



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Química

LETICIA TESSARINI

CLASSIFICAÇÃO DE RESINAS PLÁSTICAS PARA RECICLAGEM  
USANDO PROCESSAMENTO DE SINAIS ACÚSTICOS E REDES  
NEURAS ARTIFICIAIS

CLASSIFICATION OF PLASTIC RESINS FOR RECYCLING USING  
ACOUSTIC SIGNAL PROCESSING AND ARTIFICIAL NEURAL  
NETWORKS

Campinas, SP

2022

LETICIA TESSARINI

CLASSIFICAÇÃO DE RESINAS PLÁSTICAS PARA RECICLAGEM  
USANDO PROCESSAMENTO DE SINAIS ACÚSTICOS E REDES  
NEURAIS ARTIFICIAIS

Dissertação apresentada à Faculdade de Engenharia Química da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em Engenharia Química.

Orientadora: Prof.<sup>a</sup>. Dr.<sup>a</sup> Ana Maria Frattini Fileti

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL  
DA DISSERTAÇÃO/TESE DEFENDIDA PELO ALUNO  
LETICIA TESSARINI, E ORIENTADA PELA PROF(A).  
DR(A). ANA MARIA FRATTINI FILETI

Campinas, SP

2022

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

T284c Tessarini, Letícia, 1996-  
Classificação de resinas plásticas para reciclagem usando processamento de sinais acústicos e redes neurais artificiais / Letícia Tessarini. – Campinas, SP : [s.n.], 2022.

Orientador: Ana Maria Frattini Fileti.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Plásticos. 2. Reciclagem. 3. Processamento de sinais. 4. Classificação. 5. Redes neurais artificiais. I. Fileti, Ana Maria Frattini, 1965-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Química. III. Título.

**Informações Complementares**

**Título em outro idioma:** Classification of plastic resins for recycling using acoustic signal processing and artificial neural networks

**Palavras-chave em inglês:**

Plastic

Recycling

Signal processing

Classification

Artificial neural network

**Área de concentração:** Engenharia Química

**Titulação:** Mestra em Engenharia Química

**Banca examinadora:**

Ana Maria Frattini Fileti [Orientador]

Andrea Valdman

Rejane Barbosa Santos

**Data de defesa:** 12-12-2022

**Programa de Pós-Graduação:** Engenharia Química

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0001-8158-0881>

- Currículo Lattes do autor: <https://lattes.cnpq.br/890258534543840>

Dissertação de Mestrado defendida por Leticia Tessarini e aprovada em 12 de dezembro de 2022 pela banca examinadora constituídas pelos doutores:

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Ana Maria Frattini Fileti

FEQ/UNICAMP (Orientadora)

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Andrea Valdmam

Escola de Química/UFRJ

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Rejane Barbosa Santos

IFSULDEMINAS/Campus Pouso Alegre

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Dedico aos meus pais que sempre me apoiaram e me incentivaram para realização do mestrado e conclusão deste trabalho.

## AGRADECIMENTOS

Agradeço à Deus por permitir que eu realizasse este trabalho, com paciência e esforço.

Aos meus pais, Carlos e Regina e minha irmã Carol que sempre me apoiaram e auxiliaram em cada momento da minha vida. E tiveram paciência comigo estudando em casa.

A minha orientadora Ana Maria Frattini Fileti, que mesmo com a pandemia e desafios de realizar uma pesquisa a distância, aceitou me orientar, acreditou que eu era capaz e me auxiliou em todos os momentos necessários para elaboração deste trabalho.

Aos professores e colegas que tive aula durante o curso de mestrado, que de alguma forma colaboraram com a realização do trabalho.

Ao Salim, amigo do meu pai, que concedeu sua oficina e o equipamento morsa para que pudesse realizar parte dos experimentos.

Aos meus amigos que também sempre me apoiaram e tiveram paciência comigo quando não pude estar presente.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## RESUMO

Com o aumento do consumo de materiais e embalagens de origem plástica e com uma variedade de resinas que existem no mercado, há a necessidade de reciclar esses materiais corretamente, e para isso, necessita-se da separação das resinas. Normalmente, a classificação das mesmas é realizada manualmente pelos funcionários das cooperativas. Em decorrência do alto volume e da falta da simbologia nos materiais despedaçados, a identificação é difícil e as resinas acabam tomando um destino inadequado. Considerando a importância de ajudar a resolver esse problema de identificação das resinas plásticas, constatou-se a possibilidade de realizar essa classificação usando sinais do áudio quando o material é amassado. São geradas ondas sonoras diferentes para cada resina, permitindo assim a classificação e separação. Este trabalho teve como objetivo gravar os materiais sendo amassados, com amostras individuais e em conjunto em bags de pano, para que não interferisse no som dos plásticos. Com isso, um banco de dados foi criado, analisado e convertido do domínio do tempo para domínio da frequência, com a transformada rápida de Fourier (FFT). As características dos dados foram extraídas com a técnica coeficientes cepstrais de frequência mel (MFCC), que gerou um arquivo com os gêneros “categorias de resinas plásticas recicláveis”, os vetores do MFCC para cada segmento e os rótulos sendo as categorias pertencentes a cada resina, inseridos como dados de entrada. Foram utilizadas a rede neural convolucional (RNC), um tipo *feedforward* utilizado no processamento digital de imagens, e a rede neural recorrente LSTM (*long short-term memory*), que processa não apenas um único ponto, mas também sequências de dados inteiras onde a ordem é importante. Foram realizados testes com os modelos das redes neurais, com diferentes números máximos de iterações no treinamento. O desempenho das redes foi analisado por métricas, como acurácia, perda, e matriz confusão. Para a validação, foram inseridos novos dados que não foram vistos no treinamento e, posteriormente, dados nos quais as resinas foram amassadas em uma velocidade diferente da treinada. Para dados amassados com a mesma velocidade, a rede LSTM apresentou um desempenho melhor que a rede RNC, sendo alcançada uma acurácia de 97%, por ser adequada para dados sequenciais. No entanto, ao inserir novos dados somente no conjunto de teste, amassados de uma forma mais lenta do que os dados de treinamento, o desempenho da rede LSTM se deteriorou, enquanto a RNC atingiu uma porcentagem ainda aceitável. Sendo essa uma condição desfavorável e muito provável de ocorrer no dia-a-dia das cooperativas, conclui-se que o uso da RNC é o mais conveniente nestes ambientes.

