

**UNIVERSIDADE DE SÃO PAULO**

Instituto de Ciências Matemáticas e de Computação

**Aprendizado de representações com Redes Convolucionais  
para a identificação de espécies de pássaros e anuros em  
Paisagens Acústicas**

**Fábio Felix Dias**

Tese de Doutorado do Programa de Pós-Graduação em Ciências de  
Computação e Matemática Computacional (PPG-CCMC)



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

**Fábio Felix Dias**

**Aprendizado de representações com Redes Convolucionais  
para a identificação de espécies de pássaros e anuros em  
Paisagens Acústicas**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Rosane Minghim

Coorientador: Prof. Dr. Moacir Antonelli Ponti

**USP – São Carlos  
Dezembro de 2022**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados inseridos pelo(a) autor(a)

D541a      Dias, Fábio Felix  
            Aprendizado de representações com Redes  
            Convolucionais para a identificação de espécies de  
            pássaros e anuros em Paisagens Acústicas / Fábio  
            Felix Dias; orientadora Rosane Minghim;  
            coorientador Moacir Antonelli Ponti. -- São Carlos,  
            2022.  
            175 p.

            Tese (Doutorado - Programa de Pós-Graduação em  
            Ciências de Computação e Matemática Computacional) --  
            Instituto de Ciências Matemáticas e de Computação,  
            Universidade de São Paulo, 2022.

            1. Rede Neural Convolucional. 2. Indentificação  
            de Sons. 3. Paisagens Acústicas. I. Minghim,  
            Rosane, orient. II. Ponti, Moacir Antonelli,  
            coorient. III. Título.



**Fábio Felix Dias**

**Learning representations with Convolutional Networks to  
identify bird and anuran species in Soundscapes**

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Rosane Minghim

Co-advisor: Prof. Dr. Moacir Antonelli Ponti

**USP – São Carlos  
December 2022**



*Este trabalho é dedicado a quem não pôde aprender ler e escrever.*



# AGRADECIMENTOS

---

---

Inicialmente, e sobretudo, aos meus pais que sempre acreditaram e investiram na minha formação mesmo com todas as dificuldades impostas por uma renda apertada. Aos colegas do laboratório e do instituto que ajudaram direta ou indiretamente nesta pesquisa. Aos integrantes do LEEC/UNESP pelos dados e auxílios diversos. Ao Prof. Dr. Moacir Antonelli Ponti e à Profa. Dra. Rosane Minghim que orientaram o desenvolvimento desta ideia. À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo financiamento durante este doutorado. Por fim, ao Nosso Senhor Jesus Cristo, que ampara, concede ânimo e coragem para enfrentar as diversas dificuldades do caminho, através da interseção de sua mãe Maria.



*“Não há lugar para a sabedoria onde não há paciência.”  
(Agostinho de Hipona)*





# RESUMO

DIAS, F. F. **Aprendizado de representações com Redes Convolucionais para a identificação de espécies de pássaros e anuros em Paisagens Acústicas**. 2022. 175 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

A análise de Paisagens Acústicas desperta grande interesse na comunidade científica como ferramenta para auxiliar a tomada de decisões relacionadas ao monitoramento e entendimento de questões ambientais. Por exemplo, análises da diversidade e do comportamento de espécies animais, podem ajudar na compreensão do estado do ambiente onde essas espécies são encontradas. Essas análises utilizam áudios gravados de maneira autônoma em ambientes diversos, técnica que diminui custos, aumenta a capacidade de análise e diminui a influência externa nesses ambientes. Entretanto, o aumento da quantidade de gravações gera desafios para a exploração e extração de conhecimento desses dados. Nesse cenário, técnicas como Redes Neurais Convolucionais são empregadas, com resultados relevantes, para ajudar os pesquisadores em tarefas de detecção e identificação de espécies, por exemplo. Essas técnicas precisam lidar com problemas recorrentes de sons coletados em ambientes naturais e não controlados, como variação dos padrões sonoros, sobreposição de sinais e ruídos diversos. Esta pesquisa de doutorado traçou um caminho para melhorar a aplicação de redes neurais na identificação de espécies de pássaros e anuros, em sons coletados em ambientes naturais. A abordagem proposta investigou sobretudo maneiras de regularização da função de custo da rede com técnicas de quantificação; combinações de entradas para as redes, como variações de espectrogramas, características acústicas e informações sobre as gravações; e abordagens de Aprendizado Autossupervisionado para pré-treinamento das arquiteturas de rede. Com uma quantidade reduzida de amostras para treinamento, essas abordagens obtiveram resultados superiores aos de um classificador linear que usa características acústicas como entrada, melhoraram a segregação dos espaços de características em níveis distintos, incrementaram sobretudo os resultados de redes simples e alcançaram resultados próximos aos de técnicas supervisionadas empregadas para o pré-treinamento.

**Palavras-chave:** identificação de sons, quantificação, combinação de entradas, autossupervisão.



# ABSTRACT

DIAS, F. F. **Learning representations with Convolutional Networks to identify bird and anuran species in Soundscapes**. 2022. 175 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Soundscape analysis makes the scientific community interested in it as a tool to aim decision-making related to monitoring and understanding ecological questions. For instance, analysis of diversity and animal behavior can help to understand landscape health. These analyses use autonomous recorders to capture sounds from several landscapes, a technique that diminishes costs, enhances analytical capabilities, and lessens habitat disorders generated by human presence. Nevertheless, the massive amount of recordings to perform the analyses yields challenges to knowledge extraction. In this case, Convolutional Neural Networks are employed to help researchers to detect and identify animal species, for instance. These tools have to deal with issues related to sounds captured “in the wild”, such as sound variation, pattern overlap, and multiple sources of noise. As a result, this Ph.D. research constructed a path to improve the applicability of neural networks to identify birds and anuran species, inside recordings collected in natural environments. The proposed approach explored mainly the regularization of the loss function with quantification techniques; input combinations to feed networks, such as spectrogram variations, acoustic features, and recording information; and Self-supervised Learning to pretrain network architectures. In a scenario with few data samples, these approaches achieved better results than a linear classifier with acoustic features as input, improved with distinct levels the segregation of the embeddings, increased mainly the results of simple networks, and reached results close to supervised techniques used to pretrain neural networks.

**Keywords:** sound identification, quantification, input combination, self-supervised.



---

# LISTA DE ILUSTRAÇÕES

---

---

Figura 1 – Exemplos de dados uni e bidimensionais. . . . .	36
Figura 2 – Exemplo de Transformada de Fourier. . . . .	38
Figura 3 – Exemplos de variações de espectrograma . . . . .	40
Figura 4 – Banco de filtros para criação do mel-espectrograma . . . . .	41
Figura 5 – Exemplo de convolução em uma dimensão (1-D). . . . .	43
Figura 6 – Exemplo de matriz de confusão. . . . .	49
Figura 7 – Representação básica de uma RNA . . . . .	51
Figura 8 – Arquitetura de uma CNN aplicada em classificação de imagens. . . . .	55
Figura 9 – Visão geral de blocos básicos de arquiteturas de CNN. . . . .	56
Figura 10 – Estrutura básica da arquitetura usada na Barlow Twins e VICReg . . . . .	60
Figura 11 – Sumarização visual da classificação de baleias jubarte . . . . .	77
Figura 12 – Resultados de consultas em base de áudios . . . . .	78
Figura 13 – Projeção t-SNE de espaço de características . . . . .	80
Figura 14 – Comparação de resultados com <i>baseline</i> do DCLDE . . . . .	83
Figura 15 – Estrutura básica dos componentes explorados na pesquisa. . . . .	88
Figura 16 – Exemplo de quantificação em RNA . . . . .	96
Figura 17 – Arquitetura de uma CNN simples . . . . .	100
Figura 18 – Valores individuais da acurácia balanceada para 2-class e 12-class . . . . .	104
Figura 19 – Valores individuais da acurácia balanceada para anuran-class e bird-class . . . . .	106
Figura 20 – Matrizes de confusão para o cenário 12-class . . . . .	108
Figura 21 – Projeções dos espaços de características do cenário <b>bird-class</b> . . . . .	109
Figura 22 – Projeções dos espaços de características do cenário <b>12-class</b> . . . . .	110
Figura 23 – Curvas de aprendizagem para modelos no cenário <b>2-class</b> . . . . .	111
Figura 24 – Curvas de aprendizagem para modelos no cenário <b>12-class</b> . . . . .	112
Figura 25 – Passos e escolhas para treinamento de CNN . . . . .	118
Figura 26 – Arquiteturas para combinação de características . . . . .	124
Figura 27 – Valores individuais da acurácia balanceada para combinação de entradas . . . . .	129
Figura 28 – Curvas de aprendizagem para as melhores combinações de características . . . . .	133
Figura 29 – Inicialização de pesos de CNN e sua avaliação . . . . .	138
Figura 30 – Valores individuais da acurácia balanceada para autossupervisão . . . . .	145
Figura 31 – Similaridade CKA dos modelos testados . . . . .	147
Figura 32 – Projeção dos dados de teste separados em 12 espécies . . . . .	148
Figura 33 – Projeção dos dados de teste separados em pássaros e anuros . . . . .	149



# LISTA DE QUADROS

---

---

Quadro 1 – Sumarização dos artigos e suas principais características . . . . .	85
Quadro 2 – Parâmetros usados para encerrar o treinamento e salvar os modelos .	101
Quadro 3 – Parâmetros das funções para aumento de dados . . . . .	139
Quadro 4 – Extensão dos trabalhos relacionados . . . . .	157





# LISTA DE ALGORITMOS

---

---

Algoritmo 1 – Criação de espectrograma com múltiplas visões . . . . .	79
---	----



# LISTA DE TABELAS

---

---

Tabela 1 – Espécies de <b>pássaros</b> e exemplos de seus padrões sonoros. . . . .	90
Tabela 2 – Espécies de <b>anuros</b> e exemplos de seus padrões sonoros. . . . .	91
Tabela 3 – Base de dados original . . . . .	97
Tabela 4 – Parâmetros das funções para aumento de dados . . . . .	99
Tabela 5 – Acurácia balanceada dos modelos nos cenários 2-class e 12-class . . . .	103
Tabela 6 – Acurácia balanceada dos modelos nos cenários anuran-class e bird-class	105
Tabela 7 – Valores de sensibilidade gerados pelas RNAs no cenário 12-class . . . .	107
Tabela 8 – Resultados adicionais para quantificação . . . . .	112
Tabela 9 – Resumo dos resultados de quantificação . . . . .	113
Tabela 10 – Base de áudios combinada com o AudioSet . . . . .	119
Tabela 11 – Resultados de CNNs combinando diferentes entradas . . . . .	129
Tabela 12 – Resultados de CNNs combinando diferentes imagens . . . . .	131
Tabela 13 – Resultados de CNNs com quantificação . . . . .	131
Tabela 14 – Melhores resultados obtidos com combinação de entradas e quantificação	132
Tabela 15 – Resultados dos modelos pré-treinados . . . . .	144
Tabela 16 – Variação do tamanho de base de refinamento . . . . .	150
Tabela 17 – Variação do tamanho de base de pré-treinamento . . . . .	150
Tabela 18 – Variação do espaço gerado pela VICReg . . . . .	151



# LISTA DE ABREVIATURAS E SIGLAS

---

---

$H_f$	<i>Frequency Entropy</i>
$H_t$	<i>Temporal Entropy</i>
ACC	<i>adjusted classify and count</i>
ACI	<i>Acoustic Complexity Index</i>
ADA	<i>Adaptative Data Augmentation</i>
Adam	<i>Adaptive Moment Estimation</i>
ADI	<i>Acoustic Diversity Index</i>
AEI	<i>Acoustic Evenness Index</i>
ANAFCC	<i>American Northeast Avian Flight Call Classification</i>
AP	<i>average precision</i>
AR	<i>Acoustic Richness</i>
AT	<i>Adaptative threshold</i>
AW	<i>Adaptative weights</i>
BGN	<i>Background noise index</i>
Bio	<i>Bioacoustic Index</i>
CA-CNN	<i>context-adaptive neural network</i>
CC	<i>classify and count</i>
CCB	<i>Center for Conservation Bioacoustics</i>
CKA	<i>Centered Kernel Alignment</i>
cmAP	<i>class-wise mAP</i>
CNN	<i>Redes Neurais Convolucionais - Convolutional Neural Networks</i>
DCASE	<i>Detection and Classification of Acoustic Scenes and Events</i>
DCLDE	<i>International Workshop on Detection, Classification, Localization, and Density Estimation of Marine Mammals using Passive Acoustics</i>
DCT	<i>Transformada Discreta de Cosseno - Discrete Cosine Transform</i>
GDA	<i>Geometrical Data Augmentation</i>
GRU	<i>Gated Recurrent Unit</i>
H	<i>Acoustic Entropy Index</i>
HPSS	<i>harmonic percussive source separation</i>
Hz	<i>hertz</i>
IIR	<i>Infinite Impulse Response</i>

IMC	Índice de massa corporal
LDA	<i>Linear Discriminant Analysis</i>
M	<i>Median of Amplitude Envelope</i>
mAP	<i>mean average precision</i>
mba	<i>mean balanced accuracy</i>
MCP	McCulloch e Pitts
MERIDIAN	<i>Marine Environmental Research Infrastructure for Data Integration and Application Network</i>
MFCC	Coeficientes Mel-Cepstrais - <i>Mel-frequency Cepstrum Coefficients</i>
ML	Aprendizagem de Máquina - <i>Machine Learning</i>
MLP	<i>Multilayer Perceptron</i>
MoE	<i>Mixture of Experts</i>
NDSI	<i>Normalized Difference Soundscape Index</i>
NOAA	<i>U.S. National Oceanic and Atmospheric Administration</i>
NP	<i>Number of Peaks</i>
PAM	Monitoramento Acústico Passivo - <i>Passive Acoustic Monitoring</i>
PCA	<i>Principal Component Analysis</i>
PCEN	<i>Per-channel Energy Normalization</i>
PCM	<i>Pulse Code Modulation</i>
PSD	Densidade espectral - <i>Power Spectral Density</i>
ReLU	<i>rectified linear unit</i>
RMS	<i>Root Mean Square</i>
RMSProp	<i>Root Mean Square Propagation</i>
RNA	Redes Neurais Artificiais
ROC	<i>Receiver Operating Characteristic</i>
SGD	<i>Stochastic Gradient Descent</i>
SNR	<i>Signal-to-noise Ratio</i>
SONYC	<i>Sounds of New York City</i>
SPL	<i>Sound Pressure Level</i>
STFT	Transformada breve ou de tempo curto de Fourier - <i>Short-term Fourier Transform</i>
SVM	Support Vector Machine
SVM	<i>Support Vector Machine</i>
VICReg	<i>Variance-Invariance-Covariance Regularization</i>
WAV	<i>WAVEform</i>

---

# LISTA DE SÍMBOLOS

---

---

$\mathbb{C}$  — Conjunto dos números complexos

$\mathbb{R}$  — Conjunto dos números reais

$\mathbb{Z}$  — Conjunto dos números inteiros

$*$  — Operador de convolução





# SUMÁRIO

---

---

1	<b>INTRODUÇÃO</b>	31
1.1	<b>Objetivo</b>	33
1.2	<b>Contribuições e resultados obtidos</b>	34
1.3	<b>Organização do texto</b>	34
2	<b>CONCEITOS FUNDAMENTAIS</b>	35
2.1	<b>Dados multidimensionais</b>	35
2.2	<b>Processamento de sinais digitais</b>	37
2.2.1	<i>Geração de espectrogramas</i>	38
2.2.2	<i>Medidas para análise de sinais</i>	41
2.2.3	<i>Convolução</i>	42
2.3	<b>Análise de Paisagens Acústicas</b>	44
2.3.1	<i>Representação de sons ambientais</i>	45
2.4	<b>Aprendizagem de Máquina</b>	46
2.4.1	<i>Problemas de generalização</i>	48
2.4.2	<i>Avaliação de desempenho de métodos de classificação</i>	49
2.4.3	<i>Redes Neurais Artificiais</i>	51
2.4.3.1	<i>Redes Neurais Convolucionais</i>	53
2.4.3.2	<i>Exemplos de arquiteturas</i>	55
2.4.3.3	<i>Avaliação da estrutura das redes</i>	57
2.4.4	<i>Técnicas para melhoria do aprendizado</i>	58
2.4.5	<i>Aprendizado Autossupervisionado</i>	59
2.4.6	<i>Quantificação</i>	61
2.5	<b>Considerações finais</b>	63
3	<b>TRABALHOS RELACIONADOS</b>	65
3.1	<b>Classificação de sons de pássaros</b>	65
3.2	<b>Classificação de sons de anuros</b>	72
3.3	<b>Classificação de sons de pássaros e de anuros</b>	75
3.4	<b>Classificação de sons em ambiente marinho</b>	76
3.5	<b>Considerações sobre os trabalhos apresentados</b>	83

<b>4</b>	<b>LINHAS GERAIS PARA O TREINAMENTO DE REDES CONVOLUCIONAIS NA IDENTIFICAÇÃO DE VOCALIZAÇÕES DE ANIMAIS</b>	<b>87</b>
4.1	Arquivos de áudio	88
4.1.1	<i>Espécies de interesse</i>	88
4.1.2	<i>Balanceamento de classes e variação de padrões</i>	89
4.2	Representação dos padrões de som	89
4.3	Arquitetura de rede	92
4.4	Estratégias para treinamento	92
4.5	Testes e avaliação	93
4.6	Considerações finais	94
<b>5</b>	<b>QUANTIFICAÇÃO COMO REGULARIZADOR DE TREINAMENTO</b>	<b>95</b>
5.1	Metodologia aplicada	96
5.1.1	<i>Conjunto de dados</i>	97
5.1.2	<i>Aumento de dados</i>	98
5.1.3	<i>Extração de características</i>	98
5.1.4	<i>Baseline</i>	99
5.1.5	<i>Espectrogramas</i>	99
5.1.6	<i>Arquiteturas das redes neurais</i>	100
5.1.7	<i>Avaliação</i>	101
5.2	Resultados	102
5.2.1	<i>Classificação binária e classificação de pássaros e anuros</i>	103
5.2.2	<i>Classificação de pássaros e classificação de anuros</i>	104
5.2.3	<i>Análise da classificação de pássaros e anuros</i>	105
5.2.4	<i>Análise de convergência dos treinamentos</i>	107
5.2.5	<i>Resultados adicionais</i>	109
5.3	Discussão	111
5.4	Considerações finais	114
<b>6</b>	<b>COMBINAÇÃO DE ENTRADAS PARA REDE CONVOLUCIONAL</b>	<b>117</b>
6.1	Metodologia aplicada	118
6.1.1	<i>Conjunto de dados</i>	118
6.1.2	<i>Balanceamento de classes</i>	120
6.1.3	<i>Características manuais</i>	120
6.1.4	<i>Definição de um baseline</i>	121
6.1.5	<i>Tipos de representação espectral utilizados</i>	122
6.1.6	<i>Arquiteturas e suas variações</i>	122
6.1.6.1	<i>Combinando espectrogramas e características manuais</i>	124
6.1.6.2	<i>Combinando variações de espectrograma</i>	125

6.1.6.3	<i>Combinando variações de espectrograma e características manuais</i>	126
6.1.6.4	<i>Quantificação</i>	126
6.1.7	<b>Avaliação</b>	126
6.2	<b>Resultados</b>	127
6.2.1	<i>Combinando espectrogramas e características manuais</i>	127
6.2.2	<i>Combinando variações do espectrograma e características manuais</i>	128
6.2.3	<i>Quantificação</i>	130
6.2.4	<i>Comparação dos melhores resultados</i>	131
6.2.5	<i>Avaliação de curvas de aprendizagem</i>	132
6.3	<b>Discussão</b>	132
6.4	<b>Considerações finais</b>	135
7	<b>AVALIAÇÃO DE AUTOSSUPERVISÃO PARA PADRÕES SONO-ROS</b>	137
7.1	<b>Metodologia aplicada</b>	138
7.1.1	<i>Conjunto de dados</i>	138
7.1.2	<i>Aumento de dados</i>	139
7.1.3	<i>Espectrogramas</i>	140
7.1.4	<i>Arquiteturas das redes neurais</i>	140
7.1.5	<i>Tarefas de autossupervisão</i>	141
7.1.6	<i>Avaliação</i>	142
7.2	<b>Resultados</b>	143
7.2.1	<i>Análise da estrutura de similaridade das redes</i>	145
7.2.2	<i>Análise visual dos espaços de características aprendidos</i>	146
7.2.3	<i>Variação das quantidades de amostras</i>	147
7.2.4	<i>Variação do espaço gerado pela autossupervisão</i>	150
7.3	<b>Discussão</b>	151
7.4	<b>Considerações finais</b>	152
8	<b>CONCLUSÕES</b>	155
8.1	<b>Contribuições</b>	156
8.2	<b>Limitações</b>	157
8.3	<b>Trabalhos futuros</b>	158
	<b>REFERÊNCIAS</b>	159



---

# INTRODUÇÃO

---

Nas últimas décadas, a diminuição dos custos e os avanços de tecnologias para aquisição e armazenamento aumentaram substancialmente tanto a quantidade quanto a complexidade dos dados, além da velocidade da sua geração (PAULOVICH, 2008; COIMBRA, 2016). Com esses dados, gerados a partir de aplicações comerciais, financeiras, médicas, científicas etc., é possível analisar mercados e perfis de usuários, gerar sugestões de compras, definir linhas de crédito, melhorar diagnóstico e tratamento de doenças, fazer análise e monitoramento ambiental, dentre outras atividades (CHEN; ZHANG, 2014). Esses dados são encontrados não apenas em formatos estruturados, como em bancos de dados relacionais, mas também em bancos de dados NoSQL<sup>1</sup>, páginas de internet, textos, imagens, áudio e vídeo (IDREES; ALAM; AGARWAL, 2019).

Com o crescimento, tanto do volume quanto da complexidade dos dados, tornam-se necessárias técnicas para seu pré-processamento, exploração e análise, com a finalidade de encontrar tendências, padrões e informações relevantes que auxiliem a tomada de decisões. Dentre as áreas que desenvolvem algumas dessas abordagens, destaca-se a Aprendizagem de Máquina - *Machine Learning* (ML). Ela pode ser considerada como uma subárea da Inteligência Artificial, que por meio de algoritmos e modelos matemáticos busca “ensinar” máquinas como extrair padrões significativos de conjuntos de dados (RUSSELL; NORVIG, 2010; SHALEV-SHWARTZ; BEN-DAVID, 2014). As técnicas de aprendizagem são empregadas em tarefas de *i*) regressão; *ii*) agrupamento; *iii*) classificação; *iv*) quantificação; e outras. Com essas tarefas, é possível trabalhar com recuperação de informações, sistemas de veículos autônomos, reconhecimento biométrico, reconhecimento de imagens e áudio, análise de sequências genéticas, dentro outras. Para isso, os algoritmos de aprendizagem criam modelos matemáticos que representam padrões encontrados em dados de treinamento. Esses modelos são aplicados a novos dados para reconhecer os mesmos tipos

---

<sup>1</sup> Termo associado com bases de dados não relacionais, ou seja, que não utilizam estrutura tabular e suas relações.

de padrões, revelando informações relevantes sobre eles (cf. [Seção 2.4](#)).

Como mencionado, análise e monitoramento ambiental são algumas atividades que se beneficiam da quantidade de dados existente e das técnicas empregadas para sua exploração. É possível, então, mencionar atividades fundamentadas em sons ambientais ou urbanos, como o estudo de Paisagens Acústicas ([PIJANOWSKI et al., 2011a](#)). Essas paisagens são compostas por todos os sons gerados em um determinado ambiente, que definem um padrão único para ele, no tempo e no espaço ([PIJANOWSKI et al., 2011b](#)). Os conjuntos de sons de uma Paisagem Acústica são definidos como: *i*) biofonia (biophony), todos os sons produzidos por algum organismo vivo ([KRAUSE, 1987](#)); *ii*) geofonia (geophony), sons não biológicos ou geofísicos ([KRAUSE, 1987](#)); *iii*) e antropofonia (anthrophony), sons gerados direta ou indiretamente por atividades humanas ([PIJANOWSKI et al., 2011b](#)).

O estudo dessas paisagens busca descrever a diversidade e a relação entre os sons, para compreender a dinâmica do ambiente e suas alterações (cf. [Seção 2.3](#)). Com isso, é possível entender a influência do crescente tráfego de aviões e embarcações, das construções, da chegada de novas espécies, da migração ou desaparecimento das espécies atuais ou do efeito de mudanças climáticas em ambientes naturais ([SERVICK, 2014](#)). Existem tanto trabalhos com sons de ambientes naturais quanto de ambientes urbanos. Por exemplo, o projeto *Marine Environmental Research Infrastructure for Data Integration and Application Network* (MERIDIAN)<sup>2</sup> desenvolve ferramentas para análise de sons subaquáticos, o *Center for Conservation Bioacoustics* (CCB)<sup>3</sup> desenvolve ferramentas para coleta e interpretação de sons ambientais e o projeto *Sounds of New York City* (SONYC)<sup>4</sup> busca medir o impacto de ruído em áreas urbanas.

Um grande número de pesquisas está sendo desenvolvido nessa área, empregando técnicas variadas de ML para analisar, explorar, compreender e explicar a relação entre os eventos acústicos e os ambientes aos quais eles estão ligados, como se verifica ao longo do [Capítulo 3](#). Os resultados desses trabalhos sustentam tomadas de decisão relacionadas com problemas ecológicos relevantes, como detecção e controle de diversidade de espécies ([HARVEY, 2018](#)), monitoramento de áreas de preservação e suas alterações ([SÁNCHEZ-GENDRIZ; PADOVESE, 2016](#)), além da análise da qualidade ambiental e da sua biodiversidade ([RADFORD; KERRIDGE; SIMPSON, 2014](#)).

Mesmo com quantidade crescente de pesquisas na área, além da necessidade de manipulação de um volume cada vez maior de áudios, existem sérios problemas a serem enfrentados. Os eventos sonoros interessantes para os pesquisadores possuem grandes variações nas suas estruturas temporais e de frequência ([CAKIR et al., 2017](#)), dificultando a sua análise. Isso porque eles são originados em diversos tipos de ambientes, que possuem

---

<sup>2</sup> <<https://meridian.cs.dal.ca/>>

<sup>3</sup> <<https://www.birds.cornell.edu/ccb/>>

<sup>4</sup> <<https://wp.nyu.edu/sonyc/>>

características físicas e fontes de interferência distintas (PARKS; MIKSIS-OLDS; DENES, 2014). Por exemplo, a detecção apropriada de sons de animais é prejudicada por vocalizações de outros animais, ruído geofísico, sons de atividades humanas e problemas nos equipamentos de gravação (KAHL *et al.*, 2021). Também é possível detectar problemas e desvantagens relacionados com a maneira como os sons são representados para análise, o que dificulta a aplicação e a compreensão da relação dessas representações em ambientes distintos (ELDRIDGE *et al.*, 2016; KRAUSE; FARINA, 2016). Esses e outros problemas estão em aberto, criando um campo fértil para pesquisa e inovação, como a exploração e a interpretação de bases de sons, a análise da influência humana nos ambientes naturais, a influência de ruídos diversos no cotidiano das cidades, a detecção e mensuração de indivíduos e de espécies de interesse em ambientes naturais etc.

## 1.1 Objetivo

No contexto apresentado acima, a pesquisa desenvolvida neste doutorado teve por finalidade **identificar espécies de animais em gravações, a partir de um conjunto de características aprendidas por redes neurais, lidando com variações dos padrões, ruído, sobreposição e variação de intensidade dos sinais sonoros**. Assim sendo, as principais questões de pesquisa investigadas estão relacionadas com os parâmetros mais adequados da rede, como suas entradas, profundidade (quantidade de camadas), abordagem de inicialização dos pesos e estratégia para treinamento.

Os sons de interesse são vocalizações de pássaros e anuros<sup>5</sup>, que são considerados bioindicadores, ou seja, animais apropriados para medir a qualidade de um ambiente, devido à sua presença e comportamento refletirem o estado do ambiente onde eles são encontrados (MITCHELL *et al.*, 2020; STROUT *et al.*, 2017).

A aplicação de redes neurais para tarefas relacionadas com sons ambientais apresenta resultados consideráveis, devido à sua capacidade de lidar com modulações de frequência, além da identificação de padrões de tempo-frequência de sons naturais (SALAMON; BELLO, 2017). No entanto, existem vários problemas relacionados com sua capacidade de generalização, com sua habilidade em lidar com ruídos diversos, com sobreposição de sinais, com variações na intensidade desses sinais etc., problemas recorrentes na análise de sons obtidos em ambientes naturais e não controlados (BROWN; GARG; MONTGOMERY, 2019; KAHL *et al.*, 2021; LIN; FANG; TSAO, 2017; LIN; TSAO, 2020). Por causa disso, a aplicação de redes neurais é desafiadora e trabalhos com os de Dufourq *et al.* (2022) e Stowell (2022) buscam apresentar as melhores práticas para construção de modelos consistentes para resolução dos problemas destacados.

---

<sup>5</sup> Ordem de animais da classe *Amphibia*, como sapos, rãs, pererecas, dentre outras.

## 1.2 Contribuições e resultados obtidos

As contribuições e resultados desta pesquisa estão relacionadas com a regularização de funções de custo de redes neurais, para a classificação das espécies de interesse (DIAS; PONTI; MINGHIM, 2021). Além disso, existem contribuições na investigação de diferentes entradas e maneiras de combiná-las para o treinamento das redes. Por fim, a pesquisa contribui com resultados referentes a abordagens de inicialização dos pesos, para identificar como as arquiteturas usuais de redes se comportam com essa inicialização, no cenário de identificação de padrões sonoros. Descrições dessas contribuições e resultados estão no artigo citado e ao longo do texto.

## 1.3 Organização do texto

Este texto está organizado como segue. No [Capítulo 2](#), são apresentados conceitos fundamentais de dados multidimensionais, Processamento de Sinais, Análise de Paisagens Acústicas e Aprendizagem de Máquina, todos necessários para o entendimento da proposta e dos resultados obtidos. Os trabalhos relacionados são descritos no [Capítulo 3](#), com foco em pesquisas que usaram redes neurais para classificação de espécies animais, principalmente pássaros e anuros. No [Capítulo 4](#), é apresentada a abordagem proposta para a pesquisa. O [Capítulo 5](#), o [Capítulo 6](#) e o [Capítulo 7](#) relatam os resultados obtidos a partir da aplicação da proposta. Por fim, o [Capítulo 8](#) conclui o texto, destacando as contribuições, as limitações e os passos futuros para pesquisa.



---

## CONCEITOS FUNDAMENTAIS

---

---

Neste capítulo, são apresentados conceitos essenciais para fundamentação da proposta e dos experimentos, bem como uma visão geral para leitores não familiarizados com as áreas de Análise de Paisagens Acústicas ou Aprendizagem de Máquina. Assim sendo, são descritos conceitos de dados multidimensionais, que servem como entrada para algoritmos de exploração, análise e identificação de padrões, como redes neurais. Ferramentas aplicadas para processamento de sinais são recorrentes no tratamento de sons, gerando representações que facilitam sua análise. De maneira específica, as análises de Paisagens Acústicas possuem, em adição às características anteriores, suas próprias ferramentas empregadas para representação e análise de padrões sonoros relacionados com atributos ambientais. Por fim, o texto define e apresenta redes neurais, destaca exemplos de suas estruturas, além de estratégias de avaliação e melhoria de resultados dessas ferramentas, aplicadas em trabalhos de identificação de sons e no desenvolvimento da proposta de pesquisa.

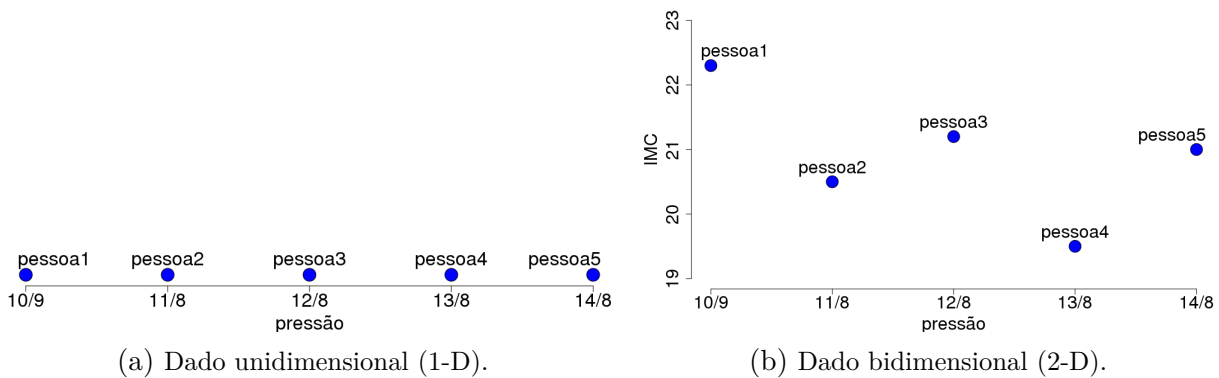
### 2.1 Dados multidimensionais

Nos dias atuais existe uma grande quantidade de dados provenientes de diversas áreas, como apontado na introdução. Esses dados variam quanto sua natureza, tipo, estrutura e significado. Uma questão importante para analisá-los está relacionada com a sua dimensionalidade, que pode ser compreendida como a quantidade de características ou atributos que os representam. Por exemplo, uma base de dados de pressão arterial de um grupo de pessoas, descreve cada pessoa por uma relação numérica. É possível representar, então, cada pessoa (registro, instância, objeto, item, exemplo, amostra) com um valor único  $x_i \in \mathbb{R}$ , sendo  $i$  o índice de cada pessoa na base de dados. Dados unidimensionais como esses podem ser representados em um gráfico para facilitar sua análise, como na [Figura 1a](#), onde o eixo  $x$  representa os valores do índice  $i$  dos dados. Se a mesma base possuir

além da pressão arterial o Índice de massa corporal (IMC) das pessoas, cada pessoa será representada por um vetor com dois valores numéricos, sendo  $\mathbf{x}_i \in \mathbb{R}^2$ . Também é possível analisar esses dados com um gráfico de dispersão (Figura 1b), onde cada eixo corresponde a uma das características dos dados (pressão, IMC).

De maneira simplificada, dados multidimensionais são dados representados como um vetor  $\mathbf{x}_i \in \mathbb{D}^n$ , sendo  $i$  o índice da amostra na base de dados,  $n > 1$  a quantidade de características e  $\mathbb{D}$  algum domínio desses atributos. Para  $n > 3$ , uma representação gráfica intuitiva é impraticável, o que gera para humanos dificuldades para identificar e explicar os padrões dos dados (COIMBRA, 2016).

Figura 1 – Exemplos de dados uni e bidimensionais.



Fonte: Elaborada pelo autor.

O termo multivariado também aparece em áreas como Estatística, Matemática ou Visualização de Dados, designando a quantidade de atributos de uma amostra e o termo multidimensional relacionado com a quantidade de índices necessários para acessar cada amostra (MUNZNER, 2014; WARD; GRINSTEIN; KEIM, 2015). Entretanto, esses termos podem aparecer na literatura de diferentes áreas de maneira inconsistente. Por convenção, ao longo deste texto o termo multidimensional estará sempre relacionado com a quantidade de atributos dos dados.

As características utilizadas para representar cada item  $i$  da base de dados podem ser de qualquer tipo, como valores numéricos (reais ou inteiros), valores categóricos (como espécies identificadas em um áudio ou seu local de coleta), textuais, entre outros. Entretanto, muitas técnicas como as de ML trabalham apenas com valores numéricos. Para isso, características como as categóricas podem ser mapeadas para um domínio numérico, usadas como rótulos dos dados, mapeadas para um domínio de cores (para usar em visualizações) ou desconsideradas a depender da situação.

Cada domínio de aplicação possui um conjunto diferente de características e de maneiras de gerá-las. Por exemplo, no contexto de redes sociais, os dados de um usuário podem considerar as relações que ele possui com outros usuários. Em um contexto bancá-

rio, os dados das transações podem ser as características de cada cliente. Para identificar a semelhança entre textos, as características de cada um deles podem estar relacionadas com as frequências das palavras que os compõem. Para detectar uma anomalia em uma imagem de satélite, características das cores, das texturas e das formas dentro da imagem podem ser consideradas. Em contextos como o de análise de Paisagens Acústicas, as características podem ser geradas a partir do sinal original do áudio, da sua representação no domínio da frequência ou de alguma combinação delas (cf. [Seção 2.3](#)).

De posse das características adequadas, os dados podem ser representados por vetores multidimensionais, o que possibilita aplicar ferramentas matemáticas, como as da Álgebra Linear, da Estatística ou da Otimização para descrever estruturas e relações dos dados, gerar modelos que os identifiquem etc., além de ferramentas computacionais como as apresentadas nas próximas seções. Vale ressaltar que os vetores de características ignoram qualquer tipo de informação que não esteja representada neles.

## 2.2 Processamento de sinais digitais

Um sinal é uma descrição da evolução temporal de algum fenômeno, que pode ser natural, como a variação das temperaturas ou da umidade em uma determinada região, ou artificial, como a variação dos preços de uma ação na bolsa de valores. O sinal pode ser compreendido como um dado unidimensional, com suas amostras sendo instantes de tempo e seu processamento são tarefas de modificação, geração ou análise por meio manual, mecânico ou digital ([PRANDONI; VETTERLI, 2008](#)).

Um sinal analógico é uma sequência infinita de valores, representada como  $x(t)$ , sendo  $x: \mathbb{R} \rightarrow \mathbb{R}$  e  $t$  o instante no tempo que o valor  $x$  do sinal aconteceu. Esse é o próprio sinal no mundo real, como um sinal de televisão via satélite. Por outro lado, um sinal discreto é uma sequência finita de valores, representada como  $x[n]$ , agora sendo  $x: \mathbb{Z} \rightarrow \mathbb{R}$ . O valor  $n$  corresponde ao “tempo”, que na verdade não está relacionado com o relógio, mas com a sequência dos valores de  $x$ .

Para que um sinal analógico  $x(t)$  seja tratado por dispositivos digitais, como o receptor da operadora de televisão ou um sistema para gravação de áudio, ele precisa ser convertido para o formato digital  $x[n]$  por um processo  $Y$ , denominado amostragem. Como exemplo desse processo, pode-se considerar o som de uma nota musical emitido por algum instrumento. Esse som é um sinal periódico que repete um determinado padrão, com uma frequência de  $f$  hertz (Hz). Valores altos dessa frequência estão relacionados com sons agudos e valores baixos com sons graves. Para que o sinal digital seja a melhor aproximação do analógico,  $Y$  deve coletar amostras de som a uma taxa de amostragem  $F_s > 2f$  (*Teorema de Nyquist*). Dessa maneira, cada segundo do som é representado no formato digital por  $F_s$  valores, que é no mínimo duas vezes maior do que a maior frequência  $f$  do sinal real

amostrado (HAYKIN; VEEN, 2001).

A sequência  $x[n]$  gerada possui as intensidades do sinal amostrado e representa o domínio do tempo do sinal. Para o som, intensidades maiores são sons com maior volume e intensidades menores sons com menor volume.

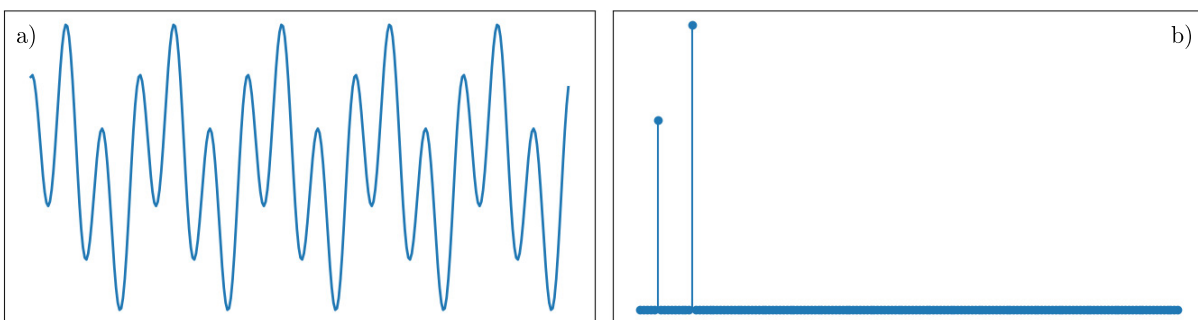
Um sinal também pode ser retratado no domínio da frequência que representa as relações entre as frequências de um sinal e suas amplitudes (espectro ou *spectrum*). Essa representação é obtida a partir da aplicação de ferramentas matemáticas, como a Transformada de Fourier (GONZALEZ; WOODS, 2010). Essa transformada é definida na sua forma discreta pela Equação 2.1 e sua inversa pela Equação 2.2, sendo  $M$  o comprimento do sinal discreto e  $X[k] \in \mathbb{C}$ . A transformada decompõe um sinal complicado, como som da voz ou um sinal de rádio, em sinais mais simples (como senos e cossenos), de maneira que possibilite verificar suas frequências, facilitando a análise e o processamento. A Figura 2 apresenta um exemplo de um sinal composto por frequências diferentes e decomposto por meio da transformada.

$$X[k] = \sum_{n=0}^{M-1} x[n] e^{-\frac{i2\pi}{M} kn} \quad (2.1)$$

$$x[n] = \frac{1}{M} \sum_{k=0}^{M-1} X[k] e^{\frac{i2\pi}{M} kn} \quad (2.2)$$

As próximas seções descrevem algumas representações desse domínio, usuais no processamento de sinais de som, além de operações que podem ser aplicadas nos dois domínios.

Figura 2 – A figura a) é um sinal formado pela soma de um seno e um cosseno com frequências e amplitudes distintas. A figura b) é o resultado da Transformada de Fourier com picos (proporcionais às suas amplitudes) nas duas frequências contidas nesse sinal.



Fonte: Elaborada pelo autor.

### 2.2.1 Geração de espectrogramas

Um espectrograma é uma representação visual do espectro de frequências de um sinal, dependente do tempo e empregado em tarefas de processamento de áudio e

fala (BRIGGS *et al.*, 2012; DONG *et al.*, 2015; STROUT *et al.*, 2017; THOMAS *et al.*, 2019; CASANOVA *et al.*, 2022), especialmente por ser compacta e compreensível (FLORENTIN; DUTOIT; VERLINDEN, 2020). Ele é obtido por meio da aplicação da Transformada breve ou de tempo curto de Fourier - *Short-term Fourier Transform* (STFT), que é uma variação da Transformada de Fourier, sendo definida pela Equação 2.3. A principal diferença entre as duas está na aplicação de uma janela  $w$ , de tamanho limitado, deslocada no tempo em  $t$  unidades, por onde o sinal  $x$  é “visto” (HAYKIN; VEEN, 2001). Existem vários tipos de janela, com formatos diferentes, como retangular, gaussiana, entre outras, sendo escolhidas a depender das necessidades da análise do sinal<sup>1</sup>.

$$X[k, t] = \sum_{n=0}^{M-1} x[n]w[n-t]e^{\left(\frac{-i2\pi}{M}kn\right)} \quad (2.3)$$

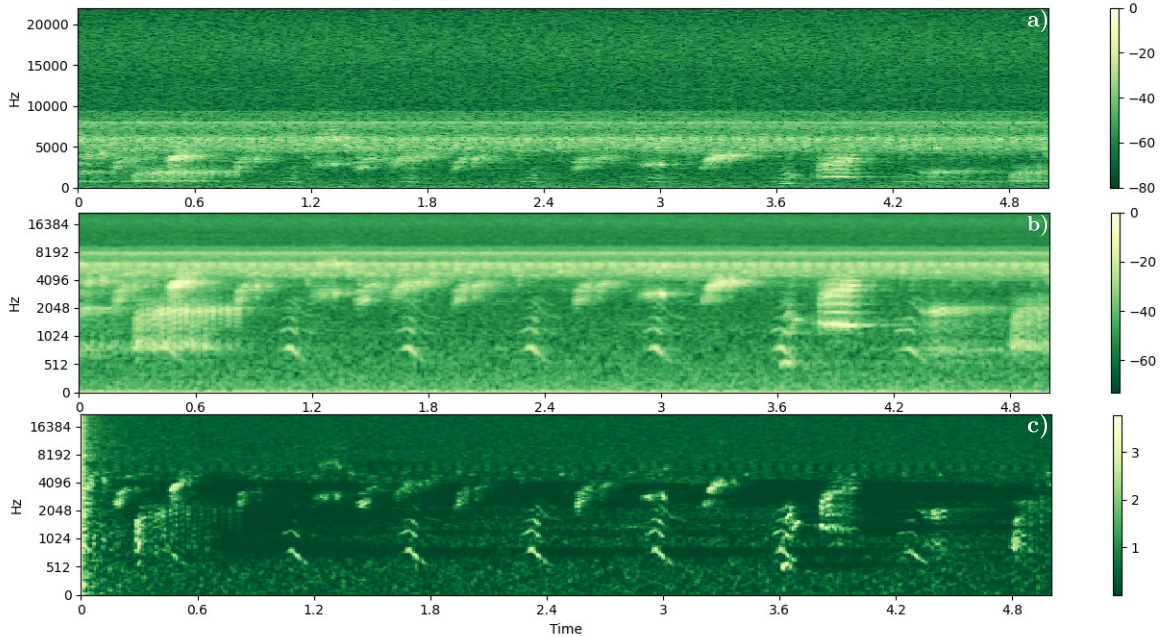
O processo de criação do espectrograma consiste em aplicar a STFT repetidas vezes, deslizando a janela sobre o sinal com algum nível de sobreposição. Assim sendo, para criação de um espectrograma é necessário identificar o tipo e tamanho da janela, além da taxa de sobreposição admitida entre os seus deslizamentos. O resultado de cada repetição é uma transformada do sinal, dentro da posição atual da janela. Os vetores de resultados gerados são concatenados como colunas de uma matriz. Dessa matriz, considera-se o quadrado do módulo dos seus valores, que podem ser convertidos para um espaço de cores, a fim de serem visualizados. A depender das necessidades de visualização, pode ser aplicada uma conversão dos resultados para *decibel*, antes do mapeamento para o espaço de cores. Com isso, o eixo vertical da imagem representa as bandas de frequência do sinal, o horizontal o tempo e as variações de cores as intensidades do sinal. Como exemplo, a Figura 3a apresenta o espectrograma de um áudio ambiental de 5 segundos, com padrões variados de sons.

Mesmo sendo aplicado em muitos trabalhos, a característica linear da faixa de frequências do espectrograma pode não ser a opção mais adequada para evidenciar padrões sonoros. Por causa disso, existem ferramentas e pesquisas que empregam variações do espectrograma como o mel-espectrograma (CAKIR *et al.*, 2017; LEBIEN *et al.*, 2020; PARASCANDOLO; HUTTUNEN; VIRTANEN, 2016; SALAMON; BELLO, 2015). Essa representação converte a escala linear dada em *Hz* para a escala Mel, buscando modelar a audição humana que possui comportamento próximo do linear para sons abaixo de 1 kHz e comportamento logarítmico acima disso (LOGAN, 2000). O mapeamento aplicado na escala original é realizado por meio da Equação 2.4 ou uma variação com logaritmo natural, sendo  $F_{hertz}$  a frequência de origem.

$$F_{mel} = 2595 \times \log_{10} \left( 1 + \frac{F_{hertz}}{700} \right) \quad (2.4)$$

<sup>1</sup> As janelas de *Hamming* e de *Hanning* são umas das mais utilizadas nesse contexto.

Figura 3 – Exemplos de *a)* espectrograma, *b)* mel-espectrograma e *c)* PCEN, gerados a partir de um áudio ambiental de 5 segundos



Fonte: Elaborada pelo autor.

Depois dessa conversão, a nova escala é dividida em  $N$  bandas associadas a filtros como os da Figura 4. Esses filtros triangulares possuem sobreposição com uma proporção de 50% e modificam a sua altura e largura até cobrirem todo o novo espectro (MATLAB, 2019). Dessa maneira, o mel-espectrograma pode ser definido como o produto interno entre uma matriz que representa um banco desses filtros e o quadrado dos valores absolutos da matriz STFT (HAN *et al.*, 2006). Um exemplo dessa representação é apresentado na Figura 3b, na qual é possível perceber que o mel-espectrograma alonga a faixa de frequências abaixo de 5 kHz e comprime frequências superiores a essa.

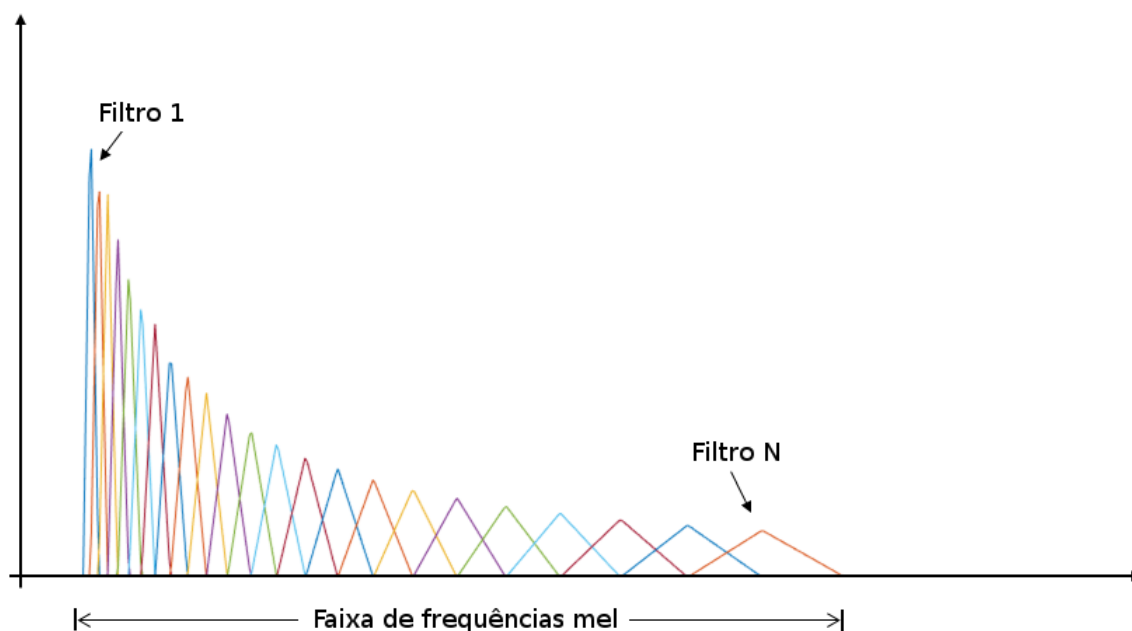
Outra maneira de representar padrões sonoros é o *Per-channel Energy Normalization* (PCEN) (WANG *et al.*, 2017), que busca reduzir distorções das frequências geradas pelo mel-espectrograma, combinando compressão e controle de ganho do sinal. É possível perceber na literatura que a aplicação dessa versão do espectrograma melhora resultados de técnicas de reconhecimento de fala, detecção de eventos acústicos, classificação de espécies, dentre outras (CRAMER *et al.*, 2020; HARVEY, 2018; LOSTANLEN *et al.*, 2018a; LOSTANLEN *et al.*, 2019). O processo é definido por meio da seguinte formulação

$$PCEN(t, f) = \left( \frac{E(t, f)}{(\varepsilon + (E \overset{t}{*} \phi_T)(t, f))^\alpha + \delta} \right)^r,$$

sendo  $t$  e  $f$  índices para tempo e frequência,  $E$  uma matriz como o mel-espectrograma,  $\phi_T$  um filtro passa baixa de primeira ordem aplicado por meio da operação  $\overset{t}{*}$ , como definido



Figura 4 – Banco de filtros para criação do mel-espectrograma



Fonte: Adaptada de [Matlab \(2019\)](#).

por ([LOSTANLEN et al., 2018a](#); [WANG et al., 2017](#)). A escala temporal  $T$  influencia o coeficiente de suavização do filtro e  $\alpha$ ,  $\varepsilon$ ,  $r$  e  $\delta$  são constantes positivas. O filtro em questão é um *Infinite Impulse Response* (IIR), aplicado para suavizar  $E$ , sendo que a divisão da fórmula executa um controle de ganho do sinal, atenuando ruído estacionário de fundo. O expoente  $r < 1,0$  e o deslocamento  $\delta$  promovem uma compressão enquanto que a subtração final contribui para a redução da faixa de valores. Um resultado desse processo também está representado na [Figura 3c](#), na qual é possível verificar um nível de redução de ruídos de fundo e manutenção de padrões abaixo de 4 kHz.

### 2.2.2 Medidas para análise de sinais

Além das representações por espectrogramas, existem outras ferramentas aplicadas no processamento e análise de sinais. Uma delas é a Densidade espectral - *Power Spectral Density* (PSD) que é aplicada para analisar a variação da energia de um sinal em relação às suas bandas de frequência, em um determinado período de tempo ([MADISETTI, 2009](#)). Essa medida é calculada a partir da integração da Transformada de Fourier, representada por  $\mathfrak{F}$  na seguinte equação

$$\varepsilon = \int_0^1 |\mathfrak{F}\{f\}|^2 df,$$

e um dos métodos utilizados para sua estimativa foi proposto por [Welch \(1967\)](#).

O PSD é aplicado em trabalhos de análise de Paisagens Acústicas, de maneira direta ou como ferramenta para calcular outras medidas como Índices Acústicos de medição da biodiversidade e de impacto do ruído nos ambientes (KASTEN *et al.*, 2012; MULLET *et al.*, 2016; SÁNCHEZ-GENDRIZ; PADOVESE, 2016).

Derivadas do mel-espectrograma, medidas como os Coeficientes Mel-Cepstrais - *Mel-frequency Cepstrum Coefficients* (MFCC) também são empregados em análise de som (LOGAN, 2000; STOWELL; PLUMBLEY, 2014). Para sua geração, a Transformada Discreta de Cosseno - *Discrete Cosine Transform* (DCT) é aplicada ao logaritmo do mel-espectrograma. Os  $n$  primeiros componentes (normalmente 12 ou 13) são considerados como os coeficientes cepstrais do sinal (LOGAN, 2000).

Além dessas medidas, também é possível listar outras usadas para análise de sinais de som, como o *Root Mean Square* (RMS) (BITTENCOURT *et al.*, 2016) que descreve de maneira simples a média da amplitude de um sinal alternado, o *Sound Pressure Level* (SPL) (SÁNCHEZ-GENDRIZ; PADOVESE, 2016) que descreve variações de pressão geradas pelo sinal, *Roughness* (RAMSAY, 2006) e *Rugosity* (MEZQUIDA; MARTÍNEZ, 2009) que medem variações no sinal, e o *Signal-to-noise Ratio* (SNR) (BEDOYA *et al.*, 2017) que representa a relação entre sinal e ruído.

