UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA ELÉTRICA

Graduação em Engenharia Eletrônica e de Telecomunicações

TRANSCRIÇÃO AUTOMÁTICA DE ÁUDIOS MONOFÔNICOS DE GUITARRA PARA TABLATURA

PEDRO HENRIQUE BITTENCOURT FERREIRA

UBERLÂNDIA

2023

TRANSCRIÇÃO AUTOMÁTICA DE ÁUDIOS MONOFÔNICOS DE GUITARRA PARA TABLATURA

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de bacharel em Engenharia Eletrônica e de Telecomunicações.

Orientadora: Dra. Milena Bueno Pereira Carneiro

Banca: Dr. Antônio Cláudio Paschoarelli Veiga, Dr. Gilberto Carrijo

UBERLÂNDIA

2023

Agradecimentos

À família, a base sólida, firme e acolhedora, fonte de sabedoria e amor.

Aos pais, que sempre acreditaram e confiaram em mim, sendo um grandioso abrigo para todos os momentos e sempre me impulsionando a pegar voo e tornar um ser humano gentil, feliz e grato.

Aos irmãos, parceiros na vida. Lutamos lado a lado, apoiando-nos e protegendo uns aos outros.

Aos avós, que provêm robustos laços de amor e sabedoria, acolhendo em todas as situações guiando-me com princípios sólidos de respeito, gratidão, ética e amor.

Aos padrinhos, que são inspiração e me motivam a crescer constantemente, seguir os sonhos e capacitar cada vez mais, profissionalmente e como ser humano.

À Pauline, namorada e parceira na jornada, de coração repleto de compaixão, presenciou o projeto desenvolver, incentivando, acolhendo nas dificuldades, sempre torcendo, apoiando e ajudando extraordinariamente com as revisões.

À Milena, orientadora que incentivou e acreditou no trabalho, dedicando-se a fazer acontecer e colocar em prática um projeto que significa tanto.

Aos amigos, rede de braços fortes, de confiança e companheirismo. Que ajudaram a levantar e continuar caminhando, aproveitar a vida, compartilhar momentos e me ensinaram que a felicidade é melhor compartilhada com aqueles que amamos.

Aos professores e mestres da infância, adolescência até a universidade, que são guias experientes que conhecem o caminho e motivam a percorrê-lo, ensinando através de exemplos de vida incorruptíveis, íntegros e sábios.

Ao Professor Abdulmotaleb El Saddik, que acreditou no meu potencial, acolheu-me e confiou desafios junto com toda a sua inestimável equipe integrando-me a uma família de incríveis pesquisadores que me inspiram.

Ao Jamilton, Carlos Henrique, Guilherme Romero e Carlos Maurício pelas oportunidades e confiança no desenvolvimento tanto técnico e profissional quanto também pessoal, como a honestidade, integridade, respeito e gentileza que moldaram minha conduta profissional.

Que os méritos deste trabalho se estendam a todos.

O caminho do conhecimento é infindável, mas que possamos percorrê-lo até o fim.

“Uma longa viagem começa com um único passo.” (Lao Tsé)

RESUMO

Este trabalho analisa a forma como a engenharia e seus recursos podem ser aplicados e utilizados nos processos musicais relacionados à transcrição automática de áudios monofônicos de guitarra para tablatura. Tendo isso em vista, o trabalho se pauta no banco de áudios desenvolvido pelo Instituto Fraunhofer de Tecnologia de Mídias Digitais (Ilmenau, Alemanha) com o intuito de utilizar, como parâmetro, os dados disponibilizados por uma instituição de credibilidade e importância neste âmbito, e verificar o papel da engenharia em paralelo e coexistência, e como forma segura de auxiliar nesses procedimentos. Trata-se, também, de uma forma comparativa e analítica de estudo acerca dos métodos *Super Flux, Probabilistic Yin* e *CREPE Pitch Tracker*.

Palavras-chave: processamento – sinais - engenharia – áudio – digital - transcrição - músicas - guitarra

ABSTRACT

This project analyzes how engineering and its resources can be applied and used in musical processes related to automatic transcription of monophonic guitar audios to tablature. With this in mind, the work is based on the audio database developed by the Fraunhofer Institute for Digital Media Technology (Ilmenau, Germany), with the intention of using the data made available by a credible and important institution in this field as a parameter, and to verify the role of engineering in parallel and coexistence, as a safe way to assist in these procedures. It is also a comparative and analytical study of the following methods Super Flux, Probabilistic Yin and CREPE Pitch Tracker.

Keywords: *processing– signal- engineering– audio– digital- transcription- songs- guitar*

Lista de figuras

Lista de tabelas

Lista de siglas e abreviaturas

ODF Função de Detecção de *Onset*

Hz Hertz

PYIN Probabilistic Yin

Fmel Frequência em Mel

FHz Frequência em Hertz

IDTM Instituto de tecnologia de mídias digitais

Lista de equações

Sumário

Capítulo 1 - INTRODUÇÃO

O conceito de música, no dicionário, é pautado pela combinação harmoniosa de sons, organização de melodias e conjunto de notas. E é nesta definição que se enquadra o papel da matemática. As estruturas musicais são repletas de concepções relacionadas, como teoria dos conjuntos e dos números, a álgebra abstrata, escalas musicais, proporção áurea e número de Fibonacci.

Essa relação direta entre música e matemática torna possível e, atualmente, necessária, a participação da engenharia. Sabe-se da existência e do papel da engenharia acústica, engenharia de áudio, processamento digital de sinais e processamento natural de linguagem neste âmbito. No entanto, a tecnologia em constante avanço abre um espaço indispensável para a engenharia e seu poder de facilitar e tornar mais práticos e eficazes os acessos a diversos recursos.

Com isso, o trabalho em questão analisa e testa os recursos da engenharia para comprovar e perceber a praticidade e eficiência da área nestes processos musicais.

1.1 OBJETIVO

O objetivo deste trabalho é analisar métodos de processamento digital de sinais para reconhecimento e extração de característica de sinais musicais de guitarra e implementar uma proposta de aplicação para transcrição automática de áudios monofônicos de guitarra para tablatura, utilizando as conclusões das análises realizadas e os conhecimentos acerca da engenharia de processamento digital de sinais e do instrumento em questão.

1.2 JUSTIFICATIVA

A importância da música no contexto social e cultural aponta a necessidade de serviços que levem em consideração e carreguem, em essência, os impactos que essa arte, consideravelmente popular e ampla, representa e causa aos indivíduos. Uma frase bastante conhecida de Friedrich Nietzsche (1844 – 1900) expressa a opinião de muitos: “Sem a música, a vida seria um erro”.

Desta forma, há que se considerar o expressivo número de ouvintes, bem como a quantidade de pessoas que tocam ou desejam tocar algum instrumento.

Uma pesquisa feita pela IFPI (Federação Internacional da Indústria Fonográfica), sobre o consumo de música em todo o mundo, mostrou que, em 2022, as pessoas passaram a ouvir 2 horas semanais a mais de música em comparação ao ano anterior. Concluiu-se que as pessoas gastam 20,1 horas ouvindo músicas semanalmente, contra as 18,4 horas de 2021. Isso aponta que, na prática, as pessoas estão ouvindo 34 músicas de 3 minutos a mais em uma semana.

A conexão que os indivíduos sentem com a música, por muitas vezes, estende-se ao desejo de inserir-se nesta esfera por meio do aprendizado. Um artigo da SABRA (Sociedade Artística Brasileira) mostra que violão e guitarra estão entre os cinco

instrumentos mais tocados no mundo, estando em primeiro e terceiro lugar, respectivamente.

No livro “Autoaprendizagem Musical – alternativas tecnológicas”, Gohn (2003) realizou um estudo sobre o uso de tecnologias na autoaprendizagem musical. Para ele, a definição de educação não-formal se configura nos “processos de ensino e aprendizagem que têm sua origem a partir da experiência prática e que usualmente não são codificados em sistemas curriculares oficializados” (GOHN, 2003, p.18).

Em contrapartida, a busca pela autoaprendizagem comumente é marcada por alguns desafios. Teoria e prática musical correlacionam-se de forma que, sem um estudo prolongado, profundo e duradouro, muitas dúvidas e dificuldades podem surgir na tentativa de aprendizagem. E, ainda que a tecnologia avance rápida e frequentemente, ela ainda não representa uma solução e não se apresenta como uma facilitadora de forma tão eficiente quanto poderia. O mesmo pode-se dizer acerca daqueles que desejam, de forma individual e independente, compor.

Por isso, a engenharia, neste contexto, surge como uma aliada indispensável quando há o interesse em contar com programas, plataformas, recursos e serviços que ofereçam suporte bem estruturado juntamente com a rapidez e praticidade da tecnologia.

Por se tratar de um projeto referencial, o trabalho considera o Instituto Fraunhofer de Tecnologia de Mídias Digitais (Ilmenau, Alemanha) como ideal foco para as análises quanto à aplicação da engenharia. Além disso, a escolha deste Instituto como fonte dos áudios analisados e trabalhados, advém da necessidade de, em busca de resultados fidedignos e esclarecedores para quem tem acesso ao trabalho e para quem, porventura utilizá-lo, futuramente como base a outros. Isso porque os áudios disponibilizados por esse banco não contêm ruídos, uma vez que são gravados em laboratório.

Por sua vez, o recurso da tablatura se mostra ideal, já que, além de ser a maneira com a qual se tornam expositivas as notas em questão, trata-se de uma forma de notação musical que descreve a posição de dedos para se tocar determinada nota e melodias em um instrumento (geralmente de cordas), apresentando-se como uma forma mais popular e acessível, sendo, inclusive, um modo mais prático de se aprender um instrumento.
Assim, a transcrição automática de áudios de guitarra e violão para tablatura torna acessíveis os estudos, conhecimentos e experiências que vão, até mesmo, para além do proposto pelo trabalho, abrindo um leque maior de possibilidades dentro do tema central.

A engenharia, neste contexto, comprovando sua eficácia neste tipo de aplicação, surge como uma aliada oferecendo mais possibilidade de bom desempenho e praticidade.

A relevância do trabalho também se justifica pela base que ele oferece e representa para outros projetos. Assim como apontam os diversos testes e estudos realizados neste exercício, a mesma aplicação, motivação e estratégia pode ser utilizada para o reconhecimento de outros tipos de sons e músicas, tornando possível ampliar o tema com profundidade e foco em distintas formas de reconhecimento de sons e

processamento de sinais. Desta forma, este trabalho pode se apresentar como uma referência concreta de estudos, representando não apenas um ponto de partida e base estrutural, como, também, um referencial de etapas, teorias, estudos, testes, programas, mapeamento de resultados e, assim, servindo como um projeto o qual pode ser consultado e considerado na produção de outros.

1.3 REFERENCIAL TEÓRICO

O curso "Processamento de Sinais de Áudio para Aplicativos Musicais", oferecido pela Universidade Pompeu Fabra de Barcelona e pela Universidade de Stanford e ministrado pelo professor Xavier Serra, do Departamento de Tecnologias de Informação e Comunicação e Diretor do Grupo de Tecnologia Musical da Universitat Pompeu Fabra em Barcelona, e por Julius O. Smith, professor de música e de engenharia elétrica na Universidade de Stanford, e supervisor de pesquisas relacionadas no Centro de Pesquisa de Computação em Música e Acústica (CCRMA); foi referência para metodologias específicas para processamento de sinal de áudio e de uso em aplicações reais, com técnicas de processamento espectral para descrição e transformação de sons. Teoria e prática do curso embasam processos de análise, sintetização, transformação e descrição de sinais de áudio no contexto de aplicações musicais. Contou-se com o apoio do professor Xavier Serra, em videochamada, para discussão de técnicas e metodologias aplicadas ao trabalho, como a seleção de melhores procedimentos para calibração de parâmetros, abordagens técnicas e motivação para continuidade deste trabalho, o qual aborda um tema bastante relevante. Além disso, o docente contribuiu com diversos materiais de estudo e referência.

Estudos do Laboratório Internacional de Áudios Erlang [1] e o website Music Information Retrieval [2] (utilizado por grandes universidades, como Harvard e Stanford), aliados ao processo de detecção de início, auxiliaram no estudo em busca do reconhecimento automático de eventos musicais em um sinal de áudio para recuperação de informações musicais, mostrando como detectar um início, o momento exato que marca o início da parte transiente de um som, ou o primeiro momento em que um transiente pode ser detectado com segurança.

Destacam-se, como outros elementos essenciais para a execução do projeto, o intercâmbio de informações com Renato Profeta [3], com doutorando na Ilmenau University of Technology, e o material didático em seu canal do YouTube [4] no qual publica, frequentemente, informações relevantes de estudos e técnicas de Processamento Digital de Sinais voltados para extração de informação em materiais sonoros.

Além disso, o artigo Monophonic Audio-Based Automatic Acoustic Guitar Tablature Transcription System with Legato Identification (2021) [5] foi usado para importante pesquisa em meio a artigos recentes (IEEE), destacando-se como um dos que mais se aproximou dos resultados desejados. Outros trabalhos como Inharmonicity-Based Method for the Automatic Generation of Guitar Tablature (2012) [6], Automatic Guitar Music Transcription (2012) [7] e Automatic Real-Time electric guitar audio transcription (2011) [8] tiveram grande importância para entendimento de técnicas e metodologias utilizadas nesse ramo de pesquisa e as soluções disponíveis.

1.4 METODOLOGIA

Algoritmos e métodos de processamento digital de sinais mais popularmente utilizados para identificação de início de notas e frequência fundamental são estudados a fim de avaliar a eficiência de resultados e encontrar o melhor ajuste de parâmetros para seu funcionamento. É utilizando o banco de áudios desenvolvido pelo Instituto Fraunhofer de Tecnologia de Mídias Digitais (Ilmenau, Alemanha) [9] que contém conjuntos de áudios exclusivamente monofônicos gravados profissionalmente e que não consideram ruídos, efeitos e técnicas de tocabilidade do instrumento. Neste contexto, alia-se diversos conhecimentos de processamento digital de sinais apresentando uma avaliação técnica dos métodos, testes e resultados para a implementação.

Para análise e recriação de um sistema de transcrição automática de áudios para tablatura, diversos artigos foram consultados. A partir deles, foi possível definir e optar pelas implementações em *Python* de métodos de processamento digital de sinais.

Foram realizados testes com os métodos comumente utilizados neste âmbito. Com isso, escolhem-se as melhores abordagens e ajustes a fim de reunir em um só sistema, os processos relevantes para a transcrição de notas identificadas em áudios monofônicos para tablatura.

Com o desenvolvimento do projeto, diversos testes de parâmetros e uso de técnicas matemáticas e de processamento de sinais foram utilizados na busca pela qualidade, acurácia e precisão dos métodos em análise. Algoritmos como *Super Flux*, para detecção de *onsets*, *Probabilistic Yin* e *CREPE*, para identificação de frequência fundamental foram analisados neste trabalho e ajustados para atingir a meta de implementação da aplicação.

O processo também incluiu estudos acerca de Escala e espectrograma Mel, *Zero-padding*, redes neurais convolucionais, rastreamento de trajetória, autocorrelação e técnicas de visão computacional tal como Filtro de Máximos.

As etapas do processo, alicerçadas pelos artigos pesquisados, tiveram, também, ordens e procedimentos adaptados por adequação, como resultado dos testes e das análises, que apontaram a necessidade de mudanças, calibração de parâmetros e adaptações para alcançar o resultado almejado.

Desta forma, as fases são listadas da seguinte forma:

- Recepção do áudio monofônico;

- Segmentação por *onset* (início e final de notas).

- Identificação de frequência fundamental : Pode-se utilizar métodos como *YIN, CREPE* e *Probabilistic YIN*, sendo o último de melhor performance nas análises deste trabalho.

- Seleção de posição: Seleção de combinação de corda e casa, referente à nota identificada.

- Transcrição para tablatura

Capítulo 2 - Métodos, Técnicas e Termos

Neste capítulo será introduzido o conhecimento acerca de termos utilizados em música, breve explicação sobre o instrumento guitarra e violão e estudos sobre os métodos que serão utilizados na implementação.

2.1 Notas Musicais e o instrumento guitarra

Notas musicais são os menores elementos que compõem uma melodia. As notas são identificadas pela sua frequência fundamental e sua oitava, por exemplo: a nota Lá na segunda oitava tem a frequência fundamental de aproximadamente 110 Hertz. A cada oitava acima, o valor da frequência duplica. Ou seja, para Lá na terceira oitava, tem-se a frequência de aproximadamente 220 Hertz.

As notas são acompanhadas de harmônicas e, para a identificação da frequência fundamental, essas são essenciais para os métodos funcionarem corretamente. Vale notar que não necessariamente a frequência fundamental é a de maior amplitude em um sinal de música. Usualmente, utiliza-se uma notação que transcreve as notas para letras. Idealizada pelo monge italiano Guido D'Arezzo (992-1050), a forma de notação que será utilizada neste trabalho é a seguinte:

Lá: A, Si: B, Dó: C, Ré: D, Mi, E, Fá: F, Sol: G

Chamados de acidentes, o sustenido e bemol são meios-tons que se encontram entre dois tons naturais por exemplo: entre C e D, temos C sustenido ou D bemol. A notação de sustenido utilizada neste trabalho é #. Então, temos após a notação da nota, a indicação (ou não) de um meio-tom. As oitavas são descritas como sufixo na notação musical. Por exemplo, para referir-se ao Lá na segunda oitava sustenido, têm-se: A#2.

Uma guitarra possui várias partes que compõem o instrumento, uma delas é o que chamamos de braço. No braço, existem combinações de corda e trastes e, cada combinação produz uma nota contendo, comumente, 6 cordas e de 20 a 24 trastes. Um aspecto importante nessa composição técnica é que uma mesma nota pode ser reproduzida em mais de uma posição no braço do instrumento tornando, portanto, desafiadora a otimização da seleção de posição que melhor se adeque à melodia que se quer tocar.