Palavras-chave: plástico; reciclagem; processamento de sinais; classificação; rede neural artificial.

## ABSTRACT

With the increase in the consumption of materials and packaging of plastic origin and with a variety of resins that exist on the market, there is a need to recycle these materials correctly, and for that, it is necessary to separate the resins. Normally, this classification is performed manually by the employees of the cooperatives. Due to the high volume and lack of symbology in the broken materials, identification is difficult and the resins end up taking an inappropriate destination. Considering the importance of helping to solve this problem of identifying plastic resins, we found the possibility of performing this classification using audio signals when the material is kneaded. Different sound waves are generated for each resin, thus allowing classification and separation. This work aimed to record the materials being kneaded, using individual samples or a set of pieces in cloth bags, so that it would not interfere with the sound of the plastics. In this way, a database was created, analyzed and converted from the time domain to the frequency domain, with the fast Fourier transform (FFT). The data characteristics were extracted with the Mel frequency cepstral coefficients (MFCC) technique, which generated a file with the genres “recyclable plastic resin categories”, the MFCC vectors for each segment and the labels being the categories belonging to each resin, entered as input data. The convolutional neural network (CNN), a type of feedforward used in digital image processing, and the recurrent neural network LSTM (long short-term memory) were employed. The last one processes not only a single point, but the entire data sequence where the order is important. Tests were performed with models of neural networks, with different maximum iterations allowed in the training procedure. Network performance was analyzed by using metrics such as accuracy, loss, and confusion matrix. For validation, new data - not seen in the training – were inputted and, later, data in which the resins were kneaded at a different speed than the trained one. For squashed data with the same speed, the LSTM network performed better than the RNC network, reaching an accuracy of 97%, as it is suitable for sequential data. However, when entering new data only in the test set, crumpled more slowly than the training data, the performance of the LSTM network deteriorated, while the RNC reached a still acceptable percentage. Since this is an unfavorable condition very likely to occur day-to-day in cooperatives, we concluded that the use of the CNN is the most convenient for these environments.

Keywords: plastic; recycling; signal processing; classification; artificial neural network.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 3.1: Porcentagem da origem das matérias-primas que chegam até as usinas de reciclagem.....  | 28 |
| Figura 3.2: Fluxograma dos quatro tipos de reciclagem plástica.....  | 30 |
| Figura 3.3: Símbolos de identificação dos materiais plásticos segundo a norma ABNT NBR 13230.....  | 31 |
| Figura 3.4: Embalagens fabricadas por mistura de resinas: (a) mistura PEAD/PEBD com símbolo normalizado (b) mistura PEAD/PEBD com símbolo normalizado em embalagem com dimensões reduzidas (c) símbolo “7 – Outros” indicando qual resina utilizou.....  | 32 |
| Figura 3.5: Comparativo dos filtros no sinal (a) sinal de entrada original (b) saída do sinal com o filtro passa-baixa com $f_c$ de 80 Hz (c) saída do sinal com o filtro passa-alta com $f_c$ de 150 Hz (d) saída do sinal com o filtro passa-banda (e) saída do sinal com o filtro rejeita-banda limite mínimo de 80 Hz e o máximo 150 Hz..... | 35 |
| Figura 3.6: Sinal original no domínio do tempo e o sinal aplicado com a transformada de Fourier no domínio da frequência.....  | 36 |
| Figura 3.7: Banco de filtros triangulares da escala Mel.....   | 39 |
| Figura 3.8: Estrutura de uma rede neural biológica.....  | 42 |
| Figura 3.9: Representação do modelo neural artificial.....   | 42 |
| Figura 3.10: Exemplo de funções de ativação.....   | 43 |
| Figura 3.11: Arquitetura básica de uma RNC.....  | 45 |
| Figura 3.12: Extração das características com filtro detecção de borda.....  | 46 |
| Figura 3.13: Exemplo utilização de padding.....  | 47 |
| Figura 3.14: Exemplo camada max-pooling.....   | 47 |
| Figura 3.15: Funcionamento de uma RNN.....   | 50 |
| Figura 3.16: Módulo de uma RNA comum com apenas uma camada.....  | 51 |
| Figura 3.17: Módulo da rede LSTM com quatro camadas.....   | 51 |
| Figura 3.18: Representação da célula LSTM (a) forget gate (b) input gate (c) cell gate e (d) output gate.....  | 53 |
| Figura 4.1: Amostras cortadas e amassadas em conjunto para caso 1 (a) Amostras de PET cortadas em aproximadamente 10 centímetro de comprimento e largura dos materiais (b) Amostras cortadas dentro do saco de pano.....   | 56 |
| Figura 4.2: Amostras inteiras das respectivas categorias: PET, PEAD, PVC, PEBD, PP, PS e Outros para caso 2.....   | 56 |
| Figura 4.3: Esquema representativo do processo de captura do áudio do caso 1 e caso 3.....   | 57 |
| Figura 4.4: Esquema representativo do processo de captura de áudio para o caso 2.....  | 57 |
| Figura 4.5: Fluxograma da etapa de pré-processamento dos dados.....  | 59 |
| Figura 4.6: Fluxograma da etapa de processamento dos dados.....  | 60 |
| Figura 4.7: Esquema ilustrativa da arquitetura da RNC utilizada na pesquisa.....   | 63 |
| Figura 4.8: Esquema ilustrativa da arquitetura da LSTM utilizada na pesquisa.....  | 66 |

