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

ESTIMAÇÃO DE MÚLTIPLOS *PITCHES* EM ÁUDIO MUSICAL POLIFÔNICO UTILIZANDO REDE NEURAL CONVOLUCIONAL

Marcus Fábio Santos da Silva

Manaus
Setembro de 2025

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

ESTIMAÇÃO DE MÚLTIPLOS *PITCHES* EM ÁUDIO MUSICAL POLIFÔNICO UTILIZANDO REDE NEURAL CONVOLUCIONAL

Marcus Fábio Santos da Silva

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

Orientadores: Waldir Sabino da Silva Júnior
Luiz Wagner Pereira Biscainho

Manaus
Setembro de 2025

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

Pós-Graduação em Engenharia Elétrica. Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI. Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

MARCUS FÁBIO SANTOS DA SILVA

ESTIMAÇÃO DE MÚLTIPLOS PITCHES EM AUDIO MUSICAL POLIFÔNICO UTILIZANDO REDE NEURAL CONVOLUCIONAL

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

Aprovado em 15 de outubro de 2025.

BANCA EXAMINADORA
Prof. Dr. Waldir Sabino da Silva Júnior - Presidente
Prof. Dr. Florindo Antônio de Carvalho Ayres Junior - Membro Titular 1 - Interno
Prof. Dr. Gabriel Matos Araújo - Membro Titular 2 - Externo

Manaus, 26 de setembro de 2025.

Documento assinado eletronicamente por **Florindo Antonio de Carvalho Ayres Júnior**, **Professor do Magistério Superior**, em 22/10/2025, às 13:46, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015 .

Documento assinado eletronicamente por **Waldir Sabino da Silva Júnior**, **Coordenador**, em 03/11/2025, às 14:34, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Gabriel Matos Araujo**, **Usuário Externo**, em 05/11/2025, às 13:24, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2817112** e o código CRC **78A25DBB**.

Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Bairro Coroado Campus Universitário, Setor Norte - Telefone: 99271-8954
CEP 69080-900 Manaus/AM - Pavilhão do CETELI. E-mail: ppgee@ufam.edu.br

Referência: Processo nº 23105.042599/2025-42 SEI nº 2817112

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S586e Silva, Marcus Fabio Santos da
Estimação de múltiplos pitches em áudio musical polifônico utilizando rede neural convolucional / Marcus Fabio Santos da Silva. - 2025.
91 f. : il., color. ; 31 cm.

Orientador(a): Waldir Sabino da Silva Júnior.
Orientador(a): Luiz Wagner Pereira Biscainho.
Dissertação (mestrado) - Universidade Federal do Amazonas, Programa de Pós-Graduação em Engenharia Elétrica, Manaus, 2025.

1. Polifonia. 2. Recuperação de informação musical. 3. Rede neural convolucional. 4. Pitch. 5. Aprendizado de máquina. I. Silva Júnior, Waldir Sabino da. II. Biscainho, Luiz Wagner Pereira. III. Universidade Federal do Amazonas. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título

---

Agradecimentos

- Primeiramente a Deus.
- À minha amada esposa Rebeca Silva.
- Aos meus amados filhos Marcus, Laís e Letícia.
- Aos meus caros orientadores e também amigos, Prof. D. Sc. Waldir Sabino e Prof. D. Sc. Luiz Wagner.

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

ESTIMAÇÃO DE MÚLTIPLOS *PITCHES* EM ÁUDIO MUSICAL POLIFÔNICO UTILIZANDO REDE NEURAL CONVOLUCIONAL

Marcus Fábio Santos da Silva

Orientadores: Waldir Sabino da Silva Júnior
Luiz Wagner Pereira Biscainho

Programa: Pós-Graduação em Engenharia Elétrica

Pitch é a percepção auditiva da altura de um som, relacionada principalmente à sua frequência fundamental. A estimativa de *pitch* em áudio musical é uma tarefa desafiadora no campo da recuperação de informação musical (MIR). Determinar com precisão a frequência fundamental (F0) das notas musicais é crucial para várias aplicações, incluindo transcrição musical, extração de melodia e análise de áudio. O problema é complicado por fatores como polifonia, ruído de fundo e variações no timbre e na dinâmica. Métodos tradicionais para estimativa de *pitch*, enfrentando essas complexidades, muitas vezes produzem resultados imprecisos ou pouco confiáveis. Recentes avanços no aprendizado profundo, particularmente o uso de redes neurais convolucionais (CNNs), têm se mostrado promissores na resolução desses desafios. As CNNs são capazes de aprender representações hierárquicas a partir de dados de áudio brutos, capturando efetivamente as características temporais e espectrais essenciais para uma estimativa de *pitch* precisa. Este trabalho explora o problema da estimativa de múltiplos *pitches* em áudio musical. Discutimos as inovações na arquitetura do modelo CREPE e estratégias de treinamento que tornam o modelo capaz de estimar múltiplos *pitches*. Os resultados demonstraram desempenho robusto em diferentes situações. Em conjuntos de validação com múltiplas frequências, o novo modelo proposto apresentou valores elevados de RPA, com média global próxima de 0,93, evidenciando sua capacidade de identificar corretamente *pitches* simultâneos.

Palavras-chave: Pitch, Recuperação de Informação Musical, Rede Neural.

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

MULTI PITCH ESTIMATION IN POLYPHONIC MUSIC AUDIO USING CONVOLUTIONAL NEURAL NETWORK

Marcus Fábio Santos da Silva

Advisors: Waldir Sabino da Silva Júnior
Luiz Wagner Pereira Biscainho

Department: Postgraduate in Electrical Engineering

Pitch is the auditory perception of the height of a sound, primarily related to its fundamental frequency. Pitch estimation in musical audio is a fundamental yet challenging task in the field of music information retrieval (MIR). Accurately determining the fundamental frequency (F0) of musical notes is crucial for various applications, including music transcription, melody extraction, and audio analysis. The problem is complicated by factors such as polyphony, background noise, and variations in timbre and dynamics. Traditional methods for estimating pitch, when dealing with these complexities, often produce inaccurate or unreliable results. Recent improvements in deep learning, especially using convolutional neural networks (CNNs), have shown potential in tackling these challenges. CNNs are capable of learning hierarchical representations from raw audio data, effectively capturing the temporal and spectral features essential for accurate pitch estimation. This work explores the problem of multi-pitch estimation in musical audio, highlighting the inherent challenges and the current state-of-the-art solutions using CNNs. We discussed the innovations in the CREPE model architecture and the training strategies that enable the model to estimate multiple *pitches*. The results demonstrated robust performance in different scenarios. In validation sets with multiple frequencies, the newly proposed model achieved high RPA values, with an overall average close to 0,93, highlighting its ability to correctly identify simultaneous *pitches*.

Keywords: Pitch, Music Information Retrieval, Neural Network.

Sumário

1 Introdução **1**

1.1 Motivação e Contexto 2

1.2 Objetivos da Dissertação 4

1.3 Organização da Dissertação 6

2 Fundamentos Teóricos **8**

2.1 Noções sobre *Pitch* 8

2.2 Transformada de Fourier de Curta Duração 10

2.3 Algoritmos para Detecção de *Pitch* - YIN e pYIN 13

2.4 Algoritmo YIN 14

2.5 Algoritmo pYIN 16

2.6 Rede Neural CREPE 18

2.6.1 Bases de Dados 22

3 Metodologia Proposta **25**

3.1 Introdução 25

3.2 Estudo de modelos monofônicos 25

3.2.1 Objetivos 26

3.2.2 Base de dados de Flauta 26

3.2.3 Implementação e testes da rede neural CREPE 27

3.2.4 Implementação e testes da rede neural FCN 31

3.3 Proposta de modelo polifônico 37

3.3.1 Mixagem da base de dados MDB-stem-synth 37

3.3.2 Implementação do modelo para estimativa de múltiplos *pitches* 37

4 Experimentos e resultados **39**

4.1 Objetivos 39

4.2 Ambiente de desenvolvimento 41

4.3 Metodologia experimental 44

4.3.1 Compreensão do Modelo CREPE 44

4.3.2 Construção da Base de Dados *MDB-stem-synth-multi* 51

4.3.3 Implementação do Modelo CREME 54

4.3.4 Procedimentos de Treinamento 59

4.4 Avaliação experimental 61

4.4.1 Avaliação com dados de validação 61

4.4.2 Avaliação pelos indicadores de erro 61

4.4.3 Teste com base de flauta *Traditional Flute Dataset* 64

4.4.4 Teste com base *Bach10-mf0-synth* 65

4.4.5 Testes com a base de dados RWC (Violão e Clarinete) 66

4.5 Discussão dos Resultados 75

4.5.1 Análise dos pontos fortes e fracos do modelo CREME 76

4.5.2 Sugestões de melhorias futuras 81

5 Conclusão **83**

5.1 Considerações Finais da Dissertação 83

5.1.1 Recapitulação dos principais resultados obtidos 83

5.1.2 Validação do objetivo de generalizar o CREPE 84

5.1.3 Aplicações práticas e estudos futuros 84

Referências Bibliográficas **86**

Lista de Figuras

2.1 Diagrama em blocos da arquitetura do CREPE. 21

3.1 Ajuste na anotação da base de flauta. 27
3.2 Diagrama de testes do FCN com a base de dados Traditional Flute. . 32

4.1 Conversão da base de dados MDB-stem-synth. 45
4.2 Diagrama para treinamento exploratório do CREPE. 46
4.3 Primeiro treinamento exploratório do CREPE. 47
4.4 Diagrama para treinamento do CREPE com aumento e randomização de dados. 48
4.5 RPA do retreinamento do CREPE. 48
4.6 Curvas de perda de treinamento e validação do treinamento do CREPE. 49
4.7 RPA do retreinamento do FCN. 50
4.8 Curvas de perda de treinamento e validação do treinamento do FCN. 51
4.9 Diagrama de treinamento otimizado do FCN. 51
4.10 RPA do treinamento corrigido do FCN. 52
4.11 Curvas de perda de treinamento e validação do treinamento corrigido do FCN. 53
4.12 Comparação do RPA para o modelo FCN após treinamento com base de música. 54
4.13 CREME: *A Convolutional Representation for Multipitch Estimation*. . 55
4.14 Diagrama para treinamento do CREME com a base de dados *MDB-stem-synth-multi*. 59
4.15 RPA do treinamento do CREME. 61
4.16 Curvas de perda de treinamento e validação do treinamento do CREME. 62
4.17 CREME: Erro por falta para teste com a base *MDB-stem-synth-multi*. 64

4.18 CREME: Alarme Falso para teste com a base *MDB-stem-synth-multi*. 65
4.19 CREME: Erro de Substituição para teste com a base *MDB-stem-synth-multi*. 66
4.20 Gráfico de comparação entre RPA do CREPE e CREME para a base de dados *Traditional Flute Dataset*. 68
4.21 Valores de RPA para teste do CREME com a base *Bach10-mf0-synth*. 69
4.22 Comparação RPA para intervalo fixo na primeira e sexta corda do violão. 73

Lista de Tabelas

3.1 RPA de estimativas do CREPE para base de dados *Traditional Flute*. 30
3.2 Média de RPA para resultados do CREPE. 31
3.3 RPA de estimativas do FCN para base de dados *Traditional Flute*. . . 34
3.4 Média de RPA para resultados do FCN. 34
3.5 RPA de estimativas do FCN para base de dados *Traditional Flute* - Novo treino. 36
3.6 Média de RPA para resultados do FCN - Novo treino. 36

4.1 RPA de estimativas do CREME para base *MDB-stem-synth-multi*. . . 63
4.2 Comparação de RPA entre CREPE e CREME para base de dados *Traditional Flute Dataset*. 67
4.3 RPA das estimativas do CREME: violão/sexta corda/intervalo crescente. 70
4.4 RPA das estimativas do CREME: violão/primeira corda/intervalo crescente. 71
4.5 RPA das estimativas do CREME: violão/sexta corda/intervalo fixo. . 72
4.6 RPA das estimativas do CREME: violão/sexta corda/intervalo fixo. . 72
4.7 RPA das estimativas do CREME: clarinete/intervalo crescente. 74
4.8 RPA das estimativas do CREME: violão e clarinete/intervalo fixo. . . 76

Capítulo 1

Introdução

A recuperação de informação musical (MIR, do inglês *music information retrieval*) pode ser definida como um campo interdisciplinar que combina elementos que vão da musicologia à ciência da computação para desenvolver métodos de recuperação e análise de informações relacionadas à música. A MIR busca a extração de informações relevantes dos dados musicais, que podem incluir sinais de áudio, partituras, letras e metadados. Isto pode facilitar, por exemplo, a organização, busca e recomendação de músicas em grandes bancos de dados, tornando mais fácil para os usuários encontrar e interagir com conteúdos musicais. A MIR é crucial na era da música digital, onde grandes quantidades de música estão disponíveis *online* e sistemas de recuperação eficientes são necessários para navegar por esses dados.

Apesar da importância da música, o processamento musical ainda é uma disciplina relativamente jovem em comparação com o processamento de fala. Enquanto os primeiros trabalhos sistemáticos em processamento de fala datam da década de 1950 [1], impulsionados pelo desenvolvimento da telefonia e sistemas de reconhecimento automático de fala, as pesquisas em processamento musical só ganharam maior consistência a partir dos anos 1990 [2]. Uma comunidade de pesquisa representada pela *international society for music information retrieval* (ISMIR), que lida sistematicamente com uma ampla gama de tópicos de análise, processamento e recuperação de música baseada analise computacional, foi formada no ano 2000 [3]. Tradicionalmente, a pesquisa musical baseada em computador tem sido conduzida principalmente com base em representações simbólicas usando notação musical ou representações MIDI. Devido à crescente disponibilidade de material de áudio digi-

talizado e à explosão do poder de computação, o processamento automatizado de sinais de áudio baseados em formas de onda está agora cada vez mais no foco dos esforços de pesquisa.

1.1 Motivação e Contexto

A indústria da música tem se beneficiado enormemente dos avanços na MIR, transformando a forma como a música é consumida, produzida e comercializada. Para os consumidores, a MIR revolucionou a descoberta musical através de sistemas de recomendação personalizados que sugerem músicas com base nos hábitos e preferências individuais de escuta [4]. Plataformas como Spotify, Apple Music e YouTube Music utilizam algoritmos sofisticados de MIR para criar *playlists* personalizadas, aumentando o engajamento e a satisfação do usuário [5]. Para artistas e produtores, as ferramentas de MIR auxiliam na produção musical oferecendo análise automatizada de música, ajudando em tarefas como mixagem, masterização e classificação de gênero [6]. A indústria também utiliza a MIR para análise de mercado, identificando tendências e entendendo as preferências do público, informando assim estratégias de *marketing* e processos de tomada de decisão [7]. Além disso, a MIR facilitou o desenvolvimento de novos modelos de negócios, como precificação dinâmica e publicidade direcionada, otimizando ainda mais a geração de receita [8].

A transcrição musical automatizada, uma aplicação da recuperação de informação musical (MIR), é o processo de converter gravações de áudio em representações simbólicas, como partituras ou arquivos MIDI. Esse processo depende fortemente de estimativas precisas de frequência fundamental. A importância da transcrição musical automatizada se estende por vários aspectos da indústria da música e da musicologia, proporcionando inúmeros benefícios a músicos, educadores e pesquisadores.

Para músicos e produtores, as ferramentas de transcrição musical automatizada podem agilizar o processo de produção e composição musical [9]. A estimativa de $F0$ (frequência fundamental de uma nota musical) permite que essas ferramentas gerem transcrições precisas de *performances* gravadas, economizando tempo e esforço em comparação com a transcrição manual [10]. Isso permite que os músicos se

concentrem mais nos aspectos criativos da composição e arranjo. Além disso, os produtores podem editar e manipular facilmente as transcrições em estações de trabalho de áudio digital, facilitando a criação de peças musicais complexas e remixagens [11].

Na educação musical, a transcrição automatizada é um ótimo recurso para estudantes e professores. Os estudantes podem usar ferramentas de transcrição para analisar e aprender com suas próprias *performances*, obtendo *insights* sobre a precisão das notas e o tempo. Os professores podem utilizar essas ferramentas para gerar partituras a partir de gravações, permitindo que forneçam aos alunos notações precisas para prática. Além disso, a transcrição automatizada auxilia no estudo de passagens musicais complexas e rápidas, ajudando os alunos a entender e executar peças desafiadoras de maneira mais eficaz.

Os musicólogos se beneficiam da transcrição automatizada, pois ela permite a análise em larga escala de obras musicais. A estimativa precisa de *pitch* (altura percebida do som, que em processamento de música pode ser considerado igual à $F0$) permite que os pesquisadores transcrevam e estudem extensos *corpora* musicais, descobrindo padrões e tendências na composição musical e nas práticas de *performance* em diferentes gêneros e períodos históricos. Isso pode levar a novas percepções sobre a evolução dos estilos musicais e a influência das mudanças culturais e tecnológicas na criação musical. A transcrição automatizada também facilita a preservação e documentação da música tradicional e folclórica, garantindo que esses artefatos culturais estejam acessíveis para estudos futuros.

A transcrição musical automatizada melhora a acessibilidade da música para indivíduos com deficiências auditivas ou aqueles que preferem representações visuais da música. Ao converter gravações de áudio em formatos visuais, como partituras, essas ferramentas tornam mais fácil para um público mais amplo se envolver e apreciar a música. Além disso, a transcrição automatizada auxilia no arquivamento de *performances* musicais, permitindo a documentação e preservação precisas de gravações de áudio. Isso é particularmente importante para *performances* ao vivo e improvisações, que podem não ter partituras escritas.

Em resumo, a transcrição musical automatizada com base na estimativa de $F0$ desempenha um papel crucial em vários aspectos da indústria da música e da academia. À medida que os algoritmos e técnicas continuam a melhorar, o impacto

da transcrição musical automatizada seguirá transformando ainda mais a forma como criamos, aprendemos e interagimos com a música.

1.2 Objetivos da Dissertação

Objetivo Geral

Apesar dos avanços significativos, a estimativa de *pitch* permanece desafiadora devido às complexidades inerentes ao áudio musical. De forma geral, este trabalho busca explorar estes desafios enfrentados pelos métodos atuais de estimativa de *pitch*, contrastando técnicas clássicas de processamento de sinal com abordagens modernas usando redes neurais convolucionais (CNNs). Métodos clássicos de processamento de sinal para estimativa de *pitch*, desde a transformada rápida de Fourier (FFT) até o algoritmo YIN, têm sido amplamente utilizados devido à sua simplicidade e eficiência computacional. No entanto, esses métodos enfrentam vários desafios [12]:

- Polifonia: Métodos clássicos têm dificuldade com música polifônica, onde várias notas são tocadas simultaneamente. Essas técnicas frequentemente falham em isolar individualmente os *pitches*, levando a erros na detecção de *pitch*.

- Ruído e Artefatos: Ruídos de fundo e artefatos de gravação podem afetar significativamente a precisão da estimativa de *pitch*. Métodos clássicos são sensíveis a esses ruídos, o que pode introduzir imprecisões nos tons detectados.

- Interferência Harmônica: Em texturas musicais complexas, harmônicos e sobretons podem interferir no processo de estimativa. Métodos clássicos podem identificar incorretamente harmônicos como frequências fundamentais.

- Faixa Dinâmica: Variações no volume e na dinâmica podem apresentar dificuldades para métodos clássicos. Passagens suaves e altas podem ser processadas de maneira diferente, afetando a consistência da estimativa de *pitch* ao longo do sinal de áudio.

Redes neurais convolucionais (CNNs) surgiram como uma ferramenta poderosa para a estimativa de *pitch*, aproveitando sua capacidade de aprender represen-

tações hierárquicas a partir de dados de áudio. Apesar de suas vantagens, métodos baseados em CNN também enfrentam vários desafios:

- Requisitos de Dados: Treinar CNNs requer grandes conjuntos de dados anotados para alcançar alta precisão. Obter esses conjuntos de dados, particularmente para diversos gêneros e estilos musicais, pode ser desafiador e exigir muitos recursos, além de consumir muitas horas de pessoas especializadas para gerar a base de dados anotada.

- Complexidade Computacional: CNNs são computacionalmente intensivas, exigindo considerável poder de processamento e memória. Essa complexidade pode limitar sua aplicação em ambientes em tempo real ou com recursos limitados, como dispositivos móveis.

- Interpretabilidade: A natureza de caixa-preta das CNNs dificulta a interpretação de como elas tomam decisões. Essa falta de transparência pode ser uma desvantagem para entender erros e melhorar o desempenho do modelo em cenários específicos.

- Polifonia: Embora as CNNs tenham mostrado melhorias sobre os métodos clássicos no manejo da polifonia, estimar com precisão *pitches* em música altamente polifônica ainda é desafiador. A presença de várias notas simultâneas pode ainda confundir o modelo, levando a imprecisões.

Abordar os desafios associados à estimativa de *pitch* baseada no uso de CNNs pode dar uma contribuição relevante para o avanço do campo de recuperação de informação musical (MIR). Este trabalho visa a otimizar a eficiência computacional, aprimorar a generalização e lidar com complexidades polifônicas. Ao enfrentar esses desafios, busco desenvolver um modelo de CNN que possa ser utilizado efetivamente em transcrição musical automática.

Objetivos Específicos

Estimativa de *pitch* monofônica é uma área bem pesquisada dentro do campo de MIR, na qual modelos de estimativa clássicos e redes neurais têm demonstrado sucesso significativo. No entanto, o desafio aumenta ao se estender esses métodos para

música polifônica, onde múltiplos *pitches* devem ser estimados simultaneamente. Este trabalho aborda esse problema.

O objetivo principal deste trabalho é adaptar a arquitetura e o processo de treinamento de uma rede neural voltada para a estimativa de *pitch* monofônica para ser capaz de estimar múltiplos *pitches* no contexto música polifônica. Como base será utilizado um modelo de rede neural convolucional bem estabelecido para estimativa de *pitch* monofônica, denominado CREPE (do inglês, *convolutional representation for pitch estimation*) [13]. Também será avaliado o modelo chamado FCN (do inglês, *fully convolutional network for pitch estimation*), que também é baseado no CREPE para estimativa de *pitch*.

Outro objetivo deste trabalho é a criação de uma base de dados para treinamento de rede neural com foco em estimativa de múltiplos *pitches*. Para isto, foi utilizada a base de dados *MDB-stem-synth*, que possui áudios musicais com anotações de frequências isoladas ao longo do tempo. Com base no *MDB-stem-synth*, o trabalho pretende efetuar combinações entre áudios para gerar uma base de dados de treinamento polifônico contendo anotações de múltiplos *pitches*.

Em resumo, sobre os objetivos específicos:

(1) Gerar um conjunto de dados de áudio anotados para estimativa de multiplos *pitches*, com base nas bases de dados *MedleyDB* e *MDB-STEM-Synth*;

(2) Implementar no *framework TensorFlow* e linguagem de programação Python os principais modelos de redes neurais convolucionais existentes (CREPE e FCN) para investigar seus limites de desempenho na estimação de *pitch*;

(3) Implementar no *framework TensoFlow* e linguagem de programação Python um modelo de CNN para realizar estimativa de multiplos *pitches*, baseado no modelo CREPE;

(4) Disponibilizar repositório dos códigos desenvolvidos.

1.3 Organização da Dissertação

Esta dissertação está organizada conforme abaixo:

- O Capítulo 1 introduz conceitos sobre a área de recuperação de informação musical (MIR), apresentando o contexto que explica a relevância desta área de pesquisa, juntamente com a motivação e objetivos para o presente trabalho de estimativa de *pitch*.

- O Capítulo 2 trata da fundamentação teórica utilizada no trabalho proposto. São abordados os principais conceitos relacionados a processamento de sinal de audio, noções sobre *pitch* e frequência fundamental $F0$, os algoritmos clássicos de estimativa de *pitch* YIN e pYIN, e por fim detalhes sobre o principal modelo de rede neural convolucional no estado-da-arte para estimativa de *pitch*, o CREPE (*convolutional representation for pitch estimation*)

- No Capítulo 3 apresentamos a proposta de trabalho, abordando as ideias relacionadas à estimativa de múltiplos *pitches* usando rede neural convolucional, delimitando as atividades e o que será produzido nesta dissertação.

- No Capítulo 4 serão descritos os experimentos realizados e os respectivos resultados obtidos ao longo do trabalho realizado.

- O Capítulo 5 é dedicado à conclusão do trabalho, relatando seus resultados, possíveis atividades futuras e melhorias que podem otimizar o desempenho do sistema desenvolvido.

Capítulo 2

Fundamentos Teóricos

2.1 Noções sobre *Pitch*

Pitch é uma propriedade subjetiva dos sons que nos permite classificá-los como *mais altos* ou *mais baixos* em frequência. É um aspecto fundamental de como experimentamos e interpretamos a música. Embora o *pitch* seja frequentemente associado à frequência fundamental ($F0$) de um som, ele não é determinado apenas por isso [14]. Em vez disso, a percepção de *pitch* é influenciada por uma interação complexa dos componentes de frequência do som, o sistema auditivo do ouvinte e fatores contextuais.

A frequência fundamental ($F0$) é a frequência mais baixa de uma sinal periódico e geralmente é considerada o principal determinante do *pitch* [15]. Para uma nota musical, a $F0$ corresponde ao *pitch* que tipicamente identificamos com essa nota. Por exemplo, a nota Lá 4 (A4) tem uma frequência fundamental de 440 Hz, que é percebida como um *pitch* específico [16].

No entanto, a relação entre *pitch* e $F0$ nem sempre é direta. Em tons complexos (usuais em música), o *pitch* percebido é influenciado pelos harmônicos ou por parciais não-harmônicas (no caso de sons não perfeitamente harmônicos) [17,18]. Os harmônicos são múltiplos inteiros da frequência fundamental que contribuem para o *timbre* [19], que é a característica perceptiva de um som que permite distingui-lo de outros sons com mesma altura e intensidade, como quando um piano e um violino tocam a mesma nota, mas ainda assim conseguimos perceber que são instrumentos diferentes [20, 21].

Nos sons do mundo real, especialmente aqueles produzidos por instrumentos musicais, o que ouvimos como *pitch* resulta da integração de múltiplos componentes de frequência. O sistema auditivo humano tem a notável capacidade de usar os harmônicos para reforçar a percepção de *pitch*, mesmo quando a frequência fundamental é ausente [18, 22, 23]. Este fenômeno é conhecido como o efeito da *fundamental ausente*, em que o cérebro infere o *pitch* de um som a partir da série harmônica [24].

Por exemplo, se um som tem harmônicos a 300 Hz, 400 Hz e 500 Hz, mas nenhum componente real a 100 Hz, os ouvintes perceberão um *pitch* correspondente a 100 Hz, o que ilustra como o sistema auditivo processa tons complexos: com base nas relações harmônicas em vez de apenas na frequência fundamental [25, 26].

A percepção de *pitch* também é influenciada por vários fatores contextuais e perceptivas. Os sons circundantes, o contexto musical e a experiência e treinamento do ouvinte podem afetar como o *pitch* é percebido [27–29]. A própria intensidade afeta a percepção de *pitch*: uma nota musical reproduzida com intensidades muito diferentes pode ser percebida como tendo alturas diferentes [30].

