UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

DOUGLAS EDUARDO SLAVIERO

USO DE CARACTERÍSTICAS SIGNIFICATIVAS EM SISTEMA DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICA

CAXIAS DO SUL

2022

DOUGLAS EDUARDO SLAVIERO

USO DE CARACTERÍSTICAS SIGNIFICATIVAS EM SISTEMA DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICA

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Orientador: Prof. Dr. André Gustavo Adami

CAXIAS DO SUL

2022

DOUGLAS EDUARDO SLAVIERO

USO DE CARACTERÍSTICAS SIGNIFICATIVAS EM SISTEMA DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICA

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado em 01/12/2022

BANCA EXAMINADORA

_____

Prof. Dr. André Gustavo Adami
Universidade de Caxias do Sul - UCS

_____

Profa. Dra. Carine Geltrudes Webber
Universidade de Caxias do Sul - UCS

_____

Prof. Dr. Daniel Luis Notari
Universidade de Caxias do Sul - UCS

Ao futuro.

AGRADECIMENTOS

Agradeço à minha família, por sempre me incentivar a seguir o meu próprio caminho. À Sabrina, uma fagulha que sempre me motiva. Ao meu amigo Jonas, sem a ajuda dele, possivelmente não teria concluído. A todos os meus amigos, que tornaram esta jornada mais leve. E ao professor Adami, pela paciência e bom humor para me orientar.

RESUMO

No decorrer dos anos a indústria da música vem se adaptando e, atualmente, está passando por um estágio de transição. A receita, que tinha seu predomínio em vendas de mídias físicas, passou a ser majoritariamente de serviços de *streaming*. Com o advento dos serviços de *streaming*, o modo de consumir e ouvir mídias de áudio se tornou uma experiência além da música. Conteúdos categorizados, gerando recomendações segundo as características e históricos dos usuários, são cada vez mais utilizados. Uma das informações que pode ser utilizada no intuito de categorizar as músicas é a língua. A partir dela é possível explorar mais pontos do seu âmbito, como reconhecimento de locutor e transcrição de letras. Trabalhos de identificação de língua em música, em sua grande maioria, exploram características estáticas do sinal de áudio propostas para o reconhecimento de fala e não o de língua. Visando contornar essa limitação, o objetivo deste trabalho foi avaliar o uso da rede *SincNet* em um modelo *deep learning* para fazer a extração de características significativas do sinal de áudio, para ser feita a identificação de língua em música. Além disso, este trabalho emprega o uso de diferentes técnicas de processamento de sinais para dirimir informações irrelevantes (por exemplo, som instrumental ou plateia) do sinal de música. Assim, o sistema proposto, primeiramente, remove os segmentos onde a voz cantante não ocorre (segmentação) e em seguida separa o sinal da voz do som instrumental (separação de áudio). O sinal de voz é alimentado na rede *deep learning* para extração de características e identificação da língua. O sistema proposto foi avaliado em uma base construída a partir das músicas de um serviço de streaming. Os resultados mostraram que as etapas de pré-processamento, segmentação e separação contribuem significativamente para o desempenho do sistema. Além disso, o sistema proposto obteve desempenho superior de aproximadamente 12% em comparação com sistema utilizando características estáticas e mesmas etapas de pré-processamento.

Palavras-chave: Identificação de Língua, Música, Extração de Características, *Deep Learning*.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

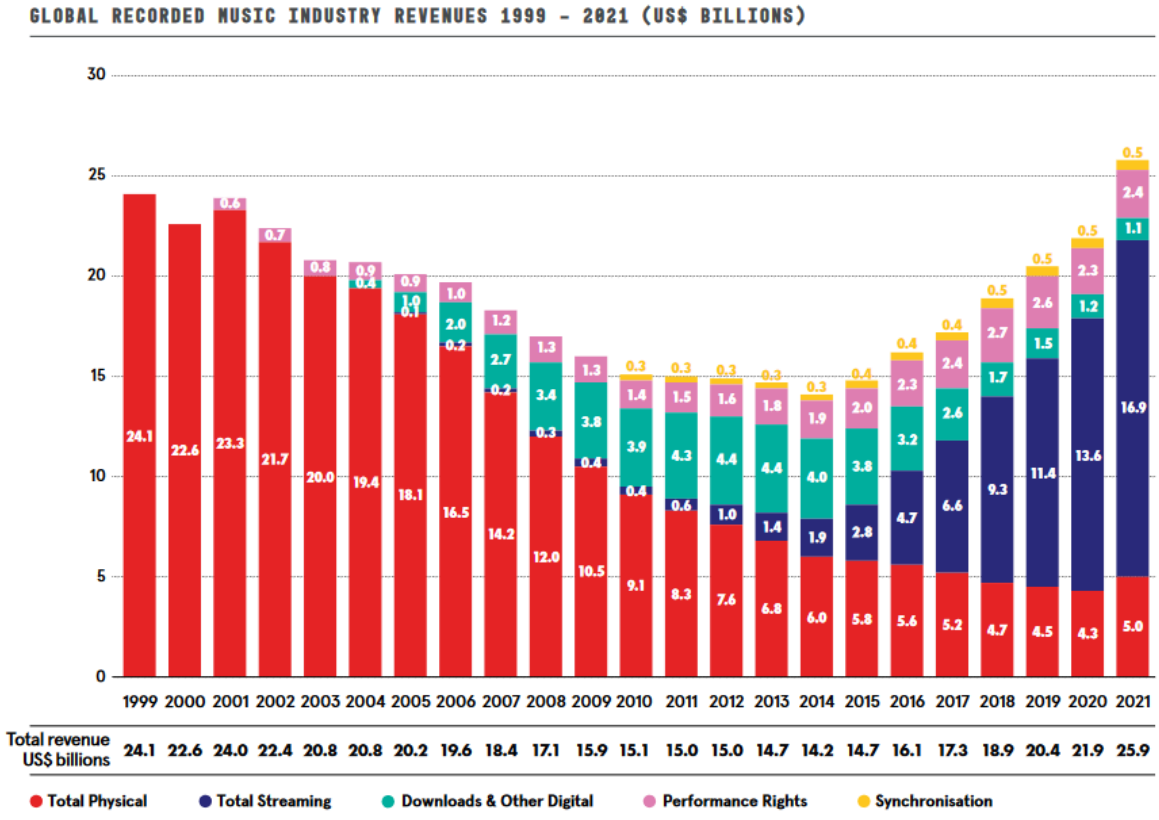
LID *Language Identification*
SLID *Singing Language Identification*
MIR *Music Information Retrieval*
MSA *Music Structure Analysis*
VAD *Voice Activity Detection*
MFCC *Mel-Frequency Cepstral Coefficients*
SDC *Shifted Delta Cepstrum*
PLP *Perceptual Linear Predictive features*
MLP *Multi-Layer Perceptrons*
GMM *Gaussian Mixture Model*
SVM *Support Vector Machines*
LPC *Linear Predictive Coefficiets*
RNA Rede Neural Artificial
DNN *Deep Neural Network*
CNN *Convolutional Neural Network*
TCN *Temporal Convolutional Network*
LSF *Line Spectral Frequency*
RNN *Recurrent Neural Network*
STFT *Short-Term Fourier Transform*
DCT *Discrete Cosine Transform*
DFT *Discrete Fourier Transform*
VP Verdadeiro positivo
VN Verdadeiro negativo
FP Falso positivo
FN Falso negativo
BPTT *Backpropagation Through Time*
RTRL *Real-Time Recurrent Learning*
LSTM *Long Short-Term Memory*
FFNN *Feed-foward Neural Network*
NLP *Natural Language Processing*
ReLU *Rectified Linear Unit*
GPU *Graphics Processing Unit*
DL *Deep Learning*

SUMÁRIO

1 INTRODUÇÃO

A indústria da música está em constante evolução e adaptação. De tecnologias de produção a modelos de negócios, todos os âmbitos vêm sendo constantemente melhorados, trazendo novas formas de se consumir músicas. Atualmente a indústria da música passa por um estágio de transição, conforme mostra a Figura 1. Há duas décadas o predomínio da receita era da venda de mídias físicas, passando a dividir uma fatia do mercado com mídias digitais, até o maior predomínio ser dos serviços de *streaming* no ano 2022. Estes serviços de *streaming*, como Spotify, Deezer e Apple possuíam cerca de 443 milhões de usuários com assinaturas pagas em suas plataformas no ano de 2020, que em conjunto com conteúdos patrocinados, resultam em cerca de 62% de toda a receita da indústria da música (IFPI, 2021).

Figura 1 – Receita anual da indústria da música entre 1999 e 2021.



Fonte: Adaptado de (IFPI, 2022).

Com o crescimento do consumo através de serviços de *streaming*, ferramentas que auxiliem, otimizem e abram novas possibilidades para conteúdos à base de áudios se tornaram mais relevantes (MORRIS; POWERS, 2015). Conteúdos categorizados, gerando recomendações dadas as características e histórico dos usuários, visando reter e manter o usuário engajado na plataforma, são cada vez mais utilizados (MAASØ; SPILKER, 2022). Com o avanço no processamento

de sinais digitais, uso de *machine learning* e grande volume de dados disponíveis, trabalhos que buscam em uma música o reconhecimento de um instrumento, a transcrição das notas musicais, classificação de gênero e reconhecimento do artista foram amplamente explorados ao decorrer do tempo (SCHEDL *et al.*, 2014). Quando o objeto é a voz, uma das primeiras características que a definem é a língua. A língua é um primeiro passo para explorar mais pontos do seu âmbito, como o reconhecimento de locutor (CAMPBELL, 1997) e diarização de locutor (ANGUERA *et al.*, 2012). A identificação de língua a partir de áudio, denominada *Language Identification* (LID), é um tópico amplamente explorado. No entanto, a identificação de língua em voz cantante em música, denominada *Singing Language Identification* (SLID), é mais recente e os trabalhos relacionados não possuem um desempenho conclusivo, comparado aos relacionados a LID.

Os trabalhos de identificação de língua utilizam em sua maioria abordagens tradicionais de *machine learning* e redes neurais *Deep Learning* (DL). Os modelos tradicionais, exploram as características e etapas de forma procedural para chegar num sistema de classificação. Utilizam e exploram características estáticas do sinal de áudio, como o *Mel-Frequency Cepstral Coefficients* (MFCC) (GROENBROEK, 2021), que transforma o sinal de áudio para uma representação que aproxima a frequência de percepção do ouvido humano. Os modelos DL além de classificarem, transformam e extraem características para trazer maior significância aos dados, resultando numa cadeia de aprendizado em prol do objetivo do sistema.

O uso de características estáticas limita o sistema de identificação, dado que as características extraídas não se adaptam ao objetivo do sistema. Visando contornar essa limitação, é possível utilizar métodos que extraiam características significativas para o problema de interesse (RAVANELLI; BENGIO, 2018). Com base nessa premissa, modelos que aprendem numa abordagem DL podem ser aplicados no problema de identificação de língua em música. Resultam em um sistema onde não somente o modelo de classificação é estimado, mas também as características são aprendidas, consequentemente, reduzindo o número de etapas necessárias e simplificando o processo num todo.

1.1 OBJETIVOS

O objetivo deste trabalho é avaliar o uso de uma rede DL para capturar informações relevantes do sinal de áudio de uma música para serem aplicadas no problema de identificação de língua em música.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta o referencial teórico, descreve conceitos sobre o trabalho com dados de áudio, informações que distinguem uma língua de outra e as técnicas mais uti-

lizados para a identificação de língua em modelos LID e SLID. Dentre as técnicas, são analisadas as etapas necessárias, características utilizadas e arquitetura do modelo. Por fim, apresenta e explica o funcionamento de modelos DL.

- O Capítulo 3 apresenta o sistema proposto utilizado para fazer a identificação de língua em música, o sistema *baseline* utilizado para fins de comparação, a base de dados construída, a experimentação aplicada para desenvolver e avaliar o modelo proposto e os resultados obtidos.
- O Capítulo 4 apresenta as conclusões do trabalho.

2 SISTEMAS DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICAS

Este capítulo define o problema e apresenta as abordagens de identificação de língua em músicas. A Seção 2.1 apresenta características que definem e compõem uma música, expandindo para áreas que exploram e desenvolvem trabalhos nesse contexto, buscando extrair e analisar suas características e estrutura. Na Seção 2.2, são apresentados detalhes relevantes da língua, detalhes que influenciam na identificação da mesma na fala humana. Na Seção 2.3, é definido o trabalho de identificação de língua, visando dar um contexto geral e amplo do problema abordado. A Seção 2.4 apresenta o que é o trabalho de identificação de língua em música, dando um contexto e abordagens realizadas pelos trabalhos que buscaram realizar a tarefa, quais seus desempenhos e métodos utilizados. Na Seção 2.5, é contextualizada a estrutura básica de uma rede neural até redes neurais profundas.

2.1 DA MÚSICA À INFORMAÇÃO

Dados de áudio comportam uma variedade de características e informações que estruturam a música. Música é um conjunto de sons organizados e tocados em sequência que podem gerar sentimentos de raiva, angústia, tristeza e felicidade (JUSLIN, 2016). Som são vibrações que viajam através do espaço, é uma onda que se propaga por si própria. Quando as vibrações de um som são regulares e repetitivas, identifica-se a frequência, definida pela quantidade de ciclos de movimentos que ocorrem por segundo (medida representada em Hertz - Hz).

Músicas são compostas por notas, escalas, harmonias, acordes e melodias. Uma nota consiste em um som com frequência e duração constante, onde são normalmente tocadas por instrumentos e pela voz humana. Na maioria das formas de música, notas são tiradas de escalas, estas que são um conjunto fixo de notas do qual são usadas para produzir músicas. Quando um conjunto de diferentes notas são tocadas simultaneamente, forma-se a harmonia. Normalmente a harmonia é descrita como um conjunto de acordes, já que acorde é um grupo específico de notas tocadas juntas. Uma característica muito importante da música é o tempo. Notas tocadas com ritmo e repetição ao decorrer do tempo tornam um conjunto de sons, vibrações em certa frequência, em música (DORRELL, 2005).

Nesse contexto, a Recuperação da Informação Musical, do inglês *Music Information Retrieval* (MIR), é uma disciplina que visa encontrar e extrair informações relevantes da música, como gênero, artista e frequência de batidas. Como descrito por Casey *et al.* (2008) MIR é focada em processamento automatizado para tornar informações sobre músicas mais acessíveis. Beneficia a indústria da música desde profissionais como professores, produtores e artistas que buscam informações específicas, a usuários finais que desejam descobrir novas músicas e consumir de forma personalizada. Sistema de identificação de música (WANG *et al.*, 2003),

recomendações de música com base em dada preferência musical (BALTRUNAS *et al.*, 2011) e monitoramento de batidas em músicas afro-latinas (FUENTES *et al.*, 2019) são alguns exemplos de aplicações na área.

Outra área importante na extração de informações de música é a Análise de Estrutura Musical, do inglês *Music Structure Analysis* (MSA). Ela trabalha com a premissa de que todas as músicas podem ser divididas em segmentos paralelos e a partir destes desenvolver e aprimorar tarefas, ferramentas e pesquisas, assim como as descritas na área do MIR. Trabalhos como separação da voz cantante de uma música (HU; LIU, 2015) e identificação de língua em uma música (IGLESIAS, 2020) são alguns exemplos. Para isso MSA considera que a estrutura de uma música pode ser dividida em 4 princípios (NIETO *et al.*, 2020):

- Homogeneidade: assume que segmentos musicais são relativamente homogêneos em relação a um determinado atributo, resultando que limites entre segmentos diferentes são considerados como um ponto de novidade.
- Novidade: implica que a música é homogênea em cada lado do limite.
- Repetição: considera segmento uma determinada sequência que se caracteriza por conta de um atributo específico, não necessariamente se repetindo no decorrer de toda sequência, mas sim com relação ao segmento anterior.
- Regularidade: assume como segmento características que ocorrem com certa regularidade dentro dos demais segmentos. Um exemplo disso é o número de batidas que ocorrem em dois segmentos igualmente rotulados.

2.2 DISCRIMINANDO LÍNGUAS

O sinal de fala é resultado de diversos níveis de informação, os quais podem ser utilizados para discriminar línguas (AMBIKAIRAJAH *et al.*, 2011; ADAMI, 2004):

- Acústica: nível mais próximo da física original da fala, onde diferentes eventos da fala podem ser distinguidos conforme a amplitude e frequência das ondas. Informações como fonotática e da palavra podem ser extraídos a partir do nível acústico, sendo que é uma das formas de informação mais fáceis de se obter a partir de áudio bruto.
- Fonotática: cada língua tem um número finito de sons possíveis e a fonotática é o estudo de padrões de sons válidos em uma língua específica. É uma ramificação da fonologia, o estudo do sistema de sons de uma língua específica. Existe uma variação de restrições fonotáticas entre as línguas, como, por exemplo, o agrupamento de fonemas /st/ que é muito comum na língua inglesa, enquanto não é permitida na língua japonesa.

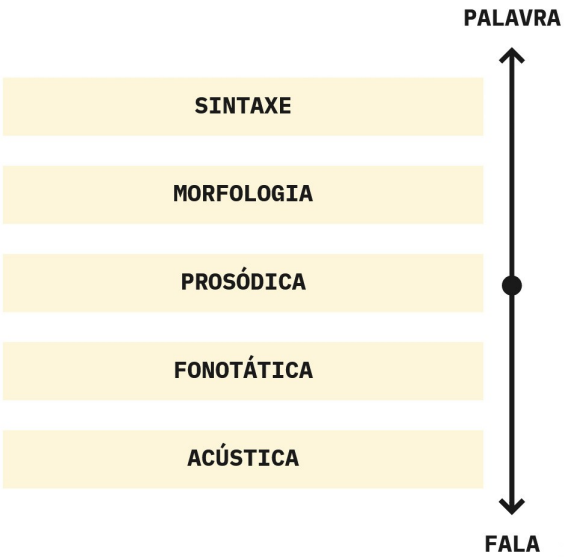
- Prosódica: línguas variam em ritmo, entonação e intensidade. Durante a fala, em conjunto com a variação de tom, volume e duração, todas essas características definem os aspectos principais do que é prosódica. Componente chave na percepção auditiva humana, prosódica engloba características que auxiliam a distinguir línguas sem entrar no detalhe da palavra. Por exemplo, linguagens tonais como o mandarim e o vietnamita, onde a entonação de uma palavra determina o significado (AMBIKAIRAJAH *et al.*, 2011).

- Morfologia: cada língua normalmente tem seu próprio conjunto de vocabulários e sua própria maneira de formar palavras. Diferentes combinações de fonemas constituem diferentes palavras, e morfologia é o campo da linguística que estuda a estrutura interna das palavras.

- Sintaxe: quando pessoas se comunicam, as palavras devem ser combinadas num modo específico que resulte numa mensagem entendível. Sintaxe é o estudo dos princípios e regras que definem o modo como as palavras numa sentença devem ser organizadas, para que assim tenham um significado. Todas as línguas têm uma variedade de regras sintáticas para organizar a distribuição das palavras, que assim formam frases e sentenças.

