SEPARAÇÃO DE FONTES SONORAS AUXILIADA POR DEEP LEARNING

Jéssica Richards Nascimento

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Carlos Eduardo Pedreira

Rio de Janeiro
Dezembro de 2021

SEPARAÇÃO DE FONTES SONORAS AUXILIADA POR DEEP LEARNING

Jéssica Richards Nascimento

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Carlos Eduardo Pedreira

Aprovada por: Prof. Carlos Eduardo Pedreita
Profª. Carolina Gil Marcelino
Prof. Hugo Tremonte de Carvalho

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2021

Richards Nascimento, Jéssica

Separação de fontes sonoras auxiliada por Deep Learning/Jéssica Richards Nascimento. – Rio de Janeiro: UFRJ/COPPE, 2021.

XIII, 38 p.: il.; 29, 7cm.

Orientador: Carlos Eduardo Pedreira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2021.

Referências Bibliográficas: p. 37 – 38.

1. separação de fontes sonoras. 2. monoaural. 3. deep-learning. I. Pedreira, Carlos Eduardo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Dedico esta dissetação a todos
que estiveram comigo nesse
caminho

Agradecimentos

Agradeço a minha família por ter me acompanhado durante todos esses anos e ter possibilitado essa jornada.

Agradeço a todos os professores que me ensinaram, sem cada um de vocês não conseguiria escrever esse trabalho. Um agradecimento especial ao professor Pedreira que aceitou orientar esse trabalho.

Obrigada aos professores Carolina Marcelino e Hugo Carvalho por terem aceitado participar da banca.

Agradeço a CAPES pela bolsa de mestrado, possibilitando esta dissertação.

Agradeço a todos os meus colegas de laboratório que me ajudaram tanto com sugestões e material para o trabalho, quanto em ideias de qual caminho seguir com a tese.

Agradeço também aos meus amigos por me apoiarem nessa jornada.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SEPARAÇÃO DE FONTES SONORAS AUXILIADA POR DEEP LEARNING

Jéssica Richards Nascimento

Dezembro/2021

Orientador: Carlos Eduardo Pedreira

Programa: Engenharia de Sistemas e Computação

Um stream auditivo é um grupo de sons que no entendimento humano pertencem à mesma cena. O uso de máscaras binárias para separar uma cena auditiva em dois ou mais streams tem se mostrado muito efetivo. Abordagens mais recentes usam métodos de aprendizado supervisionado para gerar essas máscaras. Os áudios utilizados nos experimentos foram gerados artificialmente, uma mistura de vogal falada e outro áudio. O trabalho utiliza esses áudios monoaurais, propondo encontrar uma máscara binária para o stream de interesse. Para encontrar essas máscaras duas abordagens foram utilizadas: a primeira trabalha com os coeficientes de frequ ência mel e rede neural convolucional, e a segunda com os espectrogramas dos áudios e uma rede U-Net. A primeira abordagem não se mostrou muito efetiva. A segunda apresentou melhores resultados.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SOUND SOURCE SEPARATION AUXILIATED BY DEEP LEARNING MODELS

Jéssica Richards Nascimento

December/2021

Advisor: Carlos Eduardo Pedreira

Department: Systems Engineering and Computer Science

An auditory stream is a group of sounds that in human perception belong to the same scene. The use of binary masks to segregate an auditory scene in two or more streams has shown to be very effective. More recent approaches use supervised learning models to create these binary masks. The audios used in the experiments were artificially created, a mixture of vowel sound and other audio. The work uses these monaurals audios, proposing to find a binary mask for the stream of interest. To find these masks two approaches were explored: the first one uses the mel frequency cepstral coefficients and the convolutional neural network, and the second one uses the audios spectrograms and a U-Net network. The first approach wasn't very effective. The second presented better results.

Sumário

Lista de Figuras **ix**

Lista de Tabelas **xi**

1 Introdução **1**

2 Banco de dados **4**

3 Ferramentas e métodos usados na separação de fontes sonoras **5**

3.1 Representação do sinal no domínio tempo 6

3.2 Espectrograma 7

3.3 Subtração espectral 10

3.4 Separação supervisionada de fala 11

3.5 Coeficientes cepstrais da frequência Mel - MFCC 13

4 Redes Neurais **17**

4.1 Rede Neural Convolucional (CNN) 17

4.2 U-Net 19

4.2.1 Convolução Transposta 20

5 Experimentos **24**

5.1 Primeira abordagem 24

5.2 Segunda abordagem 31

6 Conclusão **36**

Referências Bibliográficas **37**

Lista de Figuras

3.1 Amostragem de um sinal 7
3.2 Exemplo de aplicação da janela Hamming 8
3.3 Gerando um espectrograma 9
3.4 A Figura acima mostra uma separação por máscara binária. A Figura A na (esquerda e cima) é a representação da mistura vogal e música, a Figura B (direita e cima) é a representação do espectrograma do áudio da vogal pura a Figura D(direita e baixo) é a máscara binária calculada a partir da Figura B, e a Figura C(esquerda e baixo) é o espectrograma obtido após utilizar a máscara binária na Figura A. . 12
3.5 Gráfico da escala mel. 13
3.6 Banco de filtros triangulares 14
3.7 MFCC de um áudio 16

4.1 Exemplo de convolução entre o input e o kernel. 18
4.2 Exemplo de max-pooling usando o filtro 2x2 e stride 2. 19
4.3 Exemplo de segmentação de imagem. 20
4.4 Imagem do modelo original da U-Net. 21
4.5 Exemplo do método de convolução. 22
4.6 Método de convolução não tradicional. 22
4.7 Convolução apresentada como multiplicação de matrizes. 22
4.8 Exemplo de convolução transposta. 23

5.1 Matrizes de similaridade 26
5.2 Curva de aprendizado dependendo do ruído por trás 28
5.3 Classificação de vogais usando outros ruídos para treino 29
5.4 A imagem do lado esquerdo representa a imagem salva do espectrograma e a imagem do lado direito a sua respectiva máscara binária. 31

5.5 A imagem da esquerda mostra o caso em que não existe nenhum ponto interceptando as duas máscaras. A Figura do meio quando parte das máscaras se interceptam e a da direita quando elas são exatamente iguais. 32
5.6 Otimizador: sgd, função de perda: sparse categorical cross entropy, na última camada a ativação é softmax, resultado nos dados de treino: perda: 0.032, e iou: 0.1241 32
5.7 otimizador: Adam, função de perda: binary cross entropy, na última camada ativação: sigmoid, perda: 0.5865 - iou_score: 0.478, epochs : 100 33
5.8 Curva de aprendizado da U-Net usando múltiplos ruídos juntos. . . . 33
5.9 Dados de teste e seus respectivos resultados, na extrema esquerda, o espectrograma do áudio + ruído, no meio a máscara real e na direita á máscara gerada. 34

Lista de Tabelas

5.1 Resultados do primeiro treino de classificação da CNN. 25
5.2 Resultados obtidos usando as vogais puras, com a menor quantidade de ruído possível. 26
5.3 Tabela mostrando o resultado do tempo original de ínicio de término da vogal, comparado com o obtido a partir do modelo proposto . . . 30
5.4 Resultados obtidos para ruídos diferentes. 33
5.5 Resultados obtidos usando dados de treino e validação com diferentes tipos de ruídos diferentes 35

Lista de Acrónimos

ASA Análise de Cena Auditiva (do termo em inglês, *Auditory Scene Analysis*).

ASR Reconhecimento Automático de Fala (do termo em inglês, *Automatic Speech Recognition*).

CASA Análise Computacional de Cena Auditiva (do termo em inglês, *Computational Auditory Scene Analysis*).

CNN Rede Neural Convolucional (do termo em inglês *Convolutional Neural Network*).

DCT Transformadad Discreta do Cosseno (do termo em inglês, *Discrete Cosine Transform*).

DFT Transformada Discreta de Fourier (do termo em inglês, *Discrete Fourier Transform*).

IBM Máscara Binária Ideal (do termo em inglês, *Ideal Binary Mask*).

IoU Interseção dentro da União (do termo em inglês, *Intersection over Union*).

IRM Máscara Proporcional Ideal (do termo em inglês, *Ideal Ratio Mask*).

LC Critério Local.

MFCC Coeficientes Cepstrais da Frequência MEl (do termo em inglês, *Mel Frequency Cepstral Coefficients*).

ReLU Unidade Linear Retificada (do termo em inglês, *Rectified Linear Units*).

REPET Técnica de Extração de Padrão Repetitivo (do termo em inglês, *Repeating Pattern Extraction Technique*).

SNR Razão Sinal Ruído (do termo em inglês, *Signal to Noise Ratio*).

STFT Transformada de Fourier em Tempo Curto (do termo em inglês, *Short Time Fourier Transform*).

TBM Máscara Binária Desejada, (do termo em inglês, *Target Binary Mask*).

TF Tempo-Frequência.

Capítulo 1

Introdução

É comum a situação na qual se deseja escutar música em presença de ruídos externos que dificultam essa escuta. Por exemplo, quando ao escutar o rádio no carro,ocorre a passagem de uma ambul ância, a presença de sons provenientes de motor mec ânico ou mesmo outros passageiros conversando em voz alta. Em ambientes assim, consegue-se ouvir vários sons diferentes vindo de fontes sonoras completamente distintas e usualmente distingui-los com facilidade. Além dessa capacidade de distinguir os sons, o motorista consegue focar em somente um dos sons e trocar esse foco como bem desejar [1]. Por exemplo, numa roda com diferentes pessoas e conversas paralelas, consegue-se em geral focar e participar de uma das conversas ignorando completamente as demais. Soluções para esses problemas, segregação de fontes sonoras e focar em uma das fontes sonoras de interesse, podem trazer aprimoramentos na diminuição dos ruídos em aparelhos auditivos, reconhecimento de fala, fala automática, entre outros problemas relacionados aos áudios.

Bregman [2], escreve que existem semelhanças entre audição e visão. Uma cena visual pode ser descrita a partir de grupos de regiões, já que existe a possibilidade segmentar a cena usando texturas, cores ou outros atributos visuais. O mesmo fenômeno pode ser observado com a audição, quando algum som é captado pelos ouvidos, é possível descrever uma cena auditiva. Uma cena auditiva, poderia ser o som de canto de passarinhos e barulho de carros da rua, nessa cena os sons chegam aos ouvidos simultaneamente, apesar disso ainda é possível identificar que o som que chegou ao ouvido é proveniente de duas fontes sonoras distintas. Existe um campo de estudos chamados de Análise de Cena Auditiva (do termo em inglês, *Auditory Scene Analysis* (ASA)), que visa modelar a percepção auditiva.

Com a aplicação de ASA na computação surgiu a Análise Computacional de Cena Auditiva (do termo em inglês, *Computational Auditory Scene Analysis* (CASA)). Brown et al. [3], dão uma definição mais precisa de CASA. Eles definem esse campo de estudos como sendo o estudo computacional que visa atribuir a performance humana em ASA, as cenas acústicas devem ser gravadas usando um mi-

crofone(monoaurais) ou dois (binaurais). Essa definição deve-se ao fato que ao usar mais microfones para gravar, modelos que não se baseiam em ASA podem ser usados; como o beamforming ou filtração espacial, que são soluções não aplicáveis usando sons monoaurais ou binaurais e necessitam de ambientes mais controlados.

Um dos problemas mais famosos e estudados por ASA é o "problema do coquetel". McDermott [1] o define como a capacidade de distinguir um único som desejado a partir de um conjunto de sons, seja prestando atenção em uma única conversa em um restaurante cheio ou ouvir um animal se aproximando estando em uma floresta. Como vários sons chegam aos ouvidos ao mesmo tempo, eles estão misturados, por isso o primeiro desafio é classificar e separar os sons observados, já que esta mistura em si não traz grandes informações. O segundo desafio é focar em uma fonte sonora e trocar a fonte de interesse sempre que for desejado. Esses dois desafios estão interligados, já que para focar em uma fonte é necessário a separação de sons e a separação de sons é influenciada pela fonte que deseja-se ouvir.

Nos ouvidos acontece um pré-processamento de som, nesse pré-processamento acontece uma análise simplificada das frequências que compõem o som recebido [4]. Dado isso, existe a análise de Fourier que possibilita a representação física dos sons usando as suas frequências, nela os sons são representados como sendo a soma de ondas senoidais de frequências diferentes, possibilitando uma visualização mais fácil das frequências que compõem o som [5]. A análise de Fourier facilita a visualização das frequências presentes no som em relação ao tempo, porém por ser uma soma de senoidais de frequências diferentes ainda não é intuitivo a relação das frequências no sinal. Dado essa dificuldade existem representações do sinal que facilitam visualizar como as frequências que compõe o sinal mudam em relação ao tempo, uma dessas representações é o espectrograma.