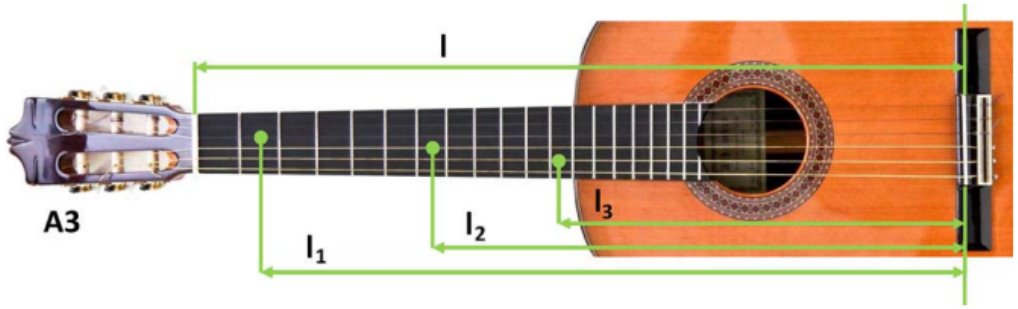

Fig.1: Três possibilidades de se tocar a nota A3. **Fonte:** [6]

A tablatura para guitarra e violão possui seis linhas, as quais indicam cada corda, sendo que a corda mais grossa (E2) é representada pela linha inferior e a mais aguda (E4) pela linha superior. Os números indicados representam os trastes a serem tocados sendo que o traste 0 (zero) representa uma corda solta, sendo tocada sem nenhum traste pressionado.

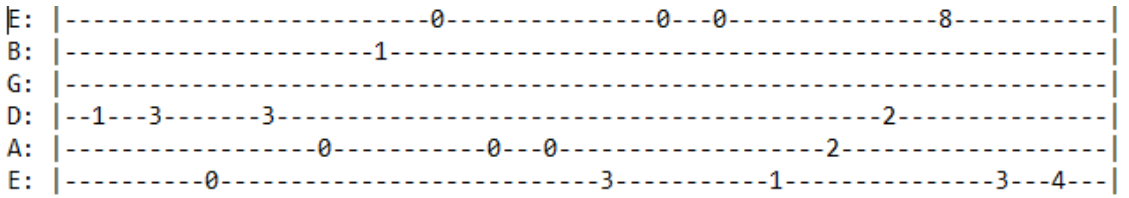

Fig.2: Exemplo de tablatura para guitarra e violão. **Fonte autoral.**

É importante esclarecer também o uso dos termos frequência fundamental e *pitch*, os quais, por mais que utilizados intercambiavelmente na literatura, possuem diferenças. A frequência fundamental é o aspecto físico do sinal enquanto o *pitch* é a percepção da altura tonal de uma nota musical pela audição humana.

2.2 Escala Mel

A audição humana não percebe eventos sonoros de forma linear, mas de uma forma altamente sofisticada dos sons. A escala Mel [10] (de *Melody*) é uma das escalas psicoacústicas que buscam trazer essa similaridade não linear do ouvido humano.

Esta escala logarítmica é utilizada amplamente em áreas como engenharia acústica, processamento de sinais, processamento de linguagem natural e estudos psicoacústicos que envolvem música e fala pelo seu ótimo ajuste à percepção humana, altura tonal e a facilidade de conversão matemática para demais escalas.

Como citado por Rafał Rolczyński (2019, p.8) “Em 1940, Stevens e Volkman atribuíram 1000 mels como 1000 Hz e pediram que participantes variassem a frequência até que percebessem que o pitch estivesse mudado em proporção com a referência. Os limiares das frequências foram marcados, resultando em um mapeamento entre a frequência real (em 8 Hz) e a escala de frequência percebida (em Mel).”

A escala Mel utiliza do fato de que a percepção humana é mais sensível à variações de *pitch* em frequências mais baixas, tornando os intervalos de percepção maiores para frequências mais altas [10].

A conversão de frequências Hertz para Mel é dada pela equação (1) cuja relação é ilustrada pelo gráfico da Figura 3.

$$Fmel = 1125 \cdot ln(1 + Fhz \div 700) \quad \textbf{(1)}$$

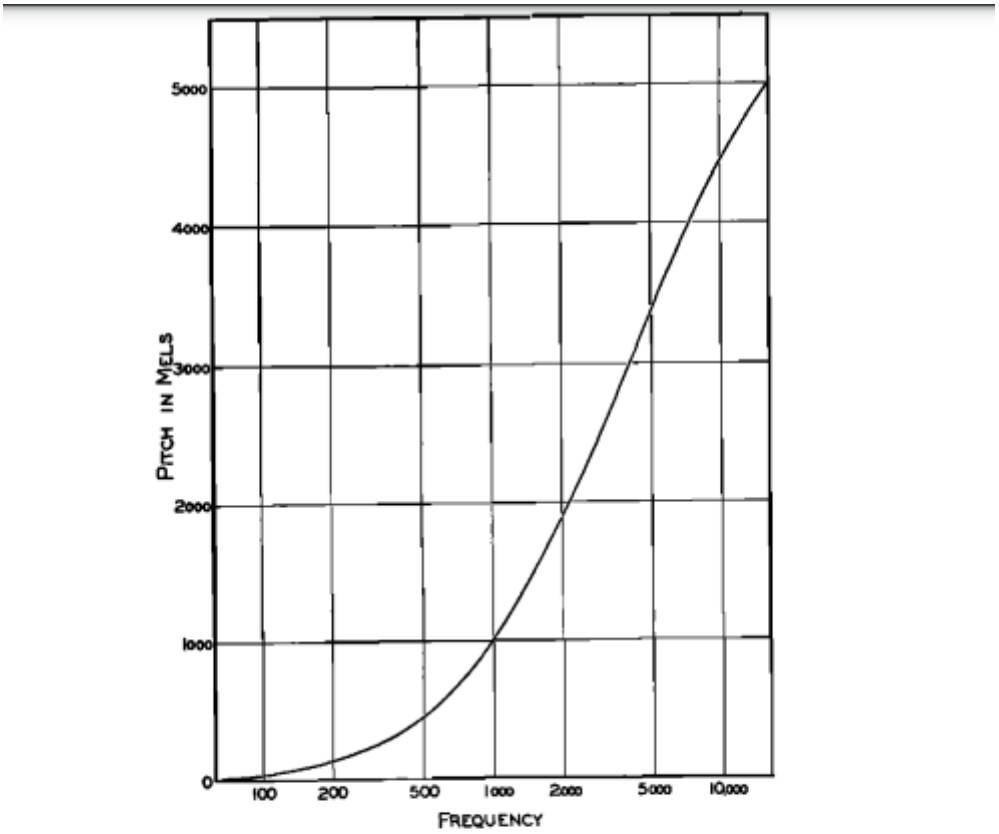

Fig. 3 - Relação escala psicoacústica Mel e frequências em Hertz. **Fonte:** [10]

2.3 *Filter Banks*

Filter Banks compõem uma coleção de filtros triangulares, podendo ser dos tipos Mel, Bark ou Constant-Q, entre outros. Neste trabalho, o *Filter Bank* está em acordo com a escala Mel [11], espaçados de forma a imitar a percepção humana de frequências. *Filter bank*s são compostos de M filtros, quantidade que define a resolução.

A construção de Filter banks se dá pela equação (2) [11], em que:

- Hm representa a magnitude do filtro de *index* m;
- k representa a frequência;
- f representa o vetor M + 2 valores de borda de filtros linearmente espaçados

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k < f(m) \\ 1 & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) < k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (2)$$

Na figura 4, vemos a construção gráfica dos filtros triangulares. Os pontos da base encontram-se na mediana dos triângulos vizinhos, os quais são espaçados de acordo com a escala Mel e a sua quantidade depende do parâmetro M.

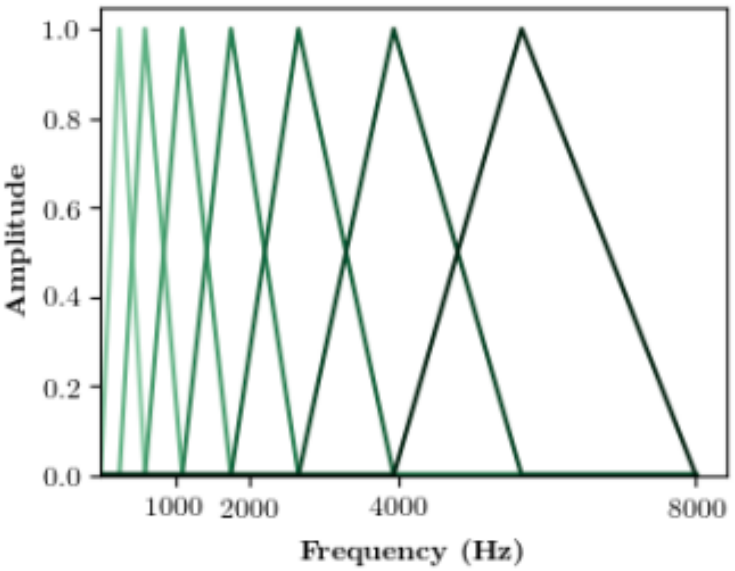

Fig. 4: Gráfico Frequência (Hz) versus Amplitude para filtros triangulares Filter Banks.
Fonte: [11]

2.4 Detecção de *Onset*: *Super Flux*

Processos de detecção de *onset* são essenciais para aplicações de transcrição de notas em áudios, já que são utilizados para a identificação do início de eventos sonoros importantes - como ataque, início de fala e notas musicais entre outros - pela diferença na magnitude, ângulo de fase, espectro complexo entre outras características. Nesta etapa, o foco é a identificação do início das notas musicais tocadas em um áudio de modo a segmentá-las para análises posteriores.

Para este trabalho utilizamos o método *Super Flux*, um método amplamente utilizado na comunidade de processamento de sinais de áudio para detecção de onsets. Trata-se de uma versão melhorada do algoritmo *Spectral Flux* que, segundo Böck, Sebastian & Widmer, Gerhard (2013, p.1), "reduz o número de detecções falso positivo pelo rastreamento de trajetórias espectrais com um filtro de máximo. Especialmente para músicas com muito uso de *vibrato*, por exemplo em óperas e performance de cordas" [12].

Este método utiliza uma abordagem baseada no espectro do sinal que, em geral, entrega resultados mais estáveis e qualitativamente melhores que demais abordagens citadas - entre elas, espectro complexo, fase, energia [13].

O funcionamento do método *Super Flux* detecta mudanças positivas da energia no tempo, similarmente ao método *Spectral* Flux. Este último compara a diferença de um mesmo bin do quadro anterior. Entretanto, em sua versão melhorada - *Super Flux* - é incluído um estágio especial de rastreamento de trajetória utilizando um filtro de máximo de forma computacionalmente eficiente e efetiva.

2.4.1 Rastreamento de trajetória

Neste estágio, o sinal é processado quadro a quadro detectando a diferença no decorrer da trajetória. A partir disso, o sinal é dividido em segmentos sobrepostos com comprimento de 2048 amostras. Uma janela do tipo Hann com mesmo comprimento é aplicada a esses segmentos e em seguida, a Transformada Discreta de Fourier converte-os ao domínio espectral.

O espectrograma de magnitude é filtrado com uso de Filter Banks para simplificação do processo de rastreamento de trajetória. Assim, trabalhando em uma escala logarítmica de frequência, mantém-se constante o espaçamento entre frequências. Segundo Sebastian Bock, Florian Krebs e Markus Schedl (2012, p.50) [14], "O uso de magnitudes logarítmicas, ao invés de representações lineares, entregam melhores resultados em diversos casos, independentemente da ODF (função de detecção de onset) utilizada.". Este processo melhora componentes espectrais fracos [15]. O Filter Banks pode ser do tipo Mel, Bark entre outros, que entregam desempenho semelhante [14]. Neste trabalho, o tipo Mel é utilizado.

De acordo com Sebastian Bock, Florian Krebs e Markus Schedl (2012, p.51) [14], para o método Super Flux que será utilizado, "investigamos diferentes tipos de bancos de filtros (Mel, Bark, Constant-Q) e descobrimos que todos eles têm melhor desempenho do que o *Spectral Flux* padrão". Além disso, constatam que todos performam similarmente, e continua: "Foi constatada a vantagem de filtrar o espectrograma a priori e, só então, tomar o logaritmo da magnitude somada (filtrada)". Para o caso deste trabalho, a seguinte ordem é feita: primeiro, filtra-se com *Filter Banks* utilizando escala Mel para, assim, tomar o logaritmo do espectrograma.

Um processo de rastreamento de trajetória simples analisa a trajetória de cada um dos blocos de frequência m retrocedendo no tempo em μ intervalos de tempo. É calculada a diferença de cada bloco com respeito à magnitude ao longo da trajetória. Entretanto, essa abordagem tem altos custos computacionais e, por isso, uma abordagem utilizando filtro de máximo foi aplicada.

2.4.2 Filtro de máximo

Filtro de máximo, um tipo de filtro bastante utilizado em aplicações de visão computacional e processamento de imagem, aloca o maior valor dentro de uma determinada janela (a qual é definida pelo contorno do filtro) e substitui a posição central com esse valor, ampliando assim a trajetória no eixo de frequência. O contorno

do filtro é definido para cobrir os vizinhos diretos do *bin* de frequência atual e é limitado ao quadro de tempo atual. A largura do filtro é escolhida para atingir um equilíbrio entre a redução de picos espúrios e a preservação de picos reais. Na figura 5 ilustra-se o rastreamento de trajetória com o uso de Filtros de máximo.

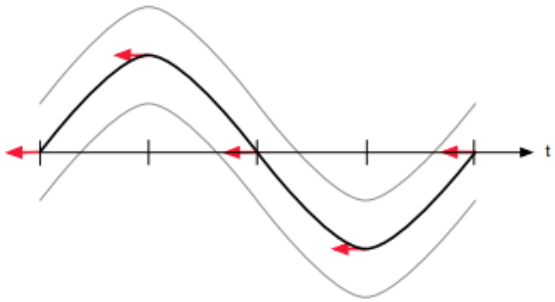

Fig. 5: Ilustração de rastreamento de trajetória com uso de Filtros de Máximo **Fonte:** [12]

Sendo Xlog,filt(n, m) o espectrograma filtrado escalado anteriormente, o espectrograma filtrado pelo Filtro de máximo é dado pela equação (3) [12].

$$Xmax\,log, filt\,(n, m) \;=\; max(Xlog, filt\,(n, m \;-\; 1 : m \;+\; 1)) \quad (3)$$

Por fim, a diferença é então calculada com respeito a esse espectrograma filtrado pelo filtro de máximo, de acordo com a equação (4):

$$SF\;*\;(n) \;=\; \sum_{m=1}^{m=N} H\,(Xlog,\; filt\,(n, m) \;-\; Xmax\,log, filt\,(n \;-\; \mu, m)) \quad (4)$$

A figura 6 mostra a diferença positiva da abordagem usando filtro de máximo (linha em negrito) e uma abordagem clássica sem o filtro de máximo (linha pontilhada).

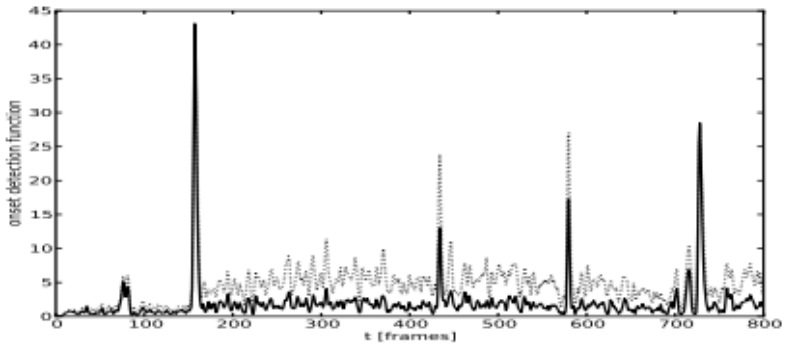

Fig. 6: Ilustração comparativa da soma positiva de métodos com e sem filtro de máximo **Fonte:** [12]

2.4.3 Detecção de Picos

A Detecção de Picos é o processo de seleção de onsets finais no método *Super Flux*, no qual seleciona-se todos os máximos locais, em uma função de detecção de limiar, como posições finais de onset, como explicado em [12] e [14].

2.5 Identificação de Frequência Fundamental: *PYIN*

A identificação de notas em um áudio é feita por um processo de identificação de frequência fundamental (comumente chamada de *pitch*). São inúmeros métodos que trazem essa solução e normalmente são focados em áudios monofônicos (em que há uma sequência de notas únicas tocada por apenas um instrumentos) e áudios polifônicos (que envolvem mais de uma nota sendo tocada simultaneamente na presença de diversos instrumentos). Como este trabalho envolve a utilização de áudios monofônicos, para suprir essa tarefa, um dos métodos escolhidos foi o *Probabilistic YIN*.

Esse método é uma modificação do conhecido algoritmo *YIN* [16] trazendo a adição de uma interpretação probabilística para os possíveis candidatos de frequência fundamental encontrados, resultando em uma precisão e revocação superiores.
A diferença no cálculo entre ambos os métodos é que em *PYIN* (*Probabilistic Yin*) a análise quadro a quadro encontra diversas possibilidades de resposta com uma probabilidade associada e guarda consigo essas informações [17].

Como concluíram Matthias Mauch e Simon Dixon (2014, p. 4) "(...)Nós demonstramos que o *PYIN* possui uma precisão e revocação superiores ao *YIN* com um banco de dados de mais de 30 horas de áudios sintetizados. Já que *PYIN* é parametrizado por uma distribuição de limiar *YIN*, o algoritmo é mais robusto à escolha de distribuição do que *YIN* é para com escolha de limiar."

Em sequência, um Modelo Oculto de Markov (HMM) é empregado para encontrar o *pitch*, mantendo a acurácia do método *YIN* e aumentando a precisão e a revocação.

