa seguinte. Após a rotulação, as características *pitch* são extraídas, tendo como referência as coordenadas rotuladas anteriormente, formando o conjunto de dados

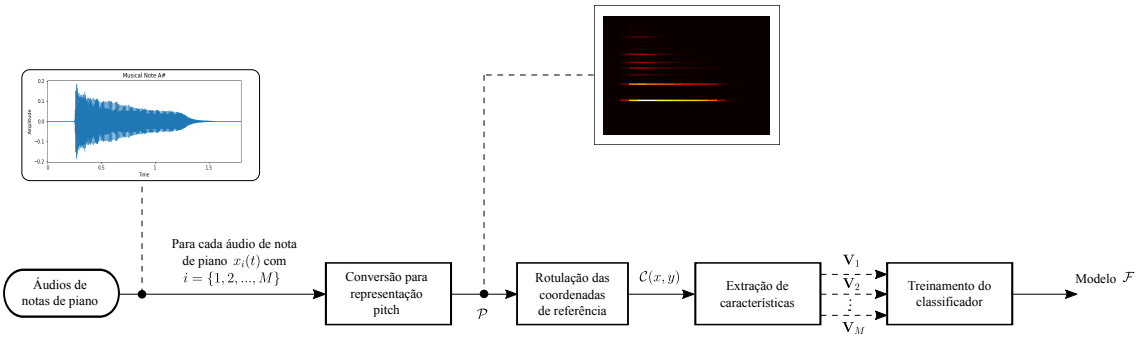

Figura 4.1: Diagrama da etapa de construção de modelo para classificação do sistema de detecção de *onsets*.

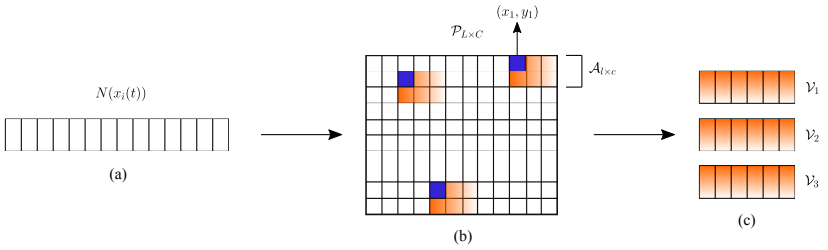

Figura 4.2: Processo de conversão do vetor com amostras de áudio $N(x_i(t))$ (a) para matriz *pitch* $\mathcal{P}_{L\times C}$ (b) e transformação das submatrizes *pitch* $\mathcal{A}_{l\times c}$ para vetores *pitch* $\mathcal{V}_i$(c).

com o qual o classificador é treinado. Após o treinamento, o modelo para classificação é gerado. A seguir, cada uma das subetapas da construção de modelo para classificação será descrita.

A primeira subetapa consiste em converter o áudio de entrada $x_i(t)$ para a representação tempo-frequência *pitch*, denominada por $\mathcal{P}$. Nesta etapa, o vetor com as amostras de áudio de entrada $N(x_i(t))$ é transformado em uma matriz $\mathcal{P}$, a representação *pitch*, utilizando um conjunto de filtros.

A representação *pitch* possui dimensões $L \times C$ e é composta por coeficientes denominados de *short-time mean square power* (STMSP). O processo de conversão é ilustrado na Figura 4.2. A dimensão L da matriz é fixa e representa as oitenta e oito bandas de frequência contidas na representação *pitch*, enquanto a dimensão C varia de acordo com a duração do sinal de áudio de entrada $x_i(t)$.

Um diferencial da representação *pitch*, e um dos motivos da escolha desta representação, é que a delimitação de cada uma das bandas de frequência é feita com base na frequência central das notas musicais contidas na escala igualmente-

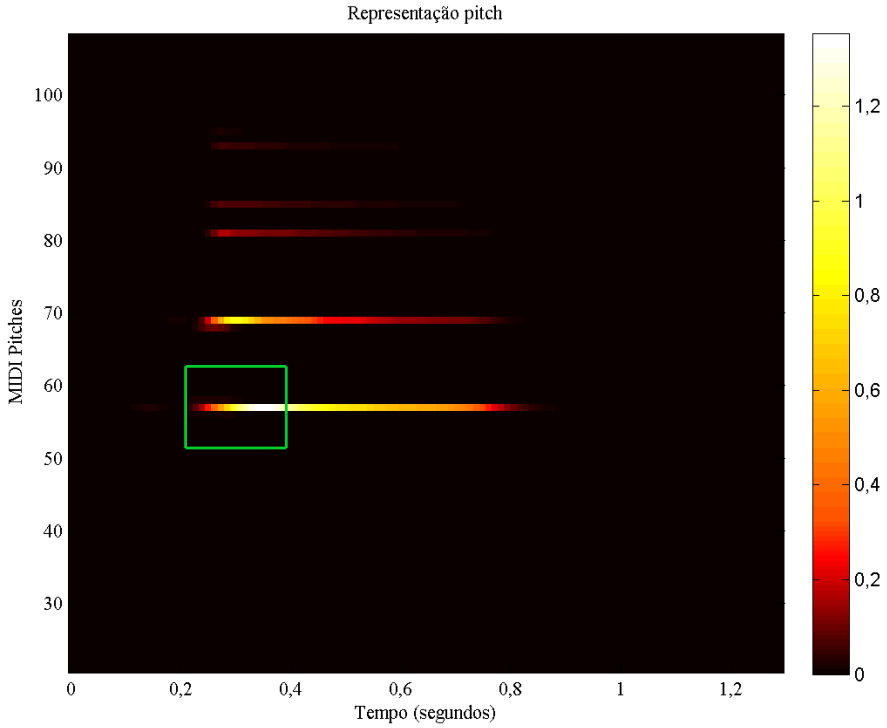

Figura 4.3: Representação *pitch* da nota de piano A3.

temperada de doze tons[1] e, dessa forma, cada banda representa uma nota musical. Logo, um acúmulo maior de energia em uma determinada banda *pitch* pode indicar a presença da nota musical representada por essa banda.

Considerando outro tipo de representação, como o espectrograma por exemplo, a relação entre frequências e notas musicais não seria explícita. Além disso, ainda fazendo comparação entre representação *pitch* e espectrograma, a representação *pitch* pode ser vista como uma representação mais compacta, pois a quantidade de coeficientes no eixo das frequências é menor.

A segunda subetapa da construção de modelo para classificação é a rotulação das coordenadas de referência, na qual coordenadas de coeficientes STMSP contidos na representação *pitch* $\mathcal{P}$ são selecionadas. No exemplo apresentado na Figura 4.2 (b), as coordenadas de referência são as coordenadas dos elementos destacados em azul na representação *pitch* $\mathcal{P}_{L\times C}$ e um exemplo de gráfico utilizado durante a rotulação manual das coordenadas está ilustrado na Figura 4.3.

Dois tipos de rotulações são realizadas na representação *pitch* $\mathcal{P}$, as rotulações

[1] Escala igualmente-temperada de doze tons é um conjunto de sons no qual a oitava é subdividida em doze passos igualmente distribuídos.

das coordenadas *onset* $\mathcal{O}$ e as rotulações das coordenadas não-*onset* $\mathcal{NO}$. Em ambos os casos, uma ou mais coordenadas são selecionadas na representação *pitch* $\mathcal{P}$. A diferença é que as submatrizes $\mathcal{A}_{l \times c}$ que serão definidas a partir de cada coordenada, na subetapa seguinte, podem conter valores de coeficientes que indicam a presença de *onsets* ou a ausência de *onsets*. Por esse motivo, as coordenadas são rotuladas como *onsets* ou não-*onsets*.

O processo de rotulação das coordenadas de referência é apresentado na Figura 4.4. Para cada representação *pitch* $\mathcal{P}$, as coordenadas de referência *onset* $\mathcal{O}$ e as coordenadas de referência não-*onset* $\mathcal{NO}$ são selecionadas, resultando nos conjuntos de rotulações $\{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_M\}$ e $\{\mathcal{NO}_1, \mathcal{NO}_2, ..., \mathcal{NO}_M\}$, respectivamente. A junção destes dois conjuntos forma o conjunto de coordenadas final $\mathcal{C}(x, y)$, que corresponde à saída deste bloco.

O número de rotulações é um parâmetro definido previamente e está dentro do intervalo $[1, 2, ..., M]$. Caso o número de rotulações escolhido seja dois, por exemplo, isso significa que, para cada tipo de rotulação, duas coordenadas de referência serão selecionadas e $\mathcal{C}(x, y)$ conterá quatro coordenadas no total.

A seleção manual das coordenadas é feita com base nas cores contidas no gráfico da representação *pitch*. Analisando a variação de amplitude e de energia no gráfico de uma nota musical de piano no domínio do tempo, apresentado na Figura 4.5, observamos que nos instantes iniciais da nota, a amplitude aumenta gradativamente até alcançar o valor máximo e, em seguida, decai até atingir o estado de repouso.

A variação de energia apresenta um comportamento similar e, como podemos observar, o início de uma nota musical é caracterizado por apresentar um aumento abrupto de energia. Na representação *pitch*, o incremento e decremento de energia são representados pela transição entre cores contidas em uma barra de cores.

Ao lado direito do gráfico da representação *pitch*, na Figura 4.3, há uma barra de cores vertical contendo valores que marcam os limites de sete intervalos. A barra de cores nada mais é do que os coeficientes STMSP mapeados em uma paleta de cores. No intervalo $[0; 0, 2]$, por exemplo, as cores que representam os limites inferior e superior desse intervalo são a cor preta e vermelho escuro, respectivamente e, entre os valores limite, há uma transição de cores na qual a tonalidade da cor fica mais

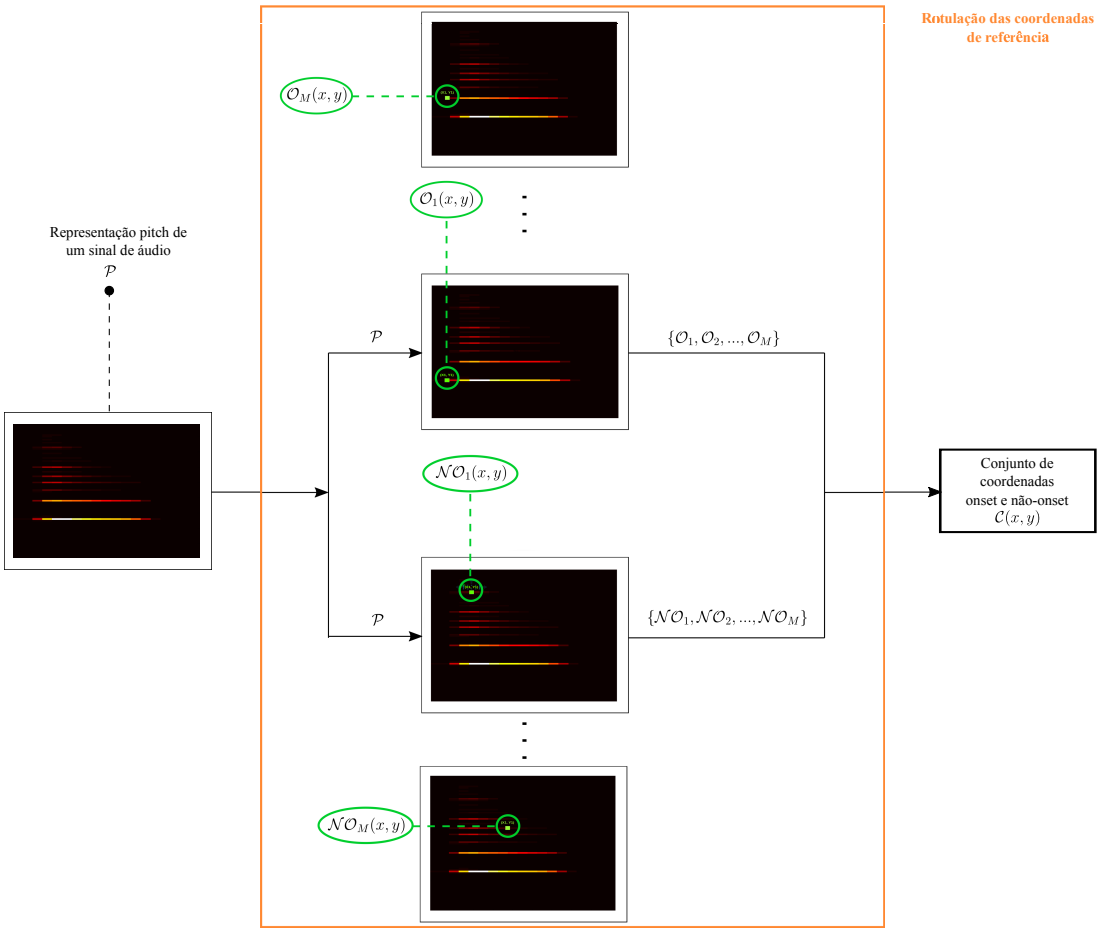

Figura 4.4: Processo de rotulação das coordenadas de referência da etapa de construção de modelo para classificação.

clara de forma gradativa.

A fase de aumento de energia, também chamada de fase de ataque, do ponto de vista das cores contidas na representação *pitch*, pode ser percebida como uma sequência encadeada de cores. A primeira cor da sequência possui a tonalidade mais escura e a última cor da sequência possui a tonalidade mais clara, representando a energia máxima contida no intervalo de tempo considerado.

Entre os MIDI *pitches* 50 e 60 da Figura 4.3, no intervalo de tempo de $0,2$ segundos a $0,4$ segundos, aproximadamente, há uma barra horizontal preenchida por um conjunto de cores. Dentro do retângulo verde, o conjunto de cores percebido inicia na cor preta e a tonalidade da cor gradativamente fica mais clara até atingir a cor branca. Baseado no mapeamento contido na barra de cores vertical, a sequência de cores descrita indica um incremento de energia e o pico de energia, no intervalo considerado, é representado pela cor branca.