Estes níveis podem ser ainda organizados em dois níveis mais abrangentes (Figura 2): um nível mais próximo da fala, onde características podem ser obtidas a partir de dados de áudios brutos, e o da palavra, onde é possível diferenciar as línguas por suas palavras, diferentes vocabulários. O nível da fala aproxima características de acústica, fonética, fonotática e prosódica. Enquanto o nível da palavra contempla características como o vocabulário, sintaxe, morfologia e gramática.

Figura 2 – Níveis de características da fala e língua.



Fonte: Adaptado de (AMBIKAIRAJAH *et al.*, 2011).

2.3 PROBLEMA DE IDENTIFICAÇÃO DE LÍNGUA

O objetivo da identificação automática de língua (ou LID) é identificar a língua falada a partir de uma amostra de áudio (ZISSMAN; BERKLING, 2001). Os sistemas de LID realizam a identificação de um pré-determinado conjunto de línguas. Assim, os sistemas podem ser divididos em conjunto fechado, onde a língua a ser identificada está no conjunto de línguas pré-definido, ou em conjunto aberto, onde a língua a ser identificada pode não estar no conjunto de línguas pré-definido.

2.3.1 Arquiteturas dos Sistemas de Identificação

Os sistemas de identificação de línguas podem ser divididos em 3 abordagens. Uma das abordagens é o modelo tradicional de *machine learning*, que se caracteriza por etapas de pré-processamento, extração de características e classificação. A segunda abordagem, é o *Deep Learning*, parte do princípio que o modelo não deve somente classificar, mas também transformar características da entrada para assim trazer maior relevância aos dados, formando uma cadeia de aprendizado que trabalhe como um sistema único e não por etapas. A terceira abordagem é o *transfer learning*, caracterizada pelo compartilhamento de aprendizado e conhecimento de um sistema consolidado para um novo sistema a ser proposto, onde ambos possuem domínios que se complementam.

Modelo Tradicional

O modelo tradicional explora as características e etapas necessárias para se chegar a um sistema de classificação ideal de forma procedural. Geralmente, os sistemas podem ser divididos em 3 etapas ordenadas (Figura 3):

1. Pré-processamento: mantém e evidencia as informações mais relevantes do sinal de áudio (informação acústica), descartando informações redundantes e irrelevantes para o processo de identificação. Etapa crucial que impacta em todo o desempenho do sistema, é necessária pelas diferentes fontes de origem que um áudio de voz pode ter. Diversas fontes e meios de gravação resultam em amostras com ruídos (por exemplo, reverberação e microfone) e sons que acompanham o dado de entrada do qual se encontra o som da fala (por exemplo, som ambiente e acompanhamento instrumental) (IGLESIAS, 2020). Algumas das tarefas mais comuns aplicadas na etapa de pré-processamento são a separação e segmentação (DESHWAL; SANGWAN; KUMAR, 2019). A separação é um processo que visa isolar a voz dos demais sons como ruídos e barulhos do ambiente, os quais podem interferir na acurácia do identificador. Existem diversas abordagens para a remoção de ruídos ou sons distintos da voz, como o filtro Wiener (EL-FATTAH *et al.*, 2014). Em músicas, isolar a voz do ruído é um trabalho muito mais complexo, por tratar-se de um som polifônico.

Uma vez feita a separação da voz, o sinal mantém a duração, onde a voz foi mantida e os demais sons foram atenuados. Como a voz é o sinal de interesse, métodos de detecção de atividade vocal (*Voice Activity Detection (VAD)*) são utilizados para segmentar o sinal em trechos que contém fala (TAN; SARKAR; DEHAK, 2020).

2. Extração de características: transforma o sinal em um conjunto de características relevantes para a discriminação entre línguas (SHARMA; UMAPATHY; KRISHNAN, 2020). Estas características podem ser extraídas usando segmentos de tamanho fixo (análise de curto prazo - *Short-term Analysis*) ou de tamanho variável (análise de longo prazo - *Long-term Analysis*). A análise de curto prazo transforma as propriedades acústicas (componentes do sinal e espectro) em vetores de características de curto prazo, a partir de segmentos de tamanho fixo. Algumas das características acústicas baseadas em segmentos de tamanho fixo incluem: *Linear Predictive Coefficiets* (LPC) (SAVIC; GUPTA; ACOSTA, 1991), tom, energia (FOIL, 1986) e espectro (TORRES-CARRASQUILLO *et al.*, 2002). A análise de longo prazo transforma as propriedades acústicas em vetores de características de longo prazo, a partir de um dado segmento, não apenas um vetor de características pode ser gerado, mas sim um ou mais. Nesta análise, as características podem ser extraídas de segmentos como frases, sílabas (ROUAS *et al.*, 2003) ou fonemas (PELLEGRINO; ANDRE-OBRECHT, 2000).

3. Classificação: utiliza as características extraídas para estimar uma medida entre o vetor de características e cada uma das classes (i.e., línguas). Portanto, a partir das características extraídas, a fase de classificação faz a predição usada para se tomar a decisão. Algum dos classificadores utilizados para LID são: *Gaussian Mixture Model* (GMM) e *Support Vector Machines* (SVM) (GOLD; MORGAN; ELLIS, 2011a).

Figura 3 – Modelo tradicional.

Fonte: O Autor (2022).

Deep Learning

Diferentemente do modelo tradicional, onde cada etapa é otimizada independentemente das demais, no modelo DL todas as etapas são otimizadas simultaneamente considerando o objetivo final do problema. A partir de uma representação dos dados, a abordagem se propõe estimar o resultado esperado com um modelo único (o qual realiza tanto a extração de características como a classificação), composto de uma ou mais redes neurais treinadas de forma conjunta

(Figura 4) (PARCOLLET; MORCHID; LINARèS, 2020; GLASMACHERS, 2017; PALAZ; COLLOBERT; DOSS, 2013).

Figura 4 – Modelo *Deep Learning*

Fonte: O Autor (2022).

O modelo DL, aplicado para a identificação de línguas, pode ser dividido em duas etapas:

- Transformação de características: o modelo DL, apesar de ser um modelo singular, em muitos casos requer uma etapa semelhante às etapas de pré-processamento e extração de características descritas na Seção 2.3.1 (PARCOLLET; MORCHID; LINARèS, 2020). A transformação de características é a primeira camada do modelo (PALAZ; COLLOBERT; DOSS, 2013). Transforma o sinal de áudio em características, reduz a quantidade de dados e seleciona informações relevantes baseado no objetivo do modelo proposto (CASTRO *et al.*, 2018). A transformação de características pode ser distinta em abordagens que resultam em características acústicas e informações estatísticas. Ambas abordagens extraem e transformam um sinal de áudio em características acústicas (por exemplo, MFCC (PADI; MOHAN; GANAPATHY, 2019), espectrogramas na escala Mel (DIELEMAN; SCHRAUWEN, 2014) e *Perceptual Linear Predictive features* (PLP)) e estatísticas (por exemplo, frequência de fonemas (RENAULT; VAGLIO; HENNEQUIN, 2021)).

- Modelo neural de identificação: esta etapa tem por objetivo realizar a extração das características e a sua classificação através de uma Rede Neural Artificial (RNA). É responsável por receber as informações do ambiente (dados de entrada), aprender por técnicas de aprendizado como *backpropagation* (AGGARWAL *et al.*, 2018) e resultar no melhor modelo dada a base de dados de treinamento, para então responder o ambiente.

  A RNA é a estrutura básica para redes neurais. A partir dela é possível expandir para modelos como: *Deep Neural Network* (DNN) caracterizada pela quantidade de camadas (GOODFELLOW; BENGIO; COURVILLE, 2016), CNN desenvolvida para trabalhar com dados bidimensionais (AGGARWAL *et al.*, 2018) e *Recurrent Neural Network* (RNN) capazes de propagarem informações de um estado anterior para o atual (STAUDEMEYER; MORRIS, 2019). Estes modelos de redes são aplicados em tarefas de reconhecimento de locutor (BAI; ZHANG, 2021; CHUNG; NAGRANI; ZISSERMAN, 2018), identificação de língua

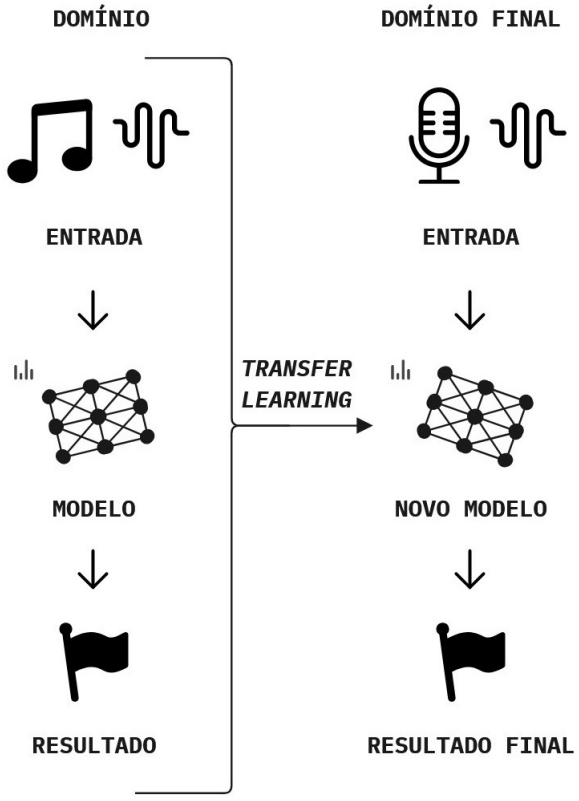
em áudio e em músicas (DIELEMAN; SCHRAUWEN, 2014; IGLESIAS, 2020; GROENBROEK, 2021; TRONG; HAUTAMÄKI; LEE, 2016).

Transfer-learning

Em muitos casos, modelos tradicionais e DL falham pela falta de dados de treinamento ou diferenças entre as características do ambiente onde foram coletados os dados e o ambiente onde o modelo é aplicado. Estes problemas motivaram no desenvolvimento do *Transfer-learning* (YANG *et al.*, 2020).

Transfer-learning é um método de aprendizado de máquina onde é extraído e usado o conhecimento de um ou mais domínios similares, compartilhando com o domínio final para aumentar o desempenho do cenário esperado (Figura 5). As características de entrada são usadas para criar conhecimento entre domínios próximos, gerando uma classificação mais assertiva a partir do conhecimento compartilhado entre os domínios (ZHUANG *et al.*, 2021). *Transfer-learning* ajuda no desenvolvimento de tarefas de classificação em áreas menos desenvolvidas e que até então possuem uma quantidade escassa de dados (YANG *et al.*, 2020), como: classificação de sentimento (WANG; MAHADEVAN, 2011), classificação de imagens (KULIS; SAENKO; DARRELL, 2011) e de línguas em texto (PRETTENHOFER; STEIN, 2010).

Figura 5 – Modelo *Transfer Learning*.



Fonte: Adaptado de (YANG *et al.*, 2020).

2.3.2 Medidas de Desempenho

Uma representação muito utilizada para a avaliação e medição do desempenho de sistemas de classificação é a matriz de confusão (GOLD; MORGAN; ELLIS, 2011b; KRUSPE, 2018; MEHRABANI; HANSEN, 2011; MUKHERJEE *et al.*, 2021). A matriz tabula o número de classificações corretas em comparação com as classificações preditas para cada língua. Em problemas de classificação de duas classes (classificação binária), a matriz possui uma dimensão de 2 por 2 como apresentado no Quadro 1 (GOLD; MORGAN; ELLIS, 2011b).

Os resultados de um classificador binário em uma matriz de confusão (Quadro 1) podem ser definidos por:

- Verdadeiro positivo (VP): itens da língua A corretamente classificados como itens da língua A.
- Falso positivo (FP): itens da língua B incorretamente classificados como itens da língua A.
- Verdadeiro negativo (VN): itens da língua B corretamente classificados como itens da língua B.
- Falso negativo (FN): itens da língua A incorretamente classificados como itens da língua B.

Quadro 1 – Matriz de confusão para classificação binária.

Classe	Classificações positivas	Classificações negativas
Positivo	VP	FN
Negativo	FP	VN

Fonte: Adaptado de (ROXBERGH, 2019).

A diagonal principal da matriz apresenta as classificações corretas enquanto as demais posições apresentam as classificações incorretas. O resultado ideal para um classificador seria uma matriz com todos os valores zerados, exceto a diagonal principal, indicando apenas classificações corretas.

Para um problema de 3 classes, a matriz de confusão poderia ser semelhante ao da Tabela 1. Neste exemplo, a classe B teve 6 classificações corretas e 2 classificações incorretas, 1 como A e 1 como C. Nesta matriz de confusão, os erros devem ser avaliados por classe, isto, é, o problema de 3 ou mais classes é transformado em um problema de duas classes do tipo um contra todos.

Diversas medidas de desempenho podem ser estimadas a partir da matriz de confusão. Algumas destas medidas, incluem:

Tabela 1 – Matriz de confusão multi-classes.

Classe	Classe A	Classe B	Classe C
Classe A	3	3	1
Classe B	1	6	1
Classe C	2	2	4

Fonte: Adaptado de (ROXBERGH, 2019).

- Acurácia: razão entre o número de classificações corretas em relação a toda amostragem de dados (ROXBERGH, 2019), conforme:

$$Acurácia = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}. \tag{2.1}$$

  Tipicamente utilizada em casos onde as classes estão balanceadas, isto é, as classes possuem o mesmo número de amostras (BIRJALI; KASRI; BENI-HSSANE, 2021).

- Precisão: apresenta a relação entre o número de classificações positivas corretas em relação a todas as classificações positivas (ROXBERGH, 2019; CHOI; WANG, 2021), conforme:

$$Precisão = \frac{\text{VP}}{\text{VP} + \text{FP}}. \tag{2.2}$$

  Utilizada para medir o quanto um sistema classifica informações relevantes, tendo alta confiabilidade da classificação (SAMMUT; PHUNG; WEBB, 2017; BIRJALI; KASRI; BENI-HSSANE, 2021).

- *Recall*: descreve a relação entre o número de classificações positivas corretas em relação às classificações a todas as amostragens positivas (ROXBERGH, 2019)(CHOI; WANG, 2021), conforme:

$$Recall = \frac{\text{VP}}{\text{VP} + \text{FN}}. \tag{2.3}$$

  Uma medida utilizada, por exemplo, em um sistema de detecção de fraudes em um banco, onde é possível lidar com nenhuma ou muito poucas transações suspeitas (SHALEV-SHWARTZ; BEN-DAVID, 2014).

- F1-score: é a média harmônica da precisão e *recall* (ROXBERGH, 2019; RENAULT; VAGLIO; HENNEQUIN, 2021), conforme:

$$F_1 = \frac{2 \times Precisão \times Recall}{Precisão + Recall}, \tag{2.4}$$

  A medida vária entre 0 e 1, onde resultados mais próximos de 1 indicam ambas precisão e *recall* satisfatórias (SAMMUT; PHUNG; WEBB, 2017).

2.4 PROBLEMA DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICA

O problema de identificação de língua em música possui o mesmo objetivo que LID, mas a fonte é uma música. Assim, a voz não é falada, mas cantante. Este tipo de tarefa pode ser utilizada na categorização de músicas por língua e no auxílio de outros modelos para música, como: identificação do humor, na identificação de letras de músicas, transcrição de letras e geração de músicas.

A voz cantante se distingue da voz falada em vários aspectos. Características prosódicas como entonação, ritmo e intensidade variam para se adequar a melodia da música cantada (TSAI; WANG *et al.*, 2004). Palavras e frases que compõem o vocabulário usado se distinguem em muitos casos de conversações normais, podendo haver uma diferente probabilidade de ocorrência em comparação com a fala, distinção causada para trazer mais emoção para quem a ouve (TSAI; WANG *et al.*, 2004; KRUSPE; FRAUNHOFER, 2014).

Um problema que torna a identificação complexa é a existência de múltiplas fontes de áudio. Uma música pode ser caracterizada pelos sons emitidos pelo cantor, *backing vocals*, instrumentos musicais harmônicos e percussivos. Assim, torna-se necessário remover ou dirimir toda informação não relevante (isto é, não relacionada com a voz cantante) para não impactar na acurácia da identificação (KRUSPE, 2018).

Em uma pesquisa bibliográfica, verifica-se que os trabalhos de identificação de língua em música iniciam em 2004 (TSAI; WANG *et al.*, 2004). A Figura 6 apresenta uma cronologia com os trabalhos realizados para resolver o problema de identificação de língua em música e voz cantante. É importante destacar que parte destes trabalhos utilizam como entrada somente a voz cantante, isto é, ignoram as interferências causadas pelas demais fontes de áudio. Nas seções a seguir, serão descritos os trabalhos publicados sobre SLID, relatando os modelos tradicionais e DL utilizados.

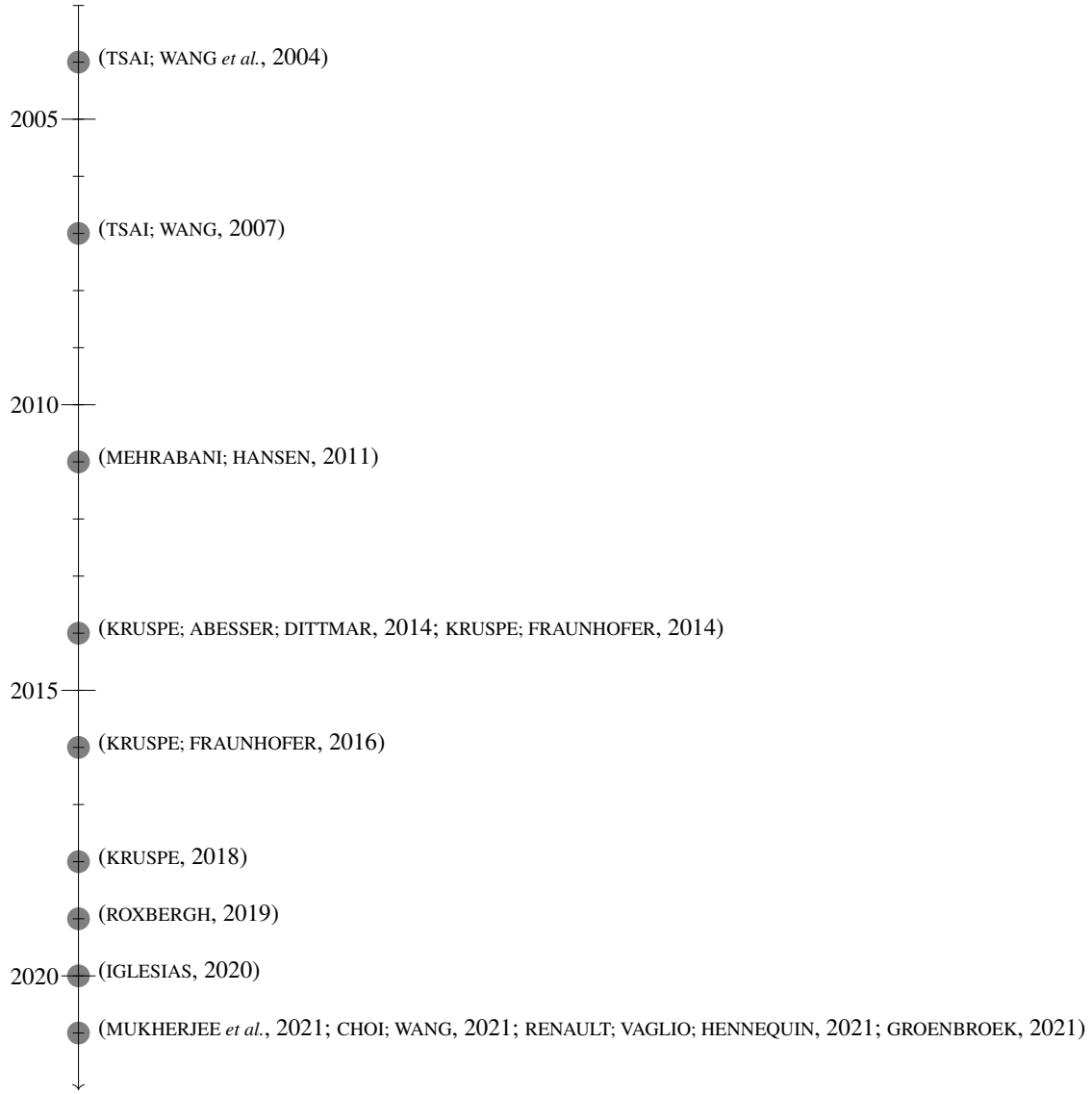
Não foram encontrados trabalhos que usem *transfer learning* para desenvolver um identificador de língua em músicas. Apesar de que Iglesias (2020) usa conceitos e práticas de compartilhamento de conhecimento para fazer a isolação da voz antes de segmentar e classificar, o autor não descreve que o conceito utilizado é de um modelo *transfer learning*.