O espectrograma comumente é gerado a partir do som originado de mais de uma fonte sonora, logo ele vai conter informações de mais de uma fonte, por esse motivo foi definido o stream. Bregman [2] define um stream auditivo como sendo um grupo de sons que no entendimento humano pertencem à mesma cena. Por exemplo, uma série de passos pode ser considerada como um acontecimento único, apesar de cada passo poder ser analisado um som separado. Por esses e outros exemplos ao citar som e eventos acústicos será em relação a parte física e stream será a representação perceptiva.

DeLiang Wang et al. [6] apresentam alguns dos métodos mais usados para a separação de fala. Existem as abordagens mais tradicionais para realizar a separação, como o aprimoramento da fala e CASA. No aprimoramento de fala o método mais usado é o da subtração espectral, já em CASA um dos modelos bastante utilizados é a Máscara Binária Ideal (do termo em inglês, *Ideal Binary Mask* (IBM)). O objetivo desse modelo é a criação de uma máscara binária para separar os streams presentes

no espectrograma. Algumas variações desse modelo são a Máscara Binária Desejada, (do termo em inglês, *Target Binary Mask* (TBM)) e Máscara Proporcional Ideal (do termo em inglês, *Ideal Ratio Mask* (IRM)). Assim em CASA as abordagens mais recentes se utilizam do aprendizado supervisionado para ajudar no problema de separação.

O objetivo deste trabalho é conseguir realizar a separação sonora de áudios que possuem mais de um stream. Para alcançar esse objetivo, o trabalho precisou de alguns marcos, como representar esses áudios no Tempo-Frequência (TF) utilizando o espectrograma, encontrar as máscaras bin árias e treinar uma rede neural para auxiliar essa separação.

A dissertação está dividida em seis capítulos. O Capítulo 2 descreve a criação da base de dados utilizada no Trabalho. O Capítulo 3 apresenta os conceitos teóricos das ferramentas utilizadas que auxiliam a separação dos streams. O Capítulo 4 apresenta os conceitos teóricos das redes neurais utilizadas nesta dissertação. O Capítulo 5 explica a construção dos experimentos e seu resultado final. O Capítulo 6 é a conclusão do trabalho.

Capítulo 2

Banco de dados

Os experimentos realizados neste trabalho usaram dados criados artificialmente. O objetivo do banco de dados usado é ser uma versão mais simplificada e controlada do problema, já que a separação de fontes sonoras é uma tarefa complexa de ser realizada.

Os dados para os experimentos utilizados nesta dissertação foram coletados a partir de quatro voluntários. Os voluntários, dois homens e duas mulheres, gravaram as vogais faladas 'a,e,i,o,u', cerca de 20 vezes. A gravação foi feita em uma sala sem acústica, mas com pouco ruído externo. Para a gravação foi utilizado um microfone de celular juntamente com um código Python que pedia para os participantes falarem as letras. Os dados foram coletados nos mesmos equipamentos para todos os voluntários.

O código python exibia uma mensagem para o voluntário falar, a partir desse momento a gravação ocorria durante dois segundos corridos, tempo estipulado experimentalmente para o voluntário falar a vogal. Não foi pedido aos participantes para mudarem as entonações das vogais, mas alguns dos áudios possuem entonações ligeiramente diferentes. Existem poucas diferenças fonéticas entre as vogais utilizadas, as principais usadas foram /á/,/é/,/i/,/ó/,/u/. Os áudios foram gravados numa frequência de 8kHz, visto que as vogais possuem frequências predominantemente baixas [7] e por muito tempo foi uma frequência empregada na telefonia móvel. Os áudios são monofônicos, possuem somente um canal.

Com esses áudios posteriormente foram criadas misturas artificiais. Os outros áudios foram extraídos da internet e utilizados para criar essa mistura. Os áudios baixados foram cortados para ter dois segundos de duração e uma frequência de amostragem de 8kHz. Os áudios selecionados foram: cachorro latindo, músicas instrumentais, barulho da chuva e som de maritacas.

Capítulo 3

Ferramentas e métodos usados na separação de fontes sonoras

Nas músicas é possível encontrar uma estrutura que se repete. Pardo et al [8] utilizam essa propriedade presente nas músicas para criar uma Técnica de Extração de Padrão Repetitivo (do termo em inglês, *Repeating Pattern Extraction Technique* (REPET)). O REPET é um algoritmo que procura elementos que se repetem em uma cena musical e separa a música em estrutura repetitiva e não repetitiva. Ele foi aplicado em músicas pop, cujo resultado da separação foi positivo. A estrutura do modelo se resume em três passos: identificar o período de repetição, modelar o segmento de repetição e extrair a estrutura de repetição.

A primeira etapa consiste em passar o áudio no domínio do tempo para o domínio da frequência, gerando um espectrograma. Usando o espectrograma um segmento de repetição é encontrado. A partir desse segmento um espectrograma de repetição é calculado e com esse espectrograma é possível realizar a separação do foreground que não se repete com o background que se repete.

Alguns conceitos presentes no artigo de Pardo et al. [8] são utilizados nesta dissertação, como a criação do espectrograma descrito na Seção 3.2 e como separar dois áudios a partir do espectrograma criado, técnica também conhecida como subtração espectral, descrita na Seção 3.3. O capítulo também aborda como representar um sinal no domínio tempo, descrito na Seção 3.1 e outros conceitos usados, como o algoritmo da IBM, descrito na Seção 3.4. Além disso na Seção 3.5 está descrito como os Coeficientes Cepstrais da Frequência Mel (do termo em inglês, *Mel Frequency Cepstral Coefficients* (MFCC)) são gerados.

3.1 Representação do sinal no domínio tempo

A física define o som como uma vibração, onda de pressão, que se propaga em um meio, liquido, gasoso ou sólido, por isso é possível representar o som pela amplitude da vibração em relação ao tempo [9].

Na natureza o som possui valores contínuos tanto no tempo quanto na amplitude, porém para ocorrer a digitalização esses valores precisam ser discretizados para serem salvos no computador. Uma maneira de discretizar esse sinal é usando sampleamento equidistante [5]. Para transformar um sinal analógico $f : \mathbb{R} \to \mathbb{R}$ em um sinal de tempo discreto a função $x : \mathbb{Z} \to \mathbb{R}$ é definida por:

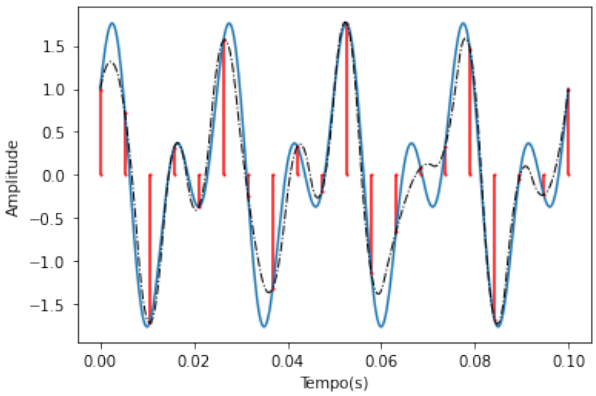
$$x(n) = f(nT). \tag{3.1}$$

Onde $T > 0$ é um número positivo que representa o período de amostragem do sinal, ou seja, a diferença de tempo T entre duas amostras consecutivas no sinal, normalmente medida em segundos. $x(n)$ é uma amostra do sinal original f medida no tempo $t = nT$. A frequência de amostragem do sinal F_s, é definida por:

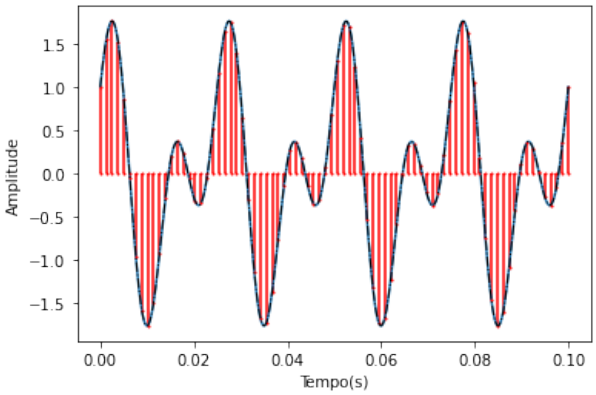
$$F_s = \frac{1}{T}. \tag{3.2}$$

A frequência de amostragem geralmente é medida pela quantidade de amostras em um segundo, Hertz (Hz).

Quando o sinal analógico é discretizado pode ocorrer perda de informação, ao escolher uma frequência de amostragem maior que o dobro da maior frequência que o sinal contém esse processo é mitigado. Esse é o teorema de sampleamento, um sinal que foi digitalizado pode ser reconstruído sem perdas de informação para sua forma analógia, caso não haja nenhuma frequência nele maior que $\Omega = \frac{F_s}{2}$ Hz. Ω é conhecido com frequência Nyquist e representa a máxima frequência que o sinal digital pode conter para não ter perdas de informação. Caso o sinal analógico tenha alguma frequência maior que Ω, então o sinal digital terá aliasing, Figura 3.1, que é quando algumas frequências presentes no sinal não são detectáveis.



(a) Sinal com Alisasing.



(b) Sinal sem Alising.

Figura 3.1: Amostragem de um sinal

O sinal original é representado nas imagens como sendo a curva azul, onda gerada artificialmente(f=80Hz, $sin(2\pi t) + cos(\pi t)$). As linhas verticais vermelhas são as amostras do sinal e a curva em traços é o sinal gerado a partir dessas amostras a partir da interpolação cúbica. 3.1a exemplo de aliasing quando a frequência de amostragem escolhida é menor que a frequência Nyquist. 3.1b $F_s = 160Hz$, a quantidade de amostras nessa figura foi reduzida a fim de melhor mostrar o gráfico.

3.2 Espectrograma

Uma representação de tempo-frequência do sinal muito utilizada é o espectrograma. Para calcular o espectrograma é necessário obter uma representação do sinal na frequência, para isso pode-se utilizar a transformada de Fourier. Ao fazer a transformada de Fourier em todo do sinal numa única vez, o resultado final conterá todas as frequências do sinal, a desvantagem desse método acontece quando o sinal não é estacionário. Pois como o espectrograma é a magnitude dessa representação, ele mostraria que todas as frequências estão presentes em todos os instantes de tempo, o que não retrata a realidade, pois um sinal não estacionário não apresenta as mesmas compenentes de frequência durante sua duração.

Uma maneira de contornar esse problema é definir um intervalo de tempo, tal que o sinal possa ser considerado estacionário nesse intervalo. Uma maneira de escolher esses intervalos é definir um tempo t e uma vizinhança em torno desse ponto, de maneira que, quanto mais distante estiver o ponto de t, menor será sua influência no sinal. Esses intervalos que serão escolhidos terão como característica a atenuação do sinal nas bordas, cujo valor fora do intervalo será zero. Para isso é usado uma função janela (do termo em inglês, *window function*).

Uma função janela muito usada é a janela Hann, ela é definida pela seguinte equação:

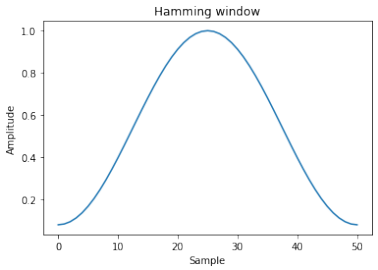
$$w[n] = \begin{cases} 1 + \frac{cos(n)}{2} & \text{se } -0.5 \leq n \leq 0.5; \\ 0 & \text{c.c.} \end{cases} \tag{3.3}$$

Além a janela Hann, outra janela bastante utilizada é a janela Hamming, definida por:

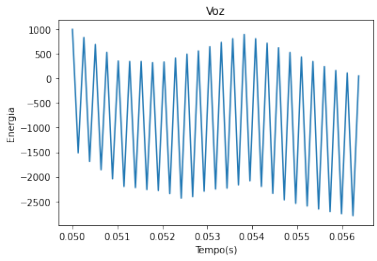
$$w[n] = \begin{cases} 0.54 - 0.46cos(2nL - 1) & \text{se } 0 \leq n \leq L - 1; \\ 0 & \text{c.c.} \end{cases} \tag{3.4}$$

Nos experimentos realizados nessa dissertação a janela escolhida foi a Hamming, a Figura 3.2 mostra um sinal atenuado por essa janela. Para extrair o sinal, utilizamos a equação abaixo, onde $x[n]$ é o valor do sinal no tempo n e $w[n]$ é o valor da janela no tempo n:

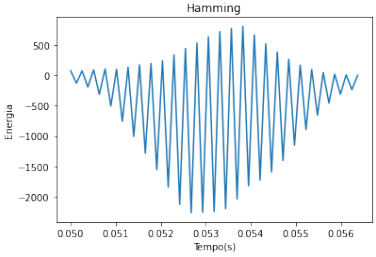
$$y[n] = w[n]x[n]. \tag{3.5}$$



(a) Janela Hamming.



(b) Áudio no domínio tempo.