Além disso, fatores psicoacústicos, como o *pitch* de tons inarmônicos ou em ambientes com ruído, destacam a complexidade da percepção de *pitch*. Tons inarmônicos, que não têm sobretons harmônicos relacionados por múltiplos inteiros, ainda podem produzir uma percepção de *pitch* estável, embora menos clara do que tons harmônicos (é o caso de um sino, por exemplo). Efeitos de ruído e mascaramento também podem alterar a percepção de *pitch*, como resultado da robustez e adaptabilidade do sistema auditivo [29, 31, 32].

Na síntese musical e de áudio, em geral, criar sons naturais e realistas requer controle preciso tanto das frequências fundamentais quanto dos harmônicos. Para tecnologias de processamento vocal e correção de *pitch*, como *Auto-Tune*, detectar e manipular *pitch* com precisão enquanto preserva harmônicos naturais é essencial para alcançar os efeitos musicais desejados [33–35].

Em resumo, *pitch* é uma qualidade perceptiva que, embora frequentemente relacionada à frequência fundamental, envolve uma interação complexa de componentes harmônicos e fatores contextuais. A capacidade do sistema auditivo humano de inferir *pitch* a partir de harmônicos e adaptar-se a várias condições de escuta destaca a sofisticação da percepção de *pitch*. No entanto, exceto por algumas raras

exceções, ele pode ser quantificado pela frequência fundamental, e assim as duas grandezas são frequentemente usados de forma intercambiável fora de estudos psicoacústicos.

2.2 Transformada de Fourier de Curta Duração

A transformada de Fourier de curta duração (STFT, do inglês *short-time Fourier transform*) é uma ferramenta fundamental no processamento de sinais, particularmente útil para analisar sinais não estacionários cujo conteúdo de frequência muda ao longo do tempo. Ao contrário da transformada de Fourier padrão, que fornece uma representação de frequência para um sinal inteiro, a STFT oferece uma representação tempo-frequencial, permitindo-nos ver como o conteúdo espectral de um sinal evolui.

A STFT é particularmente vantajosa para analisar sinais musicais devido à estrutura harmônica inerente aos sons musicais (tonais). Sinais musicais são caracterizados pelo seu conteúdo harmônico, onde cada nota normalmente é composta por uma frequência fundamental e seus múltiplos inteiros, conhecidos como harmônicos. Esses harmônicos contribuem para o timbre ou cor do som, distinguindo um instrumento de outro mesmo quando tocando a mesma nota.

Nesse contexto, a STFT é uma escolha natural para análise musical devido à sua capacidade de resolver e permitir visualizar o conteúdo harmônico dos sinais musicais e rastrear suas variações temporais. Sua flexibilidade e representação intuitiva fazem dela uma excelente ferramenta, amplamente utilizada nos trabalhos da área de recuperação de informação musical.

Para realizar uma STFT, o sinal é dividido em segmentos curtos sobrepostos (ou janelas), e a transformada de Fourier é aplicada a cada segmento. Este processo fornece um espectro de frequência localizado no tempo para cada segmento. O resultado é uma representação bidimensional do sinal nos domínios do tempo e da frequência.

A transformada de Fourier de tempo curto de um sinal contínuo $x(t)$ é defi-

nida como

$$\text{STFT}\{x(t)\}(t,\omega) = X(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(t-\tau)e^{-j\omega\tau}d\tau, \tag{2.1}$$

onde $x(t)$ é o sinal de entrada; $w(t)$ é a função janela, tipicamente uma janela de Hamming ou Hanning, usada para segmentar o sinal; ω é a frequência angular e τ é a variável de deslocamento temporal.

Para converter a STFT de tempo contínuo para a STFT de tempo discreto, amostramos o sinal a uma taxa de amostragem F_s da seguinte maneira: seja $x[n] = x(nT_s)$, onde $T_s = \frac{1}{F_s}$ é o período de amostragem. A STFT discreta é então dada por

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mH]w[n]e^{-j\frac{2\pi}{N}kn}, \tag{2.2}$$

onde m é o índice do quadro; k é o índice da banda de frequência; H é o tamanho do salto, determinando a sobreposição entre quadros adjacentes e N é o comprimento da janela.

A frequência fundamental ($F0$) é a menor frequência (não nula) de uma forma de onda periódica. Pode ser estimada a partir dos picos harmônicos detectados no espectro de magnitude obtido a partir da STFT. Se os harmônicos forem detectados nas frequências $f_1, f_2, \ldots, f_n$, a frequência fundamental f_0 pode ser inferida de

$$f_0 = \frac{F_s}{N} \operatorname{argmax}_k |X(m,k)|, \tag{2.3}$$

onde $\operatorname{argmax}_k |X(m,k)|$ é a banda de frequência com a maior magnitude na STFT.

Essa estimativa da frequência fundamental ($F0$) usando os picos de cada quadro na transformada de Fourier de tempo curto (STFT) é um método simples com fraquezas que podem afetar sua precisão e confiabilidade. Essa estratégia pode falhar em situações em que a fundamental apresenta menor amplitude em relação a harmônicos superiores, levando a estimativas incorretas. Métodos mais avançados baseados em medidas de saliência harmônica [36] exploram a regularidade do espaçamento entre harmônicos e a coerência espectral, permitindo distinguir a fundamental mesmo em cenários em que ela não é a componente mais evidente no espectro.

De forma geral, podemos citar algumas das principais limitações em modelos

baseados em STFT:

Resolução de Frequência

A resolução de frequência da STFT é determinada pelo comprimento da janela N. Uma janela mais longa proporciona melhor resolução de frequência em detrimento à resolução temporal. Inversamente, uma janela mais curta melhora a resolução temporal, mas diminui a resolução de frequência. Esse compromisso pode dificultar a estimativa precisa de $F0$, especialmente em sinais de alta complexidade, que requerem alta resolução tanto nos domínios do tempo quanto da frequência. Por exemplo, com um comprimento de janela N, o espaçamento dos *bins* de frequência é $\Delta f = \frac{F_s}{N}$. Se a frequência fundamental não estiver alinhada com os *bins* de frequência, isso pode levar a erros de estimativa [37].

Sobreposição Harmônica

Em sinais polifônicos ou ricos em harmônicos, múltiplos harmônicos e sobretons podem se sobrepor, dificultando a distinção entre a frequência fundamental e seus harmônicos. A presença de múltiplos picos próximos uns dos outros no espectro de magnitude pode levar à identificação incorreta de $F0$, particularmente se os harmônicos forem mais fortes que a frequência fundamental [38].

Ruído

Ruído no sinal de áudio introduzem picos adicionais no espectro de magnitude da STFT, que podem ser confundidos com componentes harmônicos. Ruído de fundo, imperfeições na gravação e até mesmo artefatos específicos de instrumentos podem interferir na detecção precisa dos picos, levando a uma estimativa errônea de $F0$ [39].

Efeitos de Janelamento

A escolha da função janela e seu comprimento podem introduzir vazamento espectral e lóbulos laterais, que afetam a clareza e precisão dos picos no espectro de magnitude. Embora janelas como Hamming ou Hanning reduzam o vazamento em comparação com uma janela retangular, não o eliminam completamente. O

vazamento espectral pode desfocar os picos, tornando mais difícil identificar com precisão os componentes de frequência [40].

Faixa Dinâmica

Variações na amplitude dentro do sinal podem afetar a detecção dos picos. Sons mais suaves produzem picos mais fracos, difíceis de detectar com precisão, enquanto sons mais altos dominam o espectro e obscurecem outros picos relevantes. Esse problema de faixa dinâmica pode gerar estimativas inconsistentes de $F0$ em diferentes quadros, particularmente em músicas com grandes variações dinâmicas [41].

Esmaecimento Temporal

O tamanho fixo da janela da STFT não se adapta às características variáveis do sinal. Em sinais que mudam rapidamente, uma janela longa pode esmaecer os detalhes temporais, enquanto uma janela curta pode perder detalhes sutis de frequência. Esse esmaecimento temporal pode levar a uma estimativa imprecisa de $F0$, especialmente em passagens musicais rápidas ou sinais com predominância de componentes transitórios [42].

2.3 Algoritmos para Detecção de *Pitch* - YIN e pYIN

Vários algoritmos foram desenvolvidos para a detecção de *pitch*, variando de abordagens simples no domínio do tempo a métodos complexos no domínio da frequência [43]. Entre os algoritmos de domínio do tempo mais conhecidos e amplamente utilizados estão o YIN e sua extensão probabilística, o pYIN. O algoritmo YIN [44], introduzido por Alain de Cheveigné e Hideki Kawahara em 2002, é renomado por sua precisão e robustez na detecção de *pitch*. Ele melhora os métodos tradicionais baseados em autocorrelação ao usar uma função de diferença para minimizar erros devido a ambiguidades no período do *pitch*. O algoritmo pYIN (YIN probabilístico) [45], proposto por Matthias Mauch e Simon Dixon em 2014, estende

o algoritmo YIN incorporando modelagem probabilística. Este aprimoramento aumenta a confiabilidade da detecção de *pitch*, especialmente em ambientes ruidosos ou ao lidar com sinais de áudio complexos. O pYIN usa um modelo oculto de Markov (HMM) para considerar a probabilidade de transições entre diferentes candidatos a *pitch*, proporcionando assim uma estimativa de *pitch* mais precisa e robusta.

2.4 Algoritmo YIN

Tradicionalmente, muitos algoritmos para estimação de *pitch* dependem da representação do áudio pela transformada de Fourier de curta duração (STFT). No entanto, conforme mencionado anteriormente, esta abordagem enfrenta vários desafios, como o problema resolução de frequência devido ao dilema do tamanho da janela e ao vazamento espectral (quando a energia de um sinal vaza entre componentes de frequência adjacentes). Além disso, existe o problema da complexidade computacional. A STFT envolve operações matemáticas complexas, incluindo janelamento, transformadas de Fourier e seleção de picos no domínio da frequência. Essas operações podem ser intensivas em termos computacionais, tornando a estimativa de *pitch* em tempo real desafiadora.

A principal distinção entre o algoritmo YIN [44] e muitos outros métodos de estimativa de *pitch* é depender da análise no domínio do tempo em vez da análise no domínio da frequência (STFT). O fato de o algoritmo YIN adotar uma abordagem diferente, operando diretamente no domínio do tempo, oferece algumas vantagens:

- Alta Resolução Temporal: Métodos de domínio do tempo operam diretamente no sinal de áudio, permitindo uma resolução temporal mais alta sem as limitações do tamanho da janela. Isso os torna particularmente eficazes para sinais com variações rápidas de *pitch*.

- Carga Computacional Reduzida: Métodos de domínio do tempo frequentemente envolvem operações mais simples, como cálculos de diferenças e interpolação, que podem ser mais eficientes do que a análise espectral complexa exigida pela STFT.

- Robustez ao Ruído: Métodos de domínio do tempo podem ser mais robustos

a certos tipos de ruído e artefatos que afetam as representações no domínio da frequência. Ao analisar diretamente a forma de onda, eles podem lidar melhor com sinais de áudio do mundo real com níveis de ruído variados.

Para apresentarmos o algoritmo YIN é necessário revisitar quatro equações: a autocorreção (ACF), a função de diferença (DF), a função de diferença normalizada pela média cumulativa (CMNDF) e a interpolação parabólica. As equações são descritas conforme a seguir.

A função de autocorrelação (ACF) é definida por:

$$R(\tau) = \sum_{t=0}^{N-\tau-1} x(t)x(t+\tau), \tag{2.4}$$

onde $x(t)$ é o sinal de entrada, τ é o atraso de tempo e N o tamanho da janela. Esta equação calcula a similaridade do sinal em diferentes atrasos para identificar possíveis periodicidades relacionadas a frequências existentes na janela sob análise.

Função de Diferença (DF)

O algoritmo YIN usa uma função de diferença modificada em vez de ACF:

$$d(\tau) = \sum_{j=1}^{N} \left(x(j) - x(j+\tau)\right)^2. \tag{2.5}$$

Enquanto a ACF mede a similaridade entre sinais em diferentes atrasos, a função de diferença usada no YIN foca nas diferenças, tornando-a mais robusta para sinais periódicos.

Função de Diferença Normalizada pela Média Cumulativa (CMNDF)

A DF mede a diferença quadrática entre o sinal e uma versão atrasada dele mesmo. Pequenos valores de $d(\tau)$ indicam alta similaridade e, assim, correspondem a potenciais períodos do sinal; mas essa função por si só é insuficiente para uma detecção confiável de *pitch*.

A CMNDF é definida como:

$$d'(\tau) = \begin{cases} 1, & \text{se } \tau = 0; \\ d(\tau) \left(\frac{1}{\tau} \sum_{j=1}^{\tau} d(j) \right)^{-1}, & \text{em caso contrário.} \end{cases} \tag{2.6}$$

A CMNDF melhora a detecção de *pitch* ao normalizar a função de diferença para reduzir a influência de sub-harmônicos. Quando a média cumulativa é usada como divisor, valores maiores de τ são penalizados, reduzindo sua proeminência, a menos que consistentemente tenham valores baixos de diferença. Isso ajuda a distinguir a frequência fundamental real de seus sub-harmônicos. Sem normalização, a função de diferença pode exibir mínimos locais em sub-harmônicos. Ao normalizá-la, a CMNDF garante que apenas os atrasos correspondentes à periodicidade real do sinal (frequência fundamental) tenham mínimos destacados, facilitando a identificação do *pitch* verdadeiro.

Interpolação Parabólica

Para refinar o período detectado, a interpolação parabólica é aplicada ao redor do valor mínimo da CMNDF:

$$\hat{\tau} = \tau_{\min} + \frac{d'(\tau_{\min} - 1) - d'(\tau_{\min} + 1)}{2\left(2d'(\tau_{\min}) - d'(\tau_{\min} - 1) - d'(\tau_{\min} + 1)\right)}. \tag{2.7}$$

A interpolação parabólica refina esta estimativa de período usando uma aproximação matemática para localizar um período mais preciso entre os atrasos de tempo. Em vez de usar $\tau_{\min}$ diretamente, uma curva parabólica é ajustada a este mínimo e aos seus vizinhos imediatos para encontrar uma estimativa mais precisa.

2.5 Algoritmo pYIN

O algoritmo pYIN (do inglês, *probabilistic* YIN) [45] aprimora o algoritmo YIN ao incorporar modelagem probabilística, tornando a detecção de *pitch* mais robusta e precisa, especialmente em ambientes ruidosos ou com sinais de áudio de maior complexidade. O pYIN introduz uma estrutura probabilística para detecção de *pitch*. Em vez de confiar apenas em cálculos determinísticos, os candidatos ao *pitch* são modelados como estados em um modelo oculto de Markov (HMM) [46],

onde cada estado corresponde a um *pitch* possível. O algoritmo de Viterbi [47] é utilizado para encontrar o caminho mais provável através dos estados, considerando toda a sequência de observações. Isso permite um rastreamento de *pitch* mais preciso ao longo do tempo, levando em consideração a continuidade e a probabilidade de transições de *pitch*.

Antes da apresentação do algoritmo pYIN, abordaremos alguns aspectos importantes do método HMM e do algoritmo de Viterbi. Em relação ao HMM, pode-se dizer que é um modelo estatístico onde o sistema é assumido como um processo de Markov com estados ocultos. Em pYIN, os estados ocultos representam candidatos a pitch, e os dados observados são as características do sinal de áudio computadas durante o pré-processamento. Desta forma, o HMM é aplicado conforme a seguir:

- **Estados (S)**: Representam candidatos a *pitch*.
- **Observações (O)**: Características computadas do sinal de áudio (por exemplo, valores de CMNDF).
- **Probabilidades de transição (A)**: Probabilidades de transição de um estado de *pitch* para outro.
- **Probabilidades de emissão (B)**: Probabilidades de observar uma característica específica dado um estado de *pitch*.

O algoritmo de Viterbi [47] é usado para encontrar a sequência mais provável de estados ocultos (candidatos a *pitch*), dada a sequência observada (características de áudio). Utiliza-se programação dinâmica para calcular eficientemente o caminho ótimo, que podemos resumir as etapas conforme a seguir:

- **Inicialização**: O processo começa definindo a probabilidade inicial de cada estado oculto com base na probabilidade inicial de estar em cada estado e na probabilidade de observar a primeira característica a partir desse estado:

$$\delta_1(i) = \pi_i \, b_i(O_1), \quad 1 \leq i \leq N. \tag{2.8}$$

- **Recursão**: Em cada etapa subsequente, a probabilidade de estar em cada estado é atualizada com base na melhor probabilidade do estado anterior mul-

tiplicada pela probabilidade de transição e pela probabilidade de observar a característica atual a partir do novo estado.

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \; b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N. \tag{2.9}$$

- **Término**: A última etapa consiste em encontrar a maior probabilidade dentre os estados finais, que corresponde à probabilidade da sequência de estados mais provável.

$$P^* = \max_{1 \leq i \leq N} \delta_T(i). \tag{2.10}$$

- ***Backtracking***: Para encontrar a sequência mais provável de estados, retrocede-se do estado com maior $\delta_T(i)$ para reconstruir o caminho.

Função de Densidade de Probabilidade (PDF) Gaussiana de Emissão

$$b_j(O_t) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(-\frac{(O_t - \mu_j)^2}{2\sigma_j^2} \right), \tag{2.11}$$

onde μ_j e σ_j são a média e o desvio padrão das observações para o estado j.

Probabilidades de Transição

$$a_{ij} = P(S_{t+1} = j \mid S_t = i), \tag{2.12}$$

onde S_t e S_{t+1} são os estados nos tempos t e $t+1$, respectivamente.

2.6 Rede Neural CREPE

As redes neurais convolucionais (CNNs) [48] são uma classe de modelos de aprendizado profundo especificamente projetados para processar dados com uma topologia em grade, como imagens. No entanto, são adaptáveis para processar sinais unidimensionais, como sinais de áudio. As CNNs são compostas por múltiplas camadas, predominantemente pelas camadas denominadas convolucionais, camadas de *pooling* e camadas totalmente conectadas, que permitem aprender e reconhecer padrões nos dados de entrada.

A força das CNNs reside na sua capacidade de aprender, de forma automática e adaptativa, hierarquias espaciais de características dos dados de entrada, tornando-as uma ferramenta poderosa para muitos problemas de reconhecimento de imagens, e também para análise de sinais de áudio, como processamento de linguagem natural e música.

No campo da recuperação de informação musical (MIR), os pesquisadores estão cada vez mais focados no uso de redes neurais em vez dos métodos clássicos de processamento digital de sinais (DSP, do inglês *digital signal processing*). Destacamos algumas razões para esta mudança [49]:

- Aprendizado de características: Diferentemente do DSP clássico, que requer extração manual de características, redes neurais podem aprender automaticamente características relevantes dos dados de áudio brutos. Essa habilidade reduz significativamente a necessidade de expertise específica do domínio e permite a descoberta de relações complexas nos dados.

- Adaptabilidade e generalização: Redes neurais generalizam mais efetivamente entre diferentes tipos de música e sinais de áudio do que os métodos clássicos. Podem ser treinadas em conjuntos de dados diversos, tornando-as robustas a variações na qualidade do áudio, condições de gravação e gêneros musicais.

- Desempenho e precisão: Modelos de aprendizado de máquina, particularmente arquiteturas de aprendizado profundo como CNNs, têm demonstrado desempenho superior em tarefas como estimativa de *pitch*. Sua capacidade de capturar padrões nos sinais de áudio leva a maior precisão e melhor desempenho comparado às técnicas tradicionais de DSP.

- Escalabilidade: Redes neurais podem lidar com dados em grande escala e se beneficiar dos avanços em recursos computacionais, como GPUs, para processar grandes quantidades de dados de áudio de maneira eficiente.

Em resumo, a transição do DSP clássico para aprendizado de máquina e redes neurais no MIR é motivada pela necessidade de soluções mais automatizadas, adaptáveis e de alto desempenho para processar e entender sinais de áudio complexos.

O CREPE (do inglês, *convolutional representation for pitch estimation*) é um modelo de rede neural convolucional (CNN) desenvolvido especificamente para estimar a frequência fundamental ($F0$) em sinais de áudio. Foi introduzido por Kim, Salamon, Li e Bello em 2018 [13] para fornecer uma solução precisa e eficiente para a detecção de *pitch*.

Embora os modelos YIN e pYIN sejam amplamente reconhecidos por sua precisão e eficiência na estimativa de *pitch* através da análise no domínio do tempo, o modelo de redes neurais convolucionais (CNN) tem se destacado como uma alternativa poderosa. Apesar de CREPE e os modelos YIN operarem diretamente no domínio do tempo, CREPE apresenta vantagens que o tornam uma escolha superior em muitos cenários:

- Precisão aprimorada: CREPE utiliza uma rede neural convolucional treinada em uma vasta quantidade de dados de áudio para aprender representações robustas e discriminativas do *pitch*. Isso permite uma precisão de estimativa de *pitch* que muitas vezes supera os métodos tradicionais baseados em domínio do tempo, como YIN e pYIN.

- Capacidade de generalização: Devido à sua natureza baseada em aprendizado profundo, CREPE é capaz de generalizar melhor para diferentes tipos de sinais de áudio, incluindo aqueles com ruído significativo ou características harmônicas complexas. O modelo pode adaptar-se a uma variedade de fontes sonoras sem necessidade de ajustes específicos.

- Robustez a condições adversas: CREPE é notavelmente robusto a ruídos e artefatos que normalmente prejudicariam os métodos de domínio do tempo. A capacidade da CNN de extrair características relevantes mesmo em condições adversas torna o CREPE uma ferramenta eficaz para estimativa de *pitch* em ambientes do mundo real.

- Treinamento e adaptabilidade: O modelo pode ser continuamente aprimorado com mais dados de treinamento, tornando-o cada vez mais preciso ao longo do tempo. Isso também permite a adaptação a novas condições ou tipos de sinais que podem não ter sido considerados durante o desenvolvimento inicial.

O CREPE consiste em uma rede neural convolucional profunda que opera diretamente no sinal de áudio no domínio do tempo para produzir uma estimativa de *pitch*. Um diagrama em bloco da arquitetura do CREPE é fornecido na Figura 2.1. A entrada é um trecho de 1024 amostras do sinal de áudio no domínio do tempo, utilizando uma taxa de amostragem de 16 kHz. Existem seis camadas convolucionais que resultam em uma representação latente de 2048 dimensões, que é então conectada densamente à camada de saída com ativações sigmoidais correspondentes a um vetor de saída $\hat{y}$ de 360 dimensões. A partir disso, a estimativa de *pitch* resultante é calculada de forma determinística.

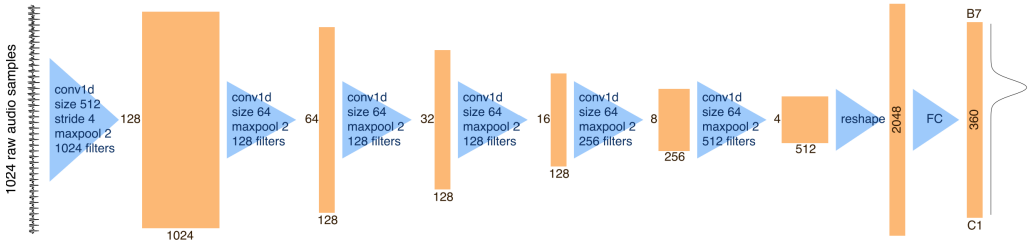

Figura 2.1: Diagrama em blocos da arquitetura do CREPE.

Cada um dos 360 nós na camada de saída corresponde a um valor de *pitch* específico, definido em *cents*. *Cent* é uma unidade que representa intervalos musicais relativos a um *pitch* de referência f_{ref} em *Hertz*, definida como uma função da frequência f em *Hertz*:

$$c(f) = 1200 \log_2 \left(\frac{f}{f_{\text{ref}}} \right), \tag{2.13}$$

onde é usado $f_{\text{ref}} = 10$ Hz nos experimentos. Esta unidade fornece uma escala de *pitch* logarítmica onde 100 *cents* equivalem a um semitom. Os 360 valores de *pitch* são denotados como $c_1, c_2, \ldots, c_{360}$, e são selecionados de modo que cubram seis oitavas com intervalos de 20 *cents* entre C1 e B6, correspondendo a $32,7$ Hz e $1975,5$ Hz. A estimativa de *pitch* resultante $\hat{c}$ é a média ponderada dos *pitches* associados c_i de acordo com a saída $\hat{y}$, que fornece a estimativa de frequência em *Hertz*:

$$\hat{c} = \frac{\sum_{i=M-4}^{M+4} \hat{y}_i c_i}{\sum_{i=M-4}^{M+4} \hat{y}_i}, \quad M = \text{argmax}_i(\hat{y}_i) \tag{2.14}$$

$$\hat{f} = f_{\text{ref}} 2^{\hat{c}/1200}. \tag{2.15}$$

As saídas alvo usadas para treinar o modelo são vetores de 360 dimensões,

onde cada dimensão representa um *bin* de frequência cobrindo 20 *cents* (o mesmo que a saída do modelo). O *bin* correspondente à frequência fundamental verdadeira recebe uma magnitude de 1. Como em [50], para suavizar a penalidade por previsões quase corretas, o alvo é suavizado com uma gaussiana na frequência, de modo que a energia ao redor de uma frequência verdadeira decai com um desvio padrão de 25 *cents*:

$$y_i = \exp\left(-\frac{(c_i - c_{\text{true}})^2}{2 \cdot 25^2}\right). \tag{2.16}$$

Dessa forma, altas ativações na última camada indicam que o sinal de entrada provavelmente tem um *pitch* próximo aos *pitches* associados aos nós com altas ativações. A rede é treinada para minimizar a entropia cruzada binária entre o vetor alvo y e o vetor estimado $\hat{y}$:

$$L(y, \hat{y}) = \sum_{i=1}^{360} \left(-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)\right), \tag{2.17}$$

onde tanto y_i quanto $\hat{y}_i$ são números reais entre 0 e 1. Esta função de perda é otimizada usando o otimizador ADAM, com a taxa de aprendizado de 0,0002. O melhor modelo é selecionado após o treinamento até que a precisão de validação não melhore mais por 32 épocas, onde uma época consiste em 500 lotes de 32 exemplos selecionados aleatoriamente do conjunto de treinamento. Cada camada convolucional é precedida por normalização em lote e seguida por uma camada de *dropout* com a probabilidade de *dropout* de 0,25.

2.6.1 Bases de Dados

O CREPE foi treinado e avaliado usando um grande conjunto de dados apresentado a seguir, que inclui uma coleção diversificada de amostras de áudio. O conjunto de dados contém gravações musicais, garantindo que o modelo possa generalizar bem para diferentes cenários. Além disso, dados sintéticos gerados usando vários instrumentos e vozes cantadas também foram empregados para estender o processo de treinamento.

- *MIR-1K* [51]: Esta base de dados contém 1000 trechos de áudio de música cantada em chinês, coletados de 110 canções de *karaoke*. Cada trecho contém

um vocal e uma trilha instrumental.

- *Bach10* [52]: A base Bach10 consiste em 10 gravações de quartetos de cordas tocando trechos de obras de J.S. Bach. As gravações foram feitas separadamente para cada instrumento (violino, viola, violoncelo e contrabaixo).