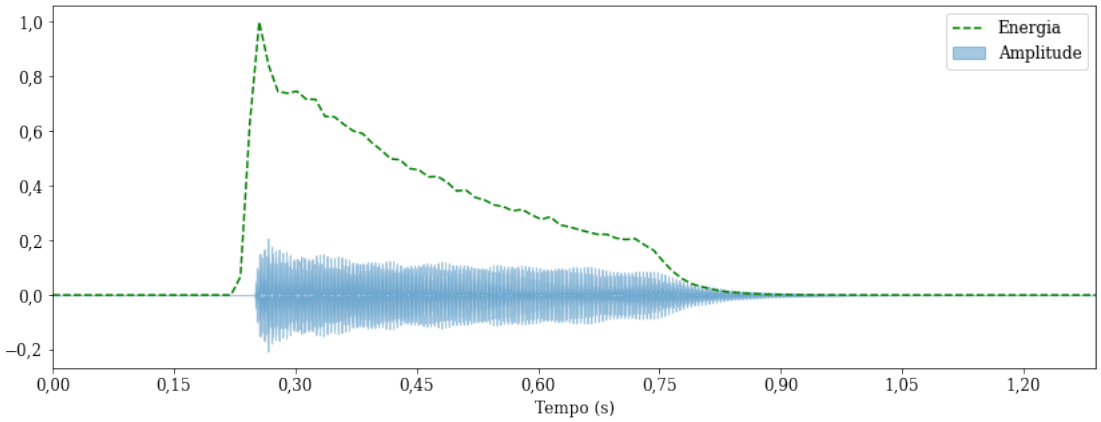

Figura 4.5: Amplitude versus energia da nota musical A3 no domínio do tempo.

Como a representação *pitch* analisada é da nota isolada de piano A3, o aumento de energia identificado corresponde à sua fase de ataque. Se fôssemos rotular a coordenada de um coeficiente situado no início da cor vermelho escuro da barra de cores horizontal, situada dentro do retângulo verde na Figura 4.3, essa coordenanda corresponderia a uma coordenada *onset*. Enquanto que coordenadas selecionadas em regiões contendo transições de cores com características de decremento de energia ou energia uniforme seriam rotuladas como coordenadas de referência não-*onset*.

Após a rotulação das coordenadas de referência, ocorre a extração de características, ilustrada na Figura 4.6, que tem como entrada o conjunto de coordenadas *onset* e não-*onset* $\mathcal{C}(x, y)$ e, como saída, um conjunto $\mathbf{V}$ composto por vetores de características *pitch* $[\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_M]$.

O primeiro passo da extração de características consiste em definir as dimensões das submatrizes $\mathcal{A}_{l \times c}$ que serão extraídas da representação *pitch* $\mathcal{P}_{L \times C}$. As Figuras 4.2 (b) e (c) ilustram um exemplo de extração de características no qual três submatrizes, com dimensões 2×3 e destacadas na cor laranja, são extraídas da representação *pitch* $\mathcal{P}$.

As coordenadas de referência, destacadas na cor azul, correspondem aos primeiros elementos de cada submatriz e servem de referência para delimitar as suas dimensões. Após a extração das características *pitch*, cada uma das submatrizes é vetorizada, formando assim, o conjunto de vetores de características *pitch* de saída $\mathbf{V} = [\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_M]$, onde o índice i representa o número de submatrizes extraídas e está dentro do intervalo $[1, 2, ..., M]$.

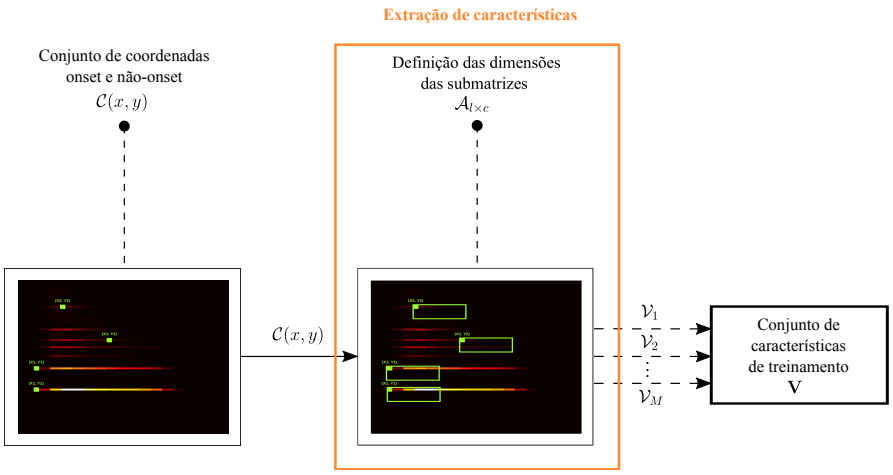

Figura 4.6: Diagrama da extração de características da etapa de construção de modelo para classificação.

A última subetapa da construção de modelo para classificação é o treinamento do classificador, apresentada na Figura 4.7. O classificador recebe como entrada o conjunto de vetores de características *pitch* $\mathbf{V}$, rotulados como *onset* ou não-*onset*, e gera como saída o modelo $\mathcal{F}$, capaz de mapear um ou mais vetores *pitch* em uma das classes: *onset* ou não-*onset*.

Os classificadores utilizados foram a SVM, *gradient boosting* e a CNN 1D. As duas primeiras técnicas são técnicas clássicas de aprendizado de máquina enquanto a terceira é uma técnica de aprendizado de máquina profundo. Além disso, os três algoritmos estão dentro da categoria de aprendizado supervisionado, ou seja, os vetores *pitch* de entrada contêm rótulos que especificam as classes as quais pertencem.

Esses três classificadores foram escolhidos por utilizarem diferentes abordagens para classificar um conjunto de dados. O *gradient boosting* faz uso de árvores de decisões para classificar os dados, a SVM utiliza hiperplanos e a CNN é uma rede neural composta, principalmente, por camadas de convolução que, como o próprio nome sugere, realizam a operação de convolução nos dados.

Cada classificador possui hiperparâmetros que precisam ser definidos antes de realizar o treinamento. Os hiperparâmetros do algoritmo *gradient boosting* são a taxa de aprendizado, subamostra, número de estimadores e profundidade máxima. A SVM, por sua vez, tem como hiperparâmetros o parâmetro de regularização C, o *kernel* e o *gamma* (γ).

Com relação a CNN 1D, além dos hiperparâmetros, a arquitetura da rede

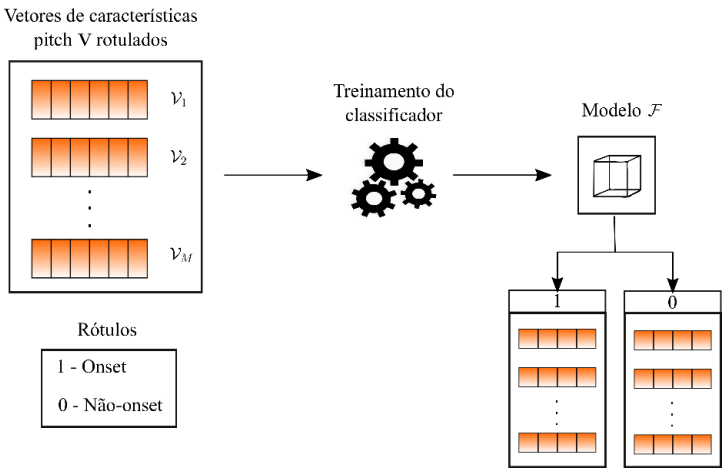

Figura 4.7: Subetapa de treinamento do classificador da etapa de construção de modelo para classificação.

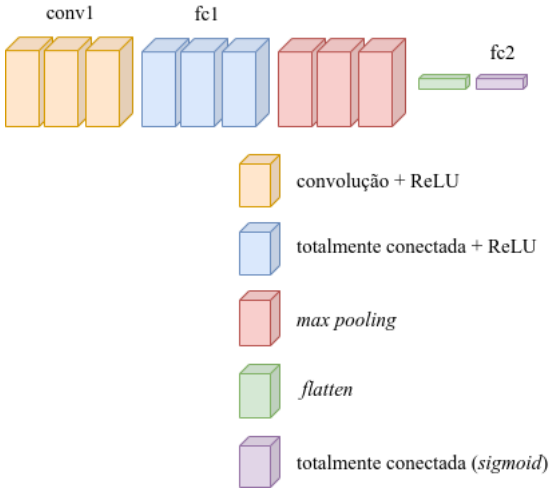

Figura 4.8: Arquitetura da rede neural convolucional 1D.

precisa ser definida. A arquitetura da CNN 1D utilizada, baseada em [57], está ilustrada na Figura 4.8 e segue a sequência: camada de convolução 1D com função de ativação ReLU, camada totalmente conectada com função de ativação ReLU, camada *max pooling*, camada *flatten* e, por último, camada totalmente conectada com função de ativação *sigmoid*.

Os hiperparâmetros da CNN 1D relacionados ao otimizador são o *learning rate* α e a constante ϵ, adicionalmente, há a função de perda, o *batch size* e o número de épocas.

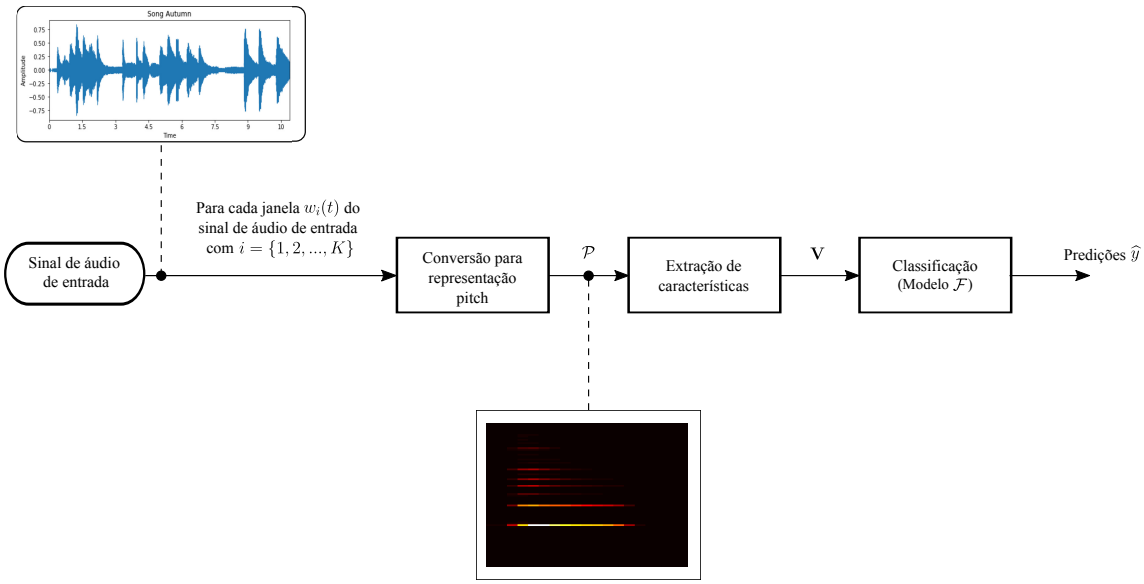

Figura 4.9: Diagrama da etapa de classificação dos vetores de características *pitch* do sistema de detecção automática de *onsets*.

4.2 Classificação dos vetores de características *pitch*

A segunda etapa de desenvolvimento do sistema de detecção automática de *onsets* é a classificação dos vetores de características *pitch*, apresentada no diagrama da Figura 4.9. Nessa etapa, janelas de áudio $w_i(t)$ de uma música instrumental de piano são fornecidas como entrada e a saída é um conjunto de predições $\hat{y}$ feitas pelo classificador.

O número de janelas i está dentro do intervalo $[1, 2, ..., K]$ e o tamanho do passo h, que define o intervalo de distância entre uma janela e a janela seguinte, é um parâmetro definido manualmente. A etapa de classificação dos vetores de características *pitch* é composta por três subetapas, descritas a seguir.

As duas primeiras subetapas também estão presentes na etapa de construção de modelo para classificação. A conversão do sinal de entrada $w_i(t)$ para a representação *pitch* $\mathcal{P}$ é igual nos dois casos. A subetapa seguinte, a extração de características, recebe como entrada a representação *pitch* $\mathcal{P}$ da janela de áudio $w_i(t)$ e não mais o conjunto de coordenadas de referência $\mathcal{C}(x, y)$, como ilustrado na Figura 4.10.

As submatrizes $\mathcal{A}_{l \times c}$ são extraídas ao longo de uma varredura realizada por

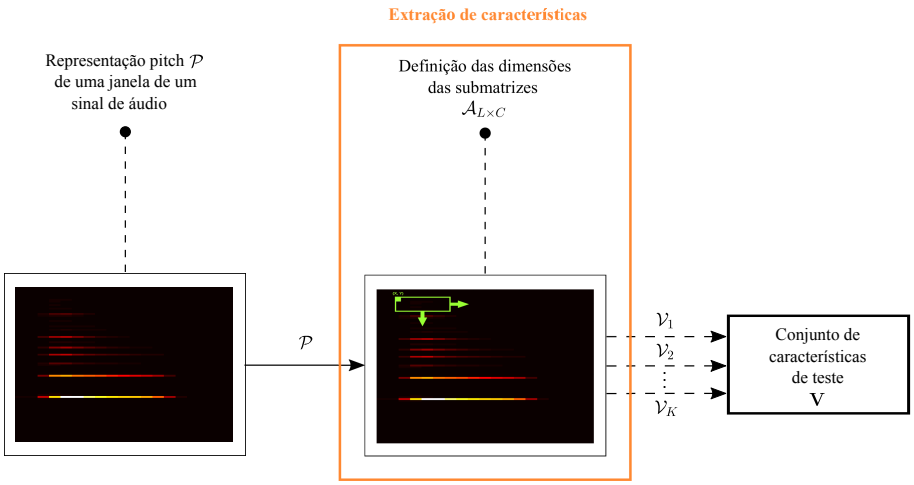

Figura 4.10: Subetapa de extração de características da etapa de classificação dos vetores de características *pitch*.

toda a representação *pitch* $\mathcal{P}$. As dimensões $l \times c$ das submatrizes são definidas previamente à varredura e as coordenadas de referência $\mathcal{C}(x, y)$ são identificadas após as delimitações das submatrizes que serão extraídas.