O processo então se divide em cinco etapas:
I - O algoritmo baseia-se no entendimento de que a diferença dada pela equação (5) [17] considerando um sinalxi,i = 1,.....,2W será pequena se houver alguma periodicidade aproximada com uma frequência fundamental f0.

$$dt(\tau) = \sum_{j=1}^{W} (xj - xj + \tau)^2, \quad (5)$$

Essa diferença pode ser calculada em termos de uma função de autocorrelação (ACF) como demonstrada pela equação (6) [17]:

$$rt(\tau) = \sum_{j=t+1}^{t+W} xj \cdot xj + \tau, (6)$$

a partir da qual pode-se descrever a equação (5) em em termos da equação (6) como demonstrada pela equação (7) [17].

$$dt(\tau) = rt(0) + rt + \tau(0) - 2 \cdot rt(\tau) \quad (7)$$

II - A próxima etapa é normalizar essa diferença em (1) para se obter uma "função de diferença normalizada de média acumulada"

III - O próximo passo é estimar a frequência fundamental. Isso é feito escolhendo o menor período τ para o qual d' tem um mínimo local e d'(τ) < s para um limite fixo s (geralmente s = 0,1 ou s = 0,15). Essa etapa segue os mesmos passos do método original YIN, com exceção de, ao final, fazer uso de uma distribuição de probabilidade (ao invés de um limiar único). No caso, ao invés da escolha de s absoluto, usa-se uma distribuição de parâmetros S dados por P(si) onde si, i = 1, ..., N são possíveis limiares. As distribuições de probabilidade utilizadas são distribuições Beta com médias 0.1, 0.15, 0.2 (α = 1 e β = 18, 11 (1/ 3) , 8), como mostrado no gráfico da Figura 7 para as distribuições beta, como descrito em [17].

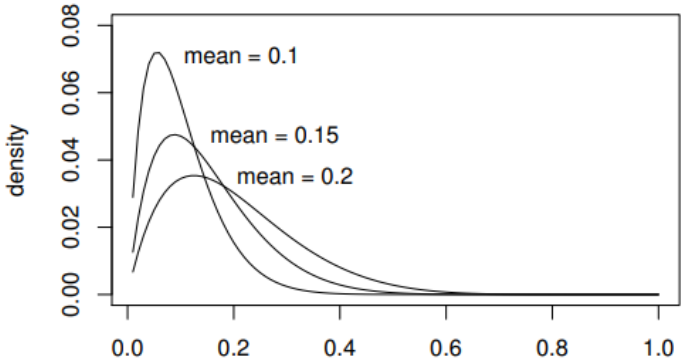

Fig. 7: Distribuição de probabilidade Beta utilizada no método *PYIN* **Fonte:** [17]

Na figura 8, é mostrado o fluxograma comparativo dos métodos *Probabilistic YIN* e *YIN* original em que se destoam na etapa em que o primeiro utiliza uma distribuição de probabilidade para melhorar seu resultado (além de outras etapas posteriores não mencionadas na ilustração da figura 8) etapa da qual herda o nome.

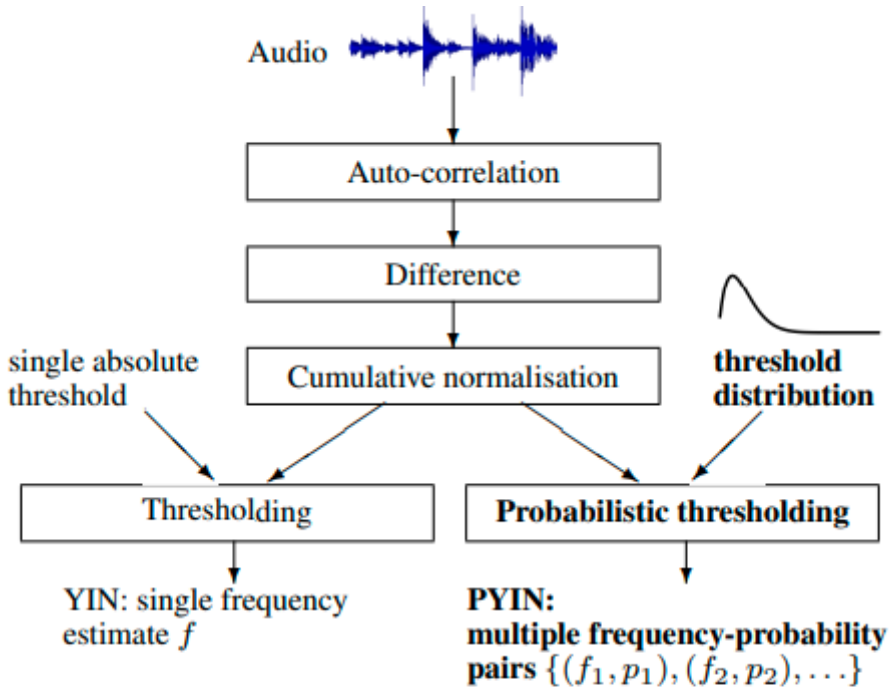

Fig. 8: Ilustração comparativa das etapas de *PYIN* e *YIN* **Fonte:** [17]

II - HMM para selecionar os candidatos

Nesta etapa é selecionado ao menos um candidato de *pitch* por quadro considerando suas probabilidades associadas para atribuir ao bloco de frequência estimada mais próxima. Cada pitch é relacionado a um estado em um Modelo Oculto de Markov (um modelo estatístico em que se busca determinar parâmetros desconhecidos a partir de parâmetros observáveis). Esse modelo então utiliza as probabilidades de cada candidato das etapas anteriores como parâmetros observáveis, como explicado em [17]. O modelo HMM é útil para capturar a suavidade temporal das frequências quadro a quadro.

2.6 Identificação de Frequência Fundamental: *CREPE*

Métodos para identificação de frequência fundamental baseados em redes neurais têm ganhado espaço no estado da arte pelos ótimos resultados que têm entregado. Um dos métodos utilizados deste tipo é o *CREPE*, um algoritmo de redes neurais convolucionais profundas que faz suas análises pelo domínio do tempo [18].

Segundo Jong Wook Kim , Justin Salamon , Peter Li , Juan Pablo Bello (2018 p.1,) [18] “de acordo com diversos estudos comparativos, o estado da arte é alcançado pelos métodos baseados em *YIN*, sendo *PYIN* o de melhor performance até o momento”. Porém, o *CREPE* traz uma nova proposta atingindo resultados que se comparam e até, em alguns casos, ultrapassam o *PYIN*.

Em seu procedimento, a rede neural recebe trechos de 1024 amostras no tempo do sinal em análise, utilizando uma taxa de amostragem de 16kHz. Na formação da rede, há seis camadas convolucionais resultando em 2048 dimensões de representação latente, as quais são então conectadas densamente à camada de resultado com ativações tipo Sigmoid, resultando em um vetor de saída de dimensão de 360. A partir disso, é calculado o *pitch* deterministicamente.

Embora as frequências sejam contínuas ao longo do tempo, o modelo *CREPE* estima a frequência de cada quadro (ou frame) de áudio independentemente, sem usar nenhuma informação temporal de quadros anteriores. Por outro lado, o modelo *pYIN* usa um modelo HMM para capturar a suavidade temporal das frequências. A figura 8 ilustra a arquitetura do método *CREPE Pitch Tracker*.

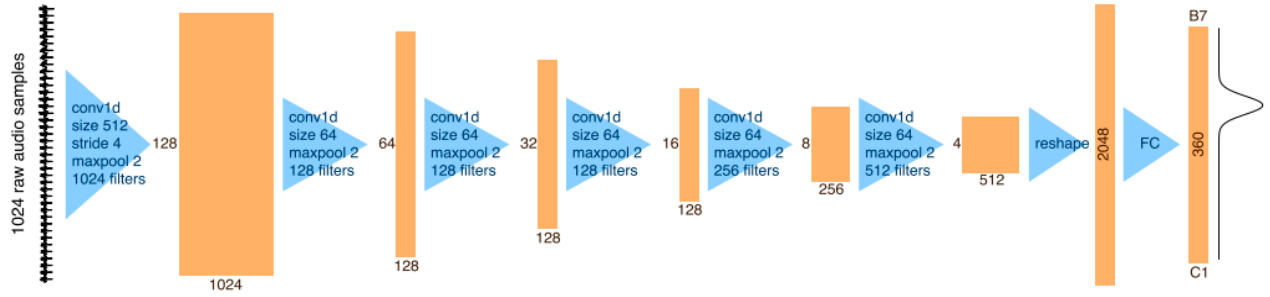

Fig. 9: Ilustração da arquitetura do método *CREPE pitch tracker*. **Fonte:** [18]

2.7 Algoritmo A-Estrela

Para encontrar a melhor posição para se tocar o conjunto de notas identificadas pela aplicação, faz-se uso do algoritmo A-Estrela. Como explicado anteriormente, a escolha de posições no braço da guitarra e/ou violão para se tocar determinadas notas não é uma tarefa simples pois uma mesma nota pode ser tocada de diversas formas. Um algoritmo de otimização é então necessário. Outras técnicas podem ser utilizadas para trazer essa solução como por exemplo a Inarmonicidade, como utilizado em [5] e [6], que utiliza informações de harmônicas do sinal da nota para identificar a posição tocada, levando em conta aspectos físicos específicos do instrumento e cordas como tensão, material, medidas. Esse método é mais complexo e foge do escopo atual deste trabalho, podendo ser uma atualização futura.

A-Estrela é um algoritmo de Busca de Caminho baseado em grafos eficiente e bastante utilizado em diversas áreas como robótica, jogos e inteligência artificial. Possui fundamentos nos algoritmos *Breadth First Search* e *Dijkstra*. Em linhas gerais, o algoritmo busca encontrar o caminho mais curto entre dois pontos.

Aplicando ao cenário de transcrição de notas para tablatura, uma vez identificadas notas, é necessário atribuí-las em posições específicas no instrumento, dentre de todas as possibilidades, de modo que a distância entre elas seja a mais otimizada, ou seja, com menor distância total, respeitando a ordem em que são tocadas.

O algoritmo de busca trabalha com o que é chamado de Nós. Cada nó é um possível caminho para se chegar ao destino. Cada nó possui seus vizinhos diretos entre os quais é escolhido o nó seguinte, buscando a menor distância total. No âmbito deste trabalho, cada possibilidade de nó é uma nota identificada e, a vizinhança de cada nó são as possibilidades da nota seguinte, até que se complete a melodia na última nota. Cada nó, representando uma nota, possui o conjunto Corda-Traste indicando a posição no instrumento como (3,2). Esse conjunto de dimensão representando a posição é utilizado pelo algoritmo calculando distâncias por meio da distância Euclidiana.

Para adaptação neste trabalho: como múltiplos conjuntos de posições são possíveis, dois nós extras são adicionados aos nós das notas, um ao início e outro ao final, chamados de INIT e END neste trabalho. Esses dois nós extras, os quais possuem posições variáveis e flexíveis, permitem ajustar a posição das notas no instrumento. Por exemplo, adicionando os pontos INIT e END na extrema esquerda, direita ou em posições centrais do instrumento, é possível manipular a posição das notas da melodia para que se encontrem em uma determinada região escolhida.

A figura 10 ilustra o funcionamento do algoritmo A-Estrela aplicada ao cenário deste trabalho:

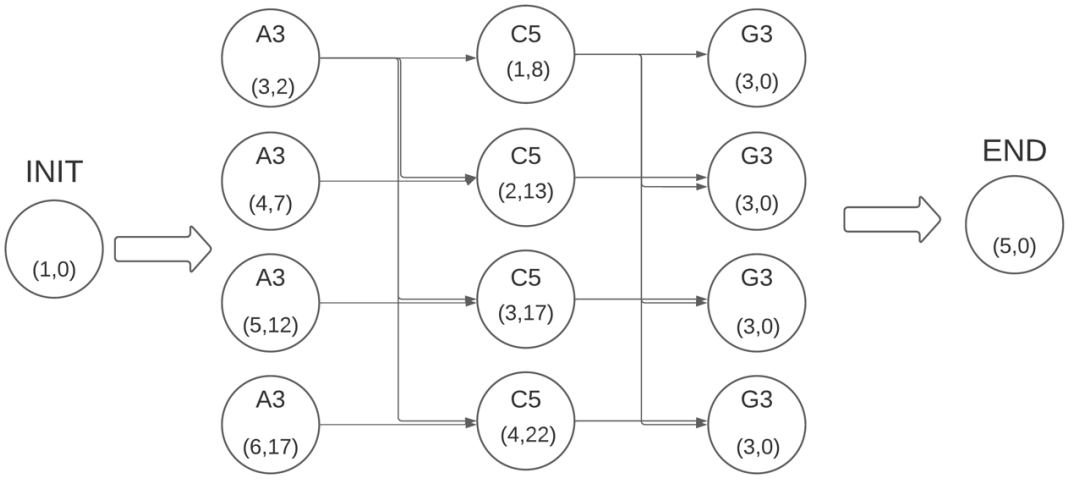

Fig. 10. Diagrama do algoritmo A-Estrela. **Fonte autoral**

No exemplo apresentado, os pontos INIT e END estão na corda 1, traste 0 (corda solta) e corda 5 traste 0, posições que se encontram à esquerda do instrumento (em posição destro), adaptando a escolha das posições das notas da música para essa região.

A aplicação faz uso de um mapeamento das posições de todas as notas encontradas no braço do instrumento, admitindo-se uma afinação padrão. Uma vez identificadas as notas no áudio em análise, o mapeamento traz as possibilidades de posição dessa nota, as quais alimentam o algoritmo A-Estrela como nós.

Capítulo 3 - Metodologia

Neste capítulo são descritos as bibliotecas utilizadas na aplicação, o dataset para teste, apresentado a arquitetura da aplicação, métricas de resultados e estratégias para os estudos e testes.

3.1 Bibliotecas

- **Librosa load** [19]:

  Para carregar arquivos de áudio em uma série de pontos flutuantes de 32 bits no tempo. Recebe como parâmetros:

  path: Caminho do arquivo de áudio de interesse para ser carregado.

  sr: Frequência de Amostragem. Foi utilizado valores de 22050 ou 44100, a depender da frequência de amostragem original de gravação do arquivo de áudio ou optando-se pela frequência de maior resolução (44100).

  mono: Converter sinal para Mono. Foi deixado no padrão para a conversão ocorrer.

  offset: Ponto de interesse no tempo (em segundos) para iniciar-se o áudio. Caso não indicado, começa-se em zero.

  duration: Duração do trecho de interesse do áudio, a partir do offset indicado. Caso não indicado, utiliza-se a duração total do áudio a partir do offset.

  dtype: Tipo do retorno dos pontos no tempo. Utilizado pontos flutuantes de 32 bits (Float 32).

  res_type: Tipo de reamostragem. Foi utilizado o valor de *'kaiser_best'* para melhor qualidade em todos os testes, referenciando a janela *Kaiser*.

- **Librosa pyin** [20] :

  Implementa método de identificação de frequência fundamental *Probabilistic Yin*. Foram utilizado os parâmetros:

  y: Pontos no tempo do áudio.

  fmin: Mínima frequência (Hertz) de interesse. Para o caso do instrumento guitarra e violão, foi utilizado a menor frequência encontrada, 82.0Hz - nota E2 (Mi, segunda oitava).

  fmax: Maior frequência (Hertz) de interesse. Para o caso do instrumento guitarra e violão, foi utilizado a menor frequência encontrada, 2093.0Hz - nota C6 (Dó, sexta oitava).

  sr: Frequência de Amostragem. Foi utilizado o mesmo valor utilizado na etapa de carregamento do sinal de áudio.

frame_length: Tamanho do quadro em amostras. Valores variados a depender do sinal utilizando técnica de variação dinâmica, como explicado em capítulos seguintes.

win_length: Tamanho da Janela em amostras. Valores variados a depender do sinal utilizando técnica de variação dinâmica, como explicado em capítulos seguintes.

hop_length: Comprimento do deslocamento/salto em amostras. Valores variados a depender do sinal utilizando técnica de variação dinâmica, como explicado em capítulos seguintes.

n_thresholds: Número de limiares para estimação de picos. Foi utilizado o valor padrão recomendado pelo método de 100.

beta_parameters: Valores para a distribuição beta. Foi utilizado o padrão recomendado pelo método de (2,18).

boltzmann_parameter: Valores para a distribuição Boltzmann. Foi utilizado o valor padrão 2.

resolution: Resolução dos bins de altura (frequência) em cents (unidade relativa de altura musical). Foi utilizado o padrão recomendado pelo método de 0.01.

max_transition_rate: Taxa de transição máxima de pitch em oitavas por segundo. Foi utilizado o padrão recomendado pelo método de 35.92.

switch_prob: Probabilidade de transição de sons vozeados para não vozeados, ou vice-versa. Foi utilizado o padrão recomendado pelo método de 0.01.

no_trough_prob: Probabilidade máxima a ser adicionada ao mínimo global se não houver depressão abaixo do limite. Foi utilizado o padrão recomendado pelo método de 0.01.

center: Modo de centralização do quadro a ser preenchido de zeros. Foi utilizado o padrão recomendado pelo método fazendo com que o sinal y seja preenchido de modo que o quadro D[:, t] esteja centralizado em y[t * hop_length].

pad_mode: Modo para preenchimento de zeros. Foi utilizado o padrão recomendado pelo método fazendo com que o sinal seja preenchido com zeros pelas ambas extremidades.

fill_na: Valor padrão para sons não vozeados. Foi utilizado o padrão recomendado pelo método fazendo com que a melhor estimativa possível seja alocada ao valor.

- **Librosa onset_strength** [21]:

Implementa método de detecção de *onset Super Flux*. Utiliza os parâmetros:

S: Espectrograma pré computado do sinal em análise.

sr: Frequência de Amostragem. Foi utilizado o mesmo valor utilizado na etapa de carregamento do sinal de áudio.

hop_length: Comprimento do deslocamento/salto em amostras.

lag: Atraso de tempo para calcular diferenças.

max_size: Tamanho (em *bins* de frequência) do filtro de máximo local.

- **Librosa onset_detect** [22]:

Implementa método de reconhecimento de picos a partir do envelope de *onset* com os seguintes parâmetros:

onset_envelope: Envelope do *onset* pré calculado (retornado do método onset_strength).

sr: Frequência de Amostragem. Foi utilizado o mesmo valor utilizado na etapa de carregamento do sinal de áudio.

hop_length: Comprimento do deslocamento/salto em amostras.

units: Unidade para retorno. Pode ser em quadros, amostras ou tempo. Não influencia na resposta. Todos os tipos foram utilizados no escopo deste trabalho para análise.