2.4.1 Modelos Tradicionais

O modelo tradicional é a abordagem mais utilizada para o desenvolvimento de identificadores de língua em música. Dentre estes trabalhos é possível separá-los pelo tipo de análise: curto e longo prazo.

A análise de curto prazo extrai características acústicas usando segmentos de tamanho fixo. Os sistemas propostos utilizam características acústicas do espectro de frequência, como *Line Spectral Frequency* (LSF) (MUKHERJEE *et al.*, 2021), PLP e MFCC (KRUSPE; FRAUNHOFER,

Figura 6 – Cronologia de trabalhos em SLID.



Fonte: do Autor (2022).

2014; KRUSPE, 2018), e dinâmicas, como *Shifted Delta Cepstrum* (SDC) (KRUSPE; FRAUNHOFER, 2014; KRUSPE, 2018).

A análise de longo prazo geralmente produz uma representação baseada em *tokens* a partir de segmentos de tamanho variável. Mehrabani e Hansen (2011) utilizam tokens obtidos a partir da análise do contorno do tom da voz. Além de extrair características de uma análise de curto prazo, Kruspe (2018) explora características de análise de longo prazo com probabilidades *a posteriori* de reconhecimento de fonemas. Tsai, Wang *et al.* (2004), Tsai e Wang (2007) realiza a tokenização do sinal de voz que modele as dinâmicas das unidades fonológicas (i.e., diferentes sons da língua).

A classificação utiliza os mesmos modelos aplicados em problemas tradicionais de reconhecimento de padrões. Os classificadores utilizam modelos como *Multi-Layer Percep-*

trons (MLP) (KRUSPE; FRAUNHOFER, 2014), *Random Forest* Mukherjee *et al.* (2021), SVM (KRUSPE; FRAUNHOFER, 2014; KRUSPE, 2018) e n-gramas (TSAI; WANG *et al.*, 2004; TSAI; WANG, 2007; MEHRABANI; HANSEN, 2011).

A Tabela 2 apresenta as acurácias obtidas pelos trabalhos propostos de SLID que utilizam o modelo tradicional. Relacionando as línguas presentes na base de dados, quais tipos de características foram utilizadas e os modelos de classificação.

Tabela 2 – Tabela de trabalhos de SLID utilizando o modelo tradicional.

Referência	Línguas	Quantidade	Características	Classificação	Acurácia
Tsai, Wang *et al.* (2004)	Inglês Mandarim	112* 112*	Tokens	N-gramas	70%
Tsai e Wang (2007)	Mandarim Inglês Japonês	346*	Tokens	N-gramas	65%
Mehrabani e Hansen (2011)[(1)]	Persa Hindu Mandarim	12**	Tokens	N-gramas	82.7%
Kruspe e Fraunhofer (2014)[(1)]	Inglês Alemão Espanhol	116-196* 116-196* 116-196*	MFCC PLP	MLP SVM	58% 72%
Kruspe (2018)[(1)]	Inglês Alemão Espanhol	116-196* 116-196* 116-196*	MFCC + PLP Probabilidade *a posteriori*	SVM	77% 63%
Mukherjee *et al.* (2021)	Inglês Bangla Hindu	22** 30** 20**	LSF	*Random Forest*	98.62%

Nota: [(1)] A base de dados possui apenas voz cantante (a capela).
* Quantidade em número de músicas.
** Quantidade em horas.

Fonte: O Autor (2022).

2.4.2 *Deep Learning*

Dentre os trabalhos que aplicam o modelo DL para fazer a identificação de língua em músicas, pode-se agrupá-los naqueles que fazem a transformação resultando em características acústicas e aqueles resultantes em informações estatísticas. Renault, Vaglio e Hennequin (2021) estima a ocorrência de fonemas no sinal de entrada e produz vetores com as probabilidades a cada segmento de tempo. Iglesias (2020) e Groenbroek (2021) utilizam MFCC. Groenbroek (2021) também utiliza representações baseadas em espectrogramas na escala Mel.

A classificação utiliza modelos de rede neural. Iglesias (2020) implementou um modelo *Temporal Convolutional Network* (TCN) com diferentes configurações (como quantidade de camadas, canais e épocas), atuando diretamente no sinal de áudio e extraindo características

MFCC. Renault, Vaglio e Hennequin (2021) utiliza um modelo RNN. Groenbroek (2021) utiliza dois modelos: DNN e VGGish (um modelo com base em CNN).

A Tabela 3 apresenta as acurácias obtidas pelos trabalhos propostos de SLID que utilizam o modelo DL. Relacionando as línguas presentes na base de dados, quais tipos de características foram utilizadas e os modelos de classificação.

Tabela 3 – Tabela de trabalhos de SLID utilizando o modelo DL.

Referência	Línguas	Quantidade[2]	Características	Classificação	Acurácia
Iglesias (2020)	Português Italiano Inglês Francês Espanhol Alemão Japonês	800 800 800 800 800 800 800	MFCC Sem extração	TCN	23% 19%
Renault, Vaglio e Hennequin (2021)[1]	Inglês Francês Alemão Italiano Espanhol	5358	Estatísticas	RNN	91.74%
Groenbroek (2021)	Inglês Holandês Alemão Francês Espanhol Português	4549 4582 4713 4607 4648 4624	MFCC *Mel Spectrograms*	DNN *VGGish*	36% 42%

Nota: [1] A base de dados possui apenas voz cantante (a capela).
[2] Quantidade é em número de músicas.

Fonte: O Autor (2022).

2.5 *DEEP LEARNING* EM IDENTIFICAÇÃO DE LÍNGUAS

O uso de redes neurais tem por objetivo extrair características e classificar numa abordagem DL. Nesta seção é abordado o contexto e funcionamento de uma RNA, a base de modelos de redes neurais, para por fim ser explicado o funcionamento de RNN e CNN, as duas principais arquiteturas DNN, ambas amplamente utilizadas em *Natural Language Processing* (NLP) (YIN *et al.*, 2017).

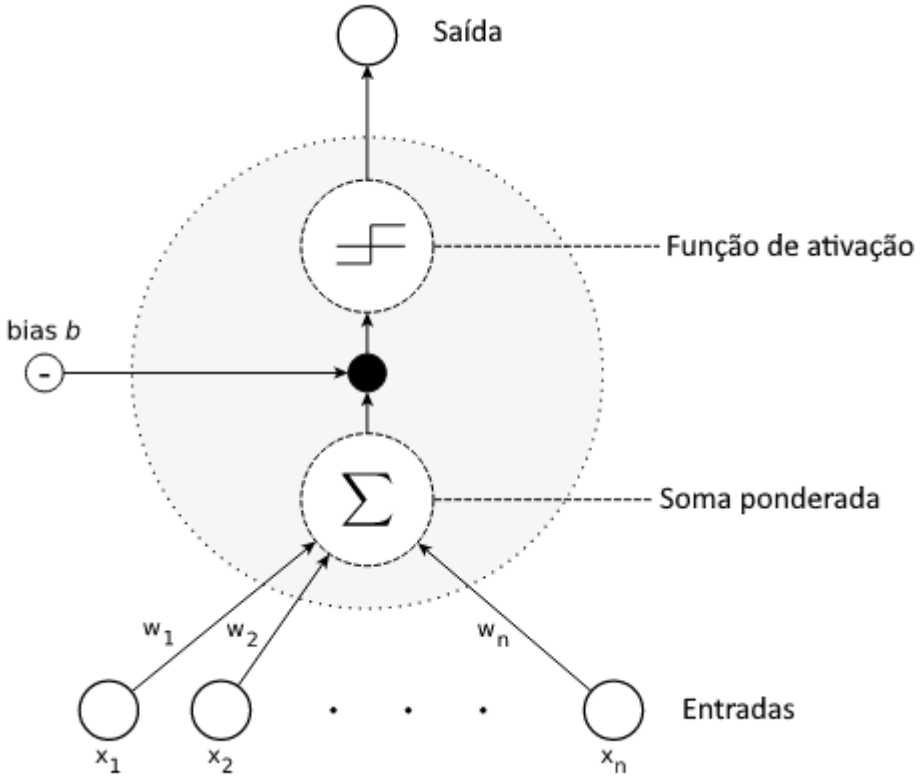
2.5.1 Rede Neural Artificial

Rede Neural Artificial (RNA) é uma técnica de aprendizado de máquina que simula o mecanismo de aprendizado em organismos biológicos (AGGARWAL *et al.*, 2018). A estrutura básica é uma rede de pequenas unidades de processamento, ou neurônios, unidas entre si por

conexões ponderadas (GRAVES, 2012). As redes neurais podem ser divididas em redes de apenas uma camada ou multi-camadas.

Redes com apenas uma camada (*single-layer network*) são a representação mais simples de uma RNA. O tipo mais básico de um neurônio artificial é chamado *perceptron*, o qual é caracterizado por um conjunto de entradas (pesos), um limite, uma função de ativação e apenas uma saída (Figura 7) (AGGARWAL *et al.*, 2018; STAUDEMEYER; MORRIS, 2019). Dado um conjunto de entradas (valores reais ponderados), todos os valores são somados. A partir deste somatório, uma função de ativação valida se o resultado é superior ao limite definido, resultando numa resposta boleana verdadeira (1), caso contrário resulta na resposta boleana falsa (-1) (STAUDEMEYER; MORRIS, 2019).

Figura 7 – Estrutura de um Perceptron.



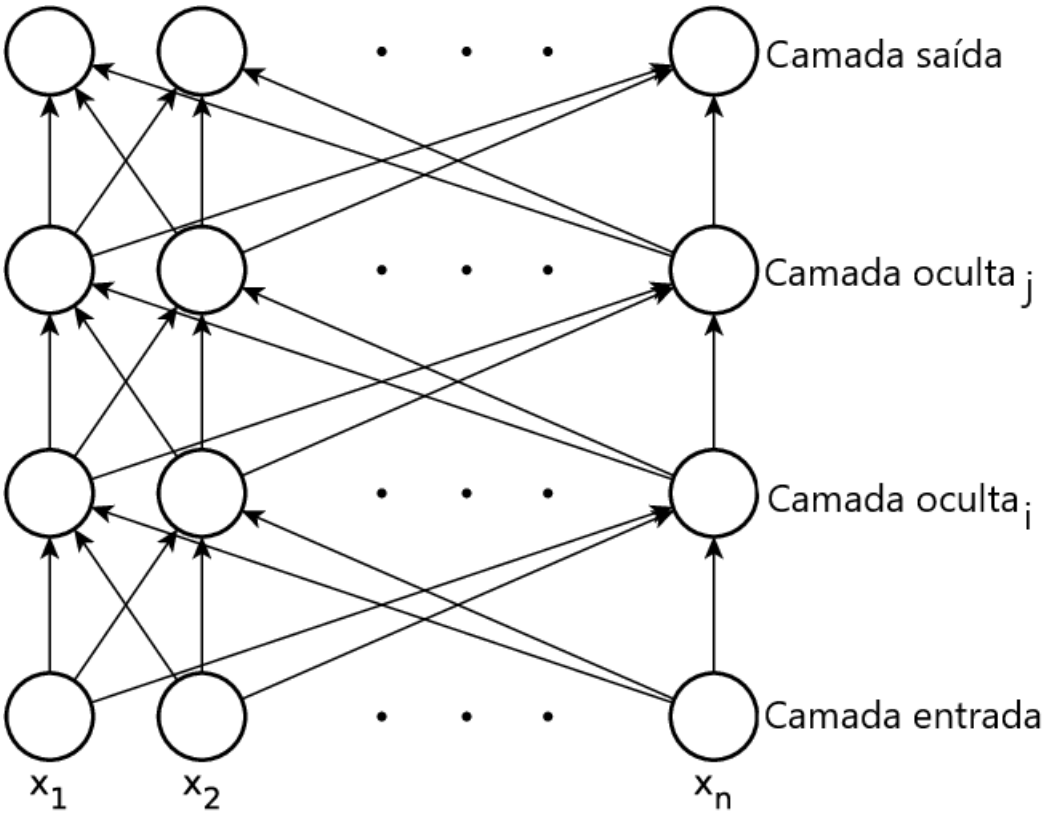
Fonte: Adaptado de (STAUDEMEYER; MORRIS, 2019).

Em uma rede neural multi-camadas os neurônios são distribuídos em 3 camadas: entrada, saída e camadas ocultas (AGGARWAL *et al.*, 2018). Os neurônios da camada de entrada recebem sinais do ambiente e os neurônios da camada de saída apresentam os sinais para o ambiente. Os neurônios que não são conectados diretamente com o ambiente são chamados neurônios ocultos, por se comunicarem apenas com outros neurônios (STAUDEMEYER; MORRIS, 2019). Uma rede neural multi-camadas pode ser dividida em duas categorias:

- *Feed-forward* (*Feed-foward Neural Network* (FFNN)): redes nas quais camadas sucessi-

vas alimentam umas às outras. Partindo da camada de entrada para a saída (AGGARWAL *et al.*, 2018), todos os neurônios em uma camada provêm a entrada de todos os neurônios da próxima camada (STAUDEMEYER; MORRIS, 2019; AGGARWAL *et al.*, 2018), conforme ilustra a Figura 8.

Figura 8 – Rede neural multi-camadas.



Fonte: Adaptado de (STAUDEMEYER; MORRIS, 2019).

- FFNN com *feedback*: rede neural com conexões de *feedback*, onde a saída de um neurônio é direcionada de volta para si. Quando redes *feed-forward* incluem conexões de *feedback*, conexões cíclicas (GRAVES, 2012), elas são chamadas Redes Neurais Recorrentes (RNN) (GOODFELLOW; BENGIO; COURVILLE, 2016).

A técnica de aprendizado mais comum em RNA é o algoritmo *backpropagation*, técnica que utiliza método de otimização gradiente descendente para aprender os pesos em redes multi-camadas (AGGARWAL *et al.*, 2018). A partir da camada de entrada, os dados são propagados através da rede até que os pesos gerados pelos neurônios cheguem a camada de saída, para então produzir o resultado da rede. A partir do resultado da rede, o algoritmo de aprendizado propaga o erro de volta. Com isso os pesos são atualizados para reduzir o erro obtido com os dados de treino atual (STAUDEMEYER; MORRIS, 2019; AGGARWAL *et al.*, 2018).

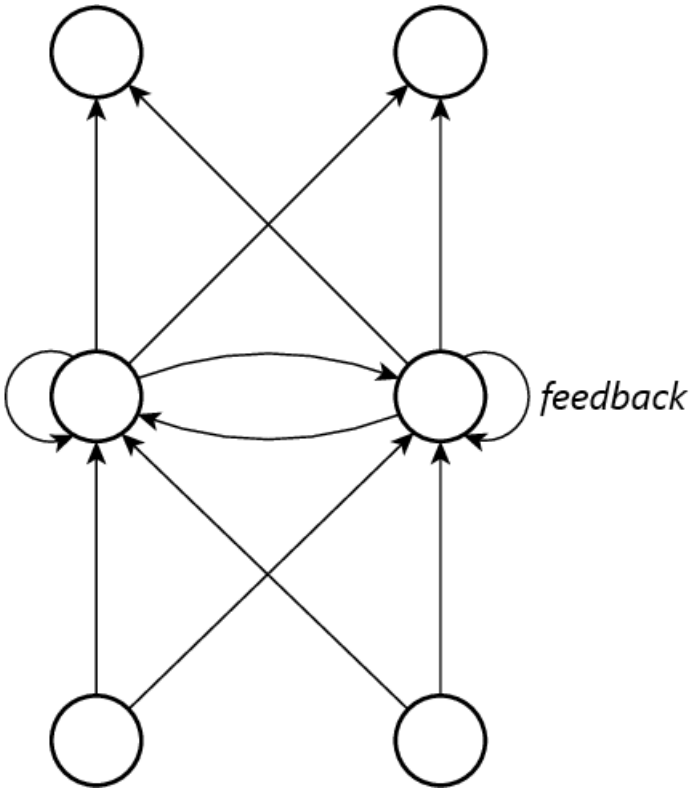
O treinamento da rede ocorre em lotes, onde cada lote é composto por uma quantidade n de instâncias para treinar a rede. Cada treinamento de lotes, é uma rodada de treinamento da

rede, esta rodada é denominada uma época. Cada rodada de treinamento da rede, é denominada como época (GÉRON, 2022). O tamanho do lote pode impactar o desempenho e tempo de treinamento do modelo, onde lotes grandes podem ser eficientemente processados com *Graphics Processing Unit* (GPU)s, mas levar a instabilidades no treinamento e resultando numa rede que não generalize como naquelas treinadas com lotes menores (GÉRON, 2022).

2.5.2 *Recurrent Neural Network*

Recurrent Neural Network (RNN) são redes neurais utilizadas para processar dados sequenciais, que considera as dependências temporais entre as características dos dados de entrada para ser feita a classificação (GOODFELLOW; BENGIO; COURVILLE, 2016). Possuem um estado interno em cada etapa da classificação com o uso de conexões de *feedback* (Figura 9), possibilitando à rede propagar informações de um evento anterior para etapas em processamento (STAUDEMEYER; MORRIS, 2019).

Figura 9 – Rede neural recorrente.