A Figura 4.11 apresenta os passos seguidos para realizar a varredura em uma representação *pitch* $\mathcal{P}$ com dimensões 3×5. Cada submatriz $\mathcal{A}$, que será extraída, tem dimensões 2×3 e os elementos destacados em vermelho são os coeficientes cujas coordenadas $c_{ij}(x, y)$ são as de referência. As submatrizes são extraídas enquanto as dimensões de $\mathcal{A}$ não extrapolarem os limites de $\mathcal{P}$ e até que toda a representação *pitch* tenha sido percorrida, adicionalmente, os tempos t_{ij} associados aos elementos de referência são armazenados, pois serão utilizados no pós-processamento.

No exemplo ilustrado na Figura 4.11, os elementos de referência são A_{11}, A_{12}, A_{13}, A_{21}, A_{22} e A_{23} e seus respectivos tempos são t_{11}, t_{12}, t_{13}, t_{21}, t_{22} e t_{23}. Todas as submatrizes extraídas da representação *pitch* P de uma janela de áudio $w_i(t)$ são vetorizadas e formam o conjunto de características *pitch* $\mathcal{V}_i$ e, ao fim da extração de características de todas as janelas de áudio, obtemos o conjunto de características de teste $\mathbf{V} = [\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_K]$.

A última subetapa da etapa de classificação dos vetores de características *pitch* é a classificação, ilustrada na Figura 4.12, na qual o conjunto de dados de teste $\mathbf{V}$ é fornecido como entrada para o modelo $\mathcal{F}$, gerado na etapa de construção

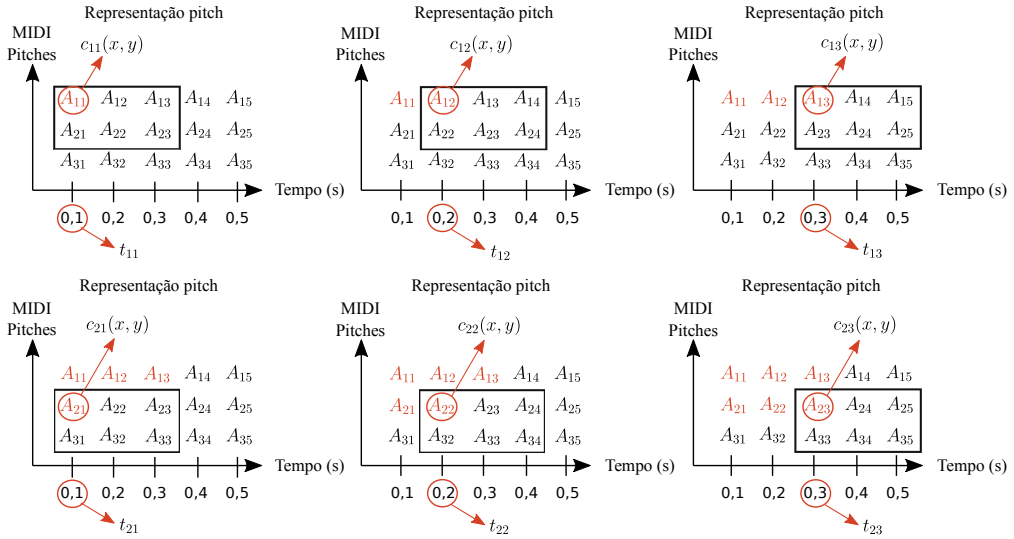

Figura 4.11: Exemplo de varredura em uma representação *pitch* P que ocorre durante a extração de características da etapa de classificação dos vetores de características *pitch*.

de modelo para classificação, e produz como saída o conjunto de predições $\hat{y}$. Cada vetor de características *pitch*, contido em $\mathbf{V}$, não possui rótulo e a predição resultante é binária, ou seja, *onset* ou não-*onset*.

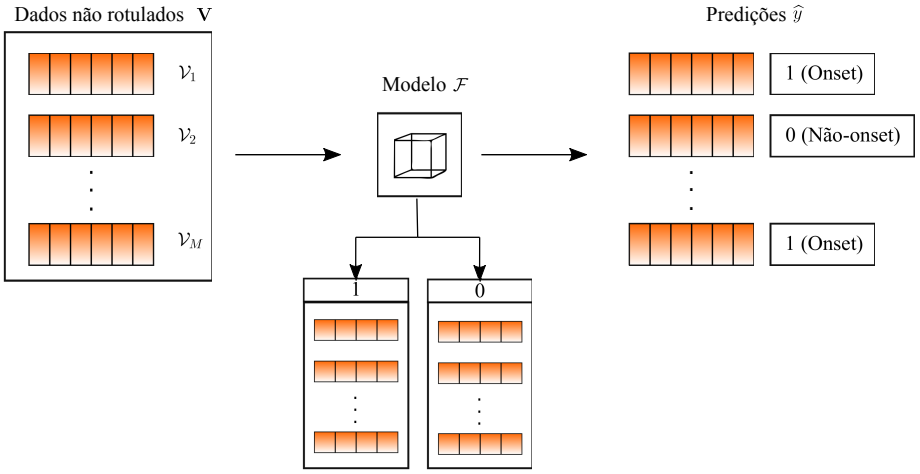

Figura 4.12: Subetapa de classificação da etapa de classificação dos vetores de características *pitch*.

4.3 Pós-processamento

O pós-processamento é a última etapa do sistema proposto, na qual buscas são realizadas para verificar quais tempos de *onsets* preditos estão dentro dos

intervalos de tempo *ground truth* dos *onsets* contidos no sinal de áudio de entrada.

Durante a extração de características da etapa de classificação dos vetores de características *pitch*, ilustrada na Figura 4.11, observamos que cada elemento de referência A_{ij}, circulado e destacado em vermelho, tem uma coordenada de referência $c_{ij}(x, y)$ e um instante de tempo de referência t_{ij}.

Como, após a extração de características, cada submatriz extraída é convertida para um vetor de características *pitch*, associado à cada vetor de características *pitch* temos uma $c_{ij}(x, y)$ e um t_{ij}. Quando a submatriz extraída for de uma área *onset*, caso da área situada dentro do retângulo verde na Figura 4.3, t_{ij} corresponderá a, aproximadamente, o instante de tempo do *onset*. Por isso, uma busca é realizada com o intuito de verificar se cada instante de tempo dos vetores de características preditos como *onset* está dentro de algum dos intervalos *ground truth* dos *onsets* $[t_{ref} - l; t_{ref} + l]$.

Os intervalos, definidos por $[t_{ref} - l; t_{ref} + l]$, delimitam as margens ou janelas de tolerância permitidas para que um dado instante de tempo seja considerado *onset* ou não. Cada intervalo é composto por t_{ref}, o instante de tempo de *onset* referência, e pela constante l, que delimita a distância, em milissegundos, que o intervalo terá, como ilustrado na Figura 4.13. Se o instante de tempo do vetor de características, predito como *onset*, estiver dentro de algum dos intervalos definidos, então a predição do classificador é considerada correta, caso contrário, a predição é considerada incorreta.

Na Figura 4.13, nos casos das janelas de tolerância localizados à esquerda e à direita, o classificador detectou os dois tempos de *onsets* corretamente, enquanto no caso do meio, o classificador deixou de detectar um tempo de *onset*. A detecção é considerada correta se houver ao menos um tempo de *onset* predito dentro da janela de tolerância.

Quando houver mais de um tempo de *onset* dentro de uma janela de tolerância, a média de todos os instantes de tempo contidos na janela é calculada e o tempo de *onset* final passa a ser o resultado da média dos tempos. A etapa de pós-processamento termina quando a busca pelos instantes de tempo de todos os vetores de características preditos como *onset* tiver sido feita.

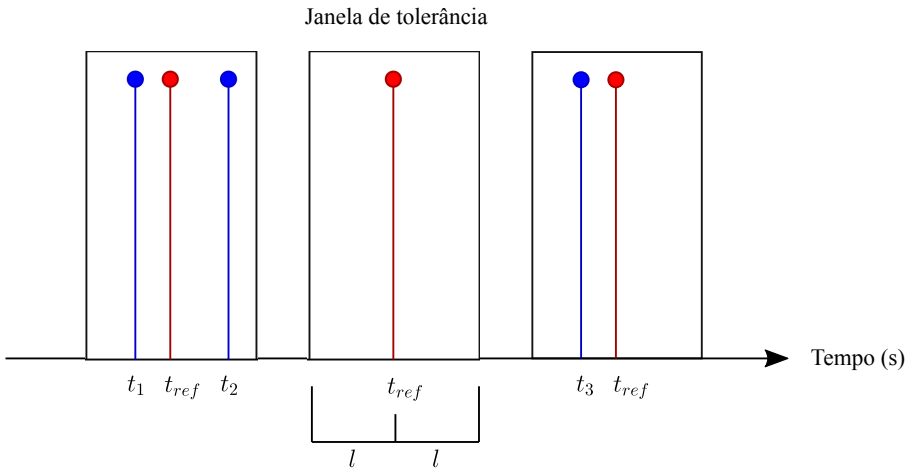

Figura 4.13: Ilustração da janela de tolerância em três casos, no primeiro, há dois tempos de *onsets* preditos dentro da janela (esquerda), no segundo, não há nenhum tempo de *onset* predito dentro da janela (meio) e, no terceiro caso, há um tempo de *onset* predito dentro da janela (direita).

4.4 Diferenças entre as abordagens empregadas

Na abordagem 1, na qual os classificadores SVM e GB são investigados, todas as três etapas da metodologia e suas respectivas subetapas são realizadas. Na abordagem 2, na qual a CNN 1D é investigada, as duas primeiras etapas são implementadas, ou seja, a etapa de construção de modelo para classificação e a etapa de classificação dos vetores de características.

Na primeira etapa da abordagem 2, a sequência de subetapas seguidas é: conversão para representação *pitch*, extração de características e, por último, o treinamento do classificador. Logo, nessa abordagem, não ocorre a rotulação das coordenadas de referência. Com relação à segunda etapa, todas as suas respectivas subetapas são realizadas.

Neste capítulo, a metodologia empregada para a construção do sistema de detecção automática de *onsets* foi abordada. As três etapas principais que a compõem foram descritas com detalhes. A primeira etapa foi subdividida em quatro etapas menores: conversão para representação *pitch*, rotulação das coordenadas de referência, extração de características e treinamento do classificador.

Na segunda etapa, a conversão para representação *pitch* e a extração de características são realizadas na sequência e, após isso, a classificação dos vetores de características é realizada. Por último, na etapa de pós-processamento, um proce-

dimento de busca é realizado para identificar instantes de tempo *onsets* preditos localizados dentro das janelas de tolerância de cada sinal de áudio. Adicionalmente, as diferenças entre as duas abordagens empregadas foi descrita.

No próximo capítulo, os experimentos realizados com o sistema proposto serão descritos, os resultados obtidos serão detalhados e uma análise destes resultados será feita.

Capítulo 5

Procedimento Experimental

Neste capítulo, as bases de dados utilizadas em cada experimento serão descritas, assim como os detalhes referentes às configurações dos experimentos realizados. Os resultados obtidos serão detalhados e, por último, uma análise dos resultados será feita.

5.1 Bases de Dados

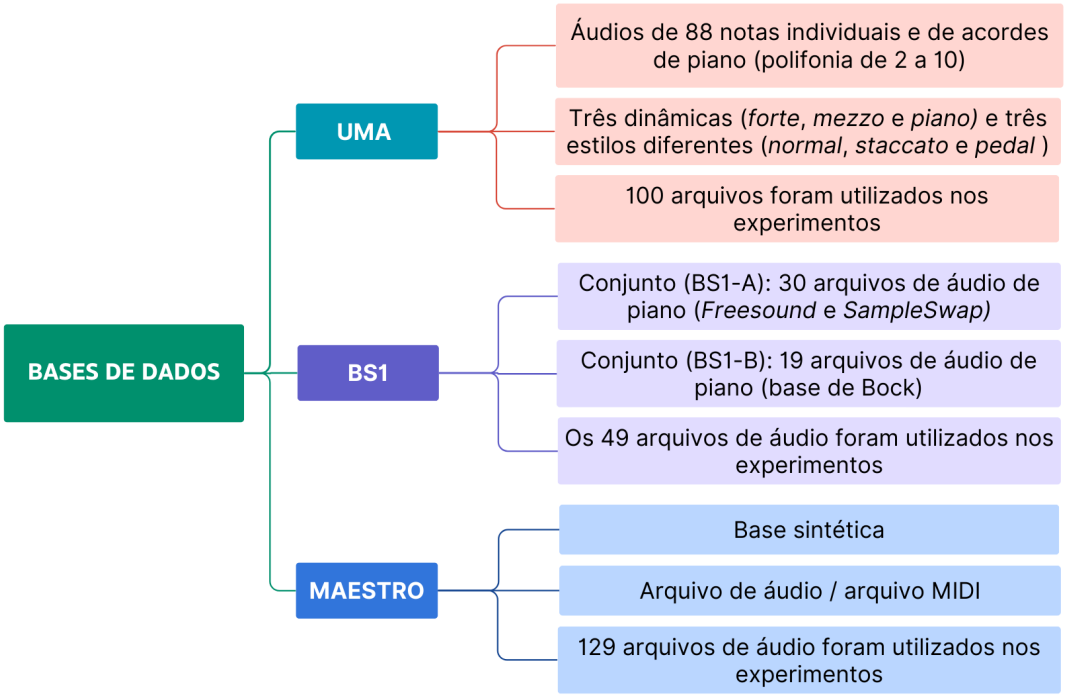

Figura 5.1: Resumo das bases de dados utilizadas nos experimentos.