### 2.2.3 Convolução

Convolução pode ser vista como uma operação entre duas funções (sinais) dependentes do tempo, que resulta em uma terceira função (GONZALEZ; WOODS, 2010). Essa operação tem aplicações no processamento de sinais e imagens, no aprendizado de máquinas (Seção 2.4.3.1), entre outras. O processo consiste em deslocar uma das funções no tempo, deslizá-la sobre a outra e retornar a soma ponderada da região onde as duas funções estão sobrepostas (HAYKIN; VEEN, 2001). A combinação das áreas sobrepostas nada mais é do que o produto interno entre elas. Essa operação no tempo discreto é nomeada como soma de convolução e descrita como

$$(x * w)[n] = s[n] = \sum_{k=-\infty}^{\infty} x[k]w[n-k]. \quad (2.5)$$

A Figura 5 apresenta um exemplo dessa operação em uma dimensão. A inversão de  $w$  é realizada por meio do termo  $-k$ , empregado na Equação 2.5 para deslocar  $w$  no tempo (ou invertê-lo em 180°), fazendo com que a operação seja comutativa. Os zeros em cinza (*padding*), à direita e à esquerda de  $x$ , são adicionados para permitir que sempre exista sobreposição entre as funções, facilitando o processo. Além disso, podem ser adicionados uns ou os elementos das extremidades de  $x$  podem ser copiados quantas vezes for necessário.

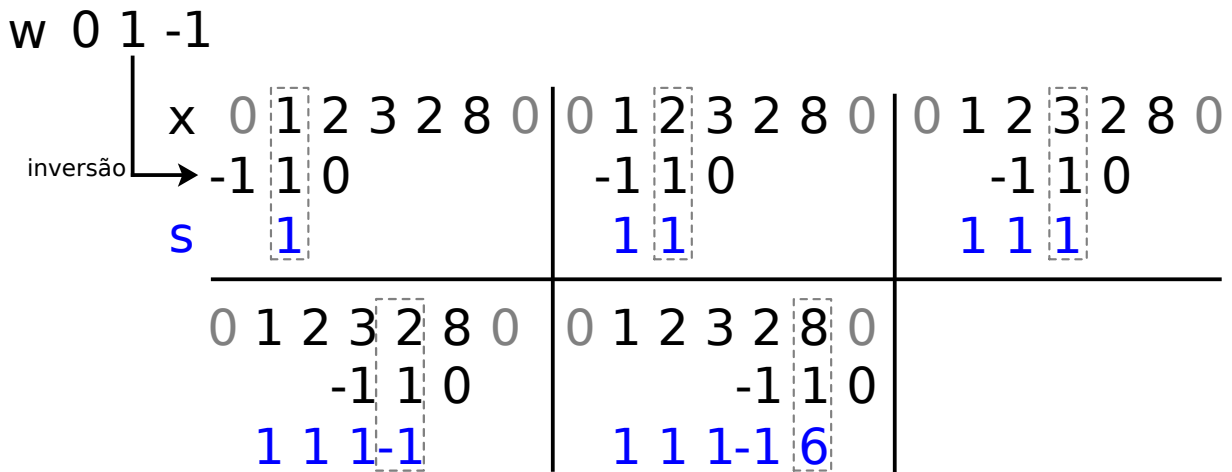


Nas implementações, é possível verificar algumas maneiras de executar esse processo. Em uma delas (filtro causal), o último elemento do  $w$  invertido (0 na imagem) é emparelhado com o primeiro de  $x$  sem adições (1 na imagem). Em outra possibilidade, como está no exemplo, o elemento central de  $w$  é associado com o primeiro de  $x$ . Além delas, o primeiro elemento do  $w$  invertido (-1 na imagem) é associado com o primeiro de  $x$ .

No primeiro caso, o tamanho do *padding* é  $m - 1$ , sendo  $m$  o comprimento de  $w$ , e essa quantidade de pontos é adicionada a cada extremidade de  $x$ ; no segundo caso, a quantidade é  $\lfloor m/2 \rfloor$ ; e no terceiro não existe necessidade de complementação. Logo, nos resultados do primeiro caso é necessário remover das extremidades a mesma quantidade de itens adicionada para o *padding*, para que o resultado possua o mesmo comprimento de  $x$  (GONZALEZ; WOODS, 2010); no segundo caso, o tamanho do resultado é o mesmo tamanho de  $x$ ; no último caso, o tamanho do resultado é inferior ao tamanho de  $x$ .

Na Figura 5, a convolução avança a cada posição, mas podem ser duas, três ou mais posições, o que tem a capacidade de gerar saídas menores. Esse passo (*stride*) não deve ser grande a ponto de gerar perda de informação. Por exemplo, nessa figura, se o passo for maior que dois, informações da sequência  $x$  serão ignoradas.

Figura 5 – Convolução em uma dimensão (1-D) dos sinais  $x$  e  $w$  resultando no sinal  $s$ . Os zeros em cinza nas extremidades indicam o *padding* de  $x$ . O retângulo cinza pontilhado é a posição atual processada.



Fonte: Elaborada pelo autor.

A convolução também é aplicada em dados com mais dimensões, com os devidos ajustes nas equações e nos processos, como ocorre em filtros de processamento de imagens e em redes convolucionais (cf. Seção 2.4.3.1). Em alguns casos, não existe a necessidade de inversão de uma das funções, como em filtros simétricos ou na correlação espacial ou cruzada<sup>2</sup>, que efetua o mesmo processo, desconsiderando a inversão de um dos operandos,

<sup>2</sup> Essa correlação não tem relação com a correlação utilizada em Estatística e Análise de dados.

gerando uma operação que não é comutativa (GONZALEZ; WOODS, 2010; GOODFELLOW; BENGIO; COURVILLE, 2016).

## 2.3 Análise de Paisagens Acústicas

Como destacado na introdução, a análise de Paisagens Acústicas busca compreender a relação entre os sons e diversos aspectos de um ambiente. Esses sons narram o que acontece no ambiente e são capazes de contar histórias sobre ele, possibilitando sua caracterização. Por exemplo, existem relações diretas e indiretas entre a saúde ambiental, a emissão de sons e a distribuição das espécies animais. Espécies de sapos podem indicar mudanças ambientais, porque requerem hábitat aquático e terrestre adequados. Elas também absorvem facilmente elementos químicos tóxicos devido à sensibilidade de sua pele. Assim sendo, mudanças nos ambientes que elas vivem, referentes a temperatura, contaminação etc., são capazes de influenciar rotina das espécies e sua emissão sonora (XIE *et al.*, 2015). Por causa dessa sua capacidade de refletir o estado do ambiente, esses animais, assim como espécies de pássaros, são usados como bioindicadores (MITCHELL *et al.*, 2020; STROUT *et al.*, 2017).

Diferentes níveis de ruído também podem influenciar a vida animal. Eles podem ser percebidos como algum tipo de ameaça, podem causar degradação sensorial ou limitar a percepção de membros da mesma espécie, de predadores, de presas ou do próprio ambiente. Essas dificuldades influenciam a interação entre predadores e presas, a reprodução, a dinâmica das espécies nos locais onde elas residem, causando estresse, gerando dor e modificação dos níveis hormonais dos animais (SHANNON *et al.*, 2016).

Além da relação entre padrões sonoros e modificações na saúde e relações sociais dos animais, algumas alterações nos sons de um ambiente podem ser explicadas pela *Hipótese do nicho acústico* (KRAUSE, 1987). Segundo essa hipótese, devido à competição, os animais tendem a modificar características temporais e de frequência dos seus sons, de maneira a evitar sobreposições. Com isso, é possível gastar menos energia para produção sonora (importante para espécies de animais pequenos) e melhorar a eficácia da comunicação, visto que assim ela enfrenta menos interferência. Dessa maneira, áreas consolidadas, como florestas primárias, possuem a maior parte das faixas do espectro sonoro preenchida. Por outro lado, áreas com algum distúrbio, como ocasionado por espécies invasoras, podem possuir lacunas no espectro ou maior sobreposição de sons (PIJANOWSKI *et al.*, 2011b). Os animais também podem ajustar suas vocalizações de acordo com características físicas do ambiente (*Hipótese de adaptação acústica*) para otimizar a propagação dos sons (EY; FISCHER, 2009) ou, ainda, ocupar ambientes adequados para emissão e percepção sonora (*Hipótese do hábitat acústico*), escolhendo ambientes tanto a partir de características físicas quanto de biodiversidade (MULLET; FARINA; GAGE, 2017).

O processo de análise das paisagens inicia com a aquisição dos sons através de sistemas como Monitoramento Acústico Passivo - *Passive Acoustic Monitoring* (PAM), para captação e gravação de sons durante longos períodos contínuos, produzindo grandes volumes de dados (THOMAS *et al.*, 2019). Com esses sistemas de gravação, é possível detectar maiores faixas de sinais, durante maiores períodos de tempo, além de minimizar a influência humana nos ambientes e os custos do processo, permitir a criação de bases para análise futura, aumentar a capacidade de detecção de espécies raras e assim por diante (NEAL, 2012; PIERETTI *et al.*, 2017; PIJANOWSKI *et al.*, 2011b; SERVICK, 2014; ZNIDERSIC *et al.*, 2020). Entretanto, com o avanço das tecnologias tanto para gravação quanto para persistência (armazenamento) de dados, o maior desafio consiste em processar seu crescente volume de maneira rápida e eficaz (ZNIDERSIC *et al.*, 2020).

### 2.3.1 Representação de sons ambientais

Para fins de análise, um arquivo de áudio mono (um canal) ou estéreo (dois canais) pode ser representado por características derivadas do domínio do tempo, da frequência ou de uma combinação deles. Essas características são analisadas separadas ou em conjunto, formando um vetor multidimensional empregado em tarefas como análise visual das paisagens (HILASACA *et al.*, 2021; PHILLIPS; TOWSEY; ROE, 2018; ZNIDERSIC *et al.*, 2020) e detecção de eventos (GAN *et al.*, 2020; GAN *et al.*, 2021; HILASACA; RIBEIRO; MINGHIM, 2021).

Um conjunto dessas características é formado pelos Índices Acústicos, que são funções matemáticas que representam elementos estruturais do som, dinâmica ou sua distribuição de energia. Além disso, podem quantificar a riqueza, a uniformidade, a regularidade ou a abundância de espécies que aparecem nos sons (SUEUR *et al.*, 2014; TOWSEY *et al.*, 2014). Os índices podem ser classificados como índices  $\alpha$  (*within-group*) e  $\beta$  (*between-group*), sendo que aqueles estão relacionados com medidas da riqueza ou a abundância relativa de um conjunto específico de áudios. Eles podem medir a intensidade sonora, a sua complexidade, que está ligada à quantidade de animais ou espécies emitindo sons, e a relação entre sons biofônicos e antropônicos. Enquanto isso, os índices  $\beta$  são desenvolvidos para diferenciar os sons dos ambientes. Entretanto, não é simples comparar conjuntos de sons, porque os sinais podem variar de maneira independente no tempo, na frequência ou na amplitude (SUEUR *et al.*, 2014).

Mesmo sendo bastante empregados nos estudos de Paisagens Acústicas, como pode ser visto em trabalhos de classificação de ambientes (GÓMEZ; ISAZA; DAZA, 2018), medição de riqueza de espécies de pássaros (ELDRIDGE *et al.*, 2018) e exploração de Paisagens Acústicas marinhas (PIERETTI *et al.*, 2017), existem controvérsias sobre a sua real eficácia para avaliação ambiental. Isso porque os índices podem não refletir todos os atributos da estrutura da paisagem e podem ser influenciados por ruídos temporários ou

permanentes. Além disso, as relações entre diversidade e complexidade acústica medidas pelos índices podem divergir em diferentes regiões do planeta (ELDRIDGE *et al.*, 2016; KRAUSE; FARINA, 2016). Logo, torna-se necessária a avaliação da aplicabilidade dos índices em diferentes regiões, como fazem os trabalhos de Harris, Shears e Radford (2016), de Eldridge *et al.* (2018) e de Jorge *et al.* (2018).

Existe uma vasta lista de índices, mas alguns deles são mais comuns na literatura, com frequência do tipo  $\alpha$ , como o *Acoustic Complexity Index* (ACI) (PIERETTI; FARINA; MORRI, 2011), *Acoustic Diversity Index* (ADI) (PEKIN *et al.*, 2012), *Acoustic Evenness Index* (AEI) (VILLANUEVA-RIVERA *et al.*, 2011), *Acoustic Richness* (AR) e o *Median of Amplitude Envelope* (M) (DEPRAETERE *et al.*, 2012), *Bioacoustic Index* (Bio) (BOELMAN *et al.*, 2007), *Background noise index* (BGN) (TOWSEY, 2017; DIAS; PONTI; MINGHIM, 2022), as entropias *Acoustic Entropy Index* (H), *Temporal Entropy* ( $H_t$ ) e *Frequency Entropy* ( $H_f$ ) (SUEUR *et al.*, 2008), *Normalized Difference Soundscape Index* (NDSI) (KASTEN *et al.*, 2012) e o *Number of Peaks* (NP) (GASC *et al.*, 2013).

Esses índices e as medidas apresentadas na Seção 2.2.2 são aplicados em tarefas relacionadas com Paisagens Acústicas, tanto em ambientes terrestres quanto subaquáticos, para análise, detecção de eventos, dentre outros (DRÖGE *et al.*, 2021; GAN *et al.*, 2020; MITCHELL *et al.*, 2020; PIERETTI *et al.*, 2017; SCARPELLI; RIBEIRO; TEIXEIRA, 2021). Além disso, existem trabalhos que empregam técnicas de ML para geração automática de características (SALAMON; BELLO, 2015; STOWELL; PLUMBLEY, 2014), buscando criar uma representação significativa dos dados a partir de uma tarefa específica de identificação. Outros trabalhos aplicam redes neurais para identificar padrões nos espectrogramas dos áudios, como pode ser visto nos trabalhos descritos no Capítulo 3.

## 2.4 Aprendizagem de Máquina

Como mencionado na introdução, a Aprendizagem de Máquina (ML) é vista como uma subárea da Inteligência Artificial que tem por finalidade propor técnicas capazes de extrair padrões significativos (conhecimento) de algum conjunto de dados (experiência) (RUSSELL; NORVIG, 2010; SHALEV-SHWARTZ; BEN-DAVID, 2014). A aplicação dessas técnicas visa a construção de sistemas que possam reconhecer padrões e tomar decisões com pouca ou nenhuma intervenção humana.

A capacidade mais importante dos algoritmos de ML está relacionada com aprender por meio de exemplos. Esse aprendizado se dá por meio de um processo de treinamento, em que um conjunto de  $n$  exemplos multidimensionais é apresentado ao algoritmo e ele busca retornar algum conhecimento sobre esses dados, aplicável em dados ainda não vistos pelo algoritmo. A depender desse algoritmo, o conhecimento pode ser representado por um conjunto multidimensional de valores que mapeiam as características relevantes

dos dados. Pode-se caracterizar o processo de treinamento como *i*) supervisionado, *ii*) não supervisionado, *iii*) semisupervisionado ou *iv*) por reforço (RUSSELL; NORVIG, 2010).

Sejam  $\mathbf{x}_i, y_i$ , com  $\mathbf{x}_i \in \mathbb{R}^d$  pares de dados e seu respectivo valor alvo  $y_i$ , disponíveis para um determinado problema. No aprendizado supervisionado, o algoritmo busca, a partir dos  $d$  atributos das instâncias  $\mathbf{x}$  de um conjunto de treinamento, encontrar a função  $\hat{f}(\mathbf{x}_i) \approx f(\mathbf{x}_i) = y_i$ . Nessa formulação,  $y_i$  é o valor alvo, que pode ser um rótulo associado à instância  $i$ , a probabilidade desse item possuir um determinado rótulo ou um valor real qualquer associado a ele (GOODFELLOW; BENGIO; COURVILLE, 2016; RUSSELL; NORVIG, 2010). No caso de tarefas de classificação, a função  $f(\cdot)$  será selecionada durante o processo de treinamento dentre de um espaço de funções admissíveis  $\mathcal{F}$  de um algoritmo de classificação (MELLO; PONTI, 2018).

O processo de treinamento é semelhante a um professor ou supervisor informando ao algoritmo qual é a saída desejada para uma amostra específica dos dados. Segundo Russell e Norvig (2010), nos casos ligados a rótulos ou suas probabilidades, o algoritmo está realizando uma tarefa de classificação de dados, enquanto que nos casos ligados a valores contínuos em geral, o algoritmo executa uma tarefa de regressão ou de previsão. Também é possível citar a quantificação, que será abordada na Seção 2.4.6. No caso supervisionado podemos ver o treinamento como uma forma de estimar a probabilidade  $P(X, Y)$ , sendo  $X$  o espaço de entrada (os dados), e  $Y$  o espaço de saída.

Por outro lado, algoritmos não supervisionados aprendem propriedades úteis da estrutura dos dados, como o grau de segregação entre eles. Nesse caso, os algoritmos não possuem supervisão, ou seja, os dados de treinamento não possuem um alvo  $y_i$  associado às características do exemplo de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma das tarefas não supervisionadas mais comuns é o agrupamento, em que o algoritmo separa os dados em grupos de exemplos mais parecidos (similares) (RUSSELL; NORVIG, 2010). A quantidade de grupos pode ser definida pelo usuário ou gerada a partir de alguma heurística aplicada sobre os dados de treinamento (ZHU; GOLDBERG, 2009).

Em particular para a tarefa de classificação, é necessária a disponibilidade de uma quantidade razoável de dados rotulados para o treinamento, de forma a garantir que haverá convergência do processo de otimização relacionado ao aprendizado (MELLO; PONTI, 2018). Entretanto, em muitos cenários, como o de análise de Paisagens Acústicas, é difícil, lento e custoso definir uma boa quantidade desses rótulos. Existem muitas abordagens que procuram mitigar esse problema, sendo uma delas o aprendizado semisupervisionado. Esse conjunto de técnicas está entre o aprendizado supervisionado e o não supervisionado, buscando estender um tipo de aprendizado com informações geradas pelo outro (RUSSELL; NORVIG, 2010; ZHU; GOLDBERG, 2009) ou ainda utilizando durante o treinamento tanto informações de pares rotulados de amostras quanto não rotulados (CAVALLARI; PONTI, 2021). Dessa maneira, tarefas de classificação usam a

informação de uma pequena parte rotulada dos dados associada com as informações da estrutura dos dados não rotulados. Segundo [Zhu e Goldberg \(2009\)](#), exemplos de mesmo rótulo formam grupos em torno da média de dados similares.

No caso do aprendizado por reforço, um agente (seja ele um *hardware* ou *software*) interage com o ambiente a partir de experiências, sendo que os estados do ambiente são sequenciais, ou seja, o estado atual depende das ações tomadas nos estados anteriores. Por exemplo, em um jogo de xadrez, o movimento atual do jogador depende das jogadas anteriores ou influencia as próximas jogadas. Essa interação é reforçada por meio de recompensas que o agente obtém com suas ações, então, o agente busca a partir de uma determinada estratégia, encontrar uma sequência de passos que maximiza o valor das recompensas obtidas. Vale ressaltar que essas recompensas podem ser esparsas, como no exemplo do xadrez, onde o agente pode não receber recompensas a cada ação, mas sim após um bom conjunto de ações que leva ao final do jogo, ganhando ou perdendo a partida ([RUSSELL; NORVIG, 2010](#)). Dessa maneira, o agente precisa de uma estratégia, um conjunto de regras que define qual ação tomar, que é uma função  $\pi$  que determina a possibilidade de se executar uma ação  $a$  a partir de um estado  $s$ , podendo ser determinística ou estocástica, neste caso definida como  $\pi(s, a) = P(A = a | S = s)$ .

### 2.4.1 Problemas de generalização

Depois do treinamento, espera-se que um algoritmo de ML, quando aplicado em dados não vistos durante o treinamento, retorne resultados satisfatórios. Por exemplo, um classificador treinado para identificar imagens de felinos deve ser capaz de identificar um gato de cor e tamanho não vistos durante o treinamento, porque felinos possuem características em comum, mesmo que algumas outras variem. Informalmente, a capacidade dos algoritmos de aprender a identificar a variação de padrões dos dados a partir de um subconjunto de treinamento é chamada de generalização ([FACELI \*et al.\*, 2011](#)). Formalmente, seja  $\mathcal{R}_{emp}(f)$  o risco empírico estimado para um determinado classificador  $f$  em um conjunto de treinamento, por exemplo, o erro de classificação medido nesse conjunto. A generalização está associada à divergência  $|\mathcal{R}_{emp}(f) - \mathcal{R}(f)|$  entre o risco empírico e o risco real  $\mathcal{R}(f)$  para os dados não vistos durante o treinamento ([MELLO; PONTI, 2018](#)).

Quando um modelo de classificação não consegue identificar padrões nem mesmo dos dados de treinamento, ele está subajustado (*underfitting*), refletindo a incapacidade do modelo em se ajustar aos dados, o que comumente indica que o espaço de funções admissíveis é muito restrito. Em outras palavras, o modelo possui poucos parâmetros ou graus de liberdade. Por outro lado, quando ele se ajusta em demasia aos dados de treinamento, ele está superajustado (*overfitting*), convergindo para um modelo que memoriza todos os dados de treinamento e não é capaz de identificar variações nos padrões, o que inviabiliza sua aplicação em dados não vistos ([MELLO; PONTI, 2018](#)).



Esses problemas podem surgir quando um classificador é muito simples para absorver a complexidade das estruturas dos dados de treinamento ou o inverso, quando o modelo é complexo em relação ao problema tratado; quando os dados de treinamento não representam de maneira razoável as variações dos dados reais; ou quando o tempo de treinamento é longo o suficiente para o modelo “memorizar” até mesmo ruídos e imperfeições dos dados. Quantidade insuficiente de amostras e classes desbalanceadas (quantidades de amostras variam entre as classes) também geram dificuldades de generalização e previsões incorretas, sobretudo das classes que possuem menos amostras (JOHNSON; KHOSHGOFTAAR, 2019; WANG; PEREZ *et al.*, 2017). Algumas técnicas empregadas para reduzir esses problemas serão apresentadas na Seção 2.4.4.

### 2.4.2 Avaliação de desempenho de métodos de classificação

Existem várias medidas empregadas para avaliar a capacidade de um classificador, muitas delas derivadas da matriz de confusão. Essa matriz apresenta as quantidades de previsões corretas (verdadeiro positivo VP ou verdadeiro negativo VN) e incorretas (falso positivo FP ou falso negativo FN) de cada classe, podendo ser organizada com as linhas da matriz representando as classes esperadas (verdadeiras) e as colunas as classes preditas, ou o inverso com as esperadas nas colunas. Logo, os valores na diagonal da matriz comunicam os acertos por classe e os demais valores os erros de previsão em cada classe (FACELI *et al.*, 2011). Considerando o exemplo da Figura 6, para a classe B, o algoritmo classificou 6 amostras de maneira correta e outras 4 de maneira incorreta.

Figura 6 – Exemplo de matriz de confusão.

		predito		
		A	B	C
esperado	A	10	0	0
	B	3	6	1
	C	5	5	0

Fonte: Elaborada pelo autor.

Algumas medidas possíveis de serem calculadas a partir dessas informações são a sensibilidade ou revocação, que divide os acertos de cada classe (valores da diagonal) pela quantidade total de amostras de cada classe, na Figura 6, somatório dos valores de cada linha, então  $VP/(VP + FN)$ . Outra medida é a precisão, que também usa os acertos de cada classe, mas divide pela quantidade de previsões feitas pelo classificador para aquela classe, na Figura 6, o somatório de cada coluna, então  $VP/(VP + FP)$ . A especificidade de uma classe usa os acertos das demais classes (VN) e divide pela quantidade total de

amostras também das demais classes, na [Figura 6](#), o somatório dos valores de uma linha, desconsiderando a classe analisada, então  $VN/(VN + FP)$ .

Existem outras maneiras de avaliação por classe, como o *F-score* que é a média harmônica da precisão e da sensibilidade. É comum ponderar essa média, atribuindo peso unitário para a precisão e peso  $k^2$  para a sensibilidade, criando uma medida *Fk-score*, como *F1-score* (ou *F-score*), *F0.5-score* etc. Outras medidas como a curva *Receiver Operating Characteristic* (ROC), que é um gráfico entre sensibilidade e o complemento da especificidade, ou a curva precisão-sensibilidade são consideradas, sendo sumarizadas pela área sob a curva ROC e pela área sob a curva precisão-sensibilidade. Para esta última, a *average precision* (AP) é uma das técnicas empregadas para cálculo da sua área.

Além de análises por classe, também é possível gerar medidas para todo o processo de classificação como a acurácia, que divide a quantidade total de acertos (somatório da diagonal) pela quantidade total de amostras preditas (somatório de toda a matriz). Entretanto, quando os dados são desbalanceados, a acurácia não reflete de maneira coerente os resultados do classificador, devido às diferentes proporções das classes. Tomando como exemplo um cenário de classificação binária onde uma classe *X* possui 1000 amostras e uma classe *Y* possui 10 amostras, se o classificador rotular todas as 1010 amostras como *X*, o valor de acurácia será de  $\approx 99\%$ , sendo um equívoco, porque mesmo acertando todas as predições para a classe majoritária, o classificador não foi capaz de prever a outra classe. Nesse tipo de cenário, pode-se considerar a aplicação de pesos proporcionais às quantidades de amostras das classes, ou seja, quanto maior a quantidade de amostras da classe menor o seu peso, ou utilizar a média da sensibilidade (acurácia balanceada), como em algumas bibliotecas de programação<sup>3</sup>.

Além disso, para obter valores confiáveis dessas medidas, é necessário que os classificadores sejam treinados em um subconjunto (treinamento) e avaliados em outro (teste), sendo este um subconjunto de dados não vistos durante o treinamento, que simulam a classificação em cenários reais. Logo, medidas como acurácia, precisão e sensibilidade são avaliadas nos dados de teste. Alguma técnica de amostragem pode ser aplicada, sendo comum a utilização de validação cruzada (*cross-validation*), onde os dados são particionados em  $k$  partições (*folds*) e  $k$  iterações são executadas, sempre considerando uma das partições como teste e as demais como treinamento. Assim, as médias das medidas são consideradas para avaliar o resultado geral do classificador ([FACELI et al., 2011](#)).

Uma abordagem aplicada para refinar as configurações de um classificador, mantendo o conjunto de testes intocado, é a criação de outra partição nos dados de treinamento, gerando um subconjunto de validação ([RUSSELL; NORVIG, 2010](#)). Dessa maneira, os dados podem ser divididos em três subconjuntos: treinamento, onde o modelo é

<sup>3</sup> <[https://scikit-learn.org/stable/modules/model\\_evaluation.html#balanced-accuracy-score](https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score)>



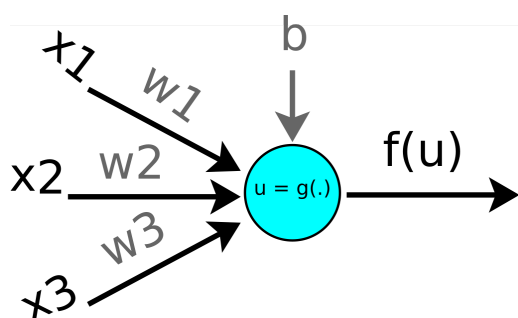
gerado; validação, onde o modelo é avaliado para refinar suas configurações; e teste, onde o modelo final é verificado. Logo, os dados podem ser particionados entre treinamento e teste, e a validação cruzada pode ser aplicada nos dados de treinamento para gerar subconjuntos de validação.

### 2.4.3 Redes Neurais Artificiais

Existem muitas abordagens empregadas em ML, uma delas são as Redes Neurais Artificiais (RNA). Elas são estruturas compostas por unidades simples (neurônios artificiais), interconectadas e organizadas em camadas (Figura 7b). Os neurônios recebem valores como entrada, associam pesos e outros valores (viés) a eles e geram um resultado (BRAGA; FERREIRA; LUDERMIR, 2007). As redes podem ser aplicadas em tarefas de classificação, regressão, previsão ou de agrupamento de dados.

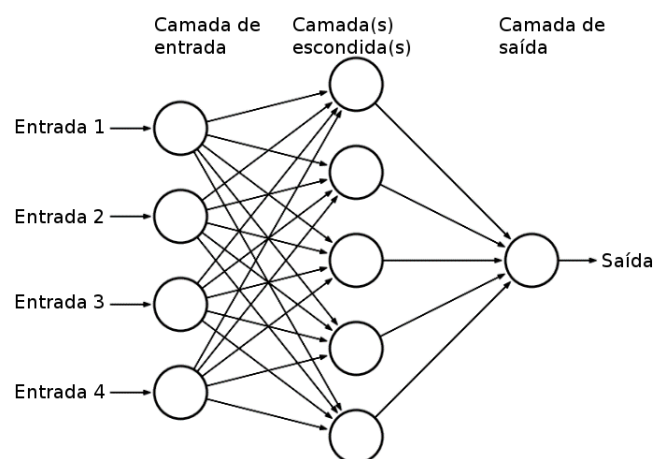
A Figura 7a representa um modelo de neurônio conhecido como McCulloch e Pitts (MCP) (MCCULLOCH; PITTS, 1943). A função  $g$  é responsável por realizar a associação entre os valores, podendo ser definida como uma combinação linear  $\mathbf{w}^T \mathbf{x} + b$ , sendo  $\mathbf{x} \in \mathbb{R}^n$  um vetor multidimensional com as características de entrada,  $\mathbf{w} \in \mathbb{R}^n$  um vetor com os pesos e  $b$  um viés usado para influenciar o aprendizado do neurônio. Enquanto isso, a função  $f$  é a função de ativação do neurônio, que transforma o resultado de  $g$  na saída do neurônio. Existem várias funções aplicadas para isso, que devem ser deriváveis, como a tangente hiperbólica ( $\tanh$ ) e a função sigmoide  $f(u) = 1/(1 + \exp^{-u})$ .

Figura 7 – Representação básica de uma RNA



(a) Modelo de um neurônio MCP com três entradas.

Fonte: Elaborada pelo autor.



(b) Arquitetura de uma rede MLP.

Fonte: Adaptada de Sharawy et al. (2016).

A Figura 7b apresenta a arquitetura de uma rede *Multilayer Perceptron* (MLP) empregada para classificação/regressão (HAYKIN, 1999). Essa rede pode ser entendida como um grafo ponderado, direcionado e acíclico (redes *feedforward*), na qual os vértices

são os neurônios e as arestas representam as conexões entre eles, de modo que a saída de um seja a entrada do outro (SHALEV-SHWARTZ; BEN-DAVID, 2014).

O conhecimento em uma MLP é armazenado nos pesos das conexões dos neurônios, que podem ser inicializados aleatoriamente ou com pesos aprendidos em outra tarefa (cf. Seção 2.4.4). Eles são atualizados a partir de um processo de otimização combinado com o algoritmo *back-propagation*, que utiliza a propagação dos erros da camada de saída para gerar as derivadas necessárias para a otimização (BRAGA; FERREIRA; LUDERMIR, 2007; HAYKIN, 1999). Esse processo consiste em apresentar amostras de treinamento (uni ou multidimensional) para a rede, calcular o erro do resultado, gerar as derivadas necessárias, atualizar os pesos a partir dessas derivadas e refazer o processo para cada subconjunto de amostras (*batch*) da base de dados.

O erro de cada neurônio  $j$  da camada de saída, em uma iteração  $t$  do treinamento, pode ser definido de maneira simples por  $e_j(t) = y_j(t) - f_j(t)$ , sendo  $y_j$  a saída desejada e  $f_j$  a gerada pelo neurônio. O objetivo é encontrar os valores de todos os pesos  $w$  da rede, de modo que uma medida geral dos erros, que possua derivada, como a soma quadrática de  $e_j$  (função de perda ou de custo) da camada de saída seja a menor possível, ou seja,

$$\min_w \quad \varepsilon(t) = \frac{1}{2} \sum_j e_j^2(t).$$

Além do desse erro quadrático e suas variações, outras funções são empregadas, como erros absolutos, funções baseadas em proximidade, entropia etc. Também é possível combinar a função de erro com termos de regularização, para evitar ambiguidades que levem a rede a problemas de generalização (GOODFELLOW; BENGIO; COURVILLE, 2016; PONTI *et al.*, 2017). Um regularizador comum é a norma  $L^2$  (ou norma Euclidiana) que contabiliza a soma do quadrado dos pesos e pode ser representada como  $\|\mathbf{w}\|_2$ , sendo  $\mathbf{w}$  o conjunto de pesos da rede. Com essa normalização, a função da otimização acima é definida como  $\varepsilon(t) + \gamma \|\mathbf{w}\|_2$ , onde  $\gamma$  é um valor que controla a influência do regularizador.

O processo de otimização pode ser realizado com algoritmos como o gradiente descendente (NOCEDAL; WRIGHT, 2006), onde o vetor dos pesos é corrigido no sentido contrário do gradiente do erro da rede, com  $\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha \nabla \varepsilon(t)$ , sendo  $\alpha$  a taxa de aprendizagem que controla quanto o valor será corrigido (tamanho do passo) naquele sentido e a função  $\varepsilon$  com ou sem regularização. O erro dos neurônios da camada de saída é conhecido, mas o erro das camadas escondidas não, porque elas não têm uma saída esperada. Assim sendo, os erros e gradientes são calculados recursivamente pelo *backpropagation*, voltando nas camadas, a partir da saída, sempre contabilizando a influência da camada atual nos erros da camada anterior.

Para um neurônio qualquer, a atualização de um de seus pesos  $i$ , com o gradiente descendente, é definida como

$$w_i(t+1) = w_i(t) - \alpha \frac{\partial \varepsilon(t)}{\partial w_i(t)}.$$

Para um neurônio da camada de saída, a derivada parcial do erro em função do peso é definida pela [Equação 2.7](#), sendo  $f'$  a derivada da função de ativação do neurônio,  $h_i$  a entrada do neurônio associada ao peso  $w_i$  e  $e$  o erro do neurônio em questão. Para um neurônio de alguma camada escondida, a derivada é definida pela [Equação 2.6](#), sendo o somatório referente a cada uma das  $k$  conexões que o neurônio atual envia sua saída. As deduções dessas fórmulas podem ser encontradas na literatura sobre redes neurais.

$$\frac{\partial \varepsilon(t)}{\partial w_i(t)} = -f'(t)h_i(t) \sum_k w_k(t)f'_k(t)e_k(t) \quad (2.6)$$

$$\frac{\partial \varepsilon(t)}{\partial w_i(t)} = -f'(t)h_i(t)e(t) \quad (2.7)$$

Como mencionado, o processo de avaliação de um modelo consiste na aplicação de alguma medida que verifique a sua capacidade de predição (cf. [Seção 2.4.2](#)). Além disso, para as RNAs, é possível acompanhar a evolução da função de custo durante os passos do treinamento. Também como citado, os dados podem ser separados em treinamento, validação e teste, sendo que a cada etapa do processo é possível apresentar os dados de validação ao modelo intermediário, para analisar suas capacidades de generalização e de predição, naquele exato instante. A partir dos resultados do modelo com essa amostra, são definidas suas melhores configurações, como quantidade de camadas e neurônios, funções de ativação e de custo, entre outras. Logo, o modelo é gerado a partir do subconjunto de treinamento, avaliado com o de validação (que não interfere no treinamento) e aplicado nos dados de teste, que simulam o ambiente no qual a rede será utilizada para efetuar predições em dados não observados durante seu treinamento.

### 2.4.3.1 Redes Neurais Convolucionais

As técnicas de Aprendizado Profundo (*Deep Learning*) são aplicadas para resolver diversos problemas em áreas como Visão Computacional, Processamento de Imagens, Processamento de Linguagem Natural, Análise de Paisagens Acústicas (cf. [Capítulo 3](#)), dentre outras. Elas são um conjunto de metodologias inspiradas em RNA com grande quantidade de camadas (dezenas ou centenas delas) ([GULLI; PAL, 2017](#)). Dentre essas técnicas estão as Redes Neurais Convolucionais - *Convolutional Neural Networks* (CNN), que podem ser vistas como redes que aplicam, de alguma maneira, o conceito de convolução (cf. [Seção 2.2.3](#)) em, pelo menos, uma de suas camadas ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

No lugar de terem pesos unitários em cada conexão, as camadas convolucionais possuem vetores ou matrizes como pesos (filtros ou *kernels*). Eles são aplicados (convoluídos)

nas entradas e os resultados são apresentados para a função de ativação, que gera saídas modificadas (mapas de ativação ou *feature maps*) que serão enviadas para a próxima camada. Essa é a principal diferença entre camadas convolucionais e camadas de uma MLP. Nesta, cada neurônio retorna um valor para cada entrada, enquanto que naquela, a saída é a entrada filtrada, podendo ter ou não suas dimensões modificadas (PONTI *et al.*, 2017). Assim sendo, o processo de aprendizagem em uma CNN busca encontrar os melhores valores para os filtros, de maneira que uma função de custo seja minimizada.

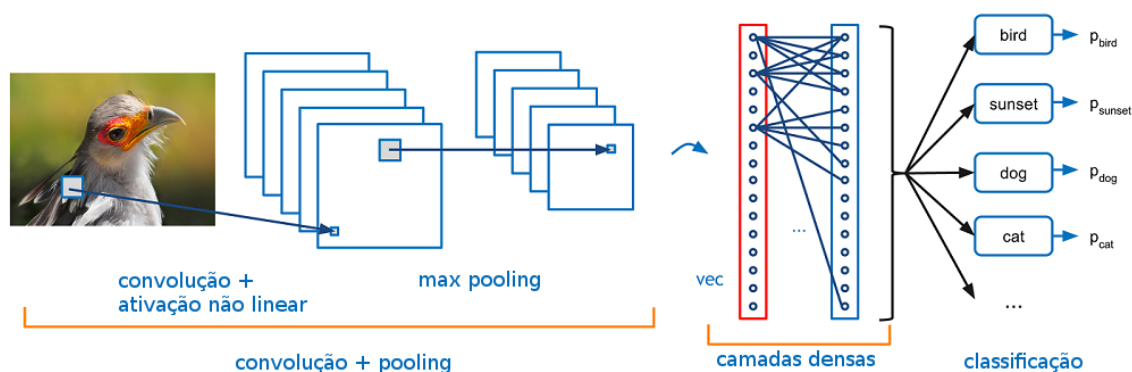
Associadas com camadas convolucionais, são aplicadas camadas de *pooling* que resumem regiões da saída convoluída utilizando, por exemplo, seu valor máximo (*max-pooling*) ou sua média (*average-pooling*). Essa sumarização busca, além da redução dos dados a serem processados, tornar a rede invariante a pequenas modificações dos dados (GOODFELLOW; BENGIO; COURVILLE, 2016). As arquiteturas de CNN empregam, depois de várias camadas convolucionais, camadas densas ou completamente conectadas que funcionam da mesma maneira que as camadas escondidas em uma MLP e executam a predição (PONTI *et al.*, 2017). Dessa maneira, a convolução e o *pooling* extraem características que servem de entrada para a predição realizada pelas camadas densas. Camadas *flatten* são adicionadas entre camadas convolucionais e densas para “nivelar” os dados, transformando, por exemplo, uma matriz em um vetor. Também são usadas camadas de *dropout*, que durante o treinamento ignoram um percentual de suas entradas, com a finalidade de evitar problemas de generalização do modelo (SRIVASTAVA *et al.*, 2014). Além delas, camadas de normalização como *batch normalization* (IOFFE; SZEGEDY, 2015) e *layer normalization* (BA; KIROS; HINTON, 2016) são consideradas para tornar o processo de treinamento mais estável e eficiente.

A Figura 8 é um exemplo da aplicação de CNN na classificação de imagens de animais. No início, uma camada convolucional é aplicada na imagem gerando  $n$  *feature maps*. Depois disso, uma camada de *pooling* é aplicada para reduzir as dimensões dos dados. Esses dados são transformados em vetores e enviados para as camadas densas. Essas camadas, por sua vez, são responsáveis por aprender, com base nas características geradas pelas camadas anteriores, as probabilidades da imagem estar associada a algum dos rótulos considerados.

As camadas convolucionais e densas utilizam diversas funções de ativação, além das sigmóides usadas pela MLP. Funções como *rectified linear unit* (ReLU), definida como  $relu(x) = \max(0, x)$ , são usadas nas camadas de convolução por serem mais simples, mais eficientes no cálculo e na propagação do gradiente. Além do mais, funções como *softmax* são postas na última camada densa, para representar a distribuição de probabilidades associada às classes de um problema com mais de duas classes (GOODFELLOW; BENGIO; COURVILLE, 2016).

No que diz respeito às funções de custo, além de funções como erro quadrático,

Figura 8 – Arquitetura de uma CNN aplicada em classificação de imagens.



Fonte: Adaptada de [Deshpande \(2016\)](#).

são aplicadas funções baseadas na ideia de entropia cruzada, que calcula a diferença entre as probabilidades geradas pelo *softmax* e a distribuição esperada das classes ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

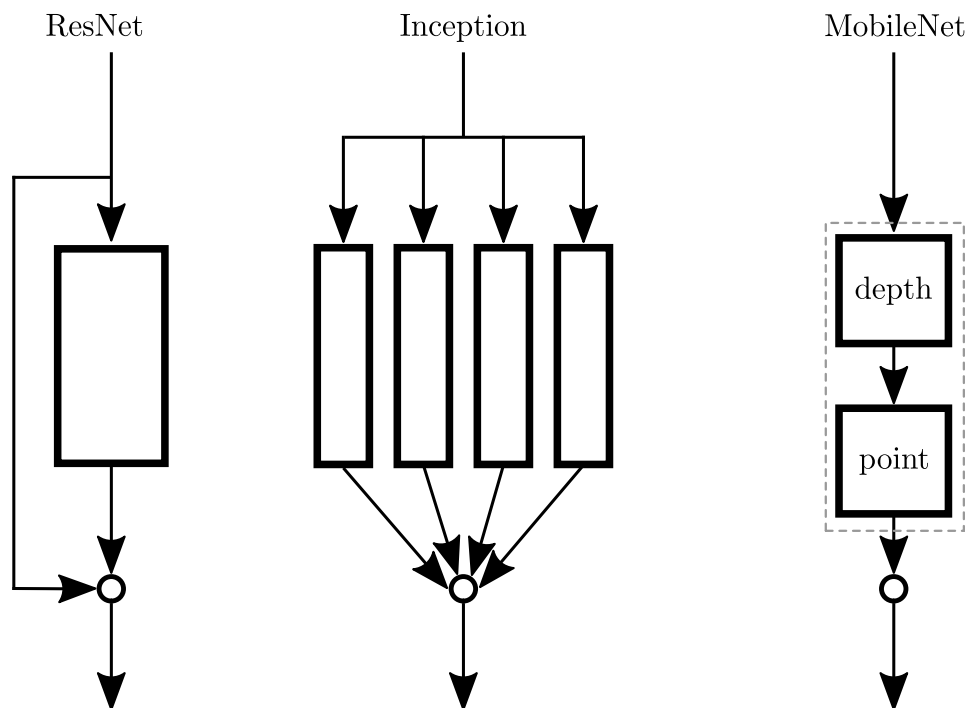
Por fim, no que diz respeito à otimização, a aplicação do gradiente descendente no cenário de *Deep Learning* não é viável. Isso porque a quantidade de exemplos a serem considerados e a quantidade de parâmetros a serem atualizados é enorme (mais de milhares), o que inviabiliza o carregamento desses dados e degrada o tempo de processamento ([PONTI et al., 2017](#)). Um dos otimizadores mais empregados nesse tipo de cenário é o *Stochastic Gradient Descent* (SGD) ([BOTTOU, 1998](#)), uma variação do gradiente descendente, que no lugar de usar todas as amostras de treinamento, pesos e saídas no processo de atualização, escolhe de maneira aleatória um subconjunto delas (*mini-batch* ou *batch*). Além dele, existem outros otimizadores empregados, como o *Root Mean Square Propagation* (RMSProp) ([GOODFELLOW; BENGIO; COURVILLE, 2016; TIELEMAN; HINTON et al., 2012](#)), *Adaptive Moment Estimation* (Adam) ([KINGMA; BA, 2014](#)) e suas variações. Cada método supõe algo sobre o gradiente produzido durante o treinamento e pondera a adaptação dos pesos utilizando diferentes momentos do gradiente e passos adaptativos. Assim, diferentes métodos pedem ajustes adequados das taxas de aprendizado e tamanho do *mini-batch* para melhor convergência ([BECHER; PONTI, 2021](#)).

#### 2.4.3.2 Exemplos de arquiteturas

Existem muitas arquiteturas de CNNs empregadas em uma vasta quantidade de tarefas de diferentes domínios. Como exemplo, pode-se mencionar a ResNet-50 ([HE et al., 2016](#)), uma rede residual que é uma arquitetura de CNN que busca melhorar a capacidade de representação sem que o gradiente do erro desapareça. Como visto, o processo de atualização dos pesos depende do cálculo do gradiente (vetor de derivadas parciais de uma função) do erro, que pode gerar valores próximos de zero (*vanishing gradient*) conforme

o *backpropagation* avança nas camadas internas, tornando a atualização dos pesos impraticável e afetando a aprendizagem do modelo. Para contornar esse problema, as ResNets são construídas com blocos residuais, como apresentado na Figura 9, que combinam a saída de um conjunto de camadas consecutivas com a entrada desse conjunto, o que evita o desaparecimento do gradiente. Com isso, essas arquiteturas podem ter mais camadas, o que melhora a sua capacidade de representação de padrões e generalização, especialmente quando a quantidade de amostras disponíveis para treinamento também é vasta (milhares de amostras). A ResNet-50 ( $\approx 50$  camadas convolucionais) é empregada em vários problemas de reconhecimento de imagem e encontrada em tarefas de classificação de sons como as descritas no Capítulo 3.

Figura 9 – Visão geral de blocos básicos de arquiteturas de CNN.



Fonte: Elaborada pelo autor.

Enquanto isso, a história da Inception-V3 (SZEGEDY *et al.*, 2016) começa com a GoogLeNet (SZEGEDY *et al.*, 2015) (Inception-V1) que utiliza abordagens para melhorar tanto a velocidade de treinamento quanto a acurácia do modelo. Uma tarefa não trivial na modelagem de uma CNN é a definição do tamanho dos filtros para trabalharem com padrões locais (filtros menores) ou globais (filtros maiores). Além disso, um aumento da quantidade de camadas pode melhorar a capacidade de representação da rede, mas torna o treinamento mais complexo e lento. Para contornar esses problemas, as Inceptions são construídas com módulos de camadas paralelas, aumentando a largura da rede como na Figura 9, com filtros pequenos, no lugar de escolher tamanhos arbitrários de filtros para cada padrão a ser processado. Por exemplo, uma entrada passa por um módulo que possui

4 blocos paralelos, sendo que um deles possui uma camada convolucional com filtros de dimensões  $1 \times 1$  seguida por duas camadas com filtros  $3 \times 3$ ; outro bloco é semelhante ao anterior, mas com uma camada com filtros de dimensões  $3 \times 3$ ; o próximo bloco contém uma camada de *pooling* seguida por convolução com filtros  $1 \times 1$ ; e o último bloco possui convoluções de tamanho  $1 \times 1$ . Os resultados desses quatro blocos são concatenados e enviados para o próximo módulo da rede. Outra ideia desenvolvida nesses modelos é a fatoração das convoluções. No lugar de uma convolução  $n \times m$ , é possível executar uma convolução com dimensões  $1 \times m$  seguida por uma convolução  $n \times 1$ , o que reduz a quantidade de operações totais. Com esse tipo de abordagem, as variações dessa arquitetura otimizam o uso dos recursos e possuem capacidade de discriminar padrões em diferentes escalas, o que melhora a acurácia das predições e impulsiona o seu uso em tarefas de reconhecimento de imagens.

No caso da MobileNet-V3 (HOWARD *et al.*, 2019), reduzir a demanda por recursos é necessário para que os modelos sejam executados em dispositivos móveis de maneira eficiente, sem perder eficácia. A arquitetura inicial parte da separação da convolução em duas etapas, uma *depthwise* e outra *pointwise* (HOWARD *et al.*, 2017), como apresentado na Figura 9. No lugar de um filtro convolucional de dimensões  $n \times m \times c$  processar uma entrada  $w \times h \times c$  e gerar uma saída de duas dimensões  $w \times h$ , na convolução *depthwise*, os  $c$  canais da entrada não são combinados durante a convolução mantendo as dimensões  $w \times h \times c$  na saída. Nesse resultado, é aplicado o passo *pointwise*, que executa uma convolução convencional, com filtros  $1 \times 1 \times c$ , agora gerando a saída com dimensões  $w \times h$ . Isso pode aparentar maior quantidade de operações, por outro lado, a quantidade de multiplicações<sup>4</sup> executadas da maneira convencional pode ser descrita como  $(w \times h \times c) \times (n \times m \times k)$ , sendo  $k$  a quantidade de filtros de uma camada. Com a separação em dois passos a quantidade de multiplicações se torna  $(w \times h \times c) \times (n \times m + k)$ . Outros artifícios como expansão e compressão das dimensões processadas nos blocos, uso dos resíduos definidos na ResNet, modificações das funções de ativação, entre outros, são considerados para melhorar a capacidade de generalização na MobileNet-V3, além de tornar o consumo de recursos mais eficaz.

### 2.4.3.3 Avaliação da estrutura das redes

Além das medidas apresentadas na Seção 2.4.2, é possível estudar a estrutura interna das redes, os pesos que elas aprendem, os espaços de características gerados, a similaridade entre camadas de uma rede e camadas de redes distintas etc. Abordagens visuais aplicam Projeção Multidimensional (NONATO; AUPETIT, 2018) e o Coeficiente de Silhueta (TAN; STEINBACH; KUMAR, 2005) para avaliar os espaços de características aprendidos pelas diversas camadas das redes, revelando estruturas e relações entre

<sup>4</sup> O tempo de execução de multiplicações é maior do que operações como adição, por isso apenas ela é considerada.



os dados e ajudando a avaliar a capacidade de gerar grupos de dados coesos com padrões similares e a segregação de padrões distintos. Essas características são comunicadas pela projeção por meio da disposição visual das amostras em um mapa de pontos, enquanto que a silhueta pode comunicar a qualidade do espaço de características (COIMBRA, 2016), retornando valores entre  $[-1, 1]$ , sendo a melhor coesão/segregação representada por valores próximos de um. Além disso, os resultados de silhueta são incrementados se a distância entre os grupos aumentar ou reduzidos se mais grupos forem adicionados.

Uma maneira simples de comparar a estrutura de duas redes, ou até mesmo de representar a estrutura interna de cada uma delas, é calcular a similaridade entre as representações geradas por suas camadas. Para isso, Kornblith *et al.* (2019) motivaram e introduziram a medida de similaridade *Centered Kernel Alignment* (CKA) que é aplicada em trabalhos de análise da estrutura de CNNs para, por exemplo, analisar variações de profundidade e largura de redes neurais (NGUYEN; RAGHU; KORNBLITH, 2020). A maneira mais simples de cálculo dessa similaridade é dada por

$$CKA(X, Y) = \frac{\|X^T Y\|_F^2}{\|X^T X\|_F \|Y^T Y\|_F},$$

onde  $X$  e  $Y$  são mapas de ativação gerados por duas camadas,  $\|\cdot\|_F$  a norma de Frobenius e os resultados de similaridade variando entre  $[0, 1]$ , com o maior valor relacionado à maior similaridade entre  $X$  e  $Y$ . Com esses valores, uma estrutura de visualização como um mapa de calor pode ser criada, com os índices das linhas e colunas representando camadas ou blocos das redes comparadas, sendo que essa comparação pode ser executada nas camadas de uma mesma rede (KORNBLITH *et al.*, 2019; NGUYEN; RAGHU; KORNBLITH, 2020).

#### 2.4.4 Técnicas para melhoria do aprendizado

Como mencionado, o treinamento de uma CNN pode enfrentar problemas como desbalanceamento de classes e pouca variação dos dados. De maneira simplista, isso pode ser resolvido com a coleta de mais amostras para o treinamento. Entretanto, a aquisição de mais dados rotulados e com boa variabilidade não é trivial, porque consome tempo, processamento e dinheiro. Dentre as abordagens empregadas para mitigar esses problemas está o aumento de dados, que busca expandir o tamanho e a variação de uma base de dados e balancear suas classes, gerando cópias das amostras existentes a partir da manipulação de suas características, de maneira que o significado original dos dados não seja perdido (FLORENTIN; DUTOIT; VERLINDEN, 2020; PONTI *et al.*, 2021). No cenário de classificação ou detecção de padrões de som, modificação de faixas de frequência (*pitch shifting*) ou amplitude do sinal (*amplitude change*), deslocamento do sinal no tempo (*time stretching*) e adição de algum tipo de ruído (*noise addition*) são algumas ferramentas



recorrentes (LOSTANLEN *et al.*, 2019; PARASCANDOLO; HUTTUNEN; VIRTANEN, 2016; SALAMON; BELLO, 2017).

Outra maneira de melhorar o processo de treinamento, sobretudo quando a quantidade de amostras é reduzida, é a Transferência de Aprendizado. Ela consiste em treinar uma rede em uma base de dados de um problema genérico, que possui quantidade razoável de amostras, como é o caso da base de imagens ImageNet (DENG *et al.*, 2009) ou da base de sons AudioSet (GEMMEKE *et al.*, 2017). Essa rede pré-treinada pode ser usada como um gerador de características ou seus pesos podem servir como inicialização para o treinamento em outra base de dados, melhorando a sua generalização no segundo problema (GOODFELLOW; BENGIO; COURVILLE, 2016). Isso acontece porque os pesos aprendidos podem ser significativos mesmo para dados de domínios diferentes (PONTI *et al.*, 2017). Por exemplo, alguns trabalhos de classificação de sons ambientais usam redes pré-treinadas na classificação de bases de imagens genéricas e refinam seus pesos em uma base específica de espectrogramas ou usam a rede para extrair características usadas como entrada para o treinamento de outros classificadores, como em LeBien *et al.* (2020), Strout *et al.* (2017) e no levantamento realizado por Dufourq *et al.* (2022).

### 2.4.5 Aprendizado Autossupervisionado

Outra maneira recorrente de pré-treinamento de uma rede é a criação de uma tarefa de classificação auxiliar, onde os rótulos possam ser criados a partir dos próprios dados e os pesos aprendidos são transferidos para tarefas específicas, alcançando resultados similares às redes treinadas em tarefas puramente supervisionadas. Em um contexto de imagens, a tarefa auxiliar pode rotacioná-las e treinar a rede para prever essa rotação (KOMODAKIS; GIDARIS, 2018), prever a vizinhança de uma região da imagem (DOERSCH; GUPTA; EFROS, 2015), prever as cores de uma imagem (ZHANG *et al.*, 2017), entre outras. Com esse processo de autossupervisão (*self-supervised learning*) (SA, 1994) é possível pré-treinar as redes com uma vasta quantidade de dados disponíveis na internet, o que leva a resultados consideráveis em tarefas como classificação de fonemas e identificação de falantes (CHI *et al.*, 2021), localização de fontes sonoras e reconhecimento audiovisual de ações (OWENS; EFROS, 2018), reconhecimento e síntese de fala (BAEVSKI *et al.*, 2020), dentre outras tarefas de Processamento de Linguagem Natural, Processamento de Áudio e Visão Computacional.