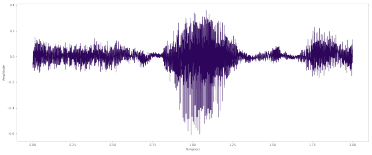
(c) Janela Hamming aplicada ao sinal de áudio da Figura 3.2b.

Figura 3.2: 3.2a: Aparência da janela hamming, de 50 amostras. 3.2b: Intervalo de um sinal de áudio. 3.2c: Intervalo 3.2b, após a janela Hamming ser aplicada para atenuaç ão das bordas.

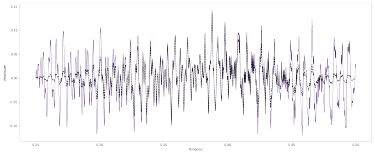
Ao aplicar a janela Hamming no sinal, é obtido um intervalo onde suas bordas estão atenuadas. Após esse passo a Transformada Discreta de Fourier (do termo

em inglês, *Discrete Fourier Transform* (DFT)) é aplicada, gerando uma representação do sinal na frequência. Esses passos contituem o algoritmo da Transformada de Fourier em Tempo Curto (do termo em inglês, *Short Time Fourier Transform* (STFT)). O último passo para obter a representação no tempo-frequência, é elevar ao quadrado a magnitude da STFT.

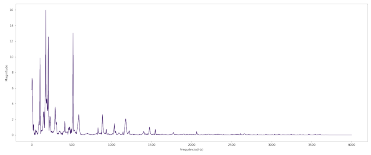
O STFT tem um parâmetro conhecido como tamanho do salto. Esse parâmetro defini quantas amostras a partir do primeiro ponto da janela anterior é necessário pular a fim de selecionar a próxima janela de sinal.



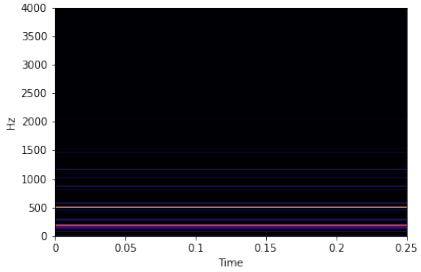
(a) Sinal no domínio tempo.



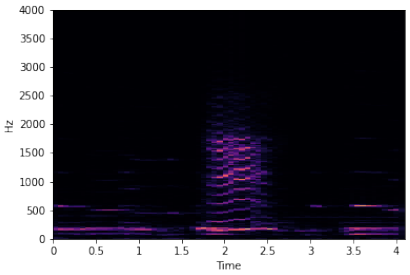
(b) Janela de tempo.



(c) FFT da janela.



(d) Espectrograma de uma janela.



(e) Espectrograma final.

Figura 3.3: Os gráficos contidos mostram os passos realizados ao gerar um espectrograma. A Figura 3.3a representa o sinal do domínio do tempo. Na Figura 3.3b a linha em azul representa um intervalo do sinal de 25ms, já a linha preta pontilhada o mesmo sinal, após aplicar uma janela Hamming. A Figura 3.3c mostra a transformada de fourier aplicada à 3.3b. A Figura 3.3d mostra o espectrograma relativo a 3.3c. A Figura 3.3e mostra o espectrograma completo do sinal de 3.3a.

Existem limitações sobre a precisão da representação do sinal no tempo-frequência. Ao aumentar o tamanho da janela de tempo, a quantidade de frequências distintas que aparecem no espectrograma aumentam, porém a quantidade de intervalos de tempo diminui, o contrário acontece quando um intervalo de tempo menor é escolhido. Esse é o princípio da incerteza. Essa limitação impede que essas duas

precisões aumentem significativamente ao mesmo tempo. O principio da incerteza diz que o tamanho da janela tempo-frequência não pode ser diminu ído o quanto se desejar, sendo limitado por $\Delta_\omega \Delta_t \geq \frac{1}{2}$ [10], onde Δ_t, Δ_ω é tamanho da janela tempo e frequência respectivamente.

3.3 Subtração espectral

A fala é muito importante para a comunicação humana, por isso conseguir separar a fala de interesse de interferências externas é desejável. Normalmente ela está corrompida por outras fontes sonoras ou reverberação causada por reflexão das ondas nas superfícies [1].

No aprimoramento de fala a separação espectral é muito utilizada. Geralmente no aprimoramento de fala é assumido que o ruído presente no som é estacionário ou mais estacionário que a fala.

No trabalho de Boll [11], os experimentos são feitos a partir de um conjunto de sons, cujos ruídos foram adicionados acusticamente ou digitalmente. O experimento assume que o ruído no ambiente é localmente estacionário, ou seja, o valor esperado da magnitude espectral antes da fala acontecer é o mesmo de quando a fala é introduzida. Mesmo quando acontece uma mudança desse ruído, existe tempo suficiente para estimar uma nova magnitude espectral antes da fala ser introduzida. Também assumem que uma redução significativa do ruído pode ser obtida removendo somente a magnitude espectral do ruído da magnitude espectral do som.

Dado um sinal $x(k)$ é possível assumir que ele é resultante da adição de um ruído $n(k)$ ao sinal de interesse $s(k)$:

$$x(k) = n(k) + s(k). \tag{3.6}$$

Pegando suas transformadas de fourier:

$$X(e^{j\omega}) = N(e^{j\omega}) + S(k^{j\omega}). \tag{3.7}$$

A subtração espectral é dada pela subtração de uma estimativa média do espectro ruidoso do espectro do sinal ruidoso. Para realizar a subtração espectral, um filtro espectral $H(e^{j\omega})$ é estimado trocando o espectro ruidoso $N(e^{j\omega})$ por um espectro mensurável. Uma das maneira de calcular a magnitude $|N(e^{j\omega})|$ é substituí-la pela média $\mu(e^{j\omega})$ que pode ser calculada a partir dos instantes de tempo que não foram detectados fala. A fase $\Theta_N(e^{j\omega})$ de $N(e^{j\omega})$ é substituída pela fase $\Theta_x(e^{j\omega})$ do sinal $X(e^{j\omega})$. É possível calcular uma estimativa espectral da subtração, $\hat{S}(e^{j\omega})$, a partir dos dados anteriores:

$$\hat{S}(e^{j\omega}) = [|X(e^{j\omega})| - \mu(e^{j\omega})]e^{j\Theta_x(e^{j\omega})}, \tag{3.8}$$

ou:

$$\hat{S}(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}), \tag{3.9}$$

com

$$H(e^{j\omega}) = 1 - \frac{\mu(e^{j\omega})}{|X(e^{j\omega})|}, \tag{3.10}$$

onde:

$$\mu(e^{j\omega}) = E\{|N(e^{j\omega})|\}. \tag{3.11}$$

Dado que $\hat{S}(e^{j\omega})$ é a estimativa espectral do sinal sem o ruído, para se ter o som com uma redução do ruído no domínio do tempo só é preciso calcular a transformada inversa de fourier.

3.4 Separação supervisionada de fala

Abordagens mais recentes tratam a separação espectral como um problema de aprendizado supervisionado. Segundo DeLiang et al. [6], essa nova abordagem foi inspirada pela máscara de tempo-frequência, utilizada em CASA. É uma abordagem similar a da subtração espectral, nela uma máscara bidimensional é aplicada em uma representação tempo-frequência do sinal a fim de extrair o sinal desejado.

Uma máscara bastante utilizada para essa tarefa é a IBM. Essa máscara contém a informação se a magnitude relativa ao intervalo t pertence ao ao sinal desejado ou não, com isso a tarefa se torna uma classificação binária. A máscara binária ideal é definida numa representação bidimensional de tempo-frequência, como o espectrograma, de um sinal ruidoso. A matriz bidimensioal correspondente à IBM pode ser conseguida usando a seguinte equação:

$$IBM = \begin{cases} 1 & \text{se } SNR(t,f) > LC; \\ 0 & \text{c.c.} \end{cases} \tag{3.12}$$

Onde t, f são o tempo e frequência respectivamente. Razão Sinal Ruído (do termo em inglês, *Signal to Noise Ratio* (SNR)) é uma medida que compara o quanto de sinal desejado existe em relação a quantidade de ruído do sinal. A máscara binária assume valores 1 onde o valor de $SNR(t,f)$, no tempo t e na frequência f ultrapassa um Critério Local (LC) definido previamente, e 0 caso contrário [12].

Além da IBM, existem outros tipos de máscaras usadas na separação espectral.

Uma delas é TBM. A diferença dela em relação a IBM é que no cálculo da TBM a energia de cada unidade TF é comparada com uma interferência fixa.

Um experimento teste usando a IBM foi realizado. O experimento utilizou uma das vogais faladas gravadas anteriormente. Essa vogal falada foi misturada artificialmente com um trecho de dois segundos de uma música. Um espectrograma foi obtido usando a função signal.stft da biblioteca scipy.

Os seguintes valores foram usados para o cálculo do stft: nperseg=512, tamanho de cada segmento, noverlap=256, número de pontos que se sobrepõe entre segmentos, nfft=2048, tamanho da FFT usada, window, foram passados os valores da janela Hamming de tamanho=512, fs=8000, o valor da frequência de amostragem, o mesmo valor em que os áudios foram gravados. A partir do cálculo do STFT, o espectrograma é obtido calculando a sua magnitude quadrada.

Para gerar uma máscara binária foi utilizado o seguinte cálculo: primeiro é descoberto o máximo de energia em cada espectro da frequência, depois calcula-se a média desses valores salvos, após isso tem:

$$M(t,f) = \begin{cases} 1 & \text{se o espectrograma}(t,f) > \text{média;} \\ 0 & \text{caso contrário.} \end{cases} \tag{3.13}$$

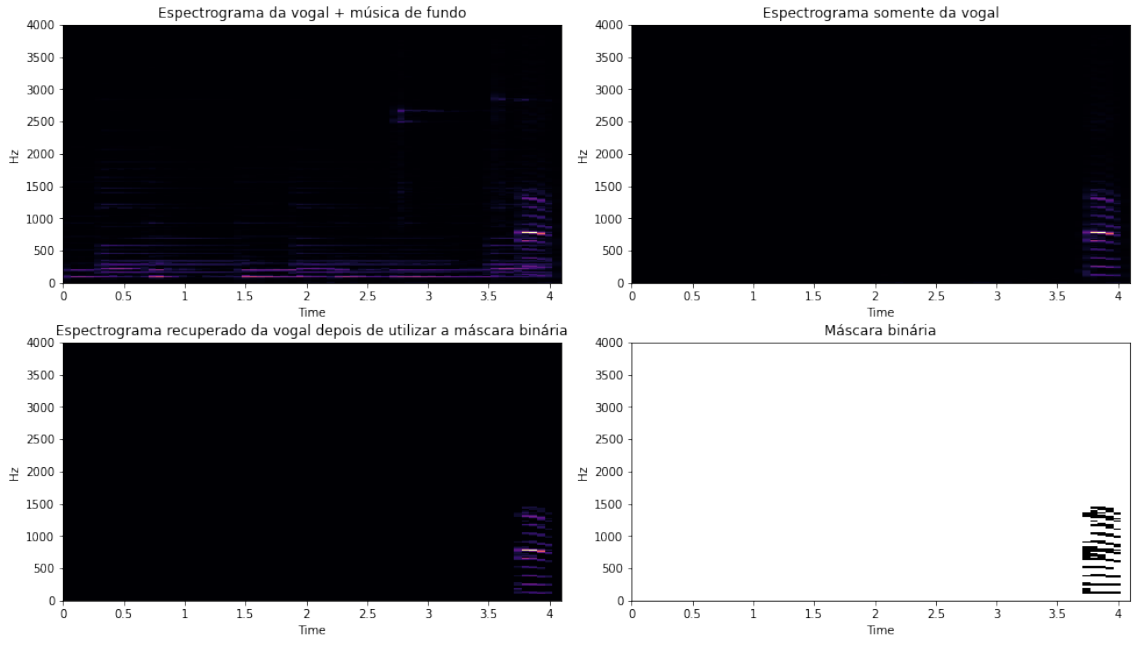


Figura 3.4: A Figura acima mostra uma separação por máscara binária. A Figura A na (esquerda e cima) é a representação da mistura vogal e música, a Figura B (direita e cima) é a representação do espectrograma do áudio da vogal pura a Figura D(direita e baixo) é a máscara binária calculada a partir da Figura B, e a Figura C(esquerda e baixo) é o espectrograma obtido após utilizar a máscara binária na Figura A.

3.5 Coeficientes cepstrais da frequência Mel - MFCC

Um modelo muito utilizado no campo de reconhecimento de som e fala automática são os MFCC. Os MFCC foram introduzidos inicialmente por Davis e Mermelstein em 1980 e são derivados de um tipo de representação cepestral de um áudio.

O cepestro, início da palavra espectro invertida, é definido como sendo o espectro do logaritmo do espectro de uma forma de onda [7]. A escala mel é uma escala perceptiva, ela mantém a relação da percepção humana dos sons, ou seja, os sons que os observadores julgaram como equidistantes continuam equidistantes nesta escala.