|  |     |
|--|-----|
| Figura 5.1: Forma de onda original do caso 1 (amassados em conjunto) e do caso 2 (amassados por peça) para todas as categorias.....  | 68  |
| Figura 5.2: Forma de onda original do caso 1 (amassados em conjunto) e do caso 2 (amassados por peça) para todas as categorias.....  | 70  |
| Figura 5.3: Gráfico da transformada rápida de Fourier da categoria PET (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.....    | 70  |
| Figura 5.4: Gráfico da transformada rápida de Fourier da categoria PEAD (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.....   | 71  |
| Figura 5.5: Espectrograma STFT tempo x frequência x amplitude para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).....                                      | 73  |
| Figura 5.6: Log espectrograma tempo x frequência x dB para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).....  | 76  |
| Figura 5.7: Espectrograma MFCC das sete categorias para caso 1 (amassadas em conjunto) e caso 2 (amassadas por peça).....  | 81  |
| Figura 5.8: Avaliação da acurácia e perda do treinamento e teste da rede neural convolucional. (a) caso 1 – amostras amassadas em conjunto (b) caso 2 – amostras amassadas por peça.....     | 82  |
| Figura 5.9: Matriz confusão da RNC para (a) caso 1 – amassadas em conjunto e (b) caso 2 – amassadas por peça.....  | 85  |
| Figura 5.10: Matriz confusão da rede neural recorrente LSTM para (a) caso 1 – amassadas em conjunto e (b) caso 2 – amassadas por peça.....   | 87  |
| Figura 5.11: Matriz confusão do conjunto de teste do caso 3 (amassadas de forma mais lenta) para (a) RNC e (b) LSTM.....   | 89  |
| Figura 0.1: Gráfico da transformada rápida de Fourier da categoria PVC (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.....    | 115 |
| Figura 0.2: Gráfico da transformada rápida de Fourier da categoria PEBD (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.....   | 115 |
| Figura 0.3: Gráfico da transformada rápida de Fourier da categoria PP (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta...       | 116 |
| Figura 0.4: Gráfico da transformada rápida de Fourier da categoria PS (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta...       | 116 |
| Figura 0.5: Gráfico da transformada rápida de Fourier da categoria OUTROS (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta..... | 117 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 5.1: Valores dos parâmetros da arquitetura da rede neural convolucional.....                  | 79 |
| Tabela 5.2: Resultado do treinamento e teste da rede neural convolucional.....                       | 80 |
| Tabela 5.3: Índice esperado e índice previsto do conjunto de teste da rede neural convolucional..... | 82 |
| Tabela 5.4: Valores dos parâmetros da arquitetura da rede neural recorrente do tipo LSTM...          | 83 |
| Tabela 5.5: Resultado do treinamento e teste da rede neural recorrente LSTM.....                     | 84 |
| Tabela 5.6: Índice esperado e índice previsto da rede neural recorrente LSTM.....                    | 86 |
| Tabela 5.7: Resultado do treinamento e teste das redes neurais para o caso 3.....                    | 88 |
| Tabela 5.8: Resumo dos resultados dos casos do treinamento e teste das redes neurais.....            | 89 |
| Tabela 5.9: Maiores e menores porcentagens de acerto da matriz confusão.....                         | 90 |

## LISTA DE QUADROS

|   |    |
|---|----|
| Quadro 3.1: Exemplos de aplicações das resinas plásticas antes e após reciclagem..... | 32 |
|---|----|

## LISTA DE ABREVIATURAS E SIGLAS

$f_s$  - Frequência de amostragem

1D – Unidimensional

2D – Bidimensionais

3D – Tridimensionais

ABNT - Associação Brasileira de Normas Técnicas

ABS - Acrilonitrila butadieno estireno

ADC - Conversor analógico-digital

ATR - Reflectância total atenuada

CNs 1D - Redes neurais convolucionais unidimensionais

DAC - Conversor digital-analógico

dB - Decibéis

DCT - Transformada discreta de cosseno

DFT - Transformada discreta de Fourier

$f_c$  - Frequência de corte

FFT - Transformada rápida de Fourier

FRBFNN - Fuzzy Radial Basis Function Neural Networks

FTIR - Infravermelho por transformada de Fourier

GPU - Unidade de Processamento Gráfico

Hz - Hertz

IA – Inteligência Artificial

IoT - Internet das Coisas

JSON - *JavaScript Object Notation*

KNN - *K-Nearest Neighbors*

LSTM - Long short-term memory

MD - Multidimensionais

MFCC - Mel-frequency cepstral coefficients

ML - Machine Learning

MLP - Rede neural de multicamadas

msec – Milissegundos

NBR - Normas Brasileiras

NFC - Comunicação de campo próximo

NIR - Espectroscopia de infravermelho próximo

PEAD - Polietileno de alta densidade

PEBD - Polietileno de baixa densidade

PELBD - Polietileno linear de baixa densidade

PET - Polietileno tereftalato

PET - Programa de Educação Tutorial

PP - Polipropileno

PS – Poliestireno

PVC - Policloreto de vinila

RAM - *Random Access Memory*

ReLU - Função linear retificada

RNA - Rede Neurais Artificiais

RNC - Rede neural convolucional

RNN - *Recurrent neural network*

SOM - mapa de auto-organização

SPI - Sociedade das Indústrias de Plásticos

STFT - Transformada de Fourier de tempo curto

SVM - Support Vector Machine

SVM - *Support Vector Machines*

Tanh - Tangente hiperbólica

TPU - *Tensor Processing Unit*

WAV - *Waveform Audio File Format*

XML - *Extensible Markup Language*

## LISTA DE SÍMBOLOS

$\int$  - Integral

$\infty$  - Infinito

$\xi$  - Csi

$\geq$  - Maior ou igual que

$\pi$  - Pi

$<$  - Menor que

$\Sigma$  ou  $\sigma$  - Sigma

$\Phi$  - Fi

$\in$  - Pertence a

## SUMÁRIO

|   |    |
|---|----|
| Capítulo 1 – INTRODUÇÃO.....                                | 19 |
| 1.1 Justificativas.....                                     | 21 |
| 1.2 Objetivos.....  | 22 |
| 1.3 Organização do trabalho.....                            | 22 |
| Capítulo 2 – REVISÃO DE LITERATURA.....                     | 23 |
| 2.1 Trabalhos relacionados.....                             | 23 |
| 2.2 Conclusão.....  | 26 |
| Capítulo 3 – FUNDAMENTAÇÃO TEÓRICA.....                     | 27 |
| 3.1 Sistema de reciclagem dos plásticos.....                | 27 |
| 3.2 Processamento de sinais.....                            | 33 |
| 3.2.1. Transformada de Fourier.....                         | 36 |
| 3.2.2. Coeficientes cepstrais de frequência mel (MFCC)..... | 38 |
| 3.3 Aprendizado de Máquinas.....                            | 40 |
| 3.3.1 Redes Neurais Artificiais (RNA).....                  | 41 |
| 3.3.1.1 Rede neural convolucional.....                      | 45 |
| 3.3.1.2 Redes neurais recorrentes.....                      | 49 |
| 3.3.1.2.1 Redes neurais recorrentes LSTM.....               | 50 |
| Capítulo 4 - METODOLOGIA.....                               | 55 |
| 4.1 Amostragem.....   | 55 |
| 4.1.1 Sistema de aquisição de dados.....                    | 57 |
| 4.2. Preparação dos sinais de áudio.....                    | 58 |
| 4.2.1. Pré-processamento.....                               | 58 |
| 4.2.2. Processamento.....                                   | 60 |
| 4.3. Classificação.....                                     | 61 |
| 4.3.1. Redes Neurais Convolucionais (RNC).....              | 62 |
| 4.3.2. Redes Neurais Recorrentes LSTM.....                  | 65 |
| 4.3.3. Validação.....                                       | 67 |
| Capítulo 5 – RESULTADOS E DISCUSSÕES.....                   | 68 |
| 5.1 Pré-processamento.....                                  | 68 |
| 5.2 Processamento.....                                      | 78 |