- **Librosa.feature.melspectrogram** [23]:

Calcula espectrograma Mel com os seguintes parâmetros:

y: Pontos no tempo do áudio.

sr: Frequência de Amostragem. Foi utilizado o mesmo valor utilizado na etapa de carregamento do sinal de áudio.

n_fft: Tamanho do quadro em amostras (para FFT). Valores variados a depender do sinal utilizando técnica de variação dinâmica, como explicado em capítulos seguintes.

hop_length: Comprimento do deslocamento/salto em amostras.

fmin: Mínima frequência (Hertz) de interesse. Para o caso do instrumento guitarra e violão, foi utilizado a menor frequência encontrada, 82.0Hz - nota E2 (Mi, segunda oitava).

fmax: Maior frequência (Hertz) de interesse. Para o caso do instrumento guitarra e violão, foi utilizado a menor frequência encontrada, 2093.0Hz - nota C6 (Dó, sexta oitava).

n_mels: Número de Mels *bands*.

win_length: Tamanho da Janela em amostras. Valores variados a depender do sinal utilizando técnica de variação dinâmica, como explicado em capítulos seguintes.

window: Tipo da janela. Utilizando o valor padrão de Janela Hann.

center: Modo de centralização do quadro a ser preenchido de zeros. Foi utilizado o padrão recomendado pelo método fazendo com que o sinal y seja preenchido de modo que o quadro D[:, t] esteja centralizado em y[t * hop_length].

pad_mode: Modo para preenchimento de zeros. Foi utilizado o padrão recomendado pelo método fazendo com que o sinal seja preenchido com zeros pelas ambas extremidades.

power: Exponente para a magnitude do espectrograma. No caso, utilizando o padrão de 2, ou seja, potência.

- **Librosa.power_to_db** [24]:

  Conversão de espectrograma de potência para dB.

  Recebe apenas a sequência de dados do áudio em potência para converter para dB.

- **CREPE** [25]:

  Implementa método *CREPE pitch tracker*.

  Recebe os parâmetros:

  sequência de dados do áudio no domínio do tempo,

  sr: frequência de amostragem do sinal,

  step_size: Tamanho do salto no tempo em milisegundos

  viterbi: para ativar técnica de suavização ou filtragem usada em processamento de sinais e reconhecimento de padrões, chamada *Viterbi smoothing*, a qual pode ajudar a reduzir os erros de decodificação e melhorar a precisão da sequência reconstruída. Nos testes estamos utilizando essa técnica.

  model_capacity: Velocidade computacional. Nos testes, utilizamos a capacidade máxima.

3.2 *Dataset*

O presente trabalho envolve o estudo de métodos para detecção de *onset* e de frequência fundamental. Tais métodos necessitam de configuração em seus parâmetros de análise como o tamanho do quadro, do passo, da janela, entre outros parâmetros

específicos para cada algoritmo. Para que se encontre esses valores ótimos, é necessário, além do estudo e análise do efeito desses parâmetros no funcionamento dos algoritmos, testes empíricos para análise da performance de cada método com seus variados parâmetros.

Para que seja feito essa análise e seleção de parâmetros, o *Dataset* IDMT-SMT-Guitar foi utilizado [9]. Trata-se de uma grande base de dados para transcrição automática de guitarra criada pelo Instituto Fraunhofer de Tecnologia de Mídias Digitais de Ilmenau, na Alemanha. Com o objetivo de fornecer um conjunto de dados padronizado e de alta qualidade para a pesquisa em reconhecimento de notas de guitarra elétrica e análise de sinal de áudio, esse dataset pode ser usado para treinar e testar algoritmos de detecção de notas, classificação de timbre e outras tarefas relacionadas à análise de áudio de guitarra elétrica.

Foram utilizados conjuntos específicos de áudios exclusivamente monofônicos gravados profissionalmente e que não consideram ruídos, efeitos e técnicas de tocabilidade do instrumento. Para os grupos de dados utilizados 2 e 3, foram gravados com interfaces de áudio *USB Tascam US-1641* e *M-Audio Fast Track Pro*, respectivamente.

A base de dados é dividida em 4 grupos. Para fins de análise e escolha de parâmetros dos métodos utilizados neste trabalho, será utilizado o grupo 2 que consiste de 3 diferentes tipos de guitarras (*Gibson Les Paul, Fender Stratocaster, Aristides 010*) tocando do traste zero (corda solta) até o traste 20 em cada uma das 6 cordas presentes no instrumento.

Esse grupo é particularmente interessante pois permite testar os métodos para as diversas notas tocadas no braço do instrumento em três modelos/marcas diferentes. Isso dá um aspecto amplo para ajuste dos parâmetros visto que a grande maioria das posições e combinações corda-traste que são utilizadas em música estão sendo testadas.

Para os testes finais que envolvem trechos de músicas com ritmos e notas variadas, utiliza-se os *Licks* (trecho de notas) presentes no grupo 2 e trechos de músicas presentes no trecho 3, todos monofônicos sem ruído e gravados profissionalmente.

3.3 Equipamentos e frequências de amostragem dos sinais

A aplicação é voltada para rodar em computadores pessoais e notebooks. Esse tipo de equipamento de uso comum apresenta algumas limitações em termos de taxas de amostragem dos sinais. Porém, a frequência de amostragem não precisa ultrapassar 44100 hertz, sendo um valor ótimo entre 22050 e 44100 (já que a audição humana capta sons entre 20 Hertz e 20 kilo Hertz). Em geral, esta faixa de frequência de amostragem é suportada pelos dispositivos e suficiente para a eficiência do sistema. Neste trabalho, foram utilizados áudios contidos no dataset IDMT-SMT-Guitar, os quais possuem frequência de amostragem máxima de 44100.

3.4 Arquitetura da aplicação

O sistema de transcrição automática de áudios de guitarra e violão monofônicos para tablatura inicia-se com a entrada do arquivo áudio e é indicado a sua frequência de amostragem.

A primeira etapa, ilustrada na figura 11, concentra-se no carregamento deste arquivo de áudio para valores quantizados de amplitude no tempo como uma sequência de pontos flutuantes 32 bits convertidos pelo método load da biblioteca Librosa. Este método recebe, juntamente ao arquivo de áudio, sua frequência de amostragem e outros parâmetros opcionais como offset e duração, para caso queira se extrair apenas um trecho do áudio para análise.

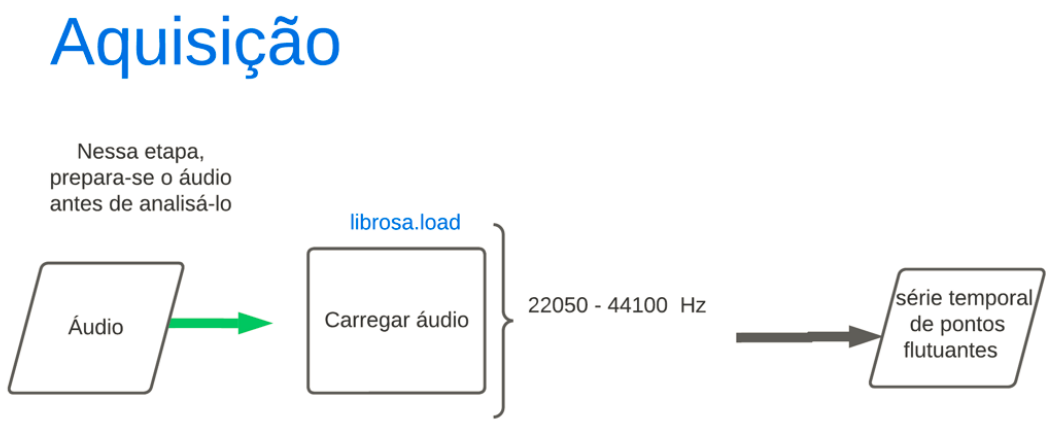

Fig 11. Etapa de aquisição do sinal. **Fonte autoral**

Após o carregamento do arquivo, os respectivos dados são passados para a etapa de Onset, a qual retornará valores de tempo de início e fim de eventos sonoros identificados no áudio, como ilustrado na figura 12. No escopo deste trabalho, esses eventos são as notas musicais tocadas pelo instrumento de forma monofônica.

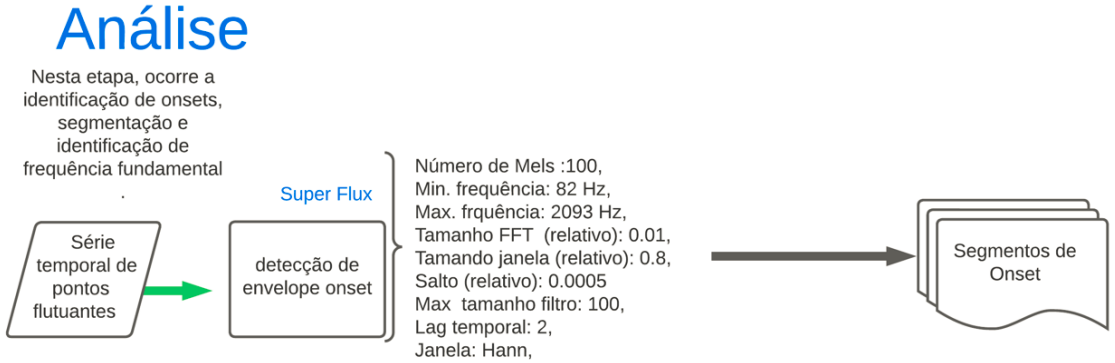

Fig 12. Etapa de análise do sinal para segmentação em Onset. **Fonte autoral**

Em posse desses valores, segmenta-se os dados do áudio, correspondendo ao início e fim dessas notas. Esses segmentos são, assim, passados à próxima etapa de identificação de frequência fundamental. O algoritmo, nesta etapa de identificação ilustrada pela figura 13, faz uso dos dados no domínio do tempo e retorna a nota identificada, informando sua oitava, em forma textual (A3, B4, C5, etc). Para cada segmento de nota, um valor de nota é encontrado e adicionado em uma lista, que é retornada no final, contendo todas as notas, em sequência, encontradas no arquivo de áudio de entrada.

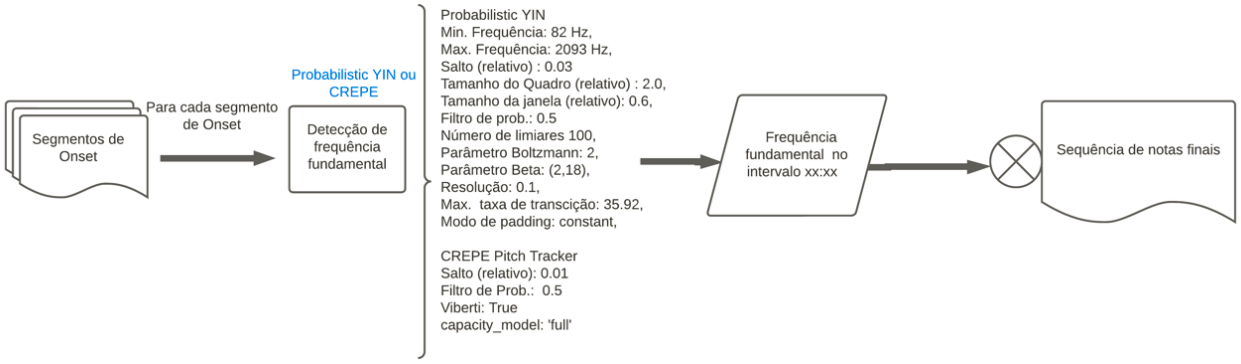

Fig 13. Etapa de análise do sinal para identificação de pitch. **Fonte autoral**

Em posse da lista final, contendo as notas identificadas no áudio, e um mapeamento de notas existentes no braço do instrumento, um algoritmo de otimização de posição é utilizado, como ilustrado pela figura 14. Ele busca encontrar, de forma lógica, as melhores posições para que as notas encontradas sejam tocadas, sendo, o fator de escolha, a proximidade entre elas.

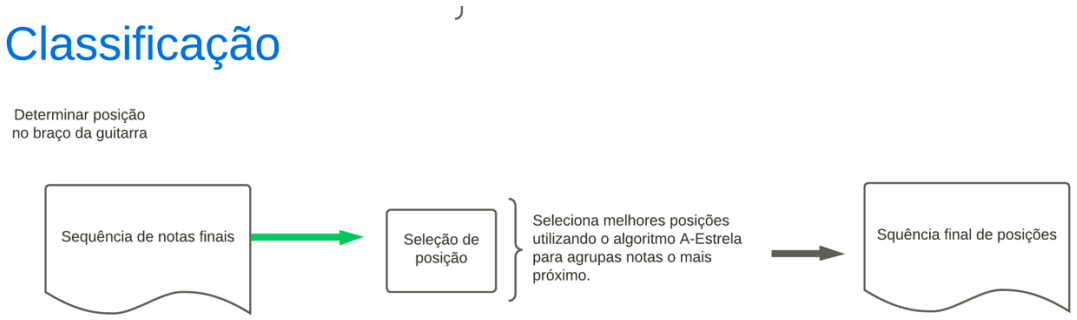

Fig 14. Etapa de classificação do sinal para seleção de posições. **Fonte autoral**

A etapa final retorna, de forma textual ASCII, a tablatura, contendo a posição de dedos para que se toque, no instrumento, as notas identificadas no áudio.

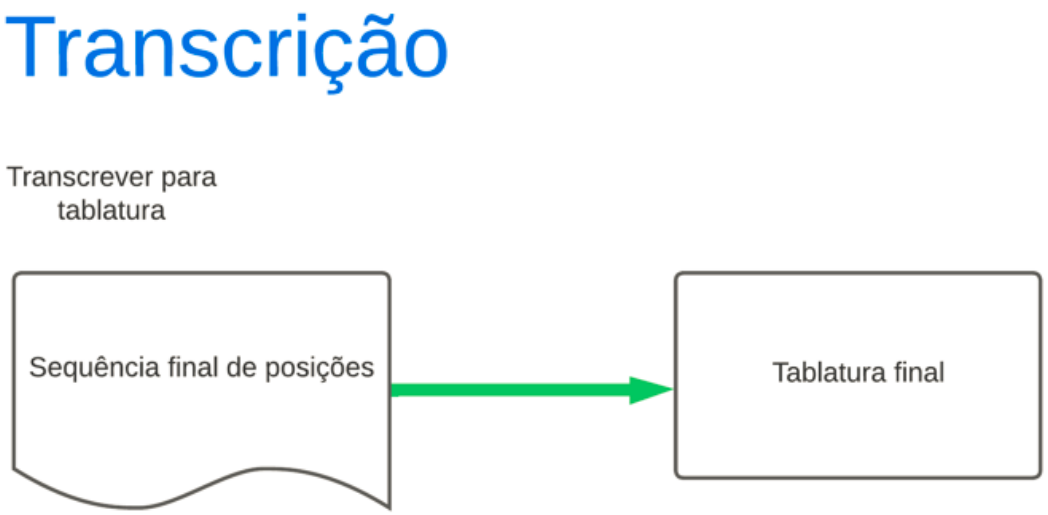

Fig 15. Etapa de transcrição do sinal para tablatura. **Fonte autoral**

Nota-se que não se faz uso de etapas de filtro pois, além de o trabalho de concentrar apenas em áudios monofônicos contidos no dataset IDMT-SMT-Guitar [9], harmônicas de altas frequências são essenciais para os métodos de reconhecimento de frequência que requerem um espectro amplo para análise.

Tratando-se da aplicação técnica do sistema em que foi escrito em linguagem Python, fazendo-se uso do paradigma de Orientação à Objetos, descreve-se o diagrama UML na figura 16.

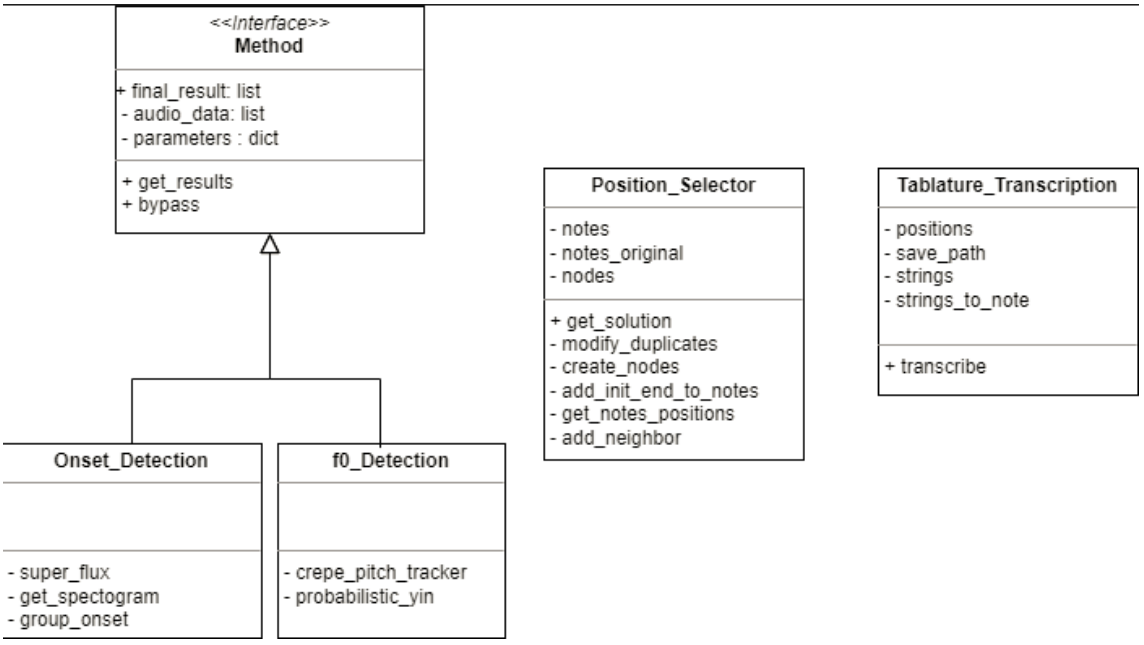

Fig 16. Diagrama UML da aplicação. **Fonte autoral**

Os métodos de detecção de onset e frequência fundamental utilizam uma interface comum *Method.* Os métodos de seleção de posição e transcrição de tablatura são implementados em outras duas classes.