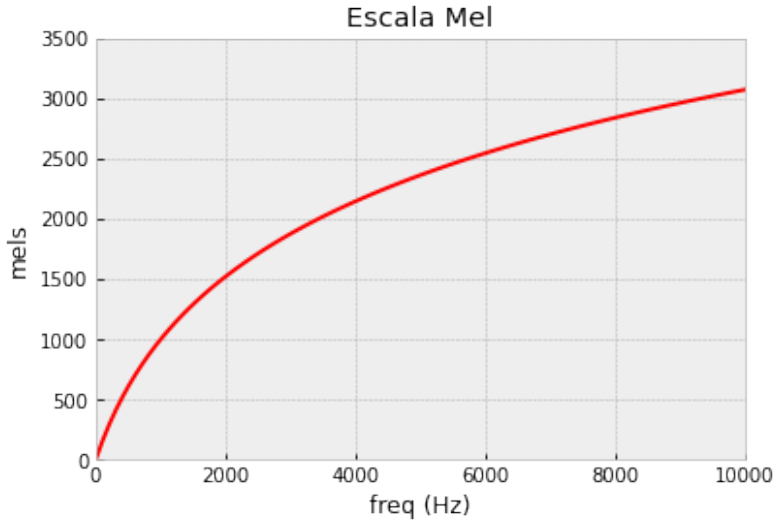


Figura 3.5: Gráfico da escala mel.

Para calcular os MFCC [13] o primeiro passo é transformar um sinal analógico em um sinal digital, após isso, um filtro pré-enfático é aplicado. O objetivo desse filtro é aumentar a quantidade de energia nas frequências mais altas, dado que elas têm uma menor quantidade de energia, possibilitando que a informação sobre essas frequências esteja mais disponível para o modelo. Após a aplicação do filtro, o espectrograma desse novo sinal é calculado usando o STFT e a janela hamming, Seç ão 3.2.

O espectro do sinal mostra a quantidade de energia em cada frequência de banda, porém a audição humana possui uma percepção diferente, ela é mais sensível nas frequências menores que 1kHz e depois se torna menos sensível, como mostrado o gráfico da escala mel, Figura 3.5. Para aproximar esse espectro da percepção humana, a escala mel é usada.

Seja um banco de filtros com M filtros ($m = 1, 2, 3, ..., M$), onde o filtro m é um

filtro triangular dado por:

$$H_m[k] = \begin{cases} 0 & k < f[m-1]; \\ \frac{k-f[m-1]}{f[m]-f[m-1]} & f[m-1] \leq k \leq f[m]; \\ \frac{f[m+1]-k}{f[m+1]-f[m]} & f[m] \leq k \leq f[m+1]; \\ 0 & k > f[m+1]. \end{cases} \tag{3.14}$$

Os filtros computam uma média das energias em torno da frequência central $f[m]$ e a quantidade de frequências englobadas por um único filtro vai aumentando conforme as frequências vão crescendo, esse fen ômeno é possível observar pela Figura 3.6.

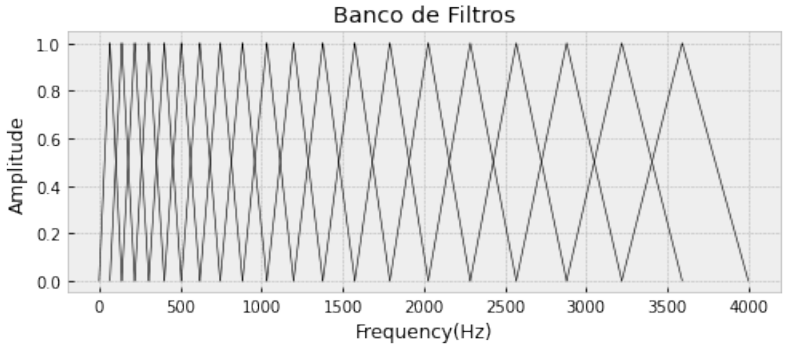


Figura 3.6: Banco de filtros triangulares

Para calcular esse banco de filtros é definido previamente a menor e a maior frequência que estarão presentes, f_{min} e f_{max} dados em Hz. Seja F_s a frequência de amostragem em Hz, M o número de filtros e N o tamanho da FFT. Para obter os pontos $f[m]$ é preciso calcular as correspondentes frequências máxima e mínima na escala mel e fazer um equidistanciamento dessas frequências usando quantidade de filtros desejados, após isso só é necessário aplicar a inversa da escala mel para obter os pontos $f[m]$.

$$f[m] = \frac{N}{F_s} B^{-1}\left(B(f_{min}) + m\frac{B(f_{max}) - B(f_{min})}{M+1}\right). \tag{3.15}$$

Onde a escala Mel é dada por:

$$B(f) = 1125ln\left(1 + \frac{1}{700}\right), \tag{3.16}$$

e sua inversa B^{-1} é dada por:

$$B^{-1}(b) = 700\left(exp\left(\frac{b}{1125}\right)\right). \tag{3.17}$$

No final de cada filtro é computada uma energia log:

$$S[m] = ln \left[\sum_{m=0}^{N-1} |X_a[k]|^2 H_m[k] \right] \ , \ 0 \leq m < M. \tag{3.18}$$

Onde o valor M varia conforme a quantidade de filtros escolhidos, 24 a 40. Cada aplicação do filtro em um intervalo de tempo retorna um coeficiente cepestral.

Apesar do cepestro ser definido como a transformada inversa de fourier do log do espectro, a Transformada Discreta do Cosseno (do termo em inglês, *Discrete Cosine Transform* (DCT)) pode ser usada ao invés da transformada discreta de fourier inversa, já que $S[m]$ tem tamanho par [13].

Para o Reconhecimento Automático de Fala (do termo em inglês, *Automatic Speech Recognition* (ASR)) normalmente são usados os primeiros 13 coeficientes, pois eles contêm a maior quantidade de informações relevantes. Como aumentar a quantidade de coeficientes não resultará em uma melhora de informação, é possível utilizar os coeficientes delta para capturar as mudanças espectrais no tempo. Um vetor de atributos com os coeficientes mais importantes podem ser formados por:

- 13 coeficientes MFCC, c_k
- 13 coeficientes delta MFCC, $\Delta c_k = c_{k+2} - c_{k-2}$
- 13 coeficientes delta de segunda ordem MFCC, $\Delta\Delta c_k = \Delta c_{k+1} - \Delta c_{k-1}$

A figura abaixo mostra um MFCC da vogal a falada, usando a biblioteca librosa do python, usando os seguinte parâmetros: *sample_rate = 8000, n_mfcc = 40, hop_length=512.*

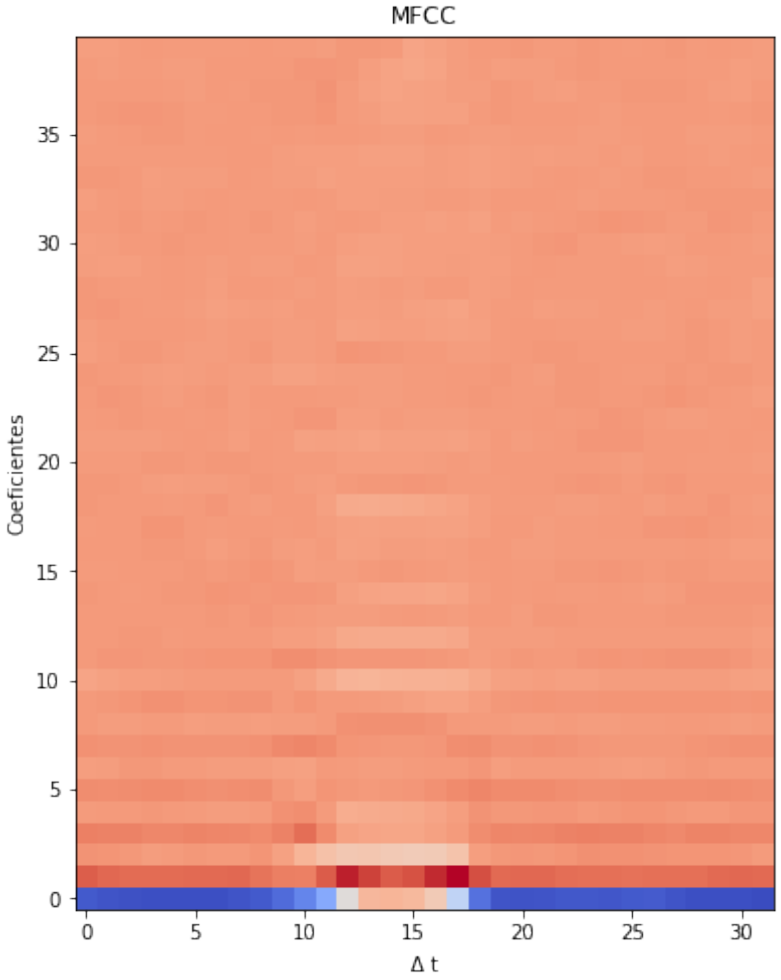


Figura 3.7: MFCC de um áudio

Capítulo 4

Redes Neurais

Nesse capítulo será apresentado de forma sucinta os modelos de redes neurais utilizados nesta dissertação. É possível observar que a separação dos sons pode ser feita somente usando a imagem da representação espectral deles. Dado isso duas redes neurais bastante populares quando se trabalha com imagens são a Rede Neural Convolucional (do termo em inglês *Convolutional Neural Network* (CNN)) e a U-Net.

4.1 Rede Neural Convolucional (CNN)

Aa Rede Neural Convolucional (do termo em inglês *Convolutional Neural Network* (CNN)) [14] é um tipo de rede neural artificial que tem como característica a capacidade de aprendizado dos filtros necessários para a classificação, por isso é um modelo muito utilizado na classificação de imagens e no processamento de vídeos [15]. Seu nome se deve ao fato de utilizar a convolução como ferramenta para auxiliar o aprendizado, essa ideia ajuda a resolver alguns desafios do aprendizado de máquina, como interações esparsas, compartilhamento de parâmetros e representações equivariantes.

A convolução na matemática é dada pela seguinte equação:

$$s(t) = (x * \omega)(t) = \int x(a)\omega(t-a)da. \tag{4.1}$$

Onde x e ω são funções definidas no domínio dos números reais. Como a CNN é um modelo computacional, é utilizado a convolução discreta ao invés da contínua.

$$s(t) = \sum_{a=-\infty}^{\infty} x(a) * \omega(t-a). \tag{4.2}$$

Ao usar a CNN na classificação de imagens, a entrada da rede são imagens, que na equação acima são representadas pela função x. A imagem pode ser representada

computacionalmente de algumas maneiras, uma delas é utilizando três matrizes onde cada uma representa a quantidade de verde, vermelho e azul presente em cada pixel, representação RGB, outra possibilidade é adicionar outra matriz ao modelo anterior para representar a opacidade da imagem, RGBA, e caso a imagem só contiver variações do branco ao preto, pode ser utilizado grayscale.

Como a maioria das imagens utiliza o modelo de RGB, então nesse caso x é um tensor de três dimensões, e a função, será o kernel (filtro), que ajuda a rede a aprender sobre a estrutura espacial da imagem, esse filtro é aprendido no modelo. A CNN na maioria das aplicações usa a convolução junto com outras funções, definiremos essas operações como sendo uma camada da rede.

Assim que a CNN recebe como entrada uma imagem, é feita a convolução dela com o kernel que gera um resultado, também conhecido como mapa de recursos. Esse output possui como tamanho final, o tamanho inicial da imagem a menos do tamanho do kernel mais um pixel, por exemplo, na Figura 4.2 o tamanho da matriz é 5x5, já o do kernel é 2x2, então o resultado final tem o tamanho do kernel a menos (5x5 - 2x2) = (3x3), mais um pixel nas dimensões (3x3) + (1x1) = (4x4).

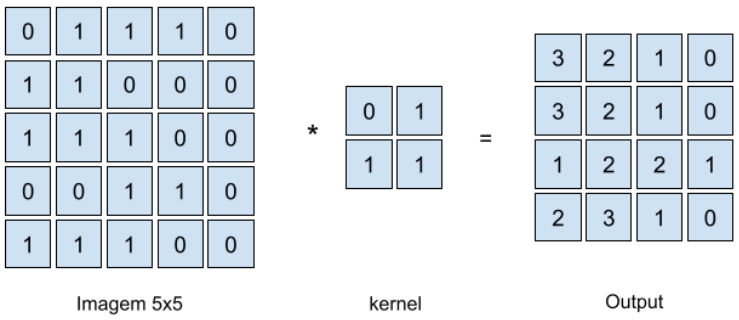


Figura 4.1: Exemplo de convolução entre o input e o kernel.