|  |     |
|--|-----|
| 5.3 Classificação.....   | 78  |
| 5.3.1 Rede Neural Convolutacional.....   | 78  |
| 5.3.2 Rede Neural Recorrente LSTM.....   | 83  |
| 5.3.3 Validação.....   | 88  |
| 5.3.4 Resumo dos resultados.....   | 89  |
| Capítulo 6 - CONCLUSÕES.....   | 91  |
| REFERÊNCIAS.....   | 93  |
| APÊNDICE A – Programa pré-processamento dos dados de áudio para taxa de amostragem de 44.100 Hz em Python desenvolvido no ambiente Google Colab..... | 102 |
| APÊNDICE B – Programa processamento dos sinais de áudio utilizando MFCC.....   | 104 |
| APÊNDICE C - Programa de treinamento do modelo de rede neural convolutacional em Python desenvolvido no ambiente Google Colab.....                   | 106 |
| APÊNDICE D – Programa para o modelo rede neural recorrente do tipo LSTM em Python desenvolvido no ambiente Google Colab.....                         | 111 |
| APÊNDICE E – Restante do gráfico da transformada de Fourier paras as categorias PVC, PEBD, PP, PS e Outros.....                                      | 115 |

## Capítulo 1 - INTRODUÇÃO

O crescimento populacional e a fácil disponibilidade de produtos industrializados levaram a um aumento significativo do consumo de embalagens, gerando resíduos sólidos de difícil degradação. O gerenciamento de resíduos sólidos é inadequado, causando alguns problemas para a saúde pública e para o meio ambiente (FERREIRA; KINTSCHNER; SUGAHARA, 2022). De acordo com PEREIRA e Dantas (2010), o plástico não é o vilão da relação material e sustentabilidade, mas sim o ciclo de vida do produto que é inadequado ao consumo sustentável. A concepção racional do produto é essencial para o reaproveitamento das matérias-primas. Para produtos descartáveis, como filme plástico, embalagens de isopor e sacolas plásticas, a reciclagem de matérias-primas é mais difícil, pois há alto risco de perda devido ao contato direto com alimentos e outros produtos químicos. Dessa maneira, as empresas que produzem plástico enfrentam o enorme desafio de garantir a sustentabilidade de uma indústria cujos produtos levam mais de 300 anos para se degradar uma vez descartados. Diante dessa situação, as indústrias de plásticos devem tomar ações estratégicas para garantir o desenvolvimento sustentável (SANTOS; AGNELLI; MANRICH, 2004).

Assim, entram as cooperativas de reciclagem que desempenham um papel vital no ciclo de vida dos aterros sanitários, pois essas cooperativas recebem o material e os destinam para o local adequado, reduzindo assim o impacto ambiental do resíduo. Normalmente, as atividades operacionais de reciclagem começam com a coleta de resíduos sólidos urbanos com potencial de reciclagem nas residências e, em seguida, transportam os resíduos para cooperativas, onde são separados, prensados, pesados, armazenados e vendidos. No entanto, as cooperativas de reciclagem muitas vezes apresentam baixo desempenho no processo de reciclagem devido à alta diversidade e os diferentes níveis de desempenho e organização, apresentando dificuldades no gerenciamento (FERREIRA; KINTSCHNER; SUGAHARA, 2022).

O plástico tornou-se um dos materiais mais produzidos no mundo nos últimos anos, por ter várias vantagens em relação a outros materiais, como ser leve, de baixo custo, vida útil estendida e resistente à degradação. Essas propriedades tornam o material versátil, aumentando seu uso e impulsionando a fabricação de embalagens plásticas, que conseqüentemente terá uma alta geração de resíduos (GONÇALVES, 2022, p. 16).

O crescimento acelerado dessa geração de resíduos, aliado à sua baixa biodegradabilidade, tornou o plástico um dos poluentes de maior preocupação globalmente. Mesmo o plástico sendo 100% reciclável, do total produzido no Brasil, recicla-se apenas

23,4% (Abiplastic, 2021). Os principais motivos para uma taxa baixa de reciclagem, é por conta da contaminação das resinas plásticas por materiais indesejados, principalmente de origem doméstica e a presença de uma grande variedade de plástico que compromete o processo de separação adequada desses materiais (PEREIRA; DANTAS, 2010). Estima-se que 24% do material triado pelas cooperativas de catadores brasileiros foi classificado como rejeito e não teve uma identificação (GONÇALVES, 2022).

Esforços ainda são necessários para melhorar a reciclagem de plásticos e, devido à grande variedade de resinas que compõem a maioria dos resíduos plásticos urbanos, a reciclagem desses materiais tornou-se um padrão de processo industrial econômico e ecologicamente correto (AHMAD, 2010, p. 1145). Para reciclar os plásticos de forma acessível, é necessário separar os resíduos precisamente antes de entrar na cadeia de reciclagem de acordo com sua categoria de resina para geração de produtos com valor agregado (COLTRO, 2008, p. 119).

A classificação é normalmente realizada pelos operadores da fábrica visual e manualmente conforme os recipientes de plástico percorrem a esteira transportadora. Essa triagem pode não ser adequada pelo alto volume e rotatividade de trabalhadores na linha e muitas vezes a falta da simbologia nas embalagens dificultam a separação consistente e reduzem a eficiência do processo (SCAVINO, 2009).