- *RWC-Synth* [45]: Esta base faz parte do *Real World Computing Music Database* [53] e foi sintetizada para conter exemplos de sinais de áudio com *pitch* conhecido.

- *MedleyDB* [54]: É uma base de dados rica e diversificada, composta por 122 faixas musicais com anotações detalhadas de *pitch* e separações de *stems* (faixas de áudio isoladas).

- *MDB-stem-Synth* [55]: Esta base é uma versão sintética das gravações do *MedleyDB*.

- *NSynth* [56]: É uma base de dados grande e diversificada que contém 305979 notas individuais de instrumentos musicais acústicos e eletrônicos. A base foi criada para treinamento de modelos de aprendizado profundo voltados para a síntese e análise de áudio.

Para avaliar objetivamente o CREPE e comparar seu desempenho com algoritmos alternativos, foi necessário utilizar dados de áudio com anotações de referência perfeitas. Isso é importante, dado que o desempenho dos algoritmos comparados já é muito alto. Diante disso, não foi utilizado um conjunto de dados como o *MedleyDB*, uma vez que seu processo de anotação inclui correções manuais que não garantem uma correspondência perfeita entre a anotação e o áudio, podendo ser afetado pela subjetividade humana. Para garantir uma avaliação perfeitamente objetiva, foram usados dois desses conjuntos de dados de áudio sintetizado com controle perfeito sobre a $F0$ do sinal resultante: o primeiro, *RWC-synth*, contém 6,16 horas de áudio sintetizado a partir do *Real World Computing Music Database* e é utilizado para avaliar o pYIN, e o segundo conjunto de dados é uma coleção de 230 *stems* (pistas de áudio isoladas) monofônicos extraídos do *MedleyDB* e ressintetizados. Este conjunto de dados consiste em 230 faixas com 25 instrumentos, totalizando 15,56 horas de áudio, referido como *MDB-stem-synth*.

O CREPE demonstrou um desempenho no estado da arte em tarefas de estimativa de *pitch*, superando estimadores de *pitch* populares, como pYIN. Sua arquitetura convolucional permite capturar detalhes minuciosos no sinal de áudio, levando a previsões de $F0$ precisas e robustas.

O modelo foi treinado usando validação cruzada *5-fold*, com uma divisão de 60/20/20 para treino, validação e teste, respectivamente. Para *MDB-stem-synth*, foram utilizados *folds* condicionais por artista, para evitar treinar e testar no mesmo artista, o que pode resultar em desempenho artificialmente elevado devido a efeitos do artista ou do álbum [57]. A avaliação da estimativa de altura de um algoritmo é medida em precisão de altura crua (RPA) e precisão de croma crua (RCA) com margens de 50 *cents* [58]. Essas métricas medem a proporção de *frames* na saída para os quais a saída do algoritmo está dentro de 50 *cents* (um quarto de tom) da $F0$ verdadeira. Foi utilizada a implementação de referência fornecida no *mir eval* [59] para calcular as métricas de avaliação.

Capítulo 3

Metodologia Proposta

3.1 Introdução

O CREPE é um modelo de rede neural convolucional considerado estado da arte para estimativa de *pitch*. Apesar da existência de vários modelos de estimativa de *pitch*, o CREPE foi escolhido sem testar outros modelos devido ao seu reconhecimento como uma solução de última geração na área. A decisão de focar exclusivamente no CREPE baseia-se em seu desempenho superior relatado em vários estudos de referência, que destacam sua robustez e precisão em tarefas de estimativa de *pitch* monofônico. Esta abordagem focada permite uma análise aprofundada e potencial melhoria de um modelo líder, com o objetivo final de aproveitar seus pontos fortes e contornar suas fraquezas em contextos polifônicos.

3.2 Estudo de modelos monofônicos

Neste estudo pretendemos avaliar o desempenho do CREPE. Adicionalmente, também avaliamos um modelo chamado FCN (do inglês, *fully-convolutional network for pitch estimation of speech signals*) [60]. O FCN foi projetado usando o CREPE como referência para atuar com sinais de fala. Este modelo mantém boa parte da arquitetura original e equações para determinar o valor de frequência estimado. O FCN busca otimizar recursos computacionais, realizando modificações para reduzir complexidade de cálculos e tempo de execução. Desta forma, o FCN também se apresenta como uma boa referência para estudo, tendo em vista já ter explorado o

CREPE e proposto melhorias no modelo.

Por fim, é importante mencionar que não buscamos reproduzir os resultados reportados no artigo original. Assumindo a credibilidade do modelo, o que se pretende é avaliar o seu desempenho sobre gravações instrumentais não vistas no treinamento.

3.2.1 Objetivos

- Avaliar o desempenho do CREPE: Testar o CREPE usando um banco de dados abrangente de flauta para avaliar sua precisão na estimativa de *pitch*. Vale mencionar que estes dados não foram vistos pelo modelo durante seu treinamento.

- Identificar limitações: Determinar os cenários específicos onde o CREPE falha ou apresenta baixo desempenho.

- Propor melhorias: Desenvolver estratégias para aprimorar a capacidade do CREPE na estimativa de *pitch*, servindo como base para implementar um modelo de estimativa de múltiplos *pitches*.

3.2.2 Base de dados de Flauta

Para testar o CREPE e avaliar seu desempenho, escolhemos o banco de dados de flauta *traditional flute dataset* criado por Brum [61]. Este conjunto de dados é composto por 30 fragmentos de áudio anotados manualmente. A construção foi feita a partir de peças reais de flauta solo, através de várias gravações de 4 peças musicais importantes do repertório de flauta. As peças musicais são:

- Allemande (primeiro movimento de BWV 1013): composta por J.S. Bach.

- Syrinx: composta por C. Debussy.

- Density 21.5: composta por E. Varese.

- Sequenza I: composta por L. Berio.

Do total de fragmentos, 10 correspondem a Allemande, 10 a Syrinx, 6 a Density 21.5 e 4 a Sequenza I. Todo o registro da flauta é coberto (de C4 a D7),

resultando em um total de 2245 eventos musicais. Adicionalmente, são fornecidos alguns arquivos que compõem o conjunto de dados:

- Arquivos de áudio: codificados como 16 *bits*, taxa de amostragem igual a 44100 Hz e extensão de arquivo *wav*.
- Anotações de frequência: anotações geradas manualmente usando o programa *Sonic Visualizer*.

3.2.3 Implementação e testes da rede neural CREPE

3.2.3.1 Preparação do Conjunto de Dados

Para utilizar a base de dados de flauta, foi necessário realizar um pré-processamento visando a compatibilizar os arquivos de áudio com a entrada do modelo CREPE. No treinamento do CREPE foram usados arquivos de áudio com taxa de amostragem de 16000 Hz. Por este motivo, é necessário reduzir a taxa de amostragem original da base de dados. O *downsampling* foi realizado com o pacote Python para análise de música e áudio chamado *Librosa* [62], amplamente conhecido e utilizado nos trabalhos da área de recuperação de informação musical.

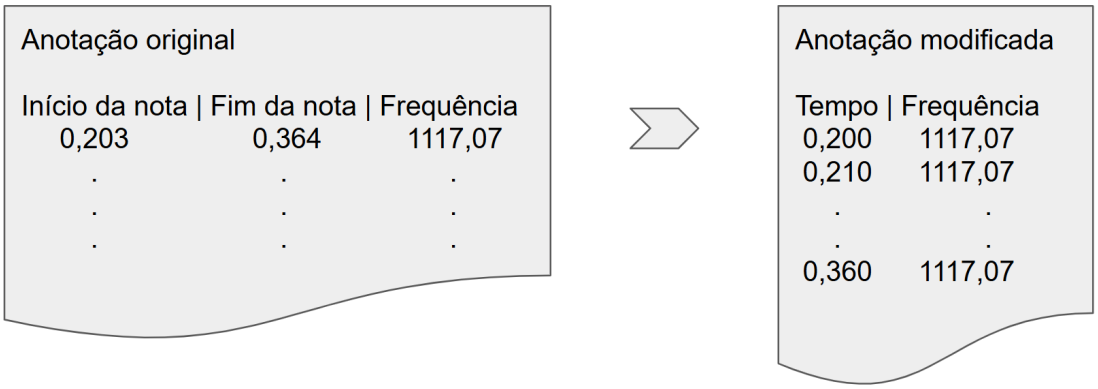

Figura 3.1: Ajuste na anotação da base de flauta.

Também foi necessário modificar as anotações originais da base de flauta. O CREPE realiza estimativas de *pitch* ao longo do tempo, fornecendo um valor de frequência a cada 10 milissegundos do audio analisado (resolução padrão que pode ser alterada, porém foi mantida nos testes). De forma diferente, as anotações da base de flauta apresentam intervalos de tempo para as frequências presentes no áudio,

destacando o tempo de início e fim das notas. Por isso, também utilizando o pacote Librosa, os intervalos de frequências foram convertidos em valores ponto a ponto ao longo do tempo conforme a Figura 3.1, com resolução de 10 milissegundos para sincronizar com o resultado do CREPE.

3.2.3.2 Testes e Análise de Desempenho

Para implementação do modelo CREPE foi utilizada a linguagem Python, juntamente com os pacotes Keras [63] e Tensorflow [64], que são *frameworks* para desenvolvimento de redes neurais. Adicionalmente, os desenvolvedores do CREPE fornecem boa parte do repositório de códigos, facilitando sua reprodução.

Para avaliação do modelo, foi usada a métrica precisão de altura crua (RPA, em inglês *Raw Pitch Accuracy*). O RPA é uma métrica de avaliação bastante empregada em trabalhos apresentados na *International Society for Music Information Retrieval* (ISMIR), usada para medir o desempenho de modelos de estimação de pitch. Esta métrica quantifica a precisão com a qual um modelo prevê o *pitch* de um sinal de áudio, incluindo um nível de tolerância para considerar diferenças entre os valores de *pitch* previstos e os reais. Para calcular o RPA dos resultados obtidos pelo CREPE, foi utilizada a biblioteca *mir_eval* introduzida em [59], que fornece a função para calcular o RPA.

O RPA é determinado pela expressão

$$\text{RPA} = \frac{\sum \left(\text{ref}_{\text{voicing}} \times \mathbf{1}_{\{|f_{\text{ref}} - f_{\text{est}}| < \text{tol}\}}\right) \text{para frequências} > 0}{\sum \text{ref}_{\text{voicing}}}, \tag{3.1}$$

onde $\text{ref}_{\text{voicing}}$ é o indicador de frequência de referência (1 se uma frequência de referência existir, 0 em caso contrário); f_{ref} é a frequência de referência em *cents*; f_{est} é a frequência estimada pelo modelo em *cents*; tol é a tolerância em *cents* dentro da qual uma frequência é considerada correta; $\mathbf{1}_{\{\cdot\}}$ é a função indicadora, igual a 1 se a diferença entre a frequência de referência e estimada for menor do que a tolerância, 0 em caso contrário; A soma é realizada onde f_{ref} e f_{est} são diferentes de 0.

Utilizamos os pacote Python *SciPy* para ler os arquivos de audio *wave* da base de dados, e o *Numpy* para gerar *frames* de 1024 amostras com sobreposição de 10 milissegundos. Além disso, considerando a normalização em lote realizada em cada camada, os *frames* são normalizados para se ajustar ao modelo.

Para cada um dos 30 fragmentos de audio da base de dados, as anotações definem valores de frequências relacionadas aos *frames*. Com isso são geradas amostras formadas por tuplas (*frame*, frequência), que alimentam a entrada da rede e fornecem a referência usada para calcular o RPA.

3.2.3.3 Resultados

As predições foram realizadas utilizando o conjunto de pesos pré-treinados do CREPE, disponibilizado pelos desenvolvedores do modelo. Os 30 arquivos de áudio foram submetidos à rede e foram calculadas as frequências ao longo do tempo. Para cada frequência estimada, o modelo fornece um valor de confiança, que determina a probabilidade de ocorrência desta frequência. Quanto maior o valor de confiança, maior a chance de a frequência estar presente no *frame* analisado.

Dessa forma, o CREPE retorna as probabilidades das frequências estimadas. A saída do modelo é um vetor de 360 componentes, onde cada valor indica a presença de uma determinada frequência. O valor final da frequência em hertz é calculado a partir da média ponderada já vista em (2.14) e (2.15), aqui novamente descritas por conveniência:

$$\hat{c} = \frac{\sum_{i=M-4}^{M+4} \hat{y}_i c_i}{\sum_{i=M-4}^{M+4} \hat{y}_i}, \quad M = \text{argmax}_i(\hat{y}_i) \tag{3.2}$$

$$\hat{f} = f_{\text{ref}} 2^{\hat{c}/1200}. \tag{3.3}$$

Na Tabela 3.1 estão os valores de RPA para as estimativas de frequência calculadas pelo CREPE, usando a base de dados *Traditional Flute Dataset*. Estes resultados apresentam dois cenários analisados para verificar o grau de confiança fornecido pelo CREPE. No primeiro caso, foi considerado qualquer valor de confiança (Conf ≥ 0) para calcular o RPA. No segundo cenário, o RPA considerou apenas as estimativas com alta confiança (Conf $\geq 0{,}9$).

Com esta abordagem, inicialmente buscamos avaliar o desempenho com baixo grau de confiança, o que sugere maior dificuldade para interpretar as frequências. Intuitivamente, se o modelo retorna valores baixos para as máximas probabilidades, espera-se que o RPA seja reduzido, dada a maior chance de erro no valor real da frequência (com uma tolerância de 50 *cents*). Por outro lado, considerando apenas estimativas com alto grau de confiança, queremos constatar que o modelo apresenta

ótimo desempenho, confirmado com valores altos de RPA.

O comportamento esperado em cada cenário ficou evidente nos resultados apresentados na Tabela 3.1. Para o caso de alta confiança, vemos o RPA próximo ou igual a 1. Também podemos observar baixos valores de RPA quando o nível de confiança é baixo. Porém, mesmo neste caso constatamos excelentes resultados de RPA, com média acima de 0,7, como demonstrado na Tabela 3.2.

	Resultado RPA	
Nome do fragmento	**Conf $\geq$ 0**	**Conf $\geq$ 0,9**
allemande_first_fragment_nicolet	0,79	0,98
allemande_first_fragment_larrieu	0,84	0,99
allemande_second_fragment_gerard	0,84	0,95
allemande_second_fragment_preston	0,81	0,96
allemande_third_fragment_rampal	0,82	0,97
allemande_third_fragment_nicolet	0,79	0,91
allemande_fourth_fragment_larrieu	0,82	0,99
allemande_fourth_fragment_gerard	0,87	0,99
allemande_fifth_fragment_preston	0,61	0,68
allemande_fifth_fragment_rampal	0,47	0,54
syrinx_first_fragment_douglas	0,96	1,00
syrinx_first_fragment_bernold	0,97	1,00
syrinx_second_fragment_dwyer	0,97	1,00
syrinx_second_fragment_bourdin	0,96	1,00
syrinx_third_fragment_rhodes	0,94	0,99
syrinx_third_fragment_douglas	0,95	0,99
syrinx_fourth_fragment_bernold	0,89	0,96
syrinx_fourth_fragment_dwyer	0,94	0,98
syrinx_fifth_fragment_bourdin	0,80	0,92
syrinx_fifth_fragment_rhodes	0,96	0,99
density_first_fragment_zoon	0,98	1,00
density_second_fragment_zoon	0,95	0,99
density_third_fragment_zoon	0,94	0,99
density_fourth_fragment_beauregard	0,90	0,99
density_fifth_fragment_beauregard	0,59	1,00
density_sixth_fragment_beauregard	0,92	0,99
sequenza_first_fragment_robison	0,78	0,98
sequenza_second_fragment_robison	0,72	0,94
sequenza_fourth_fragment_robison	0,88	0,96
sequenza_sixth_fragment_robison	0,57	0,85

Tabela 3.1: RPA de estimativas do CREPE para base de dados *Traditional Flute*.

Nas tabelas vemos que os fragmentos da peça Sequenza I tiveram pior desempenho no cenário de avaliação com baixo grau de confiança. Este resultado é esperado pela complexidade da composição. A peça é considerada de alta dificul-

dade devido ao seu uso extensivo de técnicas avançadas como multifônicos, onde o flautista produz vários sons simultaneamente, e articulações que exigem mudanças rápidas e precisas entre staccato (notas curtas e separadas) e legato (notas conectadas sem interrupção). Todos estes aspectos específicos da composição e uso do instrumento resultam em maior dificuldade para o modelo, que foi treinado em bases de áudio mais genéricas. Porém, é importante ressaltar que o baixo desempenho é relativo, sendo na prática o RPA médio acima de 0,7 um bom resultado, sugerindo a possibilidade de melhoria no desempenho se o modelo for treinado para casos particulares.

Observa-se que, no fragmento *allemande_fifth_fragment_rampal*, os valores de RPA apresentaram desempenho consideravelmente inferior (0,47 e 0,54) em relação aos demais trechos analisados. Esse resultado se deve ao fato de que a anotação das frequências de referência foi realizada de forma manual, o que introduz imprecisões. Como consequência, o modelo CREPE obteve valores reduzidos de RPA, não necessariamente refletindo uma limitação do modelo em si, mas sim a inconsistência das anotações de referência utilizadas na avaliação.

	Resultado RPA médio	
Fragmento	**Conf $\geq$ 0**	**Conf $\geq$ 0,9**
Allemande	0,82 $\pm$ 0,03	0,97 $\pm$ 0,03
Synrix	0,93 $\pm$ 0,05	0,98 $\pm$ 0,03
Density	0,88 $\pm$ 0,14	0,99 $\pm$ 0,01
Sequenza	0,74 $\pm$ 0,13	0,93 $\pm$ 0,06

Tabela 3.2: Média de RPA para resultados do CREPE.

3.2.4 Implementação e testes da rede neural FCN

3.2.4.1 Preparação do Conjunto de Dados

De forma semelhante ao caso de teste com a rede CREPE, realizamos um pré-processamento para compatibilizar os arquivos de áudio com a entrada do modelo FCN. O treinamento do FCN utilizou arquivos de áudio com taxa de amostragem de 8000Hz. Por isso, reduzimos a taxa de amostragem original da base de dados, usando o Librosa.

As anotações originais da base de flauta também foram modificadas, tendo em vista que o FCN retorna estimativas de frequências com resolução de 1 milissegundo.

E como mencionado, as anotações originais são definidas em intervalos de tempo para as frequências. Foi usado o pacote Librosa para converter as anotações em valores ponto a ponto ao longo do tempo, de forma semelhante ao processo descrito na Figura 4.1.

3.2.4.2 Testes e Análise de Desempenho

O FCN é um modelo baseado no CREPE, e por isso sua implementação foi muito semelhante, utilizando linguagem Python com os pacotes Keras e Tensorflow. Além disso, o desenvolvedor do FCN disponibiliza boa parte do repositório de códigos, auxiliando a reprodução do modelo.

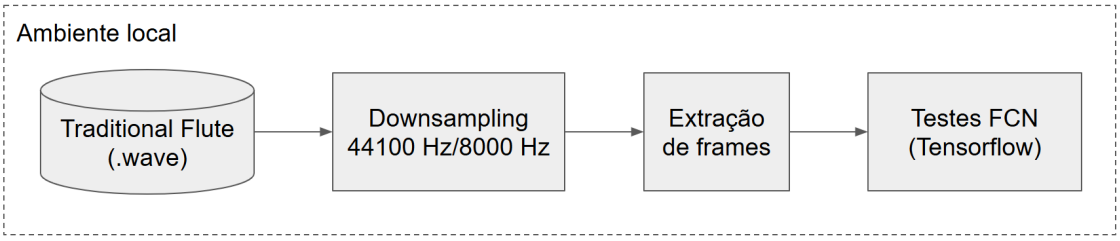

Figura 3.2: Diagrama de testes do FCN com a base de dados Traditional Flute.

Para avaliação do modelo FCN, também foi adotada a métrica *Raw Pitch Accuracy* (RPA), utilizando a biblioteca *mir_eval*. Conforme mencionado, com o RPA podemos padronizar as avaliações e comparar desempenho entre modelos de estimadores de frequência. A Figura 3.2 apresenta o diagrama do teste do modelo FCN com a base de dados *Traditional Flute*.

3.2.4.3 Resultados

De forma semelhante à etapa de testes do CREPE, usamos os pesos pré-treinados fornecidos pelo desenvolvedor do modelo. O FCN fornece as estimativas de frequências a partir do vetor de probabilidades, seguindo a mesma estratégia definida pelo CREPE. Porém, no caso do FCN este vetor de saída possui 486 componentes visando a aumentar a resolução do modelo para obter resultados mais precisos.

Os arquivos de audio da base de dados foram processados pela rede sem a necessidade de extrair *frames*. Isto é possível porque uma das alterações propostas pelo FCN é a utilização do arquivo no tamanho original na entrada da rede, apro-

veitando a característica de redes convolutivas, que permitem tamanho de entrada variável. Assim, é possível realizar as convoluções sobre o arquivo uma unica vez, diferentemente do CREPE, que executa as estimativas em *frames*.

Os resultados apresentados nas Tabelas 3.3 e 3.4 mostram um desempenho pior do que o obtido pelo CREPE. Este comportamento é esperado, uma vez que o modelo do FCN, apesar de ser uma versão do CREPE, foi proposto para trabalhar com sinais de fala. Todas as bases de dados utilizadas no treinamento do FCN são de sinais de fala, e desta forma é razoável o modelo apresentar valores de RPA abaixo dos que vemos para o CREPE. Isto fica evidente para o pior cenário (Conf ≥ 0), onde todas as estimativas com probabilidades acima de zero são avaliadas, que em determinados casos resultou em um valor abaixo de 0,5.

Por outro lado, no caso de maior confiança (Conf $\geq$ 0,9) os resultados são promissores. Comparando as Tabelas 3.2 e 3.4, vemos que os valores de RPA obtidos pela rede FCN para cada uma das composições são próximos aos números do CREPE. Este resultado é significativo, novamente levando em consideração que o FCN não utilizou áudio de música em seu treinamento, diferentemente do CREPE, que utilizou exclusivamente bases de áudio de música com diversos instrumentos. Estes resultados sugerem que o desempenho do FCN pode ser aperfeiçoado através de treinamento usando bases de áudio de música. Em tese, utilizando as bases vistas pelo CREPE, é coerente esperar melhores resultados para o RPA da rede FCN.

Levando isso em conta, visando a melhorar os resultados do RPA, realizamos um novo treinamento do FCN com uma das bases utilizadas no treinamento do CREPE. Usamos como referência a metodologia de treinamento apresentada nos artigos do FCN e CREPE. Considerando tamanho, documentação e qualidade da anotação da base de dados, selecionamos a *MDB-stem-synth*, que apresenta informações detalhadas sobre os arquivos de audio e anotações perfeitas, uma vez que utiliza a versão sintética da base de dados *MedleyDB*.

Nas Tabelas 3.5 e 3.6 apresentamos os resultados obtidos após treinamento parcial, devido à limitação de recursos computacionais que inicialmente restringiram o processo (computador local com GPU de baixa capacidade), diversas vezes ocasionando erros na execução devido problemas de memória insuficiente. Realizamos testes na rede FCN com os novos pesos para visualizar o progresso do treinamento,

	Resultado RPA	
Nome do fragmento	**Conf $\geq$ 0**	**Conf $\geq$ 0,9**
allemande_first_fragment_nicolet	0,57	0,93
allemande_first_fragment_larrieu	0,57	0,95
allemande_second_fragment_gerard	0,64	0,94
allemande_second_fragment_preston	0,66	0,90
allemande_third_fragment_rampal	0,61	0,85
allemande_third_fragment_nicolet	0,67	0,89
allemande_fourth_fragment_larrieu	0,60	0,88
allemande_fourth_fragment_gerard	0,72	0,94
allemande_fifth_fragment_preston	0,43	0,69
allemande_fifth_fragment_rampal	0,18	0,41
syrinx_first_fragment_douglas	0,50	1,00
syrinx_first_fragment_bernold	0,45	0,99
syrinx_second_fragment_dwyer	0,52	1,00
syrinx_second_fragment_bourdin	0,42	1,00
syrinx_third_fragment_rhodes	0,88	0,98
syrinx_third_fragment_douglas	0,88	0,98
syrinx_fourth_fragment_bernold	0,80	0,96
syrinx_fourth_fragment_dwyer	0,78	0,97
syrinx_fifth_fragment_bourdin	0,36	0,50
syrinx_fifth_fragment_rhodes	0,74	0,99
density_first_fragment_zoon	0,79	0,96
density_second_fragment_zoon	0,70	0,98
density_third_fragment_zoon	0,71	1,00
density_fourth_fragment_beauregard	0,43	0,95
density_fifth_fragment_beauregard	0,39	1,00
density_sixth_fragment_beauregard	0,68	0,96
sequenza_first_fragment_robison	0,48	0,95
sequenza_second_fragment_robison	0,42	0,87
sequenza_fourth_fragment_robison	0,55	0,88
sequenza_sixth_fragment_robison	0,27	0,63

Tabela 3.3: RPA de estimativas do FCN para base de dados *Traditional Flute*.

	Resultado RPA médio	
Fragmento	**Conf $\geq$ 0**	**Conf $\geq$ 0,9**
Allemande	0,63 $\pm$ 0,05	0,91 $\pm$ 0,04
Synrix	0,63 $\pm$ 0,20	0,94 $\pm$ 0,15
Density	0,62 $\pm$ 0,16	0,98 $\pm$ 0,02
Sequenza	0,43 $\pm$ 0,12	0,83 $\pm$ 0,14

Tabela 3.4: Média de RPA para resultados do FCN.

novamente utilizando a base de dados de flauta. Notamos, como esperado para o treinamento usando dados vistos pelo CREPE, que os valores de RPA foram melhorados. Vale ressaltar que no Capítulo 4 estes problemas foram resolvidos após a

utilização de serviço de computação em nuvem com maior capacidade de processamento.

Por fim, vale mencionar que na tentativa de incrementar os resultados, prosseguimos com o treinamento usando mais dados. Porém, observamos que o FCN não apresentou melhoria nos valores de RPA. Em alguns casos, vimos a rede apresentar instabilidade no treinamento, que resultava em estimativas erradas para todos os arquivos da base de dados de flauta. Supomos que um dos motivos pode ser a estratégia adotada para o particionamento dos *frames* que compõem as amostras de treino, validação e teste. No capítulo 4 é descrito um novo procedimento de treinamento do FCN que melhorou os resultados.