Se ao realizar a convoluação não for possível "aplicar"um número inteiro de kernels, então é possível adicionar algumas colunas de zeros para conseguir um valor inteiro de kernels, método conhecido como *padding*, ou eliminar algumas bordas presentes no *input*. Após essa etapa de convolução, usualmente coloca-se uma função para adicionar não linearidade à rede, essa função recebe o nome de função de ativação. Uma das funções mais usadas é a Unidade Linear Retificada (do termo em inglês, *Rectified Linear Units* (ReLU)):

$$f(x) = max(0, x). \tag{4.3}$$

A função sigmoidal também pode ser usada para esse propósito:

$$s(x) = \frac{1}{1 + e^{-x}}. \tag{4.4}$$

Após passar por essa função de ativação, existe a etapa de pooling, que reduz a quantidade de neurônios. Uma das maneiras de realizar essa operação é usando o pool máximo, Max-pooling, que gerará uma ativação máxima na região. A Figura 4.2 ilustra como funciona um filtro 2x2, usando um stride de tamanho 2. Stride é a quantidade de pixels que são pulados após o uso do filtro, por exemplo, se o stride fosse de tamanho 1, o próximo filtro seria a matriz, [[2,1],[2,1]], ao invés da [[1,0],[1,0]], como é no caso do stride 2.

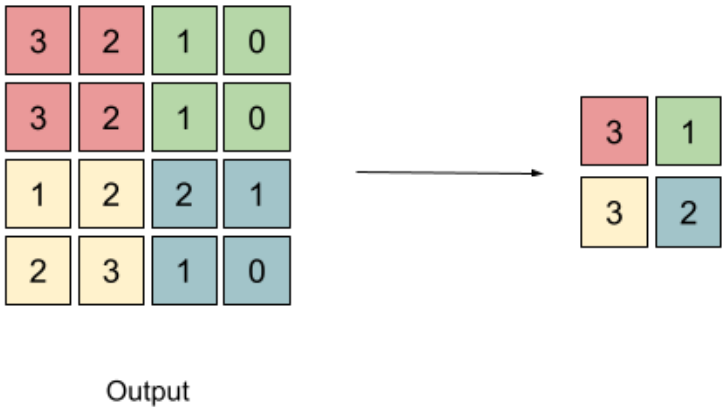


Figura 4.2: Exemplo de max-pooling usando o filtro 2x2 e stride 2.

Essas funções em sequência definem uma camada da CNN. Normalmente é usado mais de uma camada na construção da rede e a quantidade de camadas usadas depende do tamanho das imagens de entrada e da quantidade de atributos para aprender. Após definir a quantidade dessas camadas, coloca-se uma última camada totalmente conectada, ou seja, conecta todos os neurônios de saída da última camada a cada um dos neurônios de saída da rede, completando assim o modelo da CNN .

4.2 U-Net

A U-Net é uma rede neural bastante usada para a segmentação de imagens. Segmentar imagens é um problema de visão computacional, que procura particionar a imagem em segmentos ou pixels. A Figura 4.3, mostra um exemplo de uma possível segmentação, os pixels pretos são definidos como sendo background da imagem e os brancos o animal.

A U-Net tem como característica o seu formato de U, Figura 4.4, por isso **U**-Net. Ela é dividida em duas partes: uma de convolução, como a CNN , cujo objetivo

(a) Imagem original. (b) Máscara binária. (c) Imagem segmentada.

Figura 4.3: Exemplo de segmentação de imagem.

é classificar cada um dos pixels, e outra de deconvolução, expandindo a dimensão do output. A U-Net ter esse caminho de *upsampling* permite a propagação de informações de contexto às camadas superiores, além de fazer com que a rede seja simétrica a parte que diminui à dimensão, que implica na imagem da saída possuir um tamanho igual ou similar a da entrada.

A Figura 4.4 ilustra a primeira U-Net modelada [16]. Ela tem uma imagem 572x572 como entrada e cada seta para azul a direita é uma operação de convolução seguida pelo ReLU, já a seta vermelha para baixo, representa uma operação de max pool 2x2, essa estrutura se assemelha bastante a uma CNN. As setas cinzas representam a operação de corte e concatenação do tensor do caminho de *downsampling* ao de *upsampling*, essa operação é importante, já que ao fazer a convolução perdemos os pixels das bordas. Essa é uma maneira de recuperar essa informação. As setas verdes representam a convolução transposta 2x2 que diminui pela metade a quantidade de canais de atributos, seguida por uma concatenação com o mapa de atributos da caminho de compressão. A última camada do up-sampling é uma convolução 1x1 para mapear os pixels das classes desejadas. A rede possui 23 camadas convolucionais.

4.2.1 Convolução Transposta

Existem algumas estratégias possíveis para aumentar a resolução de uma imagem, como interpolação linear, quadrática, cúbica, sync, entre outras. Esses algoritmos geram combinações lineares/não lineares de valores já existentes dos pixels, não aprendendo nenhuma informação nova, ao contrário do kernel na CNN, então se for desejado que o algoritmo aprenda como aumentar a resolução, a convolução transposta pode ser usada.

A Figura 4.5 mostra um exemplo do método de convolução. A Figura 4.6,

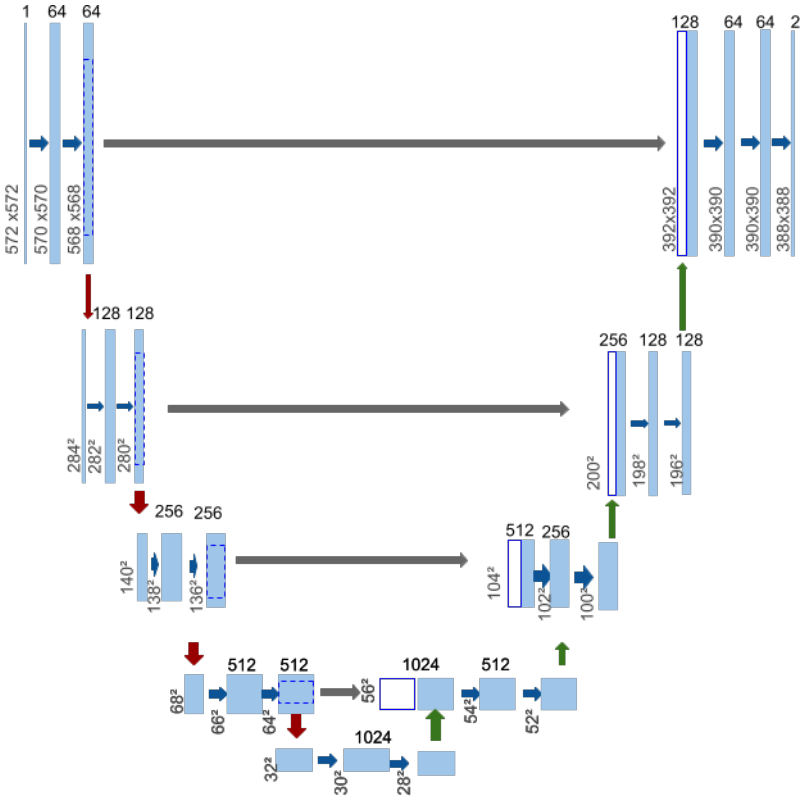


Figura 4.4: Imagem do modelo original da U-Net.

ilustra uma maneira diferente de visualizar esse método. A imagem de 4x4 pode ser achatada de maneira que seja um único vetor 16x1, Figura 4.6a, já o kernel 3x3 pode ser transformado numa matriz 4x16, Figura 4.6b, se multiplicarmos essas matrizes, teremos o mesmo resultado que se tiéssemos feito a convolução com as matrizes originais, porém esse resultado é achatado, 4x1, só um vetor, para transformá-lo no resultado esperado, só passar para a forma da matriz 2x2, Figura 4.7 [17].

A operação da convolução resulta em uma redução de dimensionalidade. No exemplo acima, a matriz 4x4 (vetor 16x1), foi para 2x2(vetor 4x1), já a convolução transposta tem o objetivo oposto, sair do vetor 4x1(matriz 2x2) e chegar no 16x1(matriz 4x4). Para isso é necessário transpor o kenrel. O kernel de 16x4, transposto passa a ter o tamanho de 4x16. Com o transposto, agora é possível fazer a multiplicação pelo vetor 4x1(matriz 2x2), recuperando assim o vetor imagem 16x1 (matriz 4x4), Figura 4.8. Esse kernel transposto pode ser aprendido na convolução transposta, o que é uma vantagem para o método.

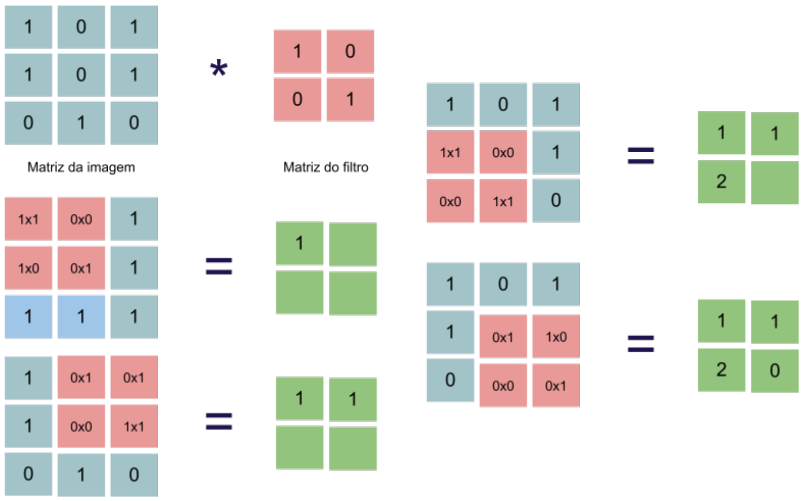
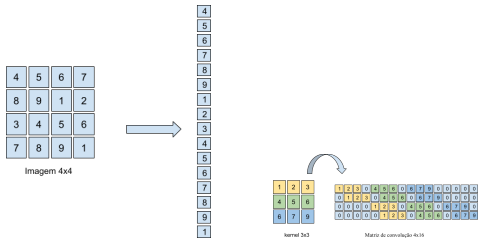


Figura 4.5: Exemplo do método de convolução.



(a) Achatamento da imagem de entrada.

(b) Transformação do Kernel de 3x3 para 4x16.

Figura 4.6

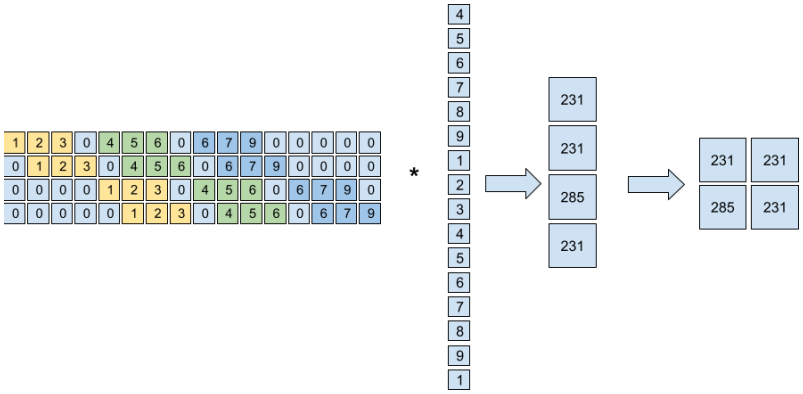


Figura 4.7: Convolução apresentada como multiplicação de matrizes.

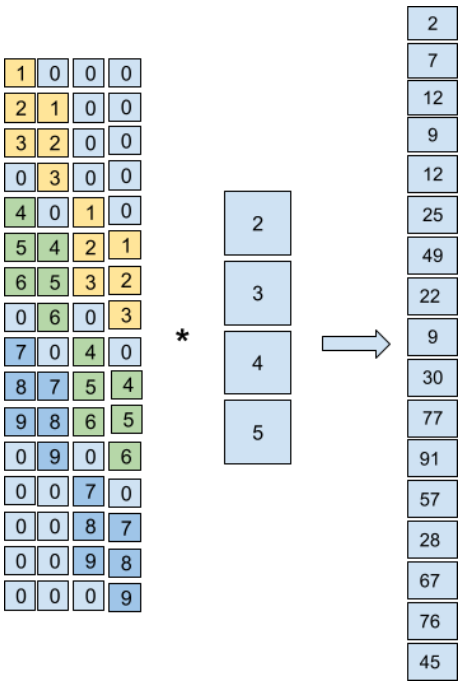


Figura 4.8: Exemplo de convolução transposta.

Capítulo 5

Experimentos

O problema abordado nesta dissertação é a separação supervisionada de fala, mais especificamente, a separação de uma vogal falada misturada a outro áudio artificialmente. Como o uso da máscara binária provou ser uma maneira muito eficiente de realizar essa separação, o trabalho foca em encontrar uma máscara para realizar a segregação dos sons.