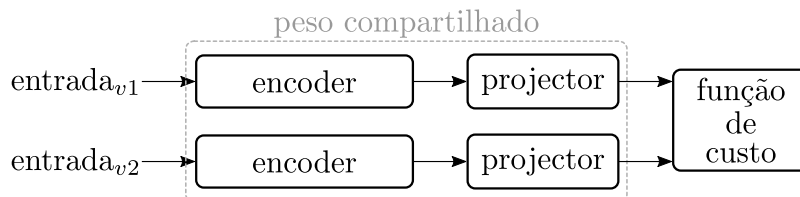
Também é comum o aprendizado não supervisionado de espaços conjuntos de características, que possam ser invariantes a distorções das entradas. Essa técnica é construída com arquiteturas que possuem mais de um ramo, que buscam aprender espaços similares para visões distintas dos dados (BARDES; PONCE; LECUN, 2022). Infelizmente, esse tipo de estrutura pode gerar o colapso do espaço de características, quando os espaços aprendidos são constantes ou sem informação relevante da estrutura dos dados. Para mi-

tigar esses problemas, abordagens como a Barlow Twins (ZBONTAR *et al.*, 2021) foram propostas. Nesse caso, como na Figura 10, a estrutura contém dois blocos consecutivos, sendo que o primeiro (*encoder*) é uma ResNet-50 sem a camada de classificação e o segundo (*projector*) é composto por 3 camadas densas de  $N$  unidades cada, uma camada de *batch normalization* antes da ativação ReLU das duas primeiras e apenas a ativação linear para a última camada densa. O *projector* é aplicado para remover redundâncias dos espaços gerados pelo *encoder*. Essa sequência de blocos é duplicada, gerando dois ramos com pesos compartilhados, como em uma rede Siamesa (BROMLEY *et al.*, 1994; CHOPRA; HADSELL; LECUN, 2005), que recebem como entradas visões diferentes de uma mesma imagem. A ideia é forçar que os espaços gerados para as visões da imagem sejam mais próximos, minimizando a redundância entre os componentes das suas características, sendo a função de custo definida como

$$L_{BT} = \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariância}} + \lambda \underbrace{\sum_i \sum_{i \neq j} C_{ij}^2}_{\text{reduz redundância}},$$

onde  $\lambda$  é uma constante positiva que balanceia a relevância dos termos da função e  $C$  é a matriz de correlação cruzada entre as saídas dos projetores de cada ramo da rede. Para o cálculo dessa matriz, os espaços gerados pelo *projector* são padronizados<sup>5</sup> e o produto interno entre os dois é dividido pelo tamanho da *batch*. O termo de invariância leva a correlação cruzada entre os espaços a se aproximar da matriz identidade, forçando a aproximação dos espaços aprendidos pelo *encoder*. Enquanto isso, o segundo termo da função reduz a redundância nesses espaços.

Figura 10 – Estrutura básica da arquitetura usada na Barlow Twins e VICReg



Fonte: Elaborada pelo autor.

Depois do treinamento, o *encoder* é extraído, configurado e refinado para uma tarefa específica. Os autores destacam a capacidade dessa abordagem de gerar resultados superiores aos obtidos por outras técnicas em tarefas de classificação semissupervisionada na ImageNet, além de resultados similares aos de técnicas do estado da arte para tarefas supervisionadas de classificação e detecção de objetos na mesma base.

<sup>5</sup> Média zero e desvio padrão igual a um para cada característica

Em uma variação dessa ideia, Bardes, Ponce e Lecun (2022) propuseram a *Variance-Invariance-Covariance Regularization* (VICReg), um método de regularização que também busca evitar o colapso dos espaços de características. Diferente de outras abordagens, ela não demanda compartilhamento de pesos entre os ramos da rede, normalização desses ramos ou das características geradas, quantização de saídas etc., nem mesmo exige que os ramos da rede sejam idênticos ou similares. Com isso, é possível considerar ramos com entradas de diferentes domínios como imagem, som e vídeo. Os autores destacam a capacidade de estabilização do processo de treinamento, levando a melhorias de desempenho com resultados próximos aos do estado da arte em várias tarefas específicas para refinamento.

Mesmo não exigindo compartilhamento de pesos ou similaridade dos ramos, o artigo da VICReg segue a mesma estrutura da Barlow Twins (cf. Figura 10) para os testes iniciais e executa testes adicionais flexibilizando essa estrutura. Os componentes também são um *encoder*, um *projector*, nomeado como *expander*, e a função de custo é definida como

$$\ell(Z_a, Z_b) = \underbrace{\lambda \frac{1}{n} \sum \text{mse}(Z_a, Z_b)}_{\text{invariância}} + \underbrace{\mu [v(Z_a) + v(Z_b)]}_{\text{variância}} + \underbrace{\nu [c(Z_a) + c(Z_b)]}_{\text{covariância}},$$

sendo  $Z_i$  uma *batch* com os  $n$  vetores aprendidos pelo *expander* de cada ramo da arquitetura e  $\lambda$ ,  $\mu$  e  $\nu$  hiperparâmetros. O termo de invariância é definido como a média do erro quadrático entre as entradas, enquanto que a variância e covariância são definidas como

$$v(Z) = \frac{1}{n} \sum \text{relu}(\gamma - \sqrt{\text{var}(Z) + \varepsilon}),$$

$$c(Z) = \frac{1}{N} \sum_i \sum_{i \neq j} C(Z)_{ij}^2, \quad C(Z) = \frac{\|Z - \text{mean}(Z)\|_2}{N - 1},$$

sendo  $N$  a quantidade de características gerada pelo *expander*,  $\gamma = 1$ ,  $\varepsilon = 10^{-4}$  e as funções *mean* e *var* calculadas ao longo da *batch*. Nessa função de custo, o termo de invariância possui o mesmo propósito da Barlow Twins, o termo de variância faz com que a variação entre os vetores aprendidos tenha desvio padrão unitário, não permitindo que eles convirjam para o mesmo ponto. O termo de covariância reduz a redundância dos espaços por meio da redução da correlação entre os vetores aprendidos.

### 2.4.6 Quantificação

Além de tarefas supervisionadas, como classificação e regressão, quantificação foi abordada em uma série de trabalhos recentes, como Moreo e Sebastiani (2022) e Schuma-

cher, Strohmaier e Lemmerich (2021). Essa é uma tarefa que busca estimar a distribuição de classes no lugar de prever os rótulos de suas instâncias (GONZÁLEZ-CASTRO; ALAIZ-RODRÍGUEZ; ALEGRE, 2013; MALETZKE *et al.*, 2019; MALETZKE; REIS; BATISTA, 2017). Esse tipo de abordagem tem sido aplicada para análise de sentimentos em redes sociais (MOREO; SEBASTIANI, 2022), para avaliação da distribuição de classes em *data streaming* (MALETZKE; REIS; BATISTA, 2017), para verificação da distribuição de classes relacionadas a plânctons em recifes de corais (BEIJBOM *et al.*, 2015), entre outras. Nesse cenário, como o foco está na estimativa das distribuições de probabilidades das classes, o que sumariza suas informações, essas distribuições podem ser diferentes nos dados de treinamento e teste, além de não pressupor dados independentes e identicamente distribuídos (BEIJBOM *et al.*, 2015; GONZÁLEZ *et al.*, 2017b).

Existem vários métodos para quantificação, desde os mais simples que aplicam apenas contagem (FORMAN, 2005; GAO; SEBASTIANI, 2016), aos mais sofisticados que buscam modelar probabilidades (BELLA *et al.*, 2010; MALETZKE *et al.*, 2019). O mais simples desses métodos é o *classify and count* (CC) (BEIJBOM *et al.*, 2015; GAO; SEBASTIANI, 2016) que define uma distribuição  $\hat{p}$ , como

$$\hat{p}_{T_e}(c_i) = \frac{\sum_{x \in T_e} \mathbb{1}[f(x) = c_i]}{|T_e|},$$

sendo  $T_e$  um conjunto de teste,  $c_i$  a classe a ser contada,  $f$  um classificador qualquer treinado, e  $\mathbb{1}[cond]$  retorna um se *cond* é verdadeira e zero para falso.

Como essa contagem é dependente da eficácia de um classificador, a quantificação pode ser imprecisa. Para corrigir variações e estimativas erradas, o quantificador *adjusted classify and count* (ACC) (FORMAN, 2005; FORMAN, 2008) é uma abordagem a ser considerada. Ele ajusta de maneira simples os resultados do CC para os dados de teste, a partir das taxas de verdadeiro e falso positivos da classificação dos dados de treinamento, definindo uma distribuição ajustada  $\hat{p}'$  da seguinte maneira

$$\hat{p}'_{T_e}(c_i) = \frac{\hat{p}_{T_e}(c_i) - FP}{VP - FP},$$

sendo que os resultados são truncados para permanecerem dentro do intervalo  $[0, 1]$ .

Para avaliar os resultados de quantificação de cada classe, é possível empregar o erro absoluto  $|\hat{p}(c_i) - p(c_i)|$ , o erro quadrático  $|\hat{p}(c_i) - p(c_i)|^2$  ou uma divergência entre as distribuições real e predita  $|p(c_i) \log \frac{p(c_i)}{\hat{p}(c_i)}|$ . Outros métodos de quantificação e avaliação dos seus resultados podem ser encontrados em trabalhos como González *et al.* (2017a).

## 2.5 Considerações finais

A escolha dos conceitos apresentados neste capítulo considerou os passos propostos no [Capítulo 4](#) e sua implementação. Assim sendo, o conteúdo sobre CNN, suas arquiteturas, avaliação e abordagens para redução de problemas encontrados durante seu treinamento é necessário para o entendimento dos demais capítulos. As ferramentas de quantificação são o foco do [Capítulo 5](#), enquanto que as variações na representação dos áudios, seja ela por vetores de características ou imagens de espectrogramas, são essenciais para o desenvolvimento do [Capítulo 6](#). Além disso, a aplicação de autossupervisão para o pré-treinamento das redes é o ponto principal do [Capítulo 7](#). Por fim, os métodos de avaliação são aplicados em todos esses capítulos.



---

## TRABALHOS RELACIONADOS

---

Neste capítulo é apresentado um levantamento de pesquisas que aplicam redes convolucionais para a identificação de espécies de pássaros e anuros a partir das suas vocalizações. Ele não visa ser uma revisão completa ou exaustiva, mas proporcionar uma visão geral sobre as práticas de *Deep Learning* empregadas para classificação de espécies de animais a partir de seus sons. Uma revisão sistemática e abrangente dessas práticas pode ser encontrada em trabalhos como o de [Stowell \(2022\)](#). Além disso, em nível de comparação, são mencionadas abordagens aplicadas em ambiente marinho, onde as características de propagação do som, de ruídos e de densidade de eventos acústicos diferem das encontradas em ambientes terrestres. Por fim, o texto apresenta uma sumarização no [Quadro 1](#) e discute esses trabalhos, buscando identificar suas semelhanças e suas relações com a proposta desta pesquisa.

### 3.1 Classificação de sons de pássaros

Esta seção descreve exemplos de trabalhos que aplicaram redes neurais para detectar e identificar espécies de pássaros em condições diversas, além de identificar outros níveis de taxonomia, como família das espécies e a ordem na qual elas se encontram. No primeiro exemplo, [Salamon et al. \(2017\)](#) exploraram técnicas para classificação de chamado de voo (*flight calls*) de pássaros, emitido durante migração noturna. Eles testaram duas técnicas, uma de aprendizagem não supervisionada de características e outra de *Deep learning*, ambas com acurácia aproximada de 94%. Uma combinação delas alcançou 96% de acurácia na classificação de 43 espécies de aves migratórias.

O processo não supervisionado (nomeado como SVM-SKM) consiste no aprendizado de um dicionário capaz de codificar a informação contida nos arquivos de áudio. Para criação do dicionário, são gerados mel-espectrogramas com janela de Hanning de tamanho 11,6 ms (256 amostras), *hop size* 1,45 ms (32 amostras) e 40 bandas mel, cobrindo

a faixa entre 2000-11025 Hz (taxa de amostragem dos arquivos de 22050 Hz). Cada mel-espectrograma foi separado em partes (*patches*) que compreendem 46 ms e todo o espectro de frequências, com janelas deslizantes com níveis de sobreposição. Esses *patches* passam por uma transformação com *Principal Component Analysis* (PCA) (PEARSON, 1901) para remover correlações e forçar que suas características possuam variância unitária (PCA-*whitennig*). O PCA retorna quantidade suficiente de componentes para representar 99% da variância dos dados. Todos os *patches* foram convertidos em vetores de  $n$  características para que uma matriz  $X \in \mathbb{R}^{n \times m}$  seja construída com todos os  $m$  *patches* gerados para a base de treinamento (SALAMON *et al.*, 2016).

A matriz  $X$  serve como entrada para o *spherical k-means* (SKM) (DHILLON; MODHA, 2001), com  $k = 256$ . Os centroides gerados por esse agrupamento são organizados em uma matriz  $D \in \mathbb{R}^{n \times k}$  (*codebook*) usada como dicionário de características. Dessa maneira, para codificar um arquivo de áudio, o processo anterior para criação de *patches* é aplicado, criando uma matriz  $M \in \mathbb{R}^{n \times p}$ , sendo  $p$  a quantidade de *patches* desse arquivo, que será multiplicada pelo dicionário para gerar a representação  $F \in \mathbb{R}^{p \times k}$ . As colunas são sumarizadas com média, desvio padrão e valor máximo, sendo que a representação final de cada arquivo é composta  $3 \times k$  posições. Essas características foram usadas como entrada para treinar um Support Vector Machine (SVM) radial.

Como CNN, Salamon *et al.* (2017) consideraram o modelo proposto por Salamon e Bello (2017), com 3 camadas convolucionais de 2 dimensões ( $5 \times 5$ ), intercaladas por *max-pooling* ( $4 \times 2$ ), sendo que a primeira possui 24 filtros e as outras duas 48. As camadas de *pooling* foram posicionadas antes de cada ativação ReLU, exceto na terceira que não possui *pooling*. No final da rede, depois de uma camada *flatten*, existem duas camadas densas com *dropout* de 0,5 antes de suas entradas e normalização  $L^2$ , com penalização  $10^{-3}$ . O SGD foi usado para otimizar a função *cross-entropy*, em 100 épocas<sup>1</sup>. A entrada para esse modelo é o mesmo mel-espectrograma usado para aprendizagem não supervisionada, entretanto com 128 bandas.

Os autores também testaram combinações das predições da rede e do classificador com características não supervisionadas. Os melhores resultados foram obtidos aplicando média geométrica entre as posições dos vetores de probabilidades retornados pela CNN e pelo SVM-SKM e retornando a classe da posição com maior valor.

Além do mais, um *baseline* foi definido com MFCCs como características de entrada para um SVM radial. Para geração dos coeficientes, foram considerados os 25 primeiros coeficientes, gerados a partir de mel-espectrogramas com 40 bandas mel. Cada componente foi sumarizado por 11 medidas estatísticas (valores mínimo e máximo, mediana, media, variância, assimetria, curtose, além da média e variância da primeira e segunda derivadas) (SALAMON; JACOBY; BELLO, 2014), gerando um vetor de características

<sup>1</sup> Sem informação de tamanho de *batch* no artigo



de 275 posições.

Para treinamento e teste os autores consideraram dados do CLO-43SD (SALAMON *et al.*, 2016), que contém 5428 clips de áudios de 43 espécies de pássaros da América do Norte. Esses arquivos foram gravados em condições diferentes (gravação com e sem ruído, gravações de pássaros em cativeiro) e possuem apenas um chamado de uma das espécies. Além desses dados, seguindo o proposto por Salamon e Bello (2017), também foram aplicadas técnicas de aumento de dados para gerar mais amostras, adicionando ruído de fundo (4 ruídos contendo geofonia, capturados à noite), compressão (*dynamic range compression*), modificação de faixas de frequência e deslocamento do sinal no tempo.

Os treinamentos foram executados com validação cruzada com  $k = 5$ , usando uma partição para validação, outra para teste e as demais para treinamento. Para analisar os resultados, foram consideradas medidas de sumarização da acurácia das partições (uso de gráficos *boxplot*).

Salamon *et al.* (2017) destacaram que a acurácia obtida pelo *baseline* é de 0,85 (valores no intervalo  $[0, 1]$ ), enquanto que os resultados médios do SVM-SKM e da CNN estão próximos a 0,94. A combinação desses dois classificadores alcançou 0,96 de acurácia para a tarefa de classificação de 43 espécies migratórias.

No segundo exemplo com sons de aves migratórias, Lostanlen *et al.* (2019) desenvolveram um modelo capaz de detectar a presença de vocalizações de aves produzidas durante migração noturna, com uma área de 76% sob a curva do gráfico de precisão-sensibilidade. Os autores combinaram e avaliaram técnicas de *Deep Learning*, diferentes entradas e aumento de dados para alcançar os resultados mais apropriados, sendo que o melhor modelo foi disponibilizado como parte de uma biblioteca de programação<sup>2</sup>.

Os pesquisadores testaram uma variação de *context-adaptive neural network* (CA-CNN) (DELCROIX *et al.*, 2015), que possui dois ramos de processamento, onde o ramo principal segue o modelo proposto por (SALAMON; BELLO, 2017), uma CNN com 3 camadas convolucionais intercaladas por camadas de *pooling* e 2 camadas densas no final. Diferente do original, a penúltima camada densa possui regularização  $L^2$ , com penalidade  $10^{-3}$  e a última tem ativação sigmoide. O ramo auxiliar da rede é composto por uma camada convolucional com 8 filtros de dimensões  $1 \times 32$  e uma camada densa com 64 unidades, ambas com ativação ReLU.

A combinação dos ramos é efetuada a partir de transformações lineares derivadas da formulação básica de um neurônio, como apresentado na Seção 2.4.3. Assim sendo, o vetor  $\mathbf{x}$  pode ser considerado como as saídas dos ramos da rede, e o viés  $b$  e os pesos  $\mathbf{w}$  são aprendidos durante o treinamento. A combinação dos ramos nesse processo foi testada a partir de três variações: *Adaptive weights* (AW), *Adaptive threshold* (AT) e *Mixture*

<sup>2</sup> <<https://github.com/BirdVox/birdvoxdetect>>

of Experts (MoE). O treinamento do modelo foi realizado com o otimizador Adam<sup>3</sup>.

Para treinamento e teste, Lostanlen *et al.* (2019) consideraram duas bases de áudios coletadas com 6 microfones cada: BirdVox-70k (LOSTANLEN *et al.*, 2018c) para detecção da presença de vocalizações de aves migratórias durante o voo, em áudios de 150 ms; e BirdVox-full-night (LOSTANLEN *et al.*, 2018b) para detecção de eventos em arquivos contínuos de aproximadamente 10 h, sendo 6 arquivos no total. Como avaliação, os autores consideraram a taxa de erros e área sob a curva do gráfico de precisão-sensibilidade, além da comparação dos resultados com a CNN proposta em Lostanlen *et al.* (2018b). O processo de treinamento foi executado com validação cruzada *leave-one-out*<sup>4</sup>, sendo que cada partição corresponde aos dados gravados por um microfone específico.

Além dos dados originais, foram empregadas técnicas de aumento de dados para melhorar o treinamento, como modificação de faixas de frequência e deslocamento do sinal no tempo, nomeadas como *Geometrical Data Augmentation* (GDA); e adição de ruídos de um microfone em arquivos gerados por outro microfone, nomeada como *Adaptative Data Augmentation* (ADA). Os autores empregaram GDA nos testes com a CA-CNN e ADA aos demais testes.

Para o ramo principal da arquitetura e para o *baseline*, foram testados como entrada o PCEN e o mel-espectrograma, enquanto que o ramo auxiliar recebe um conjunto de informações estatísticas (quartil, decil, percentil, por mil) extraídas do PSD.

Logo, foram testados 12 cenários para cada base de dados, combinando dois tipos de entrada, 3 técnicas de combinação dos ramos e uso ou não de aumento de dados. Nas detecções de presença de chamados na base BirdVox-70k, os modelos que foram treinados com aumento de dados GDA, entrada PCEN e combinação AT, alcançaram taxas de erro menores do que 10%, sendo que as demais combinações geraram taxas superiores, chegando a  $\approx 20\%$  de erro. Na base BirdVox-full-night, o melhor resultado também foi alcançado pela combinação anterior (GDA, PCEN, AT), com resultados acima de 76% de área sob a curva de precisão-sensibilidade, contra  $\approx 56\%$  do *baseline*.

Também analisando espécies migratórias, o terceiro exemplo desta seção adiciona alguns graus de complexidade ao processo. Cramer *et al.* (2020) desenvolveram a TaxoNet, uma rede neural capaz de classificar espécies de pássaros, além de identificar outros níveis dentro da taxonomia, como as famílias das espécies e a ordem onde se encontra essas famílias. Dentro de uma mesma ordem, essa CNN identifica 4 famílias e 14 espécies de aves migratórias, alcançando resultados de *micro-averaged* e *macro-averaged* da acurácia de 66,33% e 55,59%. O modelo pré-treinado está disponível na biblioteca BirdVoxClassify<sup>5</sup>,

<sup>3</sup> Os demais parâmetros, como taxa de aprendizagem, não estão descritos no texto do artigo

<sup>4</sup> As partições de teste possuem apenas uma amostra

<sup>5</sup> <<https://github.com/BirdVox/birdvoxclassify>>

uma dependência da BirdVoxDetect<sup>6</sup>.

Para isso, os autores testaram três modelos, sendo que o primeiro, usado como *baseline*, retorna um rótulo por arquivo de áudio (*single-task*). Esse modelo, que tem por base o proposto por Salamon *et al.* (2017), possui no início uma camada de *batch normalization*, seguida por três camadas convolucionais com, nessa ordem, 24, 48 e 48 filtros de dimensões  $5 \times 5$  e ativação ReLU. As duas primeiras camadas convolucionais são seguidas por *max pooling* de tamanho e *stride*  $2 \times 2$ . Depois de uma camada *flatten*, existem duas camadas densas, uma com 64 unidades e ativação ReLU e a outra com ativações *softmax* e quantidade de unidades relacionada com a tarefa de classificação específica. Em todas as camadas, o peso do viés (*bias weight*) foi desconsiderado.

Além desse, Cramer *et al.* (2020) criaram dois modelos para retorno de várias classes (*multi-task*). Um não hierárquico, onde eles usaram o mesmo modelo descrito no parágrafo anterior, mas a última camada de classificação foi substituída por três camadas densas paralelas, que retornam respectivamente a ordem (uma unidade com ativação sigmoide), a família (4 unidades com ativação *softmax*) e a espécie (14 unidades com ativação *softmax*).

O modelo hierárquico TaxoNet parte da ideia de que camadas superficiais (próximas da entrada) classificam categorias genéricas, como ordem, e camadas profundas classificam categorias específicas, como família e espécie. Dessa maneira, a arquitetura segue o fluxo básico dos dois modelos apresentados até agora, entretanto, após a camada *flatten* existem dois ramos. No primeiro, uma camada densa com uma unidade e função sigmoide identifica a presença da ordem (*coarse-level prediction*). No outro ramo, uma camada densa de 64 unidades e ativação ReLU foi aplicada e suas saídas particionadas em  $n$  subconjuntos relacionados com as famílias de espécies, de maneira que a quantidade de características em cada partição seja proporcional a quantidade de espécies em cada família. Essa separação implica que o espaço gerado pela camada densa é composto por uma soma de subespaços ortogonais, cada um deles relacionado com uma família.

Para identificação das famílias (*medium-level prediction*), cada partição gerada é processada por uma camada densa com uma unidade de ativação sigmoide. Uma concatenação das saídas dessas camadas representa a classificação das famílias, sendo que o complemento do maior resultado dessa concatenação representa a classificação de “outros” padrões (nenhuma das taxonomias de interesse). Para classificação das espécies (*fine-level prediction*) o processamento é similar, modificando apenas a quantidade de saídas das camadas densas que processam os subespaços particionados, representando a quantidade de espécies em cada família.

O treinamento dos três modelos apresentados foi executado com otimizador Adam,

<sup>6</sup> <<https://github.com/BirdVox/birdvoxdetect>>

com taxa de aprendizagem  $10^{-4}$ . A avaliação dos resultados foi feita com as medidas *micro-averaged accuracy* e *macro-averaged accuracy*. Para comparação dos resultados, o *baseline* foi treinado para classificar em separado um dos três níveis de classificação (*coarse*, *medium* e *fine*). Além de avaliar os modelos descritos, Cramer *et al.* (2020) também testaram variações das estruturas hierárquicas, por exemplo, trocando as funções de ativação da TaxoNet para *tanh* ou adicionando pesos para os resultados que não são das taxonomias de interesse (complementos das concatenações de camadas densas).

Para treinamento, os autores geraram um *benchmark* com duas bases de dados, uma para treinamento/validação *American Northeast Avian Flight Call Classification* (ANAFCC) e outra para teste BirdVox-14SD, ambas disponíveis na internet<sup>7</sup>. Os autores usaram uma taxonomia com 14 espécies de pássaros migratórios que pertencem à ordem *Passeriformes*, divididos em 4 famílias: *Passaridae* (4 espécies), *Cardinalidae* (1 espécie), *Turdidae* (2 espécies) e *Parulidae* (7 espécies).

Também foram empregadas técnicas de aumento de dados, como modificação de faixas de frequência, deslocamento do sinal no tempo e adição de ruído por meio de McFee, Humphrey e Bello (2015). Todos os arquivos foram reamostrados para 22,050 Hz e PCENs foram gerados com tamanho da janela 256, *hop size* 32, 128 bandas mel e mesmos parâmetros descritos por (LOSTANLEN *et al.*, 2018a), como  $\epsilon = 10^{-6}$ ,  $\alpha = 0,8$ ,  $\delta = 10$ ,  $r = 0,25$  e  $T_{PCEN} = 60ms$ .

A partir dos resultados analisados, Cramer *et al.* (2020) destacam a capacidade superior de predição da TaxoNet, quando comparada ao *baseline* e à versão de classificação não hierárquica. Por exemplo, se apenas a classificação de espécies for considerada, a TaxoNet obteve *micro* e *macro-averaged accuracy* (66,33% e 55,69%) superiores ao *baseline* (61,13% e 54,80%). Para classificação das famílias, a TaxoNet gerou *micro-average* de 76,50% e o *baseline* 73,80%. Por fim, para detecção (presença/ausência) da ordem *Passeriformes*, a TaxoNet obteve acurácia de 94,69%, contra 77,72% do *baseline*. Assim sendo, os autores reportam que a TaxoNet é capaz de reconhecer os níveis de taxonomia de interesse, com resultados superiores aos demais modelos testados. Mesmo assim, é necessário melhorar os resultados frente ao desbalanceamento dos dados, além de melhorar a consistência das probabilidades retornadas pela rede.

A última pesquisa desta seção é a mais abrangente, porque Kahl *et al.* (2021) desenvolveram a BirdNET, a fim de classificar 984 espécies de pássaros da América do Norte e Europa, com uma *mean average precision* (mAP) de 0,791. O modelo também alcançou F0.5-score de 0,414 para anotação de Paisagens Acústicas e correlação média de 0,251 com observações tradicionais feitas em áudios gravados durante 4 anos, que possuem 121 espécies. Aplicações fundamentadas nesse modelo estão disponíveis na internet<sup>8</sup>.

<sup>7</sup> <<https://wp.nyu.edu/birdvox/codedata/>>

<sup>8</sup> <<https://github.com/kahst/BirdNET-Analyzer>>

O modelo é uma ResNet inspirada na proposta de Zagoruyko e Komodakis (2016) e com as modificações propostas por He *et al.* (2019) e Schlüter (2018). Com isso, a arquitetura contém 157 camadas, sendo que 36 possuem pesos a serem aprendidos, totalizando em torno de 27 milhões de parâmetros, com capacidade de classificar 987 classes diferentes. A maior parte das camadas de convolução possuem filtros  $3 \times 3$  e usam *padding*, são seguidas por camadas de *batch normalization* e ativação ReLU.

O treinamento dessa estrutura usou otimizador Adam, com taxa de aprendizagem  $10^{-3}$ , decaimento de 0,5, tamanho de *batch* 32, *dropout* iniciando em 0,5, com redução de 0,1 a cada passo. Além disso, os autores usaram *early stopping*, com parada em 3 épocas se as medidas de avaliação não forem alteradas e a inicialização dos pesos foi realizada com valores pré-treinados<sup>9</sup>. O treinamento inicia com no máximo 500 amostras por classe, sendo que essa quantidade é incrementada com 1000 amostras a cada passo até o final do treinamento.

Kahl *et al.* (2021) empregaram para avaliação do modelo o mAP, *class-wise* mAP (cmAP), o F0.5-*score* e a área sob a curva ROC. Os autores também usaram a correlação de Pearson para comparar os resultados do modelo com os de observadores tradicionais.

Os dados empregados para treinamento e teste são provenientes de 3 bases, cujos arquivos possuem predominância de um som/espécie específica de pássaro. Primeiro, os autores fizeram uma varredura no eBird<sup>10</sup>, extraíndo 595 espécies dos EUA e 555 da Europa, um total inicial de 1049 espécies, sendo que algumas aparecem em ambos os continentes (ex.: *Passer domesticus*). Gravações dessas mesmas espécies foram coletadas no site Xeno-canto<sup>11</sup> e no Macaulay Library of Natural Sounds<sup>12</sup>. No final, foram considerados no máximo 500 arquivos por espécie, o que gerou um total de 226.078 arquivos de áudio, sendo que os autores eliminaram espécies que apareciam em menos de 10 arquivos, permanecendo 984 espécies.

Além das espécies anteriores, Kahl *et al.* (2021) incluíram na base sons de outras fontes coletados de 16 classes do AudioSet do Google<sup>13</sup>, do FreeField1010 e Warblr, ambas do desafio *Detection and Classification of Acoustic Scenes and Events* (DCASE)<sup>14</sup> sem sons de pássaros, e também do Macaulay, desconsiderando sons de pássaros, totalizando: outros animais ( $\approx 400$  arquivos), humanos ( $\approx 7800$  arquivos), ruído ambiental ( $\approx 10.500$  arquivos). Logo, o treinamento e avaliação utilizou 987 classes, com 3978 horas de gravação divididas em treinamento (80%), validação (10%) e teste (10%).

Os testes foram executados tanto nos 10% acima citados quanto em áudios que

<sup>9</sup> O texto não informa como o pré-treinamento foi realizado

<sup>10</sup> <<https://ebird.org>>

<sup>11</sup> <<https://www.xeno-canto.org/>>

<sup>12</sup> <<https://www.macaulaylibrary.org/>>

<sup>13</sup> <<https://research.google.com/audioset/>>

<sup>14</sup> <<http://dcase.community/>>

possuem sons diversos, coletados no Sapsucker Woods Sanctuary, Ithaca, NY, EUA. São duas bases, sendo a primeira com 286 h de 84 espécies, coletadas entre maio e julho de 2017, divididos em arquivos de 5 segundos (KAHL *et al.*, 2019) e a segunda base com 134.683 arquivos de 15 minutos coletados entre 2016 e 2019. Esta última base foi complementada com observações de *hotspot* da mesma área, disponíveis no eBird, para comparar a correlação entre detecção automática e observação tradicional.

Kahl *et al.* (2021) separaram os arquivos em partes de 3 segundos e reamostraram para 48 kHz, quando possível, com resolução de 16 bits. Foram gerados mel-espectrogramas de 64 bandas, com janela<sup>15</sup> de tamanho 512 e sobreposição de 25%. A faixa de frequências dos espectrogramas foi fixada entre 150 Hz e 15 kHz, para focar nas frequências de pássaros. Além do mais, os autores aplicaram técnicas de aumento de dados, como deslocamento e alongamento de tempo e frequência, além de adição de ruído proveniente de amostras ignoradas durante o pré-processamento. Todas essas técnicas foram executadas com probabilidade de 0,5 e um máximo de 3 aumentos por amostra.

Além de resultados apresentados no início dessa explanação, os autores pontuaram sobre a capacidade da BirdNET de replicar os padrões de observações manuais, que o tamanho do repertório de vocalizações de cada espécie não é relevante quando a base de treinamento é apropriada (qualidade das gravações e quantidade de amostras), além de salientar algumas dificuldades, como os pássaros que imitam sons de outras espécies e a redução dos resultados do modelo quando a relação SNR é baixa.

## 3.2 Classificação de sons de anuros

Esta seção apresenta trabalhos que usaram redes neurais para detectar e identificar espécies de anuros, além de avaliarem diferentes arquiteturas, Transferência de Aprendizado e combinações espectrais como entrada dos modelos. No primeiro deles, Strout *et al.* (2017) treinaram modelos de CNN para classificação de chamados de 15 espécies de anuros, conseguindo acurácia próxima de 77%. Os autores avaliaram a aplicação de, pelo menos, quatro arquiteturas diferentes, além de testes com Transferência de Aprendizado para encontrar as melhores combinações.

Os autores destacam o uso de duas abordagens. Na primeira uma rede MatConvNet<sup>16</sup> foi treinada para classificação. Na segunda abordagem, três arquiteturas foram usadas: R-CNN (GIRSHICK *et al.*, 2014), AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) e CaffeNet (JIA *et al.*, 2014), todas pré-treinadas na base ImageNet (DENG *et al.*, 2009). Esses modelos foram usados como geradores de características, sendo que os vetores dessas características foram extraídos de diferentes camadas densas dos modelos

<sup>15</sup> O texto não informa qual tipo de janela.

<sup>16</sup> <<https://www.vlfeat.org/matconvnet/>>



para servir de entrada para um classificador SVM.

O processo de treinamento foi executado com validação cruzada, com 10 partições, e os valores de média, mediana, mínimo e máximo de acurácia foram analisados. [Strout et al. \(2017\)](#) também analisaram as curvas de aprendizagem e as matrizes de confusão dos modelos.

Para treinamento e teste, os autores usaram uma base de dados com 212 chamados de 15 espécies de anuros, encontradas na Flórida, EUA. Por se tratar de uma base pequena e desbalanceada, eles empregaram Transferência de Aprendizado para melhorar os resultados de classificação. Cada chamado foi representado por um espectrograma<sup>17</sup> de dimensões  $140 \times 200$ .

Na primeira abordagem, a MatConvNet alcançou resultados abaixo de 62%. Enquanto isso, o SVM com características dos modelos pré-treinados alcançou acurácia média de  $\approx 77\%$ , quando as características foram extraídas da sétima camada densa da CaffeNet.

No segundo exemplo desta seção, para detecção da presença de uma espécie, [Xie et al. \(2022\)](#) desenvolveram representações espectrais *multi-view* e um modelo que as usa como entrada para detecção da espécie de sapos *Kroombit tinker frog*, obtendo F1-score de  $99,5 \pm 0,2$  e  $96,4 \pm 2,0$  em duas bases gravadas em períodos distintos.

O modelo, nomeado como CNN-GAP, é uma rede com 3 camadas convolucionais com filtros  $3 \times 3$  e ativação ReLU (32, 64 e 128 filtros cada camada). Todas as convoluções são seguidas por *maxpooling* de  $4 \times 2$  e *dropout* com fator 0,5. Na sequência, os autores usam uma camada *global average pooling* (GAP) e duas camadas densas, sendo que a primeira possui 1024 unidades e *droupout* com fator 0,2 e a segunda, para classificação, com 2 unidades de ativação *softmax*.

Como função de perda, [Xie et al. \(2022\)](#) usaram *binary cross-entropy* ( $L_c$ ), *focal loss* ( $L_f$ ) ([LIN et al., 2017](#)), além de uma *twin loss* que combina as duas anteriores da seguinte maneira:  $L_t = \gamma L_f + (1 - \gamma L_c)$ , sendo  $\gamma$  um peso que controla a importância dos termos.

Os autores criaram 4 tipos de *multi-view* de 3 dimensões, gerados a partir de variações do mel-espectrograma, com a finalidade de prover maior capacidade de descrição dos padrões sonoros. O mel-espectrograma foi configurado com 120 bandas, janela de 40 ms e sobreposição de 20 ms (50%). Os espectrogramas *multi-view* são criados da seguinte maneira: o *harmonic percussive source separation* (HPSS) ([DRIEDGER; MÜLLER; DISCH, 2014](#)) é uma decomposição  $S = H + P$ , sendo  $S$  o mel-espectrograma,  $H$  uma representação dos harmônicos do sinal e  $P$  uma representação de variações percussivas do sinal. A partir dessa decomposição, a representação é um tensor de 3 dimensões composto por  $\{S, H, P\}$ ; a representação *delta-based* é obtida a partir do filtro Savitzky-Golay ([PRESS;](#)

<sup>17</sup> Informações sobre a geração dos espectrogramas não estão no artigo.

TEUKOLSKY, 1990), sendo que seus resultados de primeira e segunda ordem são combinados com o mel-espectrograma original, formando uma representação de 3 dimensões como a anterior; a representação *filtered-based* é criada a partir de versões  $S_{f1}$  e  $S_{f2}$  do mel-espectrograma, geradas a partir de configurações distintas de um *neighborhood filtering* (BUADES; COLL; MOREL, 2005). Assim, a entrada da rede é  $\{S, S_{f1}, S_{f2}\}$ ; no último tipo, os autores sobrepõem cópias do mel-espectrograma para gerar uma entrada com três dimensões.

Xie *et al.* (2022) consideraram para avaliação dos resultados a acurácia, o F1-score e a curva precisão-sensibilidade, além de treinarem com validação cruzada com 5 partições para treinamento e validação, além do subconjunto de testes. Para reduzir o viés dos modelos e a influência de ruídos, os autores executaram treinamento-validação e teste em dados coletados em dias de anos diferentes, sendo que os modelos são treinados com áudios de um ano e testados nos arquivos do outro ano.

Para comparação, os autores usaram um modelo CNN-FC, que é uma variação da CNN-GAP com uma camada densa no lugar da camada GAP. Um outro modelo fundamentado na VGG (SIMONYAN; ZISSERMAN, 2014) também foi aplicado. Neste caso, são 3 blocos de convolução, sendo que cada um contém duas camadas convolucionais com filtros  $3 \times 3$  e *stride* = 2, cada uma seguida por *batch normalization* e ativação ReLU. Cada bloco é encerrado por um *maxpooling*  $2 \times 2$  e *dropout* com fator 0,2. As convoluções de cada um dos 3 blocos possuem, nessa ordem, 32, 63 e 128 filtros. Depois desses blocos, também existe uma camada *global average pooling* e duas camadas densas. A primeira tem 512 unidades e *drouput* 0,2 e a segunda é composta por 2 unidades com ativação softmax.

No treinamento de todos os modelos foi utilizado otimizador Adam, com taxa de aprendizagem  $10^{-3}$ , *batch* de tamanho 32, 200 épocas e *early stopping* para evitar *overfitting*.

Para treinamento e teste, Xie *et al.* (2022) usaram 2 arquivos de 24 horas de duração, amostrados a 16 kHz, gravados no Parque Nacional Kroombit Tops, em Queensland, Austrália, sendo um deles gravado em 12 de dezembro de 2016 e o outro em 13 de dezembro de 2017. De cada arquivo foram selecionadas 2 horas, segmentadas com uma janela deslizante de 6 segundos, sem sobreposição, gerando 1200 segmentos. Esses segmentos foram anotados por especialistas quanto a presença/ausência de vocalizações de *Kroombit tinker frog*. Por fim, cada segmento foi reamostrado para 4 kHz e a faixa de frequência entre 2 e 3,5 kHz foi selecionada por ser onde as vocalizações da espécie aparecem com maior regularidade.

Sobre os resultados alcançados, os autores destacam que a seleção de frequências melhora o F1-score, gerando resultados acima de 90 enquanto que sem essa seleção o F1-score não passa de 85. Usando essa seleção e adicionando os espectrogramas *multi-view*,



os valores de *F1-score* são superiores quando comparados com representação *singl-view*, sendo as representações fundamentadas no HPSS ou na replicação do mel-espectrograma as que apresentam resultados mais consideráveis, independente dos dados usados para treinamento (2016 ou 2017). Sobre as funções de custo, os resultados variam de acordo com os dados de treinamento, sendo que a *binary cross-entropy* obteve melhores resultados quando a rede é treinada nos dados de 2016 (*F1-score* acima de 99,6) e a *twin loss* é superior quando o treinamento é executado nos áudios de 2017 (*F1-score* próximo a 97).

Ao comparar os resultados da CNN-GAP que usa seleção de frequências, *multi-view* baseada na HPSS e *twin loss*, com os modelos de comparação CNN-FC e VGG, o *F1-score* possui média e desvio similares a esses modelos para treinamento com dados de 2016 ( $99,5 \pm 0,2$ ) e resultados até 3 pontos maiores do que eles com dados de 2017 ( $96,4 \pm 2,0$ ). Xie *et al.* (2022) ainda destacam que esses resultados são alcançados com um modelo com menos parâmetros do que os demais testados (menos de 100 mil).

### 3.3 Classificação de sons de pássaros e de anuros

Durante este capítulo, este é o único artigo que trabalha simultaneamente com sons de pássaros e de anuros. LeBien *et al.* (2020) apresentaram um procedimento para treinamento de CNNs, a fim de classificar 24 espécies de pássaros e sapos de Porto Rico. O processo consiste na coleta semiautomática de dados rotulados, na utilização de Transferência de Aprendizado e na implementação de uma função de custo, que juntas geraram mAP de 0,893 e um total de AP de 0,975, em um cenário com múltiplas classes a serem detectadas e múltiplos rótulos a serem atribuídos a cada arquivo de áudio. Códigos para treinamento e testes da arquitetura estão disponíveis na internet<sup>18</sup>.

Para isso, os autores usaram uma ResNet-50 pré-treinada na base ImageNet, substituindo suas duas últimas camadas, responsáveis pela classificação, por uma camada de *average pooling* e duas camadas densas (a primeira com 512 unidades de ativação ReLU e a segunda com 24 unidades de ativação sigmoide) intercaladas por uma camada de *dropout*, com fator 0,5.

O treinamento do modelo foi realizado com otimizador Adam, com taxa de aprendizagem  $10^{-4}$  e decaimento  $10^{-7}$ . A função de custo proposta busca reduzir a necessidade de múltiplos rótulos por amostra nos dados de treinamento, através do uso de *weak labels*<sup>19</sup>. Além disso, a avaliação dos resultados de aplicação do modelo foi feita com a precisão, sensibilidade, a curva entre essas duas medidas, com a taxa de falso positivo e com a mAP.

Para coletar os dados, LeBien *et al.* (2020) primeiro identificaram 24 espécies de

<sup>18</sup> <<https://github.com/Sieve-Analytics/arbimon2-cnn>>

<sup>19</sup> Conjunto de rótulos limitados, com ruído ou impreciso.

pássaros e anuros que possuem necessidade de conservação, em áreas de Porto Rico. A partir disso, foram escolhidos pelo menos um ou dois tipos de chamados de cada espécie para serem usados como modelos de espectrogramas para busca na plataforma ARBIMON<sup>20</sup>. Nessa mesma ferramenta, os resultados das consultas foram validados, o que diminuiu o esforço de criação da base de áudios. Para o treinamento, os arquivos de áudio recuperados foram divididos em segmentos de 2 segundos dos quais foram gerados mel-espectrogramas com janela de Hanning de tamanho 1024, com 50% de sobreposição e 124 bandas mel.

Enquanto isso, para predição, um subconjunto de 1000 arquivos (todos da Floresta Nacional El Yunque, Porto Rico) aleatórios de 1 minuto foi selecionado da base. Nesses arquivos, um janelamento foi aplicado criando segmentos de 2 segundos, com 1 segundo de sobreposição entre as janelas. O modelo foi testado nos mel-espectrogramas desses segmentos e os rótulos atribuídos foram retornados como a predição do arquivo.

Além de resultados de mAP de 0,893, segundo as considerações feitas por [LeBien et al. \(2020\)](#), os arquivos testados possuem em média entre 3 e 8 espécies detectadas. Os autores também destacam a necessidade de inclusão de variações do tamanho dos padrões usados para criar a base de treinamento, uso de aumento de dados e melhoria da eficácia da predição sem gerar perdas de acurácia.

### 3.4 Classificação de sons em ambiente marinho

Para comparação com as seções anteriores, esta seção apresenta exemplos de pesquisas que empregaram redes neurais para detectar e identificar espécies de baleias. Com isso, é possível verificar se existem abordagens similares aplicadas em gravações terrestres e subaquáticas. No primeiro desses artigos, [Harvey \(2018\)](#) usou CNNs para detecção e recuperação de áudios de baleias jubarte, em 15 anos de gravações submarinas de locais do Oceano Pacífico. Para detecção, os modelos alcançaram 90% de precisão e de sensibilidade, além de auxiliarem na geração de informações sobre a presença, sazonalidade, comportamento diário dos sons das jubartes e informações sobre a estrutura da sua população.

Foram empregadas duas abordagens de aprendizado, uma supervisionada com o treinamento de uma ResNet-50 para detecção da presença do canto de jubartes e outra não supervisionada, que usa a mesma arquitetura, mas no lugar de classificar, aprende uma função que induz vetores de características de amostras de áudio, próximas no tempo, a terem distância euclidiana similar. Para isso, os autores aplicaram a função *triplet* proposta por [Jansen et al. \(2018\)](#), que aproxima características de amostras próximas no tempo, porque é grande a possibilidade delas possuírem conteúdo similar, gerando espaços

---

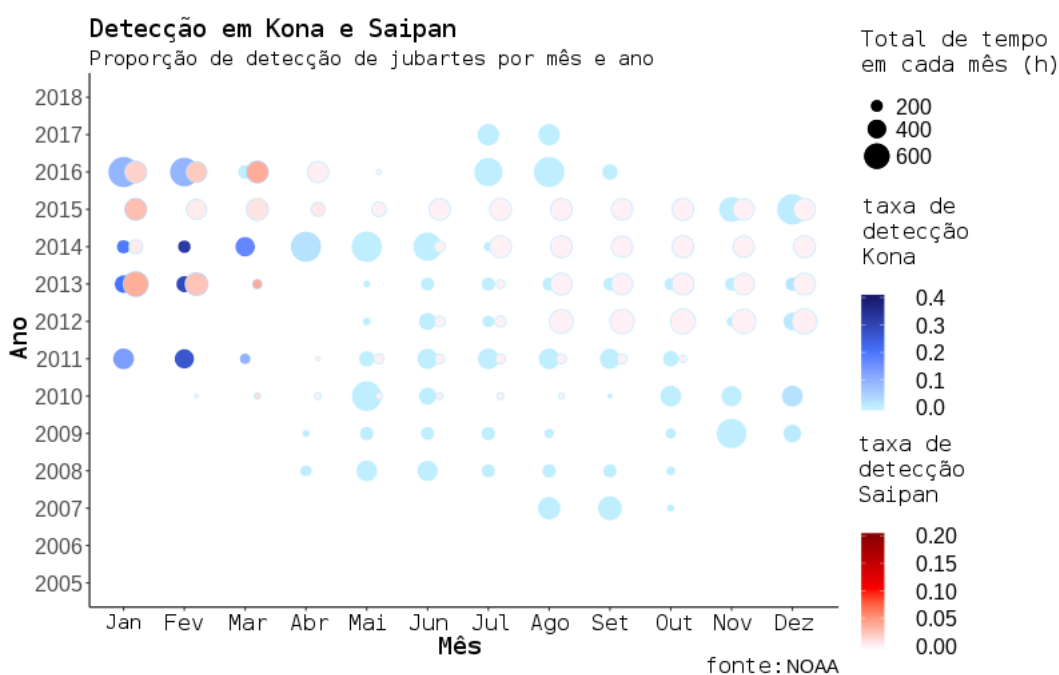
<sup>20</sup> <<https://arbimon.rfcx.org/>>

de características que representam essa similaridade. Com isso, a rede pode ser empregada como extrator de características usado para recuperação de padrões sonoros.

Os dados usados foram gravados com microfones submarinos (hidrofone) pelo *U.S. National Oceanic and Atmospheric Administration* (NOAA) e totalizam  $\approx 9,2$  terabytes de gravações. Na abordagem supervisionada, [Harvey \(2018\)](#) extraiu 0,2% desses dados, rotulados manualmente. Como entrada para os modelos, os autores utilizaram o PCEN a fim de reduzir a influência de ruído estacionário, diminuindo o erro da classificação em até 24%.

Para a detecção, o modelo alcançou 90% de precisão e de sensibilidade em áudios de teste de 75 segundos. Para visualizar isso, a [Figura 11](#) apresenta um resumo de duas áreas de gravação. Nela, é possível verificar a proporção de amostras de tempo onde jubartes foram detectadas, relacionada com o total de gravações no mês. A maior taxa de detecção nos meses iniciais dos anos de coleta é consistente com o período que as baleias migram do Alasca para as proximidades do Havaí para procriar.

Figura 11 – Sumarização visual da classificação de baleias jubarte em duas regiões do Havaí (Kona e Saipan). Os círculos indicam a proporção de horas coletadas por mês e as cores a taxa de detecção de sons de jubarte nesses áudios.

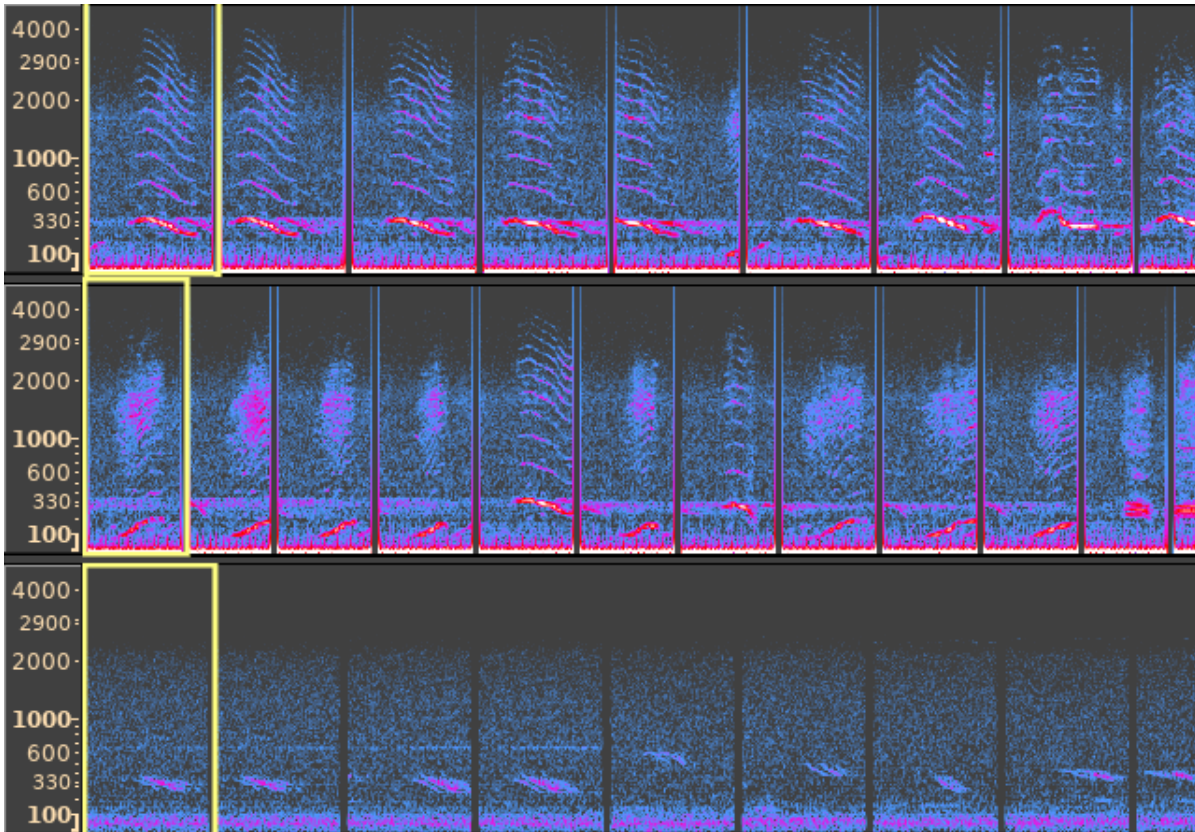


Fonte: Adaptada de [Harvey \(2018\)](#).

Para avaliar a qualidade das características geradas pelo método não supervisionado, [Harvey \(2018\)](#) selecionaram algumas amostras da base e tentaram recuperar áudios similares a elas, considerando a distância entre os vetores de características extraídos pela ResNet treinada com a função *triplet*. A [Figura 12](#) destaca os espectrogramas selecionados e algumas amostras retornadas que são similares a eles. Segundo os autores, em alguns

casos de teste foi viável recuperar centenas de gravações similares com significativa taxa de precisão.

Figura 12 – Resultados de consultas na base de áudios a partir das características geradas. Espectrogramas com moldura amarela são chaves para as consultas e os demais são áudios similares a eles.



Fonte: Harvey (2018).

No segundo exemplo com sons de baleias, [Thomas et al. \(2019\)](#) criaram um sistema de classificação capaz de diferenciar vocalizações de quatro espécies de baleias, ruídos do ambiente e sons não biológicos. Para isso, foram empregadas arquiteturas de CNNs e sobretudo, uma nova representação espectral para os padrões. Dessa maneira, o sistema alcança acurácias acima de 95%.

Os autores treinaram uma ResNet-50 e uma VGG-19 aplicando o SGD como otimizador, sendo que a taxa de aprendizagem para a primeira arquitetura é de  $10^{-3}$ , para a outra  $10^{-2}$ , mas, nos dois casos foi aplicado decaimento exponencial a um fator 10 a cada 30 épocas. O otimizador foi configurado com *momentum* = 0,9, decaimento de pesos de  $10^{-4}$ , treinamento em 100 épocas e *batch* de tamanho 128.

O processo de treinamento foi executado 10 vezes usando sementes aleatórias distintas e para avaliação foram usadas as médias da acurácia, da precisão, da sensibilidade e do F1-score. Durante o treinamento, os melhores modelos são salvos a partir da avaliação do F1-score no subconjunto de validação.

Os dados considerados por [Thomas et al. \(2019\)](#) foram coletados na costa atlântica do Canadá, durante os meses de verão e outono dos anos 2015 e 2016. O foco do desenvolvimento esteve em gravações com taxa de amostragem de 8 kHz, para capturar vocalizações inferiores a 1 kHz, como de baleia-azul (*Balaenoptera musculus*), baleia-fin (*Balaenoptera physalus*) e baleia-sei (*Balaenoptera borealis*), sons não biológicos e ruídos ambientais, além da baleia-jubarte (*Megaptera novaeangliae*). No total, são 61.884 amostras de 10 segundos, anotadas por biólogos, divididas em treinamento, validação e teste, com proporção 70:15:15, sendo que existe predominância de vocalizações de baleia-fin, com proporção 6:1. Os dados de jubartes foram adicionados em uma segunda fase de testes, contabilizando 2100 amostras de treinamento e 450 para teste.