	Resultado RPA - Novo treino	
Nome do fragmento	**Conf $\geq$ 0**	**Conf $\geq$ 0,9**
allemande_first_fragment_nicolet	0,74	0,93
allemande_first_fragment_larrieu	0,73	0,97
allemande_second_fragment_gerard	0,75	0,93
allemande_second_fragment_preston	0,77	0,94
allemande_third_fragment_rampal	0,75	0,95
allemande_third_fragment_nicolet	0,72	0,93
allemande_fourth_fragment_larrieu	0,72	0,95
allemande_fourth_fragment_gerard	0,80	0,95
allemande_fifth_fragment_preston	0,45	0,78
allemande_fifth_fragment_rampal	0,18	0,41
syrinx_first_fragment_douglas	0,94	0,99
syrinx_first_fragment_bernold	0,90	1,00
syrinx_second_fragment_dwyer	0,78	0,99
syrinx_second_fragment_bourdin	0,68	0,99
syrinx_third_fragment_rhodes	0,84	0,94
syrinx_third_fragment_douglas	0,87	0,96
syrinx_fourth_fragment_bernold	0,88	0,98
syrinx_fourth_fragment_dwyer	0,89	0,95
syrinx_fifth_fragment_bourdin	0,73	0,90
syrinx_fifth_fragment_rhodes	0,88	0,99
density_first_fragment_zoon	0,88	0,99
density_second_fragment_zoon	0,70	0,98
density_third_fragment_zoon	0,84	0,99
density_fourth_fragment_beauregard	0,57	0,99
density_fifth_fragment_beauregard	0,43	0,99
density_sixth_fragment_beauregard	0,74	0,96
sequenza_first_fragment_robison	0,63	0,94
sequenza_second_fragment_robison	0,56	0,88
sequenza_fourth_fragment_robison	0,70	0,94
sequenza_sixth_fragment_robison	0,44	0,91

Tabela 3.5: RPA de estimativas do FCN para base de dados *Traditional Flute* - Novo treino.

	Resultado RPA médio - Novo treino	
Fragmento	**Conf $\geq$ 0**	**Conf $\geq$ 0,9**
Allemande	0,75 $\pm$ 0,03	0,94 $\pm$ 0,01
Synrix	0,84 $\pm$ 0,08	0,97 $\pm$ 0,03
Density	0,69 $\pm$ 0,17	0,98 $\pm$ 0,01
Sequenza	0,58 $\pm$ 0,11	0,92 $\pm$ 0,03

Tabela 3.6: Média de RPA para resultados do FCN - Novo treino.

3.3 Proposta de modelo polifônico

Com base nos resultados obtidos nos estudos do CREPE e FCN, neste trabalho desenvolveremos um modelo de rede neural para lidar com o caso de polifonia, onde cada *frame* pode apresentar dois ou mais *pitches*.

3.3.1 Mixagem da base de dados MDB-stem-synth

A base de dados *MDB-stem-synth*, originalmente utilizada no treinamento do CREPE, foi mixada para gerar uma nova base contendo dois ou mais *pitches* por *frame*. Durante o processo de combinação, foram preservadas relações harmônicas entre os *stems* selecionados, garantindo que as frequências sobrepostas mantivessem coerência musical. Essa abordagem permitiu treinar o modelo CREME para identificar múltiplos *pitches* simultâneos de forma realista, aproximando o treinamento de situações polifônicas encontradas em contextos musicais naturais. Os detalhes são apresentados no Capítulo 4.

3.3.2 Implementação do modelo para estimativa de múltiplos *pitches*

A implementação do modelo para detecção de múltiplos *pitches* foi baseada na arquitetura do CREPE, mantendo a camada de entrada original e as seis camadas convolucionais que caracterizam o modelo de referência. Para aumentar a capacidade de aprendizagem em contextos polifônicos, foi adicionada uma camada densa antes da camada de saída, permitindo que a rede capturasse padrões hierárquicos mais complexos que favorecesse a identificação de múltiplos *pitches* simultâneos.

Além disso, a camada de saída foi expandida, aumentando a resolução do modelo para lidar com múltiplos *pitches*, em contraste com o CREPE, que é projetado para estimar apenas um *pitch* por *frame*. Essa modificação torna o CREME capaz de representar e predizer distribuições de frequências mais densas, preservando a eficiência das camadas convolucionais na extração de características relevantes do sinal de áudio.

Com essas alterações, pretende-se generalizar a detecção de frequências, mantendo o desempenho em cenários monofônicos enquanto se amplia a capacidade de estimativa em contextos polifônicos, sem alterar fundamentalmente a estrutura que

garante a robustez do CREPE. Os detalhes estão apresentados no Capítulo 4.

Capítulo 4

Experimentos e resultados

4.1 Objetivos

Esta Seção tem como objetivo apresentar os propósitos que nortearam a realização dos experimentos nesta pesquisa. Os experimentos foram concebidos tanto com fins exploratórios quanto avaliativos, buscando validar a proposta desenvolvida, denominada CREME (do inglês, *A Convolutional Representation for Multi-pitch Estimation*), um modelo de rede neural convolucional para estimar múltiplas frequências simultâneas em sinais de áudio polifônicos. O modelo foi desenvolvido a partir de uma adaptação do CREPE, modelo reconhecido como estado da arte para estimação de *pitch* em contextos monofônicos.

Os experimentos realizados podem ser agrupados em três objetivos principais:

Compreender o comportamento do modelo CREPE

Antes de iniciar a implementação do modelo CREME, considerou-se fundamental replicar o treinamento do modelo CREPE, utilizando seu código-fonte original disponibilizado publicamente [65]. O propósito desta etapa inicial não foi validar ou confrontar os resultados apresentados no artigo original, mas sim compreender em profundidade o funcionamento do modelo durante o processo de aprendizado.

A experiência obtida com o treinamento do CREPE serviu como uma etapa preparatória estratégica, proporcionando *insights* sobre o comportamento da rede em tarefas de estimação de *pitch*, bem como sobre os aspectos críticos a serem observados durante o treinamento, como as curvas de aprendizado, o comportamento

das métricas de avaliação, e os cuidados com o pré-processamento dos dados.

Além disso, essa etapa possibilitou a familiarização com a estrutura e as funcionalidades do código-fonte do CREPE, o que se mostrou vantajoso para o desenvolvimento do CREME. Ao reaproveitar componentes já implementados, como funções auxiliares para carregamento de dados, estruturas de treinamento, e rotinas de pré-processamento, foi possível acelerar o desenvolvimento e reduzir erros na construção do novo modelo.

Implementar e validar o modelo para estimação de múltiplos *pitches*

A partir da compreensão adquirida com o CREPE, foi desenvolvido o modelo CREME, com o objetivo de lidar com a tarefa mais complexa de estimação de múltiplos *pitches* em sinais de áudio polifônicos. A proposta do CREME é generalizar a abordagem do CREPE, que originalmente é limitada a sinais monofônicos, para cenários onde duas ou mais frequências fundamentais possam estar presentes simultaneamente.

Inspirado pela nomenclatura do modelo original, o nome CREME foi adotado como uma variação de CREPE, refletindo a continuidade conceitual entre os dois modelos, ao mesmo tempo em que destaca a contribuição desta pesquisa no avanço da tarefa para o domínio polifônico.

Inicialmente, o foco da implementação do CREME foi a estimação de até duas frequências simultâneas, como forma de reduzir a complexidade do problema e permitir uma validação mais controlada da abordagem. No entanto, desde o início, o modelo foi concebido para ser extensível, de modo que a arquitetura e os procedimentos de treinamento pudessem ser facilmente adaptados para lidar com mais de duas frequências em trabalhos futuros. O código final desenvolvido permite a adaptação para cenários além de duas frequências.

Avaliar o desempenho do CREME em diferentes cenários

Uma vez implementado e treinado, o modelo CREME foi submetido a uma série de testes para avaliar sua eficácia em diferentes contextos. Os testes abrangeram tanto dados sintéticos quanto dados reais, e foram conduzidos em três cenários principais:

- Dados sintéticos com múltiplos *pitches*: Utilizou-se a base de dados *MDB-stem-synth*, que é uma versão sintética da base *MedleyDB*, com anotações perfeitas das frequências presentes a cada *frame* temporal. Esta base foi empregada tanto no treinamento quanto na validação do modelo, possibilitando avaliar o desempenho do CREME em um ambiente controlado e com alto grau de fidelidade nas anotações. Também foi utilizado o conjunto *Bach10-mf0-synth*, composto por sinais sintéticos com múltiplas frequências anotadas de fagote, clarinete, saxofone e violino.

- Dados reais com uma única frequência: Para testar a capacidade do modelo CREME de manter um bom desempenho em cenários monofônicos, foi utilizada a base *Traditional Flute Dataset* [61], composta por gravações reais de flauta com uma única frequência fundamental. Esse experimento teve como objetivo verificar se a generalização do modelo para múltiplos *pitches* comprometia seu desempenho em tarefas originalmente bem resolvidas pelo CREPE.

- Cenários polifônicos com frequências próximas: Por fim, foi utilizado o conjunto *RWC Music Database*, que contém sinais reais de diversos instrumentos (como violão e clarinete), para avaliar a resolução do modelo frente a *pitches* muito próximos no espectro de frequências. Este teste foi particularmente importante para identificar os limites do CREME na tarefa de separação de componentes harmônicos em situações complexas.

4.2 Ambiente de desenvolvimento

Os experimentos desenvolvidos nesta pesquisa foram conduzidos em dois ambientes computacionais distintos: uma máquina local, utilizada durante as fases de desenvolvimento e testes iniciais dos códigos, e a plataforma *Google Colab*, empregada para a execução dos treinamentos mais intensivos, que demandaram maior poder computacional. Esta abordagem híbrida possibilitou maior flexibilidade no processo de desenvolvimento, ao mesmo tempo em que garantiu os recursos necessários para o treinamento efetivo dos modelos.

Ambiente Local

A máquina local foi utilizada principalmente para o desenvolvimento dos códigos relacionados à construção da base de dados polifônica, à reimplementação e ao retreinamento do modelo CREPE, à modelagem da arquitetura CREME e à realização de testes preliminares para validação do funcionamento correto das rotinas de treinamento. A configuração do sistema utilizado está descrita a seguir:

- Processador: Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz, 2304 MHz, 4 núcleos físicos, 8 *threads*
- Memória RAM: 16 GB
- Placa de vídeo: NVIDIA GeForce MX250 com 2 GB de memória dedicada

Apesar das limitações desse *hardware* para tarefas intensivas de aprendizado profundo, ele foi suficiente para testes rápidos de funcionalidade e verificação de integridade dos dados e modelos. A estratégia adotada consistiu em validar localmente os componentes desenvolvidos e, após sua verificação, transferi-los para o ambiente remoto de execução.

Ambiente em Nuvem

Para o treinamento dos modelos em larga escala, foi utilizado o ambiente do *Google Colab*, que fornece acesso gratuito a recursos computacionais de alto desempenho, incluindo GPUs modernas como NVIDIA A100 e L4. O ambiente no *Google Colab* foi configurado com *Python 3* como interpretador e com suporte ao *framework TensorFlow 2.10*, garantindo compatibilidade com os requisitos do modelo CREPE original e com os ajustes propostos no CREME.

Os códigos desenvolvidos foram armazenados em um repositório remoto no *GitHub*, permitindo versionamento, reprodutibilidade e fácil integração com o ambiente no *Google Colab*. Após a validação local, os *scripts* eram transferidos para o *Google Colab* por meio da clonagem do repositório. Além disso, utilizou-se o serviço *Google Drive* para o armazenamento das bases de dados, *checkpoints* de modelos, pesos salvos e demais artefatos gerados durante os experimentos.

Ferramentas e bibliotecas Utilizadas

O desenvolvimento da pesquisa foi conduzido utilizando a linguagem de programação *Python*, na versão 3.8, em conjunto com bibliotecas especializadas para manipulação de sinais de áudio, aprendizado de máquina e avaliação de modelos. A seguir, são listadas as principais bibliotecas utilizadas:

- *TensorFlow 2.10*: biblioteca principal para a construção, treinamento e inferência dos modelos de redes neurais convolucionais.
- *NumPy e Pandas*: manipulação de *arrays* e estruturas de dados tabulares.
- *SciPy*: funções científicas para análise e manipulação de sinais.
- *Librosa*: pré-processamento e manipulação de arquivos de áudio.
- *Pydub*: biblioteca para mixagem de áudio.
- *MIR Eval*: biblioteca de métricas de avaliação para tarefas de *Music Information Retrieval*, incluindo a métrica RPA utilizada nos experimentos.
- *Pyrubberband*: biblioteca para variação de *pitch* e tempo em sinais de áudio, baseada no *Rubber Band Library*.
- *Flazy*: biblioteca de gerenciamento de conjuntos de dados.
- *Wandb (Weights and Biases)*: biblioteca da plataforma utilizada para monitoramento e visualização em tempo real do treinamento, possibilitando o rastreamento de métricas, visualização de curvas de aprendizado, salvar artefatos e reprodutibilidade dos experimentos.
- *Audacity*: *software* para manipulação de arquivos de áudio usado nos experimentos de avaliação do modelo CREME.

Essa infraestrutura permitiu o desenvolvimento eficiente dos experimentos, com suporte adequado tanto à fase de prototipagem quanto à fase de treinamento em escala, assegurando a rastreabilidade e a integridade dos resultados obtidos.

4.3 Metodologia experimental

4.3.1 Compreensão do Modelo CREPE

Considerando que o modelo proposto nesta pesquisa, o CREME (*A Convolutional Representation for Multipitch Estimation*), foi concebido como uma generalização do modelo CREPE (*A Convolutional Representation for Pitch Estimation*), julgou-se fundamental realizar uma etapa inicial dedicada à reimplementação e ao retreinamento do CREPE. O principal objetivo desta etapa foi aprofundar o entendimento sobre o funcionamento do CREPE durante seu treinamento, observando seu comportamento, eventuais dificuldades, e os aspectos críticos a serem considerados na configuração do modelo e no pré-processamento dos dados.

Além de fornecer uma base teórica e prática sólida para o desenvolvimento do CREME, esta etapa teve como papel estratégico antecipar e mitigar potenciais desafios, a partir da análise de um modelo considerado mais simples em termos de tarefa, que é a estimação de apenas uma frequência fundamental por instante de tempo. Assim, a experiência obtida com o CREPE foi utilizada como alicerce para decisões de projeto e implementação no desenvolvimento do modelo para detecção de múltiplos *pitches*.

Para esta tarefa, foi utilizado o código-fonte original do CREPE, disponível publicamente em repositório online [65]. O código foi originalmente escrito em *Python*, utilizando o *framework TensorFlow* na versão 2.8. No entanto, por se tratar de uma base implementada em 2018, várias modificações e refatorações foram necessárias para garantir a compatibilidade com versões mais recentes de bibliotecas como *NumPy* e o próprio *TensorFlow*. Muitas funções e métodos anteriormente utilizados haviam sido descontinuados ou modificados, o que exigiu um trabalho de adaptação criterioso.

Além de ajustes por questões de compatibilidade, também foram realizadas melhorias pontuais com o objetivo de otimizar o desempenho do código e facilitar sua manutenção. Dentre essas melhorias, destaca-se a adoção da biblioteca *Librosa* para lidar com tarefas de leitura e pré-processamento de arquivos de áudio (por exemplo, extração de *frames* de áudio), substituindo trechos de código menos eficientes e proporcionando maior legibilidade.

Outro componente importante presente no código original do CREPE é a biblioteca *Flazy* [66], uma ferramenta própria desenvolvida pelos autores do modelo para gerenciar conjuntos de dados de forma eficiente. O *Flazy* oferece uma API funcional para leitura e manipulação de grandes volumes de dados, incluindo recursos como amostragem aleatória, *streaming* e aumento de dados em tempo real. Esta biblioteca se mostrou extremamente conveniente para a manipulação de bases de dados que não cabem inteiramente na memória, sendo responsável por alimentar o *pipeline* de treinamento com dados processados de forma incremental. Dada sua importância e robustez, a biblioteca *Flazy* também foi incorporada à implementação do modelo CREME.

Para possibilitar a utilização da biblioteca *Flazy* no gerenciamento da base de dados, foi necessário inicialmente realizar a conversão dos arquivos originais da base *MDB-stem-synth*, que se encontram no formato *WAVE*, para o formato *TFRecord*, nativamente suportado pelo *framework TensorFlow*. O *TFRecord* é um formato binário compactado e otimizado para leitura sequencial, o que contribui significativamente para reduzir o tempo de carregamento dos dados durante o treinamento e, consequentemente, para o desempenho do *pipeline* de aprendizado.

Para esse processo de conversão de formatos, descrito na Figura 4.1, foi desenvolvido um código *Python* responsável por percorrer os 230 arquivos de áudio da base *MDB-stem-synth* e realizar a extração dos *frames* de áudio, bem como das anotações associadas a cada um desses quadros, ou seja, as frequências fundamentais presentes em cada instante de tempo. Os dados extraídos foram então serializados e armazenados no formato *TFRecord*, resultando em um conjunto de dados final compatível com a biblioteca *Flazy* e pronto para ser utilizado nos experimentos subsequentes.

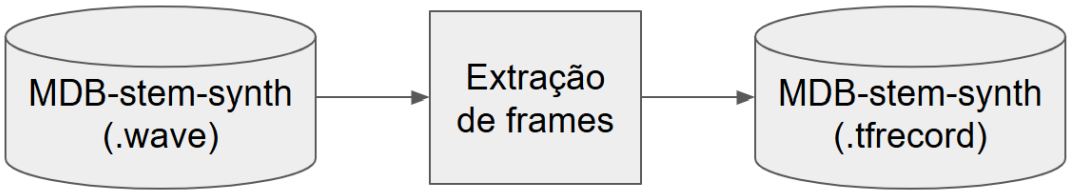

Figura 4.1: Conversão da base de dados MDB-stem-synth.

Concluído o processo de adaptação da base e ajustes no código do CREPE, foi conduzido um primeiro treinamento exploratório, com o intuito de analisar o comportamento da rede em um cenário simplificado, descrito na Figura 4.2. Neste experimento inicial, optou-se por utilizar os dados puros extraídos dos arquivos de áudio, sem aplicar técnicas de aumento de dados (como adição de ruído e variações de *pitch*) e sem qualquer forma de randomização dos *frames*. Os *frames* utilizados neste treinamento foram extraídos de forma sequencial de um mesmo arquivo, o que implica em uma forte correlação temporal entre os dados apresentados à rede, e também em menor variação de padrões nos dados apresentados a rede.

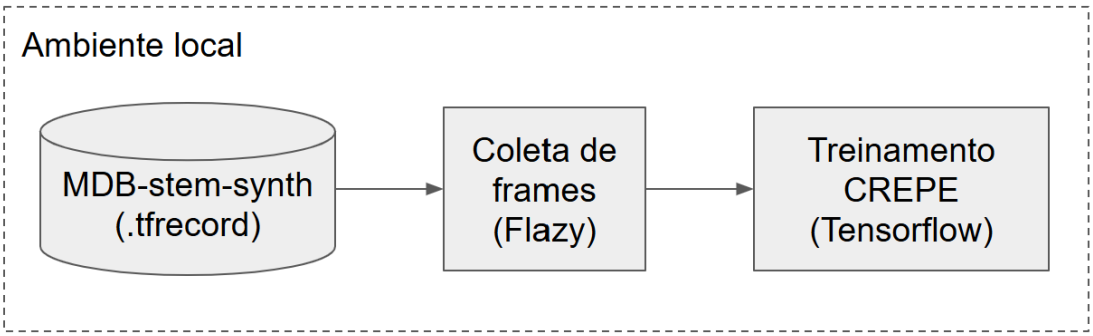

Figura 4.2: Diagrama para treinamento exploratório do CREPE.

Essa configuração intencionalmente simplificada teve como objetivo proporcionar uma visão mais clara do comportamento da rede diante de dados não manipulados, oferecendo uma base para calibração inicial de parâmetros e formulação de hipóteses quanto ao desempenho e à necessidade de técnicas adicionais de generalização.

Os resultados obtidos nesta etapa foram bastante elucidativos. Observou-se que o modelo CREPE apresentou um crescimento rápido na métrica RPA já nas primeiras épocas de treinamento. Como ilustrado na Figura 4.3, a partir de 50 épocas o modelo atingiu um valor de RPA em torno de 0,8. Paralelamente, vemos que as curvas de perda de treinamento e validação iniciaram-se próximas, mas começaram a se distanciar significativamente à medida que o treinamento prosseguia.

Esse afastamento entre as curvas e a consequente queda no desempenho de validação são indicativos claros de *overfitting*: a rede rapidamente se ajusta aos padrões específicos dos dados de treinamento, mas perde a capacidade de generalização quando confrontada com dados não vistos anteriormente. Tal comportamento evi-

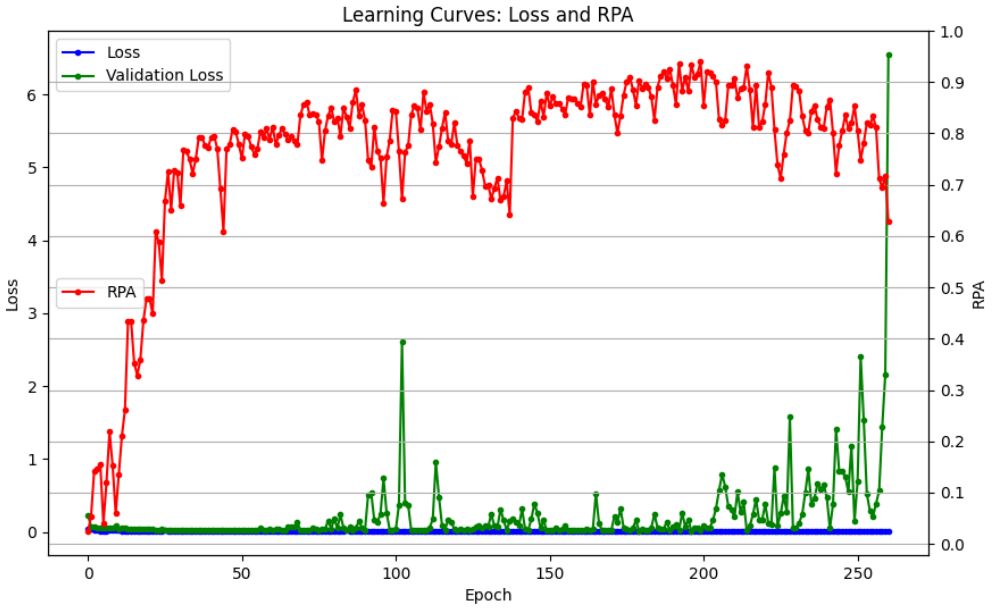

Figura 4.3: Primeiro treinamento exploratório do CREPE.

denciou uma característica crítica do CREPE, que é a sensibilidade à variação dos dados, o que reforça a importância de adotar estratégias robustas de regularização no treinamento, principalmente quando se deseja expandir o modelo para lidar com tarefas mais complexas como estimação de múltiplas frequências simultâneas.

Assim, ficou evidente a necessidade de diversificação contínua dos dados apresentados à rede durante o treinamento, por meio de técnicas como aumento de dados, incluindo adição de ruído e deslocamentos de *pitch*, e a randomização dos *frames*, garantindo que a rede tivesse acesso a *frames* oriundos de diferentes arquivos de áudio ao longo do processo de treinamento. Essas conclusões preliminares foram fundamentais para orientar as decisões no desenvolvimento do modelo CREME, que, por lidar com cenários ainda mais complexos, demandaria cuidados redobrados com relação à variação e à qualidade dos dados utilizados durante o treinamento. A Figura 4.4 apresenta o diagrama deste processo de treinamento modificado.

Diante da detecção de *overfitting*, aplicou-se aumento de dados para ampliar a diversidade dos padrões apresentados à rede. Com a biblioteca *Pyrubberband* foi realizado deslocamento de *pitch*. Também inseriu-se ruído de forma aleatória nos *frames*, variando também a intensidade desse ruído. Adicionalmente, implementou-se a randomização de *frames*, de modo que a rede não recebesse sequências provenientes

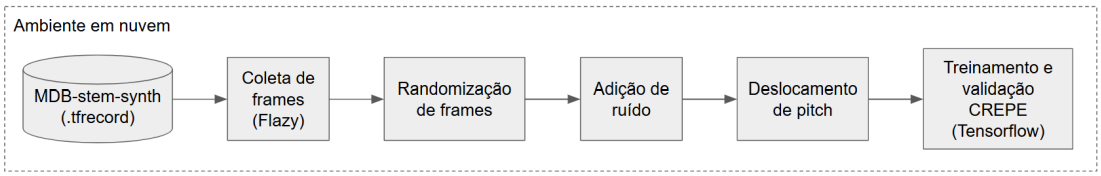

Figura 4.4: Diagrama para treinamento do CREPE com aumento e randomização de dados.

de um mesmo arquivo; em vez disso, cada novo *frame* era extraído de um arquivo distinto e apresentado de forma não sequencial durante o treinamento.

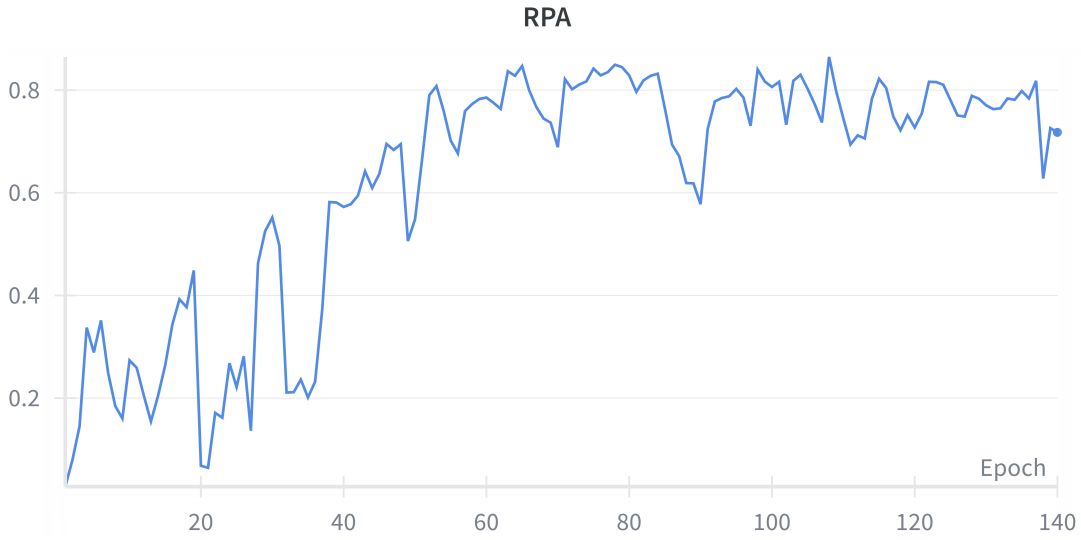

Figura 4.5: RPA do retreinamento do CREPE.

Os resultados após esses ajustes podem ser observados nas figuras Figuras 4.5 e 4.6, evidenciando um aumento progressivo dos valores de RPA e a convergência das curvas de perda de aprendizado e validação ao longo das épocas. Esses indicadores confirmam que as alterações nos parâmetros da rede e no tratamento dos dados foram eficazes para melhorar o desempenho do modelo.