Para a configuração do sistema, um arquivo de configuração (setup.py) é utilizado. Nele, os valores de parâmetros para os métodos de Onset e de identificação de frequência são selecionados (podendo ser *CREPE* ou *Probabilistic YIN*). Desta forma, o usuário tem a flexibilidade de alterar valores de parâmetros à sua vontade, facilitando, também, para testes.

No apêndice deste trabalho, encontra-se a referência para o código completo encontrado no repositório público *GitHub*.

3.5 Métricas de Resultado

Modelos de Machine Learning e sistemas de reconhecimento fazem uso de métricas para avaliar seu desempenho. Métricas como acurácia, precisão, revocação e F-medida foram amplamente utilizadas em estudos semelhantes a esse, sendo bases de comparação com os demais. Além disso, o presente trabalho faz uso da distância de Levenshtein [26], que leva o nome do matemático russo Vladimir Levenshtein (1935-2017) e mede a dissimilaridade entre duas sequências de símbolos/caracteres por meio do número mínimo de operações necessárias para que se iguale uma sequência a outra, por meio das operações de inserção, deleção e substituição de símbolos/caracteres.

Esta técnica é muito útil em áreas como teoria da informação, linguística, machine learning, reconhecimento e bioinformática, como em Khemiri, Houssemeddine & Petrovska-Delacrétaz, Dijana & Chollet, Gerard (2014) e Fiscus, Jonathan & Ajot, Jerome & Radde, Nicolas & Laprun, Christophe (2006), que a utilizam para aplicações de reconhecimento de características em áudios.

No cenário deste trabalho, no âmbito da identificação de onsets e notas, o sistema de identificação agrupa seus resultados em sequências (na linguagem Python listas). Pela naturalidade deste tipo de sistema e, consequentemente, de música e sinais com um range de notas, onsets e posições amplas a serem reconhecidos, o resultado dos métodos podem incluir, erroneamente, notas e onsets a mais ou a menos. Por exemplo, seja uma sequência de referência que represente notas em uma melodia representada por [A, B, C, D]: o sistema de reconhecimento pode trazer, como resposta, a sequência de notas [A, A, B, C, D] na qual se identifica corretamente todas as notas. Porém, há a nota A (Lá), identificada em repetição. Este cenário pode ocorrer, por exemplo, no caso de um onset identificado incorretamente quando, na verdade, há apenas uma continuação da sonoridade desta nota. O presente sistema trata, sim, essa repetição com penalização. Entretanto, se medido esse resultado com formas levianas de acurácia, precisão, revocação e f-medida – além de estarem incompatíveis os comprimentos das sequências em questão – as métricas considerariam apenas a nota A encontrada e as demais seriam ignoradas por estarem posições que destoam da sequência de referência:

Onde espera-se encontrar B, há a nota A repetida. Onde espera-se encontrar C, há a nota B, etc.

Esse motivo claramente requisita o uso de uma forma mais acurada e adaptada para que se possa medir o desempenho deste tipo de sistema de reconhecimento para que assim possa identificar esse tipo de situação, penalizar os erros e saber lidar de forma inteligente, refletindo o desempenho correto pelas métricas. Assim como apresentado em [27], no qual se defende o uso da distância de Levenshtein para o cálculo de métricas de precisão e acurácia, o presente trabalho faz uso da distância de Levenshtein aliada às métricas que serão descritas.

3.5.1 Acurácia

Acurácia é uma medida da proporção de notas identificadas corretamente em relação a uma sequência de referência, avaliando-se a performance global do método.

A seguinte equação (8) é utilizada.

$$Acurácia = 1 - \frac{Dist.\ Levenshtein}{comprimento\ da\ sequência\ de\ referência} \quad (8)$$

3.5.2 Precisão

A precisão mede a proporção de elementos identificados corretamente em relação a todos os elementos previstos, avaliando o quão preciso foi o sistema em identificar as notas corretamente, conforme a equação (9).

$$Precisão = \frac{(comprimento\ da\ sequência\ de\ referência - dist.\ Levenshtein)}{comprimento\ da\ sequência\ de\ teste} \quad (9)$$

3.5.3 Revocação

Revocação mede a proporção de notas identificadas corretamente em relação à sequência de referência conforme a equação (10).

$$Revocação = \frac{(comprimento\ da\ sequência\ de\ referência - dist.\ Levenshtein)}{comprimento\ da\ sequência\ de\ referências} \quad (10)$$

3.5.4 F-Medida

Essa métrica avalia a performance do sistema por meio de uma média harmônica combinando precisão e revocação, conforme a equação (11).

$$F\text{-}Medida = \frac{2 * (precisão * revocação)}{(precisão + revocação)} \quad (11)$$

3.5.5 Distância de Levenshtein

A distância de Levenshtein mede a dissimilaridade entre duas sequências de símbolos/caracteres por meio do número mínimo de operações necessárias para que se iguale uma sequência a outra por meio das operações de inserção, deleção e substituição de símbolos/caracteres.

3.6 Estratégia de seleção de notas

Métodos de identificação de frequência fundamental podem trazer vários resultados para um segmento de áudio em que há apenas uma nota relevante. Isso acontece com relação aos parâmetros de processamento de sinal escolhidos. Por exemplo, a seleção de tamanho de quadro e passo ajusta o sinal em segmentos para análise e cada um desses segmentos podem apresentar alguma frequência que difere das demais, dependendo do quadro e dos métodos de análise. Fatores como frequência de amostragem, número de amostras e tamanho e tipo da janela também afetam o resultado, consideravelmente. Neste sentido, um método de identificação de frequência fundamental pode, ao dividir o segmento de áudio - que possui apenas uma frequência dominante, em quadros, identificar uma ou mais frequências que destoam da esperada. Além dessa identificação, o resultado de métodos apresenta uma probabilidade de certeza a qual, ajustado um limiar, pode ajudar a definir qual frequência realmente está presente em dominância no segmento em análise.

Entretanto, como a segmentação por quadros é feita para cada quadro, pode haver a identificação de uma frequência diferente das demais encontradas. Então, mesmo após definir-se um limiar para a probabilidade, encontra-se um conjunto de várias frequências que, a cada quadro, ultrapassaram o limiar de certeza e podem ser consideradas como a frequência dominante no segmento do áudio em análise.

Para lidar com esta questão, o projeto, além de definir um limiar ótimo para cada método, utiliza uma estratégia de Contagem de Maioria para a seleção da nota dominante no segmento de áudio em análise. Essa estratégia analisa o conjunto de frequências que ultrapassaram o limiar e, a frequência que estiver em maior quantidade nesse conjunto, é escolhida como a frequência dominante que predomina no segmento do áudio, sendo, assim, definida como a nota.

A tabela (1) abaixo traz um exemplo desta análise (exemplo fictício). Admite-se um segmento do áudio que compreende o início e fim de uma única nota **G♯3.** O método de identificação de frequências fundamentais, ao dividir o segmento em quadros

(dado seu início pela coluna “Tempo”), pode identificar frequências distintas em cada um, acompanhadas da probabilidade de certeza nessa identificação.

Utilizando, por exemplo, um limiar definido de 0.5 (50%), filtra-se as candidatas que não atendem a esse pré-requisito de probabilidade. As demais - no caso do exemplo - **G♯3, G♯3 e F3** são submetidas à estratégia de Contagem de Maioria em que a nota que mais aparece neste conjunto é definida como a dominante. No exemplo, a nota definida para o segmento em análise seria **G♯3.**

Tempo	F0	Probabilidade
0.5108390022675737	G♯3	0.8911444948731202
0.5340589569160997	F3	0.7571601330199091
0.5572789115646258	A3	0.2407543944755742
0.5804988662131519	G♯3	0.9848789761259924

Tab. 1 Exemplo de resultado de identificação de pitch por segmento. **Fonte autoral**

3.7 Variação Dinâmica Relativa

Ao invés de testar parâmetros com valores estáticos, o trabalho foca em utilizar valores relativos à quantidade de amostras para estudo da relação entre os parâmetros e essa quantidade, de modo a encontrar uma relação que otimize as métricas de resultado nos testes. Esse tipo de técnica é um recurso comum em aplicações de processamento de sinais e pode trazer vantagens significativas para a análise de sinais complexos, como sinais de áudio, imagens, eletrocardiogramas, entre outros.

Essa abordagem é focada nos parâmetros que indicam o tamanho do quadro, janela e deslocamento de salto no tempo (hop). O tamanho do quadro e o tamanho do deslocamento de salto no tempo são utilizados para dividir o sinal em segmentos para análise. Quando a quantidade de amostras do sinal varia, a escolha desses parâmetros pode afetar significativamente a resolução temporal e frequencial da análise. Portanto, a variação dinâmica desses parâmetros pode ser útil para ajustar a análise de sinal de acordo com a quantidade de amostras disponíveis, para garantir que toda a informação relevante do sinal seja capturada, independentemente da sua duração. E, ainda mais no caso dos métodos de identificação de frequência fundamental, que recebem segmentos com tamanho muito variável de amostras (podendo ser muito ou até menos que o valor da taxa de amostragem), a variação dinâmica pode garantir uma análise precisa e eficiente.

Além disso, a variação dinâmica do tamanho da janela pode ser útil para ajustar a resolução de frequência da análise. A escolha do tamanho da janela pode afetar a resolução de frequência da análise de Fourier e, portanto, a capacidade de detectar eventos em diferentes frequências. A variação dinâmica do tamanho da janela pode permitir a adaptação da resolução de frequência para a aplicação em questão.

Em resumo, a variação dinâmica do tamanho do quadro, deslocamento de salto no tempo e da janela podem trazer vantagens significativas para a análise de sinais em aplicações diversas. Essa técnica permite ajustar a análise de sinal de acordo com a quantidade de amostras disponíveis e a resolução necessária para a aplicação em questão, garantindo uma análise precisa e eficiente de sinais complexos.

Capítulo 4 - Resultados experimentais

Sabe-se que para a escolha otimizada de parâmetros que tragam o melhor resultado, um método empírico se faz necessário, como é de praxe em desenvolvimento e estudo de sistemas de reconhecimento utilizando processamento natural.

Tendo em vista que os métodos analisados neste trabalho são compostos de vários parâmetros que, juntos, formam diversas combinações, faz-se uso de uma abordagem automatizada de testes nos quais varia-se cada parâmetro em intervalos específicos e, a cada iteração, as métricas são calculadas para posterior análise sobre qual combinação traz o melhor desempenho geral.

É importante ressaltar que, por limitações de carga computacional e tempo, todos os valores possíveis não foram explorados. Testa-se primeiramente em intervalos maiores e, com a análise dos resultados, explora-se intervalos menores de interesse afunilando o range de melhor resultado até que se encontre intervalos ótimos.

Pelos testes analisados nos próximos subcapítulos, o método Probabilistic Yin foi escolhido pelo seu maior desempenho quando comparado ao método CREPE para as métricas calculadas.

4.1 Super Flux

Como indicado, o método Super Flux para detecção de Onset é implementado pela biblioteca Librosa utilizando o método onset_strength [21], o método para cálculo do espectrograma Mel [23] e o método Onset_detect [22].

Para verificarmos o sucesso no reconhecimento de *onsets*, coloca-se uma tolerância de 0.5% no valor do tempo identificado para início de um *onset* em relação ao *Onset* de referência do *DataSet* [9] como feito em [5]. Para cálculo das métricas, caso o *Onset* identificado se encontre na tolerância de 0.5%, o Onset estimado é convertido para o binário 1 e, caso contrário, 0. A sequência binária final é então utilizada para o cálculo das métricas de acurácia, precisão, revocação, f-Medida utilizando distância de Levenshtein. Para cada áudio são retornadas métricas individuais. Ao final, faz-se uma média aritmética com os resultados de cada métrica para se ter o valor geral.

Os parâmetros utilizados para esse teste do método Super Flux, os quais foram explicados anteriormente, no subcapítulo sobre as bibliotecas com os métodos librosa.onset_strength e librosa.feature.melspectrogram, são:

- Max Size: Valor absoluto.
- Lag: Valor absoluto.
- Número de Mel: Valor absoluto.
- Tamanho do Deslocamento do salto: Valor relativo ao número de amostras.
- Tamanho da FFT: Valor relativo ao número de amostras.
- Tamanho da Janela: Valor relativo ao tamanho da FFT.

4.1.1 Teste 1

Inicialmente, para o primeiro teste, mantém-se fixo o valor de Max Size e Lag para que se varia os demais parâmetros. Os valores escolhidos, que trouxeram melhores resultados, são convencionais e foram utilizados em outros testes neste trabalho.. Para os demais, foram avaliados o tamanho do deslocamento e tamanho da FFT em potências de 10 para se estudar intervalos. Para o tamanho da FFT, o valor máximo de teste foi 0.1, correspondendo a 10% do número de amostras, valor máximo neste teste que, além de selecionar uma pequena quantidade de quadros, causa muito tempo computacional pelos quadros com grande número de amostras. O tamanho do deslocamento de salto teve seu valor máximo como sendo o mínimo valor de teste para o tamanho da FFT. Os valores de Mel variaram bastante. Quanto maior o valor deste último, maior a resolução, então, procura-se, inicialmente, explorar grandes intervalos. Para o tamanho da janela, escolhe-se valores acima de 80% do tamanho da FFT para que ambos fiquem com valores equivalentes pois, quanto mais estreita a janela, menor sua resolução em frequência e o tamanho da janela não pode exceder o tamanho da FFT.

- Valores Max Size: 200
- Valores Lag: 2
- Valores Tamanho do Deslocamento do salto: 0.000001,0.00001, 0.0001, 0.001
- Valores Número de Mel: 200,600,700
- Valores Tamanho da FFT: 0.001, 0.01,0.05, 0.1
- Valores Tamanho da Janela: 0.8,0.9,1.0

Maiores valores de acurácia:

A tabela 2 mostra os maiores valores de acurácia nos testes e o conjunto de parâmetros que foram utilizados. Observa-se que o tamanho da FFT se consolidou em 0.01 e, além dos parâmetros em valores fixos, outros admitiram valores diferentes para os dois melhores resultados.

Acurácia	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.7777777777777779	200	2	1e-05	200	0.01	0.8
0.7804232804232806	200	2	0.0001	200	0.01	0.8
0.7433862433862434	200	2	0.001	600	0.01	0.9

Tab 2. Teste 1 Super Flux, parâmetros para maior acurácia. **Fonte autoral**

Maiores valores de precisão:

A tabela 3 mostra os maiores valores de precisão nos testes e o conjunto de parâmetros que foram utilizados. Segue-se o mesmo padrão para acurácia.

Precisão	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.7764155943217729	200	2	1e-05	200	0.01	0.8
0.7977774296853245	200	2	0.0001	200	0.01	0.8
0.7612764550264455	200	2	0.0001	600	0.01	0.9

Tab 3. Teste 1 Super Flux, parâmetros para maior precisão. **Fonte autoral**

Maiores valores de revocação:

A tabela 4 mostra os maiores valores de revocação nos testes e o conjunto de parâmetros que foram utilizados. Segue-se o mesmo padrão para acurácia e precisão, mostrando alinhamento nos valores dos parâmetros nos melhores resultados.

Revocação	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.7777777777777779	200	2	1e-05	200	0.01	0.8
0.7804232804232806	200	2	0.0001	200	0.01	0.8
0.7433862433862434	200	2	0.001	600	0.01	0.9

Tab 4. Teste 1 Super Flux, parâmetros para maior revocação. **Fonte autoral**

Maiores valores de F-medida:

A tabela 5 mostra os maiores valores de F-medida nos testes e o conjunto de parâmetros que foram utilizados. Segue-se o mesmo padrão para acurácia, precisão, e revocação mostrando alinhamento nos valores dos parâmetros nos melhores resultados.

F-medida	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.7769008044958813	200	2	1e-05	200	0.01	0.8
0.7883768786207812	200	2	0.0001	200	0.01	0.8
0.7497968351626888	200	2	0.0001	200	0.01	0.9

Tab 5. Teste 1 Super Flux, parâmetros para maior F-medida. **Fonte autoral**

Menores distâncias de Levenshtein:

A tabela 6 mostra os menores valores de distância de Levenshtein nos testes e o conjunto de parâmetros que foram utilizados. Segue-se o mesmo padrão para acurácia, precisão, revocação e F-medida mostrando alinhamento nos valores dos parâmetros nos melhores resultados.

Distância de Levenshtein	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
4.666666666666667	200	2	1e-05	200	0.01	0.8
4.611111111111111	200	2	0.0001	200	0.01	0.8
5.388888888888889	200	2	0.001	600	0.01	0.9

Tab 6. Teste 1 Super Flux, parâmetros para menor distância de Levenshtein. **Fonte autoral**

Para o primeiro lote de testes, é observado que, para todos os casos, os valores relativos à quantidade de amostras para os parâmetros 'Número da FFT' e 'Tamanho da janela' mantiveram-se em 0.01 e 0.8, respectivamente. Para os parâmetros de deslocamento do salto e número de Mel, os maiores resultados de todas as métricas foram para 0.001 e 1 e-5 - para o tamanho do salto, e 200 para o número de Mels.

4.1.2 Teste 2

Em seguida, testa-se para uma seleção de valores de Max Size e Lag, os quais anteriormente foram fixados, além da exploração de outros valores para tamanho do salto, número de Mel e tamanho da Janela, como descrito abaixo:

- Valores Max Size: 10, 50, 100, 200
- Valores Lag: 2, 3
- Valores Tamanho do Deslocamento do salto: 0.0001, 0.0005
- Valores Número de Mel: 100, 200, 400
- Valores Tamanho da FFT: 0.01
- Valores Tamanho da Janela: 0.6, 0.8

Maiores valores de acurácia:

A tabela 7 mostra os maiores valores de acurácia nos testes e o conjunto de parâmetros que foram utilizados. Vemos uma grande variação para Max Size e Número de Mels. Os demais parâmetros de consolidam em seus valores para os melhores resultados.