Ao todo, três bases de dados foram utilizadas: UMA, BS1 e MAESTRO. Nos experimentos realizados com os classificadores máquina de vetor de suporte (SVM, do inglês, *support vector machine*) e *gradient boosting* (abordagem 1), a base UMA foi utilizada para treinamento enquanto as bases BS1 e MAESTRO foram utilizadas para teste, separadamente.

Nos experimentos realizados com a rede neural convolucional de uma dimensão (CNN 1D, do inglês, *one dimensional convolutional neural network*), ou seja, na abordagem 2, a base de dados BS1 foi utilizada tanto para treinamento como para validação e, para teste, a base MAESTRO foi utilizada. Depois, a ordem das bases foi invertida.

Um resumo do conteúdo de cada uma das bases juntamente com a quantidade de arquivos de áudio utilizados nos experimentos está descrito no diagrama da Figura 5.1.

5.1.1 UMA

A base de dados de acordes de piano UMA (do espanhol *Universidad de Málaga*) [58] é composta por gravações de áudio de notas individuais de piano e de acordes de piano. As gravações individuais são de cada uma das oitenta e oito teclas de piano enquanto as gravações dos acordes podem conter grau de polifonia que varia de dois à dez, ou seja, cada arquivo de áudio pode conter de duas à dez notas tocadas simultaneamente.

Além disso, todos os arquivos de áudio, seja de notas individuais ou de acordes, foram gravados considerando três dinâmicas diferentes, *forte*, *mezzo* e *piano*, e três estilos diferentes de tocar, *normal*, *staccato* e *pedal*. Logo, há nove versões diferentes de cada nota/acorde. A base UMA é formada por dez pastas e cada pasta contém os arquivos de áudio, no formato *.wav*, referentes a um grau de polifonia. A tabela 5.1 mostra detalhes da base.

Os sons de piano foram gravados usando uma frequência de amostragem de $44,1$ kHz e quantizados à 16 bits. Todos os arquivos de áudio são monocanais e foram gerados utilizando um sistema de produção digital de áudio profissional, no qual o piano *Kawai CA91* foi utilizado. Um detalhe importante a ser mencionado é que a base contém somente os arquivos de áudio, não há arquivos com anotações

Pasta	nº de arquivos
1	792
2	7.128
3	19.503
4	38.502
5	33.948
6	51.777
7	45.954
8	31.536
9	22.140
10	23.760
Total	275.040

Tabela 5.1: Descrição da base de dados UMA.

dos tempos dos *onsets*.

5.1.2 BS1

A base BS1 é formada por dois conjuntos de arquivos de áudio de piano, o primeiro conjunto (BS1-A) contém arquivos de áudio obtidos em [59] e [60], enquanto o segundo conjunto (BS1-B) é composto por arquivos de áudio de piano retirados da base de dados introduzida em [20], também conhecida como base de *Bock*.

O conjunto BS1-A é composto por 30 arquivos de áudio de piano monofônicos e polifônicos, dos quais 18 foram obtidos em [59] e 12 foram obtidos em [60]. Cada arquivo de áudio possui frequência de amostragem de 44, 1 kHz, está no formato *.wav* e foi convertido de estéreo para monocanal. Adicionalmente, alguns arquivos, por serem longos, foram segmentados de forma que o arquivo com menor duração possui aproximadamente dois segundos e o com maior duração possui aproximadamente dezoito segundos e meio.

Os tempos dos *onsets* contidos em cada arquivo de áudio do conjunto BS1-A foram rotulados manualmente, usando a ferramenta *Audacity*[1], para serem utilizados como referência. Cada arquivo de áudio foi exibido no *Audacity* e, através da análise visual do sinal no domínio do tempo e no domínio da frequência (espectrograma), os tempos dos *onsets* foram rotulados. Os tempos de *onsets* resultantes foram armazenados em um arquivo de texto, logo, associado à cada arquivo de áudio há

[1] https://www.audacityteam.org/

um arquivo de texto contendo os tempos dos *onsets* referência. O conjunto BS1-A possui um total de 588 tempos de *onsets*.

O conjunto BS1-B foi retirado da base de dados descrita em [20], especificamente 19 arquivos de áudio de piano. A base de dados original contém 321 arquivos de áudio de sons de vários instrumentos musicais monofônicos e polifônicos. Os arquivos de áudio possuem frequência de amostragem de 44,1 kHz, são monocanais e estão no formato *.flac*. Cada áudio possui um arquivo de texto contendo as respectivas anotações dos tempos de *onsets*. Além disso, o subconjunto de arquivos de áudio utilizado nos experimentos foi convertido de *.flac* para *.wav*. Detalhes da base toda estão descritos na Tabela 5.2.

Tipo de áudio	**nº de arquivos**	**nº de *onsets* brutos**	**nº de *onsets* combinados**
Misturas complexas	193	21.091	19.492
Pitched percussivo	60	2.981	2.795
Non-pitched percussivo	17	1.390	1.376
Instrumento de sopro	25	822	820
Instrumentos de corda com arco	23	1.180	1.177
Vocal	3	310	306
Total	321	27.774	25.966

Tabela 5.2: Descrição da base de dados de *Bock*.

Onsets brutos são todos os tempos de *onsets* rotulados manualmente enquanto o termo *onsets* combinados significa que todos os tempos de *onsets* situados dentro de um intervalo de 30 milissegundos foram combinados em um único tempo de *onset*, localizado na média aritmética de todas as posições. Instrumentos musicais como piano e guitarra fazem parte do grupo *pitched* percussivo enquanto instrumentos de percussão fazem parte do grupo *non-pitched* percussivo.

5.1.3 MAESTRO

A base de dados MAESTRO (do inglês, *MIDI and Audio Edited for Synchronous TRacks and Organization*) [61] é uma base sintética criada a partir de um processo denominado de *Wave2Midi2Wave*. No *Wave2Midi2Wave*, primeiramente, um áudio de referência é transcrito, produzindo assim a sua representação simbólica MIDI. Em seguida, essa representação é passada para um modelo de linguagem musical, que gera novas partes da música no formato MIDI, tendo como base o que foi transcrito e, por último, a nova representação MIDI gerada é sintetizada e gera o áudio da apresentação.

A base MAESTRO é composta por arquivos de áudio de músicas sintéticas de piano no formato *.wav*. Cada arquivo de áudio contém seu respectivo arquivo MIDI e, adicionalmente, há listas de arquivos de áudio da base que podem ser usadas para treinamento, validação e teste. Três versões da base MAESTRO estão disponíveis, V1.0.0, V2.0.0 e V3.0.0. Detalhes da versão V3.0.0, a versão utilizada nos experimentos, estão na Tabela 5.3.

Conjunto	**nº de arquivos**	**Duração (horas)**	**Tamanho (GB)**	**nº de notas (milhões)**
Treinamento	962	159,2	96,3	5,66
Validação	137	19,4	11,8	0,64
Teste	177	20,0	12,1	0,74
Total	1276	198,7	120,2	7,04

Tabela 5.3: Descrição da versão V3.0.0 da base de dados MAESTRO.

Cada arquivo de áudio possui frequência de amostragem de $44,1$ kHz e é estéreo. Para padronizar o subconjunto de arquivos dessa base utilizado nos experimentos, apenas quatro segundos dos arquivos de áudio foram utilizados e, a partir dos arquivos MIDI associados a cada arquivo de áudio, os seus respectivos tempos de *onsets* foram obtidos, os quais foram armazenados em um arquivo de texto. Além disso, o 129 arquivos de áudio utilizados nos experimentos foram transformados de estéreo para monocanal.

5.2 Experimentos

Os experimentos realizados foram divididos em duas partes, a primeira parte é composta pelos experimentos realizados com os classificadores SVM e *gradient boosting* (abordagem 1), que denominaremos de parte A dos experimentos, e a segunda parte ou parte B, corresponde aos experimentos realizados com a CNN 1D (abordagem 2). Os experimentos foram divididos dessa forma porque as abordagens empregadas são diferentes.

Treinamento	**Ext. de Carac.**	**Teste**	**Pós proc.**
UMA (100)	2 rotul./áudio (400)	BS1 (49)	[1, 150] ms
UMA (100)	2 rotul./áudio (400)	MAESTRO (129)	[1, 150] ms

[1]**Ext.**: Extração, **Carac.**: Características, **proc.**: processamento, **rotul.**: rotulações

Tabela 5.4: Configurações da parte A dos experimentos.

Na parte A dos experimentos, cujas configurações principais estão descritas na Tabela 5.4, para formar o conjunto de dados de treinamento, 100 arquivos de áudio da base de dados UMA foram utilizados, especificamente 10 arquivos de cada um dos dez graus de polifonia existentes na base.

Na subetapa de extração de características, 2 rotulações manuais *onset* e 2 rotulações manuais não-*onset* foram feitas por áudio, resultando em 200 vetores de características *pitch onset* e 200 vetores de características *pitch* não *onset*, totalizando 400 vetores ou observações no conjunto de treinamento.

Parâmetros	**Conf. 1**	**Conf. 2**	**Conf. 3**
Dim. da matriz	3×3	4×4	5×5
Tam. da janela (s)	$0,5$	$0,5$	$0,5$
Tam. do passo (s)	$0,5$	$0,5$	$0,5$

[1]**Conf.**: Configuração, **Dim.**: Dimensões, **Tam.**: Tamanho

Tabela 5.5: Configurações utilizadas na extração de características.

Três configurações foram utilizadas para extrair as características tanto na etapa de construção de modelo para classificação como na etapa de classificação dos vetores de características *pitch*. Essas configurações estão descritas na tabela 5.5.

Na primeira configuração, as dimensões da matriz foram 3×3, com tamanho da janela de $0,5$ segundos e tamanho do passo de $0,5$ segundos. Na segunda configuração, as dimensões da matriz foram 4×4 e os valores dos outros dois parâmetros

não se alteraram. Na terceira e última configuração, as dimensões da matriz foram 5×5 e os valores dos outros dois parâmetros permaneceram os mesmos.

Para avaliar os classificadores SVM e *gradient boosting*, as bases de dados BS1 e MAESTRO foram utilizadas, separadamente. Todos os 49 arquivos de áudio da base BS1 foram utilizados para formar o primeiro conjunto de dados de teste e 129 arquivos de áudio da base de dados MAESTRO foram utilizados para formar o segundo conjunto de dados de teste. Na extração de características, as três configurações descritas na Tabela 5.5 foram utilizadas para extrair os conjuntos de dados de teste.

Após calcular as predições, ou seja, na etapa de pós-processamento, o tamanho mais adequado da janela de tolerância, referente a cada arquivo de áudio, foi encontrado automaticamente testando vários tamanhos de janelas em um intervalo máximo de 150 milissegundos.

O tamanho da janela com o menor número de FNs (do inglês, *false negatives*) foi o tamanho escolhido e utilizado no respectivo arquivo de áudio. Adicionalmente, a média de todos os tempos de *onsets* contidos dentro de cada janela de tolerância foi calculada e o tempo resultante desse cálculo foi o tempo de *onset* final considerado.

Ao todo, 12 experimentos foram realizados na parte A. Seis experimentos utilizando a base BS1 como base de dados de teste e seis experimentos usando a base MAESTRO como base de dados de teste. Em cada experimento, os hiperparâmetros de cada um dos dois classificadores foram definidos utilizando a técnica de otimização *grid search*, na qual cinco valores foram testados para cada hiperparâmetro.

Os hiperparâmetros da SVM otimizados foram C, *gamma* (γ) e *kernel* enquanto os hiperparâmetros do *gradient boosting* otimizados foram a taxa de aprendizado, subamostra, número de estimadores e profundidade máxima.

A biblioteca utilizada para calcular as características *pitch* e gerar os gráficos da representação *pitch* foi a *chroma toolbox* [35], que contém funções desenvolvidas para serem executadas na plataforma MATLAB. Logo, na parte A dos experimentos, o MATLAB foi utilizado até a extração de características, tanto na etapa de construção de modelo para classificação como na etapa de classificação dos vetores de características *pitch*. As subetapas seguintes foram implementadas utilizando a linguagem de programação *Python*, especificamente a versão 3.7.13. As principais

bibliotecas utilizadas foram *scikit-learn*, *SciPy* e *librosa*.

Treinamento	**Validação**	**Teste**
MAESTRO (80%)	MAESTRO (20%)	BS1 (toda)
BS1 (80%)	BS1 (20%)	MAESTRO (toda)

Tabela 5.6: Informações sobre os conjuntos de dados utilizados na parte B dos experimentos.

A parte B corresponde aos experimentos realizados com a CNN 1D, que foram 8 no total, incluindo os experimentos realizados utilizando as características de espectrograma (*baseline*). A Tabela 5.6 descreve os conjuntos de dados utilizados nos experimentos. No primeiro conjunto de experimentos, a base de dados MAESTRO foi utilizada para formar os conjuntos de dados de treinamento e validação. A base BS1, por sua vez, foi utilizada para formar o conjunto de teste.

No segundo conjunto de experimentos, a base de dados BS1 foi utilizada para formar os conjuntos de dados de treinamento e validação, enquanto a base MAESTRO foi utilizada para formar o conjunto de teste. Em ambos os casos, 20% dos vetores de características da base foram utilizados para formar os conjuntos de dados de validação, o restante das instâncias foi utilizado para treinamento do classificador. As instâncias da base usadas para validação não foram usadas para treinamento.

Na abordagem empregada com a CNN 1D, primeiramente, cada arquivo de áudio foi dividido em segmentos de $0,2$ segundos, em seguida, as características *pitch* foram extraídas de cada segmento de áudio. Os vetores de características resultantes correspondem às instâncias que formaram os conjuntos de dados.