Outro experimento adicional conduzido nesta etapa foi o retreinamento da rede neural FCN. Conforme descrito na Seção 3.2, a investigação sobre o FCN fez parte das etapas exploratórias do trabalho, motivada pelo fato de já existir literatura abordando o CREPE e por entender que o estudo de uma arquitetura semelhante poderia revelar potencial de reaproveitamento de conceitos e componentes. Naquele momento, o treinamento do FCN não obteve resultados satisfatórios, apresentando

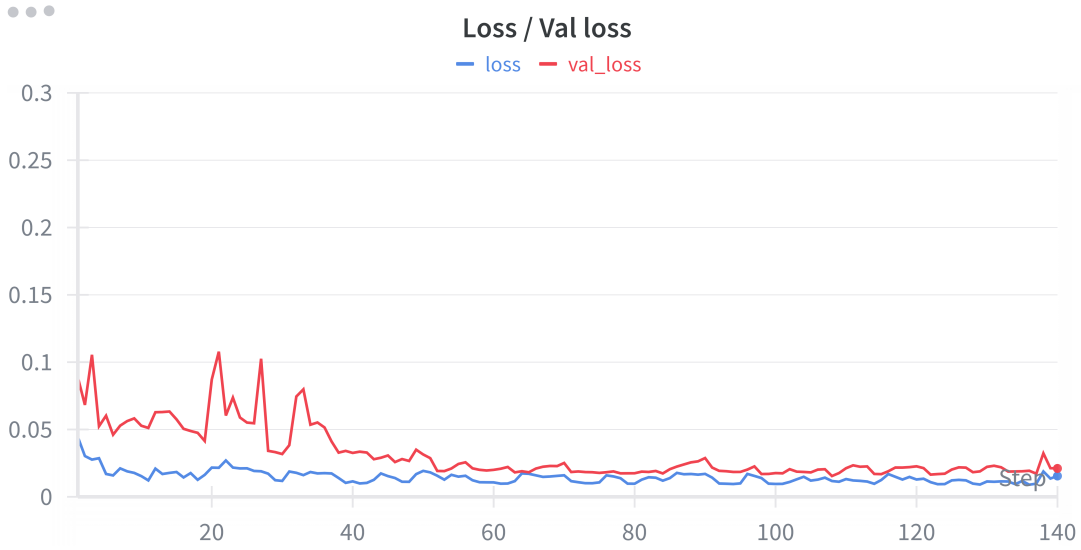

Figura 4.6: Curvas de perda de treinamento e validação do treinamento do CREPE.

desempenho aquém do esperado na tarefa de generalizar para sinais musicais, especificamente nos testes feito com a base de dados de flauta.

Com a experiência acumulada no retreinamento bem-sucedido do CREPE, tanto no que se refere à adaptação do código quanto ao pré-processamento e manipulação da base de dados, foi possível aplicar o mesmo *pipeline* para o FCN. Nesse novo experimento, partiu-se dos pesos originais do modelo (treinados em sinais de fala), adotando a estratégia de *transfer learning*: o treinamento foi iniciado a partir desses pesos e prosseguiu com a base *MDB-stem-synth*, composta por sinais musicais, visando avaliar a capacidade do FCN de produzir estimativas de *pitch* mais adequadas para este novo domínio.

Inicialmente, o método experimental foi equivalente ao utilizado no CREPE: divisão da base em conjuntos de treinamento e validação; aplicação de técnicas de aumento de dados (adição de ruído e deslocamento de *pitch*); e randomização dos frames. Os resultados desta primeira tentativa mostraram evolução gradativa do RPA ao longo das épocas, porém com elevada instabilidade, como pode ser visto nas Figuras 4.7 e 4.8. A partir de aproximadamente 200 épocas, observou-se uma tendência de queda no RPA e indícios claros de *overfitting*, evidenciados pela divergência entre as curvas de perda de treinamento (em queda) e de validação (em crescimento). Essa análise indicou que, embora as técnicas de randomização estivessem presentes, a rede ainda não estava sendo exposta a variação suficiente de

Figura 4.7: RPA do retreinamento do FCN.

padrões na base de dados.

Para contornar essa limitação, adotou-se uma estratégia de *round robin* (multiplexação) implementada na biblioteca *Flazy*: a base de treinamento foi dividida em cinco grupos, sendo quatro grupos contendo 56 arquivos de áudio para treinamento e um grupo com 6 arquivos para validação. A randomização de *frames* passou a ser realizada dentro de cada grupo, mas com alternância sistemática entre os grupos a cada ciclo de treinamento, para que fosse coletado um frame de cada grupo. Essa abordagem assegurou que, em cada época, a rede fosse exposta a uma variedade maior de padrões acústicos provenientes de diferentes arquivos, aumentando significativamente a diversidade do conjunto de treinamento. O digrama da Figura 4.9 apresenta este processo de treinamento do FCN.

O novo treinamento foi iniciado a partir dos pesos obtidos na primeira tentativa (mesmo com baixo desempenho), acelerando a convergência. O resultado dessa segunda abordagem foi notavelmente superior: conforme ilustrado nas Figuras 4.10 e 4.11, o modelo alcançou valores de RPA superiores a 0,98, com estabilidade ao longo das épocas. É necessário destacar que este desempenho superior ao do CREPE mostrado na figura 4.5 se deve ao fato de que o CREPE foi treinado do zero, sem pesos pré-inicializados, enquanto que para o FCN utilizou-se os pesos originais do modelo para realizar *transfer learning*, o que favoreceu o processo de convergência do treinamento. Também foi efetuado teste na base de dados de flauta para verificar

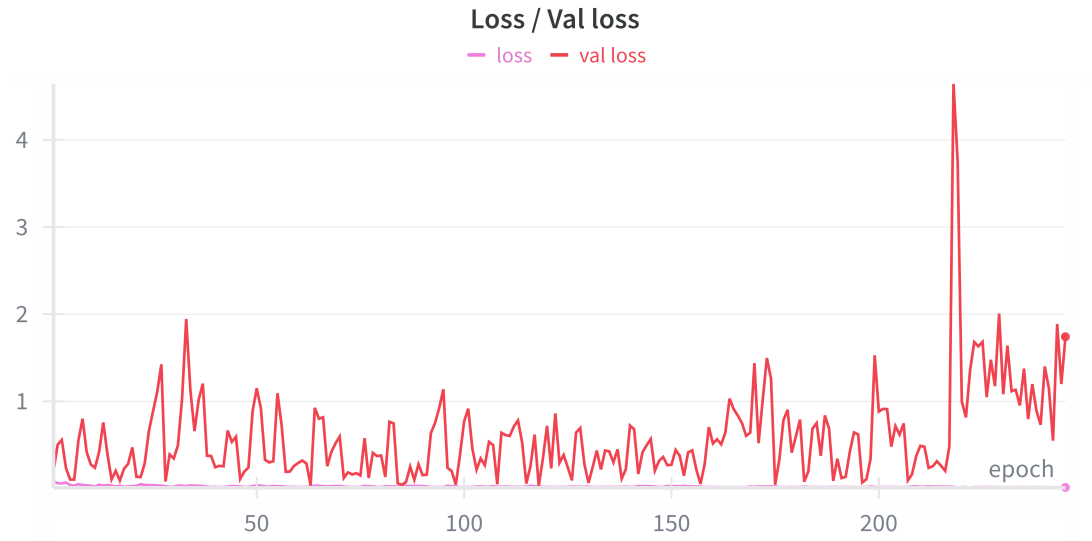

Figura 4.8: Curvas de perda de treinamento e validação do treinamento do FCN.

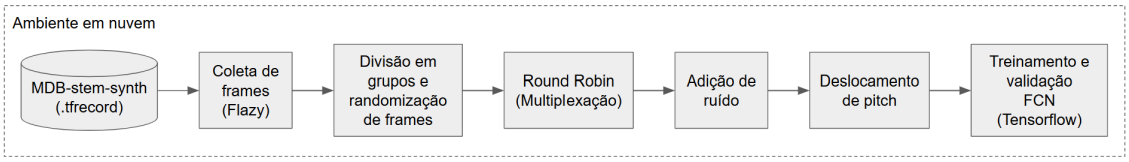

Figura 4.9: Diagrama de treinamento otimizado do FCN.

o desempenho após o treinamento e comparar com os resultados obtidos na Seção 3.2.4.3. A Figura 4.12 apresenta os resultados de RPA para cada arquivo de áudio da base *Traditional Flute Dataset*, que mostram melhoria esperada no caso de qualquer valor de confiança acima de 0.

Esse experimento foi particularmente relevante para o desenvolvimento do CREME, pois demonstrou que a estratégia de divisão da base aliada ao *round robin* não apenas mitigou problemas de *overfitting*, como também aumentou significativamente a robustez do treinamento. Essa técnica, validada com sucesso no FCN, foi incorporada como elemento central na metodologia de treinamento do CREME.

4.3.2 Construção da Base de Dados *MDB-stem-synth-multi*

Geração da base polifônica anotada

Para o treinamento do modelo CREME, foi utilizada como base de referência a *MDB-stem-synth*, escolhida por possuir anotações precisas derivadas de dados

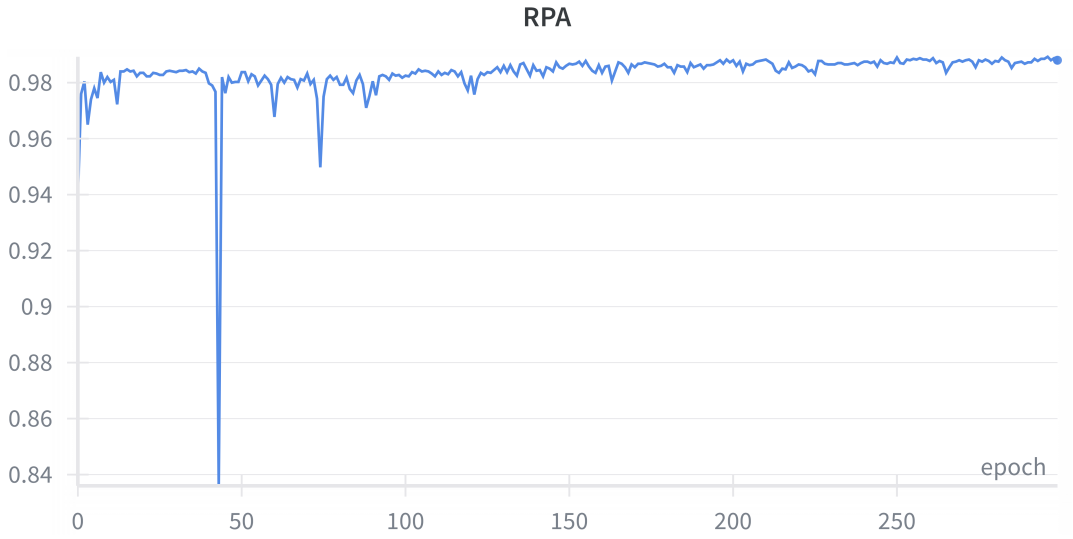

Figura 4.10: RPA do treinamento corrigido do FCN.

sintéticos. Essa característica garante marcações de frequência corretas para cada *frame* de áudio, fator fundamental para um treinamento supervisionado confiável. Outro motivo para a escolha é que essa mesma base foi utilizada no treinamento do CREPE, permitindo a comparação de resultados com maior coerência metodológica.

A *MDB-stem-synth* contém 230 *stems*, que são faixas individuais de uma música contendo elementos como bateria, baixo, vocais e outros instrumentos, permitindo que sejam mixados ou processados separadamente. Quando reproduzidos em conjunto, esses *stems* reconstituem a música original. Os arquivos seguem uma nomenclatura padronizada que indica a qual música cada *stem* pertence, o que possibilita a identificação e manipulação sistemática dos dados.

Para a construção do conjunto polifônico, foi utilizada a biblioteca *Pydub*, para manipulação e processamento de áudio em *Python*. O objetivo foi gerar arquivos de música contendo até duas frequências simultâneas e suas respectivas anotações, simulando um cenário polifônico controlado. Um ponto crítico desse processo foi garantir que as combinações fossem sempre realizadas entre *stems* da mesma música. Essa restrição assegura que a harmonia, coerência tonal e sincronismo de notas sejam preservados, resultando em exemplos mais próximos de contextos musicais reais. Combinar *stems* de músicas diferentes produziria arquivos acusticamente artificiais e pouco representativos, equivalentes a reproduzir duas músicas distintas ao mesmo tempo, o que tornaria a tarefa de detecção de frequências artificialmente

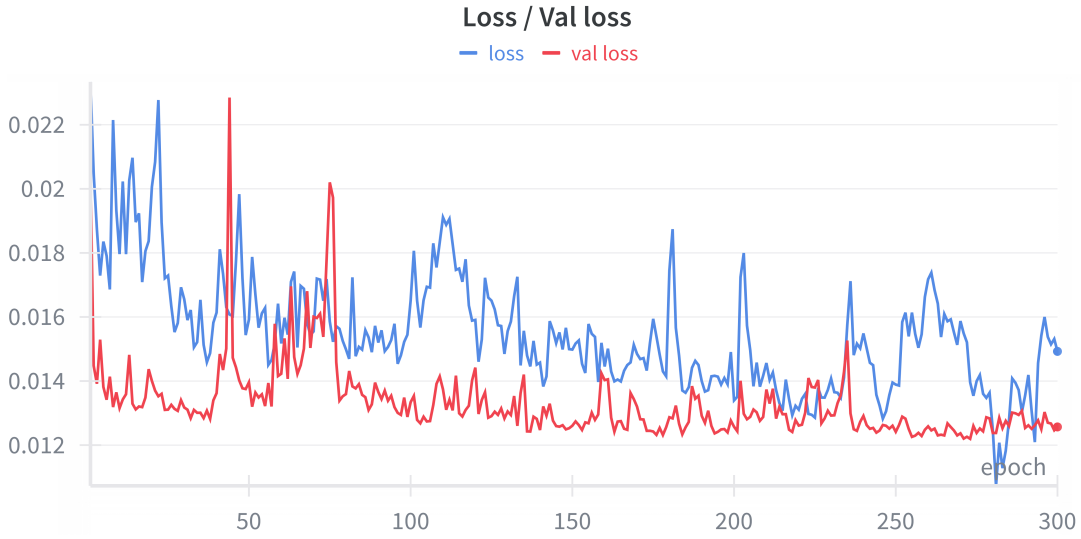

Figura 4.11: Curvas de perda de treinamento e validação do treinamento corrigido do FCN.

mais simples.

Como exemplo desse processo, podemos citar os *stems* do arquivo de música AClassicEducation_NightOwl presente na base *MDB-stem-synth*:

- AClassicEducation_NightOwl_STEM_01.RESYN.wav
- AClassicEducation_NightOwl_STEM_08.RESYN.wav
- AClassicEducation_NightOwl_STEM_13.RESYN.wav

As combinações geradas foram:

- AClassicEducation_NightOwl_STEM_01_STEM_08.RESYN.wav
- AClassicEducation_NightOwl_STEM_01_STEM_13.RESYN.wav
- AClassicEducation_NightOwl_STEM_08_STEM_13.RESYN.wav

A quantidade de stems por música varia na base *MDB-stem-synth*, e o processo de geração das combinações resultou em 453 arquivos de áudio com suas respectivas anotações, cada um contendo a indicação de duas frequências simultâneas para cada *frame* temporal. Esse novo conjunto de dados foi denominado *MDB-stem-synth-multi*, nome que será utilizado nas referências posteriores deste trabalho.

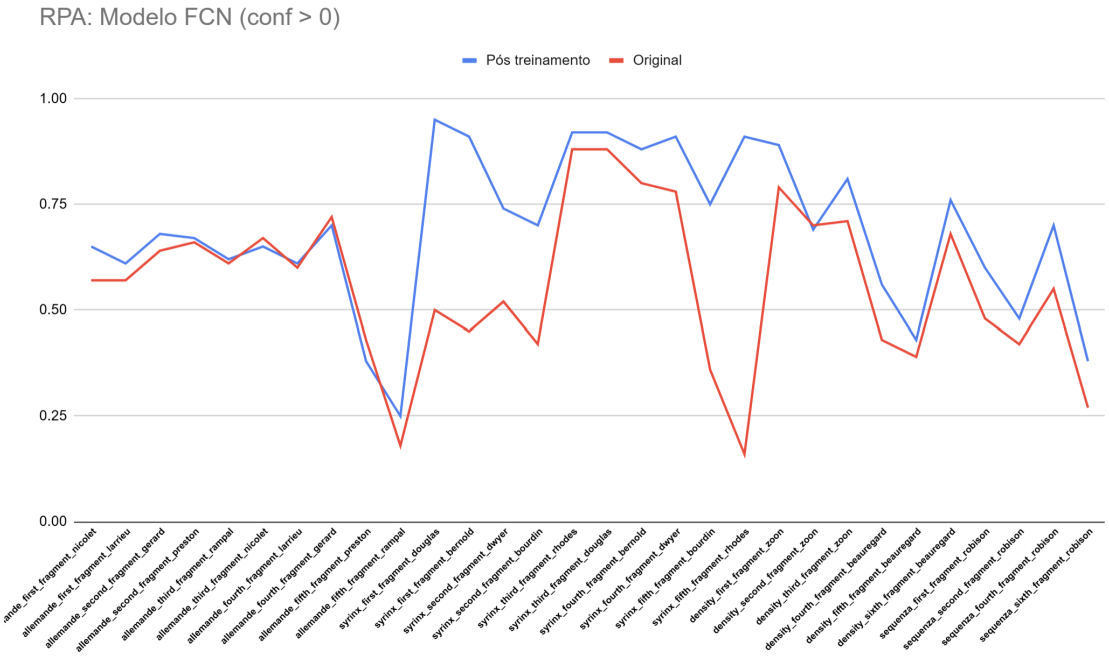

Figura 4.12: Comparação do RPA para o modelo FCN após treinamento com base de música.

Separação entre dados de treino e validação

Seguindo a estratégia que se mostrou eficiente no retreinamento do FCN, a base *MDB-stem-synth-multi* foi dividida em sete grupos com 60 arquivos cada, destinados ao treinamento, e um grupo adicional com 33 arquivos, destinado à validação. Essa organização permite combinar randomização de *frames* com a estratégia *round robin*, assegurando que o modelo seja exposto, ao longo das épocas, a amostras representativas de diferentes partes da base. O objetivo é maximizar a diversidade dos padrões observados pela rede e, consequentemente, favorecer a generalização do modelo.

4.3.3 Implementação do Modelo CREME

Ajustes na arquitetura original do CREPE para lidar com múltiplos *pitches*

O modelo CREME foi concebido com o objetivo de generalizar a proposta original do CREPE, permitindo a estimativa simultânea de mais de uma frequência fundamental em cada *frame* de áudio. Para sua implementação, partiu-se do código

previamente desenvolvido para o retreinamento do CREPE, realizando adaptações específicas na arquitetura e nas funções de saída de forma a possibilitar a detecção de múltiplos *pitches*. Inicialmente, havia o interesse em adaptar o modelo FCN para lidar com o caso de múltiplas frequências, por entender que o CREPE já havia sido explorado por este trabalho e, assim, o FCN poderia ser uma escolha mais inovadora. Entretanto, durante os testes com a base de flauta (Tabelas 3.1 e 3.5), verificou-se que o desempenho do CREPE o tornava um candidato mais promissor para servir como base para a implementação do CREME.

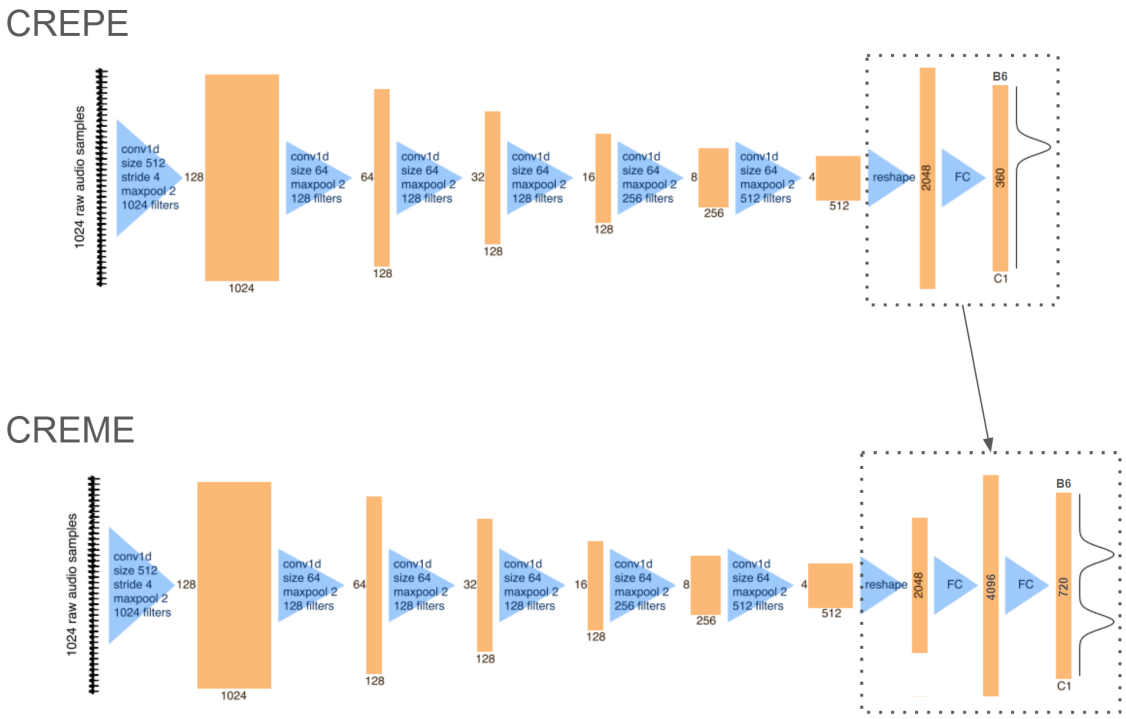

Figura 4.13: CREME: *A Convolutional Representation for Multipitch Estimation*.

A Figura 4.13 apresenta a arquitetura do modelo CREME, destacando as diferenças em relação ao CREPE. As modificações envolveram tanto ajustes estruturais no modelo quanto alterações no dimensionamento da saída para lidar com mais *bins* correspondentes a candidatos de frequência, conforme os detalhes descritos a seguir.

Estrutura de entrada e camadas convolucionais

O tamanho de entrada e a organização das camadas convolucionais do CREPE foram mantidos no CREME. A rede recebe *frames* de áudio de tamanho 1024 amos-

tras, processados por seis camadas convolucionais. Em cada camada, preservou-se a quantidade de filtros, seus tamanhos, parâmetros de *max pooling*, *stride* e *padding*, bem como a função de ativação $ReLU$ utilizada na saída de cada bloco convolucional.

A decisão de manter essa estrutura decorreu da lógica que inspirou a concepção original do CREPE, fortemente relacionada à análise espectral baseada na transformada discreta de Fourier (FFT). A FFT correlaciona o sinal de entrada com 1024 coeficientes exponenciais complexos, cada um representando uma frequência, a primeira camada convolucional do CREPE aplica 1024 filtros sobre o *frame* de entrada. Os pontos em que a correlação entre filtro e *frame* é elevada indicam a presença de componentes harmônicos específicos. No trabalho original do CREPE, os autores exploram essa analogia ao apresentar visualizações dos filtros aprendidos, interpretando-os como padrões relacionados ao conteúdo harmônico do áudio de treinamento.

Dessa forma, assim como uma FFT de tamanho 1024 retorna 1024 *bins* de frequência, a arquitetura original do CREPE já possui coerência estrutural para lidar com múltiplos componentes harmônicos, mesmo que tenha sido inicialmente projetada para estimar apenas uma frequência por *frame*.

Alterações nas camadas finais e na dimensão de saída

Após as camadas convolucionais, o CREPE original possui uma camada totalmente conectada (*fully connected*) de 2048 unidades, ligada diretamente à camada de saída de 360 neurônios. Esses 360 *bins* cobrem o range de frequências correspondente ao intervalo de notas de C1 a B6 (6 oitavas), representando as frequências candidatas à predição.

No CREME, essa parte final da arquitetura foi modificada em dois aspectos principais:

1. Camada intermediária adicional: entre a camada de 2048 unidades e a saída, foi inserida uma nova camada totalmente conectada com 4096 unidades. A motivação foi aumentar a capacidade da rede de aprender padrões hierárquicos mais complexos, requisito essencial para distinguir simultaneamente múltiplos conteúdos harmônicos.

2. Ampliação da camada de saída: o tamanho da saída foi expandido de 360 para 720 neurônios. Essa modificação teve como objetivo aumentar a resolução da rede, já que a detecção de múltiplas frequências por frame exige maior detalhamento para discriminar notas próximas, evitando a superposição de funções gaussianas usadas suavizar as anotações de frequência (detalhes na próxima Seção). A duplicação do número de *bins* amplia a granularidade da representação, potencialmente permitindo que o modelo capture variações mais sutis no espectro. Assim, a resolução da saída do modelo é:

$$6 \text{ oitavas} \times 12 \text{ semitons por oitava} \times 10 \textit{ bins} \text{ por semitom} = 720 \textit{ bins} \quad (4.1)$$

$$\text{Resolução por } \textit{bin} = \frac{100 \textit{ cents} \text{ por semitom}}{10 \textit{ bins} \text{ por semitom}} = 10 \textit{ cents} \text{ por } \textit{bin} \quad (4.2)$$

Essas alterações estruturais foram empiricamente realizadas na expectativa de se criar uma arquitetura capaz de lidar com frequências simultâneas, mantendo ao mesmo tempo a lógica central do CREPE, mas adaptando-a para cenários polifônicos controlados, que no escopo deste trabalho se concentra no caso de duas frequências.

Adaptação da função de saída e geração de rótulos para múltiplas frequências

No modelo CREPE, os rótulos de treinamento são gerados a partir de uma função gaussiana centrada na frequência fundamental real anotada para cada *frame*. Essa função suaviza os valores em torno da frequência de referência, atribuindo valores mais altos aos *bins* próximos à frequência verdadeira e valores decrescentes à medida que se afastam desse ponto. O pico da gaussiana ocorre no *bin* cuja frequência estimada está mais próxima do valor anotado.

Para o CREME, essa estratégia foi generalizada para lidar com múltiplos *pitches* por *frame*. Em vez de uma única gaussiana, utiliza-se uma soma de gaussianas independentes, cada uma centrada em uma das frequências presentes na anotação. Assim, o rótulo alvo do treinamento passa a ser a superposição dessas funções, permitindo que a rede aprenda a identificar mais de um pico simultaneamente.

A equação do CREPE para um único *pitch*, já apresentada na equação (2.16), é dada por

$$y_i = \exp\left(-\frac{(c_i - c_{\text{true}})^2}{2 \cdot 25^2}\right), \tag{4.3}$$

onde y_i é o valor do rótulo no *bin* i, c_i é a frequência correspondente ao *bin* em *cents* e c_{true} é a frequência verdadeira anotada, também em *cents*.

Para o CREME, considerando K frequências verdadeiras presentes em um mesmo *frame*, a equação é generalizada para

$$y_i = \sum_{k=1}^{K} \exp\left(-\frac{(c_i - c_{\text{true},k})^2}{2 \cdot 25^2}\right), \tag{4.4}$$

em que $c_{\text{true},k}$ representa a k-ésima frequência verdadeira em *cents*. Essa formulação gera uma curva alvo com múltiplos picos, um para cada frequência presente, possibilitando que a rede aprenda a produzir saídas multimodais. Vale notar que para $K = 1$ esta equação do CREME se reduz ao caso do CREPE.