Esses arquivos foram representados por espectrogramas criados com janela de Hanning, com 1/4 de sobreposição. Devido à grande quantidade de combinações dos parâmetros da STFT, em especial o tamanho das janelas, [Thomas et al. \(2019\)](#) apresentaram uma representação que combina vários espectrogramas, com diferentes tamanhos de janela, a partir de uma simples interpolação linear, como visto no [Algoritmo 1](#), onde a função de interpolação é definida como

$$\omega = \omega_i + \frac{\omega_{i+1} - \omega_i}{n_{i+1} - n_i}(n - n_i), \quad (3.1)$$

sendo que o  $n$  é conhecido e um ponto  $(n, \omega)$  é interpolado entre os pontos  $(n_i, \omega_i)$  e  $(n_{i+1}, \omega_{i+1})$  de espectrogramas distintos.

---

**Algoritmo 1** – Criação de uma instância da nova representação ([THOMAS et al., 2019](#))

---

- 1: **Entrada** O sinal  $x$ , uma janela  $w$  e os parâmetros de espectrograma  $\Theta = [\theta_1, \theta_2, \dots, \theta_k]$
  - 2: Inicializar resoluções  $\omega_0$  e  $n_0$  da interpolação com  $\infty$
  - 3: **para**  $i = 1$  **até**  $k$  **faça**
  - 4:     Gerar o espectrograma  $D_i = STFT(x; w, \theta_i)$  ([Equação 2.3](#))
  - 5:     Para a execução, mantém valores mínimos de  $\omega_0$  e  $n_i$
  - 6:     **se**  $\Delta\omega_i < \omega_0$  **então**
  - 7:          $\omega_0 = \Delta\omega_i$
  - 8:     **fim se**
  - 9:     **se**  $\Delta n_i < n_0$  **então**
  - 10:          $n_0 = \Delta n_i$
  - 11:     **fim se**
  - 12: **fim para**
  - 13: **para**  $i = 1$  **até**  $k$  **faça**
  - 14:     Interpola cada espectrograma  $S_i = INTERPOLATE(D_i; \omega_0, n_0)$  ([Equação 3.1](#))
  - 15: **fim para**
  - 16: Empilhar os espectrogramas criados  $Z = [S_1, S_2, \dots, S_k]$
  - 17: **Saída** Um tensor  $Z$  com  $k$  canais
- 

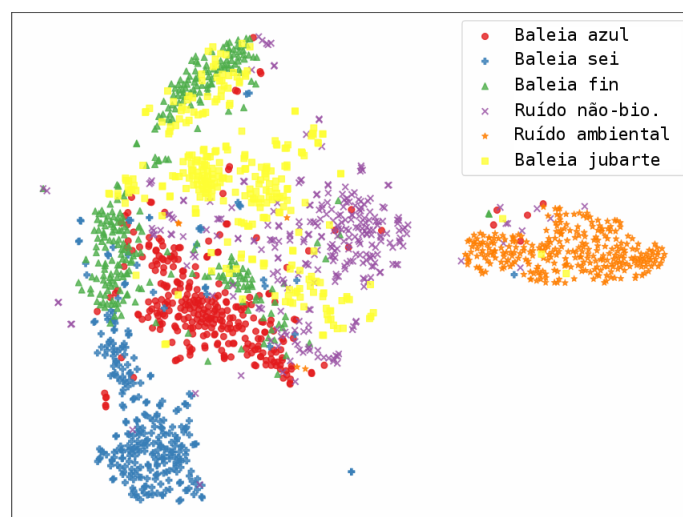
Com isso, foram treinados cinco classificadores para cada arquitetura, sendo que 3 deles possuem como entrada espectrogramas, com janelas de tamanho 256, 2048 e 16384,



respectivamente; outro com mel-espectrograma, com tamanho de janela 2048 e 128 bandas mel; e o último com a entrada proposta. Os tamanhos de janelas foram escolhidos de maneira a capturar desde curtas a longas vocalizações. Todas as representações foram truncadas para o intervalo de frequências entre 10 Hz e 1000 Hz, que representam os padrões sonoros de interesse.

Nos primeiros testes, a base de dados não continha os padrões de baleia-jubarte. Nesse cenário, a ResNet obteve  $\approx 95\%$  de acurácia e a VGG  $\approx 96\%$  com a nova representação, sendo significativamente maiores do que as representações convencionais. Isso só não acontece no teste da VGG com espectrograma gerado com janela de tamanho 2048, onde os resultados são similares aos da representação proposta. Em um segundo teste, [Thomas et al. \(2019\)](#) avaliaram a capacidade de seus detectores em generalizar para outros padrões de até 1000 Hz, aplicando Transferência de Aprendizado. Nesse caso, eles usaram o melhor modelo de VGG, treinado com a nova representação, congelaram os pesos das camadas convolucionais, adicionaram as amostras contendo padrões de baleia-jubarte nos dados de treinamento e de teste e retrainaram o modelo, atualizando apenas os pesos das camadas densas. O modelo refinado alcançou acurácia de 94,8%, e precisão e sensibilidade acima de 87%. A [Figura 13](#) apresenta o espaço de características gerado pelas camadas convolucionais, onde é possível verificar a definição das fronteiras entre as classes, mesmo que elas possuam algum nível de sobreposição.

Figura 13 – Projeção t-SNE do espaço de características gerado pela última camada convolucional após o treinamento com Transferência de Aprendizado.



Fonte: Adaptada de [Thomas et al. \(2019\)](#).

No terceiro artigo reportado nesta seção, [Kirsebom et al. \(2020\)](#) investigaram a aplicação de CNNs para detecção de sons da baleia-franca (*right whale* ou *black whale*) em arquivos gravados em regiões litorâneas do Canadá e dos Estados Unidos. O modelo gerado alcançou sensibilidade acima de 87% e precisão de 90%, demonstrando a sua capacidade

em identificar os padrões da espécie de interesse.

Para isso, os autores treinaram uma ResNet para identificar a presença de vocalizações da baleia em espectrogramas de 3 segundos. A arquitetura é composta por 8 blocos residuais precedidos por uma camada convolucional com 16 filtros de tamanho  $3 \times 3$  e inicializados com distribuição normal, usando *padding*, *stride* igual 1 e sem viés. Os blocos residuais são seguidos por 3 camadas: uma de *batch normalization*, uma de *global average pooling* e uma camada densa com ativação *softmax* <sup>21</sup>.

Kirsebom *et al.* (2020) treinaram seu modelo com otimizador Adam, com taxa de aprendizagem  $10^{-3}$ , decaimento 0,01,  $\beta_1 = 0,9$  e  $\beta_2 = 0,999$ . O processo foi executado 9 vezes com sementes diferentes para os métodos aleatórios, *batch* de tamanho 128 e 100 épocas. Na avaliação do modelo, foram empregadas as medidas *F1-score*, precisão e sensibilidade.

Para comparação, os autores converteram as matrizes de espectrogramas em vetores, aplicaram PCA para reduzir suas dimensões e usaram os resultados como entrada para uma *Linear Discriminant Analysis* (LDA). Para identificar qual a melhor quantidade de componentes principais que o PCA deveria considerar, os dados de treinamento foram divididos na proporção 85:15, para treinamento/validação, várias configurações do PCA foram testadas e as configurações que geraram o melhor resultado do LDA nos dados de validação foram utilizadas<sup>22</sup>.

Kirsebom *et al.* (2020) aplicaram uma heurística para remoção de ruídos nas gravações, que consiste em reduzir o ruído de impulso através da subtração de cada fatia de tempo (*time slice*) por sua respectiva mediana e reduzir o ruído tonal por meio da subtração das bandas de frequência também por suas medianas.

As gravações usadas para treinamento e teste são de 3 bases coletadas entre 2015 e 2019, em 6 estações ao sul do Golfo de S. Lourenço, no Canadá, e uma base coletada em 2009, no Golfo do Maine, EUA. As amostras foram rotuladas com presença e ausência de baleia-franca, por meio da aplicação do algoritmo *time-frequency based detector* (MEL-LINGER, 2004; MOUY; BAHOURA; SIMARD, 2009) e da validação de especialistas. Todos os arquivos foram reamostrados para 1 kHz e os espectrogramas foram gerados com janela de Hamming, com tamanho 0,256 segundos (256 amostras) e sobreposição de 88%, gerando imagens de dimensões  $94 \times 129$ .

Nesse cenário, os melhores modelos alcançaram resultados com sensibilidade acima de 87% e precisão acima de 90%, com significativa superioridade em relação ao LDA.

No último artigo desta seção, Shiu *et al.* (2020) também aplicaram técnicas de *Deep Learning* para detecção de vocalizações de baleia-franca, mas neste caso os autores

<sup>21</sup> Configurações encontradas no código do material suplementar do artigo

<sup>22</sup> A informação de qual a melhor dimensionalidade não aparece no artigo

compararam os resultados de várias arquiteturas com os resultados obtidos em competições de Aprendizagem de Máquina que usam os mesmos dados. As arquiteturas testadas chegaram a uma sensibilidade de 0,946 contra 0,492 de classificadores que usam características acústicas.

Os autores testaram cinco arquiteturas: uma LeNet (LECUN *et al.*, 1998), substituindo a ativação *tanh* por ReLU, as camadas de *average pooling* por camadas de *max-pooling* e adicionando *dropout* depois da primeira camada de convolução e das camadas de *pooling*; uma VGG; uma ResNet; uma arquitetura proposta por (KAHL *et al.*, 2018) e nomeada neste artigo como BirdNET<sup>23</sup>, que usa *dropout* de 0,2 depois de todas as camadas de *pooling* e regularização  $L^2$  com fator igual a 0,2 em todas as camadas convolucionais; e por fim, uma combinação de convolução de 1 dimensão com unidades recorrentes *Gated Recurrent Unit* (GRU) (CHO *et al.*, 2014), nomeada como Conv1D+GRU.

Shiu *et al.* (2020) treinaram os modelos com otimizador Adam, com taxa de aprendizagem 0,005, *batch* de tamanho 1000 e 100 épocas. Como existem diferenças nas quantidades de amostras por classe, os autores aplicaram pesos para mitigar o desbalanceamento, sendo que a classe positiva (presença de vocalização) possui um fator igual a 3. Para avaliação do desempenho, foram usadas a precisão, a sensibilidade e a quantidade de falsos positivos por hora de gravação.

Os resultados foram comparados com os classificadores usados no *International Workshop on Detection, Classification, Localization, and Density Estimation of Marine Mammals using Passive Acoustics* (DCLDE) de 2013: *multivariate discriminant analysis, generalized likelihood ratio tests, decision trees, shallow neural networks* e *boosting classifiers*, que usam 13 características acústicas para representar os dados.

Para treinar os modelos, Shiu *et al.* (2020) usaram 3 bases de áudios. Uma do DCLDE de 2013, coletada durante sete dias, no litoral de Massachusetts, EUA, entre março e abril de 2009. Dados do NOAA, coletados entre as costas de Meryland, Georgia, Carolina do Norte e Virgínia, EUA, entre setembro de 2012 e julho de 2015. E dados de uma competição do Kaggle, coletados na baía de Massachusetts, com partes de gravações de 2 segundos. De todos os arquivos, foram gerados espectrogramas com janela de Hanning de 128 ms e deslocamento de 50 ms (*hopsize = 50ms*), normalizados pela soma dos quadrados de todos os valores das matrizes. Os dados do DCLDE foram usados como treinamento para avaliar e comparar com resultados dos modelos desse evento, além de verificar a capacidade dos modelos treinados de generalizar para dados coletados em outras regiões e períodos (NOAA e Kaggle).

Para uma melhor avaliação dos resultados, os autores executaram dois cenários: em um deles, os modelos foram treinados com os dados do DCLDE e no outro, o treinamento

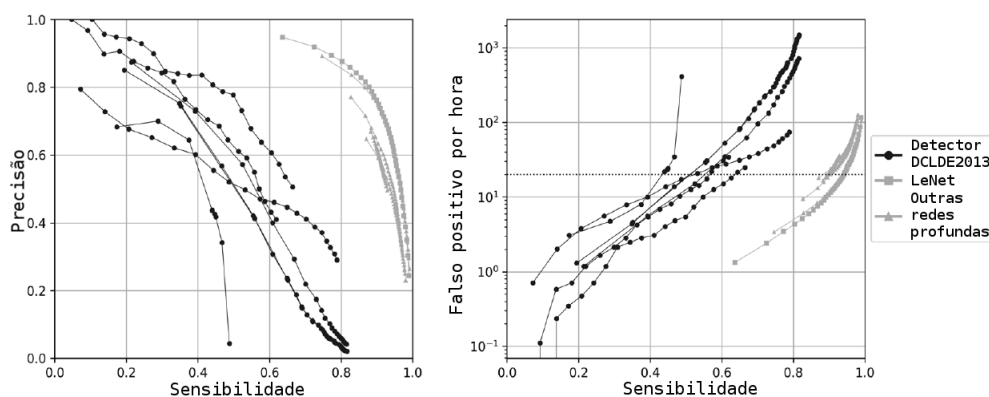
---

<sup>23</sup> Aparentemente diferente da proposta por (KAHL *et al.*, 2021)



foi executado nos dados do Kaggle. No primeiro caso, apresentado na [Figura 14](#), todas as redes alcançaram resultados de sensibilidade superiores (até 0,946) aos classificadores do DCLDE (no máximo 0,83), sendo que o LeNet e o BirdNET obtiveram os melhores resultados. No segundo caso, os modelos geraram resultados de sensibilidade de 0,883, também superiores ao *baseline*.

Figura 14 – Resultados dos classificadores do DCLDE 2013 contra as redes neurais treinadas nos mesmos dados, relacionando sensibilidade, precisão e falsos positivos por período.



Fonte: Adaptada de [Shiu et al. \(2020\)](#).

### 3.5 Considerações sobre os trabalhos apresentados

Uma análise geral sobre os aspectos dos trabalhos citados pode salientiar os resultados consideráveis da aplicação de CNN tanto para detecção da presença de animais ([LOSTANLEN et al., 2019](#); [KIRSEBOM et al., 2020](#); [XIE et al., 2022](#)) quanto para identificação de suas espécies ([KAHL et al., 2021](#); [STROUT et al., 2017](#); [THOMAS et al., 2019](#)), em ambientes terrestres e subaquáticos. Por causa desses resultados, ferramentas funcionais foram disponibilizadas com modelos pré-treinados, úteis para a comunidade de pesquisa ([CRAMER et al., 2020](#); [KAHL et al., 2021](#); [LEBIEN et al., 2020](#); [LOSTANLEN et al., 2019](#)).

É interessante perceber no [Quadro 1](#) que, mesmo lidando com sons coletados em ambientes naturais, apenas quatro desses trabalhos destacam a aplicação de filtragem como etapa de pré-processamento ([KIRSEBOM et al., 2020](#)) ou seleção de faixas de frequência ([KAHL et al., 2021](#); [THOMAS et al., 2019](#); [XIE et al., 2022](#)). Isso pode estar relacionado com as capacidades dos modelos, com as variações naturais dos padrões, com variações introduzidas por adição de dados de bases genéricas ou por técnicas de aumento de dados ([CRAMER et al., 2020](#); [KAHL et al., 2021](#); [LOSTANLEN et al., 2019](#); [SALAMON et al., 2017](#)).

Representações como o mel-espectrograma são empregadas em muitos trabalhos de classificação de sons, mesmo assim não existe certeza de que seja sempre a melhor

representação (PURWINS *et al.*, 2019). Das 11 pesquisas apresentadas, quatro usam espectrogramas, cinco utilizam mel-espectrogramas e outras três usam PCEN. Além disso, três executam algum nível de combinação de características (LOSTANLEN *et al.*, 2019; THOMAS *et al.*, 2019; XIE *et al.*, 2022). Em uma visão mais abrangente, verifica-se que parte dos trabalhos com sons lidam com alguma representação do espectro de frequências, buscando mais informação sobre os padrões. Por outro lado, modelos como a SoundNet (AYTAR; VONDRICK; TORRALBA, 2016) e a WaveNet (OORD *et al.*, 2016), não descritos aqui, utilizam o sinal do áudio para gerar representações semânticas dos sons, aplicáveis para identificação, síntese de fala e geração de música. Para que isso aconteça, durante o treinamento a SoundNet relaciona padrões de sons com seus respectivos *frames* de vídeo, enquanto que a WaveNet aplica uma sequência de transformações e estruturas residuais para gerar um modelo probabilístico e autorregressivo. Nos dois casos, os modelos geram resultados adequados, mas são mais complexos do que os reportados neste capítulo.

Sobre as arquiteturas consideradas, o Quadro 1 destaca que pelo menos 4 artigos trabalham com arquiteturas compostas por até 5 camadas (CRAMER *et al.*, 2020; LOSTANLEN *et al.*, 2019; SALAMON *et al.*, 2017; XIE *et al.*, 2022), um artigo usa arquiteturas como a AlexNet para prover características para o treinamento de um SVM (STROUT *et al.*, 2017) e a maior parte dos artigos (sete deles) usam arquiteturas profundas, como VGG e ResNet, algo semelhante ao apresentado no levantamento de Stowell (2022). Logo, arquiteturas como essas são consideradas independente do tipo de padrão (aves, anuros, baleias) ou do ambiente de coleta (terrestre, subaquático), com variações principalmente das entradas e das configurações de treinamento. Outras estratégias de aprendizado e combinação de resultados são destacadas por Salamon *et al.* (2017).

No que diz respeito à inicialização dos pesos das redes, o Quadro 1 destaca apenas 2 trabalhos que usam de maneira explícita pesos pré-treinados, executando Transferência de Aprendizado. Essa transferência é com frequência considerada quando a quantidade de dados rotulados para treinamento é reduzida, melhorando a capacidade de generalização dos modelos, sobretudo quando eles são mais profundos. É possível que os trabalhos apresentados não considerem essa abordagem por causa da quantidade de amostras e da variação dos padrões disponíveis para o treinamento. No geral, a Transferência de Aprendizado gera resultados consideráveis em aplicações com dados de diferentes domínios e também na classificação de padrões sonoros de animais (DUFOURQ *et al.*, 2022).

Por fim, essas questões referentes à representação dos sons e suas combinações, sobretudo com espectrogramas e características acústicas, à profundidade das redes, além de abordagens para melhoria do treinamento, como técnicas para aumentar a variação dos dados e para inicialização dos pesos, fazem parte dos passos de pesquisa propostos nos próximos capítulos.

Quadro 1 – Sumarização dos artigos e suas principais características, relacionadas com pré-processamento dos dados (pré): filtragem ou seleção de frequências específicas; aumento de dados (aum.); tipos de entrada dos modelos: espectrograma, mel-espectrograma, PCEN ou alguma combinação de entradas (comb.); profundidade da arquitetura: rasa ou profunda; e inicialização dos pesos: aleatória (alea.) ou pesos pré-treinados (pré-trei.)

	pré	aum.	entrada				arquitetura		inicialização	
			espec.	mel	pcen	comb.	rasa	profunda	alea.	pré-trei.
(SALAMON <i>et al.</i> , 2017)		✓		✓			✓		✓	
(LOSTANLEN <i>et al.</i> , 2019)		✓		✓	✓	✓	✓		✓	
(CRAMER <i>et al.</i> , 2020)		✓			✓		✓		✓	
(KAHL <i>et al.</i> , 2021)	✓			✓				✓	✓	
(STROUT <i>et al.</i> , 2017)			✓					✓	✓	
(XIE <i>et al.</i> , 2022)	✓			✓		✓	✓		✓	
(LEBIEN <i>et al.</i> , 2020)				✓				✓	✓	
(HARVEY, 2018)					✓			✓	✓	
(THOMAS <i>et al.</i> , 2019)	✓		✓			✓		✓	✓	
(KIRSEBOM <i>et al.</i> , 2020)	✓		✓					✓	✓	
(SHIU <i>et al.</i> , 2020)			✓					✓	✓	

Fonte: Elaborada pelo autor.



---

## LINHAS GERAIS PARA O TREINAMENTO DE REDES CONVOLUCIONAIS NA IDENTIFICAÇÃO DE VOCALIZAÇÕES DE ANIMAIS

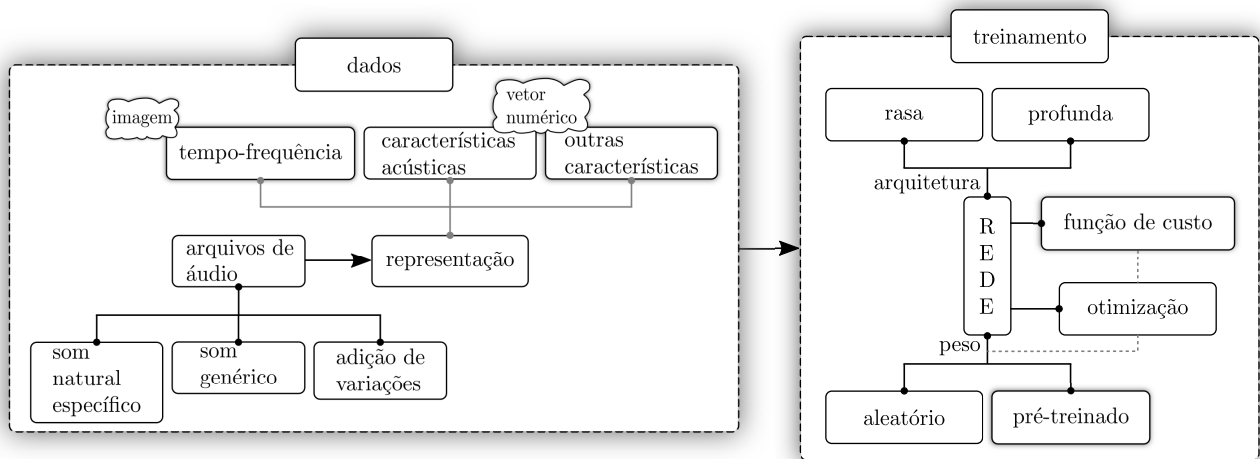
---

---

Neste capítulo são descritas as etapas da pesquisa desenvolvida neste doutorado, que, conforme os objetivos, busca aprender representações dos dados sonoros, ou seja, características, para serem empregadas na identificação de espécies de pássaros e anuros, a partir dos sons de suas vocalizações. Para isso, seguindo a [Figura 15](#), parte-se de um processo básico de análise, com dados e modelagem. A princípio, as amostras que contêm padrões de sons de interesse são adicionadas a amostras com padrões genéricos ([KAHL \*et al.\*, 2021](#)) e o conjunto resultante é aumentado ([SALAMON; BELLO, 2017](#)) para melhorar a capacidade de generalização dos modelos testados. Em seguida, desse conjunto são extraídos e combinados conjuntos diferentes de características de entrada para verificar como essas representações influenciam na robustez dos modelos, lidando com variações dos padrões de som ([LOSTANLEN \*et al.\*, 2019](#); [THOMAS \*et al.\*, 2019](#); [XIE \*et al.\*, 2022](#)). Arquiteturas com diferentes profundidades e estruturas são testadas para examinar o impacto nos resultados e como essas modificações se comportam com diferentes entradas e estratégias de treinamento. Além disso, comparações entre pesos aleatórios e aprendidos em outras tarefas também são executadas para identificar qual a melhor abordagem de inicialização ([DUFOURQ \*et al.\*, 2022](#)). Modificações na função de custo são consideradas para melhorar os espaços de características aprendidos para classificação, além da apresentação e análise de algumas modificações nas configurações dos otimizadores que aumentam a capacidade de aprendizagem das redes ([BECHER; PONTI, 2021](#)). Espera-se obter modelos que sejam capazes de identificar os padrões sonoros mesmo na presença de ruído, sobreposição de sinais e variações de volume. Devido a isso e de acordo com a ten-

dência descrita na Seção 3.5, não foram definidas etapas explícitas de pré-processamento como seleção de faixas de frequência, filtragem para atenuação de ruídos ou separação de fontes sonoras. Assim sendo, as próximas seções descrevem os passos a serem seguidos e as abordagens de avaliação dos resultados, para que sejam implementados e seus resultados analisados no Capítulo 5, Capítulo 6 e Capítulo 7.

Figura 15 – Estrutura básica dos componentes explorados na pesquisa.



Fonte: Elaborada pelo autor.

## 4.1 Arquivos de áudio

Existe uma incontável variedade de sons tanto em ambientes naturais quanto urbanos, gerados ou não por seres humanos. Esses sons possuem características distintas, o que leva ao desenvolvimento de abordagens variadas para sua manipulação e análise, o que é verificado nos trabalhos descritos no Capítulo 3 e na literatura dedicada à Paisagens Acústicas e áreas relacionadas. Também existe um empenho para construção de bases de áudios que possam ser utilizadas para exploração, análise e construção de modelos, com dados coletadas em diversos ambientes, a partir de sistemas de gravação manuais ou automáticos, como as bases Xeno-canto<sup>1</sup> e Ocean Networks<sup>2</sup>, ou coletadas em recursos, como os disponíveis no YouTube (GEMMEKE *et al.*, 2017; HERSHEY *et al.*, 2017).

### 4.1.1 Espécies de interesse

Espécies de pássaros e anuros são consideradas como bioindicadores (cf. Seção 1.1 e Seção 2.3), sendo estudadas em várias pesquisas. As vocalizações de pássaros podem ser divididas em canto (usados para demarcação, defesa de território e acasalamento) e

<sup>1</sup> <<https://www.xeno-canto.org/>>

<sup>2</sup> <<https://www.oceannetworks.ca/>>

chamado/apelo (utilizados na comunicação geral entre os animais) (WIKIAVES, 2020), enquanto que os anuros também empregam o seu coaxar para demarcação, defesa do território e para acasalamento. Esses sons apresentam variações, por exemplo, espécies de pássaros podem ter um vasto repertório de padrões de som, variando conforme a região onde a espécie é encontrada. Além de alterações nos padrões, existem espécies que imitam sons de outras espécies, possivelmente para aumentar o seu repertório (WIKIAVES, 2020), o que dificulta a sua identificação.

A Tabela 1 e a Tabela 2 listam as espécies para avaliação desta proposta além de exemplos de padrões de suas vocalizações. Essas espécies foram selecionadas por especialistas do *Spatial Ecology and Conservation Lab* da UNESP/Rio Claro, que contribuíram com dados, informações e discussões.

### 4.1.2 Balanceamento de classes e variação de padrões

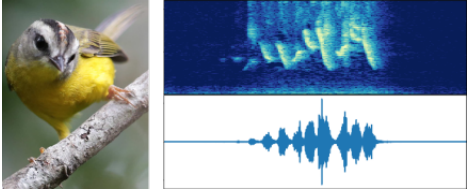
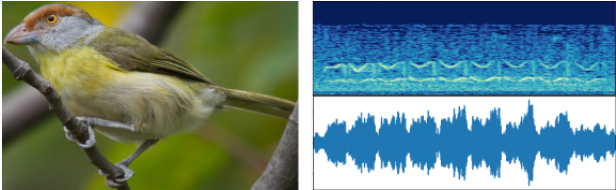
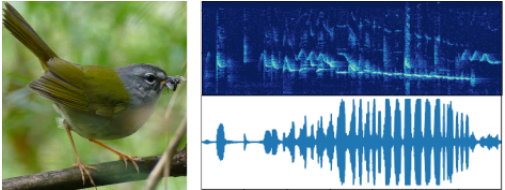
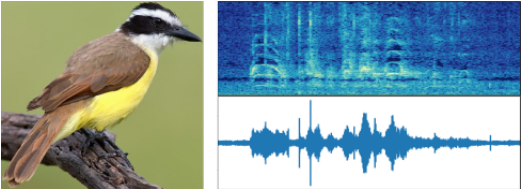
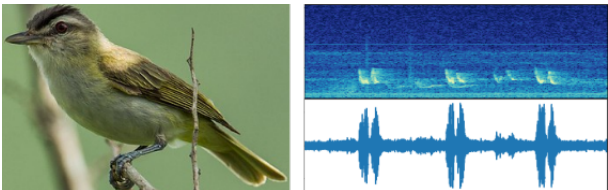
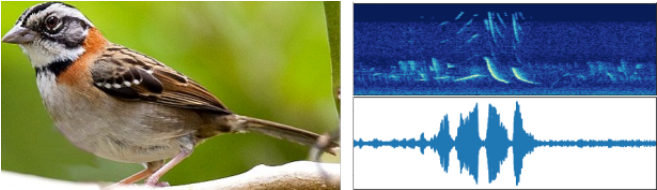
Como descrito no Capítulo 2, cenários nos quais existem poucas amostras rotuladas, pouca variabilidade dos dados ou quantidades distintas de amostras entre as classes, atrapalham a capacidade de generalização dos modelos. Por causa disso, esta pesquisa incrementa a base de áudios, criando amostras sintéticas a partir das amostras reais, por meio de técnicas de aumento de dados (SALAMON; BELLO, 2017). Com isso, mais amostras são obtidas sem a necessidade de coleta e rotulação manual, balanceando as classes de treinamento, aumentando a variação da base e criando visões diferentes dos dados. Além do mais, são adicionados sons de bases genéricas, gerados por fontes variadas, para incrementar não apenas a variabilidade dos dados mas também para que os modelos possam distinguir entre vocalizações das espécies de animais e sons diversos (KAHL *et al.*, 2021).

## 4.2 Representação dos padrões de som

Além do próprio sinal sonoro e da extração de características acústicas para representação e análise de sons, espectrogramas e suas variações também são empregadas em processos de identificação de eventos acústicos (cf. Seção 2.2.1, Seção 2.2.2 e Seção 2.3.1). Combinações de características também são usadas (LOSTANLEN *et al.*, 2019; THOMAS *et al.*, 2019; XIE *et al.*, 2022) para criar representações robustas, capazes de descrever os padrões. Como discutido na Seção 3.5, não existe consenso de qual dessas representações é mais adequada, sendo necessária a avaliação de cada caso específico, considerando a estrutura dos padrões a serem identificados, as capacidades de cada representação e suas combinações. Assim sendo, esta pesquisa investiga um conjunto de representações do espectro, para verificar quais delas melhor descrevem os padrões de interesse e suas variações, servindo como entradas para CNNs; como essas representações podem ser combinadas; e como vetores de características acústicas ao serem combinados com esses espectrogramas

impactam na acurácia das redes testadas. Por causa de variações dos padrões de vocalização dos animais, relacionadas com horário e local de sua presença, também são associados

Tabela 1 – Espécies de pássaros e exemplos de seus padrões sonoros.


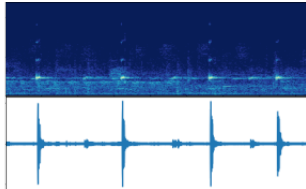

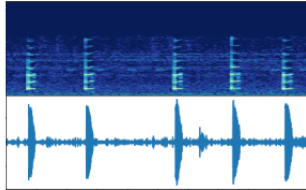

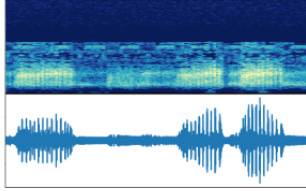

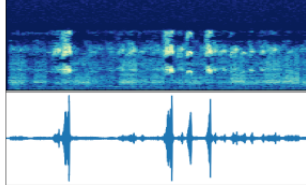

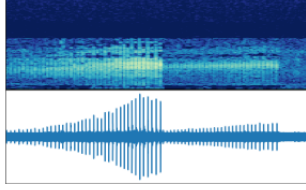
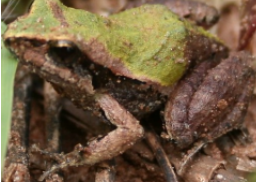
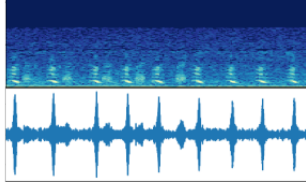
espécie	exemplo
<i>Basileuterus culicivorus</i> (pula-pula)	 <p>Foto (MAIZLISH, 2013). Som (SUEZA, 2019)</p>
<i>Cyclarhis gujanensis</i> (pitiguari)	 <p>Foto (LEMES, 2014). Som (FRAGA, 2001)</p>
<i>Myiothlypis leucoblephara</i> (pula-pula-assobiador)	 <p>Foto (FINK, 2018). Som (GODOY, 2016)</p>
<i>Pitangus sulphuratus</i> (bem-te-vi)	 <p>Foto (CHUDÝ, 2019). Som (SWACKHAMER, 2017)</p>
<i>Vireo chivi</i> (juruviara)	 <p>Foto (ATHANAS, 2013). Som (LEITE, 2020)</p>
<i>Zonotrichia capensis</i> (tico-tico)	 <p>Foto (VERONESI, 2010). Som (ANTONELLI, 2018)</p>

Fonte: Elaborada pelo autor.



às características acústicas dados referentes aos horários e aos locais de gravação.

Tabela 2 – Espécies de **anuros** e exemplos de seus padrões sonoros.

espécie	exemplo	
<i>Adenomera marmorata</i> (rãzinha-piadeira)		
	Foto (PASSOS, 2014). Som (AMPHIBIAWEB, 2020)	
<i>Aplastodiscus leucopygius</i> (perereca-flautinha)		
	Foto (PASSOS, 2020). Som (AMPHIBIAWEB, 2020)	
<i>Boana albopunctata</i> (rã-cabrinha)		
	Foto (PASSOS, 2016). Som (AMPHIBIAWEB, 2020)	
<i>Dendropsophus minutus</i> (pererequinha-do-brejo)		
	Foto (DUPONT, 2017). Som (AMPHIBIAWEB, 2020)	
<i>Ischnocnema guentheri</i> (perereca-de-folhiço)		
	Foto (PROVETE, 2008). Som (AMPHIBIAWEB, 2020)	
<i>Physalaemus cuvieri</i> (rã-cachorro)		
	Foto (PROVETE, 2007). Som (AMPHIBIAWEB, 2020)	

Fonte: Elaborada pelo autor.

## 4.3 Arquitetura de rede

A profundidade (quantidade de camadas) e a largura das redes (quantidade e tamanho de filtros por camada ou blocos de processamento paralelo) influenciam a sua capacidade de generalização e o consumo de recursos como memória (PONTI *et al.*, 2021). Por causa disso, pesquisas como as descritas ao longo do Capítulo 3 estudam diferentes arquiteturas para verificar quais delas alcançam melhores resultados para identificação de sons, arquiteturas essas que possuem diferenças tanto de profundidade quanto de largura (HERSHEY *et al.*, 2017; KONG *et al.*, 2020). Seguindo essa abordagem, esta pesquisa analisa os resultados de diferentes arquiteturas recorrentes em classificação de sons de animais, como algumas das elencadas por Stowell (2022). Também são analisadas as respostas dos modelos em relação a diferentes entradas, como as apresentadas na seção anterior, além das alterações de resultados geradas pelas configurações das próximas seções. Vale ressaltar que no escopo deste trabalho não foram executadas modificações significativas das arquiteturas consideradas, porque a construção de modelos específicos não era um dos focos da pesquisa.

## 4.4 Estratégias para treinamento

Processos de otimização como os envolvidos no treinamento de redes neurais, que buscam encontrar os melhores parâmetros a partir da avaliação de uma função objetivo, são influenciados pela inicialização desses parâmetros. Quando os parâmetros iniciais estão longe da solução, alguns métodos podem não convergir para um resultado apropriado ou, mesmo convergindo, podem apresentar comportamento instável em regiões onde a função objetivo não é convexa (NOCEDAL; WRIGHT, 2006). Uma abordagem convencional consiste em inicializar os parâmetros da rede de maneira aleatória, considerando uma distribuição de probabilidades como a gaussiana, com parâmetros fixos ou variáveis (PONTI *et al.*, 2017). Além disso, a depender da complexidade da rede e dos dados, o treinamento também depende de uma quantidade massiva de amostras (ex.: mais de 1000 por classe). Dessa maneira, pesquisas para identificação de padrões sonoros aplicam Transferência de Aprendizado (cf. Seção 2.4.4), utilizando modelos pré-treinados em bases de imagens (LEBIEN *et al.*, 2020), de sons genéricos (HERSHEY *et al.*, 2017) ou de sons naturais (LOSTANLEN *et al.*, 2019), que são refinados em problemas específicos. Como apontado por essas aplicações e no comparativo realizado por Dufourq *et al.* (2022), o uso de Transferência de Aprendizado, que é recorrente em outras áreas de aplicação, consegue melhorar a capacidade de generalização dos modelos em problemas relacionados com sons ambientais, além de reduzir a complexidade de modelagem das CNNs e o tempo necessário para definição dos seus hiperparâmetros. Assim, esta pesquisa compara resultados de modelos inicializados com pesos aleatórios com resultados de modelos pré-treinados. Estes

modelos são inicialmente treinados com tarefas de classificação, mesmo que de padrões variados, e com tarefas auxiliares de autossupervisão.

A adição de termos de regularização nas funções de custo das redes também é uma maneira de ajustar a sua capacidade de generalização (GOODFELLOW; BENGIO; COURVILLE, 2016). Esta pesquisa também define regularizadores para a função de custo e analisa seu impacto no treinamento de diferentes arquiteturas, além de verificar como essa regularização se comporta quando as entradas da rede são modificadas.

Por fim, os algoritmos empregados para otimização durante o treinamento das redes são cruciais para o sucesso desse processo. Por causa dessa importância, durante o tempo foram desenvolvidas várias abordagens de otimização que possuem suposições, processos e parâmetros específicos para aplicação em tarefas de *Deep Learning* (GOODFELLOW; BENGIO; COURVILLE, 2016). Assim sendo, técnicas de otimização também foram variadas para verificar a resposta das arquiteturas nos cenários de teste.

## 4.5 Testes e avaliação

Características acústicas como as descritas na Seção 2.2.2 e na Seção 2.3.1 são usadas na exploração e identificação de sons, apresentando resultados satisfatórios, como em Gan *et al.* (2020) e Gan *et al.* (2021). Dessa maneira, os vetores de características “manuais” propostos na Seção 4.2 são utilizados como entrada para classificadores lineares e seus resultados são utilizados como referências (*baseline*) para avaliar a qualidade dos resultados das abordagens implementadas com CNN.

Uma ferramenta recorrente para avaliação dos espaços de características de entrada de um classificador ou gerados por camadas de uma CNN é a análise por meio de métodos de visualização como Projeções Multidimensionais e de medidas de avaliação da coesão e segregação dos espaços de características (COIMBRA, 2016). Essas projeções e medidas, são empregadas nesta pesquisa para representar a estrutura dos espaços de características aprendidos pelas redes, principalmente nas camadas que antecedem a camada final de classificação, facilitando a comparação entre os modelos e com o classificador de referência.

Outra maneira de avaliar os modelos, visualiza as estruturas internas aprendidas pelas redes, por meio das similaridades entre os mapas de ativação gerados pelas suas camadas (NGUYEN; RAGHU; KORNBLITH, 2020). Por causa disso, visualizações tabulares são geradas para apresentar essas similaridades entre camadas de uma rede e comparar as estruturas dessas representações entre os modelos.

Para tornar os resultados reproduzíveis, todos os treinamentos e testes são executados com a inicialização de sementes usadas pelos métodos aleatórios. Além do mais, para avaliar o comportamento dos modelos de classificação em diferentes subconjuntos de

dados, aplica-se amostragem dos dados gerando subconjuntos distintos para treinamento e validação.

Na avaliação da classificação, são analisadas medidas que destacam a eficácia geral dos modelos e medidas específicas que reportam o comportamento deles por classe, além da visualização do desempenho das redes durante o treinamento, nos subconjuntos de treinamento e validação. Adicionado aos resultados numéricos e visuais, testes estatísticos também são executados para comparação dos resultados.

Todo o treinamento é executado considerando tarefas de classificação relacionadas com as espécies de interesse da [Seção 4.1.1](#), além de amostras genéricas, como descrito na [Seção 4.1.2](#). Já a avaliação é executada na classificação apenas das espécies de interesse para facilitar a comparação entre os capítulos de resultado.

## 4.6 Considerações finais

Os passos desta proposta seguem um fluxo comum de trabalho com CNN aplicada em problemas de classificação, como alguns dos listados no [Capítulo 3](#), mesmo assim apresenta questões relevantes para a aplicação em casos com padrões de sons de animais. Esses passos não foram executados de maneira conjunta e sequencial, sendo que os próximos capítulos descrevem os principais resultados relacionados com modificações na função de custo ([Capítulo 5](#)), com combinações de entradas das redes ([Capítulo 6](#)) e com autossupervisão ([Capítulo 7](#)). Da mesma maneira, as técnicas de avaliação dos resultados variam nesses capítulos. Os demais pontos destacados na [Figura 15](#) são analisados em paralelo com esses resultados, avaliando suas contribuições para cada um dos testes. Estruturas, parâmetros e medidas não foram relatados neste capítulo, exceto as espécies de interesse, de maneira que a proposta seja genérica o suficiente para aplicação em outros cenários com outras configurações. Logo, a descrição das suas especificidades é reportada nos próximos capítulos.

---

## QUANTIFICAÇÃO COMO REGULARIZADOR DE TREINAMENTO

---

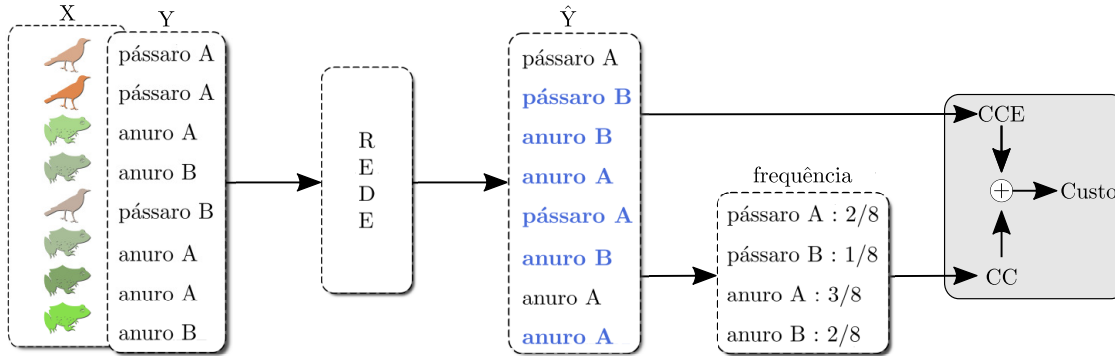
---

Este capítulo descreve a função de custo proposta em [Dias, Ponti e Minghim \(2021\)](#) que combina entropia cruzada e quantificação para melhorar a classificação de padrões sonoros de pássaros e anuros. Com isso, seu foco é a regularização descrita na [Seção 4.4](#) da proposta de pesquisa. Como detalhado na [Seção 2.4.6](#), algoritmos de quantificação são aplicados em tarefas de contagem e predição da distribuição de classes em um conjunto de dados, resumindo as classes no lugar de prever o rótulo de cada item do conjunto. Dessa maneira, o artigo usa o erro da distribuição predita como uma ferramenta de regularização do processo de aprendizagem, para gerar características capazes de discriminar os padrões de interesse, melhorando a capacidade de generalização dos modelos. A [Figura 16](#) ilustra esse processo, ressaltando a aplicação de um quantificador simples e intuitivo, que facilitou a avaliação do impacto da quantificação no treinamento. Os resultados destacam que a combinação não impacta os modelos de maneira negativa, mantendo a acurácia estável, além de melhorar métricas como a sensibilidade das classes e o Coeficiente de Silhueta dos espaços de características aprendidos pelas redes. Além disso, seguindo a proposta da [Seção 4.3](#), os testes compararam tanto arquiteturas simples, com poucas camadas, quanto arquiteturas profundas e pré-treinadas, sendo que aquelas podem superar os resultados das mais profundas na classificação dos padrões considerados ([DIAS; PONTI; MINGHIM, 2021](#)). Os códigos e melhores modelos treinados estão disponíveis no github<sup>1</sup>. Este capítulo também descreve resultados adicionais aos publicados no artigo.

---

<sup>1</sup> <<https://github.com/fabiofelix/CNN-CQ>>

Figura 16 – Exemplo de uma RNA com quantificação para identificação de pássaros e anuros a partir dos seus padrões de vocalização. Foram combinadas a entropia cruzada (CCE), que calcula a divergência entre  $Y$  e  $\hat{Y}$  e o quantificador *classify and count* (CC) que retorna a distribuição das classes a partir da predição da rede. Neste exemplo, é possível perceber uma *batch* com 8 amostras e 4 classes (pássaros A, B e anuros A e B), e mesmo a rede predizendo errado 6/8 classes (destacadas em azul), a frequência das classes está correta



Fonte: Adaptada de Dias, Ponti e Minghim (2021).

## 5.1 Metodologia aplicada

Em resumo, para avaliação da função de custo foram executados os seguintes passos dirigidos pelas ideias apresentadas no Capítulo 4. Primeiro, a quantidade de amostras por classe foi ajustada para evitar o desbalanceamento das classes visto na Tabela 3. Para isso, quatro técnicas de aumento de dados foram aplicadas aos arquivos de áudio, seguindo a proposta de Salamon e Bello (2017).

Da base aumentada, foram extraídas características acústicas usadas como entrada de um classificador, considerado como *baseline* dos experimentos, e gerados espectrogramas usados como entrada das redes neurais.

Para os testes, foram consideradas versões da ResNet-50 (HE *et al.*, 2016) e de um modelo simples de CNN sem quantificação (ResNet50 e CNN) e com quantificação (ResNet50-CQ e CNN-CQ). A função de custo proposta é o resultado da combinação linear entre a entropia cruzada e o erro absoluto da quantificação, ambos calculados para cada *batch* do treinamento, formulada como

$$\ell_{CQ}(X) = \lambda_1 CCE(X) + \lambda_2 CC_{err}(f(X)), \quad (5.1)$$

onde  $X$  é a *batch*,  $f(\cdot)$  é a rede que provê a predição das classes para computar o erro da quantificação e  $\lambda_i$  são pesos para a classificação ( $i = 1$ ) e para a quantificação ( $i = 2$ ).

Tabela 3 – Quantidades por espécie de arquivos com duração de 3 segundos

	espécie	rótulo	#trein.	#val.	#test.	Total
pássaro	<i>Basileuterus culicivorus</i>	basi_culi	376	107	54	537
	<i>Cyclarhis gujanensis</i>	cycl_guja	303	87	43	433
	<i>Myiothlypis leucoblephara</i>	myio_leuc	321	90	46	457
	<i>Pitangus sulphuratus</i>	pita_sulp	275	78	38	391
	<i>Vireo chivi</i>	vire_chiv	563	163	79	805
	<i>Zonotrichia capensis</i>	zono_cape	447	128	63	638
			2285	653	323	3261
anuro	<i>Adenomera marmorata</i>	aden_marm	89	27	13	129
	<i>Aplastodiscus leucopigylus</i>	apla_leuc	144	43	20	207
	<i>Boana albopunctata</i>	boan_albo	224	60	31	315
	<i>Dendropsophus minutus</i>	dend_minu	183	48	24	255
	<i>Ischnocnema guenteri</i>	isch_guen	104	32	15	151
	<i>Physalaemus cuvieri</i>	phys_cuvi	224	66	32	322
			968	276	135	1379
	<b>Total</b>		3253	929	458	4640

Fonte: Adaptada de Dias, Ponti e Minghim (2021).

### 5.1.1 Conjunto de dados

A Tabela 3 descreve as espécies e quantidades de arquivos de áudio considerados nos experimentos. Esses arquivos foram fornecidos pelo professor Milton C. Ribeiro<sup>2</sup> da UNESP/Rio Claro.

Os arquivos originais possuem 1 minuto cada e foram gravados no Corredor Ecológico Cantareira-Mantiqueira, em São Paulo, entre os meses de outubro<sup>3</sup> de 2016 e janeiro de 2017. Foram consideradas gravações entre 5:00 h e 8:25 h para capturar sons de pássaros e entre 18:30 h e 22:45 h, para capturar sons de anuros. Todos os arquivos são mono, no formato *WAVEform* (WAV), gravados com amostragem de 44.100 Hz, *bit-depth* igual a 16 e modulação *Pulse Code Modulation* (PCM).

Os especialistas do LEEC ouviram um subconjunto dos arquivos de 1 minuto para identificar e rotular as espécies de interesse, relacionadas na Seção 4.1.1 e listadas com suas quantidades de amostras na Tabela 3. Porções rotuladas dentro de cada arquivo foram extraídas e particionadas em segmentos de 3 segundos. Embora esses segmentos possuam outros sons, como insetos e ruído de fundo, eles não foram rotulados. O resultado desse processo gerou 4640 arquivos de 3 segundos (232 minutos no total), sendo que esse

<sup>2</sup> Spatial Ecology and Conservation Lab - LEEC. website: <<https://github.com/LEEClab>>

<sup>3</sup> O artigo reporta novembro, mas outubro é o mês correto de acordo com metadados das gravações.



conjunto foi particionado com amostragem estratificada das classes em treinamento (70%), validação (20%) e teste (10%), como apresentado na [Tabela 3](#).

Além de testes com a amostragem original dos arquivos (44.100 Hz), foram realizados testes com duas versões reamostradas em 22.050 Hz (intermediária) e 11.025 Hz (inferior). A finalidade foi analisar o impacto de amostragens menores na capacidade dos modelos de aprender características discriminantes para os padrões analisados. Para isso, a função *resample* da biblioteca *librosa*<sup>4</sup> (v0.7.2) foi aplicada com o parâmetro *type* igual a *scipy* (uso de interpolação para a reamostragem), o que diminuiu o tempo de processamento.

### 5.1.2 Aumento de dados

A base de áudios da [Tabela 3](#) não é balanceada, o que pode gerar modelos com baixa capacidade de generalização e predições imprecisas, principalmente para as classes com menos amostras ([JOHNSON; KHOSHGOFTAAR, 2019; WANG; PEREZ et al., 2017](#)). Buscando minimizar esses problemas, a quantidade de amostras de treinamento foi aumentada até alcançar um valor máximo  $\max_c = 565$  por classe, criando um total de 6780 amostras. O valor máximo é próximo da classe com maior quantidade de amostras de treinamento (*Vireo chivi*). Dessa maneira, para cada classe na coluna treinamento da [Tabela 3](#), o aumento de dados gerou  $m = \lceil (\max_c - \#class) / \#class \rceil$  cópias de algumas instâncias. Os novos arquivos foram adicionados aos dados de treinamento para extração de características e geração de espectrogramas.

Foram usadas quatro técnicas propostas por [Salamon e Bello \(2017\)](#): *pitch shifting*, *time stretching*, *noise addition* e *amplitude change*. Os dois primeiros métodos estão implementados na biblioteca *librosa*. As amplitudes de um sinal  $s$  foram modificadas aplicando  $s \times 10^{db/10}$ , enquanto que para a adição de ruído<sup>5</sup> foi considerada a adição  $s + 0,005 \times g(0, 0; 1, 0) \times \max(s) \times \mathcal{N}(0; 10^{db/10})$ , com uma distribuição uniforme de probabilidades  $g$  e uma distribuição normal  $\mathcal{N}$ . Os parâmetros dessas funções estão listados na [Tabela 4](#) e foram definidos de maneira experimental.

### 5.1.3 Extração de características

Para criação de um *baseline* para comparação, foram consideradas características acústicas descritas na [Seção 2.2.2](#) e na [Seção 2.3.1](#), como Bio,  $H_t$ ,  $H_f$ , H, ACI, AEI, M, AR, NDSI, ADI, NP, SPL, Roughness, Rugosity, RMS, PSD médio, SNR e doze coeficientes MFCC, gerando um vetor de características com 29 dimensões.

<sup>4</sup> <<https://librosa.github.io/librosa/>>

<sup>5</sup> <<https://www.kaggle.com/huseinzol05/sound-augmentation-librosa>>



Tabela 4 – Faixa de valores dos parâmetros das funções de aumento de dados. Tanto *pitch* quanto *amplitude* descartam o valor zero

	de	até	incremento
<i>stretch</i> (fator)	0,97	1,03	0,01
<i>pitch</i> (passos)	$-3 \times 10^{-4}$	$3 \times 10^{-4}$	$1 \times 10^{-4}$
<i>amplitude</i> (dB)	-0,12	0,12	0,04
<i>noise</i> (dB)	1	6	1

Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

Essas características foram extraídas com funções dos pacotes Seewave ([SUEUR; AUBIN; SIMONIS, 2008](#)) (v2.1.3), Soundecology<sup>6</sup> (v1.3.3) e tuneR<sup>7</sup> (v1.3.3), considerando como parâmetros os valores padrão dos pacotes. A rotina que calcula o MFCC retorna uma matriz de coeficientes (colunas) e componentes (linhas), sendo que as médias das colunas foram calculadas e as doze primeiras foram usadas nos testes ([DIAS, 2018](#)).

#### 5.1.4 Baseline

Como base para comparação, um *Support Vector Machine* (SVM) linear foi treinado com as características acústicas descritas. Por ser um classificador com garantias de aprendizagem, o resultado obtido pode ser considerado como um avaliador da capacidade de separação do espaço de características ([MELLO; PONTI, 2018](#)).

Foram usadas rotinas do scikit-learn<sup>8</sup> (v0.22.1) ([PEDREGOSA et al., 2011](#)), com *kernel* linear, *cost* = 1 e *iterations* =  $10^6$  para prevenir quantidades excessivas de iterações. Para definir o valor do parâmetro *cost*, foi executado um *grid search* em 12 cenários (4 cenários de classificação e 3 de taxa de amostragem dos arquivos. cf. [Seção 5.1.7](#)), variando a faixa de valores em [0, 1; 1; 10; 100; 1000]. A média da acurácia balanceada do SVM aplicado aos dados de validação da [Tabela 3](#) foi avaliada para definir os valores mais adequados para o parâmetro. Em 9/12 cenários, *cost* = 1 gerou os maiores valores de acurácia, enquanto que nos demais cenários os valores 0,1 e 1,0 alcançaram resultados similares.

#### 5.1.5 Espectrogramas

Para todas as amostras de 3 segundos, foram criados mel-espectrogramas ( $256 \times 256$ ) em escala de tons de cinza, com as rotinas da librosa, usando janela de Hanning de tamanho 2048 e 75% de sobreposição, 128 bandas mel e com os eixos verticais das imagens escalados para metade da taxa de amostragem de cada arquivo de áudio. O tamanho das

<sup>6</sup> <<http://ljvillanueva.github.io/soundecology/>>

<sup>7</sup> <<https://cran.r-project.org/web/packages/tuneR/>>

<sup>8</sup> <<https://scikit-learn.org/stable/>>

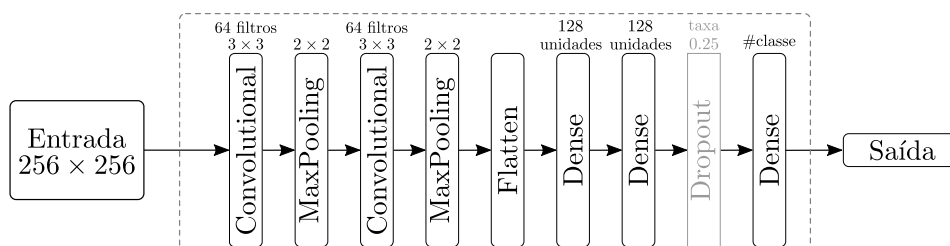
janelas e a taxa de sobreposição foram selecionadas de maneira a gerar resoluções maiores tanto na frequência quanto no tempo, para representar a maior quantidade possível de padrões.