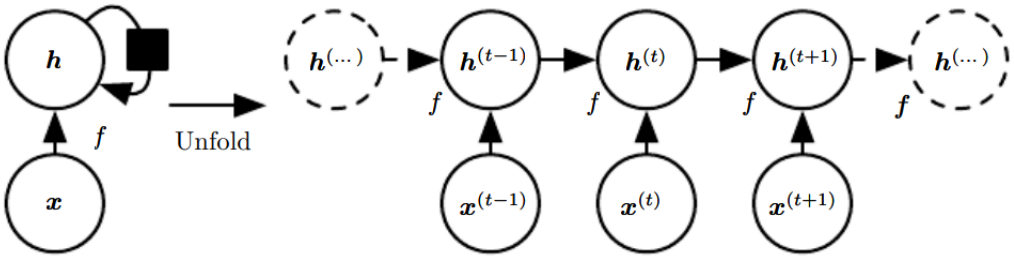
Fonte: Adaptado de (STAUDEMEYER; MORRIS, 2019).

Duas técnicas de aprendizado são comumente utilizadas para treinar a rede (STAUDEMEYER; MORRIS, 2019):

- *Backpropagation* através do tempo (*Backpropagation Through Time* (BPTT)): utiliza do fato que, por um período finito de tempo, existe uma FFNN com comportamento idêntico para toda a RNN. Para obter essa FFNN a rede é desdobrada no tempo (*unfold*),

transformação de um processo recursivo ou recorrente em um processo com uma estrutura repetitiva, correspondendo em uma cadeia de eventos conforme ilustra a Figura 10 (GOODFELLOW; BENGIO; COURVILLE, 2016; STAUDEMEYER; MORRIS, 2019).

Figura 10 – Processo de *unfold*.



Fonte: Goodfellow, Bengio e Courville (2016).

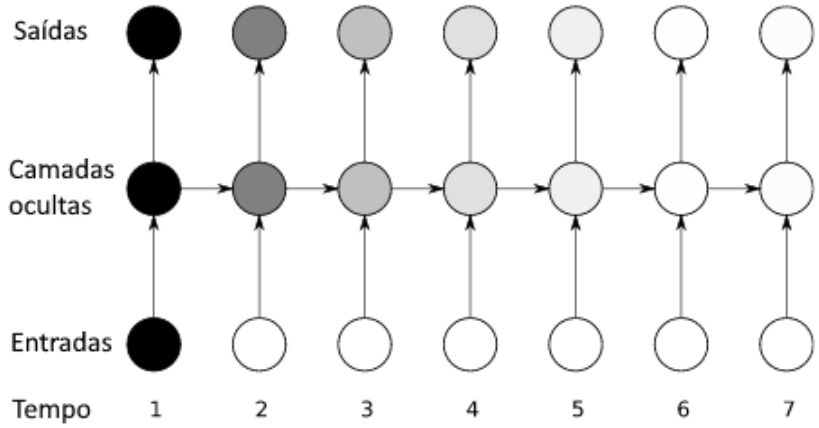
A rede neural desdobrada pode ser treinada com o algoritmo *error backpropagation*. A rede inicia num tempo t' até o tempo final t (o intervalo entre t' e t é denominado época), para cada período é calculado o erro e propagado na rede para atualizar os pesos, para por fim a RNN atualizar seus pesos com o somatório dos pesos correspondentes de cada unidade de tempo (STAUDEMEYER; MORRIS, 2019; GOODFELLOW; BENGIO; COURVILLE, 2016).

- Aprendizado recorrente em tempo real (*Real-Time Recurrent Learning* (RTRL)): não depende da propagação de erro, toda a informação necessária para ponderar os pesos é coletada no momento que o conjunto de entrada é recebido pela rede (STAUDEMEYER; MORRIS, 2019). Faz com que as alterações dos pesos sejam feitas a cada etapa de tempo, visando minimizar o erro médio da rede, erro dado a cada etapa de tempo (WILLIAMS; ZIPSER, 1989; STAUDEMEYER; MORRIS, 2019).

Ambas as técnicas utilizadas para treinar uma RNN possuem um problema: o gradiente tende a crescer ou encolher a cada passo de tempo da rede (dissipação do gradiente, Figura 11) (HOCHREITER, 1998). Com muitas etapas de tempo, o erro excede resultando em pesos oscilantes ou gera um erro pequeno resultando em tempo de treinamento excessivo, invalidando a rede (STAUDEMEYER; MORRIS, 2019; HOCHREITER; SCHMIDHUBER, 1997).

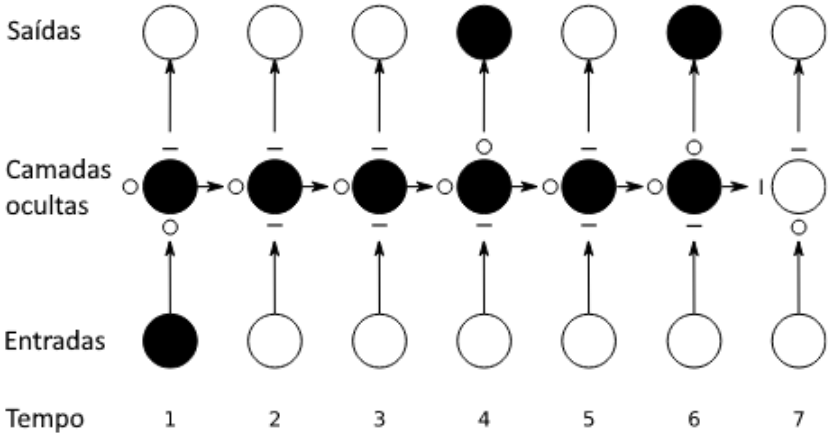
Para evitar a dissipação de gradiente, uma solução é a implementação do método LSTM (STAUDEMEYER; MORRIS, 2019). Este método é baseado na ideia de criar caminhos através do tempo, permitindo que a rede acumule informação por um longo período (GOODFELLOW; BENGIO; COURVILLE, 2016). Com isso, Goodfellow, Bengio e Courville (2016) afirmam que redes LSTM aprendem informações de longo prazo mais facilmente que RNN tradicionais. Além disso, tais autores alegam que redes LSTM preservam o gradiente através do tempo, conforme ilustra a Figura 12 (GRAVES, 2012).

Figura 11 – Dissipação do gradiente. O sombreamento dos neurônios na rede desdobrada indica a dissipação do gradiente a cada passo de tempo.



Fonte: Adaptado de (GRAVES, 2012).

Figura 12 – Dissipação do gradiente com LSTM.



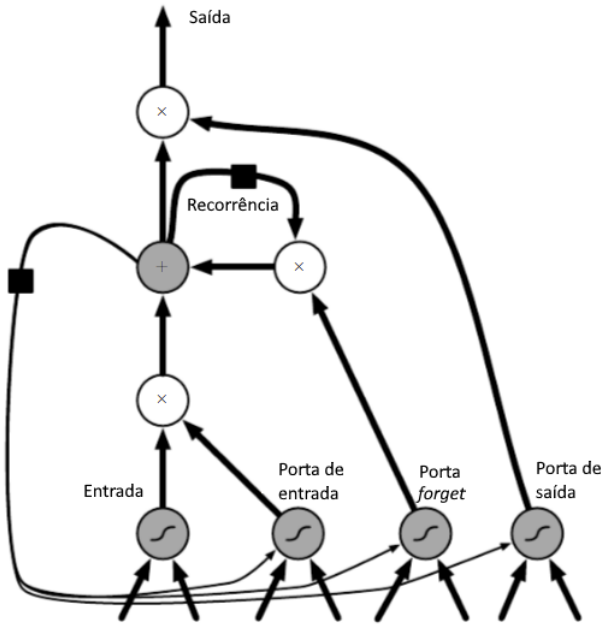
Fonte: Adaptado de (GRAVES, 2012).

Redes recorrentes LSTM são iguais a redes RNN tradicionais, exceto que na camada oculta as unidades são substituídas por blocos LSTM, ou blocos de memória (GRAVES, 2012; GOODFELLOW; BENGIO; COURVILLE, 2016). Cada célula contém uma recorrência interna, além de possuir uma entrada, saída e um sistema de unidades de portas que controlam o fluxo dos pesos (Figura 13). Possui uma porta de entrada que decide quais informações devem ser armazenadas, uma porta de saída responsável por decidir quais informações serão enviadas para as próximas células e a porta *Forget*, responsável por definir quais informações devem ser excluídas na célula (STAUDEMEYER; MORRIS, 2019; GRAVES, 2012).

2.5.3 *Convolutional Neural Network*

CNN são redes neurais utilizadas para processar dados distribuídos e representados de forma espacial com topologia do tipo grade. Áudios e imagens são exemplos de dados utilizados em redes CNNs, onde são distribuídos de forma unidimensional (1D) em intervalos de tempo

Figura 13 – Célula LSTM.



Fonte: Adaptado de (GOODFELLOW; BENGIO; COURVILLE, 2016).

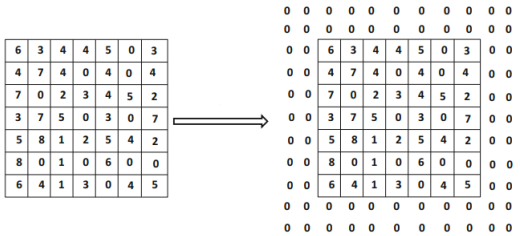
regulares, ou distribuídos de forma bidimensional (2D) (GOODFELLOW; BENGIO; COURVILLE, 2016). Como qualquer DNN, a CNN é baseada em neurônios organizados em camadas, podendo aprender representações hierárquicas. Dentre as camadas ocultas, CNNs devem possuir ao menos uma camada de convolução (*convolutional layer*), a etapa mais importante de toda a rede, e seguida normalmente por camadas de ativação e *pooling* (KATTENBORN *et al.*, 2021; GÉRON, 2022; ALZUBAIDI *et al.*, 2021).

A camada de convolução é responsável por aplicar a operação de convolução nos dados de entrada, visando gerar um mapa de características a ser utilizado como entrada na próxima camada da rede. Para isso, a camada utiliza filtros, denominados *kernels*, descritos por uma grade de valores ou números discretos. O filtro é utilizado para aplicar o produto escalar, multiplicação dos valores correspondentes e a soma para gerar um único valor escalar, por toda a grade de entrada (ALZUBAIDI *et al.*, 2021; GOODFELLOW; BENGIO; COURVILLE, 2016). Os parâmetros que definem o *kernel* e comportamento da convolução são:

- Definição de valores do *kernel*: normalmente o *kernel* é preenchido por números randômicos, para atuarem como os pesos no início do processo de treinamento. Valores que são ajustados a cada etapa de treinamento, para que o *kernel* aprenda a extrair características cada vez mais significativas (ALZUBAIDI *et al.*, 2021).
- Tamanho do *kernel*: o tamanho da grade do filtro a ser utilizado para aplicar a convolução, quanto maior o filtro menor será o tamanho do mapa de características resultante.

- *Padding*: a operação de convolução gera uma saída reduzida em relação à entrada, geralmente sendo essa uma característica indesejada por possibilitar a perda de informações nas bordas das grades (AGGARWAL *et al.*, 2018). Para contornar este problema é utilizado o *padding*, tamanho da borda a ser aplicado na grade original, conforme ilustra a Figura 14 (ALZUBAIDI *et al.*, 2021).

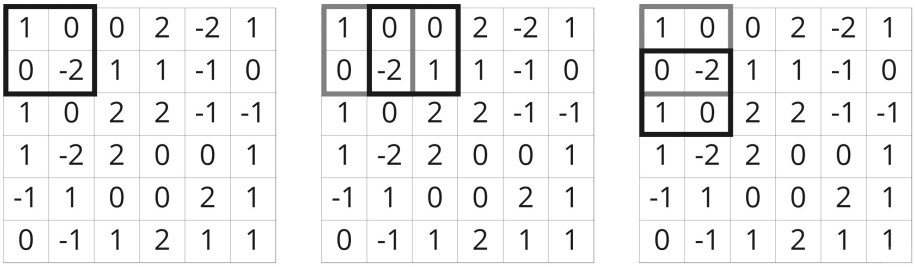
Figura 14 – Aplicação de *padding* com tamanho 2.



Fonte: Adaptado de (AGGARWAL *et al.*, 2018).

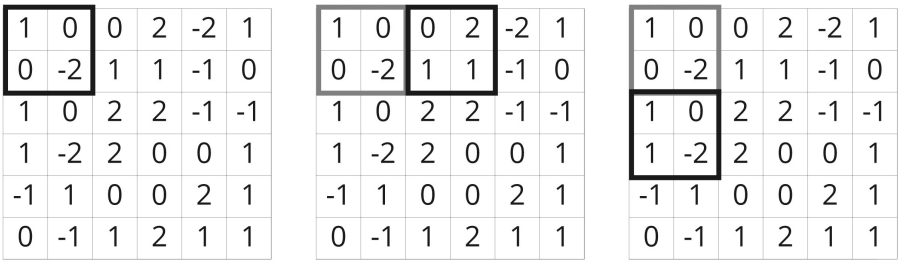
- *Stride*: para o *kernal* cobrir toda a grade ele se desloca horizontal e verticalmente, o valor do *stride* define o tamanho do deslocamento que deve ser feito. As Figuras 15 e 16 ilustram dois casos de deslocamento, com *stride* de tamanho 1 e 2 respectivamente.

Figura 15 – *Stride* de 1.



Fonte: O Autor (2022).

Figura 16 – *Stride* de 2.

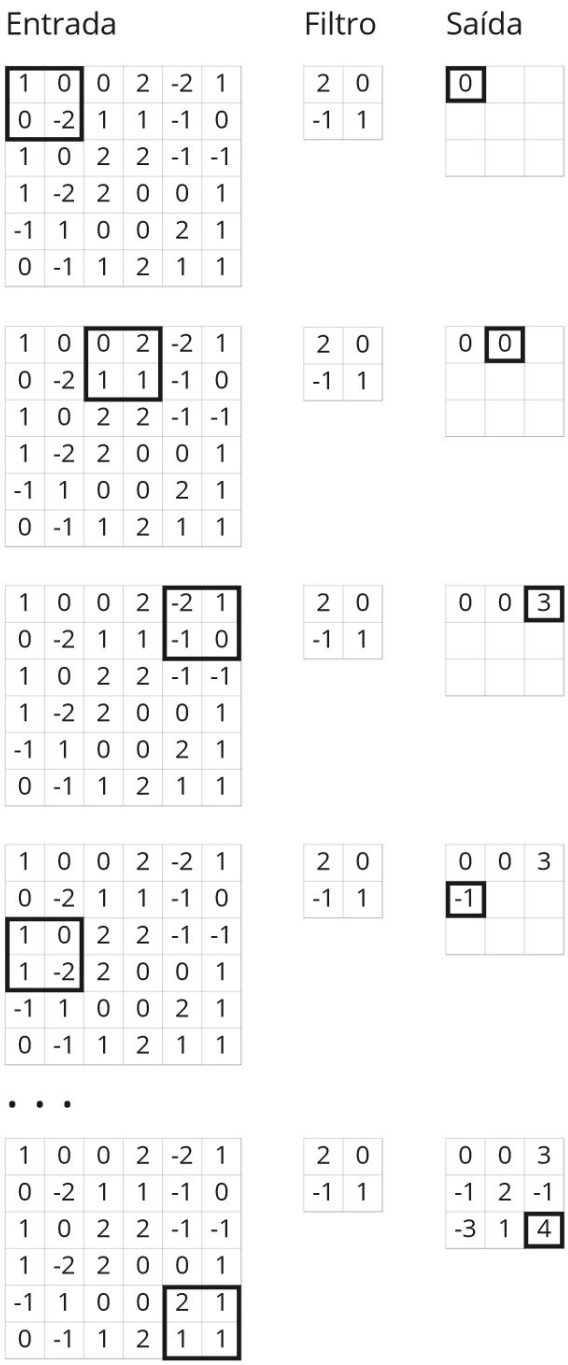


Fonte: O Autor (2022).

As camadas de uma rede CNN tem seus dados de entrada organizados em 3 dimensões, largura, altura e profundidade. A largura e a altura são respectivas as dimensões da grade, a

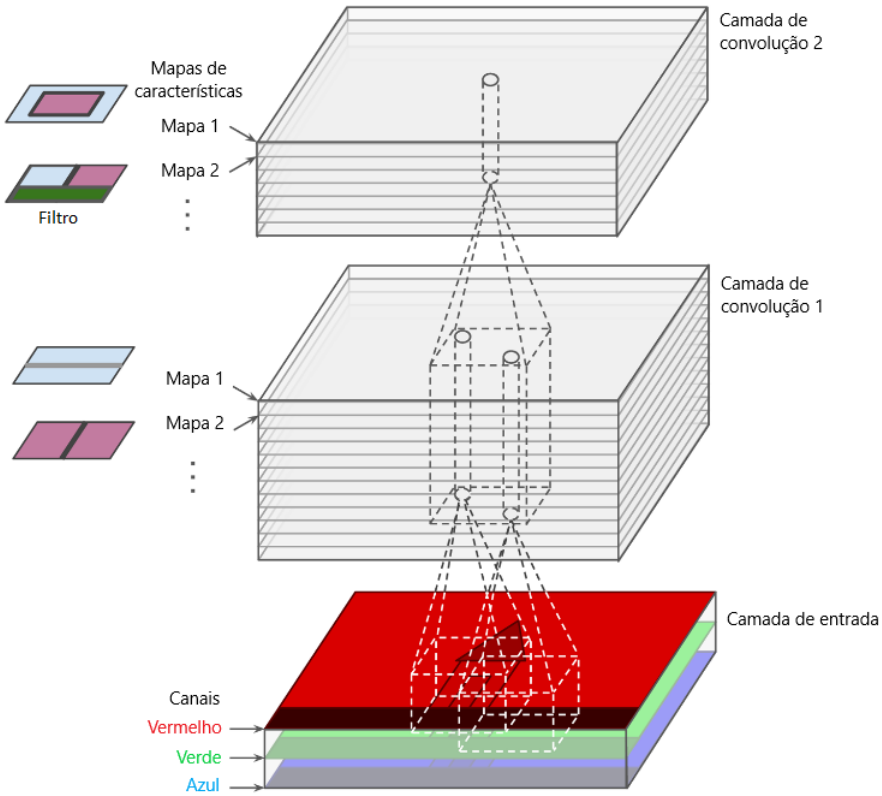
profundidade é respectiva ao número de canais (ALZUBAIDI *et al.*, 2021) de cada camada. O número de canais determina o número de mapas de características nas camadas ocultas e a quantidade de canais nos dados de entrada, por exemplo, o número de cores primarias em uma imagem (azul, verde e vermelho), onde cada cor primária representa um canal (AGGARWAL *et al.*, 2018). A Figura 17 ilustra o processo de convolução em apenas um canal e a Figura 18 ilustra o processo de convolução com canais em mais de uma camada.

Figura 17 – Camada de convolução com um *kernel* de 2x2 e *stride* de 2.



Fonte: Adaptado de (ALZUBAIDI *et al.*, 2021).

Figura 18 – Camada de convolução com canais.



Fonte: Adaptado de (GÉRON, 2022).