Adaptação do cálculo de frequência em *cents* na predição

No CREPE, o cálculo do valor de frequência estimada em *cents* é realizado por meio de uma média ponderada centrada no pico máximo da saída, utilizando os valores de ativação dos *bins* vizinhos como pesos, conforme já visto na equação (2.14):

$$\hat{c} = \frac{\sum_{i=M-4}^{M+4} \hat{y}_i c_i}{\sum_{i=M-4}^{M+4} \hat{y}_i}, \quad M = \text{argmax}_i(\hat{y}_i). \tag{4.5}$$

No CREME, essa lógica permanece válida, mas deve ser aplicada a cada pico identificado na saída. Isso significa que, para cada *frame*, o modelo retorna múltiplos picos, e o cálculo em *cents* é executado iterativamente para cada um deles. Dessa forma, obtém-se a lista completa das frequências estimadas para o intervalo temporal analisado.

Função de perda utilizada

A função de perda escolhida para o CREME é a mesma utilizada no CREPE: entropia cruzada binária (BCE, do inglês *binary cross-entropy*). É importante res-

saltar que, neste contexto, a BCE não deve ser interpretada como no caso clássico de classificação com rótulos binários *one-hot encoded*. No CREPE e no CREME, os valores-alvo são funções gaussianas contínuas, com valores variando livremente entre zero e um, o que caracteriza um problema de regressão.

O uso da BCE nesse cenário busca aproximar a saída contínua da rede às curvas-alvo, de modo que os picos previstos estejam alinhados com as frequências reais anotadas. Essa abordagem é particularmente adequada para estimação de frequência fundamental, pois acomoda variações naturais de afinação que ocorrem em sinais musicais reais.

4.3.4 Procedimentos de Treinamento

O treinamento do modelo CREME foi realizado utilizando a base de dados polifônica anotada *MDB-stem-synth-multi*, construída a partir da base original *MDB-stem-synth* e contendo até duas frequências por *frame*, conforme o procedimento descrito na Seção 4.3.2. Essa base preserva a coerência harmônica dos sinais e sincronismo de notas, pois as combinações de *stems* foram realizadas apenas entre faixas pertencentes à mesma música, garantindo maior proximidade com situações reais em que múltiplas frequências coexistem de forma harmônica. A Figura 4.14 apresenta o diagrama para treinamento do CREME.

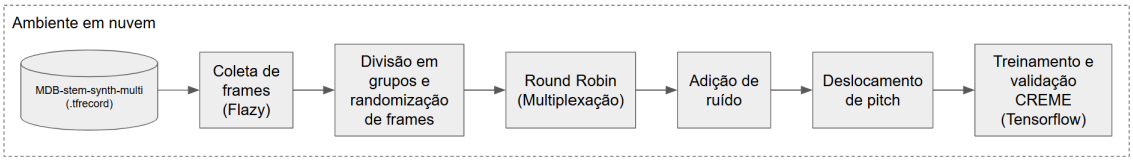

Figura 4.14: Diagrama para treinamento do CREME com a base de dados *MDB-stem-synth-multi*.

A organização da base seguiu a estratégia de divisão em oito grupos, sendo sete grupos com 60 arquivos destinados ao treinamento e um grupo com 33 arquivos para validação. A cada época, os grupos eram alternados de forma cíclica (*round robin*), garantindo que o modelo tivesse contato com todo o conjunto de dados ao longo do treinamento. Além disso, foram aplicadas técnicas de aumento de dados: adição de ruído em diferentes intensidades aleatórias, deslocamento aleatório de *pitch* e randomização de *frames* de áudio, a fim de aumentar a diversidade do conjunto

de treinamento e melhorar a capacidade de generalização do modelo.

A implementação do treinamento foi baseada no código utilizado para reprocessar o CREPE, aproveitando os pesos originais desse modelo como ponto de partida (*transfer learning*). As adaptações realizadas para o CREME incluíram a modificação da arquitetura para lidar com múltiplos *pitches* e a geração de rótulos multi-gaussianos, conforme descrito na Seção 4.3.3.

O treinamento foi conduzido com *batches* de 32 amostras, por um total de 500 épocas, sem aplicação de *early stopping*. Essa escolha foi motivada pelo objetivo de salvar todos os pesos gerados, permitindo análises posteriores de desempenho e a retomada do treinamento a partir de qualquer época específica, especialmente em casos onde a métrica de avaliação pudesse indicar que um determinado ponto do treinamento ofereceu melhor desempenho. A taxa de aprendizado foi mantida fixa em 0,0002 durante todo o procedimento, utilizando o otimizador *Adam*. A função de perda adotada foi a entropia cruzada binária (BCE), trabalhando com saídas contínuas entre 0 e 1, adequadas para regressão sobre distribuições gaussianas centradas nas frequências de referência, conforme apresentado na Seção 4.3.3.

A avaliação do desempenho foi feita usando a métrica RPA com tolerância de 50 *cents* para considerar uma frequência correta, adaptada para o contexto de múltiplas frequências. Como a saída do modelo fornece apenas os valores de pico, sem identificação direta da correspondência com as frequências anotadas, foi necessário aplicar um procedimento de associação: cada frequência estimada foi vinculada à frequência mais próxima presente na anotação do *frame*, formando pares correspondentes. A partir dessas correspondências, o valor de RPA foi calculado de forma acumulada para todas as frequências identificadas, representando assim a precisão global do modelo na detecção simultânea de múltiplos *pitches*.

A Figura 4.16 apresenta as curvas de aprendizado referentes à perda de treinamento e validação ao longo das 500 épocas de treinamento do CREME. Observa-se que ambas as curvas apresentam tendência de convergência, o que indica que o modelo conseguiu generalizar de forma consistente para o conjunto de validação, composto por dados não utilizados durante o treinamento.

Além disso, o desempenho medido pela métrica RPA apresentou valores elevados desde o início do treinamento, partindo de aproximadamente 0,89 e atingindo

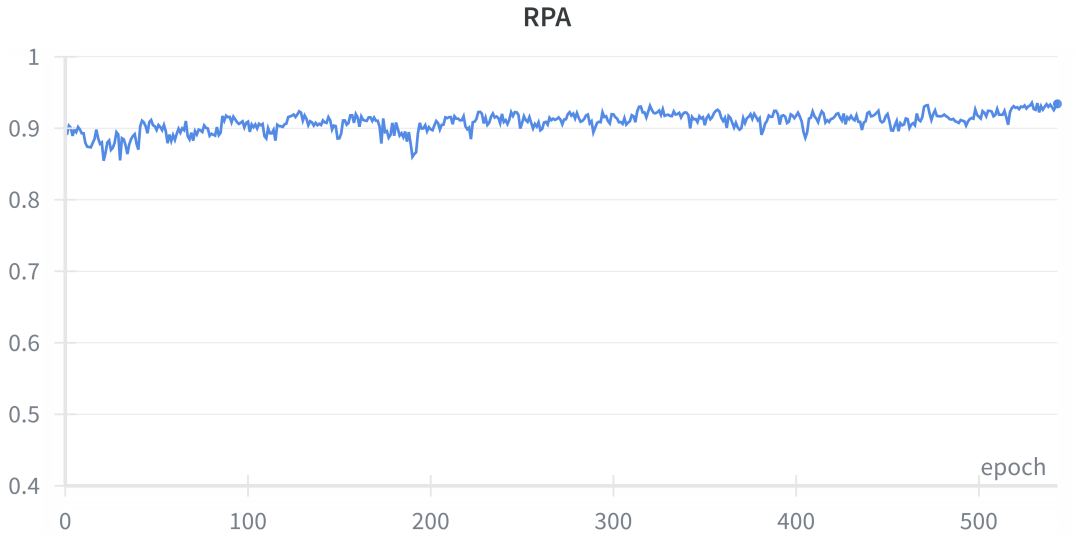

Figura 4.15: RPA do treinamento do CREME.

valores acima de 0,93 ao final do processo, conforme apresentando na Figura 4.15. Esse resultado evidencia a boa capacidade do CREME em estimar com precisão múltiplas frequências simultaneamente, reforçando a eficácia das modificações propostas em relação ao modelo CREPE original.

4.4 Avaliação experimental

4.4.1 Avaliação com dados de validação

Na Tabela 4.1 são apresentados os valores de RPA obtidos para o conjunto de dados de validação, extraídos da base *MDB-stem-synth-multi*. Esse conjunto não foi utilizado pelo CREME durante o processo de treinamento, garantindo assim uma avaliação imparcial do desempenho. Os resultados evidenciam a alta precisão do modelo, com valores variando entre um mínimo de 0,848 e um máximo de 0,992, resultando em uma média global de 0,9341 $\pm$ 0,0398. Esses valores confirmam a capacidade do CREME de estimar corretamente múltiplas frequências simultâneas.

4.4.2 Avaliação pelos indicadores de erro

Além da métrica RPA, foram avaliados três indicadores complementares, que fornecem uma visão mais detalhada do desempenho da rede no contexto de estima-

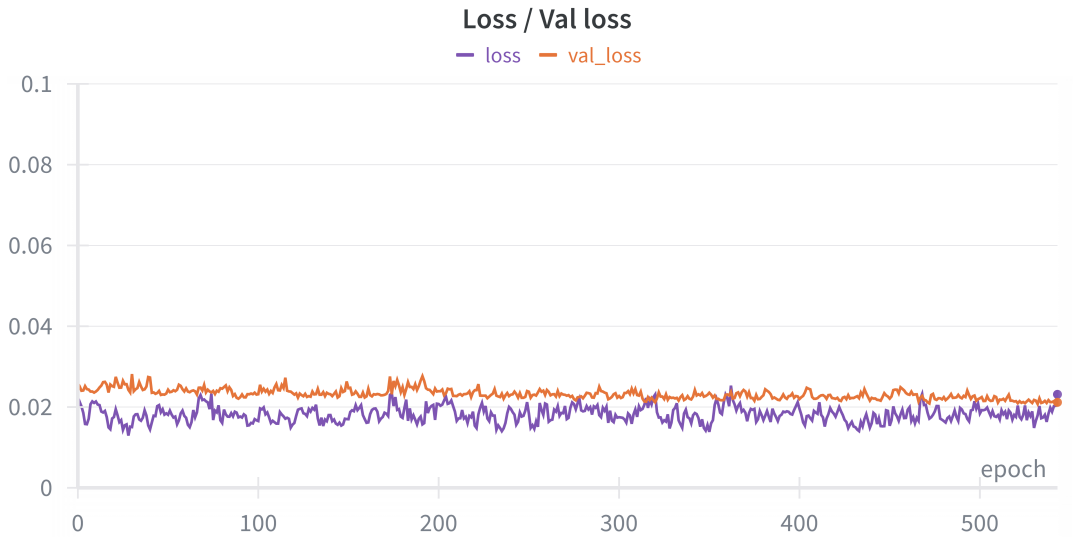

Figura 4.16: Curvas de perda de treinamento e validação do treinamento do CREME.

tiva de múltiplos *pitches*:

- Erro por falta (*Miss Error*): ocorre quando a rede estima um número de frequências menor do que o número presente na anotação de referência para um determinado *frame*. Nesse caso, uma ou mais frequências reais não são detectadas pelo modelo.

- Alarme falso (*False Alarm*): ocorre quando a rede estima um número de frequências maior do que o número presente na anotação. Nesse cenário, o modelo adiciona frequências inexistentes, que não fazem parte do sinal real.

- Erro de substituição (*Substitution Error*): ocorre quando a rede e a anotação apresentam a mesma quantidade de frequências para um *frame*, mas uma ou mais delas estão fora da tolerância aceitável para serem consideradas corretas. Essa situação indica que a frequência foi prevista, mas de forma incorreta em relação ao valor real.

A utilização desses três indicadores permite uma avaliação mais abrangente e de caráter diagnóstico, funcionando como uma verificação de sanidade do modelo. No caso de múltiplos *pitches*, tais métricas ajudam a identificar se o erro predominante é decorrente da perda de frequências reais, da adição de frequências inexisten-

Nome do arquivo	RPA
MusicDelta_LatinJazz_STEM_01_STEM_04.RESYN.wav	0,8731
MusicDelta_LatinJazz_STEM_01_STEM_05.RESYN.wav	0,9610
MusicDelta_LatinJazz_STEM_04_STEM_05.RESYN.wav	0,8734
MusicDelta_ModalJazz_STEM_02_STEM_04.RESYN.wav	0,9601
MusicDelta_ModalJazz_STEM_02_STEM_05.RESYN.wav	0,9630
MusicDelta_ModalJazz_STEM_04_STEM_05.RESYN.wav	0,9217
MusicDelta_Pachelbel_STEM_01_STEM_02.RESYN.wav	0,9584
MusicDelta_Pachelbel_STEM_01_STEM_03.RESYN.wav	0,8794
MusicDelta_Pachelbel_STEM_01_STEM_04.RESYN.wav	0,8484
MusicDelta_Pachelbel_STEM_02_STEM_03.RESYN.wav	0,9674
MusicDelta_Pachelbel_STEM_02_STEM_04.RESYN.wav	0,9364
MusicDelta_Pachelbel_STEM_03_STEM_04.RESYN.wav	0,8521
MusicDelta_Punk_STEM_02_STEM_04.RESYN.wav	0,8888
MusicDelta_Reggae_STEM_02_STEM_04.RESYN.wav	0,9169
MusicDelta_Rock_STEM_02_STEM_05.RESYN.wav	0,9107
MusicDelta_Rockabilly_STEM_02_STEM_05.RESYN.wav	0,9303
MusicDelta_SwingJazz_STEM_01_STEM_04.RESYN.wav	0,9681
MusicDelta_Vivaldi_STEM_01_STEM_03.RESYN.wav	0,9699
NightPanther_Fire_STEM_01_STEM_07.RESYN.wav	0,9615
PortStWillow_StayEven_STEM_08_STEM_10.RESYN.wav	0,9922
SecretMountains_HighHorse_STEM_01_STEM_09.RESYN.wav	0,9562
Snowmine_Curfews_STEM_02_STEM_03.RESYN.wav	0,9815
StevenClark_Bounty_STEM_02_STEM_07.RESYN.wav	0,9437
StevenClark_Bounty_STEM_02_STEM_08.RESYN.wav	0,9134
StevenClark_Bounty_STEM_07_STEM_08.RESYN.wav	0,9540
StrandOfOaks_Spacestation_STEM_01_STEM_04.RESYN.wav	0,9807
SweetLights_YouLetMeDown_STEM_02_STEM_08.RESYN.wav	0,9671
TheDistricts_Vermont_STEM_01_STEM_05.RESYN.wav	0,9428
TheScarletBrand_LesFleursDuMal_STEM_02_STEM_08.RESYN.wav	0,9440
TheSoSoGlos_Emergency_STEM_01_STEM_05.RESYN.wav	0,9044

Tabela 4.1: RPA de estimativas do CREME para base *MDB-stem-synth-multi*.

tes ou da substituição incorreta de valores próximos. O cálculo destes indicadores foi realizado utilizando a biblioteca *Mir Eval*.

Nas Figuras 4.17, 4.18 e 4.19 são apresentados os valores de erro por falta, alarme falso e erro de substituição obtidos para todo o conjunto *MDB-stem-synth-multi*. Observa-se que, entre os três indicadores, o modelo apresentou maior fragilidade no erro por falta, evidenciando que a principal dificuldade do CREME está relacionada à detecção incompleta de frequências presentes no sinal real.

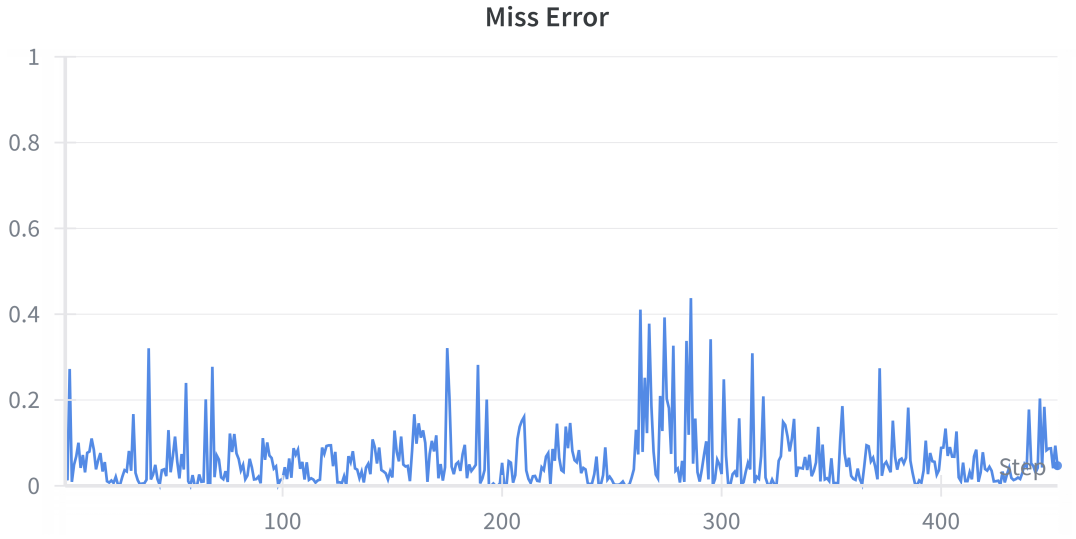

Figura 4.17: CREME: Erro por falta para teste com a base *MDB-stem-synth-multi*.

4.4.3 Teste com base de flauta *Traditional Flute Dataset*

Com o objetivo de avaliar o desempenho do CREME em um cenário monofônico, isto é, restrito a apenas uma frequência presente por *frame*, foi utilizado o conjunto de dados *Traditional Flute Dataset*, previamente descrito e empregado nos testes da Seção 3.2.3. Esse tipo de avaliação permite estabelecer uma comparação direta com o modelo CREPE.

Esse teste se mostra particularmente relevante, pois possibilita verificar se o treinamento do CREME com dados contendo duas frequências por *frame* comprometeu ou não o desempenho em situações mais simples, nas quais existe apenas um *pitch* a ser estimado. Em outras palavras, espera-se que o CREME, mesmo sendo treinado para lidar com múltiplas frequências, apresente desempenho semelhante ao CREPE quando confrontado com sinais monofônicos.

Os resultados apresentados na Tabela 4.2 indicam que o RPA obtido pelo CREME permanece consistente com aquele alcançado pelo CREPE (aqui repetindo os valores da Tabela 3.1). Observa-se que as diferenças entre os valores são pequenas, com uma média de variação de aproximadamente 0,025 44. Tal comportamento evidencia que a capacidade do CREME de estimar uma única frequência não foi prejudicada pelo treinamento voltado para múltiplos *pitches*.

A Figura 4.20 apresenta um gráfico de comparação entre as estimativas mo-

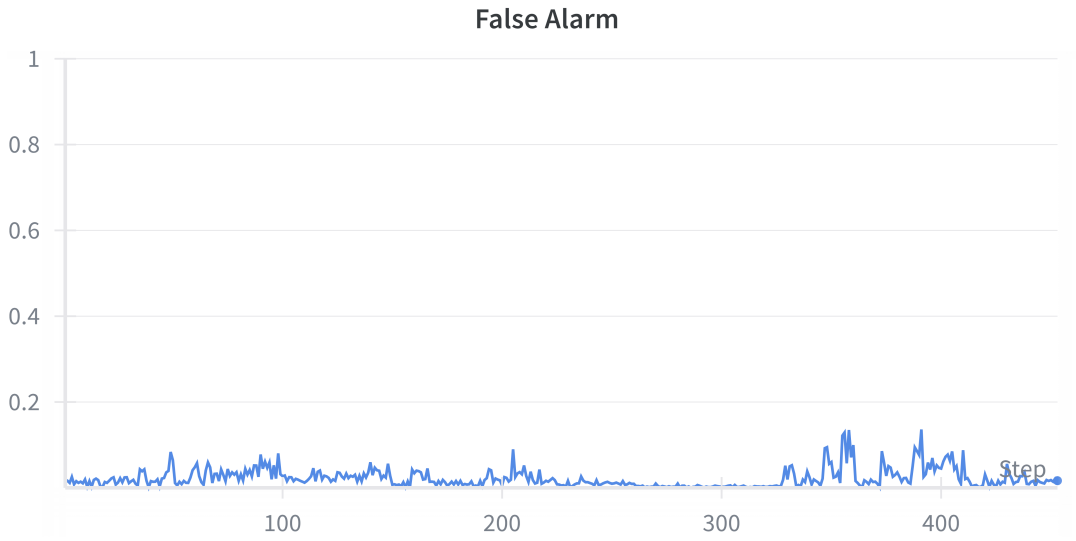

Figura 4.18: CREME: Alarme Falso para teste com a base *MDB-stem-synth-multi*.

nofônicas geradas pelo CREPE e pelo CREME. Esse resultado reforça a robustez do CREME, demonstrando que o modelo consegue generalizar para contextos mais simples sem perdas significativas de desempenho em relação ao modelo original.

4.4.4 Teste com base *Bach10-mf0-synth*

Para novamente avaliar o desempenho do CREME em um contexto polifônico semelhante ao caso apresentado na Seção 4.4.1, foi utilizada a base sintética *Bach10-mf0-synth*. Esta base é composta por 10 peças de música clássica de Johann Sebastian Bach, originalmente presentes na base Bach10, porém resintetizadas de forma a gerar anotações de *pitch* com precisão perfeita. A síntese foi realizada individualmente para cada instrumento (fagote, clarinete, saxofone e violino), produzindo arquivos de áudio e anotações separados para cada parte instrumental. Os arquivos são organizados de modo a indicar a peça a que pertencem, tanto em suas versões individuais quanto nas mixagens utilizadas para compor as versões polifônicas.

Com o objetivo de adaptar essa base para o contexto de múltiplos *pitches* do CREME, foi aplicado o mesmo procedimento descrito na Seção 4.3.2 para a geração da base *MDB-stem-synth-multi*. Em resumo, o processo consistiu em combinar dois arquivos de áudio da mesma peça, juntamente com suas respectivas anotações, para formar novos arquivos contendo dois instrumentos tocando simultaneamente.

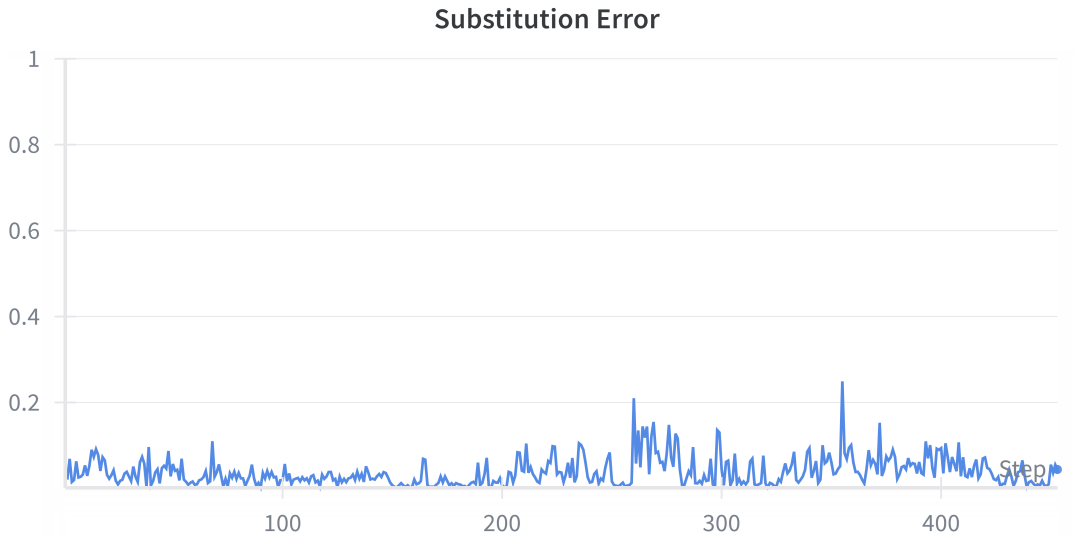

Figura 4.19: CREME: Erro de Substituição para teste com a base *MDB-stem-synth-multi*.

Dessa forma, cada *frame* da nova base resultante possui duas frequências anotadas, permitindo que o CREME fosse testado em condições controladas de polifonia com duas frequências.

Os resultados obtidos a partir dessa base podem ser vistos na Figura 4.21, que apresenta o resultado do RPA para os 60 arquivos gerados para teste. Nota-se que os valores de RPA superam 0,90 na maioria dos arquivos testados, evidenciando que o CREME mantém desempenho elevado ao lidar com material polifônico que não foi visto durante o treinamento. Este resultado reforça a capacidade de generalização do CREME e indica sua robustez no contexto envolvendo mais de uma fonte ativa no mesmo intervalo temporal.

4.4.5 Testes com a base de dados RWC (Violão e Clarinete)

O último conjunto de testes realizados com o modelo CREME teve como objetivo avaliar o desempenho da rede a partir de arquivos de áudio reais. Para isso foi utilizada a base de dados amplamente conhecida *Real World Computing Music Database* (RWC), que reúne uma extensa coleção de gravações de instrumentos musicais. Nos experimentos aqui descritos foram selecionados os arquivos de violão e clarinete, de forma a contemplar tanto um instrumento de corda quanto um instrumento de sopro.

Nome do arquivo	CREPE	CREME
allemande_first_fragment_nicolet	0.79	0.82
allemande_first_fragment_larrieu	0.84	0.80
allemande_second_fragment_gerard	0.84	0.87
allemande_second_fragment_preston	0.81	0.81
allemande_third_fragment_rampal	0.82	0.84
allemande_third_fragment_nicolet	0.79	0.82
allemande_fourth_fragment_larrieu	0.82	0.82
allemande_fourth_fragment_gerard	0.87	0.89
allemande_fifth_fragment_preston	0.61	0.51
allemande_fifth_fragment_rampal	0.47	0.39
syrinx_first_fragment_douglas	0.96	0.95
syrinx_first_fragment_bernold	0.97	0.97
syrinx_second_fragment_dwyer	0.97	0.97
syrinx_second_fragment_bourdin	0.96	0.95
syrinx_third_fragment_rhodes	0.94	0.91
syrinx_third_fragment_douglas	0.95	0.96
syrinx_fourth_fragment_bernold	0.89	0.93
syrinx_fourth_fragment_dwyer	0.94	0.93
syrinx_fifth_fragment_bourdin	0.80	0.80
syrinx_fifth_fragment_rhodes	0.96	0.90
density_first_fragment_zoon	0.98	0.98
density_second_fragment_zoon	0.95	0.84
density_third_fragment_zoon	0.94	0.85
density_fourth_fragment_beauregard	0.90	0.68
density_fifth_fragment_beauregard	0.59	0.49
density_sixth_fragment_beauregard	0.92	0.86
sequenza_first_fragment_robison	0.78	0.73
sequenza_second_fragment_robison	0.72	0.76
sequenza_fourth_fragment_robison	0.88	0.83
sequenza_sixth_fragment_robison	0.57	0.60

Tabela 4.2: Comparação de RPA entre CREPE e CREME para base de dados *Traditional Flute Dataset*.