Com relação aos parâmetros de treinamento, o número de épocas definido foi 100 e o *batch size* foi 32. O otimizador e função de perda utilizados foram *Adam* e *binary cross entropy*, respectivamente. Além disso, um *callback* foi criado para programar o comportamento da taxa de aprendizado, o chamado *learning rate scheduler*.

No *learning rate scheduler*, durante as primeiras 10 épocas, o valor da taxa de aprendizado é mantido e, após isso, esse valor diminui exponencialmente. Adicionalmente, outros dois *callbacks* foram criados, um para monitorar a perda de validação e finalizar o treinamento, caso não haja mudanças significativas, e o outro para salvar o modelo com o melhor desempenho.

Toda a parte B dos experimentos foi desenvolvida utilizando a linguagem *Python*, na mesma versão utilizada na parte A. Para calcular as características *pitch*, a biblioteca *sync toolbox*[2] [62] foi utilizada. A função que calcula as características *pitch*, tanto na *sync toolbox* como na *chroma toolbox*, emprega as mesmas técnicas, por isso essas bibliotecas foram usadas. Além da *sync toolbox*, outras bibliotecas principais utilizadas foram *Keras* e *TensorFlow*.

As métricas utilizadas para avaliar o desempenho dos classificadores foram a acurácia (Acc), precisão (PC), sensibilidade ou *recall* (RC) e *Area under the ROC curve* (AUC). As equações das três primeiras métricas estão descritas em 5.1, 5.2 e 5.3.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

$$PC = \frac{TP}{TP + FP} \tag{5.2}$$

$$sensibilidade(RC) = \frac{TP}{TP + FN} \tag{5.3}$$

Onde TP (do inglês, *true positive*) corresponde ao caso positivo em que o instante de tempo analisado é um *onset* e foi corretamente detectado. TN (do inglês, *true negative*) corresponde ao caso positivo em que o instante de tempo analisado é não-*onset* e foi corretamente detectado.

FN (do inglês, *false negative*) corresponde ao caso em que o instante de tempo não foi detectado como *onset*, mas é um *onset*. FP (do inglês, *false positive*) corresponde ao caso em que o instante de tempo foi detectado como *onset*, mas não é um *onset*.

Por último, a AUC mede a habilidade que o classificador tem em distinguir as classes *onset* e não-*onset*, e isso é feito analisando a área sob a curva ROC (do inglês, *receiver operating characteristic*).

[2] https://github.com/meinardmueller/synctoolbox

5.3 Resultados e Discussão dos Resultados

As Figuras 5.2 e 5.3 mostram representações no domínio do tempo do sinal de áudio de piano nomeado como *song_096_minor-piano-melody*, da base BS1, juntamente com retas verticais pontilhadas nas cores azul e vermelha. Esses gráficos foram gerados após a detecção de *onsets* sem a etapa de pós-processamento. As retas verticais vermelhas indicam os instantes de tempo referência, ou seja, os tempos dos *onsets* contidos no arquivo de áudio que foram marcados manualmente.

As retas verticais azuis correspondem aos instantes de tempo dos *onsets* detectados, no caso da Figura 5.2, pelo classificador *gradient boosting*, e no caso da Figura 5.3, pelo classificador SVM. Os números localizados acima das retas pontilhadas azuis correspondem a quantidade de vezes que os instantes de tempo, especificados pelos números situados abaixo das retas, foram detectados como *onset* pelo classificador. Diferente das retas azuis, as retas pontilhadas vermelhas não possuem números nas partes superior e inferior.

Analisando as Figuras 5.2 e 5.3, observamos que os classificadores detectam tempos de *onsets* não somente nas áreas próximas à região do *onset*, mas ao longo da fase de decaimento da nota musical também. Em alguns casos, até mesmo nas fases subsequentes, o que pode ser visto nos primeiros $0,3$ segundos do áudio e entre os segundos $2,6$ e $2,8$, nos dois casos.

Para fins de esclarecimento, o instante de tempo especificado como $0,0$ nos gráficos representa um tempo detectado nos instantes iniciais do áudio e não em $0,0$, pois não é possível detectar um tempo de *onset* antes do sinal de áudio iniciar. Além disso, os intervalos entre detecções são de $0,1$ segundos porque a taxa de características da representação *pitch* é de $0,1$ segundos. Logo, nessa representação há um coeficiente *pitch* a cada $0,1$ segundos e, como a informação temporal utilizada deriva da representação *pitch*, o intervalo de detecção possui essa mesma característica.

Ao longo das análises realizadas nos gráficos resultantes da detecção de *onsets*, sem o pós-processamento e especificamente na base de teste BS1, observamos que o *gradient boosting* tende a detectar *onsets* ao longo de uma região mais extensa, ou seja, ao longo de mais fases do envelope da nota musical. O classificador SVM é suavemente mais concentrado, fazendo detecções em regiões um pouco mais próximas da região do *onset*.

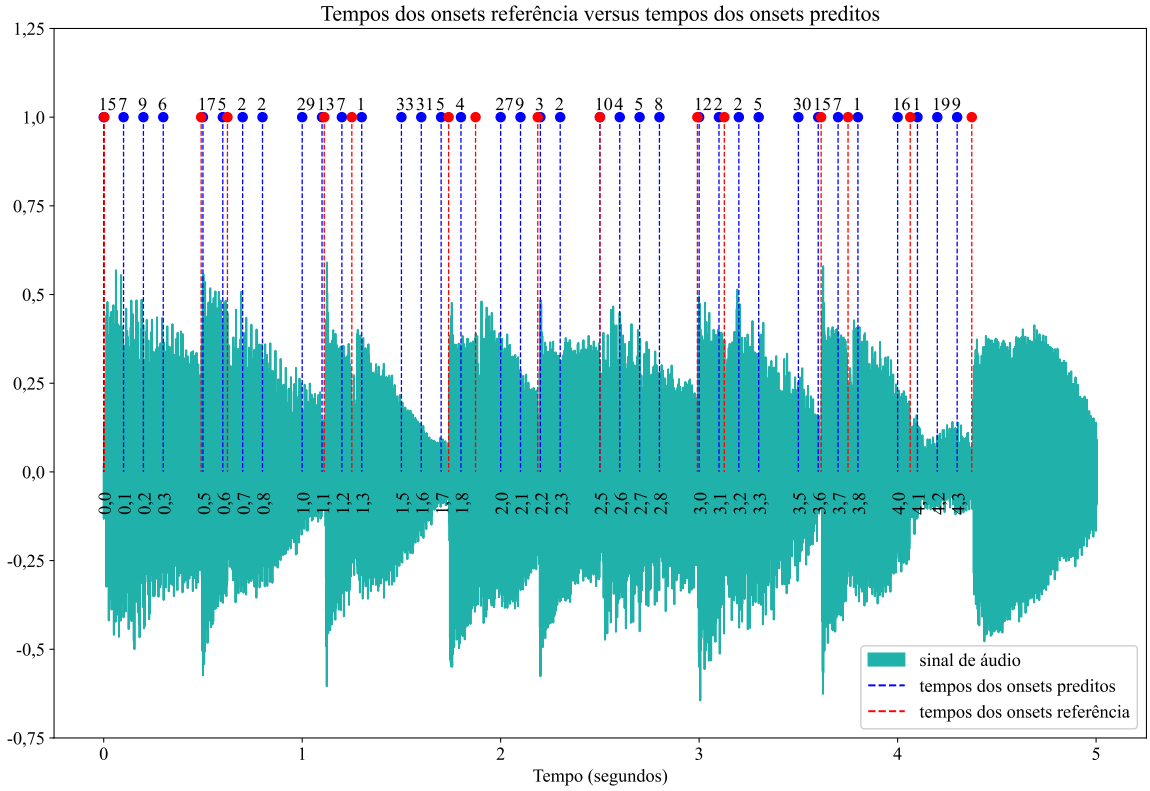

Figura 5.2: Representação do sinal de áudio *song_096_minor-piano-melody* no domínio do tempo juntamente com os tempos dos *onsets* de referência (retas verticais pontilhadas vermelhas) e tempos dos *onsets* preditos (retas verticais pontilhadas azuis) pelo classificador *gradient boosting*.

Nos primeiros $0,3$ segundos do áudio, por exemplo, a SVM fez duas detecções após a detecção mais próxima do *onset* enquanto o *gradient boosting* fez três detecções. No geral, considerando os resultados mostrados nas Figuras 5.2, 5.3 e no restante dos resultados obtidos com a base BS1, nos quais a janela de tolerância ainda não foi aplicada, os classificadores conseguiram identificar as regiões *onset* como regiões de interesse.

Apesar de haver um número considerável de detecções em regiões em que não há *onsets*, visualmente, a maioria dos tempos de *onsets* referência (retas verticais pontilhadas vermelhas) tiveram retas verticais pontilhadas azuis próximas ou sobrepondo as retas vermelhas. Isso mostra que as regiões com *onsets* não foram desconsideradas pelos classificadores. Outra questão observada foi que a SVM apre-

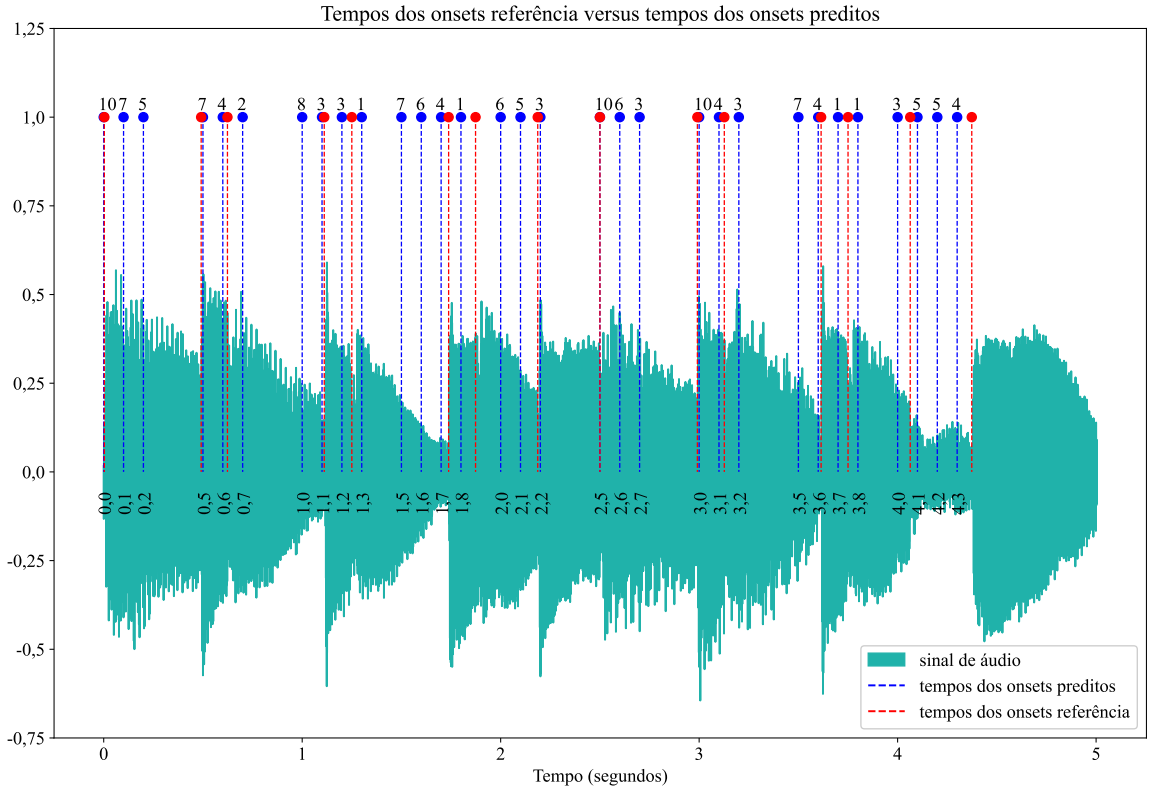

Figura 5.3: Representação do sinal de áudio *song_096_minor-piano-melody* no domínio do tempo juntamente com os tempos dos *onsets* de referência (retas verticais pontilhadas vermelhas) e tempos dos *onsets* preditos (retas verticais pontilhadas azuis) pelo classificador SVM.

sentou quedas bruscas no número de detecções quando o volume do áudio analisado era baixo.

Na análise dos gráficos resultantes da detecção de *onsets*, sem pós-processamento, tendo como base de teste a base MAESTRO, o classificar SVM apresentou mais dificuldade em detectar os tempos dos *onsets* e o *gradient boosting* apresentou um padrão de detecção repetido na maioria dos sinais de áudio. O GB detectou tempos de *onsets* a cada $0,1$ segundos desde o início até o fim do sinal de áudio, o que indica que o *gradient boosting* não conseguiu aprender os padrões que caracterizam as regiões *onset*.

Os tempos de *onsets* representados pelas retas verticais pontilhadas azuis, nas Figuras 5.2, 5.3, correspondem aos tempos de *onsets* preditos pelos classificadores,

porém, alguns desses tempos estão localizados longe da região que contém o *onset*. Por isso, para verificar se um tempo de *onset* foi detectado verdadeiramente, o tempo predito pelo classificador tem que estar próximo da região que contém o *onset*.

Dessa forma, para quantificar o quão próximo o tempo predito tem que estar, uma margem ou janela de tolerância foi definida ao redor de cada tempo de *onset* referência. Se houver algum tempo de *onset*, predito pelo classicador, dentro de uma janela de tolerância, então esse tempo de *onset* foi detectado, caso contrário, o tempo de *onset* não foi detectado.

Para definir o tamanho da janela de tolerância utilizada em cada sinal de áudio, vários tamanhos de janelas foram testados, considerando um intervalo máximo de 150 milissegundos. O gráfico ilustrado na Figura 5.9 mostra todos os tamanhos de janelas testados versus as porcentagens de *onsets* detectados. As porcentagens foram calculadas tendo como base o número total de tempos de *onsets* referência contidos no arquivo de áudio analisado. A presença de ao menos um tempo de *onset* dentro de uma janela de tolerância indica que aquele tempo de *onset* foi corretamente detectado.