A camada de ativação aplica a função de ativação para determinar se um neurônio é ativo ou não (KATTENBORN *et al.*, 2021; ALZUBAIDI *et al.*, 2021). Na rede CNN a ativação é aplicada para cada um dos valores na camada, sem alterar as dimensões por ser um mapeamento de um para um de valores de ativação (AGGARWAL *et al.*, 2018). A função de ativação mais utilizada em CNN é a *Rectified Linear Unit* (ReLU), que se resume em converter todos os valores para números positivos, conforme:

$$f(x)_{ReLU} = max(0, x). \tag{2.5}$$

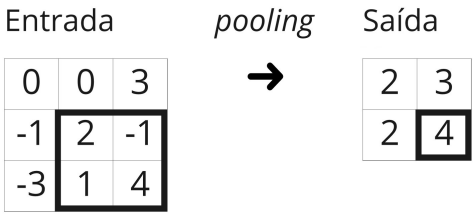
No entanto, a função ReLU possui um problema onde os neurônios "morrem", isto é, eles passam apenas a resultar no valor 0. Uma alternativa para evitar este problema é a função *Leaky* ReLU, do qual não ignora os valores negativos, mas sim utiliza um fator denotado por m:

$$f(x)_{LeakyReLU} = \begin{cases} x, \text{se } x > 0 \\ mx, \text{se } x \geq 0 \end{cases} \tag{2.6}$$

onde m é geralmente um valor muito próximo de 0 (ALZUBAIDI *et al.*, 2021; AGGARWAL *et al.*, 2018). Quando se deseja representar uma distribuição de probabilidades sobre uma variável com n valores possíveis, a função de ativação *Softmax* deve ser utilizada (GOODFELLOW; BENGIO; COURVILLE, 2016). *Softmax* é normalmente utilizada no final da rede para gerar as saídas de um classificador.

A camada *Pooling* é responsável por gerar uma subamostragem com as informações mais dominantes dos processos anteriores (ALZUBAIDI *et al.*, 2021; GOODFELLOW; BENGIO; COURVILLE, 2016). Assim como a camada de convolução, possui os parâmetros do tamanho do *kernel* e *stride*. Existem diversas modelos de *pooling* para serem aplicados, dentre todos, o mais típico é o *max-pooling* (AGGARWAL *et al.*, 2018). O *max-pooling* consiste em selecionar o maior valor dentre todos os valores num determinado retângulo, conforme ilustra a Figura 19.

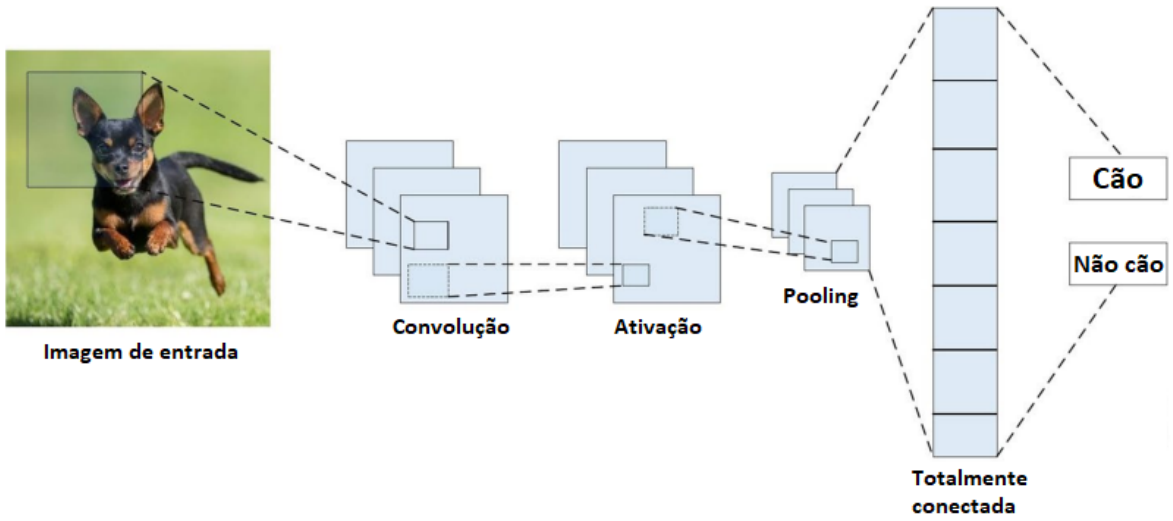
Figura 19 – Camada de *max-pooling* com um kernal de 2x2 e *stride* de 1.



Fonte: Adaptado de (ALZUBAIDI *et al.*, 2021).

A rede é completa por uma ou mais camadas totalmente conectadas (*fully connected layer*), camada que funciona exatamente da mesma forma que um FFNN. Cada neurônio da última camada com dados espaciais é conectada com cada neurônio da camada totalmente conectada, para por fim ser utilizada como o classificador da rede CNN (Figura 20).

Figura 20 – Arquitetura de uma rede CNN.



Fonte: Adaptado de (ALZUBAIDI *et al.*, 2021).

3 PROPOSTA DE UM SISTEMA DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICA

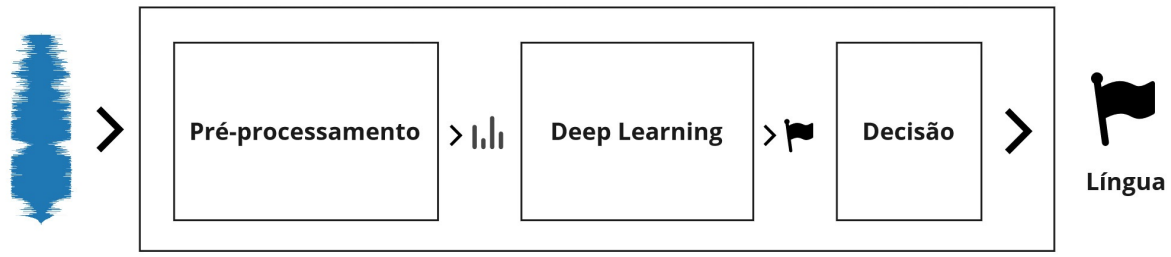
Este capítulo apresenta a proposta e avaliação de um sistema identificador de língua em música. A Seção 3.1 descreve o sistema proposto, detalhando a abordagem escolhida, as etapas de pré-processamento, as características e o modelo de classificação. A Seção 3.2 apresenta o sistema *baseline* proposto para fins de comparação. A Seção 3.3 descreve a base de dados utilizada para treinar e avaliar o modelo proposto. Finalmente, a Seção 3.4 apresenta as configurações dos experimentos e os resultados obtidos.

3.1 SISTEMA PROPOSTO

A maioria dos sistemas de identificação de língua em música utilizam características que imitam o processo humano de audição. Métodos convencionais usados para fazer a transformação de características resultam em características estáticas, como MFCC e PLP, onde não há garantia que sejam as melhores representações para a identificação de língua, pois foram criadas para reconhecer o conteúdo da mensagem falada (DAVIS; MERMELSTEIN, 1980; HERMANSKY, 1990).

Visando fornecer uma representação apropriada para identificar língua em músicas, este trabalho propõem a utilização de uma rede neural profunda para extração de características relevantes para a identificação da língua. O sistema é formado por 3 etapas (Figura 21):

Figura 21 – Sistema proposto.



Fonte: O Autor (2022).

1. **Pré-processamento**: visa remover elementos da música que não contribuem para a identificação da língua.

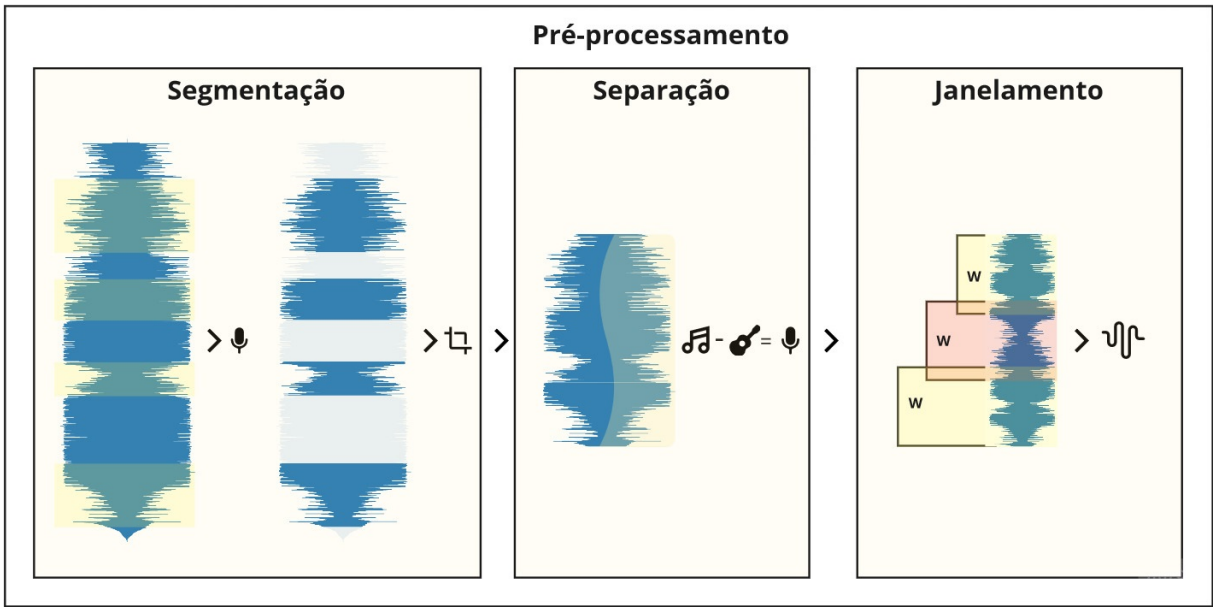
2. ***Deep Learning***: tem por objetivo transformar o sinal da música em características significativas para a discriminação das línguas e identificar a língua da música.

3. **Decisão**: tem por objetivo produzir a decisão da língua identificada pelo sistema. Como a identificação é realizada por segmento de música, a lógica do voto da maioria é aplicada para determinar a língua da música (JAMES, 1998).

3.1.1 Pré-processamento

O pré-processamento produz segmentos de áudio a partir de um sinal de música. Nesta etapa, informações não relevantes para a tarefa de identificação de língua são removidas. O pré-processamento é dividido em 3 etapas executadas em sequência: segmentação, separação e janelamento (Figura 22).

Figura 22 – Pré-processamento proposto.



Fonte: O Autor (2022).

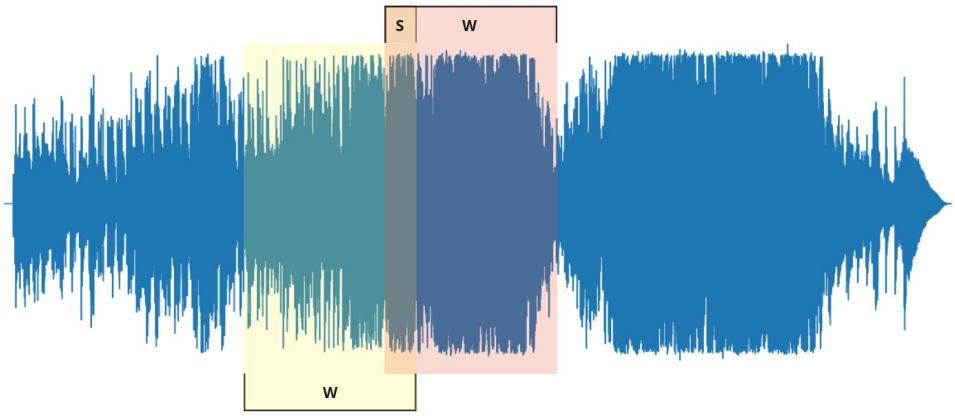
A segmentação consiste em detectar trechos do sinal de áudio com atividade vocal e remover os segmentos que não contém tal atividade. Para detectar os segmentos com atividade vocal, foi utilizado o modelo *Pyannote* (BREDIN; LAURENT, 2021), que identifica trechos com atividades vocais e as retorna de forma segmentada por tempo. O modelo *Pyannote* aplica a segmentação com a combinação de detecção de atividade vocal, detecção de mudança de locutor e detecção de sobreposição de locução numa abordagem de treinamento *end-to-end*. Igual ao sistema proposto, o modelo *Pyannote* utiliza a rede *SincNet* para transformar características relevantes. Experimentos mostraram que o modelo produz resultados superiores em comparação a outros modelos treinados para detecção de atividade de voz (BREDIN; LAURENT, 2021). A partir dos trechos segmentados, todas as partes que não possuem voz presente foram removidas, gerando um novo sinal de áudio.

A separação visa isolar a voz dos demais sons no sinal de áudio. O modelo *Demucs* (DÉFOSSEZ, 2021) foi utilizado para separar a voz cantante dos demais sons presentes em mú-

sica, gerando um sinal de áudio de mesma duração, mas com apenas a voz cantante. O modelo *Demucs* é uma extensão da arquitetura *Wave-U-Net* (STOLLER; EWERT; DIXON, 2018) que visa separar o sinal de áudio diretamente no domínio de tempo para considerar os contextos temporais do sinal. *Demucs* expande as capacidades de análise e predição com mais domínios, possuindo além do domínio temporal, um domínio com espectro e os dois domínios juntos (DÉFOSSEZ, 2021).

Por fim é aplicado o janelamento, onde o sinal de áudio é separado em pedaços de tamanho w, com uma sobreposição s, conforme ilustra a Figura 23. Estes segmentos de áudio são utilizados como entrada da rede neural profunda para a extração de característica e identificação da língua.

Figura 23 – Janelamento.

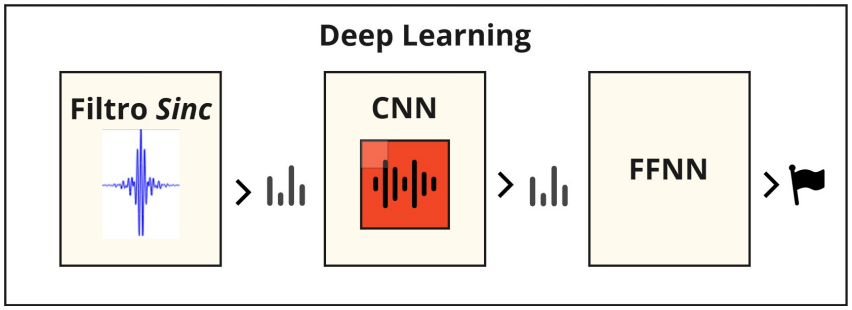


Fonte: O Autor (2022).

3.1.2 *Deep Learning*

Nesta etapa, o sinal produzido pelo pré-processamento é transformado em uma representação significativa para a tarefa de identificação da língua, utilizando redes neurais profundas. O sistema utiliza filtros *Sincs* para a transformação das características e aplica as redes CNN e FFNN para identificar a língua da música, conforme ilustrado na Figura 24.

Figura 24 – Arquitetura *Deep Learning* proposta.



Fonte: O Autor (2022).

Com base em uma rede CNN, a primeira camada da rede aplica filtros, denominados *Sinc*, que extraem informações significativas do sinal de áudio (FAINBERG *et al.*, 2019; RAVANELLI; BENGIO, 2018; TRIPATHI; SINGH; SUSAN, 2020). Como os filtros compõem a rede, eles são estimados ao decorrer do aprendizado da rede, para poderem extrair características dado o objetivo do sistema (Figura 25). A rede é denominada *SincNet* e tem produzido excelentes resultados em tarefas de reconhecimento de locutor (RAVANELLI; BENGIO, 2018; PARCOLLET; MORCHID; LINARèS, 2020), diarização de locutor (DUBEY; SANGWAN; HANSEN, 2019) e reconhecimento de emoções (MAYOR-TORRES *et al.*, 2021).

A rede *SincNet* convolve o sinal de áudio com um conjunto de funções parametrizadas, que implementam um filtro, onde as frequências de corte alta e baixa são os únicos parâmetros que a rede aprende. Em sua primeira camada aplica um conjunto de convoluções com uma função g pré-definida que depende em poucos parâmetros θ que a rede deve aprender:

$$y[n] = x[n] \times g[n, \theta], \tag{3.1}$$

onde $x[n]$ é um segmento do sinal de áudio e $y[n]$ é o resultado do filtro. A função g é composta por dois filtros de banda retangulares:

$$G[f, f_1, f_2] = rect(\frac{f}{2f_2}) - rect(\frac{f}{2f_1}), \tag{3.2}$$

onde $rect(.)$ é a função retangular, f_1 e f_2 são as frequências de corte alto e baixo a serem aprendidas. Para retornar para o domínio de tempo, é aplicada a transformada de Fourier inversa Rabiner e Schafer (2010), resultando na função g:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n), \tag{3.3}$$

onde a função $sinc()$ é definida por $sinc(x) = \sin(x)/x$. Por fim, para suavizar a descontinuidade em direção aos limites do sinal, a função g é ponderada pela função w:

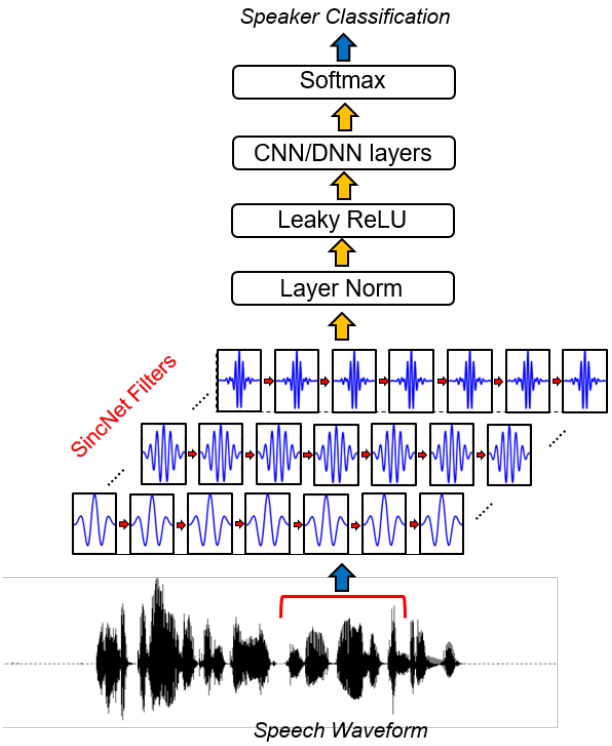
$$w[n] = 0.54 - 0.46 \cos(\frac{2\pi n}{L}), \tag{3.4}$$

que caracteriza o uso do janelamento Hamming (SHENOI, 2005). Assim, o sinal filtrado resulta em:

$$y[n] = x[n] \times g_w[n, f_1, f_2] = x[n] \times g[n, f_1, f_2] \times w[n]. \tag{3.5}$$

O uso da rede *SincNet* implica em filtros mais robustos e com rápida convergência, dado que a rede foca em parâmetros do filtro que tenham maior impacto (frequências de corte alta e baixa). A Figura 26 ilustra filtros aprendidos por uma rede CNN tradicional e uma rede *SincNet*. É possível observar o impacto dos filtros com frequências de corte alta e baixa. O resultado a ser passado para as redes subsequentes da rede DL é o sinal de áudio unidimensional filtrado em representação espacial, do qual foram extraídas as características mais relevantes para o modelo, dado o resultado esperado.

Figura 25 – Rede SincNet.



Fonte: Adaptado de (RAVANELLI; BENGIO, 2018).