Acurácia	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8042328042328043	50	2	0.0005	100	0.01	0.8
0.8015873015873016	100	2	0.0005	200	0.01	0.8
0.8015873015873017	200	2	0.0005	400	0.01	0.8

Tab 7. Teste 2 Super Flux, parâmetros para maior acurácia. **Fonte autoral**

Maiores valores de precisão:

A tabela 8 mostra os maiores valores de precisão nos testes e o conjunto de parâmetros que foram utilizados. Vemos uma grande variação para Max Size e Número de Mels. Os demais parâmetros de consolidam em seus valores para os melhores resultados.

Precisão	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8141067538126361	100	2	0.0005	100	0.01	0.8
0.8135776532835356	200	2	0.0005	100	0.01	0.8
0.8134711779448622	100	3	0.0001	100	0.01	0.8
0.8157198551935394	200	3	0.0001	100	0.01	0.8
0.8131261568955069	200	3	0.0001	200	0.01	0.8

Tab 8. Teste 2 Super Flux, parâmetros para maior precisão. **Fonte autoral**

Maiores valores de revocação:

A tabela 9 mostra os maiores valores de revocação nos testes e o conjunto de parâmetros que foram utilizados. Vemos uma grande variação para Max Size e Número de Mels. Os demais parâmetros de consolidam em seus valores para os melhores resultados.

Revocação	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8042328042328042	50	2	0.0005	100	0.01	0.8
0.8015873015873016	100	2	0.0005	200	0.01	0.8

0.8015873015873016	200	2	0.0005	400	0.01	0.8

Tab 9. Teste 2 Super Flux, parâmetros para maior revocação. **Fonte autoral**

Maiores valores de F-medida:

A tabela 10 mostra os maiores valores de F-medida nos testes e o conjunto de parâmetros que foram utilizados. Vemos uma grande variação para Max Size e Número de Mels. Os demais parâmetros de consolidam em seus valores para os melhores resultados.

F-medida	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8072009291521485	50	2	0.0005	100	0.01	0.8
0.8016450339941996	100	2	0.0005	100	0.01	0.8
0.8052651955090978	100	2	0.0005	200	0.01	0.8
0.8059068386918642	200	2	0.0005	400	0.01	0.8

Tab 10. Teste 2 Super Flux, parâmetros para maior F-medida. **Fonte autoral**

Menores distâncias de Levenshtein:

A tabela 11 mostra os menores valores de distância de Levenshtein nos testes e o conjunto de parâmetros que foram utilizados. Vemos uma grande variação para Max Size e Número de Mels. Os demais parâmetros de consolidam em seus valores para os melhores resultados.

Distância de Levenshtein	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
4.111111111111111	50	2	0.0005	100	0.01	0.8

4.166666666666667	100	2	0.0005	200	0.01	0.8
4.166666666666667	200	2	0.0005	400	0.01	0.8

Tab 11. Teste 2 Super Flux, parâmetros para menor distância de Levenshtein. **Fonte autoral**

Uma melhora ultrapassando os 80% é visto para esse segundo teste, superando o primeiro. Além disso, é observada uma discordância entre os valores para Max Size e número de Mel, isto é, nem todas as métricas possuem os mesmos valores para Max Size e número de Mel para seus maiores resultados. É visto que, para ambos parâmetros citados, todos os valores disponíveis neste testes estão presentes para os maiores resultados das métricas, mas não coincidem para todas, tendo como maioria 100 e 200 (para Max Size) e 100 e 200 (para número de Mel). Para os demais parâmetros, observa-se uma preferência clara de 0.005 para o deslocamento de salto, 2 para Lag e 0.8 para tamanho da janela para a maioria dos resultados.

4.1.3 Teste 3

Prossegue-se com um terceiro teste que explora mais valores para Max Size, número de Mel e deslocamento do salto. Os parâmetros Lag, tamanho do FFT e tamanho da janela ficam fixos para seus melhores valores encontrados em todos os testes até então.

- Valores Max Size: 25, 50, 75, 100, 150, 200
- Valores Lag: 2
- Valores Tamanho do Deslocamento do salto: 0.0001, 0.0005
- Valores Número de Mel: 100, 120, 160, 180, 200
- Valores Tamanho da FFT: 0.01
- Valores Tamanho da Janela: 0.8

Maiores valores de acurácia:

A tabela 12 mostra os maiores valores de acurácia nos testes e o conjunto de parâmetros que foram utilizados. Os parâmetros Max Size e Número de Mels admitem vários valores nos melhores resultados, ao passo que os demais parâmetros se consolidam em valores fixos.

Acurácia	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8042328042328043	50	2	0.0005	100	0.01	0.8

0.8015873015873016	75	2	0.0005	180	0.01	0.8
0.8042328042328042	75	2	0.0005	200	0.01	0.8
0.8015873015873016	100	2	0.0005	200	0.01	0.8

Tab 12. Teste 3 Super Flux, parâmetros para maior acurácia. **Fonte autoral**.

Maiores valores de precisão:

A tabela 13 mostra os maiores valores de precisão nos testes e o conjunto de parâmetros que foram utilizados. Os parâmetros Max Size e Número de Mels admitem vários valores nos melhores resultados, ao passo que os demais parâmetros se consolidam em valores fixos.

Precisão	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8141067538126361	100	2	0.0005	100	0.01	0.8
0.8161375661375566	100	2	0.0005	120	0.01	0.8
0.8140885984023237	150	2	0.0005	120	0.01	0.8
0.8140885984023237	200	2	0.0005	120	0.01	0.8

Tab 13. Teste 3 Super Flux, parâmetros para maior precisão. **Fonte autoral**.

Maiores valores de revocação:

A tabela 14 mostra os maiores valores de revocação nos testes e o conjunto de parâmetros que foram utilizados. Os parâmetros Max Size e Número de Mels admitem vários valores nos melhores resultados, ao passo que os demais parâmetros se consolidam em valores fixos.

Revocação	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8042328042328042	50	2	0.0005	100	0.01	0.8
0.8015873015873016	75	2	0.0005	180	0.01	0.8
0.8042328042328042	75	2	0.0005	200	0.01	0.8
0.8015873015873016	100	2	0.0005	200	0.01	0.8

Tab 14. Teste 3 Super Flux, parâmetros para maior revocação. **Fonte autoral**.

Maiores valores de F-medida:

A tabela 15 mostra os maiores valores de F-medida nos testes e o conjunto de parâmetros que foram utilizados. Os parâmetros Max Size e Número de Mels admitem vários valores nos melhores resultados, ao passo que os demais parâmetros se consolidam em valores fixos.

F-medida	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
0.8072009291521485	50	2	0.0005	100	0.01	0.8
0.8070172826270388	100	2	0.0005	120	0.01	0.8
0.8052651955090978	100	2	0.0005	200	0.01	0.8

Tab 15. Teste 3 Super Flux, parâmetros para maior F-medida. **Fonte autoral**.

Menores distâncias de Levenshtein:

A tabela 16 mostra os menores valores de distância de Levenshtein nos testes e o conjunto de parâmetros que foram utilizados. Os parâmetros Max Size e Número de

Mels admitem vários valores nos melhores resultados, ao passo que os demais parâmetros se consolidam em valores fixos.

Distância de Levenshtein	Max Size (absoluto)	Lag (absoluto)	Tamanho Salto (relativo)	Número de mels (absoluto)	Tamanho FFT (relativo)	Tamanho Janela (relativo)
4.111111111111111	50	2	0.0005	100	0.01	0.8
4.166666666666667	75	2	0.0005	180	0.01	0.8
4.111111111111111	75	2	0.0005	200	0.01	0.8
4.166666666666667	100	2	0.0005	200	0.01	0.8

Tab 16. Teste 3 Super Flux, parâmetros para menor distância de Levenshtein. **Fonte autoral**.

Primeiramente, não houve uma relevante melhora nos resultados, os quais, como será visto, estão condizentes com os resultados realizados na pesquisa original do método [12].

Neste último teste, é observada uma grande variação nos valores para os parâmetros Max Size e Número de Mel, contendo diversas combinações entre eles que entregam um desempenho máximo. Logo, encontram-se ótimos desempenhos para valores entre 50 a 100 para Max Size e 100 a 200 para Número de Mel, ficando a encargo do projetista e do tipo de áudio a ser analisado, o ajuste mais apropriado. Para este projeto, toma-se os valores finais:

- Max size (absoluto): 100
- Lag (absoluto): 2
- Deslocamento de salto (relativo) = 0.0005
- Número de Mel (absoluto): 200
- Tamanho da FFT (relativo): 0.01
- Tamanho da Janela (relativo): 0.8

As quais para este conjunto de áudios de testes possuem o desempenho final de:

- Acurácia: 0.8015873015873016
- Precisão: 0.8091269841269841
- Revocação: 0.8015873015873016

- F-medida: 0.8052651955090978
- Distância de Levenshtein: 4.166666666666667

E quando mudamos a tolerância do sistema para 1% temos o resultado de:

- Acurácia: 0.8941798941798944,
- Precisão: 0.9031746031746033,
- Revocação: 0.8941798941798944,
- F-medida: 0.8985675571041427,
- Distância de Levenshtein: 2.2222222222222223,

Aproximando de 90%

Sabe-se que no estudo *Maximum filter vibrato suppression for onset detection* [12], Böck, Sebastian, and Gerhard Widmer realizam testes offlines para um dataset de instrumentos de corda - também admitindo uma tolerância de 0.5%, chegaram e resultados gerais de:

- Precisão: 0.836
- Revocação: 0.701
- F-medida: 0.762

As métricas se aproximam muito com os resultados dos testes deste trabalho, provando a consistência do método. As variações se devem ao fato de o dataset, usado no artigo original do método [12], utilizar uma mistura de áudios poli (que adicionam maior complexidade) e monofônicos. No caso deste trabalho, apenas os monofônicos são utilizados tendo uma revocação e f-medida maiores. A precisão teve uma pequena variação de 2,7%.

4.2 Probabilistic Yin

Como indicado, o método Probabilistic Yin para detecção de Onset é implementado pela biblioteca Librosa utilizando o método librosa.pyin [20].

Para medir o resultado por meio das métricas, utiliza-se a distância de Levenshtein aplicada nas sequências com notas identificadas as quais estão no formato texto (string em Python). A comparação é feita diretamente.

Os parâmetros utilizados para esse teste do método Probabilistic Yin, os quais foram explicados anteriormente no subcapítulo sobre bibliotecas, são:

- Tamanho do quadro corresponde ao percentual em relação ao número de amostras
- Tamanho da janela corresponde ao percentual em relação ao tamanho do quadro
- Tamanho do passo corresponde ao percentual em relação ao número de amostras
- Filtro de probabilidade (absoluto) é a mínima porcentagem para se considerar uma predição

Os parâmetros de Onset configurados para se testar o método Probabilistic Yin são os escolhidos no subcapítulo anterior com melhor resultado referente aos testes do método Super Flux.

Não há necessidade de aplicar escala Mel e espectrograma logarítmico pois o presente método de identificação de frequência fundamental opera no domínio do tempo.

Inicialmente, testamos os seguintes pontos de interesse para os parâmetros:

- Tamanho do quadro: 0.2, 0.4, 0.6 ,0.8, 1.0, 2.0
- Filtro de probabilidade: 0.1,0.3,0.5,0.7,0.9
- Tamanho Hop: 0.02, 0.04, 0.06, 0.08
- Tamanho da Janela: 0.6, 0.8, 1.0

Sendo que, para valores do tamanho do quadro maiores que 1.0, o método faz uso do recurso de zero-*padding* centralizado e constante (abordagem padrão) em que os zeros são adicionados aos extremos do sinal, mantendo-o centralizado.

Maiores valores de acurácia:

A tabela 17 mostra os maiores valores de acurácia nos testes e o conjunto de parâmetros que foram utilizados. Vê-se que há várias combinações de Tamanho do quadro e tamanho da janela que levam aos melhores resultados, já o deslocamento de salto se consolida em 2%. Todos os valores testados para o filtro de probabilidade são possíveis.

Acurácia	Filtro de probabilidade (absoluto)	Deslocamento Salto (relativo)	Tamanho quadro (relativo)	Tamanho janela (relativo)
0.9814814814814815	0.1	0.02	0.2	0.8
0.9814814814814815	0.1	0.02	0.4	0.8
0.9814814814814815	0.1	0.02	0.6	0.8
0.9814814814814815	0.1	0.02	0.8	0.8
0.9814814814814815	0.1	0.02	1.0	0.8
0.9814814814814815	0.1	0.02	2.0	0.6
0.9814814814814815	0.3	0.02	0.2	0.8
0.9814814814814815	0.3	0.02	0.4	0.8
0.9814814814814815	0.3	0.02	0.6	0.8
0.9814814814814815	0.3	0.02	0.8	0.8

0.9814814814814815	0.3	0.02	2.0	0.6
0.9814814814814815	0.5	0.02	0.2	0.6
0.9814814814814815	0.5	0.02	0.2	0.8
0.9814814814814815	0.5	0.02	0.4	0.8
0.9814814814814815	0.5	0.02	0.6	0.8
0.9814814814814815	0.5	0.02	0.8	0.8
0.9814814814814815	0.5	0.02	2.0	0.6
0.9814814814814815	0.7	0.02	0.2	0.6
0.9814814814814815	0.7	0.02	0.2	0.8
0.9814814814814815	0.7	0.02	0.4	0.8
0.9814814814814815	0.7	0.02	0.6	0.8
0.9814814814814815	0.7	0.02	2.0	0.6
0.9814814814814815	0.9	0.02	0.2	0.6
0.9814814814814815	0.9	0.02	0.2	0.8
0.9814814814814815	0.9	0.02	0.4	0.6
0.9814814814814815	0.9	0.02	2.0	0.6

Tab 17. Teste PYIN, parâmetros para maior acurácia. **Fonte autoral**.

Maiores valores de precisão:

A tabela 18 mostra os maiores valores de precisão nos testes e o conjunto de parâmetros que foram utilizados. Vê-se que há várias combinações de Tamanho do quadro e tamanho da janela que levam aos melhores resultados, já o deslocamento de salto se consolida em 2%. Todos os valores testados para o filtro de probabilidade são possíveis.

Precisão	Filtro de probabilidade (absoluto)	Deslocamento Salto (relativo)	Tamanho quadro (relativo)	Tamanho janela (relativo)
1.0	0.1	0.02	2.0	0.6
1.0	0.1	0.02	2.0	0.8

1.0	0.3	0.02	0.2	0.8
1.0	0.3	0.02	2.0	0.6
1.0	0.3	0.02	2.0	0.8
1.0	0.5	0.02	0.2	0.8
1.0	0.5	0.02	2.0	0.6
1.0	0.5	0.02	2.0	0.8
1.0	0.7	0.02	0.2	0.8
1.0	0.7	0.02	2.0	0.6
1.0	0.7	0.02	2.0	0.8
1.0	0.9	0.02	0.2	0.8
1.0	0.9	0.02	2.0	0.6
1.0	0.9	0.02	2.0	0.8

Tab 18. Teste PYIN, parâmetros para maior precisão. **Fonte autoral**.

Maiores valores de revocação:

A tabela 19 mostra os maiores valores de revocação nos testes e o conjunto de parâmetros que foram utilizados. Vê-se que há várias combinações de Tamanho do quadro e tamanho da janela que levam aos melhores resultados, já o deslocamento de salto se consolida em 2%. Todos os valores testados para o filtro de probabilidade são possíveis.

Revocação	Filtro de probabilidade (absoluto)	Deslocamento Salto (relativo)	Tamanho quadro (relativo)	Tamanho janela (relativo)
0.9814814814814815	0.1	0.02	0.2	0.8
0.9814814814814815	0.1	0.02	0.4	0.8
0.9814814814814815	0.1	0.02	0.6	0.8
0.9814814814814815	0.1	0.02	0.8	0.8
0.9814814814814815	0.1	0.02	1.0	0.8
0.9814814814814815	0.1	0.02	2.0	0.6

0.9814814814814815	0.3	0.02	0.2	0.8
0.9814814814814815	0.3	0.02	0.4	0.8
0.9814814814814815	0.3	0.02	0.6	0.8
0.9814814814814815	0.3	0.02	0.8	0.8
0.9814814814814815	0.3	0.02	2.0	0.6
0.9814814814814815	0.5	0.02	0.2	0.6
0.9814814814814815	0.5	0.02	0.2	0.8
0.9814814814814815	0.5	0.02	0.4	0.8
0.9814814814814815	0.5	0.02	0.6	0.8
0.9814814814814815	0.5	0.02	0.8	0.8
0.9814814814814815	0.5	0.02	2.0	0.6
0.9814814814814815	0.7	0.02	0.2	0.6
0.9814814814814815	0.7	0.02	0.2	0.8
0.9814814814814815	0.7	0.02	0.4	0.8
0.9814814814814815	0.7	0.02	0.6	0.8
0.9814814814814815	0.7	0.02	2.0	0.6
0.9814814814814815	0.9	0.02	0.2	0.6
0.9814814814814815	0.9	0.02	0.2	0.8
0.9814814814814815	0.9	0.02	0.4	0.6
0.9814814814814815	0.9	0.02	2.0	0.6

Tab 19. Teste PYIN, parâmetros para maior revocação. **Fonte autoral**.

Maiores valores de F-medida:

A tabela 20 mostra os maiores valores de F-medida nos testes e o conjunto de parâmetros que foram utilizados. Vê-se que há várias combinações de Tamanho do quadro e tamanho da janela que levam aos melhores resultados, já o deslocamento de salto se consolida em 2%. Todos os valores testados para o filtro de probabilidade são possíveis.

F-Medida	Filtro de probabilidade (absoluto)	Deslocamento Salto (relativo)	Tamanho quadro (relativo)	Tamanho janela (relativo)
0.9902386908484471	0.1	0.02	2.0	0.6
0.9902386908484471	0.3	0.02	0.2	0.8
0.9902386908484471	0.3	0.02	2.0	0.6
0.9902386908484471	0.5	0.02	0.2	0.8
0.9902386908484471	0.5	0.02	2.0	0.6
0.9902386908484471	0.7	0.02	0.2	0.8
0.9902386908484471	0.7	0.02	2.0	0.6
0.9902386908484471	0.9	0.02	0.2	0.8
0.9902386908484471	0.9	0.02	2.0	0.6

Tab 20. Teste PYIN, parâmetros para maior F-medida. **Fonte autoral**.