Os tamanhos de janelas testados partiram de 10 milissegundos e incrementaram em intervalos de 20 milissegundos até alcançar 150 milissegundos. O tamanho da janela de tolerância escolhido foi o que apresentou o menor número de FNs (do inglês, *false negatives*), dessa forma, um tamanho de janela diferente foi escolhido para cada sinal de áudio.

Após aplicar a janela de tolerância com o tamanho definido, a média de todos os tempos contidos dentro de cada uma das janelas de tolerância foi calculada e os tempos médios resultantes corresponderam aos tempos de *onsets* finais. A Figura 5.4 mostra o gráfico no domínio do tempo, do arquivo de áudio nomeado como *song_ 096_ minor-piano-melody*, juntamente com os tempos de *onsets* referência (retas verticais vermelhas) e os tempos de *onset* finais (retas verticais azuis) após a subetapa de pós-processamento.

Dois casos podem ocorrer, no primeiro, há um tempo de *onset* final dentro da janela de tolerância de um dado tempo de *onset* referência, o que corresponde a um TP (do inglês, *true positive*) e, no segundo, não há, o que corresponde a um FN. Na Figura 5.4, todos os 15 tempos de *onsets* contidos no sinal de áudio foram

detectados.

Com relação a parte B dos experimentos, as Figuras 5.5 e 5.7 mostram os históricos de acurácia do classificador CNN 1D ao longo das épocas nas quais o treinamento e validação ocorreram, considerando os conjuntos de dados das bases BS1 e MAESTRO, respectivamente. As Figuras 5.6 e 5.8 mostram os históricos de perdas gerados utilizando esses mesmos conjuntos de dados.

O histórico de acurácia mostrado na Figura 5.5 é caracterizado por um aumento abrupto de acurácia nas épocas iniciais, tanto considerando o conjunto de dados de treinamento como o conjunto de dados de validação e, em seguida, há um decaimento de acurácia. A acurácia de validação, no intervalo entre as épocas 8 e 16, apresenta uma alta instabilidade, se comparada à acurácia de treinamento.

Após essa instabilidade, as acurácias de treinamento e validação iniciam um processo de estabilização e, posteriormente, se estabilizam com acurácia em torno de $0,60$ e $0,56$, respectivamente. A estabilização é atingida após cerca de 62 épocas. Quanto ao histórico de perdas, ilustrado na Figura 5.6, após cerca de 10 épocas, as perdas de treinamento e validação se reduzem a zero.

No histórico de acurácia mostrado na Figura 5.7, onde o treinamento e a validação foram realizados com o conjunto de dados da base MAESTRO, também há um aumento abrupto de acurácia nas épocas iniciais do treinamento/validação. No entanto, não ocorre uma alta instabilidade como aconteceu quando os conjuntos de dados da base BS1 foram utilizados.

Após o aumento da acurácia, um processo de estabilização inicia e as acurácias de treinamento e validação estabilizam em torno de, aproximadamente, $0,82$ e $0,76$, respectivamente. A estabilização é atingida após cerca de 100 épocas. Com relação ao histórico de perdas, ilustrado na Figura 5.8, após 10 épocas, aproximadamente, as perdas de treinamento e validação se reduzem a zero.

Na abordagem 1, a métrica utilizada para avaliar o desempenho dos classificadores foi a sensibilidade (*recall*) e os resultados obtidos das detecções dos tempos *onsets* estão descritos na Tabela 5.7. Analisando os resultados da Tabela 5.7 observamos, primeiramente, que o tamanho das matrizes utilizadas influenciaram no desempenho dos algoritmos GB e SVM, tanto na base BS1 como na base MAESTRO.

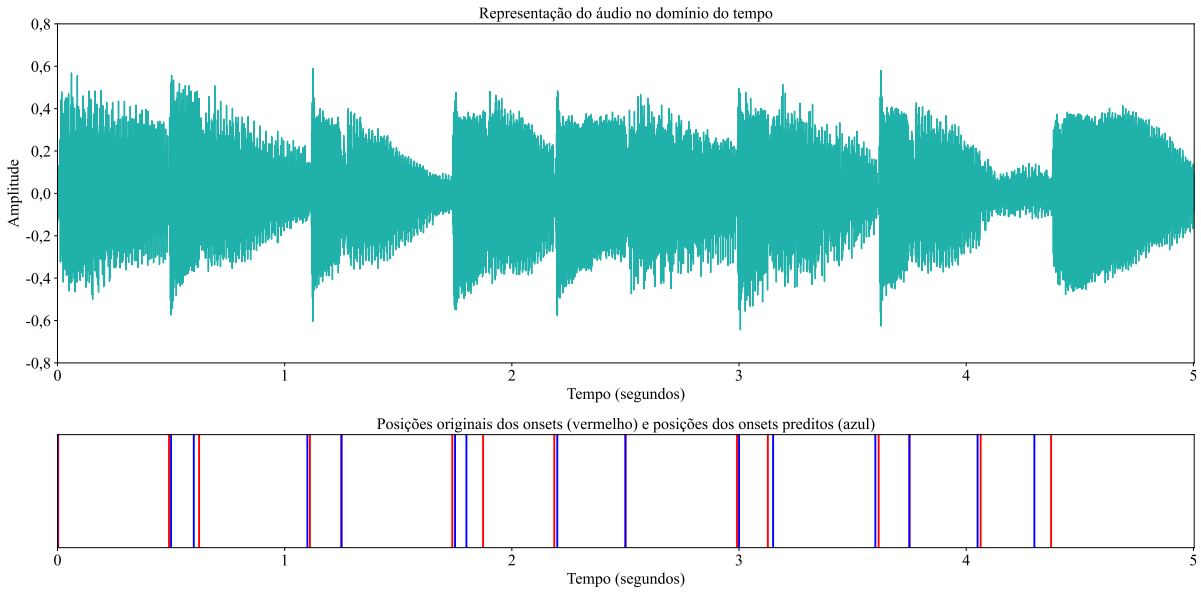

Figura 5.4: Representação do sinal de áudio *song_096_minor-piano-melody* no domínio do tempo juntamente com os tempos dos *onsets* de referência (retas verticais vermelhas) e tempos dos *onsets* preditos (retas verticais azuis) pelos classificadores *gradient boosting* e SVM após aplicar a janela de tolerância e calcular a média dos tempos *onsets* contidos nas janelas.

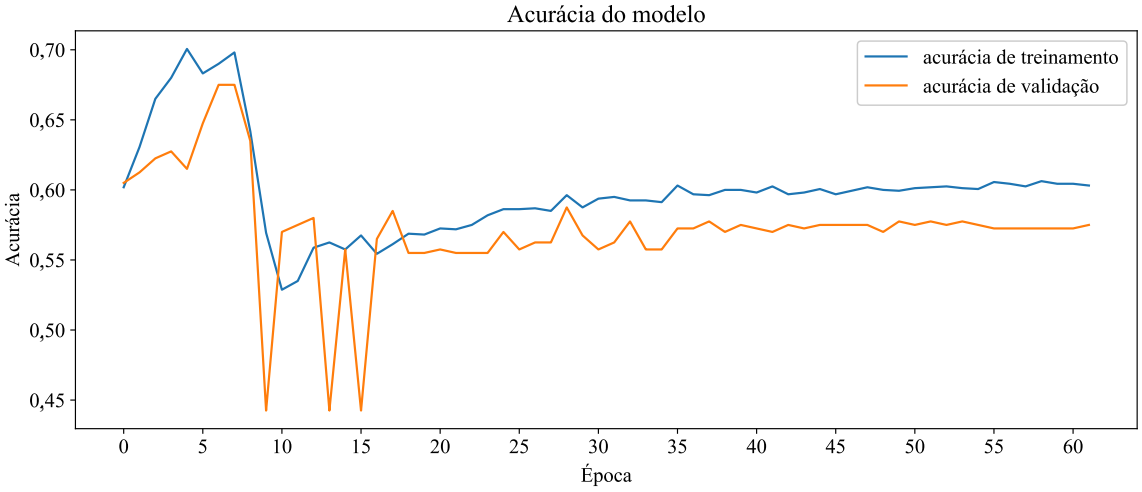

Figura 5.5: Histórico de acurácia obtido nos conjuntos de treinamento e validação ao longo das épocas na base de dados BS1 com a CNN 1D.

O valor de sensibilidade foi maior em todos os casos nos quais as dimensões da matriz utilizada foram 3×3, ou seja, na configuração 1. E, à medida que o tamanho da matriz aumentou, para 4×4 e depois para 5×5, nas configurações 2 e 3, o valor de sensibilidade diminuiu, apresentando uma maior queda na transição da configuração 2 para a 3 com a base BS1.

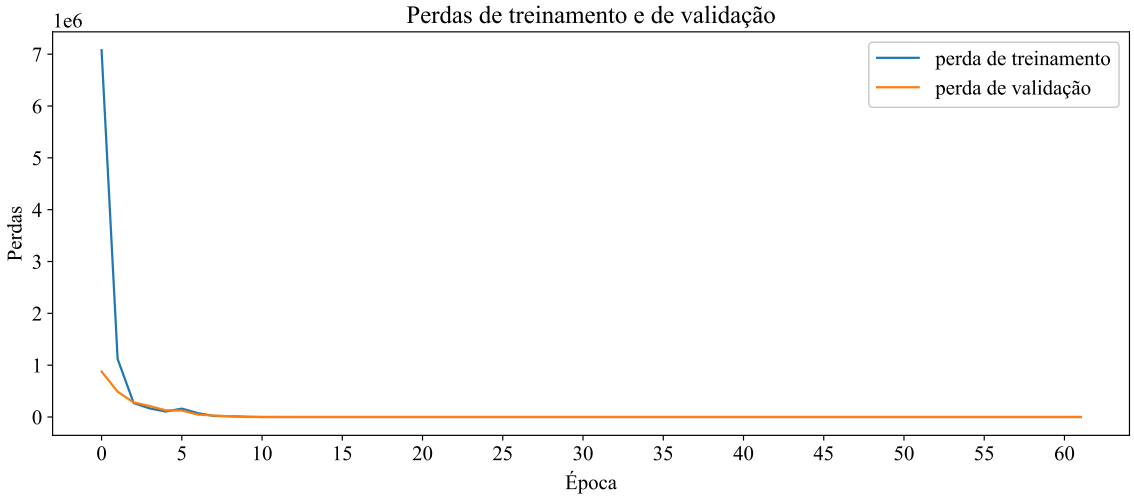

Figura 5.6: Histórico de perdas obtido nos conjuntos de treinamento e validação ao longo das épocas na base de dados BS1 com a CNN 1D.

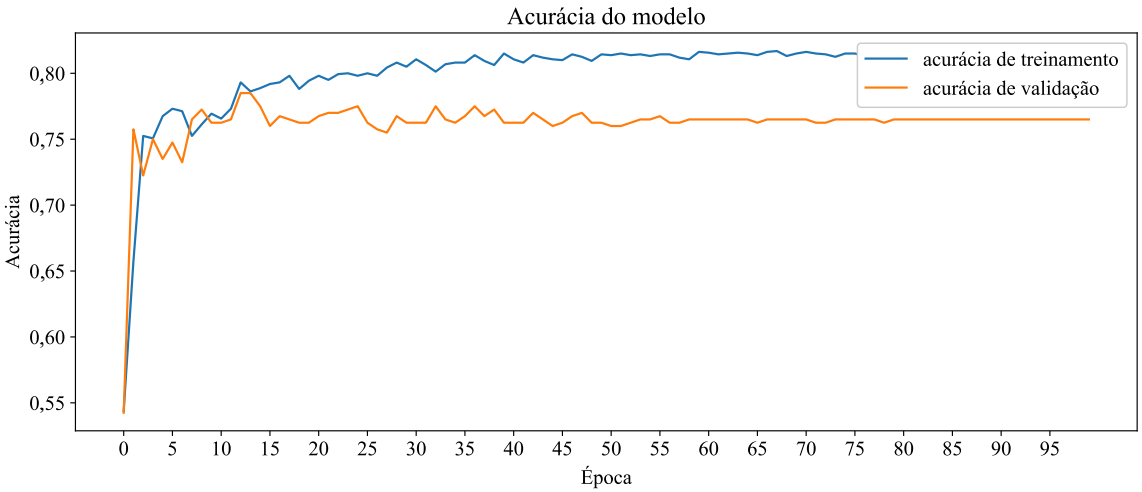

Figura 5.7: Histórico de acurácia obtido nos conjuntos de treinamento e validação ao longo das épocas na base de dados MAESTRO com a CNN 1D.

Apesar de o GB ter apresentado os melhores resultados, se comparado aos resultados obtidos com o classificador SVM, através de uma investigação mais aprofundada nas predições feitas pelo classificador, observamos que, principalmente na base MAESTRO, o GB repetiu um mesmo padrão de detecção de *onsets* na maioria dos áudios. Ou seja, mesmo nas regiões sem a presença de *onsets*, o algoritmo detectou *onsets*.