5.1 Primeira abordagem

A primeira ideia que surgiu para encontrar a máscara binária seria reutilizar vogais pré existente nos dados de treino similares o suficiente para calcular a máscara binária para os dados de teste. Como existe o áudio de uma vogal sozinha, para os dados de treino, a máscara binária seria a melhor possível. Para realizar o experimento os dados misturados foram separados em 30% dados de teste e 70% dados de treino de forma totalmente aleatória.

O raciocínio para encontrar essa vogal similar foi inicialmente classificá-las em a, e, i, o, u para limitar a quantidade de vogais e usar algum outro método para encontrar a mais parecida possível. Para classificar as misturas foi usada uma rede neural convolucional, onde as entradas eram os MFCC dos áudios e a saída sua respectiva classificação.

Para gerar os MFCC foi usada a função librosa.feature.mfcc da biblioteca librosa do python. Os parâmetros usados foram fs=8000 (fs é a frequência de amostragem), n_mfcc=16, (número de coeficientes MFCC a serem retornados), n_fft=2048 (tamanho da janela usada no fft), hop_length=512 (distância entre janelas sucessivas).

A CNN usada possui quatro camadas. As três primeiras possuem a seguinte configuração: um layer de convolução de kernel 2x2, função de ativação ReLU e pool máximo de tamanho 2, com dropout de 20% para prevenir overfitting e a última camada teve como ativação o softmax. A biblioteca usada para o cálculo

da CNN foi a tensorflow 2.0, na linguagem Python, a função de perda usada foi a categorical cross entropy, a métrica usada foi a acurácia e o otimizador adam.

O primeiro experimento realizado foi classificar os áudios segundo as vogais que eles contêm. Na Tabela 5.1 temos a classificação sendo realizada com dois dados ligeiramente diferentes. Um banco de dados era das vogais puras, sem a música adicionada, e o outro é referente aos áudios de 2 segundos que são misturados com um trecho de música de 2 segundos.

	Acertos dados de Treino	Acertos dados de teste
Dados sem a música	83%	75%
Dados com a música	67%	56%

Tabela 5.1: Resultados do primeiro treino de classificação da CNN.

Após esse experimento os áudios de vogal pura foram reduzidos de maneira que eles agora só contêm o áudio da vogal. Para conseguir extrair somente vogal de cada um dos áudios foram utilizadas métricas de similaridade, pois como o áudio ou contém o som ou não contém, é esperado que as partes que tem o silêncio sejam similares entre si e as que tem o som sejam similares entre si também.

Por isso foi calculado uma matriz de similaridade dos MFCC usando a similaridade dos cossenos:

$$cossim(A, B) = \frac{A \mathrm{x} B}{||A|| \mathrm{x} ||B||} = \frac{\sum_{i=1}^{n} A_i \mathrm{x} B_i}{\sqrt{\sum_{i=1}^{n} A_i} \mathrm{x} \sqrt{\sum_{i=1}^{n} B_i}}. \tag{5.1}$$

E a outra usando a distância euclidiana:

$$d(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)}. \tag{5.2}$$

Essa matriz de similaridades fornece uma região cujo conteúdo é similar entre si. Como a maior parte do conteúdo presente no áudio é silêncio, então na menor região contínua dessa matriz, é esperado que seja a região em que a vogal está localizada. Ao encontrar essa menor região, é possível estimar os instantes de tempo que a vogal está contida. Foi usada a matriz de similaridade de cossenos para este cálculo, porque 1 diz que é muito similar e 0 que não tem nenhuma similaridade.

A partir da matriz de similaridade de cossenos, é possível encontrar os pontos de similaridade dos MFCC, nas figuras acima por exemplo a região começa no ponto 12 e vai até o ponto 17. Cada áudio de 2s após ser digitalizado com uma frequência de amostragem de 8kHz, possui 16000 amostras, onde para o cálculo dos MFCC foram utilizadas 2048 amostras por janela. Como o pulo entre cada janela são 512 amostras, logo o total de vetores dos MFCC são 32, $ceil(\frac{16000}{512}) = 32$. Cada vetor contém informação de 2048 amostras, logo cada vetor contém informação de

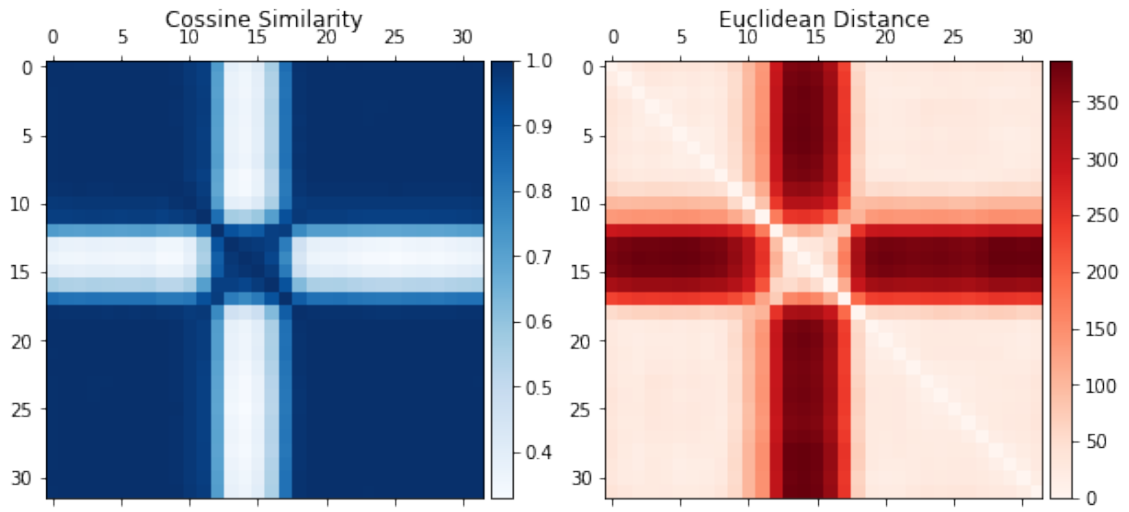


Figura 5.1: Matrizes de similaridade

256ms, período de 1 amostra $= \frac{1}{F_s} = \frac{1}{8000}$, logo $\delta_t = 2048\frac{1}{8000} = 256ms$. O primeiro vetor, v_1 abrange os instantes de 0 a 256ms, o segundo vetor, v_2 dos instantes 64ms ao 320ms, $v_3 = [128ms, 384ms]$, logo $v_n = [(n-1)64, 256 + (n-1)64]$, logo $v_{12} = [64 * (12-1), 256 + 64 * (12-1)] = [704, 960]$ e $v_{17} = [64 * (17-1), 256 + 64 * (17-1)] = [1024, 1280]$. Dado isso $t_0 = 693ms$ e $tf = 1280ms$.

Após cortar os áudios deixando majoritariamente as vogais faladas, a classificação foi realizada com os dados que não contêm pouco silêncio, Tabela 5.2. Foi utilizado um k-fold de tamanho 5, 50 épocas e tamanho de batch 20, dados de treino e teste na proporção de 70%,30% separados aleatoriamente.

Kfold	Acertos dados de Teste	Acertos dados de treino
1	78%	88%
2	94%	92%
3	91%	88%
4	88%	89%
5	89%	90%

Tabela 5.2: Resultados obtidos usando as vogais puras, com a menor quantidade de ruído possível.

É possível observar que os resultados obtidos são ligeiramente melhores que os obtidos com a vogal + silêncio. Um terceiro experimento foi realizado agora utilizando diferentes áudios misturados a vogal + silêncio, neste experimento foram utilizados a separação de 70% de dados para treino, 21% teste e 9% validação. A quantidade de épocas usadas foram 100 e o tamanho do batch 20.

A Figura 5.2 mostra curvas de aprendizado da CNN ao usar dados diferentes, é possível perceber que não era necessário o uso de 100 épocas e que em todos os casos a curva de aprendizado tem um comportamento semelhante.

A partir dessas CNN s treinadas, foi realizada a previsão dos mesmos dados

de vogal usados para treino, cuja única diferença era o ruído adicionado, a fim de verificar se na melhor das hipóteses a rede continuaria com uma boa previsão. A Figura 5.3 mostra essa relação, na coluna da esquerda estão os nomes dos áudios com suas respectivas misturas usados para o treino e na linha de baixo as misturas usadas para na previsão.

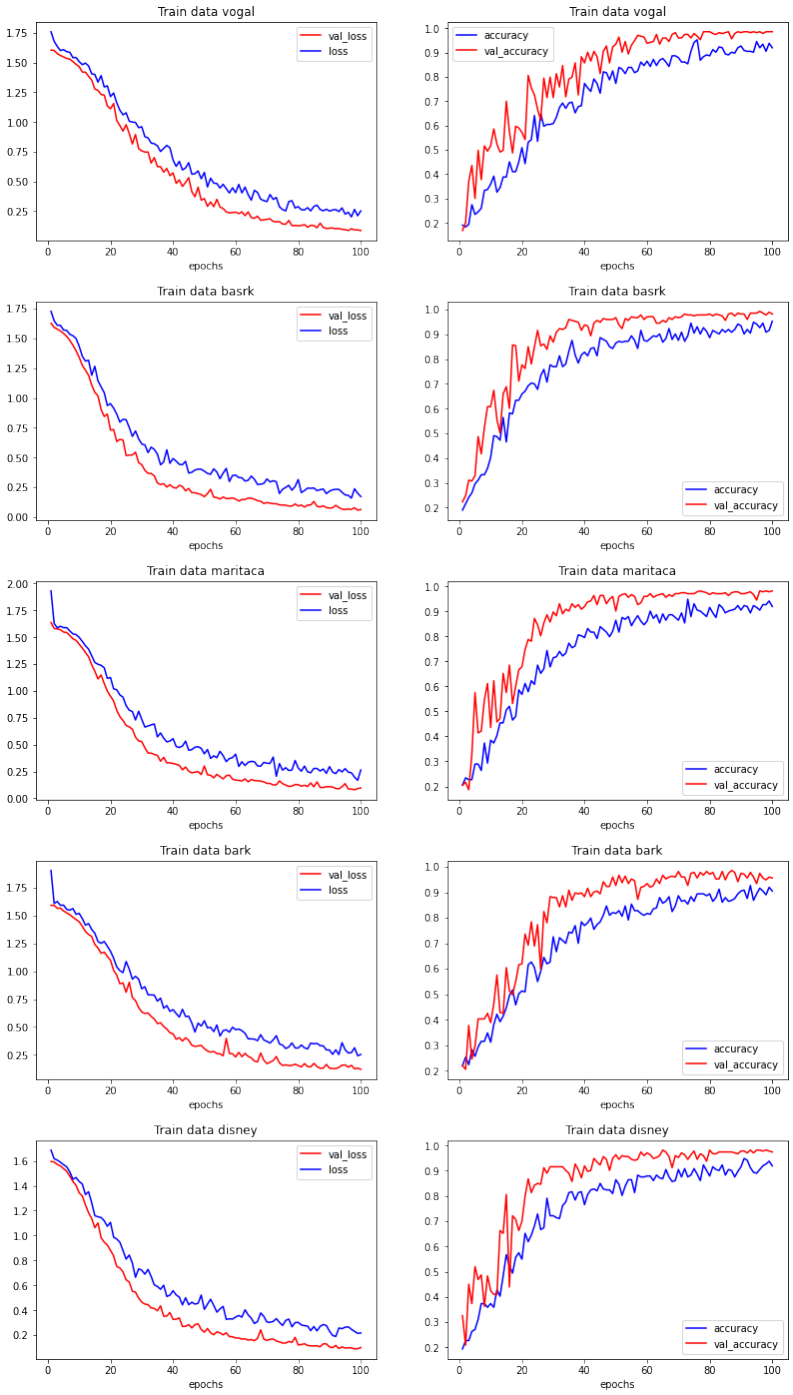


Figura 5.2: O gráfico da esquerda representa o valor de perda em relação a quantidade de épocas, onde a curva azul mostra a perda relativa aos dados de treino e a vermelha aos de validação. Já o gráfico da esquerda mostra a acurácia da classificação por épocas, onde a linha vermelha representa os dados de treino e a linha azul a acurácia relativa aos dados de validação.