Menores distâncias de Levenshtein:

A tabela 21 mostra os menores valores de distância de Levenshtein nos testes e o conjunto de parâmetros que foram utilizados. Vê-se que há várias combinações de Tamanho do quadro e tamanho da janela que levam aos melhores resultados, já o deslocamento de salto se consolida em 2%. Todos os valores testados para o filtro de probabilidade são possíveis.

Distância de Levenshtein	Filtro de probabilidade (absoluto)	Deslocamento Salto (relativo)	Tamanho quadro (relativo)	Tamanho janela (relativo)
0.3888888888888889	0.1	0.02	0.2	0.8
0.3888888888888889	0.1	0.02	0.4	0.8
0.3888888888888889	0.1	0.02	0.6	0.8
0.3888888888888889	0.1	0.02	0.8	0.8
0.3888888888888889	0.1	0.02	1.0	0.8
0.3888888888888889	0.1	0.02	2.0	0.6
0.3888888888888889	0.3	0.02	0.2	0.8

0.3888888888888889	0.3	0.02	0.4	0.8
0.3888888888888889	0.3	0.02	0.6	0.8
0.3888888888888889	0.3	0.02	0.8	0.8
0.3888888888888889	0.3	0.02	2.0	0.6
0.3888888888888889	0.5	0.02	0.2	0.6
0.3888888888888889	0.5	0.02	0.2	0.8
0.3888888888888889	0.5	0.02	0.4	0.8
0.3888888888888889	0.5	0.02	0.6	0.8
0.3888888888888889	0.5	0.02	0.8	0.8
0.3888888888888889	0.5	0.02	2.0	0.6
0.3888888888888889	0.7	0.02	0.2	0.6
0.3888888888888889	0.7	0.02	0.2	0.8
0.3888888888888889	0.7	0.02	0.4	0.8
0.3888888888888889	0.7	0.02	0.6	0.8
0.3888888888888889	0.7	0.02	2.0	0.6
0.3888888888888889	0.9	0.02	0.2	0.6
0.3888888888888889	0.9	0.02	0.2	0.8
0.3888888888888889	0.9	0.02	0.4	0.6
0.3888888888888889	0.9	0.02	2.0	0.6

Tab 21. Teste PYIN, parâmetros para menor distância de Levenshtein. **Fonte autoral**.

Para todos os casos, deslocamento de salto com valor relativo de 0.02 trouxe os melhores valores para todas as métricas.

Para o filtro de probabilidade, obteve-se os máximos resultados para todos os valores em testes. Pode-se analisar que o desempenho desse método para esse teste foi tão preciso que, independentemente do filtro, as notas identificadas no trecho correspondem à nota esperada em sua maioria.

Para o tamanho do quadro, por mais que outras métricas tenham obtido os maiores resultados para diversos valores, as métricas F-medida e Precisão tiveram melhores resultados para o tamanho de quadro relativo entre 0.2 e 2.0 (utilizando

zero-padding). E, acompanhando esses valores para tamanho do quadro relativo, temos o tamanho da janela variando, respectivamente, em 0.8 e 0.6. Não obstante, essa relação de tamanho de quadro e janela relativos de 2.0 e 0.2 e 0.8 e 0.6 foi visto em padrão para as demais métricas.

Nota-se também, o valor máximo de precisão em 100% o qual indica que, por mais que não tenha tido uma acurácia de 100%- isto é, identificado todas as notas presentes em ordem na harmonia- , todas as notas identificadas eram notas presentes no áudio de teste.

Para melhores resultados, escolhemos:

- Multiplicador tamanho do quadro relativo à amostras : 2.0
- Multiplicador tamanho da janela relativo ao tamanho do quadro: 0.6
- Tamanho Hop relativo à amostras: 0.02
- Filtro de probabilidade : 0.5

Tendo como melhores resultados:

- Acurácia: 0.9814814814814815
- Precisão: 1.0
- Revocação: 0.9814814814814815
- F-medida:0.9902386908484471
- Distância de Levenshtein: 0.3888888888888889

Resultados que se aproximam bastante dos encontrados no artigo *PYIN: A fundamental frequency estimator using probabilistic threshold distributions* [17], em que M. Mauch and S. Dixon propõem o método utilizado, provando a equivalência de performance do método.

Nessa escolha de parâmetros para o tamanho do quadro, escolhe-se 2.0 para fazer-se uso de zero-padding melhorando a resolução em frequência do sinal. Mesmo não sendo especificamente necessário nestes testes, optou-se pelo motivo de, em usos futuros da aplicação em que tenha se um número de amostras para uma nota em quantidade muito pequena (em relação à frequência de amostragem), possa se melhorar a resolução.

4.3 CREPE Pitch Tracker

Como indicado, o método CREPE pitch tracker para detecção de frequência fundamental é implementado pela biblioteca CREPE utilizando o método predict [25].

Para medir o resultado por meio das métricas, utilizamos a distância de Levenshtein aplicada nas sequências com notas identificadas que estão no formato texto (string em Python). A comparação é feita diretamente.

Os parâmetros utilizados para esse teste do método CREPE pitch tracker, os quais foram explicados anteriormente no subcapítulo sobre bibliotecas, são:

- Deslocamento do salto no tempo (relativo): Dado em milissegundos, corresponde ao percentual em relação ao número de amostras.

- Filtro de confidência (absoluto): mínima porcentagem para se considerar uma predição

Os parâmetros de Onset configurados para testar o método CREPE pitch tracker são os escolhidos no subcapítulo anterior com melhor resultado referente aos testes do método Super Flux. Não há necessidade de aplicar escala Mel e espectrograma logarítmico, pois o presente método de identificação de frequência fundamental opera no domínio do tempo.

Inicialmente, testamos os seguintes pontos de interesse para os parâmetros:

- Filtro probabilidade = 0.1,0.3,0.5,0.7,0.9
- Tamanho Step = 0.01, 0.03, 0.05, 0.07, 0.09, 0.1

(Valores menores que 0.01 tomam alto custo computacional e de tempo, não tendo um bom custo-benefício)

Maiores resultados para acurácia:

A tabela 22 mostra os maiores valores de acurácia nos testes e o conjunto de parâmetros que foram utilizados. Vê-se que todos os valores testados para o filtro de probabilidade foram possíveis nos melhores resultados e que o tamanho do salto no tempo se consolidou em 1%.

Acurácia	Filtro Probabilidade (absoluto)	Tamanho salto no tempo (relativo)
0.9629629629629629	0.3	0.01
0.9629629629629629	0.5	0.01
0.9629629629629629	0.7	0.01
0.9735449735449736	0.9	0.01

Tab 22. Teste CREPE, parâmetros para maior acurácia. **Fonte autoral**.

Maiores resultados para Precisão:

A tabela 23 mostra os maiores valores de precisão nos testes e o conjunto de parâmetros que foram utilizados. Por mais que houve um valor de 3% para o tamanho de salto no tempo, a maioria ficou em 1%.

Precisão	Filtro Probabilidade (absoluto)	Tamanho salto no tempo (relativo)
0.9728835978835979	0.3	0.01
0.9728835978835979	0.5	0.01
0.9728835978835979	0.7	0.01
0.9860971874129769	0.9	0.01
0.9781467557783348	0.9	0.03

Tab 23. Teste CREPE, parâmetros para maior precisão. **Fonte autoral**.

Maiores resultados para Revocação:

A tabela 24 mostra os maiores valores de revocação nos testes e o conjunto de parâmetros que foram utilizados. Os melhores resultados de revocação admitiram os valores de 1% e 3% para o tamanho do salto no tempo em conjunto para todos os valores de filtro de probabilidade testados.

Revocação	Filtro Probabilidade (absoluto)	Tamanho salto no tempo (relativo)
0.9629629629629629	0.3	0.01
0.9603174603174605	0.3	0.03
0.9629629629629629	0.5	0.01
0.9603174603174605	0.5	0.03
0.9629629629629629	0.7	0.01
0.9603174603174605	0.7	0.03
0.9735449735449736	0.9	0.01
0.9603174603174605	0.9	0.03

Tab 24. Teste CREPE, parâmetros para maior revocação. **Fonte autoral**.

Maiores resultados para F-medida:

A tabela 25 mostra os maiores valores de F-medida nos testes e o conjunto de parâmetros que foram utilizados. O mesmo se repete para F-medida como aconteceu na acurácia e precisão

F-medida	Filtro Probabilidade (absoluto)	Tamanho salto no tempo (relativo)
0.9678022970705898	0.3	0.01
0.9678022970705898	0.5	0.01
0.9678022970705898	0.7	0.01
0.9796038198477223	0.9	0.01
0.9688863079106981	0.9	0.03

Tab 25. Teste CREPE, parâmetros para maior F-medida. **Fonte autoral**.

Menores distâncias de Levenshtein:

A tabela 26 mostra os menores valores de distância de Levenshtein nos testes e o conjunto de parâmetros que foram utilizados. O Tamanho de salto no tempo se consolida em 1% admitindo todos os valores para o filtro de probabilidade testados.

Distância de Levenshtein	Filtro Probabilidade (absoluto)	Tamanho salto no tempo (relativo)
0.7777777777777778	0.3	0.01
0.7777777777777778	0.5	0.01
0.7777777777777778	0.7	0.01
0.5555555555555556	0.9	0.01

Tab 26. Teste CREPE, parâmetros para menores distância de Levenshtein. **Fonte autoral**.

Vemos que para todas as métricas, predomina-se o valor relativo de 0.01 para o tamanho de salto no tempo, em conjunto com todos os valores de Filtro de Probabilidade com exceção de 0.1.

Para melhores resultados, escolhemos:

- Filtro de Probabilidade: 0.7
- Tamanho do salto no tempo: 0.01

4.4 Testes Finais

Os testes realizados anteriormente contavam com áudios que tocavam, sequencialmente, nota por nota dos trastes 0 (corda solta) até o 20 (os mais utilizados usualmente) em todas as cordas, em modelos de guitarras diferentes. O espaço de tempo entre as notas é feito de modo similar, sem muita variação, o que foi essencial para se testar os métodos citados e encontrar parâmetros ótimos para áudios monofônicos de guitarra e violão. Agora, outra coleção de áudios de teste é necessária para colocar em prova a aplicação em um cenário real com áudios monofônicos tocados em cordas variadas, notas não sequenciais e com intervalo de tempo variado (ritmo).

Com a escolha de parâmetros realizada nos testes anteriores que entregam o melhor desempenho, testa-se para conjuntos de áudios diversos presentes nos dataset 2 e 3 da base de dados IDMT-SMT-Guitar [9] que possuem 6 trechos monofônicos tocados em três marcas de guitarras diferentes em 2 versões (dataset 2) e 2 músicas monofônicas (dataset 3) tocados pela guitarra Ibanez RG 2820. Os resultados finais foram:

- Acurácia: : 0.9662007583060216
- Precisão: 0.9909774436090226
- Revocação: 0.9662007583060216
- F-Medida: 0.9773000388079276
- Distância de Levenshtein: 0.5

Para um segundo teste, coloca-se em prática o algoritmo em cenários reais caseiros. Faz-se uso de 10 gravações caseiras pelo próprio autor com instrumento próprio, as quais incluem trechos de músicas gravadas pelo celular utilizando um microfone de lapela. Obteve-se os seguintes resultados:

- Acurácia: : 0.878409090909091
- Precisão: 0.8529817404817404
- Revocação: 0.878409090909091
- F-Medida: 0.865111494451153
- Distância de Levenshtein: 1.5

Os resultados demonstram um ótimo resultado para gravações caseiras e cenários práticos abrindo espaço para futuras melhorias.

4.5 Exemplo de funcionamento Aplicação

Inicialmente, configura-se o arquivo setup.py com os parâmetros dos métodos escolhidos. Por exemplo, inicia-se com o método para identificação de Onset. No primeiro parâmetro, é escolhido o método (dentre os implementados, no caso, Super Flux). A tática utilizada no exemplo abaixo é a variação dinâmica relativa, nomeada de

‘dinamic’. O parâmetro show habilita gráficos visuais e verbose habilita resultados escritos no terminal de forma verbosa. Dentro da chave methods, configuram-se os parâmetros utilizados. Caso algum não seja indicado, o algoritmo utilizará os valores padrões da biblioteca que o implementa. No caso abaixo, para Super Flux, lista-se os parâmetros utilizando a tática de variação dinâmica relativa. Caso nenhum método seja selecionado, a aplicação utiliza o padrão bypass no qual, nessa etapa, nada é feito.

```
OnsetDetection_parameters = {
        'method': 'super_flux',
        'tactic': 'dinamic',
        'show': True,
        'verbose':True,
        'methods':{
            'super_flux':{
                'static':{},
                 'dinamic':{
                          'n_fft' :0.01,
                           'hop_length' : 0.0005,
                           'win_length': 0.8,
                            'lag' : 2,
                            'n_mels' : 200,
                             'fmin' : 82,
                             'fmax' : 2093.,
                             'max_size' : 100,
                             'spectogram':'mel',
                              },
                             },
                             'bypass':{ },
                        }
}
```

Para a identificação da frequência fundamental, o esquema de configuração é similar ao explicado acima. No exemplo abaixo, constam-se configurações tanto para o método CREPE quanto para PYIN. O método a ser efetivamente utilizado é configurado no parâmetro method.

Caso nenhum método seja selecionado, a aplicação utiliza o padrão bypass, no qual nada é feito nessa etapa.

```
f0Detection_parameters = {
      'method':'probabilistic_yin',
      'tactic': 'dinamic',
      'result_type': 'max_count',
      'verbose':True,
      'show': True,
      'delete_min' : False,
      'methods':{
          'probabilistic_yin':{
                  'static':{},
                  'dinamic':{
                      'voiced_probs_filter':0.5,
                       'frame_length':2.0,
                       'win_length': 0.6,
                       'hop_length': 0.02,
                       'fmin': 82.0,
                        'n_thresholds':100,
                        },
                   },
          'crepe_pitch_tracker':{
                  'static':{},
                  'dinamic':{
                      'confidence_filter':0.7,
                      'step_size':0.01,
                                },
                           },
                           'bypass':{},
                       }
}
```

São incluídas, também, algumas breves configurações para a otimização de posição e transcrição para tablatura, que incluem os parâmetros INIT e END, explicados anteriormente, e o diretório para se salvar a tablatura gerada. O áudio para análise é incluso no arquivo *audio_samples*, indicando o diretório, frequência de amostragem, *offset* e duração caso necessários.

O programa roda com o arquivo main.py, o qual busca as configurações no arquivo citado anteriormente.

Com as configurações definidas acima, utilizando um áudio de exemplo, as seguintes informações sobre o áudio são mostradas no terminal:

```
2023-05-10 23:48:20-0300        [__main__]      [INFO]  Audio: LP_Lick1_KN.wav | Sample_rate: 44100 |
Samples: 515157 | Audio duration: 11.68156462585034 s

2023-05-10 23:48:20-0300            [OnsetDetection]            [INFO]   Method "super_flux" | Tactic
"dinamic"
```

Prosseguindo com a identificação de *Onset*, plota-se os onsets identificados pelo método configurado:

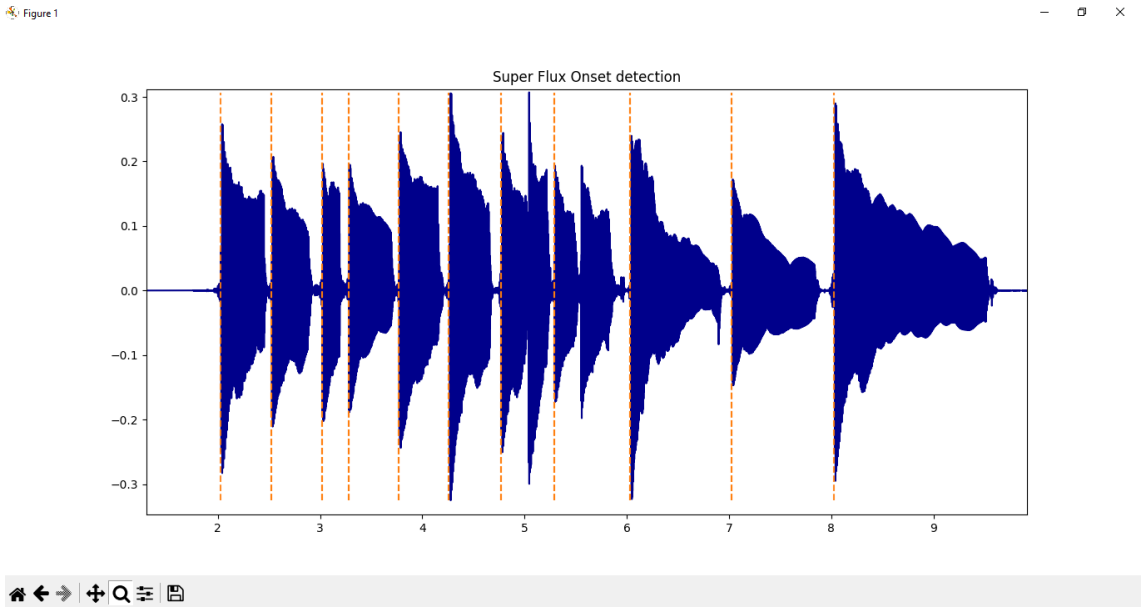

Fig 17. Exemplo de funcionamento da aplicação: Onsets. **Fonte autoral**.