E, pelo fato de, na maoria dos casos, haver pelo menos um tempo de *onset* de referência próximo ao instante de tempo predito como *onset*, a janela de tolerância

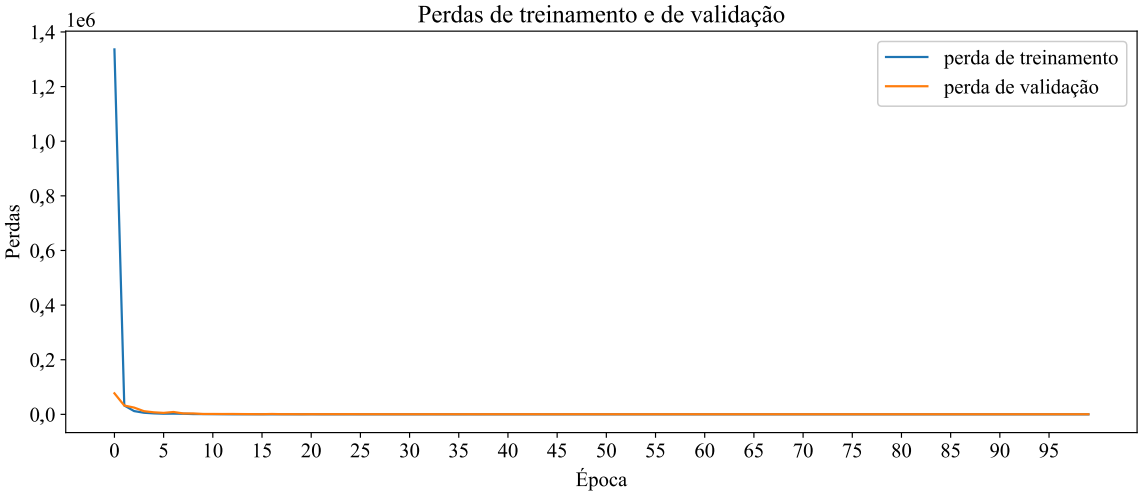

Figura 5.8: Histórico de perdas obtido nos conjuntos de treinamento e validação ao longo das épocas na base de dados MAESTRO com a CNN 1D.

Algoritmo	Configuração	Base de teste	Sensibilidade (%)
GB[1]	1	BS1	96,24
GB	2	BS1	95,03
GB	3	BS1	76,07
SVM [2]	1	BS1	85,97
SVM	2	BS1	83,50
SVM	3	BS1	65,60
GB	1	MAESTRO	96,10
GB	2	MAESTRO	93,52
GB	3	MAESTRO	91,62
SVM	1	MAESTRO	69,79
SVM	2	MAESTRO	69,43
SVM	3	MAESTRO	65,24

[1] *Gradient boosting*
[2] *Support vector machine*

Tabela 5.7: Resultados da detecção de *onsets* obtidos com os classificadores SVM e GB nas bases de dados BS1 e MAESTRO.

utilizada englobava o tempo de *onset* predito e, por consequência, uma alta quantidade de TPs foi computada. Com a base de dados BS1, houveram alguns intervalos entre a repetição desse padrão de detecção, porém, detecções em locais distantes das regiões com *onsets* continuaram. De posse dessa informação, concluímos que o GB não aprendeu os padrões que caracterizam um *onset* e permitem identificá-lo.

Os resultados dos experimentos realizados com a CNN 1D (abordagem 2) estão descritos na tabela 5.8. Tendo como base de dados de teste a BS1, sem aplicar a normalização, a métrica sensibilidade (*recall*) atingiu um valor de $98,92\%$

Algoritmo	Norm. [1]	Carac. [2]	Base de teste	Acc. (%)	P (%)	RC (%)	AUC (%)
CNN 1D	Não	*Pitch*	BS1	51,96	50,90	98,92	54,27
CNN 1D	Não	Espec.	BS1	68,55	66,33	74,77	72,60
CNN 1D	Sim	*Pitch*	BS1	52,56	51,21	98,86	55,10
CNN 1D	Sim	Espec.	BS1	68,28	63,94	83,18	79,19
CNN 1D	Não	*Pitch*	MAESTRO	67,67	69,09	70,00	67,54
CNN 1D	Não	Espec.	MAESTRO	67,95	67,38	75,96	75,63
CNN 1D	Sim	*Pitch*	MAESTRO	69,03	68,25	77,13	68,57
CNN 1D	Sim	Espec.	MAESTRO	77,75	78,35	79,85	77,82

[1] *Normalização*

[2] *Características*

Tabela 5.8: Resultados da detecção de *onsets* obtidos com o classificador CNN 1D nas bases de dados BS1 e MAESTRO.

utilizando as características *pitch*.

No entanto, os valores das outras métricas foram superiores quando as características espectrais foram utilizadas. Isso mostra que, no primeiro caso, o classificador detectou muitos FNs, porém, teve mais dificuldade em classificar as classes *onset* corretamente e, por conta disso, há um número alto de FPs. Com as características espectrais, os valores de acurácia e precisão foram maiores, apesar de o valor de sensibilidade ter sido menor.

Além disso, a métrica AUC mostra que, no segundo caso, o classificador conseguiu distinguir melhor as classes *onset* das não-*onset*. A aplicação da normalização resultou em uma melhora suave quando as características *pitch* foram utilizadas. Com as características de espectrograma, os valores de RC e AUC aumentaram cerca de $8,41\%$ e $6,59\%$, respectivamente, enquanto que os valores de acurácia e precisão diminuíram suavemente.

Utilizando a base de dados MAESTRO como base de dados para avaliação da CNN 1D, sem a normalização e considerando as características *pitch*, houve um aumento considerável nos valores das métricas acurácia, precisão e AUC, se comparado ao mesmo resultado obtido com a base BS1. Além disso, a precisão foi maior quando as características *pitch* foram utilizadas ($69,09\%$), enquanto as outras métricas apresentaram valores superiores quando as características espectrais foram usadas.

Após aplicar a normalização, com as características *pitch*, a precisão reduziu suavemente, enquanto os valores das outras métricas aumentaram. Com as caracte-

rísticas espectrais, por sua vez, houve um aumento em todas as métricas. A acurácia, por exemplo, apresentou um aumento de aproximadammente 10%. Isso mostra que a normalização teve mais impacto, positivo nesse caso, quando as características de espectrograma foram utilizadas.

No geral, na primeira abordagem, o GB apresentou o maior valor de sensibilidade, porém computou muitos TPs resultantes de uma detecção de *onsets* errônea, sendo assim, muitos FPs foram detectados. O SVM, apesar de ter apresentado um valor de sensibilidade inferior ao do GB, detectou uma quantidade menor de FPs. Além disso, os classificadores obtiveram os melhores resultados usando a configuração 1. Dessa forma, na primeira abordagem, o classificador SVM obteve o melhor desempenho.

Uma das possíveis razões de os dois classificadores terem apresentado melhor desempenho na base de dados BS1 é o fato de a maioria dos arquivos de áudio contidos na base MAESTRO terem um volume relativamente mais baixo, se comparado aos arquivos de áudio da base BS1. O que impactou no número de detecções, principalmente da SVM, já que a normalização não foi aplicada nessa abordagem.

Na segunda abordagem, utilizando a base BS1, o valor de sensibilidade foi maior utilizando as características *pitch* invés das características de espectrograma, no entanto, os valores de precisão, acurácia e AUC foram superiores no segundo caso. A normalização não teve tanto impacto positivo nos resultados como ocorreu com a base MAESTRO.

Adicionalmente, os resultados obtidos com as características de espectrograma foram superiores aos obtidos com as características *pitch* na maioria das métricas, utilizando a base MAESTRO. E, a aplicação da normalização teve um impacto positivo nos dois casos.

No geral, na segunda abordagem, o melhor resultado foi obtido utilizando a base MAESTRO, as características de espectrograma e aplicando a normalização, com acurácia de $77,75\%$, precisão de $78,35\%$, sensibilidade de $79,85\%$ e AUC de $77,82\%$. O segundo melhor resultado foi obtido utilizando as características *pitch* aplicando a normalização, com acurácia de $69,03\%$, precisão de $68,25\%$, sensibilidade de $77,13\%$ e AUC de $68,57\%$.

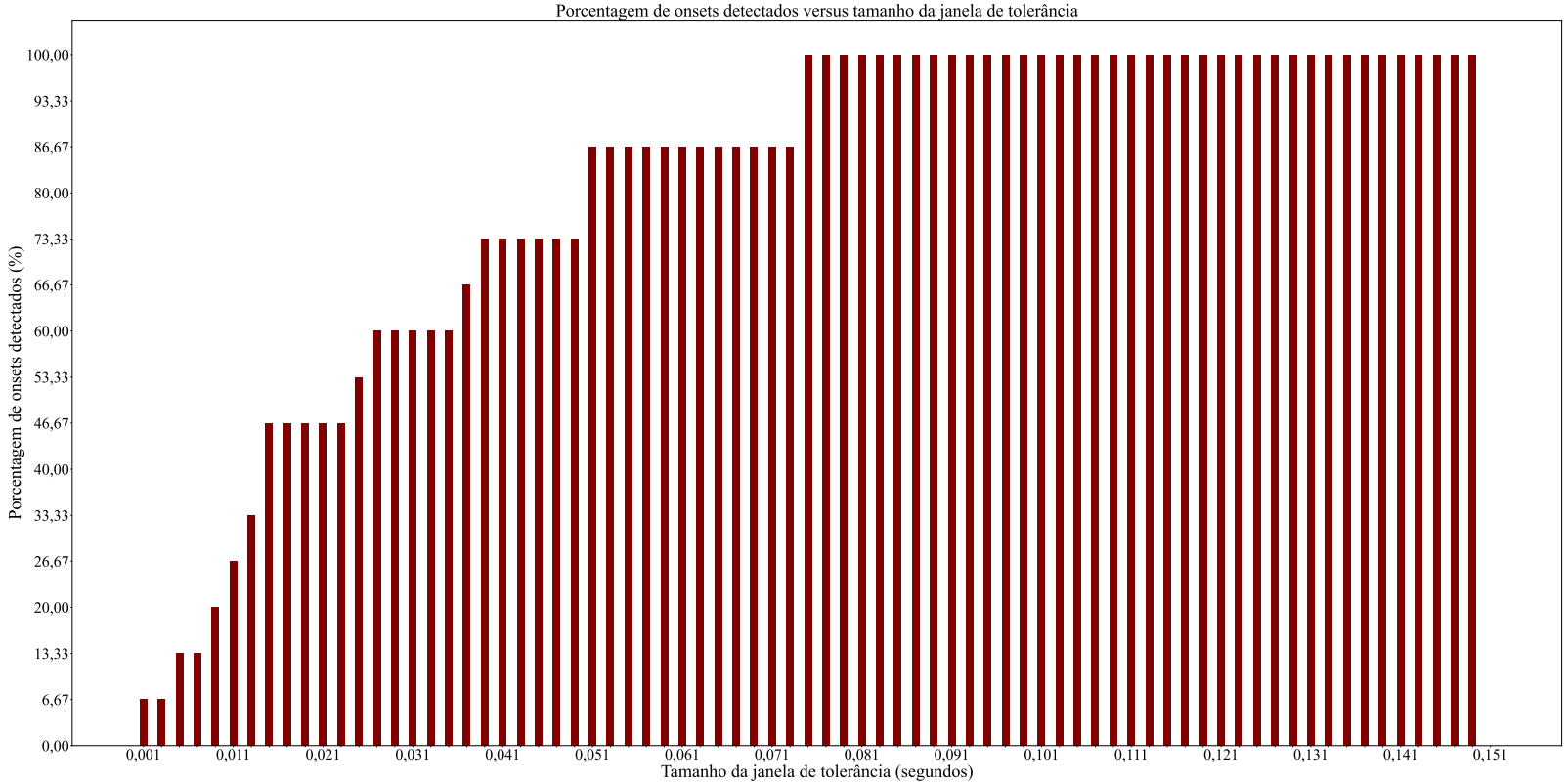

Figura 5.9: Porcentagens de *onsets* detectados pelos classificadores SVM e *gradient boosting* versus tamanho da janela de tolerância no sinal de áudio *song_096_minor-piano-melody*.

Capítulo 6

Conclusão

Um sistema automático de detecção de *onsets* em sinais de música usando aprendizado de máquina foi desenvolvido nesta dissertação. No *framework* proposto, os sinais de música de piano e a representação tempo-frequência *pitch* foram utilizados. Além disso, os classificadores SVM, *gradient boosting* e CNN 1D foram investigados.

No geral, o sistema foi dividido em três etapas principais: construção de modelo para classificação, classificação de vetores de características *pitch* e pós-processamento. A primeira abordagem empregada, na qual os classificadores SVM e *gradient boosting* foram utilizados, é composta pelas três etapas, enquanto a segunda abordagem, cujo classificador utilizado foi a CNN 1D, é composta pelas duas primeiras etapas.

Os objetivos principais da dissertação foram três. O primeiro objetivo foi construir um *framework* para detecção automática de *onsets* no qual diferentes tipos de sinais de áudio pudessem ser utilizados e cuja estrutura pudesse ser expandida. O segundo objetivo foi investigar uma representação diferente da representação do sinal de som no domínio do tempo. E, por último, propor uma solução com CNN 1D para detectar os *onsets* em sinais de música de piano.

A dissertação foi dividida em seis capítulos: introdução, fundamentos teóricos, trabalhos relacionados, metodologia para detecção de *onsets* em notas de piano, procedimento experimental e, por último, conclusão.

O primeiro capítulo forneceu o contexto dentro do qual a tarefa de detecção automática de *onsets* está inserida, introduziu o sistema proposto e os objetivos

principais desta dissertação. No segundo capítulo, os principais conceitos e técnicas empregadas ao longo do desenvolvimento do sistema foram abordados. O terceiro capítulo descreveu informações relevantes dos trabalhos relacionados ao sistema proposto e forneceu uma tabela de comparação.

No quarto capítulo, a metodologia utilizada para desenvolver o sistema de detecção de *onsets* foi descrita com detalhes. Em seguida, as bases de dados utilizadas e os experimentos realizados foram descritos no capítulo cinco, bem como com uma análise dos resultados obtidos foi feita. E, para finalizar, neste capítulo, considerações finais sobre o sistema desenvolvido estão sendo feitas.

Revisitando os três objetivos principais da dissertação, descritos no capítulo de introdução, o *framework* para detecção de *onsets* foi desenvolvido. Nesse *framework*, diferentes tipos de sinais de som podem ser explorados bem como outros tipos de classificadores de aprendizado clássico, considerando a primeira abordagem. Como saída, neste caso, o sistema fornece os tempos estimados dos *onsets*.