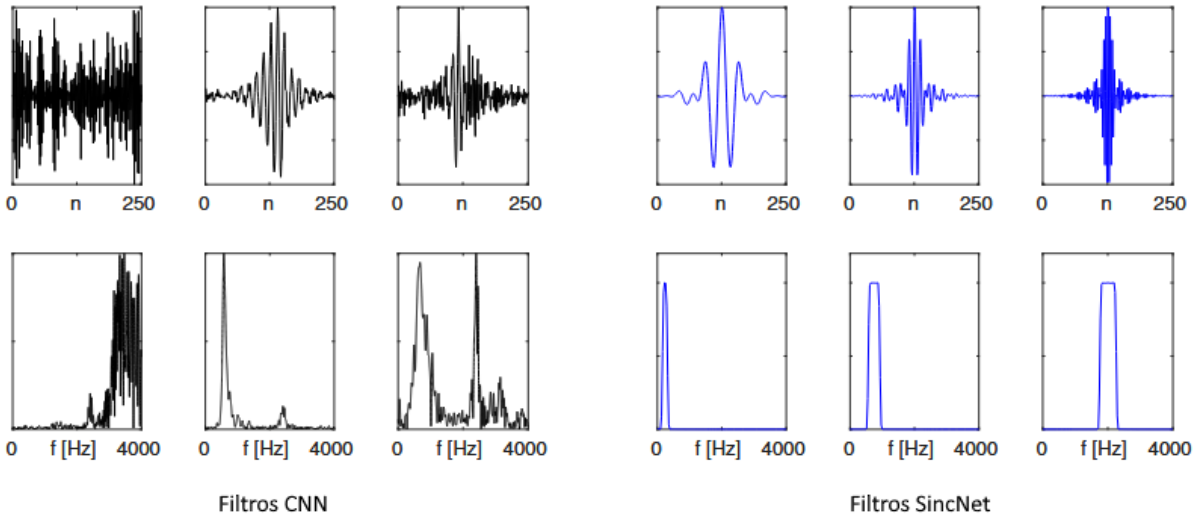
A rede CNN e FFNN são responsáveis por classificar a língua. Como a rede CNN recebe dados distribuídos de forma espacial, o sinal de áudio unidimensional é distribuído em intervalos de tempo. Cada neurônio dos dados espaciais é conectada com a rede FFNN para ser feita a classificação e gerar os erros do aprendizado da rede.

3.2 SISTEMA *BASELINE*

Como o objetivo é avaliar o uso de uma rede neural profunda que capture e aprenda informações relevantes da língua, o sistema proposto foi comparado com um sistema *baseline* que utiliza características estáticas. Com isto é possível entender o impacto gerado por uma rede neural que se adapta e busca aprender quais as características do sinal de áudio são mais relevantes dado o objetivo final do sistema.

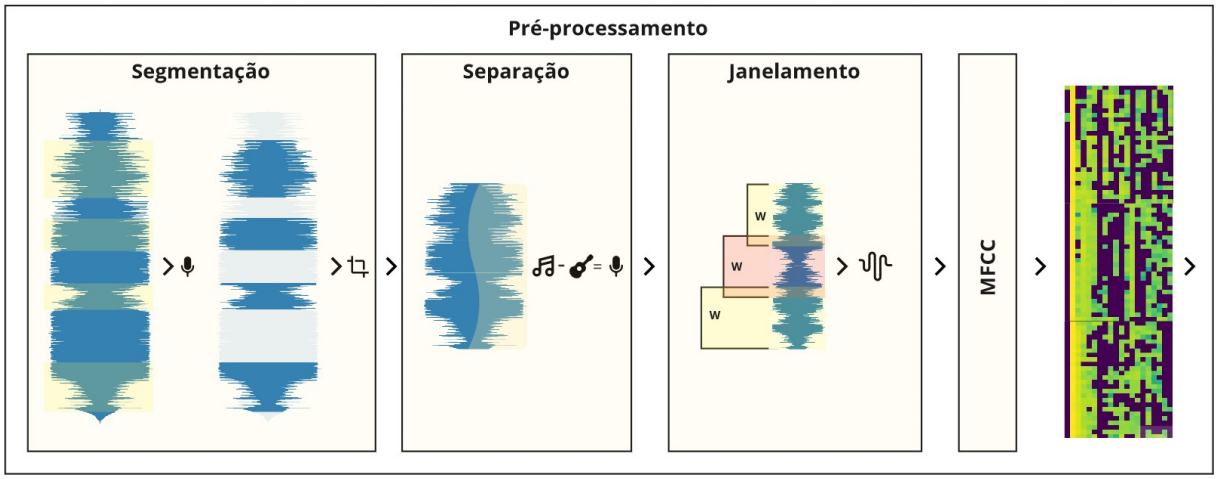
O sistema *baseline* é dividido em três etapas: pré-processamento, rede DL e decisão, assim como o sistema proposto. No entanto, como faz a extração de características estáticas, que não variam ao decorrer do aprendizado da rede, esta extração é feita na etapa de pré-processamento com o método MFCC (DAVIS; MERMELSTEIN, 1980). Portanto, a etapa de pré-processamento realiza as operações de segmentação, separação, janelamento e extração de características estáticas (Figura 27). O MFCC irá produzir um espectro bidimensional que será utilizado para alimentar a rede CNN. Diferentemente do sistema proposto, a rede DL não irá aplicar a *SincNet*, apenas CNN e FFNN (Figura 28).

Figura 26 – Exemplos de filtros aprendidos por uma rede CNN tradicional e uma rede SincNet. A primeira linha de gráficos apresenta os filtros no domínio tempo. A segunda linha mostra a resposta em frequência dos respectivos filtros.



Fonte: Adaptado de (RAVANELLI; BENGIO, 2018).

Figura 27 – Pré-processamento *baseline*.

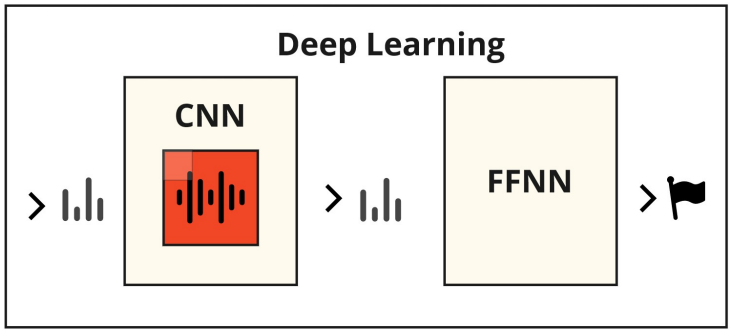


Fonte: O Autor (2022).

O MFCC é uma das características mais utilizados em aplicações de processamento de áudio para identificação de voz e língua (AMBIKAIRAJAH *et al.*, 2011). Transforma propriedades acústicas do sinal de áudio em coeficientes *cepstrais*, uma representação que aproxima a frequência de percepção do ouvido humano. O primeiro passo é aplicar a transformada de *Fourier* de curto prazo (*Short-Term Fourier Transform* (STFT)), resultando em um espectro da escala de frequência linear. O processo de extração é realizado da seguinte forma (AMBIKAIRAJAH *et al.*, 2011; KRUSPE, 2018; BENESTY *et al.*, 2008):

1. STFT: o sinal s é segmentado em segmentos s_i, cada segmento é calculado pelo janelamento Hamming para reduzir a descontinuidade nos limites dos segmentos (DELLER;

Figura 28 – Arquitetura *Deep Learning baseline*.



Fonte: O Autor (2022).

PROAKIS; HANSEN, 1993). A transformada de *Fourier* discreta (*Discrete Fourier Transform* (DFT)) é calculada para cada segmento (ALLEN, 1977; KRUSPE, 2018):

$$S_i(k) = \sum_{n=0}^{N-1} s_i(n)h(n)e^{-j2\pi kn/N}, k = 0, 1, ..., K-1 \tag{3.6}$$

onde $h(n)$ é a janela de tamanho N e K é o tamanho da transformada de *Fourier*.

2. Filtro Mel: após aplicado o STFT, o espectro é convertido para uma escala de frequência não linear, motivado pela frequência de percepção do ouvido humano. É feito aplicando convoluções no espectro com um conjunto M de filtros triangulares, espaçados conforme a escala Mel (DELLER; PROAKIS; HANSEN, 1993; ADAMI, 2004; KRUSPE, 2018), como ilustrado na Figura 29. Após aplicado o filtro, o espectrograma de frequência Mel é calculado sobre uma função logarítmica.

3. *Discrete Cosine Transform* (DCT): por fim, o espectrograma de frequência Mel é projetado em uma base de cosseno discreto com a DCT:

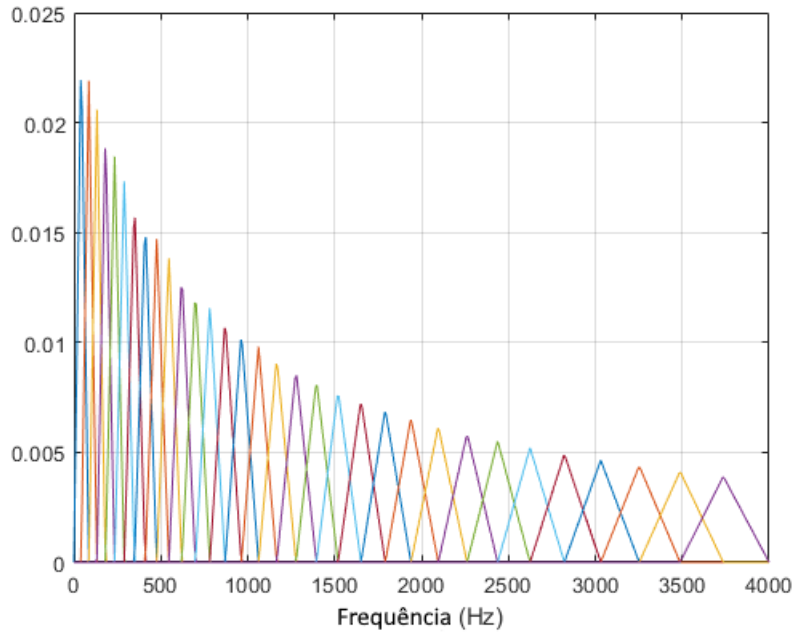
$$C_j = \sum_{m=1}^{M} X_m \times \cos\left(j+1\right) \times (m - \frac{1}{2}) \times \frac{\pi}{M}, j = 0, 1, ..., J-1 \tag{3.7}$$

onde M é o número de filtros Mel e J é o número de coeficientes cepstrais. Resultando no *cepstrum*, um espectro bidimensional (ADAMI, 2004; KRUSPE, 2018).

3.3 BASE DE DADOS

A base de dados possui 3 línguas: português, espanhol e italiano, todas originárias da língua latina. Possui 3600 músicas, divididas de forma balanceada para cada língua, 1200 cada. Para isto, a base foi construída seguindo o método adotado por Groenbroek (2021), que visa criar uma base de dados balanceada. Foi utilizada a Web API gratuita do Spotify para buscar dados de músicas e baixá-las do Youtube utilizando a ferramenta *spotdl*.

Figura 29 – Exemplo de filtro Mel.



Fonte: Adaptado de (BENESTY *et al.*, 2008).

Dentre os dados das músicas obtidos da API do Spotify, a língua não é uma das informações. Portanto, para obter músicas de uma língua específica a busca deve ser refinada em *playlists* previamente criadas que contenham músicas de uma determinada língua, para então obter o título e artista das músicas, e fazer o download do áudio da mesma no Youtube (Figura 30).

Figura 30 – Diagrama do modelo de construção da base de dados.

Fonte: Adaptado de (GROENBROEK, 2021).

Como o Spotify não fornece a informação da língua, o método utilizado por Groenbroek (2021) assume que:

1. Os resultados da pesquisa na API do Spotify são representativas da consulta.
2. As músicas presentes em *playlists* criadas pelos usuários são representativas do título da *playlist*. Por exemplo, uma *playlist* intitulada "Músicas brasileiras"terá apenas músicas da língua portuguesa.
3. A ferramenta *spotdl* consegue encontrar e baixar o áudio correto do Youtube a partir do título e artista obtidos a partir da API do Spotify.

4. Erros ocasionados pelas suposições anteriores não afetam a qualidade da base de dados o suficiente para impactar no desempenho do sistema proposto.

No entanto, não há garantia de que elas tenham apenas músicas representativas dado o título utilizado, podendo gerar uma base com diversas músicas com a classificação incorreta. Para mitigar este problema, a base foi construída utilizando a Web API gratuita do Last.fm. Da mesma forma que a API do Spotify, dentre os dados obtidos, a língua não é uma das informações. Porém, as músicas são classificadas por tags, como: gênero, estilo musical, país e língua, permitindo assim que as músicas sejam filtradas e classificadas utilizando uma ou mais tags adicionadas.

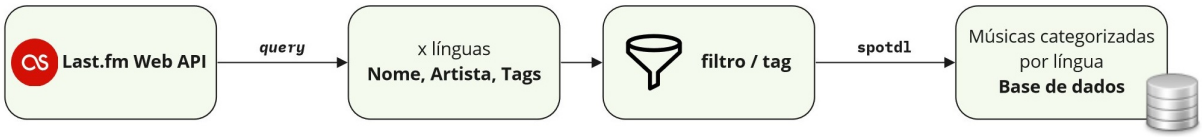
Com o intuito de construir uma base robusta e representativa para um sistema SLID, alguns requerimentos foram estabelecidos:

1. Os dados devem estar disponíveis, precisa existir uma fonte para obter uma quantidade suficiente de músicas cantadas nas línguas definidas.
2. As amostras devem comportar diferentes gêneros musicais.
3. Um artista não deve ter uma amostragem desbalanceada aos demais.
4. O número de amostragens por língua devem ser balanceada, todas com uma quantidade muito próxima ou igual.
5. Os dados devem possuir uma qualidade alta e constante entre todas as amostras.

Dado os requerimentos estabelecidos e a possibilidade de buscar e filtrar as músicas por tags, a montagem da base de dados foi construída com os seguintes passos (Figura 31):

1. Buscar na Web API gratuita do Last.fm por tag, onde a tag seja ou se relacione a língua cantada esperada.
2. Dentre todas as amostras obtidas, buscar todas as tags referentes a cada uma e remover as que tenham tags relacionadas a outra língua ou que impliquem que a música não é adequada para a base. Por exemplo, músicas da tag "Brasil"que tenham a tag "Espanhol"ou músicas que tenham a tag "Instrumental".
3. Limitar a quantidade de músicas por artistas presentes na base de dados, visando balanceada a base.
4. Utilizar a ferramenta *spotdl* para baixar o dado de áudio.

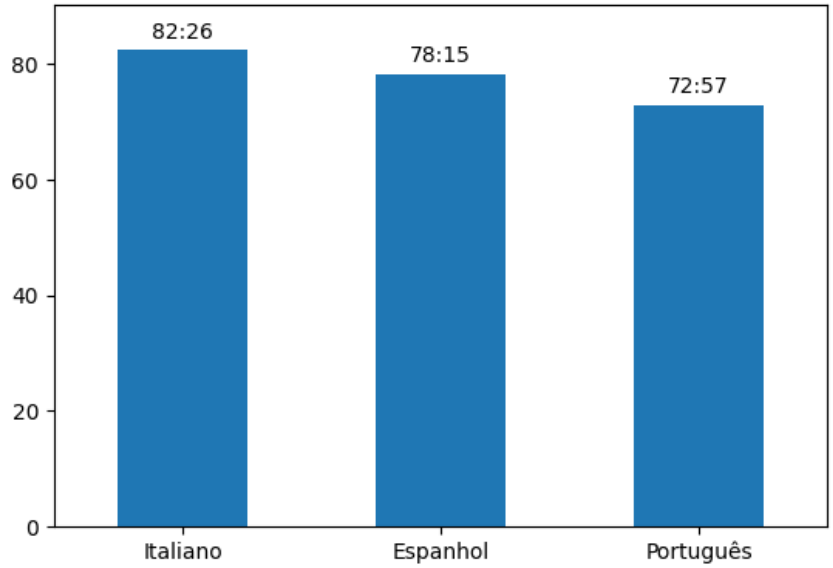
Figura 31 – Diagrama de construção da base de dado.



Fonte: O Autor (2022).

A base totaliza 233 horas e 38 minutos de música. A Figura 32 apresenta o tempo total de cada língua. Dentre todas as músicas, em sua maioria, possuem entre 3 e 5 minutos de duração, e alguns casos atípicos com duração excedendo 10 minutos ou inferior a 1 minuto (Figura 33). A quantidade de músicas por artistas foi limitada para no máximo 10, resultando que cada língua possuam ao menos 400 artistas, conforme ilustra a Figura 34.

Figura 32 – Tempo total (HH:mm) de música por língua.



Fonte: O Autor (2022).

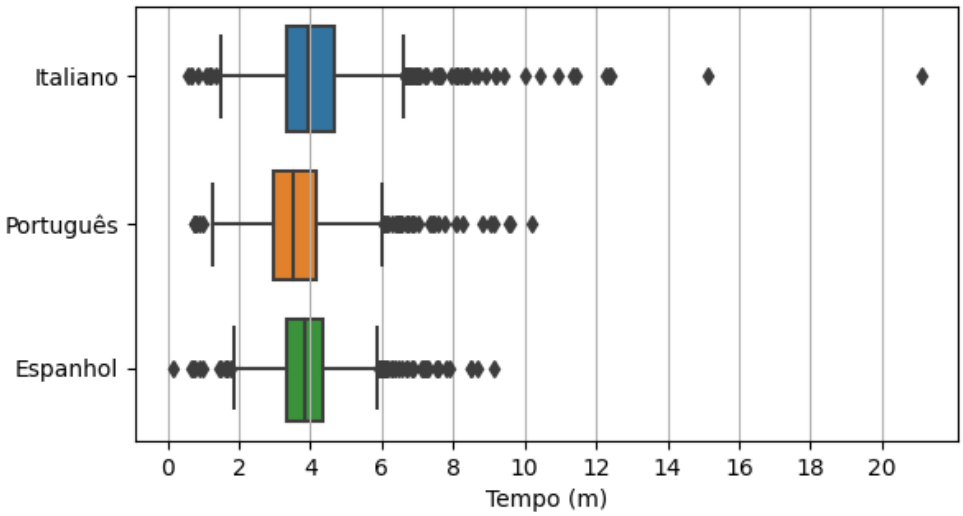
3.4 EXPERIMENTAÇÃO DO SISTEMA DE IDENTIFICAÇÃO DE LÍNGUA EM MÚSICA

A experimentação foi aplicada igualmente no sistema proposto e *baseline* com a mesma base de dados. Ambos os sistemas realizaram identificação em conjunto fechado e recebiam como dados de entrada o áudio bruto. Em todos os experimentos, foi utilizado $2/3$ da base para treinamento dos modelos e $1/3$ para teste.