Os arquivos da base RWC fornecem gravações isoladas de cada nota dos instrumentos. No caso do violão, o material disponibiliza a execução das notas percorrendo toda a extensão do instrumento, desde as cordas soltas até a décima segunda casa, abrangendo a faixa de notas de E2 (sexta corda solta) a E5 (décima segunda casa da primeira corda). Já para o clarinete, a base fornece a escala completa de notas isoladas, iniciando em D3 e alcançando até F6.

A partir desses arquivos foram conduzidos experimentos voltados para a análise dos limites de resolução do CREME, ou seja, a capacidade do modelo em distinguir notas cujas frequências estão muito próximas. Esse tipo de análise é essencial

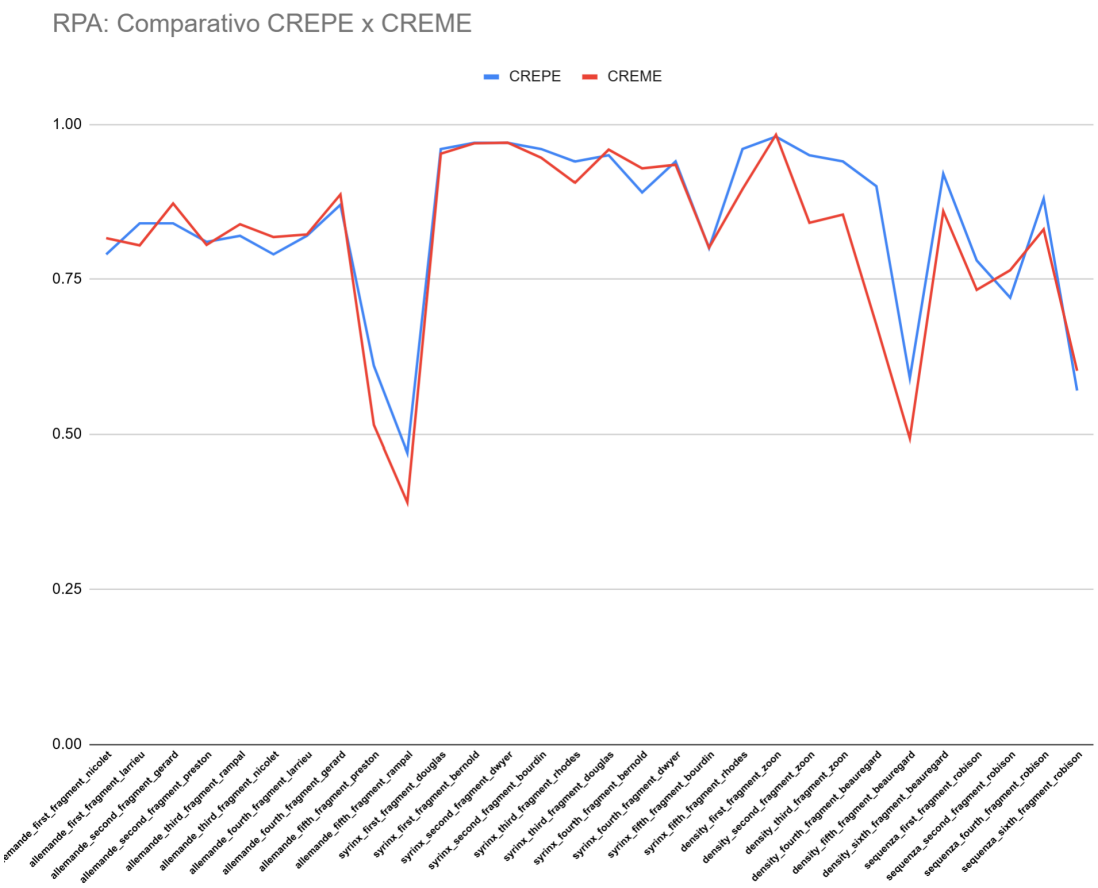

Figura 4.20: Gráfico de comparação entre RPA do CREPE e CREME para a base de dados *Traditional Flute Dataset*.

para verificar até que ponto o modelo consegue separar duas componentes fundamentais em situações de alta proximidade espectral.

Violão

Experimento 1

No caso do violão, foram planejados três cenários de teste. O primeiro cenário, apresentado a seguir, corresponde ao caso mais crítico: a análise das regiões graves, onde as notas naturalmente apresentam frequências mais próximas entre si. Para esse fim, foram extraídas as gravações da sexta corda e construídos pares de notas combinadas artificialmente usando o software Audacity. O procedimento consistiu em fixar a nota mais grave, E2, e combiná-la sucessivamente com notas subsequentes até o limite disponível na base, gerando pares como E2/F2, E2/F#2,

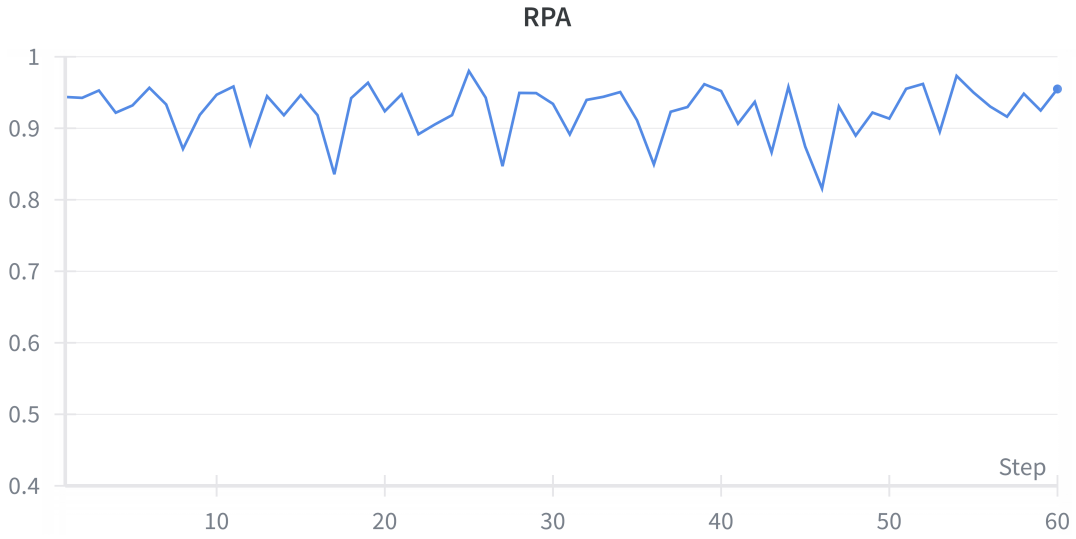

Figura 4.21: Valores de RPA para teste do CREME com a base *Bach10-mf0-synth*.

E2/G2, E2/G#2, até atingir E2/E3. Assim, foi possível avaliar o desempenho do modelo em função da distância entre as frequências, medindo o RPA e identificando a partir de qual intervalo o modelo é capaz de distinguir adequadamente as notas.

Os resultados obtidos para este cenário são apresentados na Tabela 4.3. Os valores de RPA indicam que o modelo não apresentou desempenho satisfatório, alcançando em diversos casos valores inferiores a 0.5, mesmo quando as notas já estavam suficientemente distantes para serem diferenciadas. Contudo, este resultado pode ser interpretado como esperado, considerando as características do treinamento do CREME. Como discutido anteriormente, o modelo foi treinado com bases de áudio que privilegiam combinações com coerência harmônica, ou seja, situações em que as frequências presentes pertencem a uma mesma música e mantêm entre si relações de consonância.

Esse contraste explica o fraco desempenho diante das combinações artificiais do violão, que envolvem intervalos considerados dissonantes, caracterizados por baixa correlação harmônica entre as frequências. Uma evidência dessa hipótese pode ser observada no mesmo gráfico para o caso do par E2/B2. Esse intervalo, denominado na teoria musical como quinta justa, é classificado como a combinação mais consonante, já que corresponde à relação natural entre uma frequência fundamental (E2) e seu terceiro harmônico (B2). Nessa condição, o modelo obteve o melhor desempenho entre os testes, alcançando valor de RPA superior a 0,8, o que

Nome do arquivo	Intervalo	Semitons	RPA
1-guitar-e2-f2.wav	E2/F2	1	0,4596
2-guitar-e2-fs2.wav	E2/F#2	2	0,3947
3-guitar-e2-g2.wav	E2/G2	3	0,4824
4-guitar-e2-gs2.wav	E2/G#2	4	0,4771
5-guitar-e2-a2.wav	E2/A2	5	0,4017
6-guitar-e2-as2.wav	E2/A#2	6	0,2157
7-guitar-e2-b2.wav	E2/B2	7	0,8157
8-guitar-e2-c3.wav	E2/C3	8	0,5754
9-guitar-e2-cs3.wav	E2/C#3	9	0,5964
10-guitar-e2-d3.wav	E2/D3	10	0,3526
11-guitar-e2-ds3.wav	E2/D#3	11	0,6666
12-guitar-e2-e3.wav	E2/E3	12	0,7521

Tabela 4.3: RPA das estimativas do CREME: violão/sexta corda/intervalo crescente.

reforça a interpretação de que o CREME está otimizado para capturar padrões de combinações harmônicas típicas de contextos musicais reais.

Experimento 2

No segundo cenário, foi conduzido um experimento análogo ao descrito para a região grave, mas agora direcionado à região aguda do violão. Nesse caso, a nota de referência foi a mais aguda (E5) disponível na base RWC, correspondente à décima segunda casa da primeira corda. A partir dela foram formados pares em combinação regressiva com notas mais graves, iniciando com E5/D#5, em seguida E5/D5, E5/C#5, e assim sucessivamente até atingir o limite do braço do instrumento, chegando ao par E5/E4 (corda solta). O objetivo permaneceu o mesmo: identificar a distância mínima entre notas em que o modelo CREME consegue distinguir duas frequências com maior precisão, mas agora em uma situação considerada menos desafiadora em relação à região grave, uma vez que, nas notas mais agudas, a separação em frequência é naturalmente maior.

Os resultados desse experimento estão apresentados na Tabela 4.4. De modo semelhante ao observado no primeiro cenário, o desempenho do modelo foi limitado diante das combinações artificiais geradas sem correlação harmônica, resultando, na maioria dos casos, em valores de RPA inferiores a 0,5. Diferentemente do teste na região grave, não foi possível observar aqui o comportamento de destaque associado à consonância, como ocorreu no intervalo de quinta justa (E2/B2) analisado no

Nome do arquivo	Intervalo	Semitons	RPA
1-guitar-e5-ds5.wav	E5/D#5	1	0,4436
2-guitar-e5-d5.wav	E5/D5	2	0,6519
3-guitar-e5-cs5.wav	E5/C#5	3	0,6789
4-guitar-e5-c5.wav	E5/C5	4	0,4901
5-guitar-e5-b4.wav	E5/B4	5	0,5759
6-guitar-e5-as4.wav	E5/A#4	6	0,6397
7-guitar-e5-a4.wav	E5/A4	7	0,4828
8-guitar-e5-gs4.wav	E5/G#4	8	0,5196
9-guitar-e5-g4.wav	E5/G4	9	0,5171
10-guitar-e5-fs4.wav	E5/F#4	10	0,6053
11-guitar-e5-f4.wav	E5/F4	11	0,2867
12-guitar-e5-e4.wav	E5/E4	12	0,5686

Tabela 4.4: RPA das estimativas do CREME: violão/primeira corda/intervalo crescente.

experimento anterior. Essa ausência pode ser explicada pelo fato de que a nota de referência neste caso é a mais aguda disponível na base, o que impossibilita a reprodução da relação entre frequência fundamental e seu terceiro harmônico, já que as gravações do RWC se limitam até o E5, enquanto o teste exigiu combinações apenas com notas mais graves.

Outro fator que pode ter contribuído para o baixo desempenho, mesmo em uma região onde as notas estão mais espaçadas, refere-se às propriedades acústicas do violão. As gravações do RWC podem apresentar maior influência das reverberações naturais do corpo do instrumento, além da ocorrência de ressonâncias espontâneas em cordas adjacentes. Esses fenômenos físicos produzem harmônicos adicionais que interferem diretamente nas estimativas realizadas pelo CREME, prejudicando a acurácia do modelo na identificação das duas frequências fundamentais.

Experimento 3

No terceiro e último experimento com o violão isolado, foi analisado o desempenho do modelo CREME ao distinguir duas notas separadas pela menor distância musical possível, isto é, um semitom. Nesse caso, o procedimento consistiu em gerar pares de notas consecutivas ao longo do braço do instrumento, iniciando pela região mais grave na sexta corda (primeira casa) até a região mais aguda na primeira corda (décima segunda casa). Dessa forma, foram construídos arquivos de áudio contendo pares como E2/F2, F2/F#2, F#2/G2, G2/G#2, e assim sucessivamente, sempre

mantendo o intervalo fixo de um semitom entre as notas.

Nome do arquivo	Intervalo	Casa	RPA
1-guitar-e2-f2.wav	E2/F2	1	0,4209
2-guitar-f2-fs2.wav	F2/F#2	2	0,4224
3-guitar-fs2-g2.wav	F#2/G2	3	0,4410
4-guitar-g2-gs2.wav	G2/G#2	4	0,4339
5-guitar-gs2-a2.wav	G#2/A2	5	0,4540
6-guitar-a2-as2.wav	A2/A#2	6	0,4425
7-guitar-as2-b2.wav	A#2/B2	7	0,4410
8-guitar-b2-c3.wav	B2/C3	8	0,4813
9-guitar-c3-cs3.wav	C3/C#3	9	0,4525
10-guitar-cs3-d3.wav	C#3/D3	10	0,4454
11-guitar-d3-ds3.wav	D3/D#3	11	0,4683
12-guitar-ds3-e3.wav	D#3/E3	12	0,4798

Tabela 4.5: RPA das estimativas do CREME: violão/sexta corda/intervalo fixo.

Para simplificar o método e, ao mesmo tempo, garantir representatividade, o experimento foi restrito às duas extremidades do instrumento: a sexta corda (região grave) e a primeira corda (região aguda). Essa escolha foi suficiente para analisar o comportamento do modelo tanto no início quanto no fim da escala do violão, permitindo inferir sobre o desempenho em toda a extensão do braço. Os resultados obtidos estão apresentados nas Tabelas 4.5 e 4.6. A Figura 4.22 apresenta a comparação entre os resultados que mostram a melhoria do RPA para a faixa de notas mais agudas.

Nome do arquivo	Intervalo	Casa	RPA
1-guitar-e4-f4.wav	E4/F4	1	0,5000
2-guitar-f4-fs4.wav	F4/F#4	2	0,4803
3-guitar-fs4-g4.wav	F#4/G4	3	0,4803
4-guitar-g4-gs4.wav	G4/G#4	4	0,5000
5-guitar-gs4-a4.wav	G#4/A4	5	0,5784
6-guitar-a4-as4.wav	A4/A#4	6	0,4411
7-guitar-as4-b4.wav	A#4/B4	7	0,5882
8-guitar-b4-c5.wav	B4/C5	8	0,4901
9-guitar-c5-cs5.wav	C5/C#5	9	0,5686
10-guitar-cs5-d5.wav	C#5/D5	10	0,5000
11-guitar-d5-ds5.wav	D5/D#5	11	0,5000
12-guitar-ds5-e5.wav	D#5/E5	12	0,5098

Tabela 4.6: RPA das estimativas do CREME: violão/sexta corda/intervalo fixo.

A análise mostra uma tendência de melhoria gradual na capacidade de identificação à medida que o intervalo de semitom é testado em regiões mais agudas.

Contudo, esse progresso ocorre de forma lenta. Mesmo em situações teoricamente mais favoráveis, como na primeira corda (região aguda), o modelo apresentou apenas um desempenho modesto, com valores de RPA próximos de 0,5. Esse resultado reforça as limitações do CREME quando exposto a cenários que envolvem combinações artificiais de notas com mínima distância, sem a correlação harmônica presente nos contextos musicais utilizados em seu treinamento.

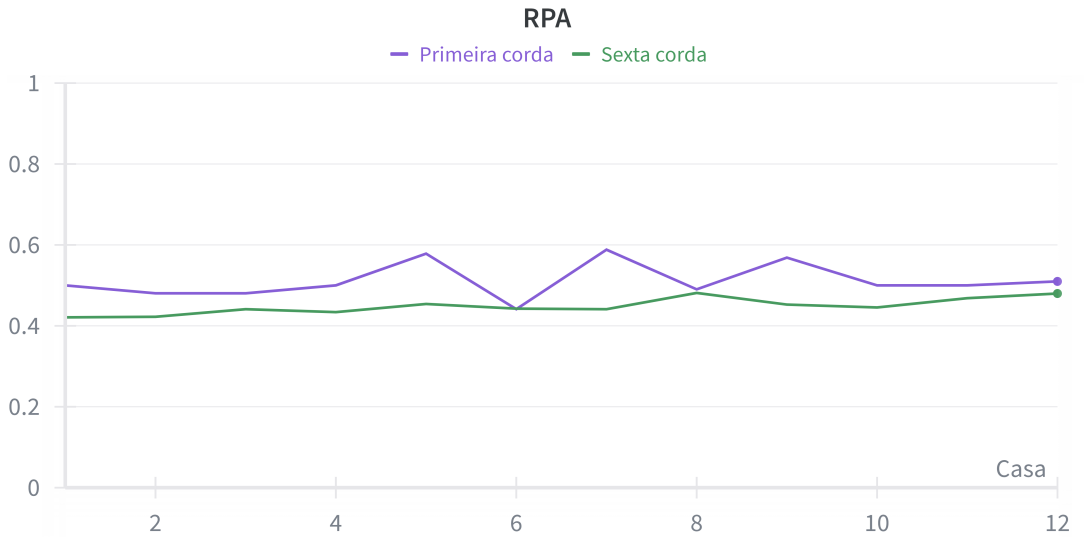

Figura 4.22: Comparação RPA para intervalo fixo na primeira e sexta corda do violão.

Clarinete

Ainda com o objetivo de avaliar o desempenho do modelo CREME em relação ao limite de resolução na distinção de duas notas próximas, foram utilizadas as gravações de clarinete da base RWC. Essa base disponibiliza todas as notas isoladas do instrumento, abrangendo sua extensão completa, desde a nota mais grave D3 até a mais aguda em F6.

De forma semelhante ao procedimento adotado com o violão, foi utilizado o software Audacity para realizar a combinação artificial de notas em pares. Nesse processo, a nota mais grave (D3) foi fixada como referência, sendo progressivamente mixada com notas vizinhas a cada intervalo de semitom. Assim, foram gerados arquivos de áudio contendo pares como D3/D#3, D3/E3, D3/F3, até completar uma oitava e chegar à combinação D3/D4.

Nome do arquivo	Intervalo	Semitons	RPA
1-clarinet-d3-ds3.wav	D3/D#3	1	0,4955
2-clarinet-d3-e3.wav	D3/E3	2	0,7477
3-clarinet-d3-f3.wav	D3/F3	3	0,7544
4-clarinet-d3-fs3.wav	D3/F#3	4	0,9910
5-clarinet-d3-g3.wav	D3/G3	5	0,9732
6-clarinet-d3-gs3.wav	D3/G#3	6	0,4910
7-clarinet-d3-a3.wav	D3/A3	7	0,9330
8-clarinet-d3-as3.wav	D3/A#3	8	0,9821
9-clarinet-d3-b3.wav	D3/B3	9	0,9977
10-clarinet-d3-c4.wav	D3/C4	10	0,9955
11-clarinet-d3-cs4.wav	D3/C#4	11	0,9598
12-clarinet-d3-d4.wav	D3/D4	12	0,9888

Tabela 4.7: RPA das estimativas do CREME: clarinete/intervalo crescente.

Os resultados obtidos estão apresentados na Tabela 4.7. Diferentemente do observado nos testes com o violão, o comportamento no caso do clarinete mostrou-se mais alinhado a uma hipótese intuitiva: à medida que a distância entre as notas aumenta, o modelo apresenta maior precisão, conseguindo distinguir de forma consistente as duas frequências. Ainda assim, em alguns casos foi possível observar dificuldades associadas a combinações dissonantes. Por exemplo, para o intervalo mínimo de um semitom (D3/D#3), o valor de RPA foi de apenas 0,49, desempenho baixo que pode ser explicado pela sobreposição de harmônicos pouco correlacionados. No entanto, já no intervalo seguinte de dois semitons (D3/E3), o RPA atingiu 0,74, e manteve-se elevado com o aumento progressivo da distância, chegando a 0,99 em diversas combinações. Uma exceção foi observada no par D3/G#3, onde o valor voltou a cair para 0,49, evidenciando novamente a dificuldade do modelo em lidar com dissonâncias.

Esses resultados contrastam com os experimentos realizados no violão e reforçam a influência das características acústicas próprias de cada instrumento no desempenho do CREME. No caso do violão, a produção sonora envolve não apenas a vibração da corda excitada, mas também fenômenos adicionais, como a ressonância do corpo do instrumento e a vibração espontânea de cordas próximas em frequências correlatas, que introduzem harmônicos suplementares e tornam o espectro resultante mais complexo. Essa complexidade adicional parece dificultar a separação das frequências fundamentais pelo modelo. Já no caso do clarinete, a produção do som ocorre a partir da vibração controlada da palheta, que gera um espectro mais estável

e com menor interferência estrutural. Essa regularidade do instrumento de sopro explica o desempenho superior do modelo CREME, que conseguiu alcançar valores de RPA mais altos e consistentes, mesmo em intervalos próximos.

Violão e Clarinete

O último experimento realizado com a base de dados RWC envolveu a combinação de notas de violão e clarinete, mantendo o objetivo de investigar os limites do modelo CREME na distinção de duas notas próximas. Para este estudo, foram consideradas as regiões da escala do braço do violão que coincidiam com o intervalo de notas do clarinete, percorrendo todas as cordas do violão e combinando cada nota com as do clarinete, sempre mantendo a distância mínima de um semitom entre as notas. Dessa forma, foi possível avaliar o desempenho do CREME ao lidar com combinações de notas próximas entre dois instrumentos diferentes.

Os resultados, apresentados na Tabela 4.8, revelam comportamento semelhante ao observado nos testes com o violão isolado. O modelo apresentou dificuldade para distinguir as notas, refletida em valores de RPA próximos de 0,5. Embora o o experimento com clarinete isolado tenha apresentado desempenho elevado, sua combinação com o violão resultou em baixo desempenho. Essa limitação é a mesma vista no caso do violão isolado, atribuída a fatores acústicos específicos do violão, como reverberações do corpo do instrumento, vibrações espontâneas de cordas adjacentes e outros elementos que enriquecem o espectro sonoro, aumentando a complexidade do sinal e dificultando a identificação precisa das frequências fundamentais pelo modelo.

4.5 Discussão dos Resultados

Nesta Seção, apresentam-se as reflexões críticas a respeito do treinamento do modelo CREME e da avaliação de seu desempenho nos testes realizados. Busca-se discutir não apenas os resultados obtidos, mas também os fatores que contribuíram para o desempenho observado, evidenciando os principais pontos fortes e limitações do modelo. Além disso, são indicadas possíveis direções para aprimoramentos futuros do modelo e estabelecer bases para sua evolução em cenários mais complexos.

Nome do arquivo	Intervalo	RPA
1-guitarclarinet-cs3-d3.wav	C#3/D3	0,5273
2-guitarclarinet-d3-ds3.wav	D3/D#3	0,5024
3-guitarclarinet-ds3-e3.wav	D#3/E3	0,5024
4–guitarclarinet-e3-f3.wav	E3/F3	0,5597
5-guitarclarinet-f3-fs3.wav	F3/F#3	0,5447
6-guitarclarinet-fs3-g3.wav	F#3/G3	0,5373
7-guitarclarinet-g3-gs3.wav	G3/G#3	0,5597
8-guitarclarinet-gs3-a3.wav	G#3/A3	0,5497
9-guitarclarinet-a3-as3.wav	A3/A#3	0,5373
10-guitarclarinet-as3-b3.wav	A#3/B3	0,4975
11-guitarclarinet-b3-c4.wav	B3/C4	0,5099
12-guitarclarinet-c4-cs4.wav	C4/C#4	0,5049
13-guitarclarinet-cs4-d4.wav	C#4/D4	0,5124
14-guitarclarinet-d4-ds4.wav	D4/D#4	0,5000
15-guitarclarinet-ds4-e4.wav	D#4/E4	0,5024
16-guitarclarinet-e4-f4.wav	E4/F4	0,5000
17-guitarclarinet-f4-fs4.wav	F4/F#4	0,4975
18-guitarclarinet-fs4-g4.wav	F#4/G4	0,5000
19-guitarclarinet-g4-gs4.wav	G4/G#4	0,4975
20-guitarclarinet-gs4-a4.wav	G#4/A4	0,5000
21-guitarclarinet-a4-as4.wav	A4/A#4	0,5024
22-guitarclarinet-as4-b4.wav	A#4/B4	0,5124
23-guitarclarinet-b4-c5.wav	B4/C5	0,5223
24-guitarclarinet-c5-cs5.wav	C5/C#5	0,5024
25-guitarclarinet-cs5-d5.wav	C#5/D5	0,5000
26-guitarclarinet-d5-ds5.wav	D5/D#5	0,4975
27-guitarclarinet-ds5-e5.wav	D#5/E5	0,5248
28-guitarclarinet-e5-f5.wav	E5/F5	0,5273

Tabela 4.8: RPA das estimativas do CREME: violão e clarinete/intervalo fixo.

4.5.1 Análise dos pontos fortes e fracos do modelo CREME

Sobre o treinamento do modelo

Um dos principais pontos fortes observados no treinamento do modelo CREME está na construção da base de dados utilizada. O processo de geração de combinações polifônicas preservou a coerência harmônica entre os sinais, uma característica que aproximou o treinamento de situações reais e contribuiu para a estabilidade do modelo durante a fase de aprendizado. Além disso, a estratégia de divisão em grupos com alternância cíclica garantiu maior diversidade e exposição do modelo ao conjunto de dados, reduzindo o risco de enviesamento para subconjuntos específicos.

Outro aspecto positivo foi a adoção de técnicas de aumento de dados, como a

inserção de ruído em diferentes intensidades, deslocamento de *pitch* e randomização de *frames*, recursos que favoreceram a capacidade de generalização do modelo e mitigaram efeitos de sobreajuste identificados nos primeiros testes.

Um ponto de fragilidade decorre do uso predominante de sinais harmonicamente consistentes, o que pode ter levado o modelo a apresentar dificuldades quando exposto a cenários mais dissonantes, como observado nos testes posteriores com combinações artificiais de notas.

Sobre os dados de validação e indicadores de erro

No que se refere à avaliação experimental com dados de validação, o CREME demonstrou desempenho consistente e robusto, alcançando valores de RPA elevados, sempre acima de 0,84 e com média global próxima de 0,93. Esse resultado reforça a capacidade do modelo em lidar com sinais contendo múltiplas frequências, preservando um nível de precisão comparável ao estado da arte em tarefas monofônicas. Um ponto forte a ser destacado é a estabilidade dos resultados, uma vez que a baixa variação entre os valores máximos e mínimos sugere que o modelo conseguiu generalizar de maneira uniforme para diferentes combinações de sinais dentro da base de validação.