### 5.1.6 Arquiteturas das redes neurais

Para avaliar o impacto da quantificação em uma arquitetura profunda, uma ResNet-50 (HE *et al.*, 2016), pré-treinada na ImageNet (DENG *et al.*, 2009), foi utilizada por ser encontrada em muitas tarefas de classificação de imagens (KORNBLITH; SHLENS; LE, 2019) e de detecção de espécies (HARVEY, 2018; LEBIEN *et al.*, 2020; THOMAS *et al.*, 2019). Os valores dos pixels de entrada foram normalizados com *max-norm*, dividindo seus valores por 255. Como as imagens possuem apenas 1 canal (tons de cinza), três cópias de cada uma foram concatenadas para formar um tensor com 3 dimensões (3D) usado no parâmetro *input tensor* da ResNet-50. Além do mais, as duas últimas camadas do modelo foram substituídas por outra *global average pooling* e uma camada densa com ativação *softmax*, que retorna a quantidade específica de classes treinadas. Para o treinamento, foram usados o otimizador SGD, com taxa de aprendizagem  $10^{-3}$ , 100 épocas de treinamento e *batch* de tamanho 50. Esse otimizador foi considerado por ser utilizado no artigo original da ResNet, a taxa de aprendizagem foi definida a partir das ponderações apresentadas em trabalhos como Becher e Ponti (2021), o tamanho de *batch* é o limite dessa arquitetura para a quantidade de memória disponível na placa de vídeo usada e a quantidade de épocas foi definida de maneira experimental.

Um modelo simples de CNN, apresentado na Figura 17, também foi testado, para verificar sua capacidade de classificar os dados em questão, comparar com a ResNet-50 e verificar a necessidade de um modelo profundo para classificar uma base pequena (com poucos milhares de amostras) e analisar o impacto da quantificação tanto em arquiteturas simples quanto profundas.

Figura 17 – Arquitetura de uma CNN simples. Os testes foram executados com e sem a camada de *dropout*



Fonte: Adaptada de Dias, Ponti e Minghim (2021).

As entradas da CNN também foram normalizadas com *max-norm*. Todas as suas camadas densas usam ativação ReLU, exceto a última camada que usa ativação *softmax*,

além das camadas de convolução e *pooling* aplicarem *padding* nas suas entradas. Os treinamentos foram realizados com e sem a camada de *dropout*. Para isso, foram usados o otimizador Adam, com taxa de aprendizagem  $10^{-4}$ , 100 épocas de treinamento e *batch* de tamanho 80. Exceto pelo otimizador, que foi definido de maneira experimental, os demais parâmetros foram escolhidos da mesma maneira que para ResNet.

Nos testes, são três cenários de ponderação investigados para verificar a relação entre quantificação e classificação na [Equação 5.1](#):  $\lambda_1 = \lambda_2 = 1,0$  (C1Q1),  $\lambda_1 = 1,0$  e  $\lambda_2 = 0,5$  (C2Q1), e  $\lambda_1 = 0,5$  e  $\lambda_2 = 1,0$  (C1Q2). Esses valores foram definidos para que os termos da função tenham a mesma importância (C1Q1) ou que um deles seja 2x mais importante do que o outro (C2Q1 e C1Q2).

Um processo de *early stopping* também foi usado para avaliar a função de perda nos dados de validação e encerrar o treinamento para evitar *overfitting*. O [Quadro 2](#) lista os parâmetros desse processo, definidos durante os testes, e do *checkpoint* para salvamento dos modelos gerados.

Quadro 2 – Parâmetros usados para encerrar o treinamento e salvar os modelos

early stopping	model checkpoint
mode = min min delta = 0,0001 patience = 20 restore best weights	mode = min   save best only

Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

Todos os modelos foram implementados com Python associado às bibliotecas Keras<sup>9</sup> (v2.2.5) e TensorFlow<sup>10</sup> (v1.10).

### 5.1.7 Avaliação

Para avaliar os resultados do SVM e das CNNs nos dados de teste, foram empregadas a acurácia balanceada, a matriz de confusão e a medida de sensibilidade (*recall* ou *sensitivity*) de cada classe, todas disponíveis no scikit-learn. Também foram executados testes *t* de Student com  $\alpha = 0,05$ , para comparar os resultados dos classificadores gerados. Nesses testes, a hipótese nula considera que as médias dos resultados comparados são iguais ou similares, conseqüentemente para a hipótese alternativa essas médias são diferentes (teste bicaudal); como a comparação é executada em resultados no mesmo conjunto de dados, existindo uma dependência entre as médias testadas, o teste é pareado. Além disso, as curvas de aprendizagem também foram analisadas (função de perda e acu-

<sup>9</sup> <<https://keras.io/>>

<sup>10</sup> <<https://www.tensorflow.org/>>

rácia categórica) para verificar a convergência dos modelos. Além da execução de testes com três taxas de amostragens distintas (cf. Seção 5.1.1), foram testados 4 cenários de classificação:

1. classificação binária com os rótulos `vire_chiv` e `phys_cuvi`, os mais frequentes de pássaros e anuros, respectivamente (2-class);
2. classificação das 6 classes de pássaros (bird-class);
3. classificação das 6 classes de anuros (anuran-class);
4. e classificação de todas as 12 classes de espécies (12-class).

Além do mais, as sementes dos métodos aleatórios do Python foram inicializadas, seguindo as orientações propostas pelo FAQ da biblioteca Keras<sup>11</sup>, com a finalidade de gerar resultados reprodutíveis. Assim sendo, tanto o treinamento do SVM quanto das redes foram executados cinco vezes, cada uma com diferentes sementes (1030, 1316, 1522, 1957 e 2359), sendo que a média e o desvio padrão da acurácia balanceada nos dados de teste foram calculados para avaliar os modelos.

Os espaços de características acústicas e aprendidos pelas redes foram visualizados com a projeção t-SNE (MAATEN; HINTON, 2008) que auxilia a avaliação das vizinhanças e da separação entre as classes (NONATO; AUPETIT, 2018). Para isso, foram consideradas as características extraídas pela penúltima camada de todas as redes (antes da última camada densa), porque nessa camada estão as características aprendidas, usadas para a classificação executada pela última camada. O Coeficiente de Silhueta (TAN; STEINBACH; KUMAR, 2005) também foi usado para avaliar os espaços de características (DIAS, 2018). Essas avaliações ajudam no entendimento dos espaços que foram criados e na identificação de possíveis dificuldades dos classificadores para diferenciar as classes de padrões sonoros.

Por fim, tanto a etapa de extração de características quanto treinamento do SVM foram executadas com um processador Intel Core i7-6850K CPU, 3,60GHz, 6 núcleos e 124 GB de memória RAM. Enquanto isso, o treinamento e teste das redes foi realizado em uma placa de vídeo NVidia Titan XP.

## 5.2 Resultados

Doravante, em todas as tabelas e figuras, as relações de pesos da função de custo definidas para a entropia cruzada (CCE) e erro da quantificação ( $CC_{err}$ ) são nomeadas

<sup>11</sup> <[https://keras.io/getting\\_started/faq/](https://keras.io/getting_started/faq/)>

como C1Q1, C2Q1 e C1Q2 (cf. Seção 5.1.6). A Tabela 5 e a Tabela 6 apresentam os melhores resultados de quantificação, a partir da avaliação da média da acurácia balanceada das combinações dos pesos anteriores.

### 5.2.1 Classificação binária e classificação de pássaros e anuros

Os resultados descritos na Tabela 5 e na Figura 18 destacam a dificuldade dos modelos em trabalhar no cenário 12-class com todas as classes de padrões, obtendo acurácia balanceada de 0,52, no melhor caso. Por outro lado, o cenário 2-class (uma classe de pássaro contra uma de anuro) é mais simples, alcançando acurácia balanceada até 0,95.

As redes neurais alcançaram média de resultados diferente do SVM, rejeitando a hipótese nula ( $p - valor \leq 0,05$ ). Essa hipótese também é rejeitada quando os resultados da ResNet50 são comparados com os da CNN, ao menos em uma das taxas de amostragem testadas; entre ResNet50-CQ e CNN-CQ (2-class), ao menos em uma taxa de amostragem; e entre CNN-CQ e CNN-CQ com dropout (12-class), na menor amostragem (11.025 Hz).

A Figura 18 facilita a identificação tanto da dispersão estreita em torno das medidas de centralidade quanto da similaridade dos resultados, independentemente da taxa de amostragem, sobretudo na Figura 18a. A Figura 18b destaca que os resultados na amostragem inferior são os mais baixos, talvez porque essa amostragem não representa toda a faixa de frequências contida nos áudios.

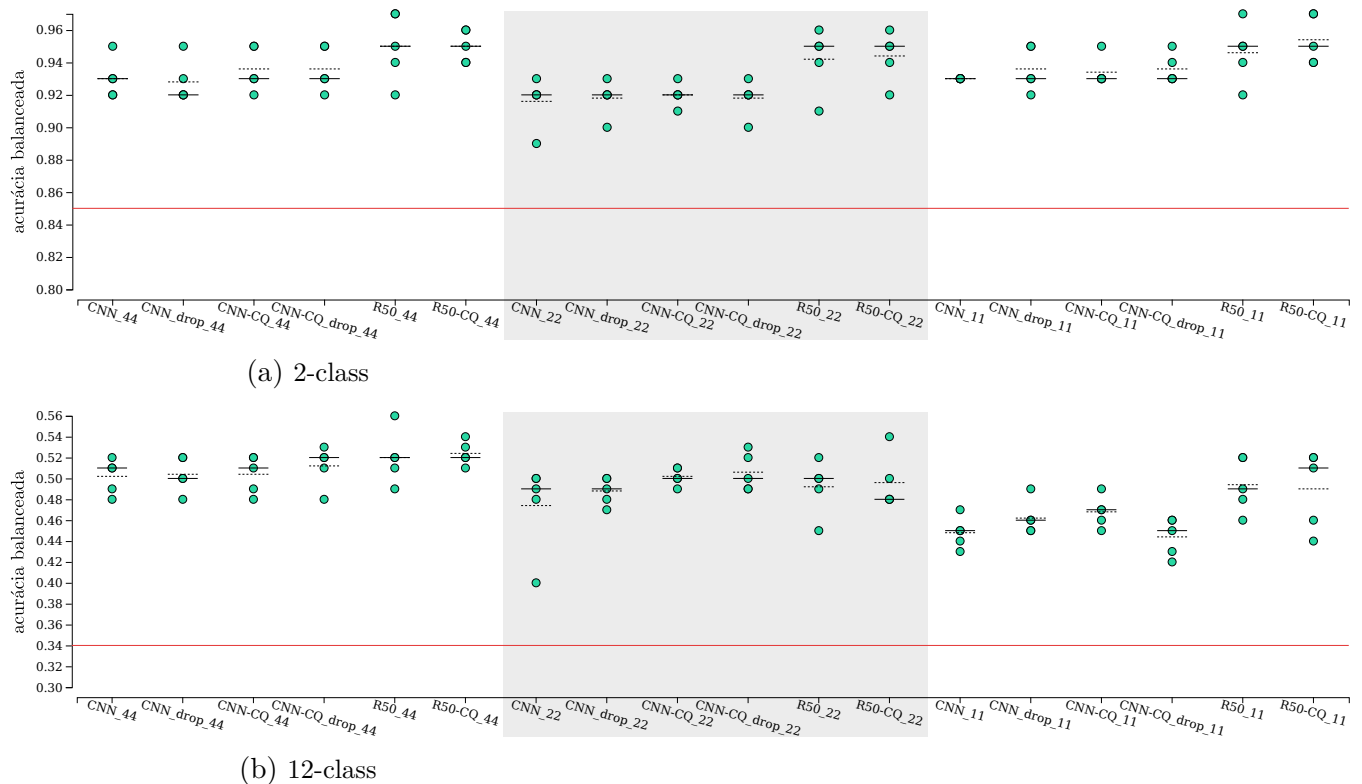
Tabela 5 – Média e desvio padrão da acurácia balanceada das RNAs aplicadas aos dados de teste. As células resumem os resultados dos modelos gerados com sementes diferentes. O grupo de linhas CNN não possui quantificação, enquanto que os grupos ResNet50-CQ e CNN-CQ apresentam apenas os melhores resultados de cada cenário

		2-class			12-class		
		44.100 Hz	22.050 Hz	11.025 Hz	44.100 Hz	22.050 Hz	11.025 Hz
SVM		0,78±0,00	0,85±0,00	0,83±0,00	0,30±0,00	0,26±0,01	0,33±0,00
ResNet50	–	0,95±0,02	0,94±0,02	0,95±0,02	0,52±0,03	0,49±0,02	0,49±0,03
ResNet50-CQ	–	*0,95±0,01	**0,94±0,02	*0,95±0,02	**0,52±0,01	**0,50±0,02	**0,49±0,04
CNN	–	0,93±0,02	0,92±0,01	0,93±0,00	0,50±0,02	0,47±0,04	0,45±0,01
	dropout	0,93±0,00	0,92±0,01	0,94±0,01	0,50±0,02	0,49±0,01	0,46±0,02
CNN-CQ	–	**0,94±0,01	**0,92±0,01	*0,93±0,01	**0,50±0,02	*0,50±0,01	*0,47±0,01
	dropout	**0,94±0,01	**0,92±0,01	*0,94±0,01	**0,51±0,02	*0,51±0,02	*0,44±0,02

\*C1Q1, \*\*C2Q1 e \*\*\*C1Q2

Fonte: Adaptada de Dias, Ponti e Minghim (2021).

Figura 18 – Gráfico de valores individuais da acurácia balanceada gerado pelas RNAs aplicadas aos dados de teste. Cada área do gráfico (branca e cinza) mostra os resultados em uma taxa de amostragem específica (da esquerda para a direita: original, intermediária, inferior). Dentro de cada área, os resultados são apresentados na seguinte ordem, da esquerda para a direita: CNN, CNN com dropout, CNN-CQ, CNN-CQ com dropout, ResNet50 e ResNet50-CQ. As linhas sólidas e tracejadas dentro de cada grupo de valores representam, respectivamente, sua mediana e a média. A Linha vermelha na base destaca o melhor resultado do SVM entre as taxas de amostragem



Fonte: Adaptada de Dias, Ponti e Minghim (2021).

## 5.2.2 Classificação de pássaros e classificação de anuros

Os resultados da Tabela 6 e da Figura 19 seguem padrões similares aos da seção anterior: redes neurais com resultados superiores aos do SVM, exceto para o cenário anuran-class (na maioria das taxas de amostragem), onde ResNet50 e ResNet50-CQ (exceto em 22.050 Hz)<sup>12</sup> obtiveram resultados próximos aos do SVM. As comparações entre as RNAs também retornam diferenças, por exemplo com bird-class a CNN tem resultados superiores aos da ResNet50 em até 11 pontos percentuais e a CNN-CQ, quando comparada com a ResNet50-CQ, alcança resultados até 9 pontos acima. Para o cenário anuran-class, a CNN obteve resultados até 11 pontos superiores aos da ResNet50, em taxas de amostragem diferentes da original. Além do mais, a Figura 19a mostra dificuldades para a ResNet50 superar as CNNs e a Figura 19b apresenta diferença não significativa entre os resultados das RNAs e do SVM.

<sup>12</sup> Corrigido em relação ao artigo que apresenta 22.100

Resumindo, o cenário 2-class (classificação de uma espécie de pássaro e uma de anuro) é o mais simples de resolver, seguido pelos cenários bird-class, anuran-class e 12-class (todas as classes de pássaros e anuros). Ao comparar os resultados da Tabela 6, os resultados do SVM na classificação de pássaros são inferiores aos seus resultados na classificação de anuros e as RNAs alcançaram resultados satisfatórios na classificação de pássaros. Além disso, com as configurações usadas para os modelos, os resultados das ResNet50s foram inferiores aos das CNNs.

### 5.2.3 Análise da classificação de pássaros e anuros

Para aprofundar a avaliação, esta seção apresenta matrizes de confusão, valores de sensibilidade e projeções t-SNE dos melhores modelos da Tabela 5, referentes a 12-class, com amostragem original (44.100 Hz). Foram considerados os resultados de cada execução (sementes diferentes) para escolher os modelos a serem reportados. A diagonal da matriz de confusão da Figura 20 tem resultados maiores ou iguais aos do SVM, exceto para os rótulos aden\_marm e dend\_minu (classes de anuros) que possuem maior confusão (valores menores), pelo menos em uma das matrizes.

A Tabela 7 lista os valores de sensibilidade resultantes dos modelos anteriores, sendo que as maiores diferenças com o SVM estão nas linhas relacionadas com os rótulos cycl\_guja, vire\_chiv, e isch\_guen (duas classes de pássaros e uma de anuro). Ao comparar a coluna ResNet50 com ResNet50-CQ, as maiores diferenças positivas aparecem em duas classes de pássaros (myio\_leuc e pita\_sulp) e duas de anuros (aden\_marm e apla\_leuc). Quando a comparação é feita entre as colunas de CNN, em sete casos os modelos

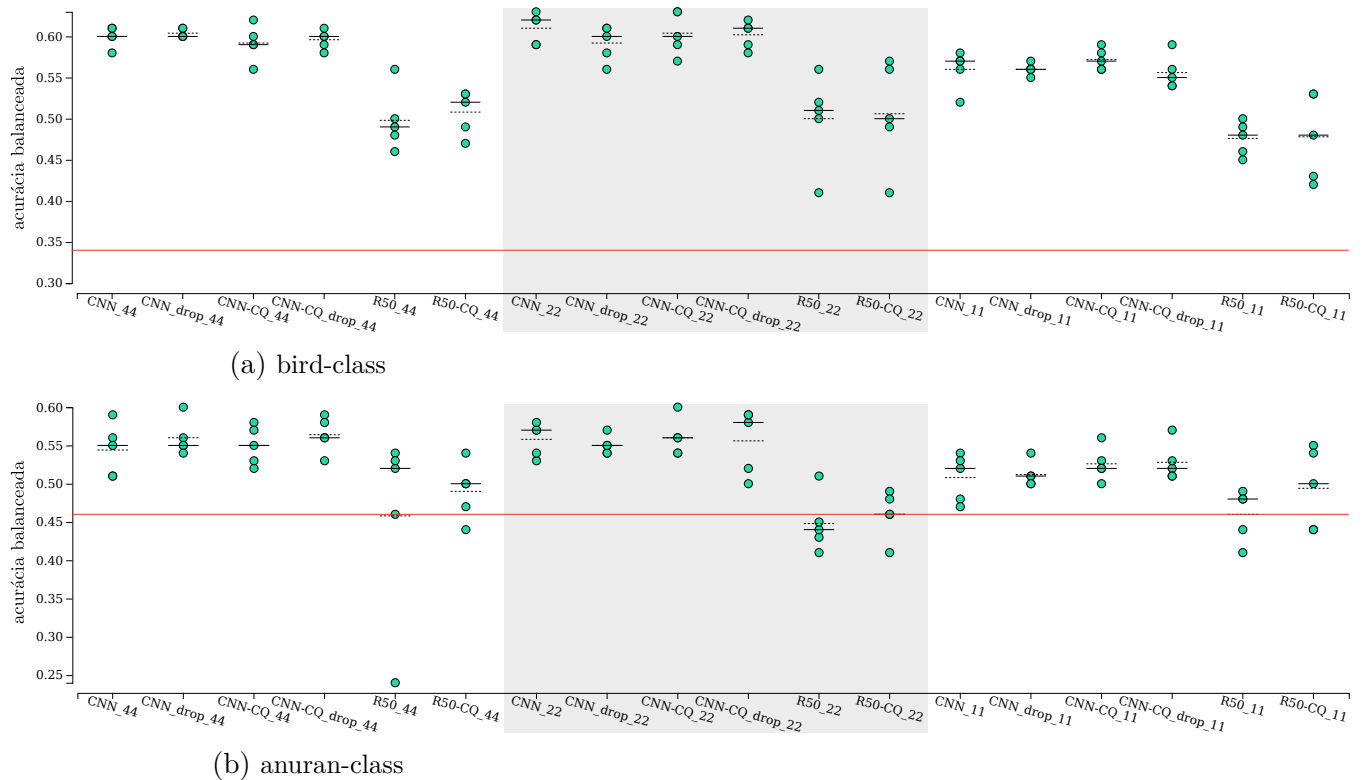
Tabela 6 – Média e desvio padrão da acurácia balanceada das RNAs aplicadas aos dados de teste. As células resumem os resultados dos modelos gerados com sementes diferentes. O grupo de linhas CNN não possui quantificação, enquanto que os grupos ResNet50-CQ e CNN-CQ apresentam apenas os melhores resultados de cada cenário

		bird-class			anuran-class		
		44.100 Hz	22.050 Hz	11.025 Hz	44.100 Hz	22.050 Hz	11.025 Hz
SVM		0,34±0,00	0,31±0,00	0,32±0,01	0,45±0,00	0,42±0,00	0,46±0,01
ResNet50	–	0,50±0,04	0,50±0,05	0,48±0,02	0,46±0,13	0,45±0,04	0,46±0,03
ResNet50-CQ	–	*0,51±0,03	**0,51±0,06	*0,48±0,05	*0,49±0,04	*0,46±0,03	**0,49±0,05
CNN	–	0,60±0,01	0,61±0,02	0,56±0,02	0,54±0,03	0,56±0,02	0,51±0,03
	dropout	0,60±0,01	0,59±0,02	0,56±0,01	0,56±0,02	0,55±0,01	0,51±0,02
CNN-CQ	–	***0,59±0,02	*0,60±0,03	**0,57±0,01	***0,55±0,03	*0,56±0,02	***0,53±0,02
	dropout	***0,60±0,01	*0,60±0,02	**0,56±0,02	***0,56±0,02	*0,56±0,04	***0,53±0,02

\*C1Q1, \*\*C2Q1 e \*\*\*C1Q2

Fonte: Adaptada de Dias, Ponti e Minghim (2021).

Figura 19 – Gráfico de valores individuais da acurácia balanceada gerado pelas RNAs aplicadas aos dados de teste. Cada área do gráfico (branca e cinza) mostra os resultados em uma taxa de amostragem específica (da esquerda para a direita: original, intermediária, inferior). Dentro de cada área, os resultados são apresentados na seguinte ordem, da esquerda para a direita: CNN, CNN com dropout, CNN-CQ, CNN-CQ com dropout, ResNet50 e ResNet50-CQ. As linhas sólidas e tracejadas dentro de cada grupo de valores representam, respectivamente, sua mediana e a média. A Linha vermelha na base destaca o melhor resultado do SVM entre as taxas de amostragem



Fonte: Adaptada de Dias, Ponti e Minghim (2021).

com quantificação alcançam resultados superiores, duas classes de anuros (aden\_marm e dend\_minu) e cinco de pássaros (basi\_culi, myio\_leuc, pita\_sulp, vire\_chiv e zono\_cape).

A Figura 21 e a Figura 22 exibem projeções t-SNE para inspeção visual dos espaços de características. Os modelos considerados para essa análise seguem o mesmo padrão dos demais desta seção, os modelos com melhores resultados de acurácia balanceada dentre as cinco execuções de cada cenário (cf. Seção 5.1.7). No geral, as projeções revelam que o espaço de características acústicas gera maior confusão visual, dificultando a identificação da separação entre as classes, enquanto que as características aprendidas pelas RNAs geram espaços com maior separação das classes. Além disso, nos espaços originais (sem projeção), o Coeficiente de Silhueta das características aprendidas (máximo de 0,68) é maior do que o das características acústicas (máximo de 0,05). Em geral, as CNNs extraíram características capazes de apresentar fronteiras mais perceptíveis entre as classes do



que as ResNets, em especial no cenário bird-class (cf. [Figura 21](#)), em que os espaços criados pelas CNNs têm silhuetas no intervalo  $[0,09;0,22]$  e as ResNets alcançam no máximo 0,026.

No cenário com todas as classes, as CNNs sem camada *dropout* extraíram características que possuem Coeficiente de Silhueta  $< 0,023$ , enquanto que quando o *dropout* é adicionado, a silhueta obtida é  $> 0,13$ . Também nas CNNs com esse tipo de regularização, a [Figura 22](#) apresenta maior coesão visual entre os grupos de pontos. Ao comparar projeções que representam espaços gerados com e sem uso da quantificação, as diferenças são imperceptíveis. Por fim, os valores de silhueta dos espaços de características são semelhantes, como no cenário 12-class que a diferença entre ResNet50 e ResNet50-CQ é de 0,0001 (menor diferença) e no cenário anuran-class a diferença entre CNN e CNN-CQ, ambas com *dropout*, é de 0,0750 (maior diferença).

#### 5.2.4 Análise de convergência dos treinamentos

Nesta seção, são reportadas avaliações da convergência dos processos de treinamento. São considerados os melhores modelos do cenário binário (2-class) e do cenário completo (12-class).

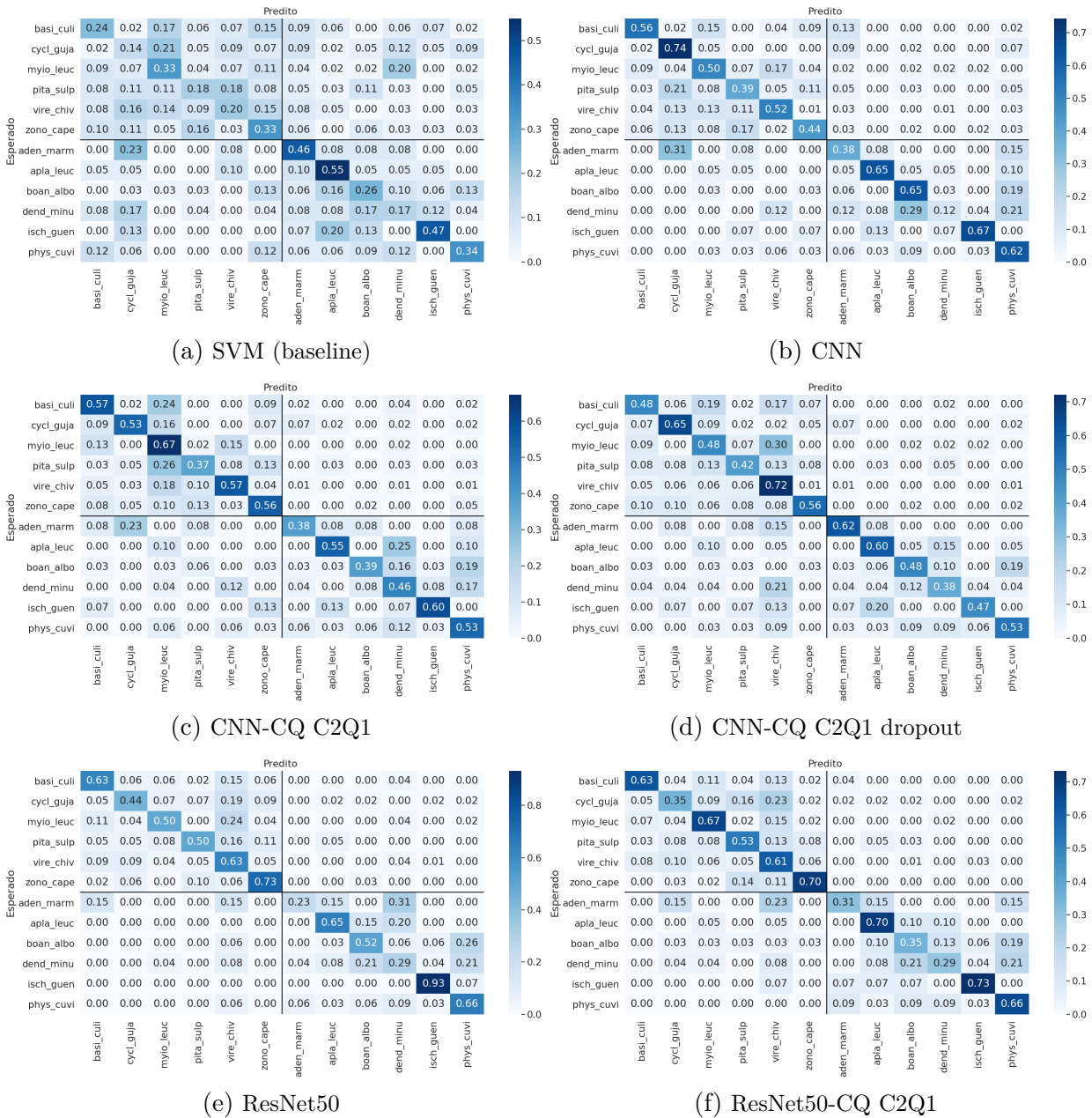
As curvas de aprendizagem na [Figura 23](#) e na [Figura 24](#), reportam como os modelos

Tabela 7 – Valores gerados pelos melhores modelos treinados/testados no cenário de **12-class**, com taxa de amostragem original, para todas as classes de pássaros seguidas das de anuros. As linhas destacadas apresentam as maiores diferenças ao comparar os resultados das redes com o SVM

rótulo	SVM	ResNet50	ResNet50-CQ C2Q1	CNN	CNN-CQ C2Q1	CNN-CQ C2Q1 dropout
basi_culi	0,24	0,63	0,63	0,56	0,57	0,48
<b>cycl_guja</b>	<b>0,14</b>	<b>0,44</b>	<b>0,35</b>	<b>0,74</b>	<b>0,53</b>	<b>0,65</b>
myio_leuc	0,33	0,50	0,67	0,50	0,67	0,48
pita_sulp	0,18	0,50	0,53	0,39	0,37	0,42
<b>vire_chiv</b>	<b>0,20</b>	<b>0,63</b>	<b>0,61</b>	<b>0,52</b>	<b>0,57</b>	<b>0,72</b>
zono_cape	0,33	0,73	0,70	0,44	0,56	0,56
aden_marm	0,46	0,23	0,31	0,38	0,38	0,62
apla_leuc	0,55	0,65	0,70	0,65	0,55	0,60
boan_albo	0,26	0,52	0,35	0,65	0,39	0,48
dend_minu	0,17	0,29	0,29	0,12	0,46	0,38
<b>isch_guen</b>	<b>0,47</b>	<b>0,93</b>	<b>0,73</b>	<b>0,67</b>	<b>0,60</b>	<b>0,47</b>
phys_cuvi	0,34	0,66	0,66	0,62	0,53	0,53
média/d.p.	0,31±0,13	0,56±0,19	0,54±0,17	0,52±0,17	0,52±0,09	0,53±0,10

Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

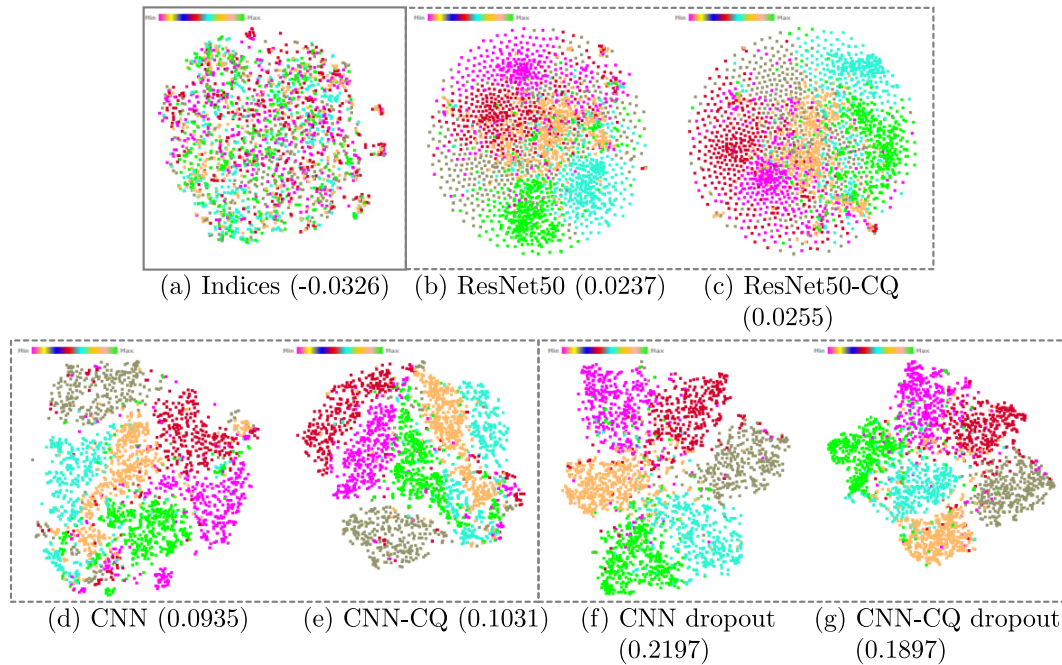
Figura 20 – Matrizes de confusão normalizada dos resultados dos melhores modelos treinados/-testados no cenário de **12-class**, com taxa de amostragem original. Subconjuntos de pássaros e anuros estão dispostos em quadrantes diferentes das matrizes



Fonte: Adaptada de Dias, Ponti e Minghim (2021).

são capazes de se ajustar aos dados de treinamento, mas são propensos a *overfitting*, principalmente em cenários com mais de duas classes, sendo que a técnica de *early stopping* é importante para minimizar esse problema. As diferenças entre as curvas na última época são maiores na ResNet50-CQ do que na CNN-CQ. Além disso, as curvas no cenário de classificação 2-class estão mais próximas do que no 12-class com os modelos de ResNet50, sugerindo ajustes mais consistentes dos modelos.

Figura 21 – Projeção t-SNE do espaço de características acústicas e dos espaços aprendidos (melhores modelos) da base de áudios aumentada usada para treinamento no cenário **bird-class**, com taxa de amostragem original. As cores dos pontos representam as classes e os valores dentro dos parênteses informam os Coeficientes de Silhueta dos espaços originais



Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

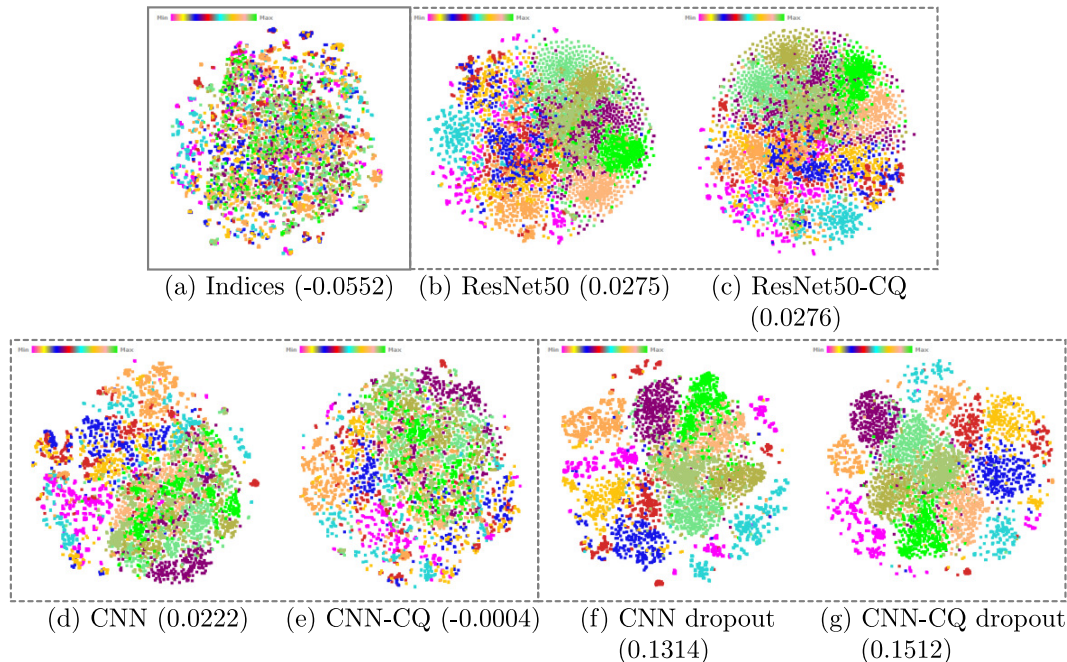
### 5.2.5 Resultados adicionais

Nesta seção, são descritos resultados com outras funções de quantificação e erro, diferente das usadas até aqui, obtidos após a publicação do artigo. É importante salientar que resultados de quantificação são satisfatórios quando os dados possuem relação temporal e a distribuição de classes varia entre janelas temporais consecutivas. Assim sendo, particionar os dados considerando uma sequência dessas janelas pode influenciar o comportamento relatado nas seções anteriores.

Na [Equação 5.1](#), o erro absoluto foi substituído pelo erro quadrático ou pela divergência das distribuições (cf. [Seção 2.4.6](#)). Os resultados dessas modificações são próximos dos apresentados até este ponto, não rejeitando a hipótese nula do teste  $t$  de Student ( $p - valor > 0,05$ ). No lugar do quantificador CC, o *adjusted classify and count* também foi testado, sendo que independente da função de erro os resultados também são próximos dos anteriores. Por outro lado, modificações da função de custo, por exemplo  $\ell_{CQ}(X) = CCE(X) \times \hat{p}$ , sendo  $\hat{p}$  a proporção retornada pelo quantificador, reduziram os resultados de acurácia dos modelos.

Para analisar a variação das frequências dos dados e seu impacto nos resultados, os dados da [Seção 5.1.1](#) foram divididos em janelas de um dia e as frequências das classes

Figura 22 – Projeção t-SNE do espaço de características acústicas e dos espaços aprendidos (melhores modelos) da base de áudios aumentada usada para treinamento no cenário **12-class**, com taxa de amostragem original. As cores dos pontos representam as classes e os valores dentro dos parênteses informam os Coeficientes de Silhueta dos espaços originais



Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

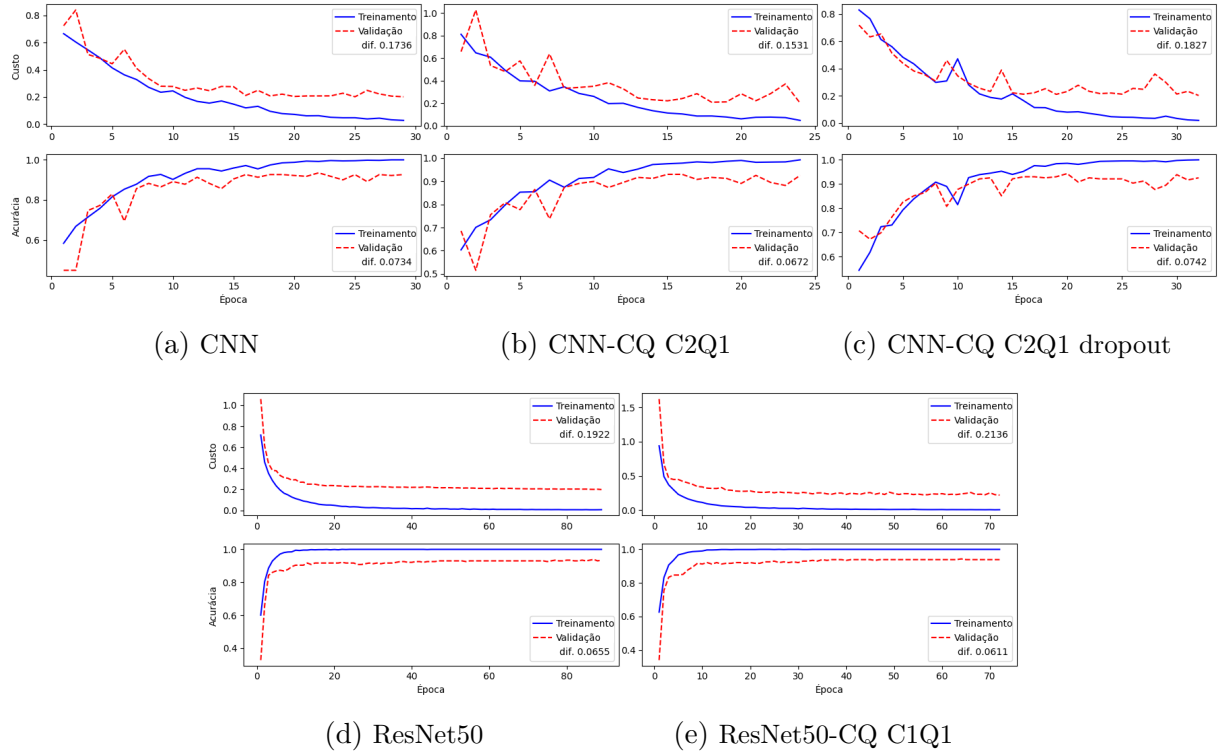
em cada um desses dias foi calculada, criando um vetor com 12 posições por janela, representando as classes de interesse. No total, são 100 dias de coleta entre 19 de outubro de 2016 e 28 de janeiro de 2017 (exceto os dias 25 e 26 de dezembro).

O teste de normalidade Shapiro-Wilk foi aplicado aos vetores de frequência, sendo que em 60% deles a hipótese nula foi rejeitada ( $p - valor \leq 0,05$ ), ou seja, as distribuições não possuem características significativas de normalidade. Por causa disso, o teste não paramétrico Mann-Whitney-Wilcoxon foi usado para comparar pares de dias consecutivos. Os resultados desses testes apresentam que em mais de 96% das comparações, a hipótese nula não foi rejeitada ( $p - valor > 0,05$ ), apresentando que existem semelhanças significativas entre as distribuições de classes dos pares de dias testados.

Considerando janelas de 10 dias e aplicando os mesmos testes, também foi encontrada evidência para aplicação do teste Mann-Whitney-Wilcoxon. Nesse caso, em todas as comparações de janelas consecutivas a hipótese nula também não foi rejeitada ( $p - valor > 0,05$ ).

A ResNet50-CQ foi treinada considerando as mesmas janelas de 10 dias, no cenário de classificação 12-class, com amostragem de 44.100 Hz. Foram executados 9 cenários, onde a rede é treinada com os dados das  $k$  primeiras janelas e testada com os dados da

Figura 23 – Curvas de aprendizagem dos melhores modelos treinados no cenário **2-class**, na taxa de amostragem original. Os valores Dif. das legendas apresentam a diferença entre treinamento e validação na última época. As curvas de acurácia são referentes à acurácia categórica



Fonte: Adaptada de Dias, Ponti e Minghim (2021).

janela  $k + 1$ , por exemplo, a rede treinada com os dados das 7 primeiras janelas (70 dias) é testada nos dados da oitava janela. Nesse treinamento os dados não foram aumentados para avaliar apenas as amostras originais.

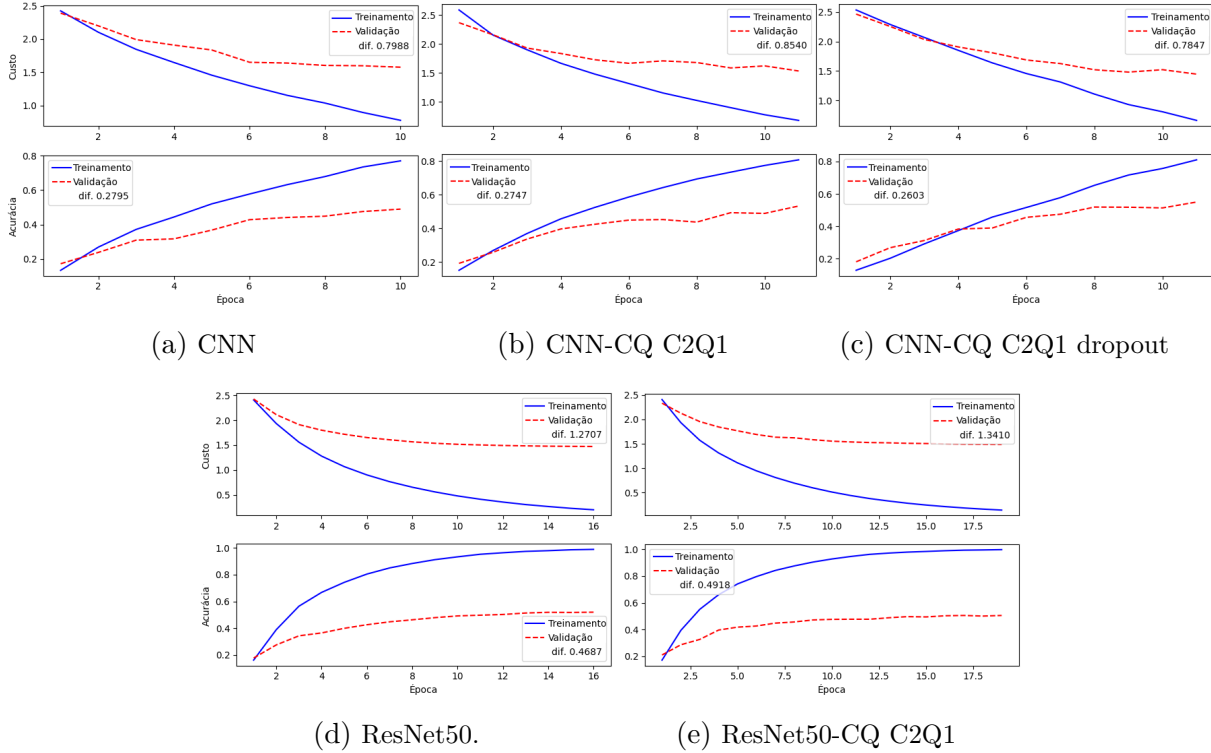
A Tabela 8 lista esses resultados, e testes  $t$  de Student foram aplicados para comparações entre os resultados com e sem quantificação. A hipótese nula do teste não foi rejeitada ( $p - valor > 0,05$ ), o que destaca similaridades entre os resultados, seguindo o padrão da Tabela 5 para o mesmo cenário. Os resultados melhoram conforme a quantidade de dados aumenta, entretanto, nos três últimos testes, esses valores diminuem, provavelmente porque existem classes sem amostras nas partições de teste (anuros: apla\_leuc e isch\_guen; e pássaro: myio\_leuc).

## 5.3 Discussão

Nesta seção, a média da acurácia balanceada (reportada na Seção 5.2) será identificada como *mean balanced accuracy* (mba). No geral, as RNAs alcançaram melhores resultados do que o SVM:  $mba \in [0,92;0,95]$  no cenário de classificação 2-class,  $mba \in [0,44;0,52]$



Figura 24 – Curvas de aprendizagem dos melhores modelos treinados no cenário **12-class**, na taxa de amostragem original. Os valores Dif. das legendas apresentam a diferença entre treinamento e validação na última época. As curvas de acurácia são referentes à acurácia categórica



Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

Tabela 8 – Média e desvio padrão da acurácia balanceada da ResNet50-CQ, no cenário 12-class, treinada em  $k$  janelas consecutivas e testada na janela  $k + 1$ . A primeira linha apresenta o melhor resultado reportado pelas configurações anteriores ([DIAS; PONTI; MINGHIM, 2021](#)), para comparação, e a linha destacada contém os melhores resultados nas partições de teste.

<b>k</b>	<b>C2Q1</b>	
-	0,52±0,03	0,52±0,01
1	0,32±0,01	0,32±0,02
2	0,25±0,03	0,28±0,02
3	0,28±0,05	0,27±0,04
4	0,33±0,03	0,31±0,02
5	0,34±0,04	0,36±0,04
<b>6</b>	<b>0,42±0,07</b>	<b>0,46±0,04</b>
7	0,30±0,13	0,36±0,04
8	0,34±0,02	0,29±0,07
9	0,35±0,05	0,35±0,03

Fonte: Elaborada pelo autor.

no cenário 12-class,  $mba \in [0,48;0,61]$  no cenário bird-class e  $mba \in [0,45;0,56]$  no cenário anuran-class. Para os resultados do cenário 2-class, além da pequena quantidade de classes, os padrões sonoros podem ter diferenças significativas entre vocalizações de pássaros e anuros, por causa disso o SVM com as características acústicas obteve valores razoáveis de  $mba \in [0,78;0,85]$ . Contudo, em cenários mais desafiadores como o 12-class, o SVM alcançou valores inferiores a esses,  $mba \in [0,26;0,33]$ .

Nos cenários intermediários, é interessante ressaltar que as redes neurais obtiveram resultados superiores na classificação de pássaros ( $mba = 0,60$ ) ao comparar com resultados da classificação de anuros ( $mba = 0,56$ ), um comportamento contrário ao do SVM que alcançou um maior desempenho na classificação de anuros. Isso pode indicar que as RNAs encontraram maior dificuldade para discriminar os padrões de anuros por causa de pouca variabilidade.

A ResNet50 (arquitetura profunda) obteve resultados similares ao da CNN (arquitetura rasa) tanto no cenário 2-class ( $mba \approx 0,95$  e  $mba \approx 0,93$ , respectivamente) quanto no cenário 12-class ( $mba \approx 0,50$  e  $mba \approx 0,48$ , respectivamente). Entretanto, tanto para bird-class quanto para anuran-class, a CNN obteve  $mba$  maior do que a ResNet50, o que pode estar relacionado com o maior refinamento desses cenários intermediários. Isso pode estar associado à especificidade dos padrões de som, um maior grau de refinamento das características dos sons ou à necessidade de um conjunto maior de amostras, imposta por arquiteturas mais profundas como a ResNet50. É importante salientar, que na base de áudios considerada, a sobreposição de classes pode ser um comportamento natural, porque padrões similares de vocalização podem acontecer em diferentes espécies de animais.

Tabela 9 – Resumo dos resultados. A partir da maior acurácia balanceada (ac.balan.), foram selecionadas a menor taxa de amostragem (tx.amost.) e o modelo com menor desvio padrão dos resultados (melhor modelo)

tarefa	tx.amost.↓	melhor modelo	ac.balan.↑
2-class	11 kHz	ResNet50 ResNet50-CQ	0,95
bird-class	22 kHz	CNN	0,61
anuran-class	22 kHz	CNN CNN-CQ	0,56
12-class	44 kHz	ResNet50-CQ	0,52

Fonte: Adaptada de [Dias, Ponti e Minghim \(2021\)](#).

A [Tabela 9](#) resume os resultados, onde é possível verificar que problemas com mais classes precisam ser tratados com arquivos de áudio gravados com taxa de amostragem elevada. Além do mais, o uso da quantificação na função de perda apresenta efeitos positivos na aprendizagem de características para o contexto de Paisagens Acús-

ticas. Especialmente, essa função, em alguns casos, aumentou os valores de medidas de classificação como a sensibilidade reportada na [Tabela 7](#). Claro que essa melhoria varia entre as classes, o modelo e, no caso da CNN, a aplicação ou não de uma camada de *dropout*. Por outro lado, a quantificação melhorou a sensibilidade média das classes (até 1%) e reduziu o desvio padrão do *mba*, além de não degradar os resultados ou gerar *overfitting*, em particular quando os resultados finais de acurácia tanto no subconjunto treinamento quanto de validação são considerados.

A análise visual dos espaços de características ratifica a melhoria alcançada pelas RNAs, quando comparados com o espaço das características acústicas. Os valores de silhueta deste espaço são sempre inferiores às silhuetas das características aprendidas pelas redes, o que reflete nos resultados inferiores do SVM. As projeções das características aprendidas pelas CNNs sugerem uma melhor formação de agrupamentos do que as obtidas pelas ResNet50s, justificando os resultados mais apropriados de uma arquitetura mais rasa. Pode-se adicionar a isso, a camada de *dropout*, que quando combinada com a quantificação melhora os valores de silhueta, por exemplo, no cenário bird-class a CNN alcançou 0,095 de silhueta e sua versão com *dropout* obteve 0,219. Tanto a inspeção visual quanto o Coeficiente de Silhueta destacam variações sutis entre espaços gerados com e sem a função de custo com quantificação. Isso está relacionado com a pequena variação da acurácia balanceada gerada pelos modelos com quantificação e o aumento ou diminuição da sensibilidade das classes.

Por fim, a quantificação é mais apropriada quando existe variação nas proporções de classes entre janelas temporais consecutivas. Nos testes adicionais relatados, não existe variação significativa nas frequências de classes em dados consecutivos. Logo, para melhor particionar os dados para treinamento dos modelos é necessário obter uma base real ou sintética cujas janelas temporais possuam maior variação, para verificar o comportamento dos modelos criados com a função de custo proposta.

## 5.4 Considerações finais

Este capítulo descreveu os resultados reportados por [Dias, Ponti e Minghim \(2021\)](#), que provê evidências empíricas sobre a combinação da entropia cruzada e do quantificador *classify and count*, cobrindo o ponto levantado na proposta sobre regularização do treinamento. Sobre variações na profundidade das arquiteturas, foram testadas a ResNet50 e uma CNN simples, com suas variações, revelando que esta mais simples alcança resultados melhores para discriminar classes de pássaros ou classes de anuros, além das duas arquiteturas obterem resultados similares para diferenciar pássaros de anuros. A adição da quantificação na função de custo gerou resultados estáveis, mesmo quando a taxa de amostragem dos arquivos foi reduzida. Além do mais, essa função de custo obteve níveis de



---

generalização razoáveis ao comparar os erros de treinamento e teste, e, o mais importante, melhorou métricas individuais como a sensibilidade de algumas classes e o Coeficiente de Silhueta dos espaços de características em cenários com uma quantidade significativa de classes.

Testes com outras configurações e arquiteturas também são recomendados para adequar a profundidade das arquiteturas ao tamanho da base de dados. Algumas mudanças de configurações, sobretudo de otimizadores, e outras arquiteturas são testadas com resultados promissores no [Capítulo 6](#).