3.4.1 Configuração Experimental

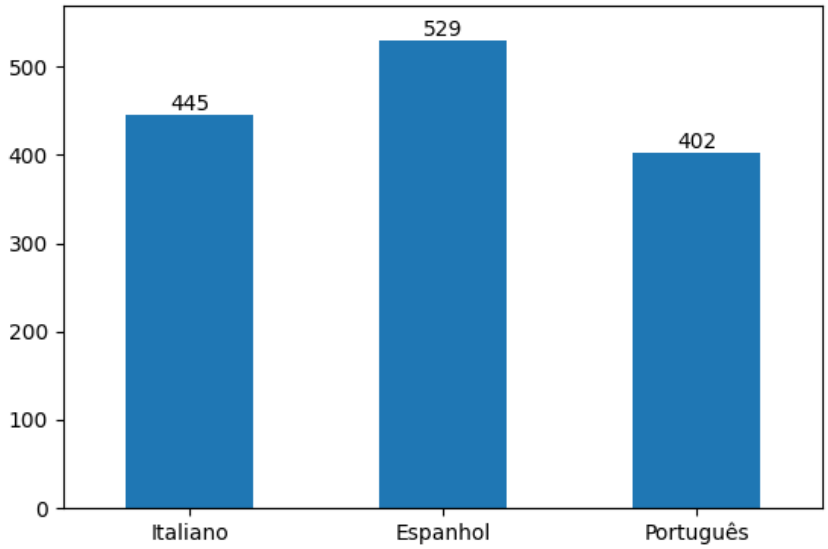
Para a execução da experimentação, os sistemas utilizaram configurações semelhantes. Em ambos os sistemas foram utilizados janelamentos de $200ms$, uma sobreposição de $10ms$,

Figura 33 – Duração das músicas em minutos.



Fonte: O Autor (2022).

Figura 34 – Quantidade de artistas por língua.



Fonte: O Autor (2022).

com 1200 lotes de 128 instâncias para cada época treinada. Todas as camadas de convolução aplicam um *stride* e *padding* de tamanho 1. Após cada convolução é aplicada uma camada de ativação. As camadas de ativação utilizam a função de ativação *Leaky* ReLU, para evitar que existam neurônios com valores zerados. A última camada de toda a rede é uma camada de ativação com a função *Softmax*.

Os dois sistemas aplicam segmentação e separação na etapa de pré-processamento. Para fins de análise de impacto do pré-processamento no desempenho do sistema, foram realizados experimentos com o sinal de áudio completo, isto é, sem remover informações irrelevantes do sinal de áudio e sem separar a voz dos demais sons. Assim, é possível verificar o quanto cada um destes processamentos auxilia na identificação da língua.

O sistema proposto aplica uma rede DL, onde sua primeira camada deve fazer a transformação de características relevantes dado o objetivo final de toda a rede. Para isso utiliza filtros *Sinc* na primeira camada da rede CNN, seguida por uma FFNN. A rede CNN possui 3 camadas de convolução. A primeira obrigatoriamente aplica o filtro *Sinc*, possui como entrada apenas 1 canal e resulta em 80 canais após a convolução, com um *kernel* de tamanho 251. As próximas duas camadas de convolução não aplicam o filtro e possuem ambas *kernel* de tamanho 5 e resultam em canais de tamanho 60. Cada camada de convolução possui uma camada de *max-pooling* em seguida, todas com um *kernel* de tamanho 3 e *stride* de tamanho 1.

O sistema *baseline* aplica o método MFCC para fazer a extração de características estáticas, passadas para a rede DL. Uma rede CNN seguida por uma FFNN. O MFCC é configurado para extrair 20 coeficientes *cepstrais*, com janelamento da transformada de *Fourier* de tamanho 256 e aplicando 48 filtros Mel. A rede CNN possui 3 camadas de convolução, todas com *kernel* de tamanho 3. Os números de canais das camadas de convolução são, respectivamente, 1 canal, 30 canais e por fim 60 canais. Cada camada de convolução possui uma camada de *max-pooling* em seguida, todas com um *kernel* de tamanho 2 e *stride* de tamanho 1.

O Quadro 2 apresenta todos os modelos estabelecidos para fazer a experimentação do sistema de identificação de língua em música. Para cada sistema, proposto e *baseline*, 3 configurações possíveis: completo com segmentação e separação, apenas segmentação e sem nenhum tratamento contra ruídos e informações irrelevantes no sinal de áudio.

Quadro 2 – Modelos de experimentação.

Modelo	Segmentação	Separação	Janelamento	MFCC	Filtro Sinc	CNN / FFNN
Proposto	X	X	X		X	X
	X		X		X	X
			X		X	X
Baseline	X	X	X	X		X
	X		X	X		X
			X	X		X

Fonte: O Autor (2022).

3.4.2 Resultados

Os resultados foram obtidos ao executar por 50 épocas todos os modelos de experimentação. A Tabela 4 apresenta a relação das acurácias obtidas a cada 10 épocas para a configuração sem etapas de separação e segmentação nos sistemas proposto e *baseline*. Em ambos os sistemas, a melhor acurácia foi obtida na época 50. Com base na tabela, é possível identificar que o uso de características significativas impactam positivamente na identificação de língua em música em comparação à características estáticas. O sistema proposto obteve um desempenho aproximadamente 12% melhor que o sistema *baseline* de mesma configuração.

A remoção dos segmentos de áudio que não contém voz contante não trouxe uma melhora significativa no desempenho da identificação, somente vantagem na quantidade de pro-

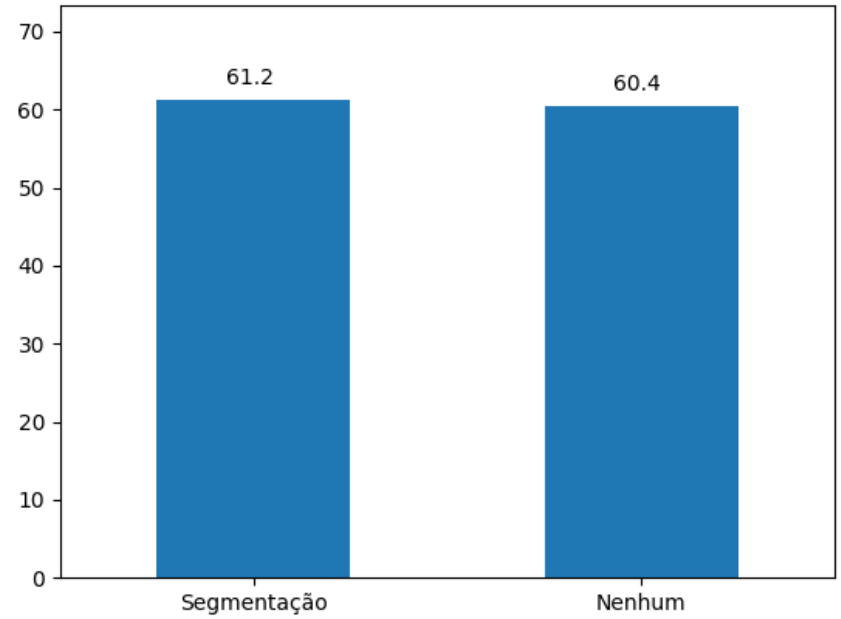
Tabela 4 – Tabela de acurácia (%) por época para os sistemas proposto e *baseline* sem aplicação de separação e segmentação.

Épocas	Sistema proposto	Sistema *baseline*
10	50.9	44.3
20	55.3	50.6
30	56.1	47.9
40	58.2	48.8
50	**60.4**	**53.8**

Fonte: O Autor (2022).

cessamento. Conforme ilustrado os desempenhos do sistema proposto com e sem segmentação na Figura 35, a diferença de desempenho relativa é de apenas $1,3\%$. É possível especular que as redes estão aprendendo a ignorar as informações dos segmentos que não possuem voz cantantes. Entretanto, o resultado mostra que os segmentos que não possuem voz cantante não afetam o desempenho da identificação e que a sua remoção resulta em menos demanda por processamento computacional. A Figura 36 apresenta a duração em minutos das músicas com e sem a segmentação para cada língua. A redução de dados é em torno de 64%, reduzindo também o tempo de processamento.

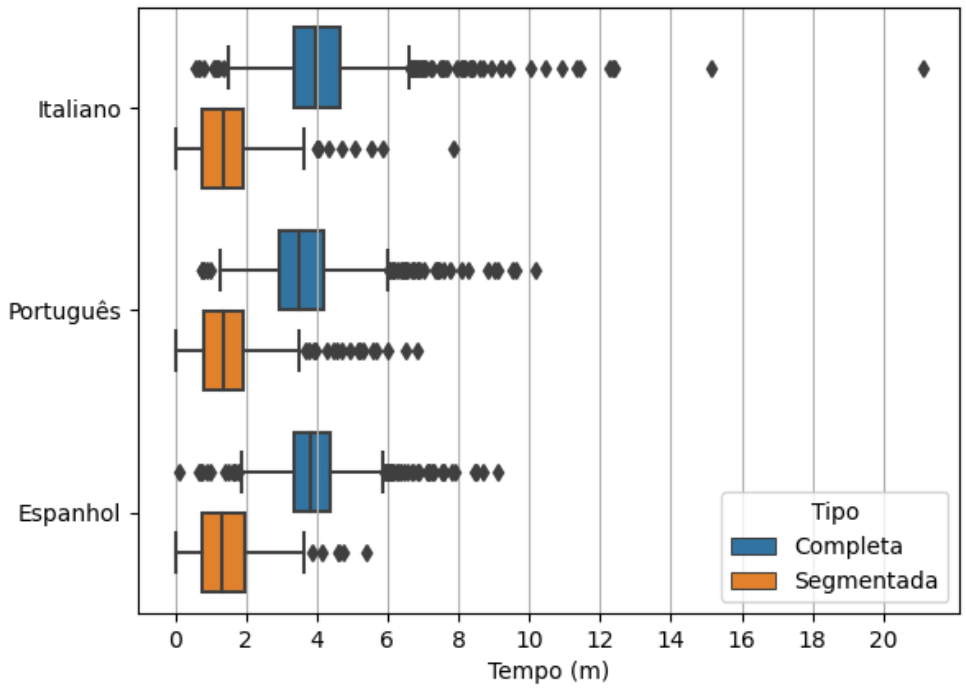
Figura 35 – Acurácias do sistema proposto com e sem a segmentação no pré-processamento.



Fonte: O Autor (2022).

Ao utilizar o sistema proposto completo com todas as etapas de pré-processamento, segmentação e separação, foi obtido o melhor desempenho, acurácia de $68,5\%$. Além de ser realizada a segmentação, é aplicada a separação, isto é, remover os ruídos do sinal de áudio mantendo apenas a voz cantante. Conforme a Figura 37 apresente, o desempenho do sistema com segmentação e separação foi superior em aproximadamente 12% ao sistema com apenas

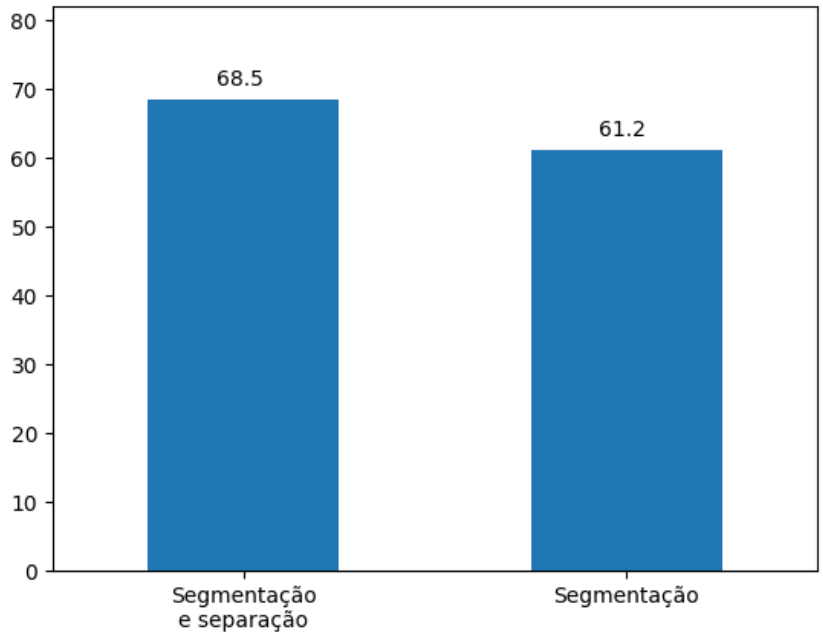
Figura 36 – Duração das músicas completas e segmentadas (sem os segmentos que não possuem voz cantante) em minutos



Fonte: O Autor (2022).

segmentação. Isto mostra o impacto que o ruído, como som instrumental, causa na tarefa de identificação de língua em música.

Figura 37 – Acurácias do sistema proposto com segmentação e separação.

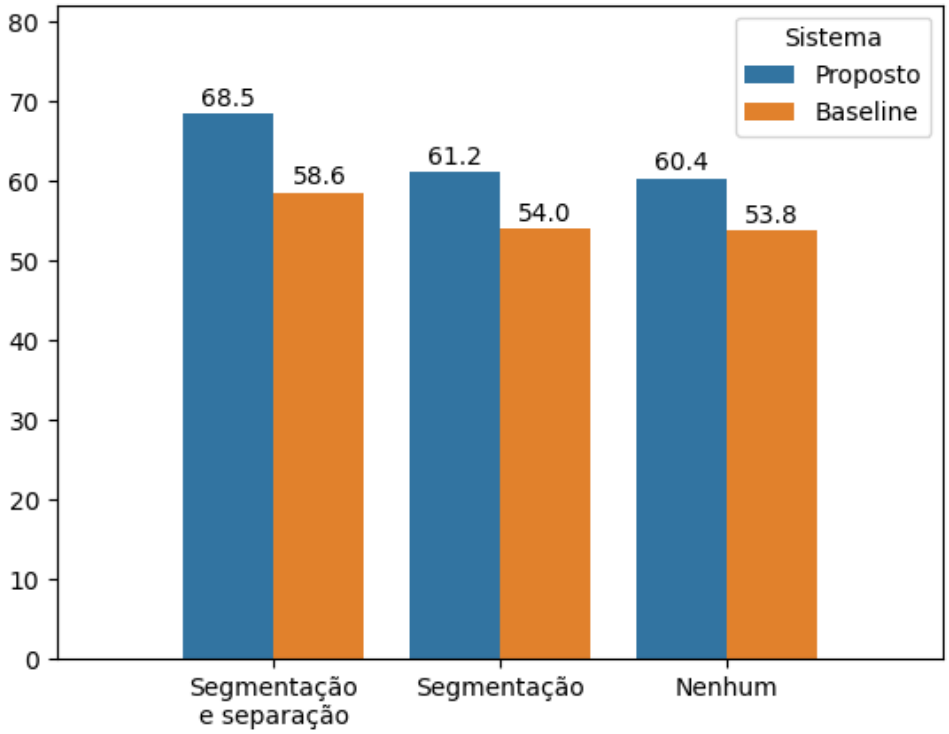


Fonte: O Autor (2022).

As mesmas ponderações para o sistema proposto se repetem no sistema *baseline*, ao

analisarmos as configurações de pré-processamento. A Figura 38 apresenta os resultados de ambos os sistemas, proposto e *baseline*, para todas as configurações estabelecidas. Em todas as configurações o sistema proposto obteve um melhor desempenho, principalmente na configuração completa com segmentação e separação, onde obteve um desempenho superior em 16.9% relativo ao sistema *baseline*.

Figura 38 – Acurácias por sistema e configurações estabelecidas.



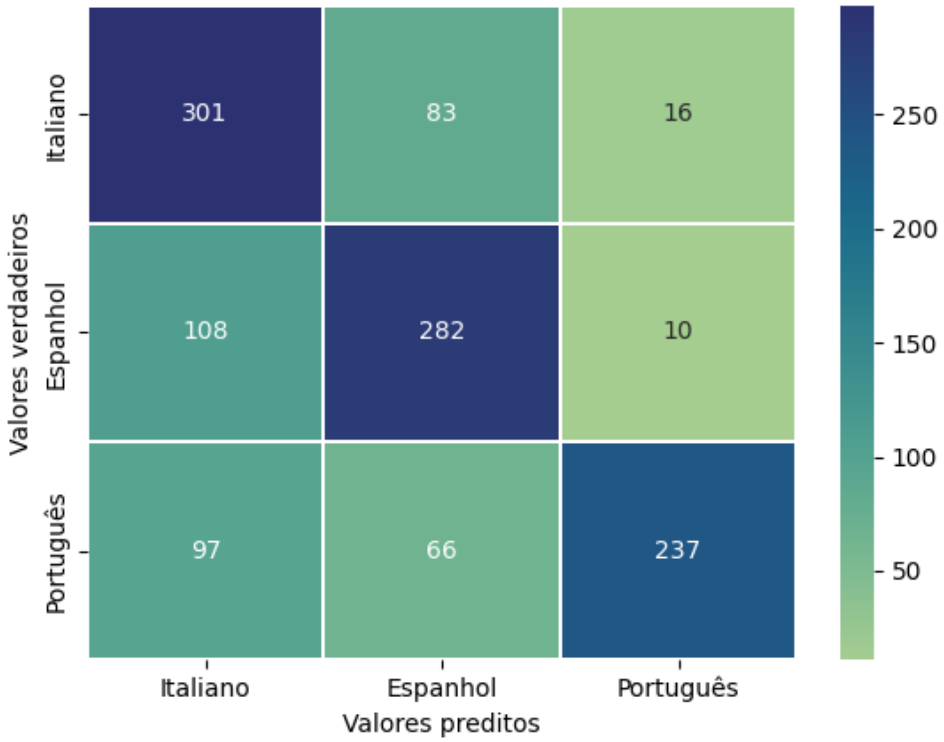
Fonte: O Autor (2022).

Dentre as línguas utilizadas para testar o sistema, a que teve o maior número de classificações corretas foi a língua italiana. A Figura 39 apresenta a matriz de confusão do sistema proposto, onde é possível notar que, apesar de italiano ter sido a língua com mais classificações corretas, foi também a que teve a maior quantidade de falso positivos (Seção 2.3.2). Isto mostra uma tendência do sistema em classificar línguas de origem latina, como a língua italiana, onde aproximadamente 42% de toda base de teste foi classificada como língua italiana.

O sistema proposto, em comparação aos trabalhos de SLID utilizando DL apresentados na Seção 2.4.2, obteve uma acurácia superior que os trabalhos de Iglesias (2020) e Groenbroek (2021). Ambos os trabalhos utilizaram características estáticas para fazer a identificação de língua, tendo dentre eles a melhor acurácia de 42%. Portanto, o sistema proposto apresentou um desempenho que é no mínimo 63% melhor que ambos. Porém, os trabalhos possuem base de dados com uma quantidade superior de línguas, fator que pode ter impactado no desempenho.

Dentre os três trabalhos de SLID utilizando DL, apenas o trabalho de Renault, Vaglio e Hennequin (2021) obteve uma acurácia superior de 91.74%. Diferente dos demais trabalhos,

Figura 39 – Matriz de confusão sistema proposto.



Fonte: O Autor (2022).

eles utilizaram características estatísticas das músicas com base na ocorrência de fonemas para fazer a identificação. O sistema utiliza uma rede para estimar os fonemas, os quais são utilizados posteriormente para fazer a identificação da língua. Isto implica que para treinar o modelo é necessário de um reconhecedor de fonemas da língua de interesse, além das letras de música para estimar tal reconhecedor, diferente do sistema proposto onde apenas o sinal de áudio é necessário.