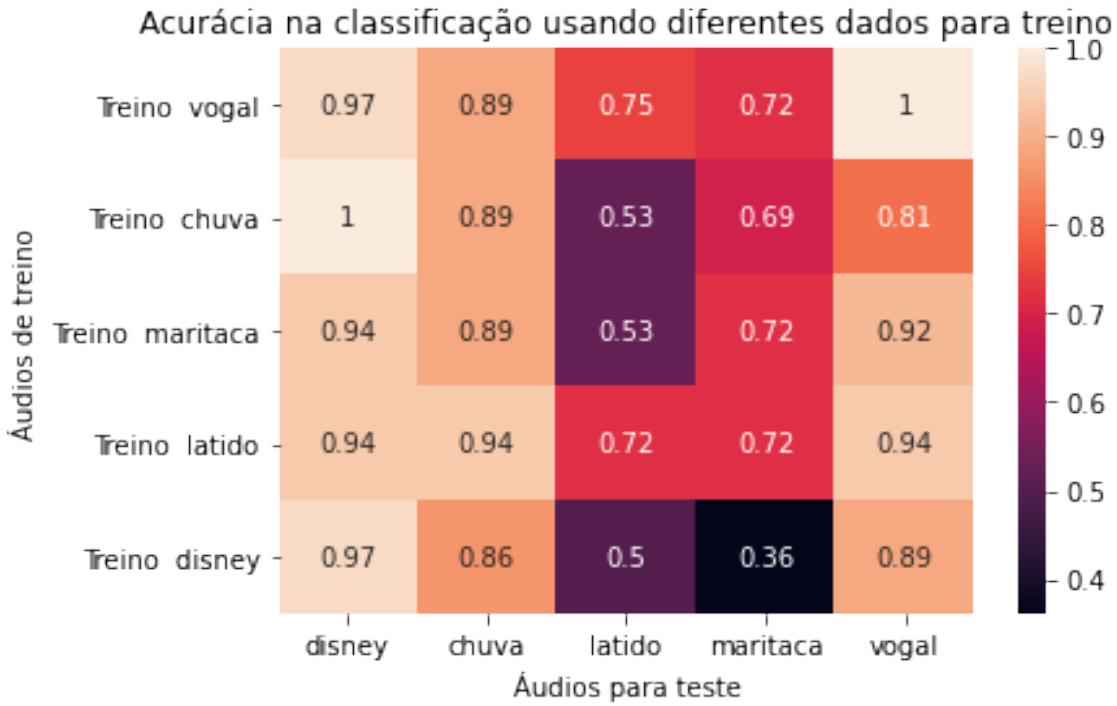


Figura 5.3: A tabela acima mostra como a acurácia quando usamos os mesmos áudios de vogais para o treino e para teste, cuja diferença são os ruídos adicionados.

A fim de conseguir extrair as vogal mais semelhante, seria necessário conseguir descobrir os instantes de tempo em que a vogal está presente no áudio, para isso foi tentado o seguinte método:

1. Os dados das vogais sem o silêncio foram separados em treino e teste, proporção de 70% para 30%.

2. Uma CNN com os mesmos parâmetros do primeiro experimento foi treinada usando esses dados. A partir dessa CNN treinada foi tentado encontrar o intervalo em que a vogal estava presente usando 100% do banco de dados de vogal + silêncio.

3. Para cada áudio, gerava uma previsão sobre sua vogal, após essa previsão era investigado se ela ficaria melhor ou pior caso zerasse o vetor mais a esquerda de coeficientes do MFCC , caso houvesse uma melhora nessa previsão o próximo vetor era zerado, caso houvesse uma piora, então assumia que aquele vetor era importante, então o mesmo processo era realizado na extremidade direita do MFCC , até encontrar o vetor que piorava.

4. Para retirar o intervalo era só pegar o primeiro vetor a esquerda que houvesse uma piora na classificação e o primeiro vetor a direita.

O método descrito apresenta algumas falhas e limitações, tais como a impossibilidade de extrair o conteúdo da fonte desejada quando os áudios contêm muitas frequências similares.

A tabela abaixo, mostra os resultados obtidos ao usar esse método. É possível observar que o método falha em encontrar os intervalos corretos na maioria dos casos, como esse método foi testado com todos os dados, então até mesmo nos áudios que são similares aos áudios que foram usados para o treinamento o método falha. A Tabela 5.3 mostra que em 44.75% dos casos, o intervalo extraído não contém parte da vogal.

	$Tc_0 < To_0$	$Tc_0 = To_0$	$Tc_0 > Tc_0$
$Tc_f < To_f$	$\sim 16.9\%$	$\sim 0.3\%$	0%
$Tc_f = To_f$	$\sim 19.68\%$	$\sim 1.02\%$	$\sim 1.78\%$
$Tc_f > To_f$	$\sim 31.44\%$	$\sim 2.81\%$	$\sim 26.07\%$

Tabela 5.3: Tabela que mostra o resultado geral dos tempos calculados onde a vogal começa e termina, Tc_0 significa, Tempo calculado de início da vogal, To_0 é o verdadeiro tempo de início da vogal, Tc_f é o tempo calculado em que a vogal termina, To_f é o tempo verdadeiro de témino da vogal.

5.2 Segunda abordagem

Dado que a abordagem usada no experimento anterior não obteve um resultado satisfatório, uma outra abordagem foi explorada.

Romain Hennequin et al. [18] lançaram uma nova ferramenta para auxiliar a separação musical. O modelo treinado consegue separar a música em: 2 faixas; vocal e acompanhamento, 4 faixas; vocal, bateria, baixo e acompanhamento e 5 faixas; vocal, bateria, baixo, piano/teclado e restante. A rede neural utilizada para realizar essa separação de faixas foi a U-Net de 12 camadas, 6 codificadoras e 6 decodificadoras, ela é usada para estimar uma máscara para cada faixa. A função de perda é a norma L1 entre espectrograma de entrada e a o espectrograma desejado de saída.

Essa segunda abordagem também utiliza a U-NeT, porém seu objetivo é gerar diretamente uma máscara binária para os áudios das misturas. Uma U-NeT de 10 camadas é usada para esse experimento, 5 de codificação e 5 de decodificação.

A U-Net original é baseada numa entrada de 512x512, nesse experimento como na entrada da rede foram usadas imagens geradas a partir dos espectrogramas, a entrada tem um tamanho diferente, que é o mesmo do espectrograma, 1024x64, cada pixel representa uma energia da janela $\Delta_t \Delta_\omega$.

Essas imagens estão em *grayscale*, de maneira que foi preservada a intensidade das frequência, onde quanto mais próximo de 1 for o valor do pixel, maior é a quantidade da frequência no espectrograma original. As máscaras binárias foram geradas da mesma maneira que a mostrada na Seção 3.4. A Figura 5.4 mostra um espectrograma que contém somente a vogal e na figura ao lado, sua respectiva máscara binária.

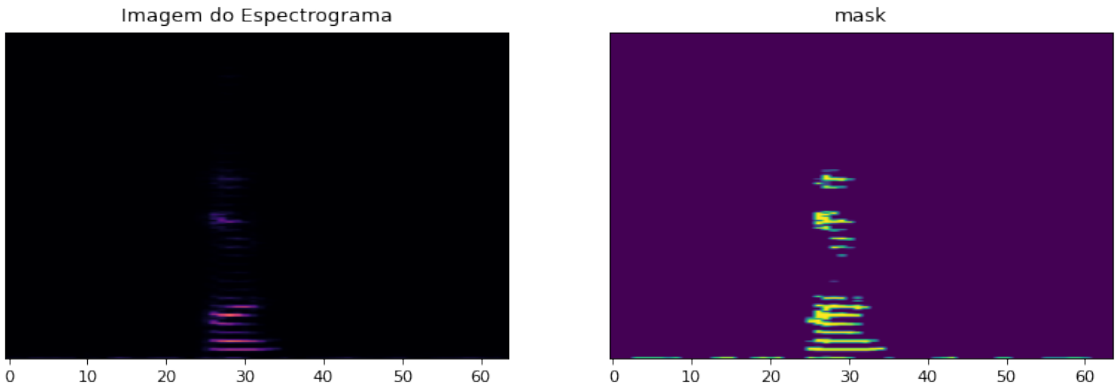


Figura 5.4: A imagem do lado esquerdo representa a imagem salva do espectrograma e a imagem do lado direito a sua respectiva máscara binária.

A métrica de acurácia usada foi a Interseção dentro da União (do termo em inglês, *Intersection over Union* (IoU)), essa métrica calcula a porcentagem da máscara

prevista pela rede neural que se intercepta com a máscara desejada. Quanto mais próxima elas se parecem, mais perto de 1 é o resultado, Figura 5.5. O objetivo de utilizar essa métrica é que é desejado que a máscara binária calculada seja a mais próxima da máscara binária real.

$$IoU = \frac{\text{Interseção}}{\text{União}}. \tag{5.3}$$

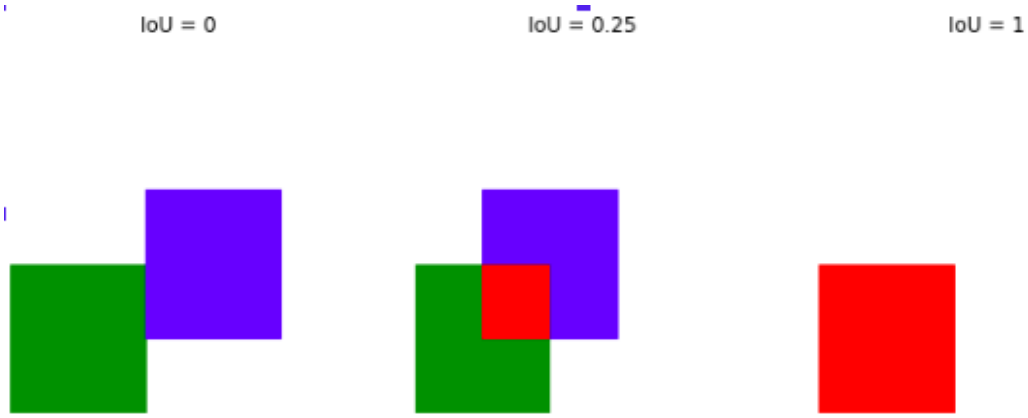


Figura 5.5: A imagem da esquerda mostra o caso em que não existe nenhum ponto interceptando as duas máscaras. A Figura do meio quando parte das máscaras se interceptam e a da direita quando elas são exatamente iguais.

Foram treinados alguns modelos, usando a proporção dos dados de 70% para treino, 21% validação e 9% teste.

Os primeiros testes foram realizados usando o mesmo tipo de ruído para todos os áudios, os gráficos nas Figuras 5.7 e 5.6 ilustram os dois melhores resultados.

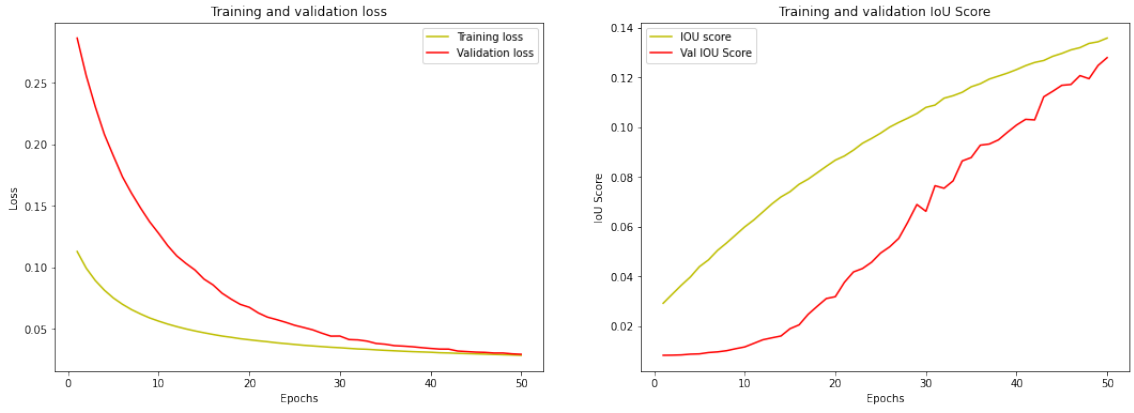


Figura 5.6: Otimizador: sgd, função de perda: sparse categorical cross entropy, na última camada a ativação é softmax, resultado nos dados de treino: perda: 0.032, e iou: 0.1241 .

Já que usando o otimizador Adam a função de perda binary cross entropy, com a última camada ativação sendo a função sigmoid e 100 épocas foi obtido o melhor

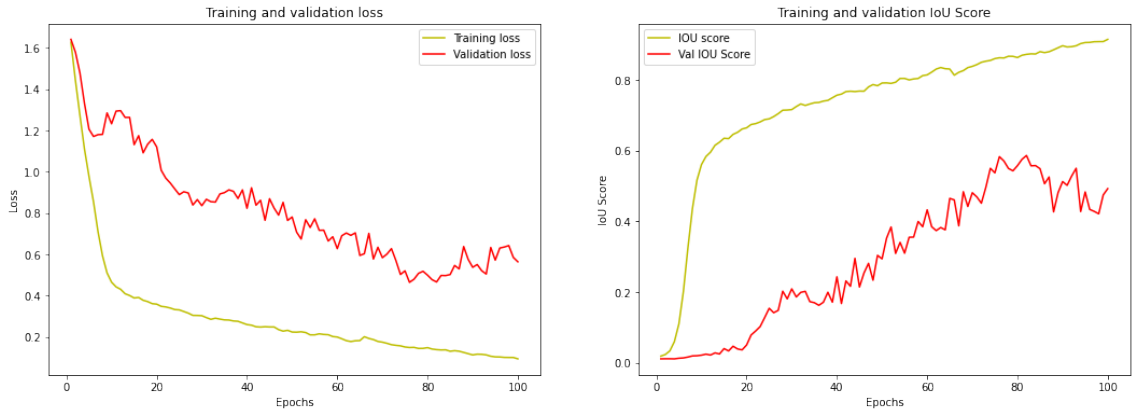


Figura 5.7: otimizador: Adam, função de perda: binary cross entropy, na última camada ativação: sigmoid, perda: 0.5865 - iou_score: 0.478, epochs : 100

resultado então essa mesma configuração foi usada para treinar U-Net com dados com ruídos diferentes, como mostrado na Tabela 5.4

	Loss	IoU score
Latido	1.6176	0.0922
Chuva	1.2296	0.1446
Instrumental	1.2303	0.1419
Maritaca	1.3613	0.1089

Tabela 5.4: Resultados obtidos para ruídos diferentes.