---

## COMBINAÇÃO DE ENTRADAS PARA REDE CONVOLUCIONAL

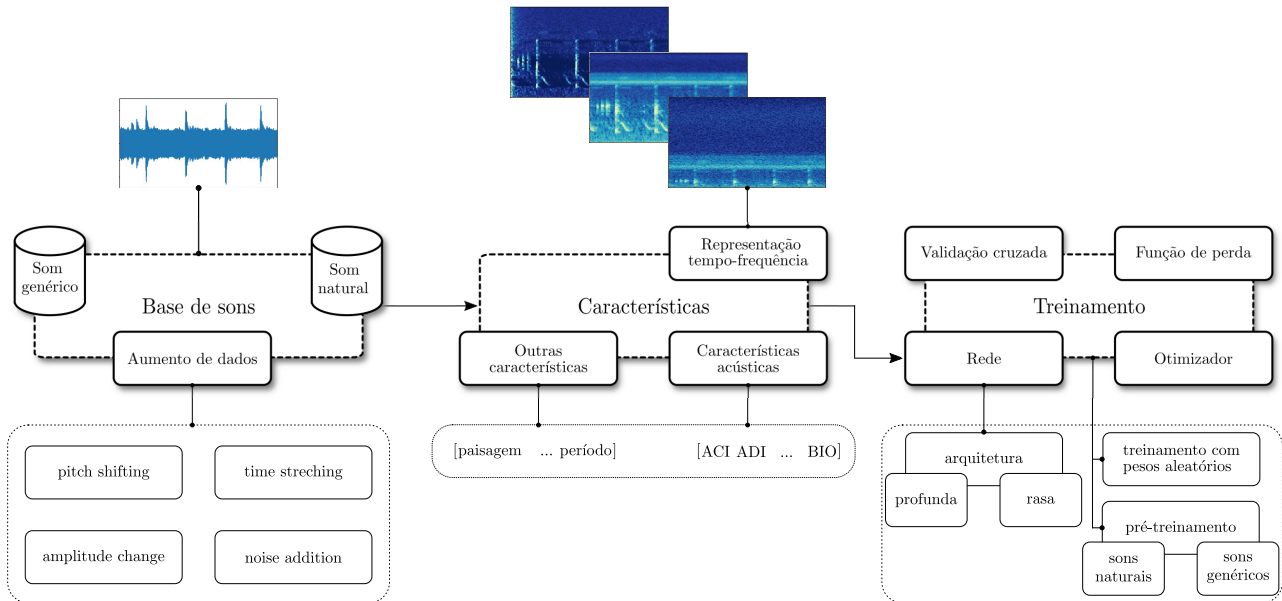
---

Este capítulo descreve estudos sobre a combinação de entradas para redes neurais, que leva em consideração, combinações de espectrograma e características como índices acústicos, para melhorar a classificação de padrões sonoros de pássaros e anuros. A finalidade é investigar se as combinações melhoram resultados de classificação, como definido na [Seção 4.2](#) da proposta de pesquisa. Como descrito na [Seção 2.2.1](#), tanto o espectrograma quanto suas variações são empregados em vários trabalhos de processamento de áudio, com resultados relevantes. Características acústicas, como as descritas na [Seção 2.2.2](#) e na [Seção 2.3.1](#), também são recorrentes em pesquisas que avaliam Paisagens Acústicas. Dessa maneira, imagens do espectro, características acústicas e suas combinações conseguem sumarizar e representar atributos importantes do som, comunicando informações ecológicas importantes. Este capítulo descreve a investigação dessas combinações, verificando o comportamento de redes convolucionais rasas (com poucas camadas), profundas ou largas (com filtros em paralelo), treinadas com pesos aleatórios ou pré-treinadas em bases genéricas ou específicas, em vários cenários de combinações de características de entrada. Além disso, algumas considerações são apresentadas sobre as etapas e as decisões tomadas na identificação de sons naturais com redes neurais, como as que são exibidas na [Figura 25](#), relacionadas com otimizadores, aumento de dados etc., variações elencadas para experimentos em partes da proposta, como discutido na [Seção 4.4](#). Escolhas apropriadas podem tornar os modelos mais robustos em cenários com grandes variações de ruído e volume dos sons. Os resultados destacam que as combinações de características possuem impacto mais significativo em redes rasas, tanto pré-treinadas quanto com pesos inicializados de maneira aleatória. Os códigos e melhores modelos treinados estarão disponíveis no [github](#)<sup>1</sup>.

---

<sup>1</sup> <<https://github.com/fabiofelix/CNN-Input-Combination>>

Figura 25 – Conjunto de passos investigados e escolhas específicas realizadas em cada um desses passos



Fonte: Elaborada pelo autor.

## 6.1 Metodologia aplicada

Em resumo, o procedimento aplicado para avaliar os modelos e as diferentes combinações de entradas é fundamentado no proposto no Capítulo 4 e segue os blocos destacados na Figura 25. Primeiro, os dados de treinamento listados na Tabela 10 foram balanceados pela aplicação de técnicas de aumento de dados para sons. No segundo passo, um conjunto de características foi extraído desses arquivos e um classificador linear foi treinado para criar um *baseline* para os experimentos. Por fim, variações de espectrogramas foram geradas e combinadas com as mesmas características do *baseline* para treinar diferentes arquiteturas de CNN.

### 6.1.1 Conjunto de dados

Os sons naturais e seus rótulos são os mesmos descritos na Seção 5.1.1, coletados em paisagens naturais, fornecidos pelo professor Milton C. Ribeiro<sup>2</sup> da UNESP/Rio Claro, rotulados por integrantes do mesmo grupo e explorados em outros trabalhos, como Scarpelli, Ribeiro e Teixeira (2021), Hilasaca *et al.* (2021) e Hilasaca, Ribeiro e Minghim (2021). Como descrito nessas referências, as gravações foram realizadas em áreas com diferentes coberturas de vegetação, sendo nomeadas como: área aberta (sobretudo áreas de agricultura e pastagem), floresta (áreas remanescentes de Mata Atlântica) e brejos (fragmentos de florestas próximos a corpos d'água).

<sup>2</sup> Spatial Ecology and Conservation Lab - LEEC. website: <<https://github.com/LEEClab>>

Tabela 10 – Quantidades por espécie de arquivos com duração de 3 segundos agrupados por grupos de espécies

	espécie	rótulo	#trein.	#test.	Total
pássaro	<i>Basileuterus culicivorus</i>	basi_culi	483	54	537
	<i>Cyclarhis gujanensis</i>	cycl_guja	390	43	433
	<i>Myiothlypis leucoblephara</i>	myio_leuc	411	46	457
	<i>Pitangus sulphuratus</i>	pita_sulp	352	39	391
	<i>Vireo chivi</i>	vire_chiv	724	81	805
	<i>Zonotrichia capensis</i>	zono_cape	574	64	638
			2934	327	3261
anuro	<i>Adenomera marmorata</i>	aden_marm	116	13	129
	<i>Aplastodiscus leucopiggyus</i>	apla_leuc	186	21	207
	<i>Boana albopunctata</i>	boan_albo	283	32	315
	<i>Dendropsophus minutus</i>	dend_minu	229	26	255
	<i>Ischnocnema guenteri</i>	isch_guen	136	15	151
	<i>Physalaemus cuvieri</i>	phys_cuvi	290	32	322
			1240	139	1379
outros		animal	108	12	120
		human	109	11	120
		natural	109	11	120
			326	34	360
<b>Total</b>			4500	500	5000

Fonte: Elaborada pelo autor.

Seguindo o indicado por Kahl *et al.* (2021), foram adicionadas amostras do Google AudioSet (GEMMEKE *et al.*, 2017), a fim de melhorar o aprendizado das redes, permitindo que elas diferenciem os sons das espécies de interesse de sons secundários e apresentados como sons genéricos na Figura 25. Considerando a ontologia<sup>3</sup> associada ao AudioSet, arquivos de diferentes classes foram baixados e salvos com os mesmos atributos dos arquivos da UNESP: formato WAV, taxa de amostragem de 44.100 Hz, *bit-depth* igual a 16 e modulação PCM. Como esses arquivos possuem aproximadamente 10 segundos, eles foram particionados em segmentos de 3 segundos. Com essa adição, os resultados de acurácia balanceada do *baseline* não foram alterados de maneira significativa, mas a média da medida de sensibilidade das classes aumentou em 1,25 ponto percentual.

Assim sendo, além das 12 espécies descritas na Tabela 10, as amostras do AudioSet foram agrupadas em três classes, considerando sons de animais, como cachorro, pássaros, insetos etc.; sons naturais, como vento, chuva, trovão etc.; e sons gerados direta ou indiretamente por humanos, como motores, piano, música etc. Por fim, a tarefa principal é

<sup>3</sup> <<https://research.google.com/audioset/ontology/index.html>>

treinar os modelos para identificar as 15 classes apresentadas na tabela.

O resultado da combinação dos conjuntos de áudios (UNESP + AudioSet) foi dividido com amostragem estratificada das classes em treinamento (90%) e teste (10%), com um total de 5000 arquivos de 3 segundos (250 minutos no total). Durante o processo de treinamento das CNNs, foi aplicada validação cruzada com  $k = 5$ , sendo que em cada iteração uma das partições foi considerada como subconjunto de validação.

Para baixar os arquivos do AudioSet, foi empregada a biblioteca `youtube-dl`<sup>4</sup> (v2021.4.26) disponível em Python.

### 6.1.2 Balanceamento de classes

O conjunto de dados da Tabela 10 é desbalanceado e algumas técnicas de aumento de dados (SALAMON; BELLO, 2017) foram empregadas para reduzir os problemas relacionados com esse desequilíbrio, como baixa capacidade de generalização dos modelos e predições imprecisas para classes com menos amostras (JOHNSON; KHOSHGOFTAAR, 2019; WANG; PEREZ *et al.*, 2017). Para isso, foram usadas as mesmas técnicas de aumento, parâmetros e implementações abordadas na Seção 5.1.2: *pitch shifting*, *time stretching*, *noise addition* e *amplitude change*.

O processo de aumento gerou  $m = \lceil (\max_c - \#class) / \#class \rceil$  cópias modificadas de cada arquivo de 3 segundos e as adicionou ao subconjunto de treinamento para extração de características e geração de espectrogramas. Quando a validação cruzada é aplicada, primeiro os arquivos originais são divididos em 3600 ( $k - 1$  partições) para treinamento e 900 para validação. Depois disso, aquelas amostras das partições de treinamento são aumentadas até que cada classe (no início com  $\#class$  amostras) possua  $\max_c = 580$  amostras, criando 8700 arquivos. A definição do valor de  $\max_c$  está relacionada à classe com maior quantidade de amostras (*Vireo chivi*) nas partições de treinamento. Portanto, em cada iteração da validação cruzada, existem 8700 arquivos para treinamento, 900 para validação e 500 para teste.

### 6.1.3 Características manuais

Foram extraídas 30 características acústicas como as descritas na Seção 2.2.2 e na Seção 2.3.1: Bio,  $H_t$ ,  $H_f$ , H, ACI, AEI, M, AR, NDSI, ADI, NP, SPL, Roughness, Rugosity, RMS, média do PSD, SNR, BGN e 12 MFCCs. Além disso, informações sobre o local (área aberta, floresta e brejo) e período de coleta (manhã e noite) foram incorporadas. Essas informações foram codificadas com *one-hot encoding*<sup>5</sup> em 5 novas características e

<sup>4</sup> <<http://ytdl-org.github.io/youtube-dl/>>

<sup>5</sup> Codificação que transforma uma informação de entrada em vetores de zeros e uns, onde cada posição corresponde a um dos valores de entrada possíveis

combinadas com as características acústicas, gerando um vetor de 35 posições. A adição das características de local e período aumentaram em  $\approx 48\%$  os resultados do *baseline* (de 0,29 para 0,43).

As características acústicas foram extraídas com os pacotes do R: Seewave (SUEUR; AUBIN; SIMONIS, 2008) (v2.1.3), Soundecology<sup>6</sup> (v1.3.3) e tuneR<sup>7</sup> (v1.3.3). Todas as rotinas de extração que possuem o parâmetro frequência máxima usaram 22.050 Hz (ADI, AEI, BIO e o parâmetro biomax do NDSI), porque essa é a maior frequência capturada pela taxa de amostragem dos arquivos de áudio. Para as rotinas que dependem da Transformada de Fourier, foi considerada janela de Hanning com tamanho 1024 e taxa de sobreposição de 10% (PSD, SPL, SNR e H), para não elevar os tempos de processamento e o consumo de memória. Os valores do SPL e SNR são calculados a partir dos resultados do PSD. A rotina do seewave que calcula o índice H foi reimplementada para ajustar problemas de consumo de memória e retornar os valores de  $H_t$ ,  $H_f$ , do envelope de Hilbert (usado como entrada pelas medidas RMS e BGN), resultados da função interna *meanspec* (usada como entrada pelas medidas Roughness, NP e M), além de ser configurada com os parâmetros de Fourier descritos antes no parágrafo. O índice AR foi calculado com os valores dos índices  $H_t$  e M, no lugar de usar a função das bibliotecas, reduzindo tempo de processamento. Para o índice ACI do soundecology, o parâmetro *cluster\_size* foi inicializado com 1, para permitir cálculo de arquivos de 3 segundos. Para a rotina do tuneR que calcula o MFCC, foram considerados os parâmetros padrão. Ela retorna uma matriz de coeficientes (colunas) e componentes (linhas), sendo que as médias das colunas foram calculadas e as doze primeiras foram usadas nos testes (DIAS; PEDRINI; MINGHIM, 2021; DIAS; PONTI; MINGHIM, 2021). O código escrito para extrair essas características está disponível no github<sup>8</sup>.

#### 6.1.4 Definição de um *baseline*

Seguindo trabalhos anteriores (cf. Seção 5.1.4), um SVM linear também foi empregado como *baseline*, considerando os parâmetros *cost* = 1, *iterations* = 10<sup>6</sup> e o vetor de características da seção anterior como entrada. Em experimentos, normalizações das entradas do SVM não melhoraram de maneira significativa a classificação, por causa disso, não foram executadas normalizações das características manuais. Esse classificador foi utilizado devido às suas garantias de aprendizagem e por utilizar hiperplanos como superfície de decisão, de forma que métricas computadas sobre esses classificadores funcionam como medida da separação das classes no espaço de características (MELLO; PONTI, 2018). Nesse caso, assim como para as CNNs, também foi executada validação cruzada com  $k = 5$  nos dados no subconjunto de treinamento, treinando o classificador com  $k - 1$

<sup>6</sup> <<http://ljvillanueva.github.io/soundecology/>>

<sup>7</sup> <<https://cran.r-project.org/web/packages/tuneR/>>

<sup>8</sup> <<https://github.com/fabiofelix/AudioTools>>

partições, desconsiderando a partição de validação, porque os hiperparâmetros do classificador não foram refinados, e testando no subconjunto de testes da Tabela 10. Todas as funções executadas estão disponíveis no scikit-learn<sup>9</sup> (v0.22.1) (PEDREGOSA *et al.*, 2011) do Python.

### 6.1.5 Tipos de representação espectral utilizados

Para todos os arquivos de áudio foram geradas imagens em escala de tons de cinza ( $256 \times 256$ ) dos espectrogramas (spec), mel-espectrogramas (mel) e PCEN (pcen) com funções da biblioteca librosa<sup>10</sup> (v0.7.2), considerando janela de Hanning de tamanho 2048, sobreposição de 75% (ou *hop length* de 25%) e escala do eixo vertical das imagens de metade da taxa de amostragem dos arquivos. Tanto o tamanho quanto a sobreposição das janelas contribuem para criação de uma representação que possua boa resolução de frequência e de tempo para representar uma variedade maior de padrões. O mel-espectrograma foi configurado para retornar 128 bandas mel e o PCEN usa  $\delta = 2,0$ ,  $r = 0,5$  e  $\alpha = 0,98$ , como no artigo original (WANG *et al.*, 2017).

### 6.1.6 Arquiteturas e suas variações

Foram usadas 4 arquiteturas de rede: uma CNN2D simples inicializada com pesos aleatórios (DIAS; PONTI; MINGHIM, 2021), um modelo multitarefas não hierárquico (nomeado aqui como BirdVox) pré-treinado com dados do ANAFCC (CRAMER *et al.*, 2020) (cf. Seção 3.1), além da ResNet-50 (HE *et al.*, 2016) e da Inception-V3 (SZEGEDY *et al.*, 2016), ambas pré-treinadas com a ImageNet (DENG *et al.*, 2009). A ResNet-50 é uma arquitetura comum para classificação de espécies animais (HARVEY, 2018; LEBIEN *et al.*, 2020; THOMAS *et al.*, 2019), a Inception possui uma estrutura com filtros em paralelo de tamanhos distintos, o que varia a largura da rede e pode facilitar o aprendizado de padrões de diferentes tamanhos, a BirdVox apresentou bons resultados na classificação de pássaros e a CNN2D foi considerada para comparar os demais modelos com um modelo simples inicializado com pesos aleatórios. O uso de modelos pré-treinados na ImageNet é uma abordagem recorrente para Transferência de Aprendizado em cenários de classificação de padrões de imagens de espectrogramas, como pode ser visto no Capítulo 3 e no levantamento de Dufourq *et al.* (2022).

Todas essas arquiteturas foram treinadas e avaliadas com spec, mel, pcen e combinações dessas entradas, sendo que todas essas imagens foram normalizadas com *max-norm*, dividindo seus valores por 255. Nos casos em que as entradas não foram combinadas, foi realizada a menor quantidade possível de modificações nas arquiteturas para manter suas características originais.

<sup>9</sup> <<https://scikit-learn.org/stable/>>

<sup>10</sup> <<https://librosa.github.io/librosa/>>



A CNN2D tem a mesma estrutura proposta na [Figura 17](#), duas camadas convolucionais, seguidas por camadas de *pooling* e 3 camadas densas (*dropout* não foi considerado nestes testes). Diferente do original, o otimizador SGD foi aplicado para minimizar a função de custo com taxa de aprendizagem  $10^{-2}$ , *momentum* = 0,9, 100 épocas e *batch* de tamanho 80. O otimizador, taxa de aprendizagem e momentum elevaram os resultados do modelo, enquanto que o tamanho de *batch* é o limite dessa arquitetura para a quantidade de memória disponível na placa de vídeo.

A ResNet-50 também foi preparada como descrito no capítulo anterior e as mesmas configurações foram usadas na Inception-V3. Como todas as imagens usadas estão em tons de cinza (apenas 1 canal), três cópias de cada entrada foram concatenadas e usadas como entrada (3 canais) da ResNet e da Inception. As duas últimas camadas do modelo também foram substituídas por uma *global average pooling* e uma camada densa com ativação *softmax* e a quantidade específica de classes treinadas. Diferente do [Capítulo 5](#), o treinamento da ResNet usou otimizador Adam, com taxa de aprendizagem  $10^{-4}$ , 100 épocas e tamanho de *batch* igual a 30. As diferenças nas configurações de otimização geraram melhorias significativas nos resultados, como será apresentado na seção de resultados, e o tamanho de *batch* também é dependente da quantidade de memória de vídeo disponível. Para Inception, a otimização da função de custo foi realizada com RMSProp, com taxa de aprendizagem  $10^{-3}$ , também com 100 épocas, mas *batch* de tamanho 80. Para essa arquitetura, o otimizador e suas configurações trouxeram os melhores resultados e o tamanho de *batch* foi inicializado da mesma maneira que para os modelos anteriores.

Com a BirdVox, a entrada foi modificada para receber a resolução de imagens considerada nos testes ( $256 \times 256$ ), as quatro camadas densas do topo da rede (com 64, 1, 5 e 15 unidades) foram substituídas por uma camada densa com 64 unidades e as mesmas configurações originais da arquitetura (inicialização He Normal ([HE et al., 2015](#)), regularização  $L^2$  com fator  $10^{-3}$  e desconsiderando o uso de viés) e uma camada densa com ativação *softmax* e a quantidade de classes treinadas. O otimizador Adam também foi empregado para o treinamento desse modelo, com taxa de aprendizagem  $10^{-4}$ , 100 épocas e tamanho de *batch* igual a 80. As configurações de otimização desse modelo seguem o seu artigo original.

Todos os modelos foram implementados em Python (v3.5.2), associado com as bibliotecas Keras<sup>11</sup> (v2.2.5) e TensorFlow<sup>12</sup> (v1.10). Para carregar os pesos da BirdVox, considerou-se o modelo disponível na biblioteca birdvoxclassify<sup>13</sup> (v0.2.0).

---

<sup>11</sup> <<https://keras.io/>>

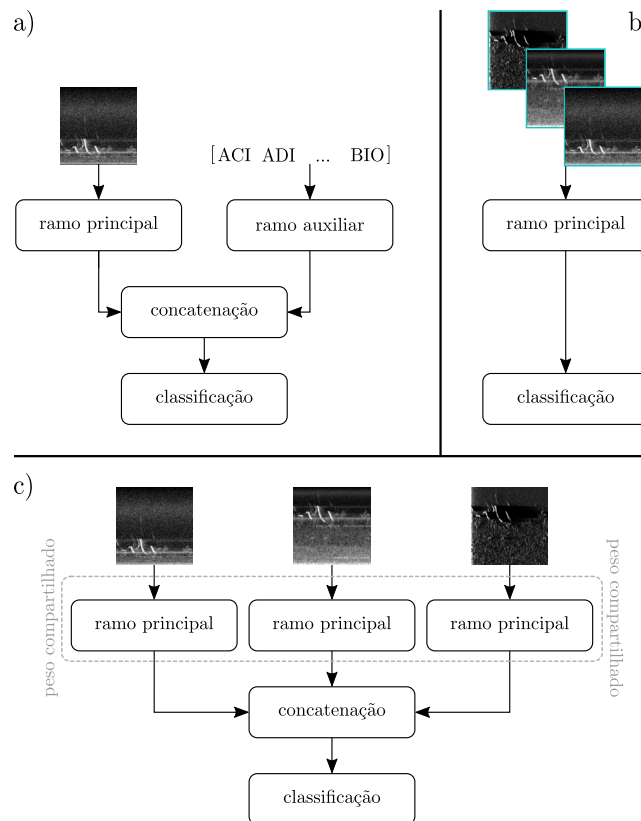
<sup>12</sup> <<https://www.tensorflow.org/>>

<sup>13</sup> <<https://github.com/BirdVox/birdvoxclassify>>

### 6.1.6.1 Combinando espectrogramas e características manuais

Inspirado por um exemplo de rede neural *context-adaptive* (LOSTANLEN *et al.*, 2019), um ramo auxiliar processa características manuais, como descrito na Figura 26a. Foram analisadas três configurações desse ramo auxiliar: uma camada densa com 128 unidades de ativação ReLU, uma camada de *batch normalization* ou uma combinação de *batch normalization* seguida pela mesma camada densa descrita no início. As características manuais não foram normalizadas antes do treinamento das redes. No ramo principal das CNNs, depois da camada *flatten* (CNN2D e BirdVox) ou *global average pooling* (ResNet50 e InceptionV3), as camadas originais de classificação foram substituídas por uma camada densa de 128 unidades e ativação ReLU. A BirdVox também aplica nessa camada as configurações das suas próprias camadas (inicialização, regularização e viés). Os resultados dos ramos principal e auxiliar são concatenados e enviados para duas camadas densas, sendo a primeira com 128 unidades de ativação ReLU e a última com ativação *softmax* e a quantidade de classes treinada.

Figura 26 – Representação das arquiteturas para processamento das combinações de características. *a)* Combinação de espectrogramas com características manuais, *b)* imagem com 3 canais, uma para cada variação de espectrograma e *c)* rede com 3 ramos, um para cada variação de espectrograma. As bordas azuis em *b)* são para facilitar a visualização nesta imagem, não sendo consideradas nas entradas dos modelos.



Fonte: Elaborada pelo autor.

### 6.1.6.2 Combinando variações de espectrograma

Além do treinamento separado com spec, mel e pcen, foram consideradas combinações dessas representações, como representado na [Figura 26](#). A hipótese levantada é que a combinação de representações diferentes pode melhorar o processo de aprendizagem das CNNs, de maneira similar ao trabalho de [Xie et al. \(2022\)](#). Por exemplo, spec torna mais visíveis harmônicos em frequências altas, mel exhibe melhor padrões inferiores (até 8 kHz) e pcen retorna uma versão filtrada do sinal.

Em um experimento (cf. [Figura 26b](#)), as representações foram combinadas em uma imagem de 3 canais (3-channels) e passadas para os modelos. Então, a entrada da CNN2D foi modificada para suportar 3-channels. Na BirdVox, depois da entrada, uma camada *lambda*<sup>14</sup> foi adicionada para retornar a média entre os 3 canais e enviar para o restante do modelo. Esse modelo também usa, depois da cada *flatten*, uma camada de 128 unidades e mesmas configurações das suas demais camadas densas (inicializador, regularizador e viés) antes da camada de classificação. A ResNet50 e a InceptionV3 foram pré-treinadas com uma entrada de 3 canais, dessa maneira apenas as duas camadas do topo foram modificadas seguindo as configurações da seção anterior.

No outro experimento, configurado como na [Figura 26c](#), a modelagem seguiu a ideia de redes Siamesas ([BROMLEY et al., 1994](#); [CHOPRA](#); [HADSELL](#); [LECUN, 2005](#)), construindo uma arquitetura com três ramos idênticos (3-inputs) e pesos compartilhados, mas que recebem entradas diferentes (spec, mel ou pcen). Como ramos, tanto a CNN2D quanto BirdVox foram consideradas até a primeira camada densa (128 unidades) após a camada *flatten*, enquanto que a ResNet50 e a InceptionV3 vão até a camada de *global average pooling*. Esses ramos são concatenados em cada modelo e o resultado passa por mais duas camadas densas (uma com 128 unidades e a outra com unidades relacionadas com as classes de treinamento) para classificar os arquivos de áudio. Nesse experimento, os tamanhos de *batch* para o treinamento foram modificados por causa da quantidade de memória disponível na placa gráfica utilizada: 30 amostras para CNN2D, 70 para BirdVox, para ResNet50 o tamanho foi reduzido para 15 amostras e InceptionV3 para 30.

Nesses dois experimentos as entradas são vistas de maneiras distintas. Na [Figura 26b](#), a rede aprende um espaço de características a partir da combinação dos três canais de entrada, enquanto isso, na [Figura 26c](#) a rede aprende três espaços de características que representam visões distintas do mesmo padrão de entrada e os combina para executar a classificação dos dados.

---

<sup>14</sup> Usada para adicionar uma função arbitrária como camada da rede

### 6.1.6.3 Combinando variações de espectrograma e características manuais

As estratégias descritas nas duas seções anteriores também foram combinadas. No experimento de 3-channels, as entradas de espectrogramas foram configuradas como na [Figura 26b](#) e combinadas com o ramo auxiliar descrito na imagem [Figura 26a](#). Para isso, foram seguidos os passos para 3-channels da seção [Seção 6.1.6.2](#), considerando as configurações da [Seção 6.1.6.1](#) para o ramo auxiliar.

No experimento de 3-inputs, tanto a CNN2D quanto a BirdVox adicionaram o ramo auxiliar na concatenação representada na [Figura 26c](#), e descrita na [Seção 6.1.6.2](#). Enquanto isso, a ResNet50 e a InceptionV3 adicionaram uma camada densa com 128 unidades e ativação ReLU depois da camada *global average pooling* e concatenaram os resultados das três cópias dessa estrutura com os ramos que processam características manuais.

### 6.1.6.4 Quantificação

Por fim, também foram realizados testes com a função de perda definida na [Seção 5.1](#) que combina entropia cruzada e quantificação. O objetivo é comparar os resultados atuais com quantificação e avaliar o impacto das modificações no conjunto de dados e no processo de treinamento com os resultados descritos no capítulo anterior. Para comparar com o artigo original, a função de perda foi adicionada aos modelos (inicialmente sem as combinações das seções anteriores) e apenas o mel-espectrograma foi usado como entrada. Os pesos da [Equação 5.1](#) foram inicializados seguindo o capítulo anterior:  $\lambda_1 = \lambda_2 = 1,0$  (C1Q1);  $\lambda_1 = 1,0$  e  $\lambda_2 = 0,5$  (C2Q1); e  $\lambda_1 = 0,5$  e  $\lambda_2 = 1,0$  (C1Q2).

### 6.1.7 Avaliação

Para avaliação, foram empregadas a acurácia balanceada, as curvas de aprendizagem dos modelos (função de perda e acurácia categórica) e a medida de sensibilidade (*recall*) das classes, todas disponíveis na biblioteca scikit-learn. Os modelos foram treinados para identificar 15 classes (cf. [Seção 6.1.1](#)), mas a análise da acurácia balanceada levou em consideração apenas as 12 classes relacionadas às espécies de interesse (cf. [Tabela 10](#)) para facilitar a comparação com o capítulo anterior. Também foram executados testes *t* de Student com  $\alpha = 0,05$ , para comparar os resultados dos classificadores gerados. Nesses testes, a hipótese nula considera que as médias dos resultados comparados são iguais ou similares, consequentemente para a hipótese alternativa essas médias são diferentes (teste bicaudal); como a comparação é executada em resultados no mesmo conjunto de dados, existindo uma dependência entre as médias testadas, o teste é pareado.

Para facilitar a reprodução dos resultados, tanto para SVM quanto para as redes neurais, as sementes dos métodos aleatórios do Python foram inicializadas (1030),

seguindo o descrito pelo FAQ da biblioteca Keras<sup>15</sup>. Como descrito nas seções anteriores, a validação cruzada usou  $k = 5$  e a média e o desvio padrão da acurácia balanceada nos dados de teste foram usadas para avaliação dos resultados.

Por fim, as etapas do *baseline* foram executadas com um processador Intel Core i7-6850K CPU, 3,60GHz, 6 núcleos e 124 GB de memória RAM. O treinamento e teste das redes foi executado com uma placa de vídeo NVidia Titan XP, com driver v387.26, Cuda v9.0 e cuDNN v7.0.5.15.

## 6.2 Resultados

Esta seção descreve os resultados experimentais dos testes com diferentes combinações de entradas para a CNN. Os resultados foram comparados com os obtidos por um SVM treinado com características manuais e avaliado com acurácia balanceada  $\in [0, 1]$ . Os modelos foram treinados para classificar 15 classes de sons, sendo que 12 delas (espécies de interesse) foram consideradas para avaliar os resultados, facilitando comparações com resultados do capítulo anterior. Foram testados dois tipos de entradas: imagens (spec, mel e pcen) e características manuais (características acústicas e informações sobre local e período de coleta). Seguindo a [Figura 26](#), as imagens foram combinadas com as demais características usando 3 tipos de ramos auxiliares: camada densa (dense128), camada de *batch normalization* (bnorm) e camada de *batch normalization* seguida por uma camada densa (bn+d128). As imagens de entrada foram combinadas de maneira a criar um tensor com 3 dimensões (3-channels) e em arquiteturas de pesos compartilhados com três ramos que recebem representações diferentes do espectro de frequências (3-inputs). Essas combinações, 3-channels e 3-inputs, também foram associadas com os ramos que processam características manuais, como descrito antes no parágrafo. Por fim, uma função de custo ponderada foi avaliada com 3 casos de pesos, C1Q1, C2Q1 e C1Q2 (cf. [Seção 6.1.6.4](#)), para comparar os resultados atuais com os resultados do [Capítulo 5](#). Todos os classificadores foram treinados com validação cruzada, com  $k = 5$ .

### 6.2.1 Combinando espectrogramas e características manuais

A [Tabela 11](#) apresenta os resultados da combinação de imagens e características manuais. Na primeira coluna de valores, os resultados com mel-espectrograma são, na sua maioria, superiores aos alcançados com as demais imagens, mesmo que a hipótese nula do teste  $t$  de Student não seja rejeitada ( $p - \text{valor} > 0,05$ ) quando são comparados os resultados mel/spec e mel/pcen para CNN2D e ResNet50, respectivamente. Para InceptionV3, qualquer comparação desses resultados possui grandes diferenças, por exemplo, com mel, essa arquitetura alcançou resultados 15 pontos percentuais acima dos resultados com spec.

<sup>15</sup> <[https://keras.io/getting\\_started/faq/](https://keras.io/getting_started/faq/)>

Contudo, BirdVox obteve resultados maiores com o pcen do que com spec (até 7 pontos percentuais a mais) ou mel (até 4 pontos percentuais a mais).

A adição de outras características processadas com dense128 sempre gera medidas menores do que 0,36, enquanto que sem combinações os resultados são maiores do que 0,47. Na sequência, as colunas bnorm e bn+d128 possuem resultados maiores ou iguais aos da coluna sem combinações, por exemplo, para BirdVox com pcen, a média da acurácia balanceada é 0,62 (bn+d128) contra 0,56 (sem combinações), com desvio padrão menor. Os resultados de ResNet50 com spec e InceptionV3 com mel (ambos combinados com bn+d128) são as únicas exceções, nas quais os resultados com combinação foram inferiores, alcançando 0,60 contra 0,66 no caso da ResNet50. Essas colunas (bnorm e bn+d128) contêm resultados similares, dentro da faixa [0,51;0,77] e com nenhuma diferença significativa entre elas, por exemplo, ao comparar os resultados alcançados pela InceptionV3 quando mel é a imagem de entrada.

O mel-espectrograma com combinações também obteve resultados iguais ou superiores às demais representações, exceto no caso da BirdVox cujos resultados com pcen são os melhores, sempre maiores do que 0,60. Mesmo com combinações, tanto CNN2D quanto BirdVox obtiveram resultados iguais ou menores a 0,62, enquanto que ResNet50 e InceptionV3 obtiveram resultados que chegam a 0,77. Entretanto, as adições de outras características influenciaram mais a CNN2D e a BirdVox (resultados até 7 pontos percentuais maiores) do que as arquiteturas mais profundas. Além do mais, exceto na coluna dense128, todos os resultados são superiores aos do SVM. Na [Figura 27](#), é possível verificar de maneira mais fácil essas diferenças entre os melhores resultados para cada modelo e a diferença entre CNN2D/BirdVox e ResNet50/InceptionV3.

### **6.2.2 Combinando variações do espectrograma e características manuais**

A [Tabela 12](#) reporta os resultados das combinações das imagens de espectrogramas, além de adicionar características manuais. Nas primeiras duas colunas sem características manuais, os resultados de 3-channels são maiores ou iguais aos resultados com 3-inputs. Por exemplo, a ResNet50 com 3-channels alcançou um resultado 68 pontos percentuais superior ao seu resultado com 3-inputs.

Da mesma maneira que na [Tabela 11](#), quando uma camada densa processa outras características, os resultados são reduzidos principalmente para 3-channels, por exemplo, de 0,73 para 0,19 no caso da InceptionV3. Em seguida, a maioria das comparações entre bnorm (ou bn+d128) e as colunas sem combinações apresentam resultados similares (hipótese nula do teste estatístico não rejeitada). Entretanto, todos os resultados da CNN2D, os resultados da ResNet50 (com 3-inputs e bn+d128) e da InceptionV3 (com 3-inputs e

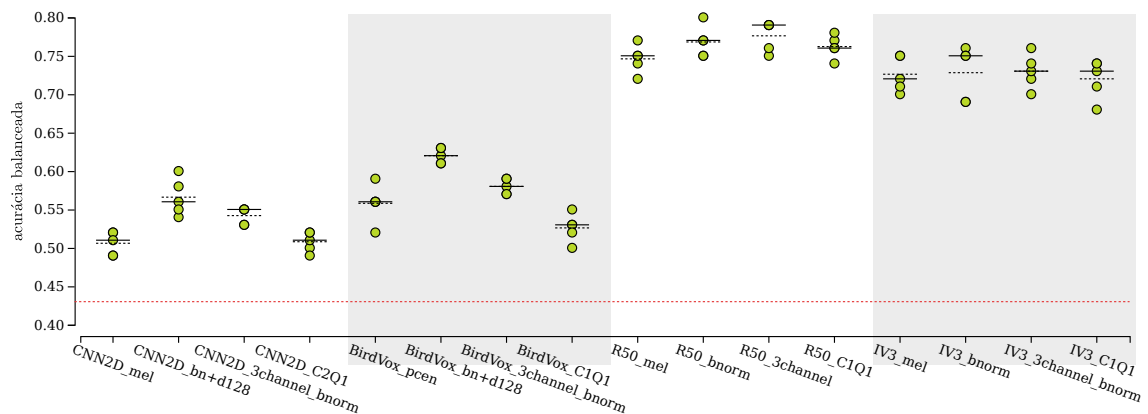
Tabela 11 – Média e desvio padrão da acurácia balanceada dos modelos aplicados ao subconjunto de testes. A primeira coluna de valores apresenta os resultados dos modelos apenas com imagens como entrada e as demais colunas combinam essas representações com características manuais processadas por diferentes ramos auxiliares. Os valores em destaque identificam os melhores resultados de cada modelo (maior média e menor desvio padrão)

			dense128	bnorm	bn+d128
CNN2D	spec	0,49±0,02	0,08±0,00	0,56±0,01	0,56±0,02
	mel	0,51±0,02	0,08±0,00	0,56±0,02	<b>0,57±0,02</b>
	pcen	0,48±0,02	0,08±0,00	0,54±0,02	0,55±0,01
BirdVox	spec	0,49±0,02	0,32±0,11	0,51±0,01	0,51±0,02
	mel	0,52±0,04	0,34±0,15	0,52±0,02	0,54±0,02
	pcen	0,56±0,02	0,35±0,13	0,61±0,03	<b>0,62±0,01</b>
ResNet50	spec	0,66±0,04	0,24±0,16	0,68±0,02	0,60±0,13
	mel	0,75±0,02	0,25±0,17	<b>0,77±0,02</b>	0,77±0,02
	pcen	0,70±0,08	0,25±0,17	0,75±0,02	0,73±0,02
InceptionV3	spec	0,58±0,04	0,08±0,00	0,63±0,02	0,62±0,03
	mel	<b>0,73±0,02</b>	0,20±0,26	0,73±0,03	0,71±0,02
	pcen	0,66±0,04	0,09±0,01	0,68±0,03	0,69±0,02

SVM (baseline) 0,43±0,03

Fonte: Elaborada pelo autor.

Figura 27 – Gráfico de valores individuais da acurácia balanceada gerado pelas CNNs aplicadas aos dados de teste. Cada área do gráfico (branco e cinza) mostra os melhores resultados de um modelo específico (da esquerda para a direita: CNN2D, BirdVox, ResNet50 e InceptionV3). Existe sobreposição entre pontos, por causa disso os resultados de alguns modelos apresentam quantidades diferentes de pontos. As linhas sólidas e tracejadas dentro de cada grupo de valores representam, respectivamente, sua mediana e a média. A linha vermelha na base representa a média das acurácias balanceadas do SVM



Fonte: Elaborada pelo autor.



bnorm ou bn+d128) alcançaram aumentos significativos. Por exemplo, para InceptionV3, enquanto a primeira coluna 3-inputs exibe 0,08, sua combinação com bn+d128 alcançou 0,34.

Tanto bnorm quanto bn+d128 geraram resultados similares, exceto para InceptionV3 com 3-channels, onde o processamento com bn+d128 reduziu os resultados em 5 pontos percentuais quando comparado com a bnorm. Em todas as combinações, 3-channels alcançou resultados maiores ou iguais aos de 3-inputs, por exemplo, ResNet50 com bnorm, resultado de 0,77 contra 0,15. A CNN2D e a BirdVox alcançaram valores de até 0,58, enquanto isso, ResNet50 e InceptionV3 chegaram a 0,78.

Todas as colunas de 3-channels, exceto as com dense128 e InceptionV3 com bn+d128, apresentam resultados maiores ou iguais às da primeira coluna da [Tabela 11](#). Um exemplo é a BirdVox com 3-channels e bnorm comparada com BirdVox e entrada pcen obtiveram, respectivamente, média de acurácia balanceada de 0,58 e 0,56. Em todos os casos de 3-channels, exceto com dense128, os resultados são superiores aos do SVM.

Uma comparação entre as combinações contidas na [Tabela 11](#) e [Tabela 12](#) destacam que a CNN2D e a BirdVox obtiveram resultados maiores com a adição de características manuais do que com a combinação 3-channels, por exemplo, a BirdVox com pcen e bn+d128 alcançou 0,62, com 3-channels obteve 0,57 e com 3-channels e bnorm alcançou um resultado intermediário de 0,58 (cf [Figura 27](#)). Por outro lado, tanto para ResNet50 quanto para InceptionV3 as comparações entre as entradas não rejeitaram a hipótese nula do teste estatístico ( $p - value > 0.05$ ), apresentando similaridade entre os seus resultados. Por exemplo, a ResNet50 com mel (ou 3-channels) combinado com bnorm gerou 0,77 e com 3-channels o resultado foi 0,78.

### 6.2.3 Quantificação

A [Tabela 13](#) descreve os resultados da função de custo com quantificação. Apenas o mel-espectrograma foi considerado como entrada porque essa representação alcançou os melhores resultados na [Tabela 11](#) (coluna sem combinações) e para comparar com os resultados anteriores. Ao comparar os resultados de acurácia balanceada e sensibilidade gerados pelos modelos, não existe diferença significativa entre os resultados com ou sem quantificação. Além disso, na [Figura 27](#), é possível verificar a relação da acurácia balanceada entre os modelos com quantificação e com as combinações descritas nas seções anteriores. Também foi calculado o Coeficiente de Silhueta dos dados de treinamento (4500 amostras), usando as características extraídas pela penúltima camada de cada modelo, porque nessa camada estão as características aprendidas, usadas para a classificação executada pela última camada. Em todos os casos, a quantificação melhorou os valores desse coeficiente em pelo menos 0,02 ponto (por exemplo, ResNet50 de 0,34 para 0,36), exceto para InceptionV3, onde a silhueta diminuiu de 0,40 para 0,36.



Tabela 12 – Média e desvio padrão da acurácia balanceada dos modelos aplicados ao subconjunto de testes. As duas primeiras colunas de valores apresentam os resultados dos modelos com **combinações de imagens** como entrada e as demais colunas combinam essas representações com características manuais processadas por diferentes ramos auxiliares. Os valores em destaque identificam os melhores resultados de cada modelo (maior média e menor desvio padrão)

	dense128			
	3-inputs	3-channels	3-inputs	3-channels
CNN2D	0,49±0,01	0,49±0,02	0,08±0,00	0,08±0,01
BirdVox	0,31±0,21	0,57±0,02	0,21±0,18	0,43±0,13
ResNet50	0,10±0,04	<b>0,78±0,02</b>	0,09±0,01	0,25±0,18
InceptionV3	0,08±0,00	0,73±0,03	0,10±0,04	0,19±0,24
	bnorm		bn+d128	
	3-inputs	3-channels	3-inputs	3-channels
CNN2D	0,53±0,02	<b>0,54±0,01</b>	0,53±0,03	0,54±0,02
BirdVox	0,21±0,19	<b>0,58±0,01</b>	0,35±0,20	0,57±0,02
ResNet50	0,15±0,09	0,77±0,02	0,24±0,03	0,76±0,01
InceptionV3	0,31±0,02	<b>0,73±0,02</b>	0,34±0,03	0,68±0,04

SVM (baseline) 0,43±0,03

Fonte: Elaborada pelo autor.

Tabela 13 – Média e desvio padrão da acurácia balanceada dos modelos aplicados ao subconjunto de testes. Resultados gerados apenas com mel-espectrograma como entrada. Os valores em destaque identificam os melhores resultados de cada modelo (maior média e menor desvio padrão)

	mel	C1Q1	C2Q1	C1Q2
CNN2D	0,51±0,02	0,51±0,02	<b>0,51±0,01</b>	0,50±0,01
BirdVox	0,52±0,04	<b>0,53±0,02</b>	0,53±0,02	0,53±0,02
ResNet50	0,75±0,02	<b>0,76±0,01</b>	0,75±0,04	0,76±0,01
InceptionV3	<b>0,73±0,02</b>	0,72±0,03	0,71±0,03	0,71±0,03

SVM (baseline) 0,43±0,03

Fonte: Elaborada pelo autor.

#### 6.2.4 Comparação dos melhores resultados

A Tabela 14 apresenta uma comparação dos melhores resultados da Tabela 11, Tabela 12, Tabela 13 e dos resultados apresentados por Dias, Ponti e Minghim (2021). A ResNet50 com mel, combinada com bnorm, ou 3-channels alcançou resultados adequados como destacado nas seções anteriores. Esses modelos também foram treinados com quantificação, para comparar com os resultados da seção anterior. Nessa comparação, existe grande diferença entre os resultados do artigo prévio e os resultados deste capítulo, em

especial por causa da mudança do otimizador e suas configurações. Além disso, a hipótese nula do teste  $t$  de Student não foi rejeitada ( $p > 0,05$ ), na comparação de resultados com e sem quantificação, indicando semelhança significativa entre eles.

Tabela 14 – Melhores resultados dos testes relacionados com a **ResNet50**. As primeiras colunas descrevem as principais configurações dos modelos e as duas primeiras linhas os resultados anteriores. As linhas 6 e 8 foram calculadas para apresentar o impacto da quantificação nos melhores resultados de combinação de características. A linha destacada apresenta o maior valor de acurácia e os demais valores em negrito resultados similares, considerando intervalo de confiança

Entrada	Otim.	Quant.	Acurácia Balan.
mel (DIAS; PONTI; MINGHIM, 2021)	SGD	–	0,52±0,03
mel (DIAS; PONTI; MINGHIM, 2021)	SGD	C2Q1	0,52±0,01
mel	Adam	–	0,75±0,02
mel	Adam	C1Q1	<b>0,76±0,01</b>
mel+bnorm carac.	Adam	–	<b>0,77±0,02</b>
mel+bnorm carac.	Adam	C2Q1	<b>0,77±0,02</b>
<b>3-channels</b>	<b>Adam</b>	–	<b>0,78±0,02</b>
3-channels	Adam	C1Q1	<b>0,77±0,02</b>

Fonte: Elaborada pelo autor.

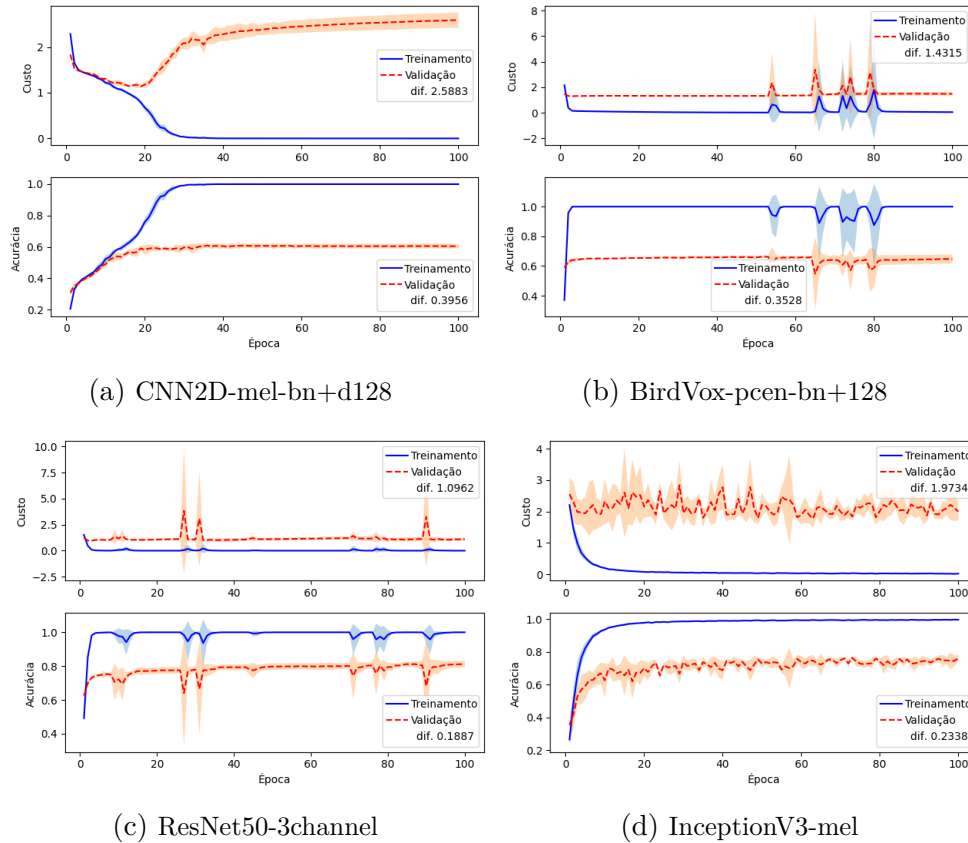
### 6.2.5 Avaliação de curvas de aprendizagem

A Figura 28 exibe as curvas de aprendizagem das redes com melhores resultados, descritas até agora. No geral, o treinamento alcança um vale onde as curvas da função de perda e da acurácia permanecem monótonas. A CNN2D começa a diminuir a perda (treinamento e validação) e melhorar a acurácia por volta da época 20, quando o modelo apresenta sobreajuste, aumentando a diferença entre as curvas de treinamento e validação. Na última época, a ResNet50 obtém a menor diferença entre as curvas: 1,0962 (perda) e 0,1887 (acurácia), e a InceptionV3 alcançou a maior variação das curvas de validação.

## 6.3 Discussão

No geral, com as configurações usadas, o mel-espectrograma atingiu os melhores resultados, tanto combinando com outras características quanto não. Por exemplo, para ResNet50 sem combinações, o mel como entrada gerou resultados que são até 14% superiores aos das demais representações, e para InceptionV3, usando *batch normalization* para processar características manuais, o mel obteve resultados até 16% maiores do que as demais imagens. Sugere-se que o alongamento realizado pelo mel-espectrograma nas frequências inferiores do espectro destaca padrões importantes que não são claramente

Figura 28 – Curvas de aprendizagem dos melhores modelos para cada arquitetura. As curvas no centro são a média das execuções da validação cruzada e a área em torno delas o desvio padrão. Os valores dif. das legendas comunicam a diferença entre treinamento e validação na última época. As curvas de acurácia são referentes à acurácia categórica



Fonte: Elaborada pelo autor.

identificáveis no espectrograma regular, enquanto que o pcen atenua informações importantes para diferenciar os padrões específicos dos dados trabalhados.

O ramo que processa as características manuais apenas com camada densa gerou resultados insuficientes (menores do que 0,36), porque as características não foram normalizadas antes do treinamento. Para o *baseline*, normalizações não influenciaram os resultados, por causa disso, nenhum pré-processamento foi aplicado antes de usar as características nas CNNs, para garantir condições similares para as comparações. Mesmo assim, as demais combinações geraram resultados maiores ou iguais aos das arquiteturas que processam apenas imagens, por exemplo, a BirdVox com o pcen, usando uma camada de *batch normalization* como ramo auxiliar, alcançou resultados de 5 pontos percentuais acima dos resultados da arquitetura com a mesma imagem de entrada, mas sem combinações. O ramo com *batch normalization* também gerou resultados similares (hipótese nula não rejeitada com  $p - valor > 0,05$ ) aos resultados que possuem essa mesma camada associada com uma camada densa, entretanto, com menos parâmetros para serem

treinados.

Diferente das demais arquiteturas, a BirdVox apresentou melhores resultados com pcen e suas combinações do que com mel-espectrograma. Essa arquitetura foi pré-treinada com o pcen e ainda existe a possibilidade de melhorar os parâmetros de geração dessa representação de maneira manual ou por meio do treinamento de uma rede específica (WANG *et al.*, 2017), podendo alcançar melhores resultados.

Na sequência, o caso 3-channels atingiu resultados superiores ao 3-inputs, independente da combinação com características manuais. Por exemplo, para ResNet50, 3-channels gerou resultados 8x (sem outras combinações) e 5x (combinação usando *batch normalization*) maiores do que 3-inputs. A hipótese levantada é de que o gradiente não foi suficiente para atualizar os pesos compartilhados e a origem da ResNet50 e da InceptionV3, treinadas com entradas com 3 dimensões, também contribuiu para 3-channels obter melhores resultados. Mais uma vez, as combinações usando uma camada densa como ramo auxiliar reduziram os resultados das classificações (menores do que 0,45). No geral, as combinações com características manuais não modificaram os resultados de maneira significativa (hipótese nula não rejeitada com  $p - valor > 0,05$ ). Contudo, essas combinações demonstraram resultados consideráveis em todos os cenários com CNN2D, em todas as combinações de InceptionV3 e 3-inputs, e para ResNet50 com 3-inputs, combinada com a sequência de *batch normalization* e camada densa.

As combinações de entradas impactaram mais na CNN2D e BirdVox do que na ResNet50 e InceptionV3 (cf. Figura 27). Sugere-se que um ramo auxiliar com poucas camadas e tamanho de saída reduzida, em relação ao ramo principal, pode não impactar em arquiteturas profundas como a ResNet50 (mais de 50 camadas) e como a InceptionV3 (mais de 90 camadas). Além do mais, as combinações de representações espectrais também não adicionaram informações relevantes para as arquiteturas profundas.

A aplicação de quantificação, seguindo Dias, Ponti e Minghim (2021), não apresentou modificações relevantes na acurácia balanceada ( $\pm 1$  ponto percentual) ou na medida de sensibilidade das classes. Quanto ao Coeficiente de Silhueta, a quantificação não gerou modificações significativas, apresentando um incremento de 0,02 ponto, exceto para InceptionV3 que reduziu os valores (0,04 ponto) quando a quantificação foi aplicada, o que ratifica sua diminuição na acurácia. No geral, esses resultados seguem o descrito no capítulo anterior que destacou que a função de perda com quantificação não influencia a acurácia mas gera modificações sutis na silhueta e na sensibilidade.

De modo geral, as comparações com o trabalho descrito no Capítulo 5 destacaram que para usar a ResNet50 (melhores resultados) e Transferência de Aprendizado, nos dados considerados, é necessário substituir o otimizador SGD (HE *et al.*, 2016), ou pelo menos, suas configurações, pelo Adam para ajustar melhor os padrões de interesse, provavelmente devido a maneira pela qual o Adam adapta suas taxas durante o processo

de treinamento (KINGMA; BA, 2014). Com Adam, a ResNet50 gerou resultados até 50% superiores aos resultados anteriores. Mais informações empíricas a respeito da definição de otimizadores e seus parâmetros podem ser encontradas em trabalhos como Becher e Ponti (2021).

Com a avaliação das curvas de aprendizagem, é possível notar que independente da profundidade da arquitetura, da sua largura ou da técnica de treinamento aplicada (pré-treinada ou inicializada com pesos aleatórios), existe uma dificuldade de generalização dos modelos (função de perda no treinamento próxima de zero e na validação  $> 1$ ), gerando acurácia balanceada menor ou igual a 0,78. Em um cenário com poucas amostras rotuladas (menos do que 10 mil) para treinar modelos profundos, pode-se inspecionar os parâmetros das arquiteturas, além de técnicas de regularização e treinamento para melhorar a generalização dos modelos.

## 6.4 Considerações finais

Este capítulo descreveu resultados que abordam uma série de testes com combinações de entradas, previstos na proposta desta pesquisa, para melhorar a representação aprendida por redes neurais, para identificação de padrões sonoros. Foram testadas quatro arquiteturas, uma inicializada com pesos aleatórios e as demais pré-treinadas com imagens de padrões de sons naturais e com uma base genérica de imagens, também cobrindo pontos levantados sobre profundidade, largura e inicialização discutidos na proposta. As evidências empíricas sugerem que o mel-espectrograma é a representação mais apropriada para os dados classificados, exceto para a BirdVox que tem o PCEN como melhor escolha de entrada. A combinação de imagens com características manuais pode ser implementada com a adição de um ramo simples que contém apenas uma camada de *batch normalization*. Essas combinações são apropriadas para arquiteturas pequenas, por exemplo, com duas ou três camadas convolucionais como a BirdVox, mas geram melhorias sutis na ResNet50 e InceptionV3, mesmo com representações com 3 dimensões das variações do espectrograma. Por fim, a função de custo com quantificação apresentou resultados similares aos do Capítulo 5.