A análise dos indicadores de erro complementa essa interpretação, oferecendo uma visão mais detalhada das limitações do CREME. Entre os três indicadores, a fragilidade mais evidente concentrou-se no erro por falta, isto é, na incapacidade do modelo de detectar todas as frequências efetivamente presentes em um mesmo *frame*. Esse comportamento indica que, embora o modelo consiga reconhecer múltiplos *pitches*, ainda apresenta tendência a subestimar a quantidade de fontes sonoras ativas em contextos polifônicos. Em contrapartida, os valores de alarme falso e substituição permaneceram relativamente baixos, o que demonstra que quando o modelo prediz uma frequência, ela tende a ser consistente com a realidade, e não um artefato espúrio. Esse ponto representa uma vantagem importante, pois reduz o risco de o CREME introduzir frequências inexistentes, o que seria mais prejudicial em aplicações práticas, como transcrição automática ou análise musical.

Testes com base monofônica

Outro aspecto relevante pôde ser observado nos experimentos com a base monofônica de flauta. Os resultados mostraram que a adaptação do modelo para lidar com múltiplas frequências não comprometeu sua precisão em cenários mais simples. De fato, o CREME manteve desempenho semelhante ao CREPE, com diferenças marginais de RPA, o que reforça sua flexibilidade e abrangência. Esse achado constitui um dos pontos fortes mais significativos do trabalho, uma vez que evidencia que a extensão para múltiplos *pitches* não implica perda de desempenho nos casos monofônicos.

Testes com base polifônica

A avaliação com a base Bach10-mf0-synth reforçou a evidência de que o CREME apresenta desempenho sólido em cenários polifônicos controlados. O fato de o modelo alcançar valores de RPA superiores a 0,90 na maior parte dos arquivos testados mostra não apenas sua precisão, mas também sua capacidade de generalização para conjuntos de dados que não fizeram parte do treinamento. Um ponto forte particularmente relevante é a consistência dos resultados mesmo em materiais complexos como peças clássicas de Bach, que apresentam sobreposição harmônica entre instrumentos de timbres distintos. Esse comportamento sugere que a arquitetura do CREME, aliada ao processo de treinamento baseado em combinações de *stems* harmonicamente coerentes, consegue capturar padrões estruturais que extrapolam os limites da base original de treinamento.

Por outro lado, mesmo com a elevada precisão média, os resultados também indicam que o modelo ainda pode sofrer limitações em contextos de polifonia mais densa, onde múltiplos instrumentos interagem em regiões espectrais próximas. A escolha metodológica de restringir os testes a pares de instrumentos permite um diagnóstico claro do desempenho, mas não cobre situações mais complexas que podem surgir em práticas musicais reais. Assim, embora a robustez frente a dois *pitches* simultâneos seja um ponto forte, a extrapolação para cenários com três ou mais frequências simultâneas permanece não explorada nesta etapa de avaliação.

Experimentos com violão

O primeiro experimento com violão destacou uma das limitações mais evidentes do CREME. Ao lidar com notas graves, especialmente em pares próximos como E2/F2 ou E2/F#2, o modelo apresentou valores de RPA frequentemente inferiores a 0,5, mesmo em situações em que a distância entre as frequências já seria teoricamente suficiente para permitir distinção confiável. Esse comportamento revela um ponto fraco importante: a dificuldade do modelo em lidar com sinais complexos e ricos em harmônicos, como os produzidos por instrumentos de corda.

Por outro lado, esse resultado também evidencia um ponto forte metodológico da abordagem adotada, na medida em que demonstra que o CREME não foi superajustado a um único tipo de sinal, mas sim que seu desempenho depende das propriedades acústicas intrínsecas ao material testado. A queda de desempenho nas regiões graves do violão aponta, portanto, para a necessidade de ampliar a diversidade do treinamento, incorporando dados que incluam exemplos de interferências não harmônicas, típicas de instrumentos de cordas, de modo a tornar o modelo mais robusto a tais condições. Assim, embora os valores de RPA baixos possam ser interpretados como uma limitação do modelo, eles também fornecem pistas valiosas sobre as direções em que futuras melhorias devem se concentrar.

O segundo experimento com violão, voltado para a região aguda, evidencia uma tendência semelhante ao primeiro, reforçando algumas limitações do CREME e confirmando certos padrões de comportamento do modelo. Apesar de as notas estarem naturalmente mais distantes em frequência, os valores de RPA permanecem, em grande parte dos casos, abaixo de 0,5, indicando que a rede ainda enfrenta dificuldades ao lidar com combinações artificiais de notas sem correlação harmônica. Essa observação confirma o ponto fraco do modelo identificado no experimento anterior: o treinamento com bases harmônicas tende a otimizar o desempenho para combinações consonantes, dificultando a generalização para pares dissonantes ou não naturais.

Em termos de pontos fortes, o experimento demonstra que o modelo mantém consistência no padrão de comportamento entre diferentes regiões do braço do violão, permitindo identificar claramente quais fatores externos, como a falta de consonância e os efeitos acústicos do instrumento, são responsáveis pela queda de desempenho.

Esses *insights* fornecem subsídios importantes para orientar estratégias futuras de aprimoramento, como a inclusão de exemplos de pares dissonantes e a diversificação de gravações de instrumentos de corda durante o treinamento.

O terceiro experimento, no qual se avaliou a capacidade do CREME de distinguir notas separadas pelo menor intervalo musical possível, reforça os padrões observados nos experimentos anteriores e evidencia de maneira mais clara os limites do modelo. Apesar de analisar as extremidades do braço do violão e manter constante o intervalo de um semitom entre as notas, os resultados indicam que o modelo ainda apresenta desempenho modesto, com valores de RPA próximos a 0,5, mesmo nas regiões agudas, teoricamente mais favoráveis à diferenciação de frequências.

Esse comportamento confirma um ponto fraco recorrente do CREME: a rede é otimizada para lidar com combinações harmônicas naturais, como as presentes nas bases de treinamento, mas apresenta dificuldades diante de combinações artificiais ou dissonantes, especialmente quando a distância entre as notas é mínima. Além disso, fatores acústicos do violão, como ressonâncias e reverberações do corpo do instrumento, interferem na clareza das frequências fundamentais, contribuindo para a queda de desempenho observada nos pares de notas mais próximos. Por outro lado, o experimento evidencia uma tendência de melhoria gradual na identificação conforme os intervalos se deslocam para regiões agudas, indicando que o modelo consegue explorar a separação natural de frequência quando disponível.

Experimento com clarinete

O experimento com o clarinete evidencia aspectos importantes sobre os pontos fortes do modelo CREME em comparação com instrumentos de corda. Ao longo da escala do instrumento, o modelo apresentou desempenho consistente e coerente com a expectativa teórica: à medida que a distância entre as notas aumenta, a capacidade de distinção do CREME melhora, refletindo-se em valores de RPA elevados. Esse comportamento confirma a força do modelo em contextos polifônicos quando os sinais apresentam harmônicos bem definidos e pouco interferidos por ressonâncias adicionais.

Mesmo assim, os resultados indicam que o CREME ainda apresenta limitações diante de combinações dissonantes. O desempenho reduzido observado para

intervalos mínimos, como D3/D#3 (RPA igual a 0,49), e em casos específicos de maior dissonância, como D3/G#3, sugere que a rede encontra dificuldades quando os harmônicos das frequências próximas não possuem correlação natural, mesmo em instrumentos de sopro com espectro relativamente limpo.

Em contraste com os testes realizados com o violão, os resultados do clarinete reforçam a hipótese de que as propriedades acústicas do instrumento exercem impacto direto sobre a capacidade de estimativa de múltiplos *pitches* do CREME. Enquanto o violão apresenta interferências significativas devido à ressonância do corpo do instrumento e à vibração de cordas adjacentes, o clarinete produz um espectro mais estável e previsível, permitindo que o modelo consiga identificar com maior precisão as frequências fundamentais. Dessa forma, o experimento demonstra que o CREME é mais robusto em contextos acústicos menos complexos e sugere que ajustes adicionais no treinamento poderiam ajudar a superar as dificuldades encontradas em instrumentos com espectros mais ricos, como o violão.

4.5.2 Sugestões de melhorias futuras

A análise crítica realizada nos testes experimentais indica algumas direções claras para aprimorar o desempenho do modelo CREME em cenários polifônicos e complexos. Primeiramente, em relação à detecção de múltiplas frequências, um ponto a ser explorado é a redução do erro por falta, que mostrou-se o indicador mais crítico entre os avaliados. A tendência do modelo em subestimar a quantidade de fontes sonoras ativas sugere que a inclusão de dados adicionais com maior densidade polifônica e maior diversidade espectral poderia auxiliar a rede a aprender padrões mais robustos de ativação simultânea de múltiplas frequências.

No contexto de instrumentos de corda, como evidenciado nos experimentos com violão, os baixos valores de RPA decorrem de interferências acústicas adicionais, como reverberações do corpo do instrumento, vibrações espontâneas de cordas próximas e harmônicos suplementares. Para mitigar esse efeito, uma sugestão é ampliar do conjunto de treinamento com exemplos que reproduzam essas condições naturais de interferência, incluindo combinações artificiais de notas dissonantes ou pares de instrumentos cujas frequências se sobrepõem parcialmente. Essa abordagem permitiria ao modelo generalizar melhor para sinais não harmônicos e reduzir

a sensibilidade a artefatos típicos da produção sonora de instrumentos de corda.

Além disso, embora o CREME tenha apresentado desempenho sólido em pares de instrumentos isolados, a avaliação combinando violão e clarinete evidenciou limitações quando instrumentos com características acústicas distintas coexistem. Futuras melhorias poderiam incluir o treinamento com conjuntos de dados que combinem diferentes famílias instrumentais, proporcionando ao modelo experiência prévia com contextos de interferência espectral heterogênea. Essa estratégia ajudaria a aumentar a robustez do CREME para situações de polifonia real, onde múltiplos timbres interagem de forma complexa.

Por fim, considerando que o treinamento atual foi realizado prioritariamente com dados harmonicamente coerentes, outra direção relevante é incorporar técnicas de aumento de dados que gerem combinações menos naturais, com pares dissonantes e variações espectrais artificiais, mantendo a correspondência de anotações precisas. Essa abordagem não apenas aumentaria a diversidade do conjunto de treinamento, mas também permitiria ao modelo aprender a lidar com cenários adversos, melhorando sua capacidade de generalização e a acurácia em sinais que não seguem padrões harmônicos convencionais.

Em resumo, as sugestões de melhorias futuras concentram-se na ampliação e diversificação dos dados de treinamento, na inclusão de combinações dissonantes e na exposição do modelo a interferências acústicas típicas de instrumentos complexos, de modo a fortalecer sua capacidade de estimativa de múltiplos *pitches* em contextos polifônicos mais desafiadores.

Capítulo 5

Conclusão

5.1 Considerações Finais da Dissertação

5.1.1 Recapitulação dos principais resultados obtidos

Nesta dissertação, foi apresentado o desenvolvimento e a avaliação do modelo CREME, uma extensão do CREPE, voltado para a estimativa de múltiplos *pitches* simultâneos em sinais musicais. O objetivo principal consistiu em generalizar a abordagem de detecção monofônica para cenários polifônicos, mantendo a precisão em contextos mais simples e expandindo a capacidade de lidar com múltiplas frequências.

Os resultados demonstraram que o CREME alcança desempenho robusto em diferentes situações. Em conjuntos de validação com múltiplas frequências, o modelo apresentou valores elevados de RPA, com média global próxima de 0,93, evidenciando sua capacidade de identificar corretamente *pitches* simultâneos. A análise por indicadores de erro mostrou que a principal limitação do modelo está na subestimação de frequências presentes, enquanto a incidência de alarmes falsos e erros de substituição permaneceu baixa.

Nos testes com bases polifônicas controladas, como o *Bach10-mf0-synth*, o CREME manteve RPA superiores a 0,90, mesmo em peças com sobreposição harmônica de instrumentos distintos. Os experimentos com violão revelaram limitações ao lidar com sinais complexos de cordas, principalmente em combinações artificiais sem correlação harmônica, com RPA em torno de 0,5. Já o clarinete isolado apre-

sentou desempenho mais próximo do ideal, indicando que a estabilidade espectral de instrumentos de sopro favorece a estimativa precisa de múltiplos *pitches*. Experimentos combinando violão e clarinete demonstraram que o modelo ainda enfrenta dificuldades quando instrumentos com características acústicas distintas coexistem.

5.1.2 Validação do objetivo de generalizar o CREPE

Os resultados obtidos confirmam que o objetivo da dissertação de estender o CREPE para cenários polifônicos foi alcançado de maneira consistente. O CREME mostrou capacidade de generalizar bem em contextos harmônicos coerentes com dois *pitches* simultâneos, mantendo desempenho elevado mesmo em combinações de instrumentos e frequências que não haviam sido vistas durante o treinamento. Além disso, os experimentos permitiram identificar com clareza os fatores que influenciam a acurácia em sinais mais complexos, como a presença de harmônicos adicionais, ressonâncias de instrumentos de corda e combinações dissonantes, evidenciando os limites atuais do modelo. Essa capacidade de generalização é especialmente relevante em cenários com múltiplas fontes sonoras simultâneas, onde a identificação precisa de *pitches* é desafiadora devido à sobreposição de frequências e à interação entre harmônicos.

5.1.3 Aplicações práticas e estudos futuros

O CREME apresenta potencial para diversas aplicações práticas, incluindo transcrição musical automática, análise de performances polifônicas e reconhecimento de acordes. Para estudos futuros, uma estratégia promissora seria expandir o treinamento do modelo para incluir mais de duas frequências simultâneas, aumentando o alcance de *pitches* que o CREME pode estimar. O código atualmente implementado para o CREME [67] já suporta bases de dados com múltiplas frequências por *frame*, facilitando essa expansão. Além disso, estudos futuros podem explorar a ampliação e diversificação dos conjuntos de treinamento, incluindo combinações dissonantes, exposição a interferências acústicas típicas de instrumentos complexos e avaliação em cenários polifônicos mais densos, com três ou mais *pitches* simultâneos. Tais abordagens têm o potencial de aumentar a robustez e a precisão do modelo, aproximando seu desempenho do limite teórico de resolução de frequência

em sinais polifônicos reais.

Em síntese, o CREME representa um avanço significativo na estimativa automática de múltiplos *pitches*, fornecendo uma base sólida para pesquisas futuras e aplicações práticas em música e processamento de áudio.

Referências Bibliográficas

[1] LABS, B., "Audrey: Early Speech Recognition System", `https://en.wikipedia.org/wiki/Speech_processing`, 1952, Primeiro sistema de reconhecimento de fala desenvolvido pela Bell Labs, capaz de reconhecer dígitos falados.

[2] MOORER, J. A., "Signal Processing for Computer Music", *Journal of the Audio Engineering Society*, v. 39, n. 7/8, pp. 541–553, 1991, Revisão fundamental do estado da arte em processamento musical no início da década de 1990.

[3] MULLER, M., *Fundamentals of Music Processing*. 2nd ed. Springer: Gewerbestrasse 11, 6330 Cham, Switzerland, 2021.

[4] MCCALLUM, R., O'NEILL, P., "How Music Recommender Systems Impact the Music Industry", *First Monday*, v. 26, n. 3, 2021, Discute como sistemas de recomendação moldam o consumo de música, como Spotify e Apple Music.

[5] CHOPRA, A., ROY, A., HERREMANS, D., "MIRFLEX: Music Information Retrieval Feature Library for Extraction", *arXiv preprint arXiv:2411.00469*, 2024, Biblioteca modular para extração de características musicais usadas em recomendação, classificação, etc.

[6] ZIEMER, T., KIATTIPADUNGKUL, P., KARUCHIT, T., "Novel Recording Studio Features for Music Information Retrieval", *arXiv preprint arXiv:2101.10201*, 2021, Explora recursos de estúdio para apoiar produção musical e classificação de gênero com MIR.

[7] SCHOOL, H. B., "Spotify's Machine Learning Strategy", Harvard Business School Digital Initiative, 2021, Descrição das estratégias de recomendação do Spotify usando filtragem colaborativa, NLP e áudio.

[8] GROUP, F. M., "How Machine Learning is Transforming Music Marketing and Trends Analysis", Flourish & Prosper, 2023, Aborda o uso de machine learning em marketing musical, previsão de tendências e personalização publicitária.

[9] PONS, J., "Estimating pitch in polyphonic music", *arXiv preprint arXiv:1901.05079*, 2019, Apresenta técnicas de estimativa de pitch em música polifônica, fundamentais para transcrição musical automatizada.

[10] MAUCH, T., "Computer-aided melody note transcription using the CREPE model", *Proceedings of the TENOR Conference*, 2015, Demonstra a aplicação do modelo CREPE para transcrição de notas melódicas, destacando sua eficácia na estimativa de F0.

[11] SINGH, S., WANG, R., QIU, Y., "DEEPF0: End-To-End Fundamental Frequency Estimation for Music and Speech Signals", *arXiv preprint arXiv:2102.06306*, 2021, Apresenta o modelo DeepF0, que utiliza convoluções dilatadas para estimativa precisa de F0 em sinais musicais e de fala.

[12] BERNARD, D., "Pitch estimation in monophonic and polyphonic audio signals", *Bachelor Final Project*, 2022, Apresenta uma visão geral dos métodos comumente utilizados na estimativa de pitch monofônico e polifônico, destacando as dificuldades associadas à música polifônica.

[13] KIM, J. W., SALAMON, J., LI, P., et al., "CREPE: A Convolutional Representation for Pitch Estimation". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161–165, 2018.

[14] HARTMANN, W. M., *Signals, Sound, and Sensation*. American Institute of Physics: Melville, NY, 1998.

[15] PLACK, C. J., OXENHAM, A. J., *The Psychophysics of Pitch*. Springer, 2005.

[16] MOORE, B. C. J., *An Introduction to the Psychology of Hearing*. 6th ed. Brill, 2012.

[17] PLACK, C. J., OXENHAM, A. J., "Pitch perception", *Current Biology*, v. 18, n. 16, pp. R855–R858, 2008.

[18] TERHARDT, E., "Pitch, consonance, and harmony", *Journal of the Acoustical Society of America*, v. 55, n. 5, pp. 1061–1069, 1974.

[19] MCADAMS, S., "Perspectives on the contribution of timbre to musical structure", *Computer Music Journal*, v. 23, n. 3, pp. 85–102, 1999.

[20] GREY, J. M., "Multidimensional perceptual scaling of musical timbres", *Journal of the Acoustical Society of America*, v. 61, n. 5, pp. 1270–1277, 1977.

[21] SETHARES, W. A., *Tuning, Timbre, Spectrum, Scale*. 2nd ed. Springer, 2005.

[22] LICKLIDER, J. C. R., "On the Frequency Analysis of Sounds by the Ear", *Journal of the Acoustical Society of America*, v. 27, n. 1, pp. 123–128, 1954.

[23] SCHOUTEN, J. F., RITSMA, R. J., CARDOZO, B. L., "Pitch of the Residue", *Journal of the Acoustical Society of America*, v. 34, n. 8, pp. 1418–1424, 1962.

[24] HARTMANN, W. M., "Pitch, Periodicity, and Auditory Organization", *The Journal of the Acoustical Society of America*, v. 99, n. 6, pp. 3584–3596, 1996, Este estudo explora como o sistema auditivo humano percebe o *pitch* a partir da série harmônica, mesmo na ausência da frequência fundamental.

[25] CEDOLIN, L., DELGUTTE, B., "Pitch of Complex Tones: Rate-Place and Interspike Interval Representations in the Auditory Nerve", *Journal of Neurophysiology*, v. 94, n. 1, pp. 347–362, 2005.

[26] HOUTSMA, A. J. M., SMURZYNSKI, J., "Pitch identification and discrimination for complex tones with many harmonics", *The Journal of the Acoustical Society of America*, v. 87, n. 1, pp. 304–310, 1990.

[27] GUEST, D. R., OXENHAM, A. J., "The role of pitch and harmonic cancellation when listening to speech in harmonic background sounds", *The Journal of the Acoustical Society of America*, v. 145, n. 4, pp. 3011, 2019.

[28] WHITEFORD, K. L., OXENHAM, A. J., "Learning for pitch and melody discrimination in congenital amusia", *Cortex*, v. 103, pp. 164–178, 2018.

[29] DELGUTTE, B., "Place and temporal theories of pitch", *Scientific American*, v. 275, n. 5, pp. 49–57, 1996.

[30] DOUBT, T. J., RICHARDS, V. M., "Pure-tone anomalies. I. Pitch-intensity effects and diplacusis in normal ears", *The Journal of the Acoustical Society of America*, v. 97, n. 5, pp. 2978–2988, 1995.

[31] GUEST, D. R., OXENHAM, A. J., "Harmonicity helps hearing in noise: detection and discrimination effects for harmonic vs. inharmonic tones", *Biology*, v. 12, n. 12, pp. 1522, 2022.

[32] KAWASE, T., DELGUTTE, B., LIBERMAN, M. C., "Neural representation of sound amplitude in the auditory cortex: effects of noise masking", *Journal of Neurophysiology*, v. 70, n. 6, pp. 2533–2549, 1993.

[33] ROADS, C., *The Computer Music Tutorial*. MIT Press: Cambridge, MA, 1996.

[34] DEPALLE, P., HÉLIE, T., "Analysis and Synthesis of Quasi-Harmonic Sounds: A Review of Sinusoidal Modeling Techniques", *Acoustics Today*, v. 12, n. 3, pp. 24–33, 2016.

[35] HILDEBRAND, A., "System and Method for Pitch Correction of Vocal Performances", US Patent 5,997,776, 1999, Antares Audio Technologies.

[36] MÜLLER, M., "Fundamentals of Music Processing: Using Python and Jupyter Notebooks", In: *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*, chap. 8, pp. 437–449, Springer, 2015.

[37] OPPENHEIM, A. V., SCHAFER, R. W., BUCK, J. R., *Discrete-time signal processing*. Prentice Hall, 1999.

[38] KLAPURI, A., "Multiple fundamental frequency estimation by harmonicity and spectral smoothness", *IEEE Transactions on Speech and Audio Processing*, v. 11, n. 6, pp. 804–816, 2006.

[39] KAY, S. M., *Fundamentals of Statistical Signal Processing: Estimation Theory*. v. 1. Prentice Hall, 1993.

[40] HARRIS, F. J., "On the use of windows for harmonic analysis with the discrete Fourier transform". In: *Proceedings of the IEEE*, v. 66, n. 1, pp. 51–83, 1978.

[41] RABINER, L. R., GOLD, B., "Theory and application of digital signal processing", *Prentice-Hall, Englewood Cliffs*, v. 15, n. 2, pp. 235–243, 1975.

[42] DUDA, R. O., HART, P. E., "Pattern Classification and Scene Analysis", *A Wiley-Interscience Publication*, pp. 271–272, 1973.

[43] MCLEOD, A., WYVILL, G., "Evaluation of the YIN pitch tracking algorithm across languages and instruments", *The Journal of the Acoustical Society of America*, v. 143, n. 3, pp. EL202–EL208, 2018.

[44] DE CHEVEIGNÉ, A., KAWAHARA, H., "YIN, a fundamental frequency estimator for speech and music", *The Journal of the Acoustical Society of America*, v. 111, n. 4, pp. 1917–1930, 2002.

[45] MAUCH, M., DIXON, S., "pYIN: A fundamental frequency estimator using probabilistic threshold distributions". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659–663, 2014.

[46] RABINER, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, v. 77, n. 2, pp. 257–286, 1989.

[47] VITERBI, A. J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Transactions on Information Theory*, v. 13, n. 2, pp. 260–269, 1967.

[48] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., *Deep Learning*. MIT Press, 2016.

[49] HUMPHREY, E. J., BELLO, J. P., LECUN, Y., "Deep learning for music information retrieval: promises, challenges, and opportunities", *IEEE Signal Processing Magazine*, v. 30, n. 2, pp. 82–86, 2013.

[50] BITTNER, R. M., MCFEE, B., SALAMON, J., et al., "Deep Salience Representations for F0 Tracking in Polyphonic Music". In: *Proceedings of the 18th ISMIR Conference*, 2017.

[51] HSU, C.-L., JANG, R. B., "On the Improvement of Singing Voice Separation for Monaural Recordings Using the MIR-1K Dataset", *IEEE Transactions on Audio, Speech, and Language Processing*, 2009.

[52] DUAN, Z., HAN, Y., PARDO, B., et al., "Multiple Fundamental Frequency Estimation by Modeling Spectral Peaks and Non-Peak Regions", *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[53] GOTO, M., HASHIGUCHI, H., NISHIMURA, T., et al., "RWC Music Database: Popular, Classical and Jazz Music Databases". In: *Proceedings of the 3rd ISMIR Conference*, v. 2, pp. 287–288, 2002.

[54] BITTNER, R. M., SALAMON, J., TIERNEY, M., et al., "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research". In: *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference*, 2014.

[55] SALAMON, J., GOMEZ, E., ELLIS, D. P., "An Analysis/Synthesis Framework for Automatic F0 Annotation of Multitrack Datasets". In: *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference*, 2017.

[56] ENGEL, J., RESNICK, C., ROBERTS, A., et al., "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders", *arXiv preprint arXiv:1704.01279*, 2017.

[57] STURM, B. L., "Classification accuracy is not enough", *Journal of Intelligent Information Systems*, v. 41, n. 3, pp. 371–406, 2013.

[58] SALAMON, J., GOMEZ, E., ELLIS, D. P. W., et al., "Melody extraction from polyphonic music signals: Approaches, applications, and challenges", *IEEE Signal Processing Magazine*, v. 31, n. 2, pp. 118–134, 2014.

[59] RAFFEL, C., MCFEE, B., HUMPHREY, E. J., et al., "mir_eval: A Transparent Implementation of Common MIR Metrics". In: *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014.

[60] ARDAILLON, L., ROEBEL, A., "Fully-Convolutional Network for Pitch Estimation of Speech Signals". In: *Proceedings of Interspeech 2019*, Graz, Austria, September 2019.

[61] BRUM, J. P. B., "Traditional Flute Dataset for Score Alignment", `https://www.kaggle.com/jbraga/traditional-flute-dataset`, 2018.

[62] MCFEE, B., RAFFEL, C., LIANG, D., et al., "librosa: Audio and Music Signal Analysis in Python". In: *Proceedings of the 14th Python in Science Conference*, pp. 18–25, 2015.

[63] CHOLLET, F., OTHERS, "Keras", `https://keras.io`, 2015.

[64] ABADI, M., AGARWAL, A., BARHAM, P., et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", 2015, Software available from tensorflow.org.

[65] KIM, J. W., SALAMON, J., LI, P., et al., "CREPE: A Convolutional Representation for Pitch Estimation", `https://github.com/marl/crepe`, 2018, Acesso em: 11 ago. 2025.

[66] CHOI, J., "Flazy: Functional, lazy-evaluated dataset manipulation library for ML in Python", `https://github.com/jongwook/flazy`, 2025, Acesso em: 28 ago. 2025.

[67] FÁBIO, M., "CREME: Estimador de múltiplos pitches para sinais musicais", `https://github.com/marcus-fabio/creme`, 2025, Acesso em: 27 ago. 2025.