Há necessidade de melhorar as propriedades de separação dos materiais plásticos das cooperativas, aumentando assim a renda e criando condições para a qualidade de vida e trabalho dos cooperados (FERREIRA; KINTSCHNER; SUGAHARA, 2022). Além disso, tem poucos estudos sobre resíduos cooperativos no Brasil com foco nas embalagens plásticas, o maior componente de resíduos. Como as embalagens plásticas são produzidas e descartadas em massa, e existem na seção de rejeitos, ou seja, enviadas para disposição final, é necessário entender os desafios da reciclagem plástica e sua separação. A dificuldade de acesso a esses dados limita a percepção das proporções do problema e dificulta o desenvolvimento de estratégias para reduzir a produção de rejeitos (GONÇALVES, 2022, p. 17).

Existem várias técnicas para identificar as resinas, das mais simples e de menor custo - como teste de chama e densidade - mas que demandam tempo, assim como técnicas mais sofisticadas, usando equipamentos específicos, geralmente com custo mais alto, como: espectroscopia de raios infravermelhos, espectroscopia de ressonância magnética nuclear, absorção, fluorescência e métodos analíticos, como difração de raios-X (CANCIAM, 2020). Com o desenvolvimento de tecnologias, há muitos estudos envolvendo Inteligência Artificial (IA), como técnicas de *Machine Learning* (ML), alguns exemplos de algoritmos utilizados

para classificação, são *Support Vector Machine* (SVM) e Rede Neurais Artificiais (RNA) que envolvem imagens, vídeos, sons e processamento de fala.

Dentre as técnicas existentes para classificação, os métodos escolhidos no presente trabalho para identificação das resinas plásticas para reciclagem, foram dois modelos de rede neurais artificiais. Foram utilizados dados de áudio, que quando o material é amassado, geram ondas sonoras que diferem uma resina da outra e com as características extraídas pela técnica coeficientes cepstrais de frequência mel (MFCC). Essas características foram utilizadas como entrada nas redes, realizou-se o treinamento e teve a classificação das resinas. Analisou-se também, os sinais de áudio originais e no domínio da frequência, com a utilização da transformada rápida de Fourier (FFT), assim como os espectrogramas gerados pela técnica de extração. Com essas análises e técnicas conseguiu classificar as resinas plásticas com uma porcentagem aceitável, de modo a ajudar os trabalhadores das cooperativas nesse quesito de identificação e contribuir para que os materiais tomem o destino adequado na reciclagem.

### 1.1 Justificativas

O problema de pesquisa foi identificado por meio do Programa de Educação Tutorial (PET) da Engenharia Química Unicamp que tem parceria com catadores e cooperados de resíduos. Essas organizações possuem acordos com líderes do setor de embalagens para garantir que os resíduos sólidos gerados no país cheguem ao lugar correto.. Seu objetivo é incluir essas organizações na produção profissional, para poderem aumentar seus níveis de renda e emprego (Gaia Social). Foi relatado que um trabalhador que passava pelas cooperativas conseguia separar melhor os diferentes materiais plásticos, pois estava familiarizado com as diferentes categorias de resinas plásticas apenas ouvindo o som produzido ao amassar os materiais.

No entanto, atualmente, há uma dificuldade de separação de materiais plásticos para os outros trabalhadores dessas cooperativas, pois eles não conseguem identificar a qual categoria cada resina pertence, como o referido funcionário. Portanto, teve-se o incentivo de estudar os áudios das resinas plásticas ao amassar e utilizar os arquivos de áudio como dado de entrada em redes neurais artificiais para classificar as sete resinas plásticas. Assim, com o uso da Inteligência Artificial, as cooperativas de reciclagem podem utilizar o programa realizado no estudo para ajudarem a resolver o problema de classificação. Com isso, irá facilitar o trabalho dos funcionários e deixar o processo mais eficiente. E os materiais terão um destino adequado, aumentando a qualidade de reciclagem dos mesmos.

## 1.2 Objetivos

Considerando a importância da classificação de resinas plásticas para uma reciclagem mais eficiente, o presente trabalho teve como objetivo desenvolver um método para classificação dos plásticos através do sinal de áudio, com a utilização de redes neurais artificiais. Os dados experimentais foram obtidos por gravação do áudio dos plásticos ao serem amassados. Foram extraídas suas características e então utilizadas como entradas nas redes neurais artificiais, encarregadas pela classificação das sete resinas plásticas.

As seguintes etapas foram propostas para desenvolver o trabalho:

- Obter os dados de áudio pelo amassamento de sete resinas plásticas (PET, PEAD, PVC, PEBD, PP, PS e Outros) em diferentes condições, por um smartphone;
- Adquirir os áudios em uma situação desfavorável, onde as amostras foram amassadas de forma mais lenta e inseridas somente no conjunto de teste;
- Analisar as características dos sinais emitidos por meio do emprego da transformada rápida de Fourier (FFT) e extração pela técnica coeficientes cepstrais de frequência mel (MFCC);
- Desenvolver modelos neurais, usando rede neural convolucional (RNC) e rede neural recorrente LSTM (*Long short-term memory*) para classificação das resinas plásticas;
- Avaliar o treinamento e teste dos modelos usando critérios como: acurácia, perda e matriz confusão;
- Comparar os resultados das diferentes condições de amassamento para as redes neurais RNC e LSTM.

## 1.3 Organização do trabalho

O trabalho está organizado da seguinte forma: O Capítulo 2 apresenta uma revisão da literatura com os principais trabalhos desenvolvidos na área de estudo. O Capítulo 3 contém uma fundamentação teórica das técnicas utilizadas na pesquisa. No Capítulo 4 estão descritos os materiais e metodologia utilizada para o desenvolvimento do trabalho. No Capítulo 5 apresenta a discussão dos resultados obtidos e no Capítulo 6 têm a conclusão e sugestões para trabalhos futuros.

## Capítulo 2 – REVISÃO DE LITERATURA

Neste capítulo são apresentados os principais trabalhos da literatura desenvolvidos e relacionados com a classificação de resíduos, plásticos ou não plásticos, utilizando técnicas mais comuns e de aprendizado profundo, como processamento de imagens e sinais.