Os testes melhoraram os resultados de redes rasas e geraram variações mínimas em modelos profundos pré-treinados. Mesmo assim, estes modelos alcançaram os melhores resultados porque não foram inicializados com pesos aleatórios, seguindo o que foi apresentado na revisão de Dufourq *et al.* (2022). Também, existem problemas na generalização que geraram resultados de classificação desfavoráveis, com erro maior do que 20%, o que é uma questão a ser investigada. Assim sendo, em trabalhos futuros, é importante avaliar outros tipos de entradas, seus parâmetros de criação, refinamentos da estrutura das arquiteturas e avaliação de abordagens de regularização e treinamento para melhorar

a capacidade de generalização dos modelos. Além disso, é importante conduzir uma avaliação aprofundada sobre otimizadores e suas configurações no cenário de identificação de sons.

---

## AVALIAÇÃO DE AUTOSSUPERVISÃO PARA PADRÕES SONOROS

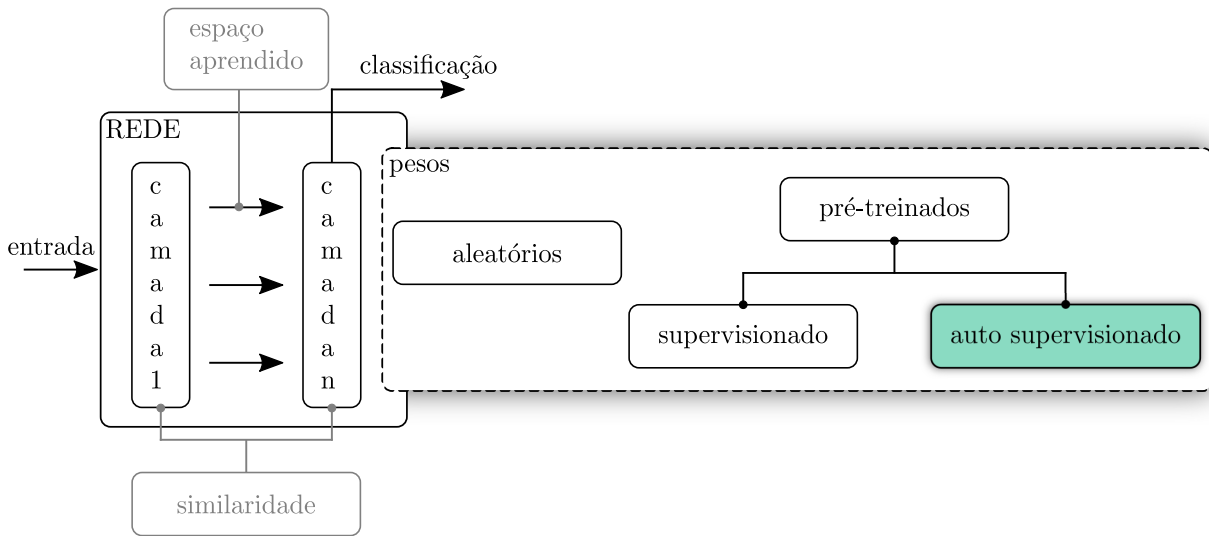
---

Este capítulo descreve uma sequência de testes para avaliar o impacto de técnicas de Aprendizado Autossupervisionado em um cenário de identificação de sons naturais. Logo, seu foco é a inicialização dos pesos das redes, definida na [Seção 4.4](#) da proposta de pesquisa. Como visto na [Seção 2.4.4](#), o treinamento de um classificador em um problema genérico e a transferência do conhecimento adquirido nessa tarefa para uma mais específica, é uma abordagem empregada para melhorar a capacidade de generalização dos modelos. Tarefas não supervisionadas também são consideradas para esse pré-treinamento, obtendo resultados significativos, como é destacado na [Seção 2.4.5](#). Este capítulo aplica essas tarefas para o pré-treinamento de modelos que são refinados para identificação de padrões sonoros de espécies animais. Diferente de trabalhos que usam bases como a ImageNet para as tarefas de autossupervisão, este capítulo descreve uma análise do comportamento dessas tarefas em uma base com quantidade reduzida de amostras para o pré-treinamento. A [Figura 29](#) ilustra os tipos de inicialização de pesos testados, ressaltando os caminhos para avaliação dos resultados, como análise dos espaços de características aprendidos e da estrutura interna de similaridade entre as camadas das redes. Além disso, são testadas arquiteturas com diferentes profundidades e larguras (com filtros em paralelo), seguindo o proposto na [Seção 4.4](#). Os resultados destacam a capacidade da autossupervisão de gerar modelos que obtêm medidas próximas de modelos pré-treinados em tarefas supervisionadas na ImageNet. Os códigos e melhores modelos treinados estarão disponíveis no [github](#)<sup>1</sup>.

---

<sup>1</sup> <<https://github.com/fabiofelix/Sound-Self-Supervised>>

Figura 29 – Apresentação de estratégias para inicialização dos pesos de uma RNA e abordagens para avaliação dos resultados da rede.



Fonte: Elaborada pelo autor.

## 7.1 Metodologia aplicada

Em resumo, o procedimento executado para avaliar a aplicação de autossupervisão em diferentes arquiteturas de rede está alicerçado no proposto no [Capítulo 4](#). Primeiro, os dados de treinamento listados na [Tabela 10](#) foram balanceados pela aplicação de técnicas de aumento de dados para sons. Essa base aumentada é usada como entrada para o refinamento dos modelos pré-treinados. Essas mesmas técnicas foram empregadas para criação das visões apresentadas aos modelos autossupervisionados. O segundo passo gera uma base para comparação, treinando modelos com pesos aleatórios ou pré-treinados em tarefas genéricas de classificação. No próximo passo as técnicas de autossupervisão são aplicadas às arquiteturas e seus pesos são refinados com uma tarefa específica de classificação de padrões de sons.

### 7.1.1 Conjunto de dados

Os arquivos e rótulos são os mesmos descritos na [Seção 6.1.1](#), com 3 segundos de duração, coletados em paisagens naturais, fornecidos por pesquisadores do LEEC<sup>2</sup> da UNESP/Rio Claro e explorados em outros trabalhos, como [Scarpelli, Ribeiro e Teixeira \(2021\)](#), [Hilasaca et al. \(2021\)](#) e [Hilasaca, Ribeiro e Minghim \(2021\)](#). Além deles, também são listados sons genéricos extraídos do Google AudioSet ([GEMMEKE et al., 2017](#)). Dessa maneira, a tarefa principal está relacionada com identificar as 15 classes dispostas na tabela.

<sup>2</sup> Spatial Ecology and Conservation Lab - LEEC. website: <<https://github.com/LEEClab>>



Os dados da [Tabela 10](#) foram divididos com amostragem estratificada das classes em treinamento (90%) e teste (10%) e durante o treinamento das redes, foi aplicada validação cruzada com  $k = 5$ , sendo que em cada iteração uma das partições foi considerada como subconjunto de validação. Assim sendo, para as tarefas de refinamento de classificadores de padrões sonoros, são usados os subconjuntos de treinamento, validação e teste, enquanto que nas tarefas auxiliares de autossupervisão testadas, são usados os subconjuntos de treinamento e validação.

### 7.1.2 Aumento de dados

O conjunto de arquivos descrito na [Tabela 10](#) foi aumentado tanto para melhorar o balanceamento das classes, evitando problemas associados ([JOHNSON; KHOSHGOFTAAR, 2019](#); [WANG; PEREZ \*et al.\*, 2017](#)), quanto para criar as visões necessárias para aplicação das técnicas de autossupervisão.

O procedimento aplicado para balanceamento das classes segue os passos previstos na [Seção 6.1.2](#), aplicando *pitch shifting*, *time stretching* e *noise addition*, mas com os parâmetros apresentados no [Quadro 3](#). Isso porque mudanças na amplitude do sinal não impactavam de maneira significativa nos resultados e as modificações de parâmetros incrementou os resultados em pelo menos 2 pontos percentuais (de 0,75 para 0,77 na ResNet-50).

Quadro 3 – Faixa de valores dos parâmetros das funções de aumento de dados. O *stretch* descarta o valor um.

	valores
<i>stretch</i> (fator)	de 0,7 a 1,3 com incrementos de 0,1
<i>pitch</i> (passos)	$[-12, -6, -3, 3, 6, 12]$
<i>noise</i> (dB)	de 2 a 12 com incrementos de 2

Fonte: Elaborada pelo autor.

No caso do Aprendizado Autossupervisionado, para cada uma das amostras das *batches* de treinamento ou validação, são criadas duas visões alternativas. Para isso, escolhe-se de maneira aleatória duas das funções e respectivos parâmetros descritos no [Quadro 3](#), que geram as modificações das amostras de áudio. Ao aplicar a validação cruzada, os arquivos originais são divididos em 3600 ( $k - 1$  partições) para treinamento e 900 para validação. Depois disso, as visões desses subconjuntos são geradas e apresentadas aos modelos. Portanto, em cada iteração da validação cruzada, existem 7200 arquivos para treinamento e 1800 para validação.

### 7.1.3 Espectrogramas

Para todas as amostras de 3 segundos, foram gerados mel-espectrogramas ( $256 \times 256$ ) em escala de tons de cinza, usando janela de Hanning de tamanho 2048, sobreposição de 75% (ou *hop length* de 25%), 128 bandas mel e com os eixos verticais das imagens escalados para metade da taxa de amostragem de cada arquivo de áudio. Tanto o tamanho quanto a sobreposição das janelas contribuem para criação de uma representação que possua boa resolução de frequência e de tempo para representar uma variedade maior de padrões. As funções usadas são da biblioteca *librosa*<sup>3</sup> (v0.8.1).

### 7.1.4 Arquiteturas das redes neurais

Foram usadas 4 arquiteturas de rede: uma proposta por Salamon e Bello (2017) e nomeada aqui como SimpleCNN, composta por três camadas convolucionais com 24, 48 e 48 filtros de dimensões  $5 \times 5$ , respectivamente. Essas convoluções são seguidas por camadas de *max pooling* (exceto a última convolução) com fator  $4 \times 2$  e ativação ReLU. Depois de uma camada *flatten*, existem duas camadas densas, uma com 64 unidades de ativação ReLU e a outra com ativação *softmax* e a quantidade de classes especificada. Essas camadas também possuem *dropout* com fator 0,5 antes de suas entradas e normalização  $L^2$ , com fator  $10^{-3}$ . As outras arquiteturas são uma MobileNet-V3 (Large) (HOWARD *et al.*, 2019), uma ResNet-50 (HE *et al.*, 2016) e uma Inception-V3 (SZEGEDY *et al.*, 2016), todas pré-treinadas com a ImageNet (DENG *et al.*, 2009), quando necessário. A ResNet-50 é uma arquitetura comum para classificação de espécies animais (HARVEY, 2018; LEBIEN *et al.*, 2020; THOMAS *et al.*, 2019), a Inception possui uma estrutura com filtros em paralelo de tamanhos distintos, o que varia a largura da rede e pode facilitar o aprendizado de padrões de diferentes tamanhos, a MobileNet é um modelo construído para consumir poucos recursos, o que facilita sua aplicação em dispositivos móveis e a SimpleCNN apresentou bons resultados na classificação de pássaros.

Para a ResNet-50 e Inception-V3, as duas últimas camadas foram substituídas por uma *global average pooling* e uma camada densa com ativação *softmax* e a quantidade específica de classes. Para a MobileNet-V3, as quatro últimas camadas foram substituídas por uma camada convolucional, com filtros  $1 \times 1$  de quantidade igual à quantidade de classes de treinamento, ativação linear e precedida por um *dropout* de fator 0,2; uma camada *flatten*; e uma ativação *softmax*. Essas configurações são as mesmas contidas nas bibliotecas usadas para implementação dos experimentos.

Para o treinamento, tanto a SimpleCNN quanto a MobileNet-V3 usaram otimizador SGD, com taxa de aprendizagem  $10^{-2}$ , mas o treinamento da MobileNet-V3 também considera *momentum* = 0,9. Enquanto isso, para a ResNet-50 é aplicado o otimizador

<sup>3</sup> <<https://librosa.github.io/librosa/>>

Adam com taxa  $10^{-4}$ , e para Inception-V3, RMSProp com taxa  $10^{-3}$ . Em todos os casos de treinamento, foram executadas 100 épocas com *batch* = 80, exceto para a ResNet-50 que possui *batch* = 30. Na autossupervisão, foram aplicadas essas mesmas configurações, com exceção do valor de *batch*: 50 para SimpleCNN, MobileNet-V3 e Inception-V3; 30 para ResNet-50, que são usados tanto para o pré-treinamento quanto para refinamento dos pesos. Todas as configurações de otimização foram realizadas a partir dos melhores resultados retornados por cada modelo e os tamanhos de *batch* estão relacionados com a capacidade máxima da memória de vídeo utilizada. As imagens de entrada das redes foram normalizadas com *max-norm*, dividindo seus valores por 255.

Para que fosse possível a comparação da SimpleCNN com as demais arquiteturas, ela foi pré-treinada em uma versão reduzida da ImageNet, que considerou  $\approx 28\%$  (358.727 amostras) dos dados de treinamento, para evitar problemas de generalização do modelo, e manteve as quantidades de validação (50.000 amostras) e teste (100.000 amostras). Dessa maneira, os subconjuntos usados têm uma proporção de  $\approx 70,5\%$  (treinamento),  $\approx 9,8\%$  (validação) e  $\approx 19,7\%$  (teste). Esse treinamento foi executado seguindo as mesmas configurações de otimização e épocas apresentadas no parágrafo anterior para esse modelo, mas com *batch* = 256 por causa da memória de vídeo usada.

Todos os modelos foram implementados em Python (v3.8.10), associado com as bibliotecas Keras<sup>4</sup> (v2.7.0) e TensorFlow<sup>5</sup> (v2.7.0).

### 7.1.5 Tarefas de autossupervisão

As tarefas de autossupervisão testadas são a Barlow Twins e a VICReg descritas na Seção 2.4.5. De maneira resumida, arquiteturas com dois ramos que compartilham pesos, formados por um *encoder* e um *projector* (ou *expander*), geram espaços de características com  $N$  dimensões que são processadas pelas respectivas funções de custo. Diferente de testes reportados por Bardes, Ponce e Lecun (2022) para classificação de sons, os ramos possuem a mesma estrutura e processam dados de mesma origem. Nesse cenário, as entradas desses ramos são os mel-espectrogramas das visões criadas com as técnicas da Seção 7.1.2 e os pesos das arquiteturas foram inicializados de maneira aleatória. As funções de custo de cada uma das tarefas foram configuradas com os valores de parâmetros definidos nos seus respectivos artigos: para a Barlow Twins  $\lambda = 0,005$ ; e para VICReg  $\lambda = \mu = 25$ ,  $\nu = \gamma = 1$  e  $\epsilon = 10^{-4}$ .

Para o *encoder*, são avaliados resultados com as quatro arquiteturas mencionadas na seção anterior, sendo que a SimpleCNN é configurada até sua camada *flatten*, a ResNet-50 e a Inception-V3 até a camada *global average pooling*, enquanto que a MobileNet-V3 também usa essa camada de *pooling* no lugar suas 4 últimas camadas. Enquanto isso,

<sup>4</sup> <<https://keras.io/>>

<sup>5</sup> <<https://www.tensorflow.org/>>

cada uma das 3 camadas densas do *projector* possui  $N = 512$  unidades, sendo que as duas primeiras são seguidas por uma camada de *batch normalization* e ativação ReLU, enquanto a última possui apenas a ativação linear.

Após o pré-treinamento, o *encoder* é extraído e configurado da mesma maneira que na Seção 7.1.4, sendo que os pesos das camadas adicionadas são aleatórios, enquanto que todos os pesos aprendidos na autossupervisão são refinados. Como esse pré-treinamento é executado com validação cruzada, onde  $k - 1$  partições são usadas para treinamento e uma partição para validação, para refinamento foi utilizado o modelo que obteve menor valor da função de custo na partição de validação.

### 7.1.6 Avaliação

Para avaliação dos modelos refinados, foi empregada a acurácia balanceada disponível na biblioteca *scikit-learn*<sup>6</sup> (v1.0.1) (PEDREGOSA *et al.*, 2011) do Python. Os erros das funções de custo da autossupervisão também foram analisados, tanto nos dados de treinamento quanto de validação. Também foram executados testes  $t$  de Student com  $\alpha = 0,05$ , para comparar os resultados dos classificadores gerados. Nesses testes, a hipótese nula considera que as médias dos resultados comparados são iguais ou similares, consequentemente para a hipótese alternativa essas médias são diferentes (teste bicaudal); como a comparação é executada em resultados no mesmo conjunto de dados, existindo uma dependência entre as médias testadas, o teste é pareado.

Para todas as redes, as sementes dos métodos aleatórios do Python foram inicializadas (1030), seguindo o descrito pelo FAQ da biblioteca Keras<sup>7</sup>. Como descrito nas seções anteriores, a validação cruzada usou  $k = 5$ , tanto para o pré-treinamento quanto para o refinamento dos modelos. No caso do pré-treinamento os modelos foram avaliados e selecionados quanto ao erro da função de custo nos dados de validação e a para o refinamento dos modelos a média e o desvio padrão da acurácia balanceada nos dados de teste foram usadas para avaliação da classificação.

Para comparação com os resultados da autossupervisão, os modelos foram treinados para identificar 15 classes (cf. Seção 7.1.1), em duas situações: na primeira, todos os modelos foram treinados com pesos aleatórios; e na segunda, com pesos aprendidos na ImageNet. As análises de acurácia balanceada e de visualização levaram em consideração apenas as 12 classes relacionadas às espécies de interesse (cf. Tabela 10) para facilitar a comparação com os demais capítulos.

Também foram consideradas duas estratégias de avaliação que modifica as quantidades de amostras empregadas nos treinamentos. Na primeira, uma amostragem estratificada das classes é aplicada para gerar dois subconjuntos dos dados de treinamento

<sup>6</sup> <<https://scikit-learn.org/stable/>>

<sup>7</sup> <[https://keras.io/getting\\_started/faq/](https://keras.io/getting_started/faq/)>

da [Tabela 10](#), um com 50% e outro com 20% desses dados. Esses subconjuntos são usados para refinar os modelos pré-treinados com as abordagens de autossupervisão. Nesses dois casos, quando o aumento de dados é executado, as amostras das classes são aumentadas para 290 e 116, respectivamente, refletindo as quantidades de amostras da classe majoritária nas partições de treinamento. Na outra abordagem, a quantidade de amostras para pré-treinamento é duplicada, adicionando outras 4500 amostras não rotuladas aos dados de treinamento da [Tabela 10](#), também provenientes das gravações da UNESP/Rio Claro. Dessa maneira, os modelos são pré-treinados com essa nova base e refinados com os dados da [Tabela 10](#), além dos subconjuntos reduzidos, mencionados antes. Com isso, é possível verificar o comportamento dos modelos com mais amostras para pré-treinamento e com variação da quantidade de amostras para o refinamento na tarefa de classificação.

Assim como na [Seção 5.1.7](#), os espaços de características aprendidos pelas redes foram avaliados por meio do Coeficiente de Silhueta ([TAN; STEINBACH; KUMAR, 2005](#)) e de visualizações com a projeção t-SNE ([MAATEN; HINTON, 2008](#)), por facilitar a identificação das vizinhanças dos dados e a separação entre as classes ([NONATO; AUPETIT, 2018](#)). Para isso, foram extraídas as características aprendidas pela penúltima camada de todas as redes (antes da última camada densa), após o refinamento dos modelos, porque nessa camada estão as características aprendidas, usadas para a classificação executada pela última camada. As rotinas empregadas para silhueta e projeção também estão disponíveis no scikit-learn.

Neste capítulo, também são gerados mapas de calor para cada modelo, representando a similaridade CKA (cf. [Seção 2.4.3.3](#)) entre as suas camadas convolucionais, depois do refinamento dos modelos. No caso da SimpleCNN, por possuir menos convoluções, as similaridades foram calculadas entre todas as suas camadas, exceto a camada *flatten*.

Por fim, a maioria dos treinamentos e testes das redes foram executados com uma placa de vídeo NVidia Titan XP, com driver v470.86, Cuda v11.2.152 e cuDNN v8.1.0. No caso do pré-treinamento da SimpleCNN na classificação da ImageNet, foi usada uma placa de vídeo NVidia RTX A5000, com driver v470.74, Cuda v11.4 e cuDNN v8.1.0, mesma configuração considerada para os testes da [Seção 7.2.4](#), que varia o valor  $N$  do espaço gerado pelo *projector*.

## 7.2 Resultados

Esta seção descreve os resultados experimentais dos testes com Aprendizado Autossupervisionado. Os resultados dos modelos pré-treinados em tarefas auxiliares de autossupervisão foram comparados com os resultados dos mesmos modelos treinados com pesos aleatórios e com pesos aprendidos em tarefas de classificação da ImageNet, avaliados por meio de acurácia balanceada  $\in [0, 1]$ , Coeficiente de Silhueta  $\in [-1, 1]$ , mapas de calor

da similaridade  $CKA \in [0, 1]$  e projeção t-SNE. Nesse cenário, foram testadas duas tarefas autossupervisionadas: Barlow Twins e VICReg; e quatro arquiteturas de redes para serem refinadas: SimpleCNN, MobileNetV3, ResNet50 e InceptionV3, todas usando mel-espectrograma como entrada. Todos os testes foram executados com validação cruzada, com  $k = 5$ .

A Tabela 15 e a Figura 30 resumizam os resultados dos testes. Ao analisar os valores de cada coluna de acurácia da tabela e comparar os resultados da ResNet50 com os da InceptionV3, verifica-se que a hipótese nula do teste  $t$  de Student não foi rejeitada ( $p$ -valor  $> 0,05$ ), o que significa similaridade entre as médias dos resultados desses modelos. Além disso, os resultados desses modelos são sempre superiores aos outros dois modelos, normalmente com mais de 3 pontos percentuais de diferença. Em todas as situações com pesos pré-treinados, os valores de silhueta dos espaços aprendidos pela ResNet50 são superiores aos demais modelos, estando entre  $[0, 10; 0, 23]$ , enquanto que os demais modelos não ultrapassam 0,15. No cenário com inicialização aleatória dos pesos, a InceptionV3 obteve os melhores resultados de silhueta e acurácia.

Tabela 15 – Resultados dos modelos treinados com pesos aleatórios, pré-treinados na ImageNet e pré-treinados com tarefas de autossupervisão. Tanto o Coeficiente de Silhueta quanto a média e o desvio padrão da acurácia balanceada são referentes aos dados de teste. A silhueta foi calculada a partir do modelo treinado na partição de validação cruzada com melhor acurácia balanceada. Os valores destacados são os melhores resultados da tabela.

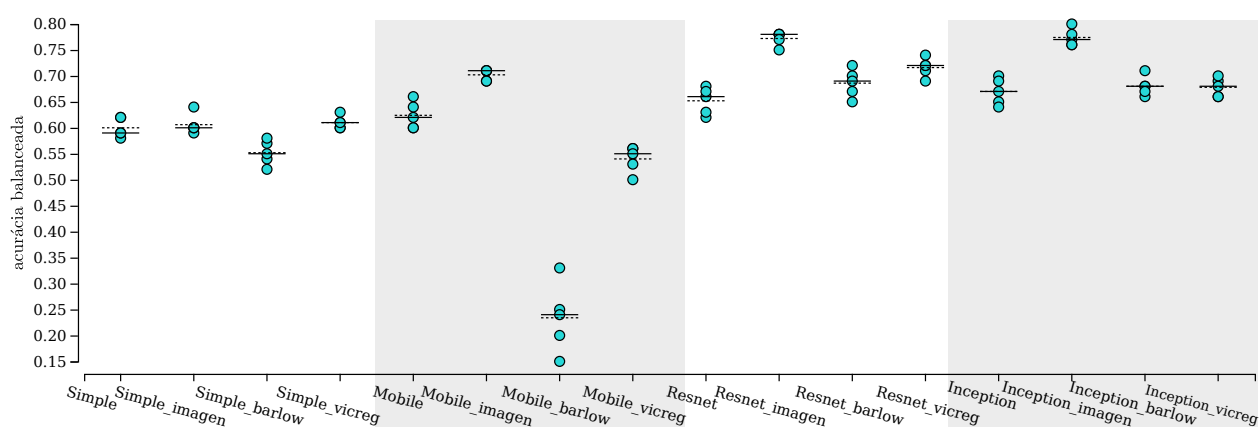
	Aleatório		ImageNet	
	silhueta	acurácia	silhueta	acurácia
SimpleCNN	0,0065	0,60 $\pm$ 0,02	0,0340	0,61 $\pm$ 0,02
MobileNet-V3	0,0863	0,62 $\pm$ 0,03	0,1500	0,70 $\pm$ 0,01
ResNet-50	0,0769	0,65 $\pm$ 0,03	<b>0,2240</b>	<b>0,77<math>\pm</math> 0,01</b>
Inception-V3	0,1150	0,67 $\pm$ 0,03	0,1303	0,77 $\pm$ 0,02
	Barlow Twins		VICReg	
	silhueta	acurácia	silhueta	acurácia
SimpleCNN	-0,0031	0,55 $\pm$ 0,02	0,0142	0,61 $\pm$ 0,01
MobileNet-V3	-0,1213	0,23 $\pm$ 0,07	0,0085	0,54 $\pm$ 0,03
ResNet-50	0,1041	0,69 $\pm$ 0,03	0,1374	0,72 $\pm$ 0,02
Inception-V3	0,0451	0,68 $\pm$ 0,02	0,0109	0,68 $\pm$ 0,02

Fonte: Elaborada pelo autor.

Ao verificar as linhas da tabela para MobileNetV3, ResNet50 e InceptionV3, os pré-treinamentos com Barlow Twins e VICReg reduziram os valores de silhueta ( $< 0, 14$ ) e acurácia balanceada ( $\leq 0, 72$ ), quando comparados com os modelos inicializados com pesos da ImageNet (silhueta  $< 0, 23$  e acurácia  $\leq 0, 77$ ). Ao comparar as tarefas de autossupervisão com os pesos aleatórios, os resultados da MobileNetV3 sempre são inferiores aos pesos aleatórios, enquanto que os da ResNet50 e da InceptionV3 são similares (hipó-

tese nula não rejeitada com  $p\text{-valor} > 0,05$ ) ou maiores do que os alcançados com pesos aleatórios. O caso da Barlow Twins com MobileNetV3 como *encoder* apresenta a maior redução de silhueta ( $< 0,12$ ) e acurácia ( $\leq 0,23$ ). No caso da SimpleCNN, enquanto a Barlow Twins reduziu os resultados, a VICReg manteve a acurácia balanceada estável, com um pequeno incremento da silhueta, quando comparada com o modelo inicializado com pesos aleatórios. Comparando os resultados da Barlow Twins com a VICReg, exceto para InceptionV3, a VICReg incrementa os resultados em mais de 3 pontos percentuais. Essas diferenças e similaridades entre os resultados de cada modelo ficam mais visíveis ao analisar a [Figura 30](#).

Figura 30 – Gráfico de valores individuais da acurácia balanceada gerado pelas CNNs aplicadas aos dados de teste. Cada área do gráfico (brancas e cinzas) mostra os resultados de um dos modelos. Dentro de cada área, os resultados são apresentados na seguinte ordem, da esquerda para a direita: modelo treinado com pesos aleatórios, modelo pré-treinado na ImageNet e pré-treinado com tarefas de autossupervisão. As linhas sólidas e tracejadas dentro de cada grupo de valores representam, respectivamente, sua mediana e a média. Existe sobreposição entre pontos, por causa disso os resultados de alguns modelos apresentam quantidades diferentes de pontos.



Fonte: Elaborada pelo autor.

### 7.2.1 Análise da estrutura de similaridade das redes

A [Figura 31](#) apresenta mapas de calor com o padrão de similaridade CKA entre as camadas das redes para comparação e avaliação das estruturas aprendidas com as técnicas de autossupervisão. Esses padrões variam de acordo com o modelo e com o tipo de inicialização de seus pesos. Para a SimpleCNN, modificações surgem sobretudo em torno da sua quinta camada, sendo que o pré-treinamento executado com a Barlow Twins obtém os valores de similaridade mais elevados ( $\approx 1,0$ ), o que pode estar relacionado com a redução dos resultados de acurácia em 5 pontos percentuais.

Para a MobileNetV3, as diferenças mais visíveis surgem ao comparar os modelos pré-treinados com o modelo inicializado com pesos aleatórios, sendo que as principais diferenças são apresentadas em torno da diagonal principal do mapa de calor, como pode



ser visto próximo da camada 5, entre as camadas 15 e 25, e acima da camada 35. As modificações dessas similaridades são mais drásticas quando o modelo é inicializado com os pesos aprendidos na Barlow Twins, o que gerou resultados inferiores a 0,25 de acurácia balanceada.

No caso da ResNet50, existe semelhança entre os resultados com pesos aleatórios e os resultados com tarefas autossupervisionadas, onde as similaridades entre as camadas possuem os maiores valores na maior parte dos mapas, o que força os resultados desses modelos a estarem em torno de 0,68 de acurácia balanceada, diferente do que ocorre com o pré-treinamento com a ImageNet.

Todos os mapas da InceptionV3 apresentam semelhanças e variações sutis em torno da diagonal, como pode ser percebido nas variações até a camada 10 e acima da camada 80. As semelhanças mantêm os resultados de acurácia abaixo de 0,70 para as tarefas de autossupervisão e as pequenas modificações com o pré-treinamento na ImageNet eleva a acurácia para 0,77.

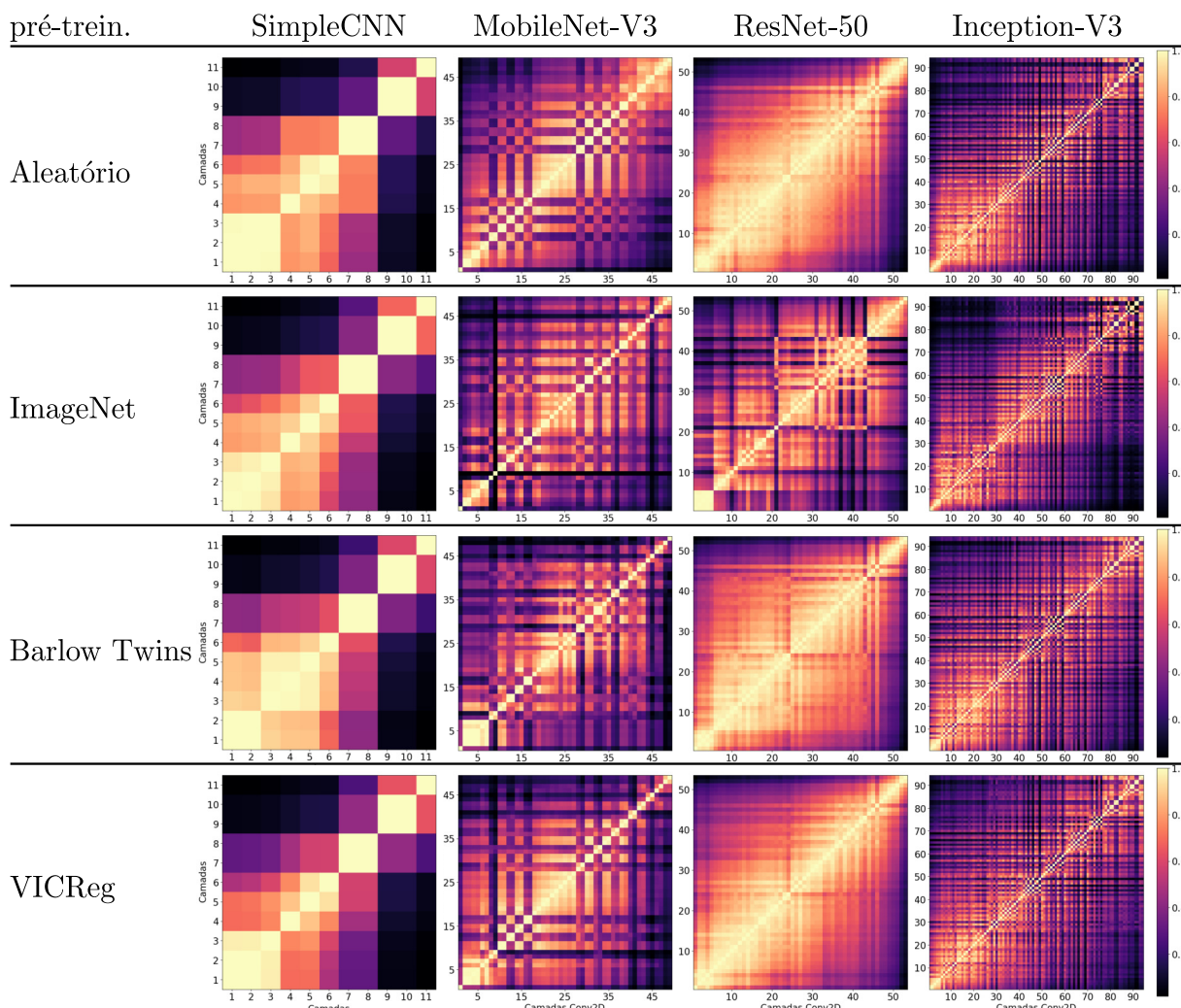
### 7.2.2 Análise visual dos espaços de características aprendidos

A [Figura 32](#) e a [Figura 33](#) exibem projeções t-SNE para inspeção visual dos espaços de características aprendidos, tanto para modelos treinados com pesos aleatórios quanto pré-treinados na ImageNet ou com autossupervisão. No geral, o que mais influencia a segregação das classes é a profundidade dos modelos, sendo que a SimpleCNN demonstra maior confusão visual entre as classes, seguida pela MobileNetV3, o que reflete os valores do Coeficiente de Silhueta da [Tabela 15](#). Os baixos resultados da MobileNetV3 com a Barlow Twins, com silhueta negativa e acurácia balanceada de 0,23, são representados pelo alto grau de sobreposição da respectiva projeção. A maior separação entre as classes surge na segunda linha de projeções, ratificando os resultados superiores da [Tabela 15](#), além dessa segregação ser mais difícil de perceber com modelos inicializados com pesos aleatórios ou conforme a autossupervisão é aplicada. Também é possível identificar que a região central das projeções possui maior sobreposição das classes.

Como complemento, a [Figura 33](#) apresenta as mesmas projeções, mas identificando dois grupos de padrões: pássaros e anuros. Dessa maneira, é possível identificar que independente da inicialização dos pesos, os modelos conseguem distinguir entre esses dois grupos de amostras, sendo que a inicialização com pesos aprendidos na ImageNet alcança a maior segregação visual. Conforme as tarefas de autossupervisão são empregadas, as fronteiras dos dois grupos se aproximam, tendendo para os resultados com pesos aleatórios, o que impacta nos resultados gerais de classificação da [Tabela 15](#).



Figura 31 – Similaridade CKA dos modelos treinados com pesos aleatórios, pré-treinados na ImageNet e pré-treinados com tarefas de autossupervisão. Cada modelo considerado corresponde à respectiva partição da validação cruzada com melhor acurácia balanceada nos dados de teste. As similaridades são calculadas entre as camadas convolucionais de cada modelo, exceto a SimpleCNN na qual o cálculo é executado para todas as suas camadas.

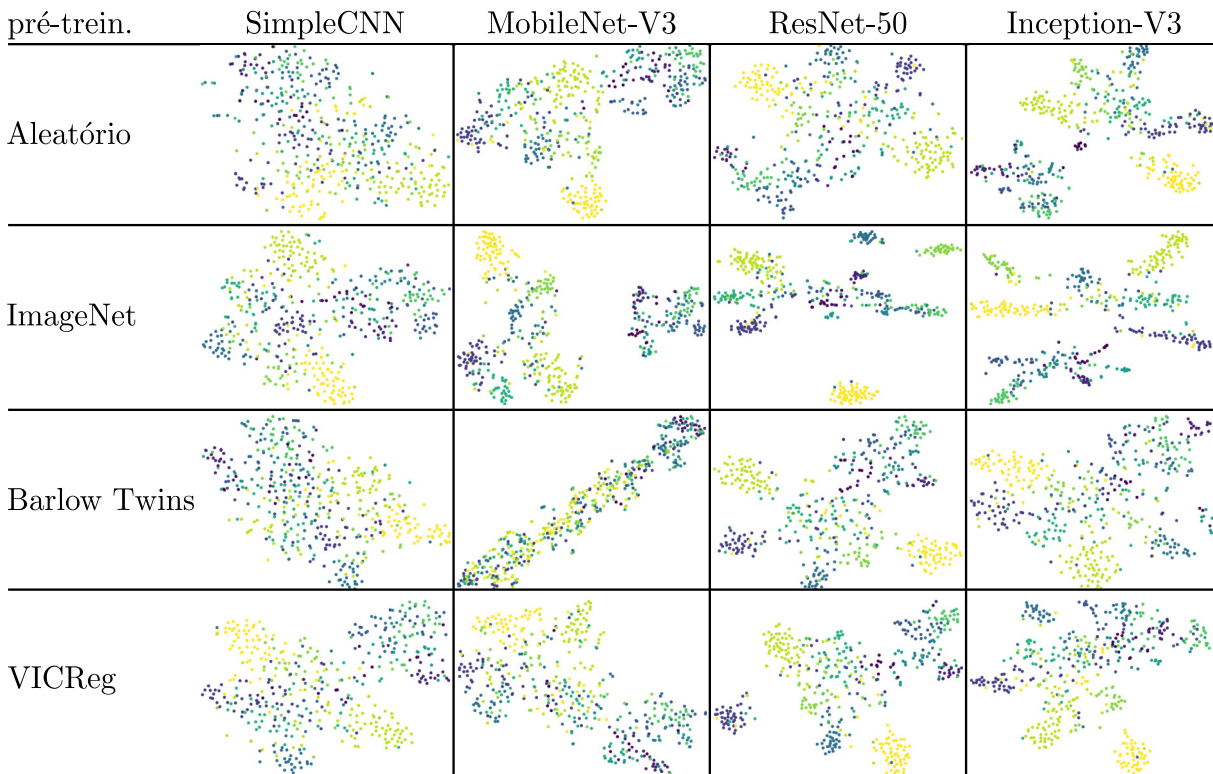


Fonte: Elaborada pelo autor.

### 7.2.3 Variação das quantidades de amostras

A Tabela 16 apresenta o refinamento dos modelos pré-treinados com subconjuntos dos dados apresentados na Tabela 10. Ao comparar os resultados da coluna 100% (quantidade original de amostras) com as colunas 50% e 20% para cada técnica de autossupervisão, a hipótese nula do teste  $t$  de Student é rejeitada ( $p$ -valor  $\leq 0,05$ ), evidenciando uma diferença significativa entre os resultados das colunas comparadas. As únicas exceções aparecem com a SimpleCNN e a MobileNetV3 no pré-treinamento com a Barlow Twins e coluna 50%. Nesse caso, todas as diferenças indicam redução dos resultados quando o tamanho da base de refinamento diminui.

Figura 32 – Projeção t-SNE dos dados de teste, com características extraídas da penúltima camada de cada modelo após o seu treinamento. As linhas da imagem apresentam modelos treinados com pesos aleatórios, pré-treinados na ImageNet e pré-treinados com tarefas de auto supervisão. Cada modelo considerado corresponde à respectiva partição da validação cruzada com melhor acurácia balanceada nos dados de teste. As cores dos pontos representam as classes de interesse.



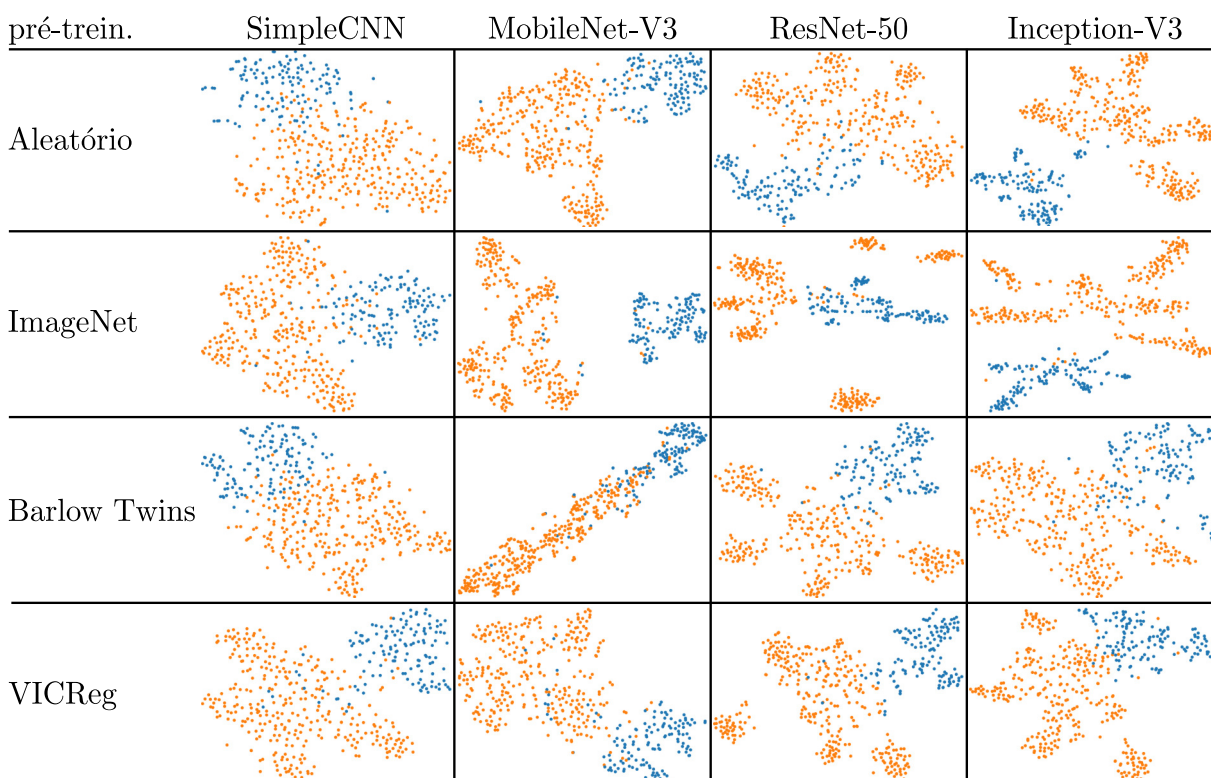
Fonte: Elaborada pelo autor.

Quando são comparadas as colunas da Barlow Twins com as respectivas colunas da VICReg, também existem diferenças entre os resultados, exceto para a coluna 100% (InceptionV3) e a coluna 50% (SimpleCNN, ResNet50 e InceptionV3). Nessa comparação, os resultados da VICReg são iguais ou maiores do que os da Barlow Twins, sendo menos impactados pela redução da quantidade de amostras.

Para a Tabela 17, o tempo médio de pré-treinamento com o dobro da quantidade de amostras é de 3 dias para cada modelo, enquanto que no pré-treinamento original esse tempo é próximo de 24 h. No comparativo entre as quantidades de amostra para refinamento e entre as respectivas colunas da Barlow Twins e da VICReg, as semelhanças e conclusões seguem praticamente o mesmo padrão do pré-treinamento com os dados originais da Tabela 10. A diferença surge na Tabela 17 ao comparar as colunas da Barlow Twins com as respectivas colunas da VICReg, nas quais não existem diferenças entre as colunas 100% (ResNet50 e InceptionV3) e a coluna 50% (InceptionV3).

Ao comparar os resultados da Tabela 16 e da Tabela 17, os resultados da Barlow Twins na segunda tabela são diferentes da primeira (hipótese nula rejeitada com p-valor

Figura 33 – Projeção t-SNE dos dados de teste, com características extraídas da penúltima camada de cada modelo após o seu treinamento. As linhas da imagem apresentam modelos treinados com pesos aleatórios, pré-treinados na ImageNet e pré-treinados com tarefas de autossupervisão. Cada modelo considerado corresponde à respectiva partição da validação cruzada com melhor acurácia balanceada nos dados de teste. As cores dos pontos representam **pássaros e anuros**.



Fonte: Elaborada pelo autor.

$\leq 0,05$ ), exceto para a coluna 100% (ResNet50 e InceptionV3). No caso da VICReg, o padrão é invertido, na maior parte dos casos os resultados da segunda tabela são similares aos da primeira (hipótese nula **não** rejeitada com p-valor  $> 0,05$ ), exceto para a coluna 50% (SimpleCNN, MobileNetV3 e ResNet50) e para a coluna 20% (ResNet50). Com isso, o aumento dos dados para pré-treinamento possui maior influência nos modelos pré-treinados com Barlow Twins. As variações para esse pré-treinamento são na maior parte dos casos positivas, por exemplo, ao comparar os resultados da ResNet50 para a coluna 100%, existe um aumento de 2 pontos percentuais na [Tabela 17](#), enquanto que para a MobileNetV3, na coluna 20%, existe um aumento de 12 pontos percentuais. Enquanto isso, as variações para a VICReg são na maior parte dos casos negativas, por exemplo, ao comparar os resultados da ResNet50 para a coluna 20%, existe uma redução de 5 pontos percentuais na [Tabela 17](#).

Tabela 16 – Resultados dos modelos **refinados** com todos os dados da Tabela 10 (100%) e com subconjuntos de 50% e 20% desse dados.

	Barlow Twins			VICReg		
	100%	50%	20%	100%	50%	20%
SimpleCNN	0,55± 0,02	0,53± 0,03	0,41± 0,03	0,61± 0,01	0,55± 0,01	0,47± 0,02
MobileNet-V3	0,23± 0,07	0,15± 0,06	0,11± 0,03	0,54± 0,03	0,44± 0,03	0,29± 0,02
ResNet-50	0,69± 0,03	0,59± 0,02	0,38± 0,03	0,72± 0,02	0,60± 0,02	0,43± 0,02
Inception-V3	0,68± 0,02	0,56± 0,02	0,48± 0,02	0,68± 0,02	0,59± 0,03	0,51± 0,02

Fonte: Elaborada pelo autor.

Tabela 17 – Modelos **pré-treinamentos** com todos os dados da Tabela 10 mais 4500 amostras não rotuladas, gravadas pelo mesmo grupo de pesquisa. Os resultados dos modelos refinados com todos os dados da Tabela 10 (100%) e com subconjuntos de 50% e 20% desse dados.

	Barlow Twins			VICReg		
	100%	50%	20%	100%	50%	20%
SimpleCNN	0,08± 0,00	0,08± 0,00	0,08± 0,00	0,59± 0,03	0,54± 0,01	0,46± 0,02
MobileNet-V3	0,32± 0,08	0,27± 0,03	0,23± 0,01	0,51± 0,01	0,40± 0,01	0,28± 0,02
ResNet-50	0,71± 0,01	0,61± 0,01	0,44± 0,02	0,72± 0,02	0,64± 0,01	0,48± 0,02
Inception-V3	0,71± 0,03	0,61± 0,03	0,52± 0,01	0,69± 0,02	0,61± 0,04	0,50± 0,02

Fonte: Elaborada pelo autor.

### 7.2.4 Variação do espaço gerado pela autossupervisão

Tanto o artigo da Barlow Twins (ZBONTAR *et al.*, 2021) quanto da VICReg (BARDES; PONCE; LECUN, 2022) afirmam que o aumento do tamanho  $N$  do espaço de características gerado pelo *projector* (ou *expander*) influencia de maneira positiva os resultados. Por causa disso, além dos testes reportados até aqui com  $N = 512$ , foram testados os valores 1024, 2048, 4096 e 8192, sendo este último o tamanho padrão dos artigos, por possuir os melhores resultados.

Os testes foram executados com a VICReg e ResNet50, por apresentarem os melhores resultados na Tabela 15. As configurações do modelo e do treinamento são as mesmas listadas na Seção 7.1, exceto pelo tamanho de *batch* do pré-treinamento que foi configurado para 90, sendo que para  $N = 8192$  esse tamanho é igual a 80, por causa da memória disponível na placa de vídeo. No refinamento dos modelos, o tamanho de *batch* foi reduzido para 30 em todos os casos, também por causa da memória disponível.

Os resultados de acurácia balanceada descritos na Tabela 18 são similares aos reportados na Tabela 15, no mesmo cenário com VICReg, visto que a hipótese nula do teste  $t$  de Student não foi rejeitada ( $p - \text{valor} > 0,05$ ). Ao comparar as duas primeiras linhas



da tabela, mesmo não sendo significativo, é possível perceber uma redução da acurácia balanceada e do Coeficiente de Silhueta. Da mesma maneira, a variação do tamanho  $N$  do espaço de características criado para aplicação da função de custo, não proporcionou modificações significativas dos resultados.

Tabela 18 – Resultados de ResNet-50, pré-treinada com VICReg variando o tamanho  $N$  do espaço. Tanto o Coeficiente de Silhueta quanto a média e o desvio padrão da acurácia balanceada são referentes aos dados de teste. A silhueta foi calculada a partir do modelo gerado na partição de validação cruzada com melhor acurácia balanceada. O resultado da primeira linha corresponde à [Tabela 15](#).

#N	Silhueta	Acurácia
512	0,1374	0,72 $\pm$ 0,02
512	0,1142	0,70 $\pm$ 0,02
1024	0,1142	0,69 $\pm$ 0,02
2048	0,1093	0,70 $\pm$ 0,02
4096	0,0968	0,68 $\pm$ 0,02
8192	0,1079	0,69 $\pm$ 0,02

Fonte: Elaborada pelo autor.

## 7.3 Discussão

De maneira geral, os resultados das redes inicializadas com pesos aleatórios ou com pesos aprendidos na ImageNet são os limites inferior (média de acurácia balanceada  $\in [0,60;0,67]$ ) e superior (média de acurácia balanceada  $\in [0,61;0,77]$ ) dos resultados. Com isso, a depender do modelo usado como *encoder*, os resultados da Barlow Twins estão próximos dos pesos aleatórios, com acurácia balanceada máxima de 0,69, enquanto que a VICReg alcança resultado igual a 0,72. Esses comportamentos acompanham os descritos por [Bardes, Ponce e Lecun \(2022\)](#), onde a VICReg obteve resultados similares ou superiores aos da Barlow Twins, evidenciando a capacidade daquela para aprender espaços de características mais robustos. Além disso, mesmo com uma quantidade reduzida de amostras, 4500 de 15 classes de imagens de padrões sonoros, a VICReg gerou modelos capazes de obter resultados com até 5 pontos percentuais de diferença dos modelos pré-treinados em tarefas de classificação na base de imagens ImageNet, que possui mais de um milhão de amostras de 1000 classes de imagens genéricas. Mesmo assim, variações da quantidade de amostras de pré-treinamento são necessárias para avaliar os impactos nos resultados.

A aplicação da Barlow Twins e da VICReg impacta de maneiras diferentes nos resultados e na estrutura interna aprendida por modelos com profundidade e largura diferentes. Por exemplo, ao verificar a [Figura 31](#), modificações sutis da MobileNetV3

levaram a grandes variações dos resultados de acurácia (de 0,70 com ImageNet para 0,23 com Barlow Twins). Por outro lado, variações expressivas como as apresentadas para a ResNet50 impactaram menos nos resultados (de 0,77 com ImageNet para 0,72 com VICReg). Esse tipo de comportamento pode estar relacionado com variações específicas da similaridade de grupos particulares de camadas, mas constatações sobre isso demandam uma quantidade maior de testes para compreender essas relações.

A capacidade dos modelos em diferenciar pássaros de anuros, como aparece na [Figura 33](#), destaca que a maior dificuldade para as arquiteturas testadas está relacionada com discriminar entre as classes de pássaros e entre as classes de anuros. Essa dificuldade está relacionada com a sobreposição das classes, percebida sobretudo na região central das projeções da [Figura 32](#). Isso faz com que nenhuma das arquiteturas ou abordagens de inicialização alcance médias de acurácia balanceada acima 0,77 ou Coeficiente de Silhueta maior do que 0,23.

A redução da quantidade de amostras para refinamento impacta as duas abordagens de autossupervisão, sendo que os resultados da VICReg  $\in [0,29;0,60]$  são iguais ou maiores do que os da Barlow Twins  $\in [0,11;0,59]$ , quando a quantidade é reduzida para 50% ou 20% das amostras. O mesmo padrão acontece quando a base de pré-treinamento é duplicada, VICReg  $\in [0,28;0,64]$  e Barlow Twins  $\in [0,08;0,61]$ . Esse comportamento da VICReg pode estar relacionado com a capacidade da sua função de custo de aprender pesos mais significativos para inicialização do modelo para refinamento. Por outro lado, a duplicação da base para pré-treinamento apresenta maior impacto na Barlow Twins, chegando a melhorar em 12 pontos percentuais os valores de colunas equivalentes das tabelas de resultados, enquanto que o impacto na VICReg foi menor ou negativo, chegando a reduzir os resultados em 5 pontos percentuais. Além disso, a duplicação da base para pré-treinamento aumenta o tempo de processamento em aproximadamente 3x.

A variação do tamanho do espaço de características de saída da VICReg não influenciou de maneira significativa os resultados da tarefa de classificação. Conforme o valor de  $N$  aumenta, os resultados tendem a saturar, mantendo-se sem modificações relevantes. Esse comportamento difere do descrito por [Zbontar et al. \(2021\)](#) e [Bardes, Ponce e Lecun \(2022\)](#), onde um valor maior de  $N$  proporciona melhores resultados no refinamento dos modelos. Logo, mais testes são necessários para elucidar esse desempenho do modelo.

## 7.4 Considerações finais

Este capítulo descreveu os resultados obtidos a partir do pré-treinamento de CNNs com técnicas de autossupervisão, contemplando os testes de inicialização de pesos sugeridos no capítulo de proposta desta pesquisa. Foram testadas as abordagens Barlow Twins

e VICReg, ambas testadas com quatro arquiteturas de CNN, cobrindo também pontos levantados sobre profundidade e largura discutidos na proposta. As evidências empíricas sugerem que, no cenário proposto, a VICReg apresenta resultados superiores aos da Barlow Twins, inicializando modelos que convergem para resultados mais adequados, próximos do resultado de outras abordagens de pré-treinamento, como a utilização de tarefas supervisionadas fundamentadas na ImageNet.

Com isso, os resultados sugerem que é possível aplicar esse tipo de técnica para pré-treinamento das redes, mesmo com uma quantidade pequena de amostras de treinamento (menos de 1000 por classe), obtendo resultados acima de 0,70 de acurácia. Entretanto, ainda é necessário compreender melhor os impactos da variação dessa quantidade de amostras, tanto para pré-treinamento quanto para refinamento, do tamanho do espaço de características gerado para aplicação das funções de custo, do tamanho da *batch* para treinamento e até mesmo da utilização de ramos distintos com entradas diferentes, principalmente na VICReg. Além disso, a eficiência do pré-treinamento para bases maiores precisa ser melhorada. Essas questões devem nortear os trabalhos futuros, além da necessidade de estudar maneiras de romper uma barreira dos resultados que está em torno de 0,77 de acurácia balanceada.