Os gráficos apresentados em 5.8 mostram a curva de aprendizado da U-Net quando treinada com ruídos diferentes.

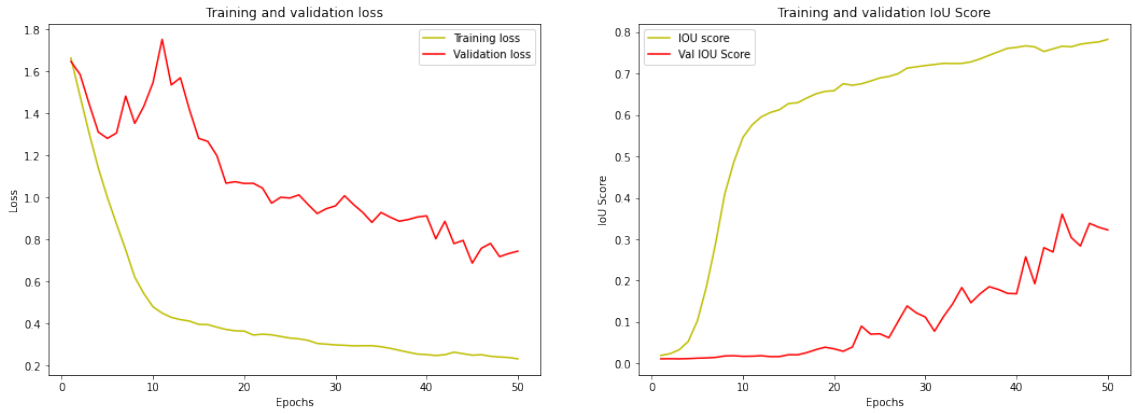
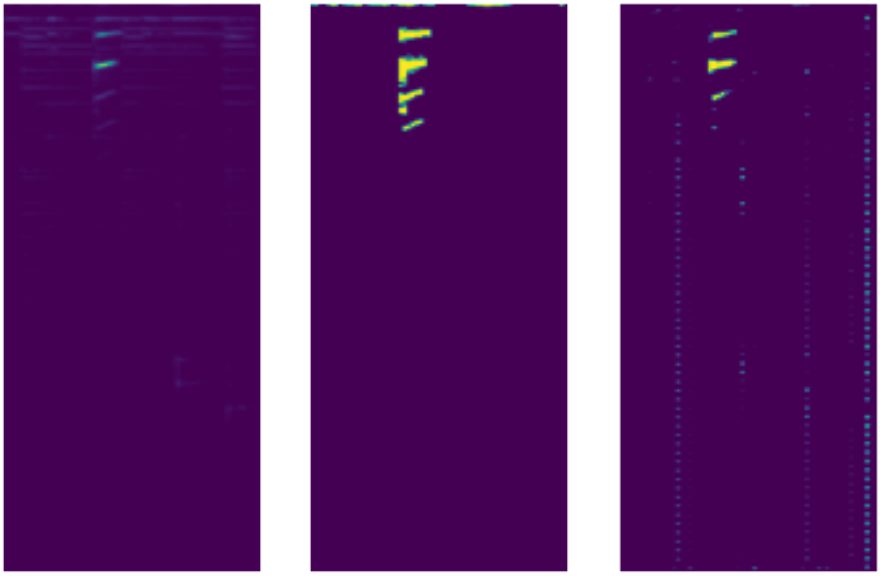


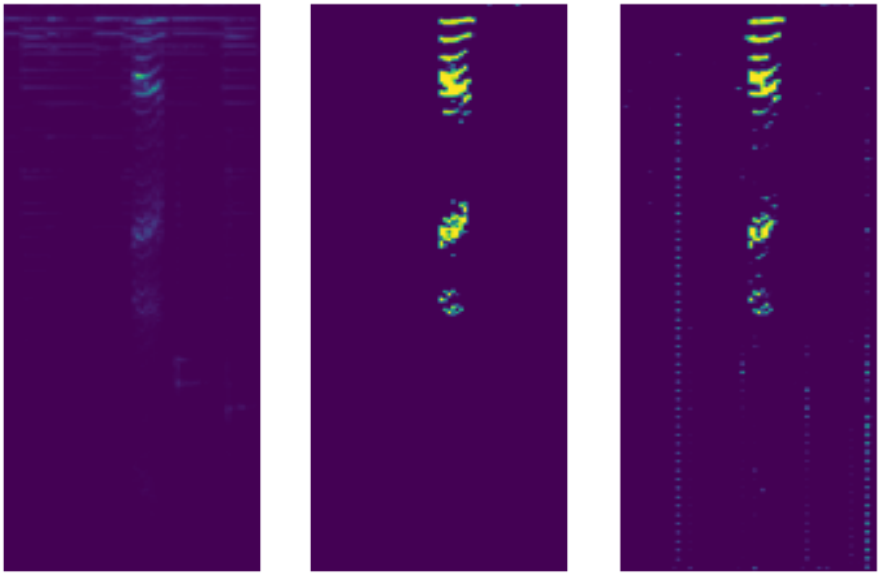
Figura 5.8: Curva de aprendizado da U-Net usando múltiplos ruídos juntos.

As Figuras 5.9 mostram as algumas máscaras binárias dos dados de teste geradas pela UNeT. Nessas figuras o espectrograma do áudio com a mistura adicionada é representado pelas figuras na coluna a esquerda, a máscara desejada, que é a calculada diretamente do espectrograma da vogal pura é representada pela figuras

no meio e na coluna da direita contém as máscaras geradas pela U-Net.



(a) Exemplo de resultado



(b) Exemplo de resultado

Figura 5.9: Dados de teste e seus respectivos resultados, na extrema esquerda, o espectrograma do áudio + ruído, no meio a máscara real e na direita á máscara gerada.

Um último experimento foi realizado a fim de verificar como a rede geraria as máscaras caso os dados de entrada houvessem múltiplos ruídos. Para isso foi sorteado aleatoriamente qual dos ruídos seria adicionado, chuva, latido, música instrumental ou barulho de maritacas aos dados de treino. Para cada dado de treino um novo sorteio acontecia. A partir desses dados e suas respectivas máscaras binárias a U-Net foi treinada.

Com essa U-Net foi feita a previsão em diferentes banco de dados. Todos os testes a seguir utilizaram os mesmos dados de teste que o treinado pela rede. A Tabela 5.5 mostra o resultado de IoU Score quando usados dados diferentes para gerar as máscaras. Os dados sem ruído são relativos aos dados vogal+silêncio. Os dados com ruídos diferentes foram gerados da mesma maneira que os dados de treinamento desse experimento, os com latido,chuva, instrumental e maritaca, são os dados das vogais misturadas artificialmente com latido de cachorro, chuva, música instrumental e barulho de maritacas, respectivamente.

É possível perceber pela Tabela 5.5 que a pontuação de IoU varia bastante dependendo dos dados usados para treino, mostrando que o método não é muito robusto. Apesar de não ter havido uma preocupação no balanceamento dos ruídos utilizados, há um comportamento muito similar no score dos ruídos únicos. O IoU score de ruídos diferentes é inferior a sem ruído, para melhorar talvez fosse necessário aumentar o banco de dados. Um experimento que não foi investigado é quando todos os áudios de treino possuem um ruído diferente, a partir dos experimentos propostos provavelmente a rede teria um desempenho ruim. Uma maneira de tentar melhorar o desempenho da rede teria sido deixar as proporções das imagens de entrada bem similares, como no experimento original.

	Loss	IoU score
sem ruído	0.6135	0.4632
ruídos diferentes	0.8867	0.2625
latido	1.0168	0.1757
chuva	1.0251	0.1743
instrumental	1.0306	0.1677
maritaca	1.0120	0.1840

Tabela 5.5: Resultados obtidos usando dados de treino e validação com diferentes tipos de ruídos diferentes

Capítulo 6

Conclusão

Nessa dissertação foram testados dois modelos para a separação de sons que usaram métodos muito difundidos no campo de visão computacional. Isso só é possível por causa da imagem da representação tempo-frequência de um áudio. Ao mudar o domínio tempo para TF e vice-versa não acarreta em mudanças significativas no conteúdo do sinal.

Os áudios que esse trabalho utiliza são áudios muito simples e controlados, para que possuam a menor quantidade de ruído e informação possível, por isso as vogais foram escolhidas. Os ruídos presentes nos áudios foram adicionados artificialmente após a coleta dos dados, gerando uma mistura controlada e artificial para gerar os experimentos.

O primeiro modelo proposto foi uma junção de CNN com MFCC , onde primeiro foi feita uma classificação e após uma tentativa de separação do sinal. Apesar da classificação dos sons do algoritmo utilizado ter tido uma eficácia boa, a separação dos sons mostrou-se insatisfatória sendo necessário buscar outro modelo.

A segunda solução proposta foi trabalhar com a U-NeT juntamente com uma máscara binária. O conceito de máscara binária é bastante utilizado para segmentação de imagens, um campo muito explorado que já possui algoritmos mais recentes que a U-NeT. Quando a máscara binária é utilizada para os áudios, é necessário um threshold para criá-la, assemelhando-se muito à geração de máscaras binárias usando intensidade do pixel, método que é pouco utilizado na literatura mais recente.

A partir dessas IBMs criadas e os espectrogramas dos áudios, foi possível gerar um banco de dados para o treinamento supervisionado para a rede. Os resultados do segundo método superaram os resultados do primeiro modelo sugerido, mas ainda não foi um resultado muito satisfat ório. Outras abordagens poderiam ter sido utilizadas como tentar gerar diretamente o espectrograma.

Referências Bibliográficas

[1] MCDERMOTT, J. H. "The cocktail party problem", *Current Biology*, v. 19, n. 22, pp. R1024–R1027, 2009.

[2] BREGMAN, A. "Auditory Scene Analysis: The Perceptual Organization of Sound". v. 95, 01 1990. doi: 10.1121/1.408434.

[3] BROWN, G. J., COOKE, M. "Computational auditory scene analysis", *Computer Speech Language*, v. 8, n. 4, pp. 297–336, 1994.

[4] FLANAGAN, J. L. "Speech Analysis, Synthesis and Perception". 1971.

[5] MÜLLER, M. *Fundamentals of Music Processing*. Springer, Cham, 2015.

[6] WANG, D., CHEN, J. "Supervised Speech Separation Based on Deep Learning: An Overview", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. PP, 08 2017.

[7] OPPENHEIM, A., SCHAFER, R. "From frequency to quefrency: a history of the cepstrum", *IEEE Signal Processing Magazine*, v. 21, n. 5, pp. 95–106, 2004.

[8] BRYAN PARDO, ZAFAR RAFII, Z. D. *Audio Source Separation in a Musical Context*. Springer, 2018.

[9] GOLDBERG, M. B. E. *Introduction to Digital Audio Coding and Standards*. Springer, 2003.

[10] MERTINS, A., MERTINS, D. "Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications", 11 2001.

[11] BOLL, S. "Suppression of acoustic noise in speech using spectral subtraction", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 27, n. 2, pp. 113–120, 1979.

[12] WANG, Y., WANG, D. "Towards Scaling Up Classification-Based Speech Separation", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 21, n. 7, pp. 1381–1390, 2013.

[13] XUEDONG HUANG, ALEX ACERO, H.-W. H. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Pearson, 2001.

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

[15] ACADEMY, D. S. "Deep Learning Book". 2021. Disponível em: <`https://www.deeplearningbook.com.br/`>.

[16] RONNEBERGER, O., FISCHER, P., BROX, T. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015.

[17] DUMOULIN, V., VISIN, F. "A guide to convolution arithmetic for deep learning", *ArXiv*, v. abs/1603.07285, 2016.

[18] HENNEQUIN, R., KHLIF, A., VOITURET, F., et al. "SPLEETER: A FAST AND STATE-OF-THE ART MUSIC SOURCE SEPARATION TOOL WITH PRE-TRAINED MODELS". 2019.