### 2.1 Trabalhos relacionados

Como o consumo de embalagens plásticas está aumentando exponencialmente nas últimas décadas e vendo que ainda há uma necessidade para aprimorar o processo de classificação desses materiais de forma mais rápida e menos dispendiosas para as cooperativas de reciclagem, vários estudos e pesquisas de técnicas envolvendo inteligência artificial, como *Machine Learning* estão sendo investigadas e implementadas para esse fim.

Em 2006, WAHAB *et al.* utilizaram vários modelos de redes neurais artificiais e um mapa de auto-organização (SOM) para identificar resinas PET (polietileno tereftalato) e não-PET. As amostras de plástico eram jogadas em uma tremonha e caíam em uma posição horizontal que eram transportadas ao longo da correia e utilizavam um sistema automatizado de espectroscopia de infravermelho próximo (NIR) que detectava o material e enviava para um identificador que maximizava a precisão e melhorava a qualidade, utilizando processamento de imagens e reconhecimento de padrões. As imagens foram comparadas com um banco de dados que continha todas as características conhecidas das garrafas. Porém, o sistema não conseguia identificar novas amostras, fora dessas características, quando apareciam na esteira.

SCAVINO *et al.* em 2009 também utilizaram essas técnicas de SOM e RNA para identificação de garrafas plásticas. As imagens capturadas na esteira transportadora eram digitalizadas e pré-processadas usando algoritmos de filtragem padrão e suas características eram extraídas. A extração das garrafas era realizada com braço pneumático, que era acionado conforme a categoria de plástico. As garrafas de PET permaneceriam na esteira transportadora, enquanto as garrafas que não eram PET, retiradas. Ao combinar todas essas tecnologias, conseguiram alcançar 97% de reconhecimento por imagem.

O método estudado por BORGES *et al.* (2015), para identificar automaticamente os materiais recicláveis por um sensor de metal e programar a coleta desses materiais. O processo de identificação foi conseguido colocando uma película fina de cristal de quartzo ou outro material cristalino no objeto a ser identificado, e a película fina oscilava em uma

frequência predeterminada. Quando um campo elétrico era aplicado para deformar o cristal, o mesmo acumulava carga em resposta ao estresse mecânico aplicado. Essa oscilação dependia da espessura do cristal de quartzo que poderia ser projetada de diferentes maneiras, resultando em diferentes frequências que eram associadas a um banco de dados conhecidos para caracterizar a frequências semelhantes aos materiais recicláveis. O sinal era enviado para um dispositivo móvel com um aplicativo, onde recebia as informações por reconhecimento de imagem ou comunicação de campo próximo (NFC). Essas informações podiam ser apenas sobre materiais reciclados ou sua localização e vibração.

ROH e OH (2018) utilizaram o método de infravermelho por transformada de Fourier (FTIR) com reflectância total atenuada (ATR) para identificação dos plásticos pretos. O método FTIR é usado para várias análises que diferem as estruturas moleculares dos polímeros. Para utilizar esta técnica, era necessário eliminar o ruído de fundo. Para o reconhecimento, os autores utilizaram três técnicas, a primeira era a correção da linha de base e detecção de pico, onde era aplicada uma correção utilizando múltiplas janelas de deslocamento da linha de base e dos postos de detecção de pico para poder extrair as características do espectro original. Esses espectros eliminados eram utilizados como variáveis de entrada em um algoritmo de classificação competitivo chamado FRBFNN (*Fuzzy Radial Basis Function Neural Networks*), classificando os plásticos em até três categorias, como PP (polipropileno), PS (poliestireno) e PET. A segunda técnica, consistia em combinar conhecimentos especializados e algoritmos de classificação inteligentes, os quais forneciam as posições dos picos característicos que identificava os plásticos. A última técnica, era utilizar a transformação difusa, que poderia extrair o espectro original e melhorar o desempenho do sistema. Em relação a recursos computacionais, a tecnologia baseada na transformação difusa foi melhor que o recurso técnico de primeiro tipo.

DHARMANA *et al.* (2019) perceberam que quando um material era triturado, também era possível realizar a classificação dos resíduos através de sinais de áudio gravados com microfone. As ondas sonoras obtidas diferenciavam materiais recicláveis, como plástico, alumínio, vidro e papel, fazendo com que a separação pudesse ser realizada. Para identificar cada material reciclável, era necessário analisar o sinal de áudio e utilizar a técnica MFCC para extração das características do sinal sonoro. Para isso, foi necessário dividir o sinal de áudio em pequenos segmentos, de modo a reduzir a distorção do sinal, diminuindo para zero. FFT era utilizado para converter cada segmento do sinal no domínio do tempo para o domínio da frequência, realizando assim a resposta de amplitude da intensidade do sinal no espectro. Após isso, realizava a classificação utilizando rede neural de multicamadas (MLP), a qual

possuía camadas ocultas entre camadas de entrada e saída. No início, a camada de rede de entrada foi fornecida com uma amostra de dados de treinamento. O erro quadrático foi calculado realizando uma soma dos valores previstos na saída da rede com os conhecidos e os valores dos pesos eram atualizados até que a soma do erro ficasse menor que o limite predefinido. O sistema com a técnica MFCC atingiu 85% de precisão.

Em 2020, LU *et al.* aplicaram redes neurais convolucionais unidimensionais (CNs 1D) para classificar resíduos na fonte com base em dados acústicos. Essa rede foi proposta para extrair automaticamente características de dados acústicos e alcançar alta precisão de classificação. Neste estudo, foram coletados 16 resíduos de diferentes tamanhos e condições, e um dispositivo de coleta de som foi projetado conforme a categoria de impacto de queda livre dos materiais. Todos os sons produzidos pelos resíduos foram registrados e inseridos na rede como dados. Antes da classificação sonora, todas as amostras foram agrupadas com o mesmo número de cada tipo em cada grupo. A rede é ajustada de forma supervisionada para minimizar o erro de previsão das etiquetas de resíduos. O método de validação cruzada foi usado para treinar e chegar a um modelo confiável e estável. Aplicou-se uma abordagem experimental ortogonal para estudar os efeitos dos hiperparâmetros RNC e encontrar a melhor combinação de parâmetros no modelo. O método proposto mostrou ser eficiente para ser aplicado a aterros de forma prática para separação automática de fontes de resíduos.