A aplicação segue imprimindo na tela em extenso os *onsets* identificados em unidades de amostra, tempo, amplitude e quadros.

```
2023-05-10 23:51:48-0300        [__main__]      [INFO]  Onset Final Result: {'onset_samples':
[(89436, 111281), (111281, 133126), (133126, 144691), (144691, 166279), (166279, 187867), (187867,
210226), (210226, 233356), (233356, 265995), (265995, 309942), (309942, 353889), (353889, 515157)],
'onset_times': [(2.028027210884354, 2.5233786848072564), (2.5233786848072564, 3.018730158730159),
(3.018730158730159, 3.2809750566893423), (3.2809750566893423, 3.770498866213152), (3.770498866213152,
4.260022675736962), (4.260022675736962, 4.76702947845805), (4.76702947845805, 5.291519274376418),
(5.291519274376418, 6.031632653061225), (6.031632653061225, 7.028163265306122), (7.028163265306122,
8.024693877551021), (8.024693877551021, 11.68156462585034)], 'onset_amplitudes':
[-0.0039429847674314475, 0.008506056429223403, -0.0035296029026117117, -0.01733955211112636,
0.008858910873076737, -0.0070404021228853515, -0.020948095728221795, 0.03997616972383058,
```

```
0.003745406591789419, -0.0022129991065995766, 0.004634752150215624], 'onset_frames': array([ 348,
433,  518,  563,  647,  731,  818,  908, 1035, 1206, 1377],

      dtype=int64)}
```

Então, inicia-se o procedimento de identificação de frequência fundamental com o método escolhido para cada segmento. Informações sobre o intervalo de tempo, amostras as quais pertencem esse segmento no áudio original e informações relativas ao método de identificação de frequência fundamental são mostrados na figura 18 (imagem cortada).

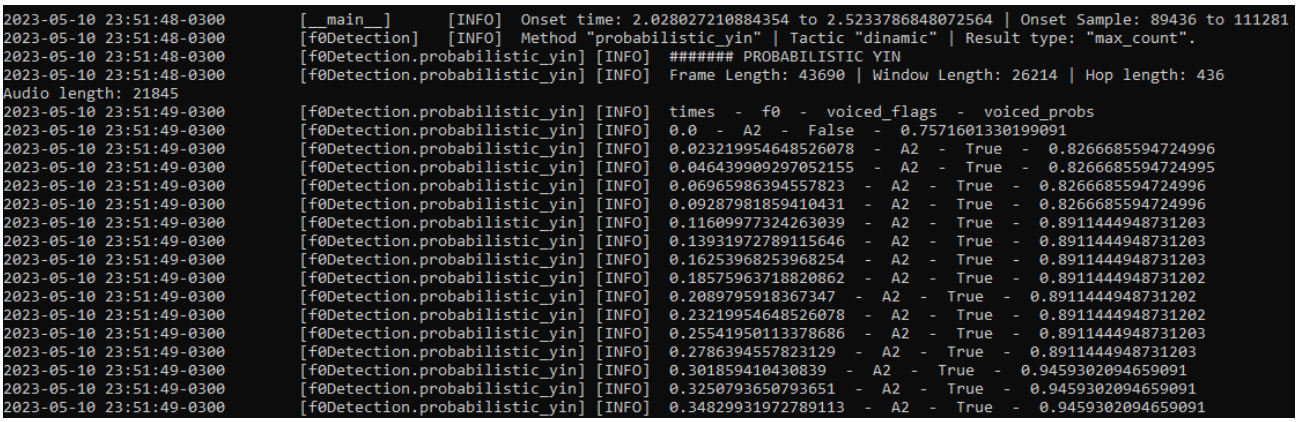

Fig 18. Exemplo de funcionamento da aplicação: Identificação de frequência fundamental por fração de tempo. **Fonte autoral**.

A aplicação, então, plota (caso configurado para isso), para cada frequência fundamental estimada, o espectrograma, mostrado na figura 19.

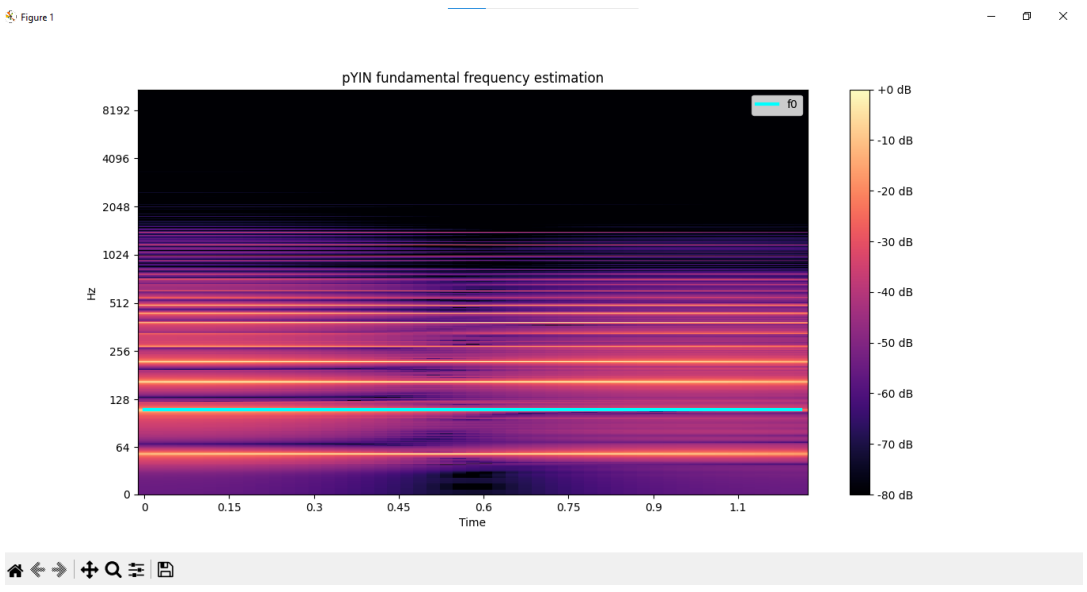

Fig 19. Exemplo de funcionamento da aplicação: Identificação de frequência fundamental. **Fonte autoral**.

Além disso, informa o filtro de probabilidade configurado e a lista de frequências encontradas para que seja escolhida utilizando a estratégia de seleção de notas pela contagem da maioria. As frequências são convertidas para a notação em nota

para que isso seja feito. A figura 20 mostra a lista de frequências fundamentais encontradas após o filtro de probabilidade.

```
2023-05-10 23:58:47-0300        [f0Detection.probabilistic_yin] [INFO]  Disable Fmin deletion
2023-05-10 23:58:47-0300        [f0Detection.probabilistic_yin] [INFO]  Voiced Probs filter set to 0.5
2023-05-10 23:58:47-0300        [f0Detection.probabilistic_yin] [INFO]   New F0: [110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.090945
2279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.090
9452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.
0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 1
10.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816
, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279816, 110.0909452279
816, 110.0909452279816, 110.0909452279816]
```

Fig 20. Exemplo de funcionamento da aplicação: Lista de frequências. **Fonte autoral**.

Desta forma, é imprimida na tela a frequência nota estimada para esse segmento, conforme mostra a figura 21.

```
2023-05-11 12:02:15-0300        [f0Detection.get_results]       [INFO]  Max count note: A2
2023-05-11 12:02:15-0300        [__main__]      [INFO]  F0  Result: A2
2023-05-11 12:02:16-0300        [__main__]      [INFO]
```

Fig 21. Exemplo de funcionamento da aplicação: Resultado final da identificação de frequência fundamental. **Fonte autoral**.

Esse processo é repetido para cada um dos segmentos de *Onset.* Ao final, mostra-se a sequência de notas identificadas, as posições selecionadas e indica-se o arquivo contendo a tablatura transcrita em um arquivo .txt em ASCII salvo no diretório indicado nas configurações da aplicação e as notas encontradas no áudio.

2023-05-11 22:47:16-0300 [__main__] [INFO] Audio:AR_Lick1_FN.wav

2023-05-11 22:47:16-0300 [__main__] [INFO] Predicted Notes: ['A2', 'C3', 'G3', 'E3', 'D3', 'C3', 'A2', 'E3', 'C3', 'D4', 'G3', 'C3']

2023-05-11 22:47:16-0300 [Position_Selector] [INFO] Positions Selected: [(0, 0), (5, 0), (5, 3), (3, 0), (4, 2), (4, 0), (5, 3), (5, 0), (4, 2), (5, 3), (2, 3), (3, 0), (5, 3), (0, 0)]

2023-05-11 22:47:16-0300 [Tablature_Transcription.transcribe] [INFO] Tabs saved at C:\Users\pedrohbf\Desktop\Projetos\Python\AGT_tabs\tabs\AR_Lick1_FN_tabs.txt

O procedimento é equivalente ao método CREPE.

A tablatura encontrada é demonstrada na figura 22.

```
AR_Lick1_FN_tabs - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
1:  |--------------------------------------------------|
2:  |--------------------------------------3-----------|
3:  |----------0-------------------------------0-------|
4:  |--------------2---0-----------2-------------------|
5:  |--0---3---------------3---0-------3-----------3---|
6:  |--------------------------------------------------|
```

Fig 22. Exemplo de funcionamento da aplicação: Tablatura. **Fonte autoral**.

Capítulo 5 - Conclusão

Pelos testes individuais de cada método de identificação de frequência fundamental, no caso CREPE e Probabilistic Yin, ambos tiveram resultados bem próximos em suas métricas. No entanto, o melhor resultado pende, com leves variações, para o Probabilistic Yin, que segue como o método no estado da arte com melhores resultados entregues para identificação de frequência fundamental em áudios monofônicos. CREPE alcança excelentes resultados estando equiparável com o estado da arte utilizando redes neurais convolucionais.

O método de identificação de *Onsets* Super Flux executou papel essencial para a segmentação do áudio em notas, facilitando as etapas posteriores. Pelos testes, o método mantém sua ótima performance indicada pelos estudos deste, em áudios monofônicos.

Os testes finais tiveram uma performance relevante, abrindo espaço para melhorias, para ser adaptado e utilizado em mais cenários reais e caseiros com gravações que contenham ruídos e sejam gravadas pelo usuário.

O projeto cumpre com o objetivo de estudar métodos para transcrição automática de áudios monofônicos de guitarra e trazer uma proposta de implementação utilizando o dataset IDMT-SMT-Guitar como base para calibração de parâmetros relativos à quantidade de amostras e testes, considerando os áudios monofônicos.

5.1 Trabalhos futuros

Adaptar a aplicação para reconhecimento de acordes e áudios polifônicos em geral é um desafio para ampliar as funcionalidades, além de amplificar para o uso com áudios contendo ruídos e com gravações caseiras. É possível, também, utilizar coeficientes de Inarmonicidade - levando em conta tensão, medidas, diâmetro e material que compõem o instrumento e suas cordas - para se identificar a posição exata na qual foram tocadas as notas no áudio. Diversos trabalhos como [28] e [29] utilizam Inteligência Artificial para a identificação das notas e em [30] são utilizadas técnicas de visão computacional para identificar diretamente as posições tocadas em um vídeo, nas quais pode-se trabalhar em conjunto com os métodos abordados neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

Sites:

ARAÚJO, Patrícia. Estudo da IPFI mostra os hábitos de consumo de música no mundo. **MCT**, 2021. Disponível em: <http://mct.mus.br/tag/ifpi/>. Acesso em: dia, mês e ano.

Pra que serve a música? . Super Abril Ciência, 31 de Julho de 2004. Disponível em: <https://super.abril.com.br/ciencia/para-que-serve-a-musica/>

Ep. 20 -Música e Engenharia | Maua Cást. Blog da Mauá, 8 de Dezembro de 2022. Disponível em: <https://blog.maua.br/2022/12/ep-20-musica-e-engenharia-mauacast/>

PONTES, Márcio Miranda. Sociedade Artística Brasileira, 15 de Abril de 2020. Disponível em: <https://www.sabra.org.br/site/instrumentos-musicais/>

Nunca se ouviu tanta música quanto nos nossos dias. ARKADE. 22 de Novembro de 2022. Disponível em: <https://www.arkade.com.br/nunca-se-ouviu-tanta-musica-quanto-nos-nossos-dias-de-acordo-com-uma-pesquisa/>

PETRIN, Natália. Estudo Prático. 8 de Junho de 2015. Disponível em: <https://www.estudopratico.com.br/a-matematica-e-a-musica-relacao-e-escalas/>

Artigos e materiais:

[1] Juan Pablo Bello, Laurent Daudet, Samer A. Abdallah, Chris Duxbury, Mike E. Davies, and Mark B. Sandler: **A Tutorial on Onset Detection in Music Signals.** IEEE Transaction on Speech and Audio Processing 13(5-2), 2005, pp. 1035–1047.

[2] https://musicinformationretrieval.com/

[3] **Guitars.AI.** Disponível em: https://github.com/GuitarsAI

[4] **Guitars.AI.** Disponível em: https://www.youtube.com/channel/UCyAyQAu_PTX5h1Ni4q0ShHQ

[5] M. K. Boloyos, T. K. Libunao, J. Masilungan, F. d. Leon, C. R. Lucas and C. T. Tolentino, "Monophonic Audio-Based Automatic Acoustic Guitar Tablature Transcription System with Legato Identification," TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON), Auckland, New Zealand, 2021, pp. 516-521, doi: 10.1109/TENCON54134.2021.9707430.

[6] I. Barbancho, L. J. Tardon, S. Sammartino and A. M. Barbancho, "Inharmonicity-Based Method for the Automatic Generation of Guitar Tablature," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 6, pp. 1857-1868, Aug. 2012, doi: 10.1109/TASL.2012.2191281.

[7] L. Alcabasa and N. Marcos, "Automatic Guitar Music Transcription," 2012 International Conference on Advanced Computer Science Applications and

Technologies (ACSAT), Kuala Lumpur, Malaysia, 2012, pp. 197-202, doi: 10.1109/ACSAT.2012.78.

[8] X. Fiss and A. Kwasinski, "Automatic real-time electric guitar audio transcription," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 2011, pp. 373-376, doi: 10.1109/ICASSP.2011.5946418.

[9] Kehling, Christian, Männchen, Andreas, & Eppler, Arndt. (2023). IDMT-SMT-Guitar Dataset (1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7544110

[10] S. S. Stevens, J. Volkmann, E. B. Newman; A Scale for the Measurement of the Psychological Magnitude Pitch. *J Acoust Soc Am* 1 January 1937; 8 (3): 185–190. https://doi.org/10.1121/1.1915893

[11] Rolczyński, Rafał. (2019). Synthetic Boosted Automatic Speech Recognition. 10.13140/RG.2.2.35163.11042.

[12] Böck, Sebastian, and Gerhard Widmer. “Maximum filter vibrato suppression for onset detection.” 16th International Conference on Digital Audio Effects, Maynooth, Ireland. 2013.

[13] MÜLLER,Meinard. Novelty Comparison. Audio Labs Erlangen. Disponível em <https://www.audiolabs-erlangen.de/resources/MIR/FMP/C6/C6S1_NoveltyComparison.html>.

[14] Böck, Sebastian & Krebs, Florian & Schedl, Markus. (2012). EVALUATING THE ONLINE CAPABILITIES OF ONSET DETECTION METHODS.

[15] MÜLLER,Meinard. Novelty Spectral. Audio Labs Erlangen. Disponível em <https://www.audiolabs-erlangen.de/resources/MIR/FMP/C6/C6S1_NoveltySpectral.html>.

[16] De Cheveigné, Alain, and Hideki Kawahara. “YIN, a fundamental frequency estimator for speech and music.” The Journal of the Acoustical Society of America 111.4 (2002): 1917-1930.

[17] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 659-663, doi: 10.1109/ICASSP.2014.6853678.

[18] Kim, Jong & Salamon, Justin & Li, Peter & Bello, Juan. (2018). CREPE: A Convolutional Representation for Pitch Estimation. Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on.

[19] LIBROSA.LOAD. Documentação da biblioteca. Disponível em: https://librosa.org/doc/main/generated/librosa.load.html

[20] LIBROSA.PYIN. Documentação da biblioteca. Disponível em: https://librosa.org/doc/main/generated/librosa.pyin.html

[21] LIBROSA.ONSET_STRENGTH. Documentação da biblioteca. Disponível em: https://librosa.org/doc/main/generated/librosa.onset.onset_strength.html

[22] LIBROSA.ONSET_DETECT. Documentação da biblioteca. Disponível em: https://librosa.org/doc/main/generated/librosa.onset.onset_detect.html

[23] LIBROSA.FEATURE.MELSPECTROGRAM. Documentação da biblioteca. Disponível em: https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html

[24] LIBROSA.POWER_TO_DB. Documentação da biblioteca. Disponível em: https://librosa.org/doc/main/generated/librosa.power_to_db.html

[25] CREPE. Documentação da biblioteca. Disponível em: https://pypi.org/project/crepe/

[26] Levenshtein, V. I. (1966), 'Binary codes capable of correcting deletions, insertions and reversals.', *Soviet Physics Doklady* 10 (8), 707--710.

[27] Young, Brian & Faris, Tom & Armogida, Luigi. (2021). Levenshtein Distance as a Measure of Accuracy and Precision in Forensic PCR-MPS Methods. 10.1101/2021.01.03.425149.

[28] J. d. J. Guerrero-Turrubiates, S. E. Gonzalez-Reyna, S. E. Ledesma-Orozco and J. G. Avina-Cervantes, "Pitch estimation for musical note recognition using Artificial Neural Networks," 2014 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 2014, pp. 53-58, doi: 10.1109/CONIELECOMP.2014.6808567.

[29] E. J. Humphrey and J. P. Bello, "From music audio to chord tablature: Teaching deep convolutional networks toplay guitar," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 6974-6978, doi: 10.1109/ICASSP.2014.6854952.

[30] Paleari, Marco & Huet, Benoit & Schutz, Antony & Slock, Dirk. (2008). A multimodal approach to music transcription. Proceedings - International Conference on Image Processing, ICIP. 93 - 96. 10.1109/ICIP.2008.4711699.

Apêndices

Apêndice I - Teste 1 Super Flux

Disponível em: https://github.com/pbitts/AGT_tabs/tree/main/Tests

Apêndice II - Teste 2 Super Flux

Disponível em: https://github.com/pbitts/AGT_tabs/tree/main/Tests

Apêndice III - Teste 3 Super Flux

Disponível em: https://github.com/pbitts/AGT_tabs/tree/main/Tests

Apêndice IV -Teste PYIN

Disponível em: https://github.com/pbitts/AGT_tabs/tree/main/Tests

Apêndice V -Teste CREPE

Disponível em: https://github.com/pbitts/AGT_tabs/tree/main/Tests

Apêndice VI - Repositório da Aplicação

Disponível em: https://github.com/pbitts/AGT_tabs