---

## CONCLUSÕES

---

Esta pesquisa de doutorado teve como foco identificar espécies de animais por meio de características de seus sons aprendidas por Redes Neurais Convolucionais. Os principais desafios para essa identificação estão relacionados com as variações dos padrões sonoros e de suas intensidades, com ruídos diversos e com sobreposição de sons. Com isso, o próprio processo de treinamento das redes deve ser capaz de gerar modelos que lidem com essas questões, evidenciando que escolhas apropriadas podem produzir modelos mais robustos. Por causa disso, nenhuma etapa de pré-processamento, como seleção de faixas de frequência, filtragem para atenuação de ruídos ou separação de fontes sonoras, foi definida. Nesse caso, as etapas da pesquisa abordaram sobretudo *i)* a definição de uma função de custo apropriada, *ii)* estratégias para combinação de entradas e *iii)* abordagens para inicialização dos pesos. Dessas etapas, a combinação de entradas gerou os melhores resultados, sendo que os maiores impactos foram observados em arquiteturas com menos de 10 camadas. Além disso, no lugar de treinar as CNNs com grandes bases específicas de sons, os testes foram executados em dados com uma quantidade reduzida de amostras (menos de 1000 por classe), verificando como diferentes arquiteturas se comportam nesse cenário, quando associadas às etapas propostas. A partir da análise geral dos resultados dessas etapas, é possível constatar sua capacidade de gerar características apropriadas para a classificação dos padrões sonoros, mesmo que esses resultados sejam apenas similares ou apresentem incrementos sutis ao serem comparados com os de redes sem as mesmas etapas. Avaliações específicas, como de coesão ou segregação de espaços de características e de medidas relacionadas com matrizes de confusão, destacam a capacidade da proposta de melhorar os espaços aprendidos em torno de classes específicas. A aplicação das etapas também manteve o processo de treinamento estável, não degradando de maneira considerável os resultados dos modelos. Assim sendo, a proposta implementada alcançou resultados aceitáveis, podendo ser aplicada e refinada para a identificação de espécies de pássaros e anuros, enfrentando os cenários e desafios apresentados.

## 8.1 Contribuições

Seguindo as etapas abordadas, a primeira contribuição desta pesquisa está ligada à aplicação de métodos de quantificação como o *classify and count*, na regularização do treinamento das redes. Essa regularização foi capaz de aperfeiçoar a estrutura dos espaços de características aprendidos, impactar de maneira positiva nos valores de medidas específicas como a sensibilidade das classes, mesmo que de maneira sutil, e de manter o processo de treinamento estável.

Na sequência, está a análise de entradas das redes, como variações de espectrogramas, características acústicas, informações sobre a gravação e combinações dessas características. A partir dessas análises, verificou-se que uma combinação simples de imagens do espectro com características manuais, incrementou os resultados, em especial de arquiteturas rasas com menos de 10 camadas. Para isso, as características manuais foram melhor processadas por um ramo auxiliar das arquiteturas, com uma camada de *batch normalization*. Além disso, a adição de informações relacionadas com local e data de gravação melhoraram os resultados do *baseline*. A combinação de variações de espectrograma em três canais também foi capaz de gerar resultados consideráveis para modelos profundos e pré-treinados como a ResNet. No caso de entrada única, o mel-espectrograma foi capaz de representar de maneira aceitável os padrões de interesse, quando comparado com as demais representações de tempo-frequência testadas.

A terceira contribuição da pesquisa, é a avaliação de abordagens para inicialização dos pesos das redes. Com isso, indícios foram apresentados de que tarefas de autossupervisão como a VICReg são capazes de gerar resultados próximos aos obtidos por tarefas supervisionadas na ImageNet, mesmo utilizando uma quantidade menor de amostras para o processo de pré-treinamento e refinamento dos modelos.

Essas contribuições, abordadas em cada um dos capítulos de resultados, geraram três artigos que estão em diferentes fases de elaboração:

- **Capítulo 5:** A classification and quantification approach to generate features in soundscape ecology using neural networks (DIAS; PONTI; MINGHIM, 2021);
- **Capítulo 6:** Combination of input spectrogram representations and hand-crafted features in convolutional networks for natural sounds classification (em preparação para ressubmissão);
- **Capítulo 7:** Evaluation of self-supervised learning to identify bird and anuran species (em preparação para submissão).

O Quadro 4 retoma os trabalhos reportados no Capítulo 3, adicionando os capítulos de resultado da tese. Como isso, é possível visualizar sua integração com outros trabalhos

que aplicam redes neurais no cenário de classificação de espécies animais a partir de suas vocalizações. Como proposto no [Capítulo 4](#), a única característica que não é considerada nesses capítulos é o pré-processamento dos arquivos de som.

Quadro 4 – Extensão do [Quadro 1](#) com os capítulos apresentados nesta tese, resumindo suas principais características, relacionadas com pré-processamento dos dados (pré): filtragem ou seleção de frequências específicas; aumento de dados (aum.); tipos de entrada dos modelos: espectrograma, mel-espectrograma, PCEN ou alguma combinação de entradas (comb.); profundidade da arquitetura: rasa ou profunda; e inicialização dos pesos: aleatória (alea.) ou pesos pré-treinados (pré-trei.)

	pré	aum.	entrada				arquitetura		inicialização	
			espec.	mel	pcen	comb.	rasa	profunda	alea.	pré-trei.
(SALAMON <i>et al.</i> , 2017)		✓		✓			✓		✓	
(LOSTANLEN <i>et al.</i> , 2019)		✓		✓	✓	✓	✓		✓	
(CRAMER <i>et al.</i> , 2020)		✓			✓		✓		✓	
(KAHL <i>et al.</i> , 2021)	✓			✓				✓		✓
(STROUT <i>et al.</i> , 2017)			✓					✓		✓
(XIE <i>et al.</i> , 2022)	✓			✓		✓	✓		✓	
(LEBIEN <i>et al.</i> , 2020)				✓				✓	✓	
(HARVEY, 2018)					✓			✓	✓	
(THOMAS <i>et al.</i> , 2019)	✓		✓			✓		✓	✓	
(KIRSEBOM <i>et al.</i> , 2020)	✓		✓					✓	✓	
(SHIU <i>et al.</i> , 2020)			✓					✓	✓	
Capítulo 5		✓		✓			✓	✓	✓	✓
Capítulo 6		✓	✓	✓	✓	✓	✓	✓	✓	✓
Capítulo 7		✓		✓			✓	✓	✓	✓

Fonte: Elaborada pelo autor.

## 8.2 Limitações

Mesmo com as contribuições obtidas, não é possível entender de maneira adequada o comportamento de quantificadores na regularização, principalmente em cenários onde existem variações consideráveis de frequência das classes entre dados de treinamento e teste. Outro problema está relacionado com a definição dos parâmetros usados na construção de espectrogramas, que são dependentes da estrutura das vocalizações. Mesmo usando uma configuração que busca abranger a maior quantidade de padrões, não é possível garantir que as usadas aqui são as mais adequadas. Essas mesmas questões podem ser levantadas sobre as demais características associadas com esses espectrogramas. Por fim, mesmo que os modelos consigam segregar pássaros de anuros e gerem resultados superiores a um SVM linear (cf. [Tabela 13](#)) com características acústicas como entrada, ainda

é necessário melhorar a diferenciação entre as espécies de pássaros e entre as espécies de anuros. Por causa disso, os resultados das arquiteturas e processos empregados estão saturados em torno de 0,77 de acurácia balanceada (cf. [Tabela 14](#) e [Tabela 15](#)).

### 8.3 Trabalhos futuros

A partir das limitações apresentadas, os próximos passos de pesquisa precisam aperfeiçoar a função de custo com outras técnicas de quantificação ([GONZÁLEZ et al., 2017a](#)) e outros regularizadores, além de testar variações das frequências das classes. Também são necessários estudos tanto com parametrização das representações de tempo-frequência quanto da aplicação de outras representações, como *harmonic percussive source separation* (HPSS) ([DRIEDGER; MÜLLER; DISCH, 2014](#)). Na sequência, são sugeridas avaliações de outras tarefas de Aprendizado Autossupervisionado e suas respectivas funções de custo. Por fim, também se fazem necessárias análises de otimizadores e seus parâmetros, aplicados ao cenário específico de identificação de sons naturais.

## REFERÊNCIAS

---

---

- AMPHIBIAWEB. **AmphibiaWeb**. 2020. Disponível em: <<https://amphibiaweb.org>>. Acesso em: 08/19/2020. Citado na página 91.
- ANTONELLI, V. **XC424989**. 2018. Disponível em: <<https://www.xeno-canto.org/424989>>. Acesso em: 08/19/2020. Citado na página 90.
- ATHANAS, N. **Chivi Vireo**. 2013. Disponível em: <<https://flic.kr/p/WhHXT7>>. Acesso em: 08/19/2020. Citado na página 90.
- AYTAR, Y.; VONDRICK, C.; TORRALBA, A. Soundnet: Learning sound representations from unlabeled video. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2016. p. 892–900. Citado na página 84.
- BA, J. L.; KIROUS, J. R.; HINTON, G. E. Layer normalization. **arXiv preprint arXiv:1607.06450**, 2016. Citado na página 54.
- BAEVSKI, A.; ZHOU, Y.; MOHAMED, A.; AULI, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. **Advances in Neural Information Processing Systems**, v. 33, p. 12449–12460, 2020. Citado na página 59.
- BARDES, A.; PONCE, J.; LECUN, Y. VICReg: Variance-Invariance-Covariance regularization for Self-supervised learning. In: **ICLR 2022-10th International Conference on Learning Representations**. [S.l.: s.n.], 2022. Citado nas páginas 59, 61, 141, 150, 151 e 152.
- BECHER, A. R.; PONTI, M. A. Optimization Matters: Guidelines to improve representation learning with Deep Networks. In: SBC. **Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2021. p. 595–606. Citado nas páginas 55, 87, 100 e 135.
- BEDOYA, C.; ISAZA, C.; DAZA, J. M.; LÓPEZ, J. D. Automatic identification of rainfall in acoustic recordings. **Ecological indicators**, Elsevier, v. 75, p. 95–100, 2017. Citado na página 42.
- BEIJBOM, O.; HOFFMAN, J.; YAO, E.; DARRELL, T.; RODRIGUEZ-RAMIREZ, A.; GONZALEZ-RIVERO, M.; GULDBERG, O. H. Quantification in-the-wild: data-sets and baselines. **arXiv preprint arXiv:1510.04811**, 2015. Citado na página 62.
- BELLA, A.; FERRI, C.; HERNÁNDEZ-ORALLO, J.; RAMIREZ-QUINTANA, M. J. Quantification via probability estimators. In: IEEE. **2010 IEEE International Conference on Data Mining**. [S.l.], 2010. p. 737–742. Citado na página 62.
- BITTENCOURT, L.; BARBOSA, M.; SECCHI, E.; JR, J. L.-B.; AZEVEDO, A. Acoustic habitat of an oceanic archipelago in the Southwestern Atlantic. **Deep Sea Research Part I: Oceanographic Research Papers**, Elsevier, v. 115, p. 103–111, 2016. Citado na página 42.

- BOELMAN, N. T.; ASNER, G. P.; HART, P. J.; MARTIN, R. E. Multi-trophic invasion resistance in Hawaii: bioacoustics, field surveys, and airborne remote sensing. **Ecological Applications**, Wiley Online Library, v. 17, n. 8, p. 2137–2144, 2007. Citado na página 46.
- BOTTOU, L. Online algorithms and stochastic approximations. In: SAAD, D. (Ed.). **Online Learning and Neural Networks**. Cambridge, UK: Cambridge University Press, 1998. Revised, oct 2012. Disponível em: <<http://leon.bottou.org/papers/bottou-98x>>. Citado na página 55.
- BRAGA, A. d. P.; FERREIRA, A. C. P. d. L.; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC Editora Rio de Janeiro, Brazil, 2007. Citado nas páginas 51 e 52.
- BRIGGS, F.; LAKSHMINARAYANAN, B.; NEAL, L.; FERN, X. Z.; RAICH, R.; HADLEY, S. J. K.; HADLEY, A. S.; BETTS, M. G. Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach. **The Journal of the Acoustical Society of America**, ASA, v. 131, n. 6, p. 4640–4650, 2012. Citado na página 39.
- BROMLEY, J.; GUYON, I.; LECUN, Y.; SÄCKINGER, E.; SHAH, R. Signature verification using a "siamese" time delay neural network. **Advances in neural information processing systems**, v. 6, 1994. Citado nas páginas 60 e 125.
- BROWN, A.; GARG, S.; MONTGOMERY, J. Automatic rain and cicada chorus filtering of bird acoustic data. **Applied Soft Computing**, Elsevier, v. 81, p. 105501, 2019. Citado na página 33.
- BUADES, A.; COLL, B.; MOREL, J.-M. A non-local algorithm for image denoising. In: IEEE. **2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)**. [S.l.], 2005. v. 2, p. 60–65. Citado na página 74.
- CAKIR, E.; PARASCANDOLO, G.; HEITTOLA, T.; HUTTUNEN, H.; VIRTANEN, T. Convolutional recurrent neural networks for polyphonic sound event detection. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, IEEE, v. 25, n. 6, p. 1291–1303, 2017. Citado nas páginas 32 e 39.
- CASANOVA, E.; WEBER, J.; SHULBY, C. D.; JUNIOR, A. C.; GÖLGE, E.; PONTI, M. A. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2022. p. 2709–2720. Citado na página 39.
- CAVALLARI, G. B.; PONTI, M. A. Semi-supervised siamese network using self-supervision under scarce annotation improves class separability and robustness to attack. In: IEEE. **2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2021. p. 223–230. Citado na página 47.
- CHEN, C. P.; ZHANG, C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. **Information Sciences**, Elsevier, v. 275, p. 314–347, 2014. Citado na página 31.

- CHI, P.-H.; CHUNG, P.-H.; WU, T.-H.; HSIEH, C.-C.; CHEN, Y.-H.; LI, S.-W.; LEE, H.-y. Audio bert: A lite bert for self-supervised learning of audio representation. In: IEEE. **2021 IEEE Spoken Language Technology Workshop (SLT)**. [S.l.], 2021. p. 344–350. Citado na página 59.
- CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014. Citado na página 82.
- CHOPRA, S.; HADSELL, R.; LECUN, Y. Learning a similarity metric discriminatively, with application to face verification. In: IEEE. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**. [S.l.], 2005. v. 1, p. 539–546. Citado nas páginas 60 e 125.
- CHUDÝ, A. **Great Kiskadee**. 2019. Disponível em: <<https://flic.kr/p/25m7ViZ>>. Acesso em: 08/18/2020. Citado na página 90.
- COIMBRA, D. B. **Multidimensional projections for the visual exploration of multimedia data**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado nas páginas 31, 36, 58 e 93.
- CRAMER, J.; LOSTANLEN, V.; FARNSWORTH, A.; SALAMON, J.; BELLO, J. P. Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers. In: IEEE. **ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2020. p. 901–905. Citado nas páginas 40, 68, 69, 70, 83, 84, 85, 122 e 157.
- DELCROIX, M.; KINOSHITA, K.; HORI, T.; NAKATANI, T. Context adaptive deep neural networks for fast acoustic model adaptation. In: IEEE. **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2015. p. 4535–4539. Citado na página 67.
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255. Citado nas páginas 59, 72, 100, 122 e 140.
- DEPRAETERE, M.; PAVOINE, S.; JIGUET, F.; GASC, A.; DUVAIL, S.; SUEUR, J. Monitoring animal diversity using acoustic indices: Implementation in a temperate woodland. **Ecological Indicators**, Elsevier, v. 13, n. 1, p. 46–54, 2012. Citado na página 46.
- DESHPANDE, A. **A Beginner's Guide To Understanding Convolutional Neural Networks**. 2016. Disponível em: <<https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>>. Acesso em: 09/26/2019. Citado na página 55.
- DHILLON, I. S.; MODHA, D. S. Concept decompositions for large sparse text data using clustering. **Machine learning**, Springer, v. 42, n. 1-2, p. 143–175, 2001. Citado na página 66.



- DIAS, F. F. **Uma estratégia para análise visual de Paisagens Acústicas com base em seleção de características discriminantes**. Dissertação (Mestrado) — Universidade de São Paulo, 2018. Citado nas páginas 99 e 102.
- DIAS, F. F.; PEDRINI, H.; MINGHIM, R. Soundscape segregation based on visual analysis and discriminating features. **Ecological Informatics**, Elsevier, v. 61, p. 101184, 2021. ISSN 1574-9541. Citado na página 121.
- DIAS, F. F.; PONTI, M. A.; MINGHIM, R. A classification and quantification approach to generate features in soundscape ecology using neural networks. **Neural Computing and Applications**, Springer Science and Business Media LLC, v. 34, n. 3, p. 1923–1937, sep 2021. Citado nas páginas 34, 95, 96, 97, 99, 100, 101, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 121, 122, 131, 132, 134 e 156.
- \_\_\_\_\_. Implementing simple spectral denoising for environmental audio recordings. **arXiv preprint arXiv:2201.02099**, 2022. Citado na página 46.
- DOERSCH, C.; GUPTA, A.; EFROS, A. A. Unsupervised visual representation learning by context prediction. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1422–1430. Citado na página 59.
- DONG, X.; TOWSEY, M.; ZHANG, J.; ROE, P. Compact features for birdcall retrieval from environmental acoustic recordings. In: IEEE. **Proceedings of the 2015 IEEE 15th International Conference on Data Mining Workshops**. [S.l.]: IEEE Computer Society, 2015. p. 1–6. Citado na página 39.
- DRIEDGER, J.; MÜLLER, M.; DISCH, S. Extending harmonic-percussive separation of audio signals. In: **ISMIR**. [S.l.: s.n.], 2014. p. 611–616. Citado nas páginas 73 e 158.
- DRÖGE, S.; MARTIN, D. A.; ANDRIAFANOMEZANTSOA, R.; BURIVALOVA, Z.; FULGENCE, T. R.; OSEN, K.; RAKOTOMALALA, E.; SCHWAB, D.; WURZ, A.; RICHTER, T. *et al.* Listening to a changing landscape: Acoustic indices reflect bird species richness and plot-scale vegetation structure across different land-use types in north-eastern Madagascar. **Ecological Indicators**, Elsevier, v. 120, p. 106929, 2021. Citado na página 46.
- DUFOURQ, E.; BATIST, C.; FOQUET, R.; DURBACH, I. Passive acoustic monitoring of animal populations with transfer learning. **Ecological Informatics**, Elsevier, v. 70, p. 101688, 2022. Citado nas páginas 33, 59, 84, 87, 92, 122 e 135.
- DUPONT, B. **Lesser Treefrog (Dendropsophus minutus)**. 2017. Disponível em: <<https://flic.kr/p/HnWyYn>>. Acesso em: 08/19/2020. Citado na página 91.
- ELDRIDGE, A.; CASEY, M.; MOSCOSO, P.; PECK, M. A new method for ecoacoustics? Toward the extraction and evaluation of ecologically-meaningful soundscape components using sparse coding methods. **PeerJ**, PeerJ Inc., v. 4, p. e2108, 2016. Citado nas páginas 33 e 46.
- ELDRIDGE, A.; GUYOT, P.; MOSCOSO, P.; JOHNSTON, A.; EYRE-WALKER, Y.; PECK, M. Sounding out ecoacoustic metrics: Avian species richness is predicted by acoustic indices in temperate but not tropical habitats. **Ecological indicators**, Elsevier, v. 95, p. 939–952, 2018. Citado nas páginas 45 e 46.



- EY, E.; FISCHER, J. The "acoustic adaptation hypothesis- a review of the evidence from birds, anurans and mammals. **Bioacoustics**, Taylor & Francis, v. 19, n. 1-2, p. 21–48, 2009. Citado na página 44.
- FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. **Inteligência Artificial: Uma abordagem de aprendizado de máquina**. [S.l.]: LTC Editora Rio de Janeiro, Brasil, 2011. Citado nas páginas 48, 49 e 50.
- FINK, E. "Myiothlypis leucoblephara- White-browed Warbler, Olivflanken-Waldsanger). 2018. Disponível em: <<https://flic.kr/p/2hhJYCi>>. Acesso em: 08/16/2020. Citado na página 90.
- FLORENTIN, J.; DUTOIT, T.; VERLINDEN, O. Detection and identification of european woodpeckers with deep convolutional neural networks. **Ecological Informatics**, Elsevier, v. 55, p. 101023, 2020. Citado nas páginas 39 e 58.
- FORMAN, G. Counting positives accurately despite inaccurate classification. In: SPRINGER. **European Conference on Machine Learning**. [S.l.], 2005. p. 564–575. Citado na página 62.
- \_\_\_\_\_. Quantifying counts and costs via classification. **Data Mining and Knowledge Discovery**, Springer, v. 17, n. 2, p. 164–206, 2008. Citado na página 62.
- FRAGA, R. M. **XC572550**. 2001. Disponível em: <<https://www.xeno-canto.org/572550>>. Acesso em: 08/17/2020. Citado na página 90.
- GAN, H.; ZHANG, J.; TOWSEY, M.; TRUSKINGER, A.; STARK, D.; RENSBURG, B. J. van; LI, Y.; ROE, P. Data selection in frog chorusing recognition with acoustic indices. **Ecological Informatics**, Elsevier, v. 60, p. 101160, 2020. Citado nas páginas 45, 46 e 93.
- GAN, H.; ZHANG, J.; TOWSEY, M.; TRUSKINGER, A.; STARK, D.; van Rensburg, B. J.; LI, Y.; ROE, P. A novel frog chorusing recognition method with acoustic indices and machine learning. **Future Generation Computer Systems**, Elsevier, v. 125, p. 485–495, 2021. Citado nas páginas 45 e 93.
- GAO, W.; SEBASTIANI, F. From classification to quantification in tweet sentiment analysis. **Social Network Analysis and Mining**, Springer, v. 6, n. 1, p. 19, 2016. Citado na página 62.
- GASC, A.; SUEUR, J.; PAVOINE, S.; PELLENS, R.; GRANDCOLAS, P. Biodiversity sampling using a global acoustic approach: contrasting sites with microendemics in New Caledonia. **PLoS One**, Public Library of Science, v. 8, n. 5, p. e65311, 2013. Citado na página 46.
- GEMMEKE, J. F.; ELLIS, D. P. W.; FREEDMAN, D.; JANSEN, A.; LAWRENCE, W.; MOORE, R. C.; PLAKAL, M.; RITTER, M. Audio set: An ontology and human-labeled dataset for audio events. In: **Proc. IEEE ICASSP 2017**. New Orleans, LA: [s.n.], 2017. Citado nas páginas 59, 88, 119 e 138.
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 580–587. Citado na página 72.

- GODOY, F. I. d. **XC421842**. 2016. Disponível em: <[www.xeno-canto.org/421842](http://www.xeno-canto.org/421842)>. Acesso em: 08/18/2020. Citado na página 90.
- GÓMEZ, W. E.; ISAZA, C. V.; DAZA, J. M. Identifying disturbed habitats: A new method from acoustic indices. **Ecological Informatics**, Elsevier, v. 45, p. 16–25, 2018. Citado na página 45.
- GONZÁLEZ-CASTRO, V.; ALAIZ-RODRÍGUEZ, R.; ALEGRE, E. Class distribution estimation based on the Hellinger distance. **Information Sciences**, Elsevier, v. 218, p. 146–164, 2013. Citado na página 62.
- GONZÁLEZ, P.; CASTAÑO, A.; CHAWLA, N. V.; COZ, J. J. D. A review on quantification learning. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 50, n. 5, p. 1–40, 2017. Citado nas páginas 62 e 158.
- GONZÁLEZ, P.; DÍEZ, J.; CHAWLA, N.; COZ, J. J. del. Why is quantification an interesting learning problem? **Progress in Artificial Intelligence**, Springer, v. 6, n. 1, p. 53–58, 2017. Citado na página 62.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens. tradução: Cristina Yamagami e Leonardo Piamonte**. [S.l.]: Pearson Prentice Hall, São Paulo, 2010. Citado nas páginas 38, 42, 43 e 44.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016. Citado nas páginas 44, 47, 52, 53, 54, 55, 59 e 93.
- GULLI, A.; PAL, S. **Deep Learning with Keras**. [S.l.]: Packt Publishing, 2017. ISBN 9781787129030. Citado na página 53.
- HAN, W.; CHAN, C.-F.; CHOY, C.-S.; PUN, K.-P. An efficient MFCC extraction method in speech recognition. In: IEEE. **2006 IEEE international symposium on circuits and systems**. [S.l.], 2006. p. 4–pp. Citado na página 40.
- HARRIS, S. A.; SHEARS, N. T.; RADFORD, C. A. Ecoacoustic indices as proxies for biodiversity on temperate reefs. **Methods in Ecology and Evolution**, Wiley Online Library, v. 7, n. 6, p. 713–724, 2016. Citado na página 46.
- HARVEY, M. **Acoustic Detection of Humpback Whales Using a Convolutional Neural Network**. 2018. Disponível em: <<https://ai.googleblog.com/2018/10/acoustic-detection-of-humpback-whales.html>>. Acesso em: 11/05/2018. Citado nas páginas 32, 40, 76, 77, 78, 85, 100, 122, 140 e 157.
- HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2nd. ed. [S.l.]: Prentice Hall, 1999. Citado nas páginas 51 e 52.
- HAYKIN, S. S.; VEEN, B. V. **Sinais e sistemas**. [S.l.]: Bookman, 2001. Citado nas páginas 38, 39 e 42.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015. Citado na página 123.

\_\_\_\_\_. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. Citado nas páginas [55](#), [96](#), [100](#), [122](#), [134](#) e [140](#).

HE, T.; ZHANG, Z.; ZHANG, H.; ZHANG, Z.; XIE, J.; LI, M. Bag of tricks for image classification with convolutional neural networks. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 558–567. Citado na página [71](#).

HERSHEY, S.; CHAUDHURI, S.; ELLIS, D. P.; GEMMEKE, J. F.; JANSEN, A.; MOORE, R. C.; PLAKAL, M.; PLATT, D.; SAUROUS, R. A.; SEYBOLD, B. *et al.* CNN architectures for large-scale audio classification. In: IEEE. **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2017. p. 131–135. Citado nas páginas [88](#) e [92](#).

HILASACA, L. H.; RIBEIRO, M. C.; MINGHIM, R. Visual Active Learning for labeling: A case for Soundscape Ecology data. **Information**, Multidisciplinary Digital Publishing Institute, v. 12, n. 7, p. 265, 2021. Citado nas páginas [45](#), [118](#) e [138](#).

HILASACA, L. M. H.; GASPAR, L. P.; RIBEIRO, M. C.; MINGHIM, R. Visualization and categorization of ecological acoustic events based on discriminant features. **Ecological Indicators**, Elsevier, v. 126, p. 107316, 2021. Citado nas páginas [45](#), [118](#) e [138](#).

HOWARD, A.; SANDLER, M.; CHU, G.; CHEN, L.-C.; CHEN, B.; TAN, M.; WANG, W.; ZHU, Y.; PANG, R.; VASUDEVAN, V. *et al.* Searching for mobilenetv3. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 1314–1324. Citado nas páginas [57](#) e [140](#).

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Citado na página [57](#).

IDREES, S. M.; ALAM, M. A.; AGARWAL, P. A study of big data and its challenges. **International Journal of Information Technology**, Springer, v. 11, n. 4, p. 841–846, 2019. Citado na página [31](#).

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 448–456. Citado na página [54](#).

JANSEN, A.; PLAKAL, M.; PANDYA, R.; ELLIS, D. P.; HERSHEY, S.; LIU, J.; MOORE, R. C.; SAUROUS, R. A. Unsupervised learning of semantic audio representations. In: IEEE. **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2018. p. 126–130. Citado na página [76](#).

JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. In: **Proceedings of the 22nd ACM international conference on Multimedia**. [S.l.: s.n.], 2014. p. 675–678. Citado na página [72](#).

- JOHNSON, J. M.; KHOSHGOFTAAR, T. M. Survey on deep learning with class imbalance. **Journal of Big Data**, Springer, v. 6, n. 1, p. 1–54, 2019. Citado nas páginas [49](#), [98](#), [120](#) e [139](#).
- JORGE, F. C.; MACHADO, C. G.; NOGUEIRA, S. S. da C.; NOGUEIRA-FILHO, S. L. G. The effectiveness of acoustic indices for forest monitoring in Atlantic rainforest fragments. **Ecological Indicators**, Elsevier, v. 91, p. 71–76, 2018. Citado na página [46](#).
- KAHL, S.; STÖTER, F.-R.; GOËAU, H.; GLOTIN, H.; PLANQUE, R.; VELLINGA, W.-P.; JOLY, A. Overview of BirdCLEF 2019: large-scale bird recognition in soundscapes. In: CEUR. **Working Notes of CLEF 2019-Conference and Labs of the Evaluation Forum**. [S.l.], 2019. p. 1–9. Citado na página [72](#).
- KAHL, S.; WILHELM-STEIN, T.; KLINCK, H.; KOWERKO, D.; EIBL, M. Recognizing birds from sound-the 2018 BirdCLEF baseline system. **arXiv preprint arXiv:1804.07177**, 2018. Citado na página [82](#).
- KAHL, S.; WOOD, C. M.; EIBL, M.; KLINCK, H. BirdNET: A deep learning solution for avian diversity monitoring. **Ecological Informatics**, Elsevier, v. 61, p. 101236, 2021. Citado nas páginas [33](#), [70](#), [71](#), [72](#), [82](#), [83](#), [85](#), [87](#), [89](#), [119](#) e [157](#).
- KASTEN, E. P.; GAGE, S. H.; FOX, J.; JOO, W. The remote environmental assessment laboratory’s acoustic library: An archive for studying soundscape ecology. **Ecological Informatics**, Elsevier, v. 12, p. 50–67, 2012. Citado nas páginas [42](#) e [46](#).
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014. Citado nas páginas [55](#) e [135](#).
- KIRSEBOM, O. S.; FRAZAO, F.; SIMARD, Y.; ROY, N.; MATWIN, S.; GIARD, S. Performance of a deep neural network at detecting North Atlantic right whale upcalls. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 147, n. 4, p. 2636–2646, 2020. Citado nas páginas [80](#), [81](#), [83](#), [85](#) e [157](#).
- KOMODAKIS, N.; GIDARIS, S. Unsupervised representation learning by predicting image rotations. In: **International Conference on Learning Representations (ICLR)**. [S.l.: s.n.], 2018. Citado na página [59](#).
- KONG, Q.; CAO, Y.; IQBAL, T.; WANG, Y.; WANG, W.; PLUMBLEY, M. D. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, IEEE, v. 28, p. 2880–2894, 2020. Citado na página [92](#).
- KORNBLITH, S.; NOROUZI, M.; LEE, H.; HINTON, G. Similarity of neural network representations revisited. In: **Proceedings of the 36th International Conference on Machine Learning**. [S.l.]: PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 3519–3529. Citado na página [58](#).
- KORNBLITH, S.; SHLENS, J.; LE, Q. V. Do better imagenet models transfer better? In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 2661–2671. Citado na página [100](#).
- KRAUSE, B. Bioacoustics, habitat ambience in ecological balance. **Whole Earth Review**, v. 57, p. 14–18, 1987. Citado nas páginas [32](#) e [44](#).

- KRAUSE, B.; FARINA, A. Using ecoacoustic methods to survey the impacts of climate change on biodiversity. **Biological Conservation**, Elsevier, v. 195, n. January, p. 245–254, 2016. Citado nas páginas 33 e 46.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012. Citado na página 72.
- LEBIEN, J.; ZHONG, M.; CAMPOS-CERQUEIRA, M.; VELEV, J. P.; DODHIA, R.; FERRES, J. L.; AIDE, T. M. A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network. **Ecological Informatics**, Elsevier, p. 101113, 2020. Citado nas páginas 39, 59, 75, 76, 83, 85, 92, 100, 122, 140 e 157.
- LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 82.
- LEITE, G. **XC573115**. 2020. Disponível em: <<https://www.xeno-canto.org/573115>>. Acesso em: 08/19/2020. Citado na página 90.
- LEMES, W. M. C. **Pitiguari / Rufous-browed Peppershrike (Cyclarhis gujanensis)**. 2014. Disponível em: <<https://flic.kr/p/noho35>>. Acesso em: 08/16/2020. Citado na página 90.
- LIN, T.-H.; FANG, S.-H.; TSAO, Y. Improving biodiversity assessment via unsupervised separation of biological sounds from long-duration recordings. **Scientific reports**, Nature Publishing Group, v. 7, n. 1, p. 4547, 2017. Citado na página 33.
- LIN, T.-H.; TSAO, Y. Source separation in ecoacoustics: A roadmap towards versatile soundscape information retrieval. **Remote Sensing in Ecology and Conservation**, Wiley Online Library, v. 6, n. 3, p. 236–247, 2020. Citado na página 33.
- LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K.; DOLLAR, P. Focal loss for dense object detection. In: **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2017. Citado na página 73.
- LOGAN, B. Mel frequency cepstral coefficients for music modeling. In: **In International Symposium on Music Information Retrieval**. [S.l.: s.n.], 2000. Citado nas páginas 39 e 42.
- LOSTANLEN, V.; SALAMON, J.; CARTWRIGHT, M.; MCFEE, B.; FARNSWORTH, A.; KELLING, S.; BELLO, J. P. Per-channel energy normalization: Why and How. **IEEE Signal Processing Letters**, IEEE, v. 26, n. 1, p. 39–43, 2018a. Citado nas páginas 40, 41 e 70.
- LOSTANLEN, V.; SALAMON, J.; FARNSWORTH, A.; KELLING, S.; BELLO, J. P. Birdvox-full-night: A dataset and benchmark for avian flight call detection. In: **IEEE 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2018b. p. 266–270. Citado na página 68.
- \_\_\_\_\_. **BirdVox-70k: a dataset for species-agnostic flight call detection in half-second clips**. [S.l.]: Zenodo, 2018c. Citado na página 68.



- \_\_\_\_\_. Robust sound event detection in bioacoustic sensor networks. **PloS one**, Public Library of Science San Francisco, CA USA, v. 14, n. 10, p. e0214168, 2019. Citado nas páginas 40, 59, 67, 68, 83, 84, 85, 87, 89, 92, 124 e 157.
- MAATEN, L. v. d.; HINTON, G. Visualizing data using t-SNE. **Journal of Machine Learning Research**, v. 9, n. Nov, p. 2579–2605, 2008. Citado nas páginas 102 e 143.
- MADISETTI, V. K. **The Digital Signal Processing Handbook**. [S.l.]: CRC Press, 2009. Citado na página 41.
- MAIZLISH, A. **Golden-crowned Warbler**. 2013. Disponível em: <<https://flic.kr/p/dYq4dQ>>. Acesso em: 08/16/2020. Citado na página 90.
- MALETZKE, A.; REIS, D. dos; CHERMAN, E.; BATISTA, G. Dys: a framework for mixture models in quantification. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, n. 01, p. 4552–4560. Citado na página 62.
- MALETZKE, A. G.; REIS, D. M. dos; BATISTA, G. E. Quantification in data streams: Initial results. In: IEEE. **2017 Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2017. p. 43–48. Citado na página 62.
- MATLAB. **Mel spectrogram**. 2019. Disponível em: <<https://www.mathworks.com/help/audio/ref/melspectrogram.html>>. Acesso em: 31/10/2019. Citado nas páginas 40 e 41.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 51.
- MCFEE, B.; HUMPHREY, E. J.; BELLO, J. P. A software framework for musical data augmentation. In: **ISMIR**. [S.l.: s.n.], 2015. v. 2015, p. 248–254. Citado na página 70.
- MELLINGER, D. K. A comparison of methods for detecting right whale calls. **Canadian Acoustics**, v. 32, n. 2, p. 55–65, 2004. Citado na página 81.
- MELLO, R. F.; PONTI, M. A. **Machine learning: a practical approach on the statistical learning theory**. [S.l.]: Springer, 2018. Citado nas páginas 47, 48, 99 e 121.
- MEZQUIDA, D. A.; MARTÍNEZ, J. L. Platform for bee-hives monitoring based on sound analysis. a perpetual warehouse for swarm’s daily activity. **Spanish Journal of Agricultural Research**, v. 7, n. 4, p. 824–828, 2009. Citado na página 42.
- MITCHELL, S. L.; BICKNELL, J. E.; EDWARDS, D. P.; DEERE, N. J.; BERNARD, H.; DAVIES, Z. G.; STRUEBIG, M. J. Spatial replication and habitat context matters for assessments of tropical biodiversity using acoustic indices. **Ecological Indicators**, Elsevier, v. 119, p. 106717, 2020. Citado nas páginas 33, 44 e 46.
- MOREO, A.; SEBASTIANI, F. Tweet sentiment quantification: An experimental re-evaluation. **Plos one**, Public Library of Science San Francisco, CA USA, v. 17, n. 9, p. e0263449, 2022. Citado nas páginas 61 e 62.

- MOUY, X.; BAHOURA, M.; SIMARD, Y. Automatic recognition of fin and blue whale calls for real-time monitoring in the St. Lawrence. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 126, n. 6, p. 2918–2928, 2009. Citado na página 81.
- MULLET, T. C.; FARINA, A.; GAGE, S. H. The acoustic habitat hypothesis: An ecoacoustics perspective on species habitat selection. **Biosemiotics**, Springer, v. 10, n. 3, p. 319–336, 2017. Citado na página 44.
- MULLET, T. C.; GAGE, S. H.; MORTON, J. M.; HUETTMANN, F. Temporal and spatial variation of a winter soundscape in south-central Alaska. **Landscape ecology**, Springer, v. 31, n. 5, p. 1117–1137, 2016. Citado na página 42.
- MUNZNER, T. **Visualization Analysis and Design**. [S.l.]: CRC Press, 2014. Citado na página 36.
- NEAL, L. **Detection and Segmentation of Bird Song in Noisy Environments**. Dissertação (Mestrado) — Oregon State University, 2012. Citado na página 45.
- NGUYEN, T.; RAGHU, M.; KORNBLITH, S. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. **arXiv preprint arXiv:2010.15327**, 2020. Citado nas páginas 58 e 93.
- NOCEDAL, J.; WRIGHT, S. **Numerical optimization**. [S.l.]: Springer Science & Business Media, 2006. Citado nas páginas 52 e 92.
- NONATO, L. G.; AUPETIT, M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 25, n. 8, p. 2650–2673, 2018. Citado nas páginas 57, 102 e 143.
- OORD, A. v. d.; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A.; KAVUKCUOGLU, K. Wavenet: A generative model for raw audio. **arXiv preprint arXiv:1609.03499**, 2016. Citado na página 84.
- OWENS, A.; EFROS, A. A. Audio-visual scene analysis with self-supervised multisensory features. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 631–648. Citado na página 59.
- PARASCANDOLO, G.; HUTTUNEN, H.; VIRTANEN, T. Recurrent neural networks for polyphonic sound event detection in real life recordings. In: IEEE. **2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2016. p. 6440–6444. Citado nas páginas 39 e 59.
- PARKS, S. E.; MIKSIS-OLDS, J. L.; DENES, S. L. Assessing marine ecosystem acoustic diversity across ocean basins. **Ecological Informatics**, Elsevier, v. 21, p. 81–88, 2014. Citado na página 33.
- PASSOS, M. d. A. **Adenomera marmorata**. 2014. Disponível em: <<http://michelpassosherpetolife.blogspot.com/search/label/Adenomera%20marmorata>>. Acesso em: 06/18/2022. Citado na página 91.

\_\_\_\_\_. **Boana albopunctata**. 2016. Disponível em: <<http://michelpassosherpetolife.blogspot.com/search/label/Boana%20albopunctata>>. Acesso em: 06/18/2022. Citado na página 91.

\_\_\_\_\_. **Aplastodiscus leucopygius**. 2020. Disponível em: <<http://michelpassosherpetolife.blogspot.com/search/label/Aplastodiscus%20leucopygius>>. Acesso em: 06/18/2022. Citado na página 91.

PAULOVICH, F. V. **Mapeamento de dados multi-dimensionais-integrando mineração e visualização**. Tese (Doutorado) — Universidade de São Paulo, 2008. Citado na página 31.

PEARSON, K. On lines and planes of closest fit to systems of points in space. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901. Citado na página 66.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado nas páginas 99, 122 e 142.

PEKIN, B.; JUNG, J.; VILLANUEVA-RIVERA, L.; PIJANOWSKI, B.; AHUMADA, J. Modeling acoustic diversity using soundscape recordings and LIDAR-derived metrics of vertical forest structure in anetropical rainforest. **Landscape Ecology**, Springer, v. 27, n. 10, p. 1513–1522, 2012. Citado na página 46.

PHILLIPS, Y. F.; TOWSEY, M.; ROE, P. Revealing the ecological content of long-duration audio-recordings of the environment through clustering and visualisation. **PloS one**, Public Library of Science, v. 13, n. 3, p. e0193345, 2018. Citado na página 45.

PIERETTI, N.; FARINA, A.; MORRI, D. A new methodology to infer the singing activity of an avian community: The Acoustic Complexity Index (ACI). **Ecological Indicators**, Elsevier, v. 11, n. 3, p. 868–873, 2011. Citado na página 46.

PIERETTI, N.; MARTIRE, M. L.; FARINA, A.; DANOVARO, R. Marine soundscape as an additional biodiversity monitoring tool: A case study from the Adriatic Sea (Mediterranean Sea). **Ecological indicators**, Elsevier, v. 83, p. 13–20, 2017. Citado nas páginas 45 e 46.

PIJANOWSKI, B. C.; FARINA, A.; GAGE, S. H.; DUMYAHN, S. L.; KRAUSE, B. L. What is soundscape ecology? An introduction and overview of an emerging new science. **Landscape Ecology**, Springer, v. 26, n. 9, p. 1213–1232, 2011. Citado na página 32.

PIJANOWSKI, B. C.; VILLANUEVA-RIVERA, L. J.; DUMYAHN, S. L.; FARINA, A.; KRAUSE, B. L.; NAPOLETANO, B. M.; GAGE, S. H.; PIERETTI, N. Soundscape ecology: the science of sound in the landscape. **BioScience**, Oxford University Press, v. 61, n. 3, p. 23–216, 2011. Citado nas páginas 32, 44 e 45.

PONTI, M. A.; RIBEIRO, L. S. F.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about Deep Learning for Computer Vision but were



- afraid to ask. In: BRAZILIAN COMPUTER SOCIETY - SBC. **SIBGRAPI - Conference on Graphics, Patterns and Images**. [S.l.], 2017. Citado nas páginas 52, 54, 55, 59 e 92.
- PONTI, M. A.; SANTOS, F. P. d.; RIBEIRO, L. S. F.; CAVALLARI, G. B. Training deep networks from zero to hero: avoiding pitfalls and going beyond. In: IEEE. **2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2021. p. 9–16. Citado nas páginas 58 e 92.
- PRANDONI, P.; VETTERLI, M. **Signal processing for communications**. [S.l.]: EPFL press, 2008. Citado na página 37.
- PRESS, W. H.; TEUKOLSKY, S. A. Savitzky-Golay smoothing filters. **Computers in Physics**, American Institute of Physics, v. 4, n. 6, p. 669–672, 1990. Citado na página 74.
- PROVETE, D. B. **Physalaemus cuvieri 02**. 2007. Disponível em: <<https://flic.kr/p/exNK9f>>. Acesso em: 06/18/2022. Citado na página 91.
- \_\_\_\_\_. **Ischnocnema guentheri**. 2008. Disponível em: <<https://flic.kr/p/afGq8f>>. Acesso em: 08/19/2020. Citado na página 91.
- PURWINS, H.; LI, B.; VIRTANEN, T.; SCHLÜTER, J.; CHANG, S.-Y.; SAINATH, T. Deep learning for audio signal processing. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 13, n. 2, p. 206–219, 2019. Citado na página 84.
- RADFORD, A. N.; KERRIDGE, E.; SIMPSON, S. D. Acoustic communication in a noisy world: can fish compete with anthropogenic noise? **Behavioral Ecology**, Oxford University Press UK, v. 25, n. 5, p. 1022–1030, 2014. Citado na página 32.
- RAMSAY, J. O. **Functional data analysis**. [S.l.]: Wiley Online Library, 2006. Citado na página 42.
- RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. 3. ed. [S.l.]: Pearson, 2010. Citado nas páginas 31, 46, 47, 48 e 50.
- SA, V. R. de. Learning classification with unlabeled data. **Advances in neural information processing systems**, Morgan Kaufmann Publishers, p. 112–112, 1994. Citado na página 59.
- SALAMON, J.; BELLO, J. P. Unsupervised feature learning for urban sound classification. In: IEEE. **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2015. p. 171–175. Citado nas páginas 39 e 46.
- \_\_\_\_\_. Deep convolutional neural networks and data augmentation for environmental sound classification. **IEEE Signal Processing Letters**, IEEE, v. 24, n. 3, p. 279–283, 2017. Citado nas páginas 33, 59, 66, 67, 87, 89, 96, 98, 120 e 140.
- SALAMON, J.; BELLO, J. P.; FARNSWORTH, A.; ROBBINS, M.; KEEN, S.; KLINCK, H.; KELLING, S. Towards the automatic classification of avian flight calls for bioacoustic monitoring. **PloS one**, Public Library of Science San Francisco, CA USA, v. 11, n. 11, p. e0166866, 2016. Citado nas páginas 66 e 67.

SALAMON, J.; BELLO, J. P.; FARNSWORTH, A.; KELLING, S. Fusing shallow and deep learning for bioacoustic bird species classification. In: IEEE. **2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)**. [S.l.], 2017. p. 141–145. Citado nas páginas [65](#), [66](#), [67](#), [69](#), [83](#), [84](#), [85](#) e [157](#).

SALAMON, J.; JACOBY, C.; BELLO, J. P. A dataset and taxonomy for urban sound research. In: ACM. **Proceedings of the 22nd ACM international conference on Multimedia**. [S.l.], 2014. p. 1041–1044. Citado na página [66](#).

SÁNCHEZ-GENDRIZ, I.; PADOVESE, L. Underwater soundscape of marine protected areas in the south Brazilian coast. **Marine Pollution Bulletin**, Elsevier, v. 105, n. 1, p. 65–72, 2016. Citado nas páginas [32](#) e [42](#).

SCARPELLI, M. D.; RIBEIRO, M. C.; TEIXEIRA, C. P. What does Atlantic Forest soundscapes can tell us about landscape? **Ecological Indicators**, Elsevier, v. 121, p. 107050, 2021. Citado nas páginas [46](#), [118](#) e [138](#).

SCHLÜTER, J. Bird identification from timestamped, geotagged audio recordings. In: **CLEF (Working Notes)**. [S.l.: s.n.], 2018. Citado na página [71](#).

SCHUMACHER, T.; STROHMAIER, M.; LEMMERICH, F. A comparative evaluation of quantification methods. **arXiv preprint arXiv:2103.03223**, 2021. Citado na página [62](#).

SERVICK, K. Eavesdropping on Ecosystems. **Science (New York, N.Y.)**, American Association for the Advancement of Science, v. 343, n. February, p. 834–837, 2014. Citado nas páginas [32](#) e [45](#).

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge University Press, 2014. Citado nas páginas [31](#), [46](#) e [52](#).

SHANNON, G.; MCKENNA, M. F.; ANGELONI, L. M.; CROOKS, K. R.; FRISTRUP, K. M.; BROWN, E.; WARNER, K. A.; NELSON, M. D.; WHITE, C.; BRIGGS, J. *et al.* A synthesis of two decades of research documenting the effects of noise on wildlife. **Biological Reviews**, Wiley Online Library, v. 91, n. 4, p. 982–1005, 2016. Citado na página [44](#).

SHARAWY, M.; SAYED, N. A. E.; ZAYED, H.; ABDEL-RAHIM, N. M.; SHALTOUT, A. Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: case study el Burullus Lake. **International Journal of Environmental Science and Development**, v. 7, n. 4, p. 295, 2016. Citado na página [51](#).

SHIU, Y.; PALMER, K.; ROCH, M. A.; FLEISHMAN, E.; LIU, X.; NOSAL, E.-M.; HELBLE, T.; CHOLEWIAK, D.; GILLESPIE, D.; KLINCK, H. Deep neural networks for automated detection of marine mammal species. **Scientific reports**, Nature Publishing Group, v. 10, n. 1, p. 1–12, 2020. Citado nas páginas [81](#), [82](#), [83](#), [85](#) e [157](#).

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014. Citado na página [74](#).

- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 54.
- STOWELL, D. Computational bioacoustics with deep learning: a review and roadmap. **PeerJ**, PeerJ Inc., v. 10, p. e13152, 2022. Citado nas páginas 33, 65, 84 e 92.
- STOWELL, D.; PLUMBLEY, M. D. Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. **PeerJ**, PeerJ Inc., v. 2, p. e488, 2014. Citado nas páginas 42 e 46.
- STROUT, J.; ROGAN, B.; SEYEDNEZHAD, S. M.; SMART, K.; BUSH, M.; RIBEIRO, E. Anuran call classification with deep learning. In: IEEE. **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2017. p. 2662–2665. Citado nas páginas 33, 39, 44, 59, 72, 73, 83, 84, 85 e 157.
- SUEUR, J.; AUBIN, T.; SIMONIS, C. Seewave, a free modular tool for sound analysis and synthesis. **Bioacoustics**, Taylor & Francis, v. 18, n. 2, p. 213–226, 2008. Citado nas páginas 99 e 121.
- SUEUR, J.; FARINA, A.; GASC, A.; PIERETTI, N.; PAVOINE, S. Acoustic indices for biodiversity assessment and landscape investigation. **Acta Acustica United with Acustica**, S. Hirzel Verlag, v. 100, n. 4, p. 772–781, 2014. Citado na página 45.
- SUEUR, J.; PAVOINE, S.; HAMERLYNCK, O.; DUVAIL, S. Rapid acoustic survey for biodiversity appraisal. **PloS One**, Public Library of Science, v. 3, n. 12, p. e4065, 2008. Citado na página 46.
- SUEZA, L. **XC452754**. 2019. Disponível em: <<https://www.xeno-canto.org/452754>>. Acesso em: 08/16/2020. Citado na página 90.
- SWACKHAMER, J. **XC521789**. 2017. Disponível em: <[www.xeno-canto.org/521789](http://www.xeno-canto.org/521789)>. Acesso em: 08/18/2020. Citado na página 90.
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9. Citado na página 56.
- SZEGEDY, C.; VANHOUCHE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 2818–2826. Citado nas páginas 56, 122 e 140.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining. 1st**. [S.l.]: Boston: Pearson Addison Wesley. XXI, 2005. Citado nas páginas 57, 102 e 143.
- THOMAS, M.; MARTIN, B.; KOWARSKI, K.; GAUDET, B.; MATWIN, S. Marine mammal species classification using convolutional neural networks and a novel acoustic representation. In: SPRINGER. **Joint European Conference on Machine Learning and Knowledge Discovery in Databases**. [S.l.], 2019. p. 290–305. Citado nas páginas 39, 45, 78, 79, 80, 83, 84, 85, 87, 89, 100, 122, 140 e 157.

- TIELEMAN, T.; HINTON, G. *et al.* Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. **COURSERA: Neural networks for machine learning**, v. 4, n. 2, p. 26–31, 2012. Citado na página 55.
- TOWSEY, M.; WIMMER, J.; WILLIAMSON, I.; ROE, P. The use of acoustic indices to determine avian species richness in audio-recordings of the environment. **Ecological Informatics**, Elsevier, v. 21, p. 110–119, 2014. Citado na página 45.
- TOWSEY, M. W. The calculation of acoustic indices derived from long-duration recordings of the natural environment. 2017. Disponível em: <<https://eprints.qut.edu.au/110634/>>. Citado na página 46.
- VERONESI, F. **Rufous-collared Sparrow**. 2010. Disponível em: <<https://flic.kr/p/7GUmWX>>. Acesso em: 08/19/2020. Citado na página 90.
- VILLANUEVA-RIVERA, L.; PIJANOWSKI, B.; DOUCETTE, j.; PEKIN, B. A primer of acoustic analysis for landscape ecologists. **Landscape Ecology**, Springer, v. 26, n. 9, p. 1233–1246, 2011. Citado na página 46.
- WANG, J.; PEREZ, L. *et al.* The effectiveness of data augmentation in image classification using deep learning. **Convolutional Neural Networks Vis. Recognit**, v. 11, p. 1–8, 2017. Citado nas páginas 49, 98, 120 e 139.
- WANG, Y.; GETREUER, P.; HUGHES, T.; LYON, R. F.; SAUROUS, R. A. Trainable frontend for robust and far-field keyword spotting. In: IEEE. **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2017. p. 5670–5674. Citado nas páginas 40, 41, 122 e 134.
- WARD, M. O.; GRINSTEIN, G.; KEIM, D. **Interactive data visualizaton: foundations, techniques, and applications**. 2. ed. [S.l.]: CRC Press, 2015. 578 p. ISBN 978-1482257373. Citado na página 36.
- WELCH, P. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. **IEEE Transactions on Audio and Electroacoustics**, IEEE, v. 15, n. 2, p. 70–73, 1967. Citado na página 41.
- WIKIAVES. **A vocalização das aves**. 2020. Disponível em: <[https://www.wikiaves.com.br/wiki/a\\_vocalizacao\\_das\\_aves](https://www.wikiaves.com.br/wiki/a_vocalizacao_das_aves)>. Acesso em: 08/19/2020. Citado na página 89.
- XIE, J.; TOWSEY, M.; ZHANG, J.; DONG, X.; ROE, P. Application of image processing techniques for frog call classification. **2015 IEEE International Conference on Image Processing (ICIP)**, p. 4190–4194, 2015. Citado na página 44.
- XIE, J.; ZHU, M.; HU, K.; ZHANG, J.; HINES, H.; GUO, Y. Frog calling activity detection using lightweight CNN with multi-view spectrogram: a case study on kroombit tinker frog. **Machine Learning with Applications**, Elsevier, v. 7, p. 100202, 2022. Citado nas páginas 73, 74, 75, 83, 84, 85, 87, 89, 125 e 157.
- ZAGORUYKO, S.; KOMODAKIS, N. Wide residual networks. **arXiv preprint arXiv:1605.07146**, 2016. Citado na página 71.

ZBONTAR, J.; JING, L.; MISRA, I.; LECUN, Y.; DENY, S. Barlow twins: Self-supervised learning via redundancy reduction. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2021. p. 12310–12320. Citado nas páginas [60](#), [150](#) e [152](#).

ZHANG, R.; ZHU, J.-Y.; ISOLA, P.; GENG, X.; LIN, A. S.; YU, T.; EFROS, A. A. Real-time user-guided image colorization with learned deep priors. **arXiv preprint arXiv:1705.02999**, 2017. Citado na página [59](#).

ZHU, X.; GOLDBERG, A. B. Introduction to semi-supervised learning. **Synthesis lectures on artificial intelligence and machine learning**, Morgan & Claypool Publishers, v. 3, n. 1, p. 1–130, 2009. Citado nas páginas [47](#) e [48](#).

ZNIDERSIC, E.; TOWSEY, M.; ROY, W. K.; DARLING, S. E.; TRUSKINGER, A.; ROE, P.; WATSON, D. M. Using visualization and machine learning methods to monitor low detectability species - The Least Bittern as a case study. **Ecological Informatics**, Elsevier, v. 55, p. 101014, 2020. Citado na página [45](#).