SÁNCHEZ *et al.* (2022) projetaram um sistema automatizado para classificar os elementos no processo de reciclagem. Isso foi feito aplicando técnicas de aprendizado profundo e processamento de áudio, além de interconectar bandas de frequência, braços robóticos e sensores de áudio através de vários protocolos de comunicação IoT (Internet das Coisas). Vidro e plástico foram os materiais classificados usando o som. Utilizaram apenas garrafas de gargalo padrão, pois o design da garrafa tem um diâmetro fixo. O layout do sistema incluiu um transportador de entrada que transporta as garrafas até o final da linha sendo parado por sensores ultrassônicos. O robô pegava a garrafa e seguia a trajetória determinada por todo o subsistema de som. Esse subsistema consistia em uma haste de metal e um microfone que trabalhavam juntos para gerar e gravar o som da garrafa. Visto que a classificação determinava se a garrafa era de vidro ou plástico, o robô colocava na rota de embarque apropriada. Dois métodos diferentes foram usados para classificação de sons e três modelos foram avaliados. O primeiro, utiliza uma RNC para classificação, dado que tanto o plástico quanto o vidro produzem sons com frequências específicas, que podem ser facilmente identificadas quando espectrogramas de mel são gerados a partir desses sons. Para o segundo modelo tem as mesmas entradas que o primeiro, mas o ruído de fundo foi filtrado. O terceiro

modelo é uma RNC que usa a extração de MFCC como entrada, o que apresentou melhores resultados.

## 2.2 Conclusão

Os trabalhos citados neste capítulo utilizam técnicas aprimoradas, como infravermelho com transformada de Fourier e refletância total, que requerem equipamentos específicos e podem acabar sendo dispendiosas. Outras pesquisas aplicadas a técnicas orientadas a dados, como o aprendizado profundo, podem resolver diversos problemas em sua totalidade, podendo extrair características dos dados brutos e utilizar métodos de regressão ou classificação para encontrar o destino correto. A maioria dos estudos utiliza o processamento de imagens para detectar resíduos e outros materiais, exigindo recursos computacionais adicionais para o processamento dos dados. E ao extrair recursos de imagens, como formas, bordas e cores, podem variar de amostra para amostra, criando desafios ao preparar esses conjuntos de dados para treinamento do modelo (LU *et al.*, 2020).

Portanto, algumas tentativas foram feitas no campo do processamento de áudio. Os estudos visam classificar uma variedade de materiais recicláveis, não apenas plásticos e os que se concentram apenas em resinas plásticas, detectam apenas dois grupos de PET e não-PET. Desta forma, a classificação das sete resinas plásticas (PET, PEAD, PVC, PEBD, PP, PS e Outros) para reciclagem utilizando aprendizagem de máquina com processamento de sinal ainda não foi estudada.

Como não encontrou trabalhos em que utilizavam processamento de áudio para classificação das sete resinas plásticas para reciclagem e analisando que as propriedades dessas resinas contêm ricas informações. Este trabalho estudou e entendeu os dados de áudio das resinas, no qual, os sinais foram processados, extraídas as características e classificados.

## Capítulo 3 – FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda as principais teorias estudadas para o desenvolvimento do trabalho e os métodos utilizados, envolvendo processo de reciclagem dos plásticos, aplicações das resinas, processamento de sinais de áudio - transformada rápida de Fourier e extração das características-, e também a aprendizagem de máquina para classificação, usando dois modelos de rede neural artificial: a convolucional e a recorrente LSTM.

### 3.1 Sistema de reciclagem dos plásticos

O consumo de embalagens plásticas tem aumentado nos últimos anos devido ao seu baixo peso, facilidade de fabricação e processamento, alta resistência ao impacto, baixo custo de energia para produção e transformação, bom isolante térmico e resistente à oxidação, não precisando de tratamento anticorrosivo (KIPPER; MÄHLMANN; RODRÍGUEZ, 2009). Com todas essas características e com uma vasta aplicação, estima-se que mais de 8,3 bilhões de toneladas métricas de plástico virgem tenham sido produzidas no mundo até 2017 e esse número só aumenta ao passar dos anos (GEYER; JAMBECK; LAW, 2017).

Em uma sociedade de consumo influenciada pela industrialização e pelo crescimento populacional, há impactos socioambientais, como excessos de resíduos plásticos urbanos, um dos maiores desafios da atualidade. A fim de diminuir esta consequência e contribuir com uma economia viável, entra a reciclagem desses resíduos, que apresenta uma economia de matéria-prima (KIPPER; MÄHLMANN; RODRÍGUEZ, 2009).

No Brasil, o plástico é um dos materiais menos reciclados, mesmo sua reinserção na cadeia produtiva apresentando características interessantes e para aumentar as taxas de reciclagem de forma sustentável, é necessário investimento em tecnologias inovadoras, que sejam viáveis economicamente. A respeito disso, a reciclagem começa desde a coleta seletiva adequada, o ponto-chave para o retorno desses materiais para reutilização ou reaproveitamento em novos processos produtivos, propagando a cadeia de suprimentos reversa. Essa diz respeito à logística como etapas de atividades econômicas e processamento necessários para converter matérias-primas em produtos visando a satisfação do cliente (KIPPER; MÄHLMANN; RODRÍGUEZ, 2009).

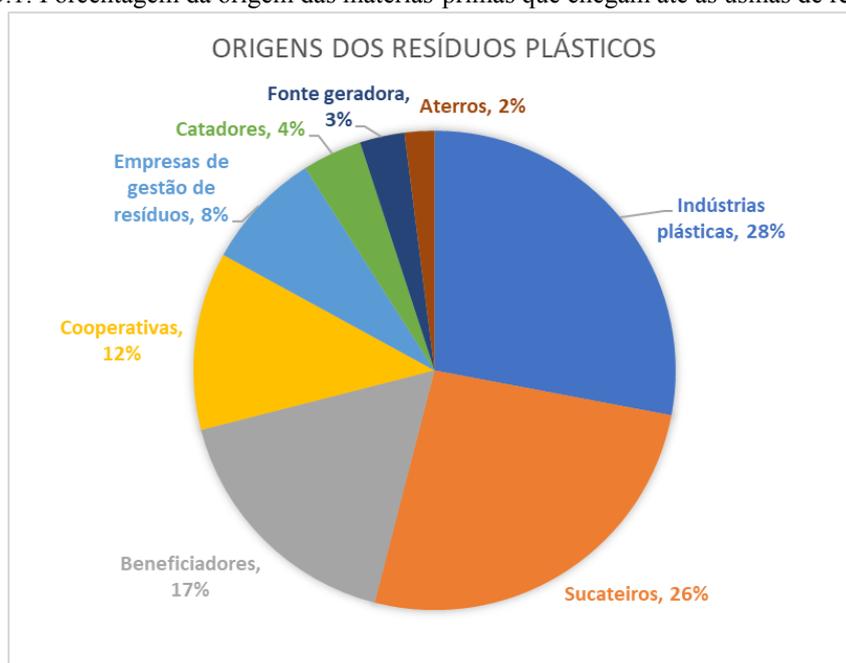
Assim como em outros países desenvolvidos, a rentabilidade do mercado brasileiro de reciclagem em embalagens plásticas indica a atratividade das iniciativas comerciais do setor, com consequências socioeconômicas diretas associadas à melhoria da qualidade de vida da