Na segunda abordagem, na qual a CNN 1D é utilizada, o sistema é capaz de identificar se há *onsets* dentro de um segmento de áudio ou não, sem fornecer os instantes de tempo dos *onsets*. Além disso, o *framework*, no geral, pode ser expandido.

Com relação ao segundo objetivo, uma representação diferente da representação do sinal de som no domínio do tempo foi investigada, especificamente a representação tempo-frequência *pitch*. Na representação *pitch* é possível identificar relações entre as bandas de frequências que a compõem com as notas musicais contidas na escala igualmente temperada. Além disso, essa representação pode ser considerada mais compacta, se comparada com a representação tempo-frequência espectrograma.

Com respeito ao último objetivo, uma solução para detectar *onsets* em sinais de música de piano utilizando a CNN 1D foi proposta na qual é possível classificar um segmento de áudio como tendo *onset* ou não.

Nos experimentos realizados utilizando a primeira abordagem, o *gradient boosting* apresentou valores de sensibilidade maiores do que a SVM, nas duas bases de teste BS1 e MAESTRO. No entanto, através de uma investigação mais aprofundada, observamos que o *gradient boosting* fez muitas detecções espúrias e, portanto, na análise final, o classificador SVM obteve o melhor desempenho na detecção dos

tempos dos *onsets*.

Nos experimentos realizados com a CNN 1D na base de dados BS1, ou seja, na segunda abordagem, a métrica sensibilidade foi superior quando as características *pitch* foram utilizadas invés das características de espectrograma (*baseline*). Na base MAESTRO, o desempenho da CNN 1D foi superior usando as características de espectrograma.

Com relação aos trabalhos futuros, podemos explorar outros tipos de representações de sinal de música como, por exemplo, a representação tempo-frequência *chroma*. Uma representação caracterizada por possuir apenas doze bandas de frequências no total resultantes da junção das bandas de frequência *pitch*. Podemos investigar se a redução no número de coeficientes da representação iria impactar ou não na detecção dos tempos de *onsets*.

Outro ponto que poderia ser trabalhado seria migrar a parte da primeira abordagem, implementada utilizando a plataforma MATLAB, para linguagem *Python*. O que facilitaria a investigação e alterações na subetapa de extração de características, pois a utilização da biblioteca *chroma toolbox* está atrelada ao uso de versões específicas do MATLAB.

Exemplos de parâmetros que poderiam ser investigados são o tamanho da janela de áudio selecionada, tamanho do passo e tamanho da matriz. Por último, com relação à segunda abordagem, uma forma de estimar os instantes de tempo dos *onsets* contidos nos segmentos de áudio poderia ser investigada, bem como a utilização de outros modelos da CNN 1D.

Referências Bibliográficas

[1] COBOS, M., LOPEZ, J. J., "Listen up — The present and future of audio signal processing", *IEEE Potentials*, v. 29, n. 4, pp. 40–44, 2010.

[2] DARJI, M. C., "Audio Signal Processing: A Review of Audio Signal Classification Features". In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2017.

[3] SILVA, N. E. M., AMOEDO, D. A., UCHÔA, M. S., et al., "Sound Pressure Level Measurement System in an Industrial Production Line using STM32 Platform". In: *Proceedings of the 2022 IEEE International Conference on Consumer Electronics (ICCE)*, 2022.

[4] VALADÃO, M. D., AMOEDO, D. A., TORRES, G. M., et al., "Automatic Video Labeling with Assembly Actions of Workers on a Production Line Using ResNet". In: *Proceedings of the 2022 IEEE International Conference on Consumer Electronics (ICCE)*, 2022.

[5] MATSUO, A. K., GUTERRES, B. A., COLARES, N. D., et al., "A Low-cost IoT Mobile System for Monitoring Vital Signs of Elderly People". In: *Proceedings of the 2022 Symposium on Internet of Things (SIoT)*, 2022.

[6] JOMBO, G., ZHANG, Y., "Acoustic-Based Machine Condition Monitoring—Methods and Challenges". In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2023.

[7] ISTRATE, D., BOUDY, J., MEDJAHED, H., et al., "Medical Remote Monitoring Using Sound Environment Analysis and Wearable Sensors". In: *Biomedical Engineering*, 2009.

[8] FERREIRA, D., JÚNIOR, W. S. S., CARVALHO, C., "Localização em Ambientes Internos Baseada em Aprendizado Supervisionado Utilizando Estações de Rádio FM". In: *Anais da VIII Escola Regional de Informática de Goiás*, 2020.

[9] MOROCUTTI, T., SCHMID, F., KOUTINI, K., et al., "Device-Robust Acoustic Scene Classification via Impulse Response Augmentation", submitted.

[10] COSTA, L. R., UCHOA, A. M., GIUSTI, R., et al., "Detecting Test Audio Signal Automatically with Fourier Analysis and Pattern Recognition". In: *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE-TW)*, 2021.

[11] SILVA, M. O., FERREIRA, D. A. O., OUCHI, K. Y., et al., "Automated Bright Pixel Detection System on LCD Displays". In: *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021.

[12] MÜLLER, M., *Information Retrieval for Music and Motion*. Springer, 2007.

[13] LERCH, A., KNEES, P., "Machine Learning Applied to Music/Audio Signal Processing", *Electronics*, 2021.

[14] CHEN, Y., FENG, Y., "Singing Melody Extraction Based on Joint Network with Res-CBAM". In: *Proceedings of 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2022.

[15] ABEBER, J., MÜLLER, M., "Jazz Bass Transcription Using a U-Net Architecture", *Electronics*, 2021.

[16] MUTHUMARI, M., AKASH, V., PRUDHVICHARAN, K., et al., "A Novel Model for Emotion Detection with Multilayer Perceptron Neural Network". In: *Proceedings of 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2022.

[17] ESTOLAS, E. A. L., MALIMBAN, A. F. V., NICASIO, J. T., et al., "Automatic Beatmap Generating Rhythm Game Using Music Information Retrieval

with Machine Learning for Genre Detection". In: *Proceedings of 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2020.

[18] DIXON, S., "Onset Detection Revisited". In: *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, 2006.

[19] FAGHIH, B., CHAKRABORTY, S., YASEEN, A., et al., "A New Method for Detecting Onset and Offset for Singing in Real-Time and Offline Environments". In: *Applied Sciences*, 2022.

[20] BOCK, S., KREBS, F., SCHEDL, M., "Evaluating the Online Capabilities of Onset Detection Methods". In: *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, 2012.

[21] SILVA JR, W. S., ARAUJO, G., DA SILVA, E. A. B., et al., "Facial fiducial points detection using discriminative filtering on principal components". In: *Proceedings IEEE International Conference on Image Processing (ICIP)*, pp. 2681 – 2684, 10 2010.

[22] SILVA JR, W. S., DA SILVA, E. A. B., GOLDENSTEIN, S., "Reconhecimento de Padrões utilizando Filtros de Correlação com Análise de Componentes Principais, Tese de doutorado, UFRJ", 06 2010.

[23] BERNARDO, G., KAZUHIRO, F., et. al., "Advances in subspace learning and its applications". In: *Proceedings Conference on Graphics, Patterns And Images (SIBGRAPI-EST)*, pp. 35–41, 10 2021.

[24] JESUS, A., JUNIOR, W., BITAR, N., et al., "Reconhecimento de Placas Veiculares em Cenários Complexos utilizando o Método do Subespaço Mútuo e Redes Neurais Convolucionais". In: *Proceedings XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT)*, 10 2022.

[25] JESUS, A., SANTOS, E., et. al., "Reconhecimento de Placas Veiculares em Cenários Complexos utilizando o Método de Subespaço, Dissertação de mestrado, UFAM", 11 2021.

[26] CHUAN, C., CHEW, E., "The Effect of Key and Tempo on Audio Onset Detection Using Machine Learning Techniques: A Sensitivity Analysis". In: *Proceedings of 8th IEEE International Symposium on Multimedia (ISM'06)*, 2006.

[27] JACQUES, C., ROEBEL, A., "Automatic Drum Transcription with Convolutional Neural Networks". In: *Proceedings of the 21th International Conference on Digital Audio Effects (DAFx)*, 2018.

[28] WANG, X., XU, W., LIU, J., et al., "Transition-Aware: A More Robust Approach for Piano Transcription". In: *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx2021)*, 2021.

[29] BERNARDO, G., LINCON, S., et. al., "A semi-supervised convolutional neural network based on subspace representation for image classification", *EURASIP Journal on Image and Video Processing*, v. 2020, 06 2020.

[30] BERNARDO, G., et. al., "Pattern-set Representations using Linear, Shallow and Tensor Subspaces", *CLEI Electronic Journal*, v. 25, 04 2022.

[31] KWON, T., JEONG, D., NAM, J., "Polyphonic Piano Transcription using Autoregressive Multi-State Note Model". In: *Proceedings of the 21th International Conference on Music Information Retrieval (ISMIR)*, 2020.

[32] HAWTHORNE, C., SIMON, I., SWAVELY, R., et al., "Sequence-to-Sequence Piano Transcription with Transformers". In: *Proceedings of the 22th International Conference on Music Information Retrieval (ISMIR)*, 2021.

[33] MAGALHÃES, T. N., LOUREIRO, M. A., "Training a Convolutional Neural Network for Note Onset Detection on the Clarinet". In: *18th Brazilian Symposium on Computer Music (SBCM)*, 2021.

[34] III, J. O. S., *Mathematics of the Discrete Fourier Transform (DFT), with Audio Applications*. W3K, 2007.

[35] MÜLLER, M., EWERT, S., "Chroma Toolbox: Matlab Implementations for Extracting Variants of Chroma-Based Audio Features". In: *Proceedings*

of the 12th International Conference on Music Information Retrieval (ISMIR), 2011.

[36] MED, B., *Teoria da Música*. 4th ed. MusiMed, 1996.

[37] WALKER, J. S., DON, G. W., *Mathematics and Music: Composition, Perception, and Performance*. CRC Press, 2013.

[38] LERCH, A., *An Introduction to Audio Content Analysis*. 2nd ed. IEEE Press and John Wiley & Sons, Inc., 2023.

[39] MÜLLER, M., *Fundamentals of Music Processing Using Python and Jupyter Notebooks*. 2nd ed. Springer International Publishing, 2021.

[40] LOY, G., *Musimathics: The Mathematical Foundations of Music*. The MIT Press, 2006.

[41] HERRMANN, B., MAESS, B., JOHNSRUDE, I. S., "Aging Affects Adaptation to Sound-Level Statistics in Human Auditory Cortex", *The Journal of Neuroscience*, 2018.

[42] MEUNIER, S., VANNIER, M., CHATRON, J., et al., "Asymmetry in perceived duration between up-ramp and down-ramp sounds as a function of duration", *The Journal of the Acoustical Society of America*, 2014.

[43] BELLO, J. P., DAUDET, L., ABDALLAH, S., et al., "A Tutorial on Onset Detection in Music Signals". In: *IEEE Transactions on Speech and Audio Processing*, 2005.

[44] HUZAIFAH, M., "Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks". 2017.

[45] POUR, A. F., ASGARI, M., HASANABADI, M. R., "Gammatonegram based speaker identification". In: *Proceedings of 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2014.

[46] ZIABARY, P. A., VEISI, H., "A Countermeasure Based on CQT Spectrogram for Deepfake Speech Detection". In: *Proceedings of 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 2021.

[47] JAMES, G., WITTEN, D., TIBSHIRANI, R., et al., *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

[48] NATEKIN, A., KNOLL, A., "Gradient boosting machines, a tutorial", *Frontiers in Neurorobotics*, Dec. 2013.

[49] NIELSEN, M., *Neural Networks and Deep Learning*. 2019.

[50] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., *Deep learning: Adaptive Computation and Machine Learning*. The MIT Press, 2016.

[51] STEINER, P., JALALVAND, A., STONE, S., et al., "Feature Engineering and Stacked Echo State Networks for Musical Onset Detection". In: *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, 2021.

[52] SCHLÜTER, J., BÖCK, S., "Improved Musical Onset Detection with Convolutional Neural Networks". In: *Proceedings of 2014 IEEE International Conference on Acoustics, Speech an Signal Processing (ICASSP)*, 2014.

[53] EYBEN, F., BÖCK, S., SCHULLER, B., et al., "Universal Onset Detection with Bidirectional Long-Short Term Memory Neural Networks". In: *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, 2010.

[54] TAN, H. L., ZHU, Y., CHAISORN, L., et al., "Audio Onset Detection using Energy-based and Pitch-based Processing". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010.

[55] C. G. V. D. BOOGAART, R. L., "Note Onset Detection for the Transcription of Polyphonic Piano Music". In: *Proceedings of 2009 IEEE International Conference on Multimedia and Expo*, 2009.

[56] YOU, W., DANNENBERG, R. B., "Polyphonic Music Note Onset Detection Using Semi-Supervised Learning". In: *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[57] DATATECHNOTES, "Classification with Keras 1D CNN Model in Python", https://www.datatechnotes.com/2020/02/classification-example-with-keras-cnn.html, 2022-12-01.

[58] BARBANCHO, A. M., BARBANCHO, I., TARDÓN, L. J., et al., *Database of Piano Chords: An Engineering View of Harmony*. Springer, 2013.

[59] TEAM, F., "Freesound", https://freesound.org/, 2022-09-01.

[60] BECKER, C., "SampleSwap", https://sampleswap.org/, 2022-09-01.

[61] HAWTHORNE, C., STASYUK, A., ROBERTS, A., et al., "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset". In: *International Conference on Learning Representations*, 2019.

[62] MÜLLER, M., ÖZER, Y., KRAUSE, M., et al., "Sync Toolbox: A Python Package for Efficient, Robust, and Accurate Music Synchronization". In: *Journal of Open Source Software*, 2021.