Os trabalhos que utilizam o modelo tradicional para fazer a identificação de línguas, apresentados na Seção 2.4.1, possuem desempenho entre 58% e 98.62%. O trabalho com desempenho mais baixo, de Kruspe e Fraunhofer (2014), utilizou características estáticas com a aplicação do método MFCC para extração de características. O trabalho com desempenho mais alto, de Mukherjee *et al.* (2021), utilizou características estáticas LSF com um classificador *Random Forest*. O desempenho do trabalho de Mukherjee *et al.* (2021) foi superior ao do sistema proposto em aproximadamente 44%. No entanto, é importante considerar que as bases utilizadas, apesar da mesma quantidade de línguas (3), são bem diferentes. Mukherjee *et al.* (2021) utilizou as línguas: inglês, bangla e hindu, onde apenas duas são de mesma origem, enquanto na base utilizada para testar o sistema proposto, todas são originárias da língua latina. Línguas de origens diferentes tendem a ter maior facilidade de discriminação do que línguas de mesma origem (ADAMI, 2004).

4 CONCLUSÕES

É notável a constante evolução que a indústria da música vivencia. O modo como músicas são e tem sido consumidas mudou drasticamente num intervalo de tempo relativamente curto, houve um tempo que para ser músico era preciso de uma gravadora, especializada e preparada para gravar e distribuir de forma eficaz e rentável. Hoje, com um aplicativo como Garage Band, criatividade e uma plataforma como o Youtube, é possível ser ouvido no mundo todo em alguns minutos.

Assim como publicar músicas se tornou algo mais fácil e palpável, ouvir músicas se tornou algo mais viável, disponível e atemporal. Plataformas como Youtube, Spotify e Deezer disponibilizam ferramentas onde com uma *internet* razoável é possível ouvir a sua música favorita repetidas vezes, no carro, em casa, na rua e academia, sem ter que depender da programação de uma rádio, ou ter que comprar um disco ou CD para tocar no seu aparelho em casa.

Com toda essa facilidade e demanda de produzir e ouvir, a categorização de músicas se tornou uma área a ser explorada para ajudar tanto quem produz, quanto quem quer apenas ouvir. Dentre as diversas categorizações possíveis de se fazer com música, uma delas é a categorização por língua. Hoje vivemos num mundo globalizado e a língua é uma barreira que pode que ser quebrada com o auxílio de ferramentas especializadas. A categorização de músicas por língua é uma etapa primordial para a tradução e transcrição das músicas, fazendo com que possam ser compreendidas por diversas pessoas, pois músicas em muitos contextos são mais que um amontoado de sons, possuem significado e querem passar um sentimento.

Dado esse contexto, este trabalho desenvolveu um identificador de língua em música. Para a definição do sistema proposto, foi feito o estudo dos meios de distinguir línguas, métodos e abordagens utilizados para ser feita a identificação tanto em música quanto apenas de locutor. A partir disso foi identificado que características estáticas podem ser voláteis, não trazendo o melhor desempenho para o sistema. Para contornar este problema, o presente trabalho desenvolveu um identificador utilizando características significativas do sinal de áudio, implementado com o uso de rede *SincNet*, que aplicam filtros para extrair características significativas do sinal de áudio, dado o objetivo do sistema. Como a rede *SincNet* é um modelo DL, os filtros são melhorados a cada etapa de treinamento da rede conforme a rede aprende.

Os resultados mostraram que o uso de características significativas produzem uma melhor representação para o problema de identificação de línguas do que características estáticas. A capacidade da rede aprender e se adaptar para extrair características do sinal de áudio, dado o objetivo do sistema, obteve um desempenho de aproximadamente $16,9\%$ melhor em comparação ao uso de características estáticas (MFCC). A adição das etapas de pré-processamento tornou o desempenho do sistema melhor, aproximadamente 12% em comparação ao mesmo

modelo sem pré-processamento, dado que explicita como os trechos irrelevantes e ruídos do sinal de áudio impactam negativamente na identificação de língua.

O sistema proposto obteve resultados coerentes dado o objetivo do trabalho, como esperado obteve um desempenho superior ao *baseline*, apresentando o ganho que o uso de características significativas podem trazer para tarefas de identificação em áudio. Obteve uma acurácia de 68.5%, um número relativamente baixo, que pode ter ocorrido dado a base de dados utilizada, a qual possui 3 línguas de mesma origem, portanto existindo bastante semelhança entre elas. Com a aplicação de mais tempo de treinamento, em uma base com maior número de músicas, esse desempenho talvez possa ser ainda mais melhorado.

REFERÊNCIAS

ADAMI, A. G. **Modeling prosodic differences for speaker and language recognition**. Tese (Doutorado) — Oregon Health & Science University, Portland, USA, 2004.

AGGARWAL, C. C. *et al.* **Neural networks and deep learning**. [S.l.]: Springer, 2018. v. 10. 978–3 p.

ALLEN, J. Short term spectral analysis, synthesis, and modification by discrete fourier transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 25, n. 3, p. 235–238, 1977.

ALZUBAIDI, L. *et al.* Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. **Journal of big Data**, Springer, v. 8, n. 1, p. 1–74, 2021.

AMBIKAIRAJAH, E. *et al.* Language identification: A tutorial. **IEEE Circuits and Systems Magazine**, v. 11, n. 2, p. 82–108, 2011.

ANGUERA, X. *et al.* Speaker diarization: A review of recent research. **IEEE Transactions on audio, speech, and language processing**, IEEE, v. 20, n. 2, p. 356–370, 2012.

BAI, Z.; ZHANG, X.-L. Speaker recognition based on deep learning: An overview. **Neural Networks**, Elsevier, v. 140, p. 65–99, 2021.

BALTRUNAS, L. *et al.* Incarmusic: Context-aware music recommendations in a car. In: SPRINGER. **International conference on electronic commerce and web technologies**. [S.l.], 2011. p. 89–100.

BENESTY, J. *et al.* **Springer handbook of speech processing**. [S.l.]: Springer, 2008. v. 1. 179–179 p.

BIRJALI, M.; KASRI, M.; BENI-HSSANE, A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. **Knowledge-Based Systems**, v. 226, p. 107134, 2021. ISSN 0950-7051.

BREDIN, H.; LAURENT, A. End-to-end speaker segmentation for overlap-aware resegmentation. In: **Proceedings of Interspeech 2021**. [S.l.: s.n.], 2021. p. 3111–3115.

CAMPBELL, J. P. Speaker recognition: A tutorial. **Proceedings of the IEEE**, IEEE, v. 85, n. 9, p. 1437–1462, 1997.

CASEY, M. A. *et al.* Content-based music information retrieval: Current directions and future challenges. **Proceedings of the IEEE**, IEEE, v. 96, n. 4, p. 668–696, 2008.

CASTRO, F. M. *et al.* End-to-end incremental learning. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018.

CHOI, K.; WANG, Y. **Listen, Read, and Identify: Multimodal Singing Language Identification of Music**. [S.l.]: arXiv, 2021.

CHUNG, J. S.; NAGRANI, A.; ZISSERMAN, A. VoxCeleb2: Deep Speaker Recognition. In: **Proceedings of Interspeech 2018**. [S.l.: s.n.], 2018. p. 1086–1090.

DAVIS, S.; MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 28, n. 4, p. 357–366, 1980.

DÉFOSSEZ, A. Hybrid spectrogram and waveform source separation. In: **Proceedings of the 2021 Workshop on Music Source Separation (ISMIR)**. [S.l.: s.n.], 2021.

DELLER, J.; PROAKIS, J.; HANSEN, J. Discrete-time processing of speech signals. In: **Discrete-time processing of speech signals**. [S.l.: s.n.], 1993.

DESHWAL, D.; SANGWAN, P.; KUMAR, D. Feature extraction methods in language identification: a survey. **Wireless Personal Communications**, Springer, v. 107, n. 4, p. 2071–2103, 2019.

DIELEMAN, S.; SCHRAUWEN, B. End-to-end learning for music audio. In: **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2014. p. 6964–6968.

DORRELL, P. Sound and music. In: ____. **What is Music?: Solving a Scientific Mystery**. [S.l.]: Lulu.com, 2005. p. 63–86.

DUBEY, H.; SANGWAN, A.; HANSEN, J. H. L. Transfer learning using raw waveform sincnet for robust speaker diarization. In: **ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2019. p. 6296–6300.

EL-FATTAH, A. *et al.* Speech enhancement with an adaptive wiener filter. **International Journal of Speech Technology**, Springer, v. 17, n. 1, p. 53–64, 2014.

FAINBERG, J. *et al.* Acoustic model adaptation from raw waveforms with sincnet. In: **2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)**. [S.l.: s.n.], 2019. p. 897–904.

FOIL, J. Language identification using noisy speech. In: **ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing**. [S.l.: s.n.], 1986. v. 11, p. 861–864.

FUENTES, M. *et al.* Tracking beats and microtiming in afro-latin american music using conditional random fields and deep learning. In: **Proceedings of the 20th International Society for Music Information Retrieval Conference**. Delft, The Netherlands: [s.n.], 2019. p. 251–258.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. [S.l.]: "O'Reilly Media, Inc.", 2022.

GLASMACHERS, T. Limits of end-to-end learning. In: PMLR. **Asian Conference on Machine Learning**. [S.l.], 2017. p. 17–32.

GOLD, B.; MORGAN, N.; ELLIS, D. **Speech and audio signal processing: processing and perception of speech and music**. [S.l.]: John Wiley & Sons, 2011.

____. **Speech and audio signal processing: processing and perception of speech and music**. [S.l.]: John Wiley & Sons, 2011. 235–238 p.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613. Disponível em: <https://books.google.com.br/books?id=Np9SDQAAQBAJ>.

GRAVES, A. Supervised sequence labelling. In: **Supervised sequence labelling with recurrent neural networks**. [S.l.]: Springer, 2012.

GROENBROEK, H. **A Machine Learning Approach to Automatic Language Identification of Vocals in Music**. Dissertação (Mestrado) — University of Groningen, 2021.

HERMANSKY, H. Perceptual linear predictive (plp) analysis of speech. **The Journal of the Acoustical Society of America**, v. 87, n. 4, p. 1738–1752, 1990. Disponível em: <https://doi.org/10.1121/1.399423>.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, World Scientific, v. 6, n. 02, p. 107–116, 1998.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.

HU, Y.; LIU, G. Separation of singing voice using nonnegative matrix partial co-factorization for singer identification. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, IEEE, v. 23, n. 4, p. 643–653, 2015.

IGLESIAS, D. del C. **End-to-end Learning for Singing-Language Identification**. 56 p. Dissertação (Mestrado) — KTH, School of Electrical Engineering and Computer Science (EECS), 2020.

INTERNATIONAL FEDERATION OF THE PHONOGRAPHIC INDUSTRY. **Global Music Report in 2021**. 2021. Disponível em: <https://www.ifpi.org/ifpi-issues-annual-global-music-report-2021/>. Acesso em: 21 mar. 2022.

____. **Global Music Report in 2022**. 2022. Disponível em: <https://www.ifpi.org/ifpi-global-music-report-global-recorded-music-revenues-grew-18-5-in-2021/>. Acesso em: 23 mar. 2022.

JAMES, G. M. **Majority vote classifiers: theory and applications**. [S.l.]: Stanford University, 1998.

JUSLIN, P. N. Emotional Reactions to Music. In: **The Oxford Handbook of Music Psychology**. Oxford University Press, 2016. ISBN 9780198722946. Disponível em: <https://doi.org/10.1093/oxfordhb/9780198722946.013.17>.

KATTENBORN, T. *et al.* Review on convolutional neural networks (cnn) in vegetation remote sensing. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 173, p. 24–49, 2021.

KRUSPE, A. M. **Application of automatic speech recognition technologies to singing**. Tese (Doutorado) — Technical University of Munich, Munich, Germany, 2018.

KRUSPE, A. M.; ABESSER, J.; DITTMAR, C. A gmm approach to singing language identification. In: AUDIO ENGINEERING SOCIETY. **Audio Engineering Society Conference: 53rd International Conference: Semantic Audio**. [S.l.], 2014.

KRUSPE, A. M.; FRAUNHOFER, I. Improving singing language identification through i-vector extraction. In: **DAFx**. [S.l.: s.n.], 2014. p. 227–233.

____. Phonotactic language identification for singing. In: **INTERSPEECH**. [S.l.: s.n.], 2016. p. 3319–3323.

KULIS, B.; SAENKO, K.; DARRELL, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: **CVPR 2011**. [S.l.: s.n.], 2011. p. 1785–1792.

MAASØ, A.; SPILKER, H. S. The streaming paradox: Untangling the hybrid gatekeeping mechanisms of music streaming. **Popular Music and Society**, Taylor & Francis, v. 45, n. 3, p. 300–316, 2022.

MAYOR-TORRES, J. M. *et al.* Interpretable sincnet-based deep learning for emotion recognition from eeg brain activity. In: **2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)**. [S.l.: s.n.], 2021. p. 412–415.

MEHRABANI, M.; HANSEN, J. H. L. Language identification for singing. In: **2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2011. p. 4408–4411.

MORRIS, J. W.; POWERS, D. Control, curation and musical experience in streaming music services. **Creative Industries Journal**, Taylor & Francis, v. 8, n. 2, p. 106–122, 2015.

MUKHERJEE, H. *et al.* Identifying language from songs. **Multimedia Tools and Applications**, Springer, v. 80, n. 28, p. 35319–35339, 2021.

NIETO, O. *et al.* Audio-based music structure analysis: Current trends, open challenges, and applications. **Transactions of the International Society for Music Information Retrieval**, Ubiquity Press, v. 3, n. 1, 2020.

PADI, B.; MOHAN, A.; GANAPATHY, S. End-to-end language recognition using attention based hierarchical gated recurrent unit models. In: **ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2019. p. 5966–5970.

PALAZ, D.; COLLOBERT, R.; DOSS, M. M. **End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks**. [S.l.]: arXiv, 2013.

PARCOLLET, T.; MORCHID, M.; LINARèS, G. E2e-sincnet: Toward fully end-to-end speech recognition. In: **ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2020. p. 7714–7718.

PELLEGRINO, F.; ANDRE-OBRECHT, R. Automatic language identification: an alternative approach to phonetic modelling. **Signal Processing**, v. 80, n. 7, p. 1231–1244, 2000. ISSN 0165-1684.

PRETTENHOFER, P.; STEIN, B. Cross-language text classification using structural correspondence learning. In: **Proceedings of the 48th annual meeting of the association for computational linguistics**. [S.l.: s.n.], 2010. p. 1118–1127.

RABINER, L.; SCHAFER, R. **Theory and applications of digital speech processing**. [S.l.]: Prentice Hall Press, 2010.

RAVANELLI, M.; BENGIO, Y. Speaker recognition from raw waveform with sincnet. In: **2018 IEEE Spoken Language Technology Workshop (SLT)**. [S.l.: s.n.], 2018. p. 1021–1028.

RENAULT, L.; VAGLIO, A.; HENNEQUIN, R. Singing language identification using a deep phonotactic approach. In: **ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2021. p. 271–275.

ROUAS, J.-L. *et al.* Modeling prosody for language identification on read and spontaneous speech. In: **2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).** [S.l.: s.n.], 2003. v. 6, p. I–40.

ROXBERGH, L. **Language classification of music using metadata**. 2019.

SAMMUT, C.; PHUNG, D.; WEBB, G. **Encyclopedia of Machine Learning and Data Science**. Springer US, 2017. (Springer reference). ISBN 9781489975027. Disponível em: <https://books.google.com.br/books?id=Ja_-CQAAQBAJ>.

SAVIC, M.; GUPTA, S.; ACOSTA, E. An automatic language identification system. In: **Acoustics, Speech, and Signal Processing, IEEE International Conference on**. Los Alamitos, CA, USA: IEEE Computer Society, 1991. p. 817–820.

SCHEDL, M. *et al.* Music information retrieval: Recent developments and applications. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 8, n. 2-3, p. 127–261, 2014.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. [S.l.]: Cambridge University Press, 2014.

SHARMA, G.; UMAPATHY, K.; KRISHNAN, S. Trends in audio signal feature extraction methods. **Applied Acoustics**, v. 158, p. 107020, 2020. ISSN 0003-682X.

SHENOI, B. A. **Introduction to digital signal processing and filter design**. [S.l.]: John Wiley & Sons, 2005. 249–302 p.

STAUDEMEYER, R. C.; MORRIS, E. R. **Understanding LSTM–a tutorial into long short-term memory recurrent neural networks**. [S.l.]: arXiv, 2019.

STOLLER, D.; EWERT, S.; DIXON, S. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. In: GÓMEZ, E. *et al.* (Ed.). **Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018**. [s.n.], 2018. p. 334–340. Disponível em: <http://ismir2018.ircam.fr/doc/pdfs/205_Paper.pdf>.

TAN, Z.-H.; SARKAR, A. kr.; DEHAK, N. rvad: An unsupervised segment-based robust voice activity detection method. **Computer Speech & Language**, v. 59, p. 1–21, 2020. ISSN 0885-2308.

TORRES-CARRASQUILLO, P. A. *et al.* Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In: CITESEER. **Interspeech**. [S.l.], 2002.

TRIPATHI, M.; SINGH, D.; SUSAN, S. Speaker recognition using sincnet and x-vector fusion. In: RUTKOWSKI, L. *et al.* (Ed.). **Artificial Intelligence and Soft Computing**. Cham: Springer International Publishing, 2020. p. 252–260.

TRONG, T. N.; HAUTAMÄKI, V.; LEE, K.-A. Deep language: a comprehensive deep learning approach to end-to-end language recognition. In: **Odyssey**. [S.l.: s.n.], 2016. v. 2016, p. 109–116.

TSAI, W.-H.; WANG, H.-M. Automatic identification of the sung language in popular music recordings. **Journal of New Music Research**, Taylor & Francis, v. 36, n. 2, p. 105–114, 2007.

TSAI, W.-H.; WANG, H.-M. *et al.* Towards automatic identification of singing language in popular music recordings. In: **ISMIR 2004, 5th International Conference on Music Information Retrieval**. [S.l.: s.n.], 2004.

WANG, A. *et al.* An industrial strength audio search algorithm. In: CITESEER. **Ismir**. [S.l.], 2003. v. 2003, p. 7–13.

WANG, C.; MAHADEVAN, S. Heterogeneous domain adaptation using manifold alignment. In: **Twenty-second international joint conference on artificial intelligence**. [S.l.: s.n.], 2011.

WILLIAMS, R. J.; ZIPSER, D. Experimental analysis of the real-time recurrent learning algorithm. **Connection science**, Taylor & Francis, v. 1, n. 1, p. 87–111, 1989.

YANG, Q. *et al.* **Transfer Learning**. [S.l.]: Cambridge University Press, 2020.

YIN, W. *et al.* **Comparative study of CNN and RNN for natural language processing**. 2017.

ZHUANG, F. *et al.* A comprehensive survey on transfer learning. **Proceedings of the IEEE**, v. 109, n. 1, p. 43–76, 2021.

ZISSMAN, M. A.; BERKLING, K. M. Automatic language identification. **Speech Commun.**, Elsevier Science Publishers B. V., NLD, v. 35, n. 1–2, p. 115–124, aug 2001. ISSN 0167-6393.