população, geração de renda, conservação dos recursos naturais e diminuição dos problemas ambientais (FORLIN; FARIA, 2002).

Esforços em grande escala ainda são necessários para melhorar o efeito de reciclagem dos plásticos, devido à grande variedade de resinas que compõem a maioria dos resíduos plásticos municipais. A reciclagem desses materiais tornou-se um padrão de processo industrial econômico e ambientalmente correto. Para reciclar os plásticos de forma acessível, é necessário separar os resíduos economicamente e automática conforme a sua classificação para que se possa gerar produtos de valores agregados (AHMAD, 2010).

Os resíduos plásticos têm origem por meio de indústrias plásticas, sucateiros, beneficiadores, cooperativas, empresas de gestão de resíduos, catadores, fonte geradoras e aterros, conforme mostra o gráfico da Figura 3.1, com suas respectivas porcentagens (Abiplastic, 2021).

Figura 3.1: Porcentagem da origem das matérias-primas que chegam até as usinas de reciclagem.



Fonte: Adaptado de Abiplastic, 2021.

Nas cooperativas, a reciclagem de plásticos pode ser entendida como a implementação de processos e tecnologias para otimizar o uso de energia, matérias-primas, produtos e materiais utilizados na fabricação de embalagens, mantendo com segurança sua função com novas matérias-primas ou produtos, com conceitos de impacto social, saúde e ambiental (FORLIN; FARIA, 2002). Nesse cenário, caminhos potencial ou economicamente viáveis para a reciclagem de materiais plásticos envolvem quatro principais formas de

reaproveitamento, a primária e secundária, envolvendo a reciclagem mecânica; a terciária (reciclagem química) e por último a quaternária, sendo uma reciclagem térmica (HIWATASHI, 1999).

A reciclagem primária é realizada pela própria empresa ou empresa especializada em reprocessar os resíduos, e a reciclagem secundária é a recuperação de resíduos plásticos operados com presença de diferentes resinas que são incompatíveis (FORLIN; FARIA, 2002). Ambos os tipos de reciclagem ocorrem por meio da reciclagem mecânica, que envolve o processamento mecânico de materiais plásticos para moldá-los fisicamente em diferentes formas. Os materiais termoplásticos são vantajosamente adequados para este processo, mantendo na maioria as propriedades físicas, químicas e mecânicas do polímero original (HIWATASHI, 1999, p. 32).

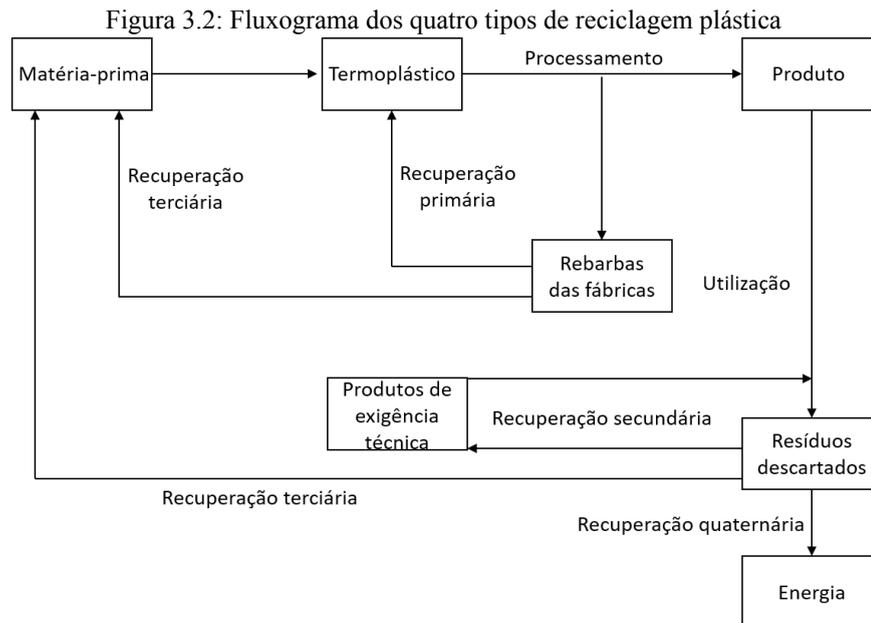
A implementação do método de reciclagem mecânica proporciona coleta seletiva, limpeza de contaminantes, delaminação (material de embalagem compósito), separação e seções de embalagem através da compatibilidade das propriedades do polímero. Na usina de reciclagem, os materiais selecionados são decompostos mecanicamente na forma de pellets, que passam por um processo de extrusão seguido de resfriamento súbito, após moagem e secagem, para produzir material reciclado que é vendido para a indústria de plásticos para uso na fabricação de novos produtos ou outros materiais (FORLIN; FARIA, 2002).

O principal mercado consumidor do plástico reciclado em pellets é a indústria de plásticos, que utiliza o material para produzir baldes, cabides, garrafas de alvejante, eletrodutos e autopeças, entre outros. Consistindo em identificar e separar diferentes resinas e de avanços tecnológicos em equipamentos e técnicas. Como em qualquer outro processo industrial, a reciclagem mecânica deve ser economicamente viável e, entre outros fatores, garantir o fornecimento contínuo de materiais recicláveis, tecnologia adequada para os diferentes produtos e valor de comercialização para compensar a quantidade de energia aplicada no processo (FORLIN; FARIA, 2002).

A reciclagem terciária visa decompor os polímeros em monômeros, oligômeros e outros produtos químicos por meio de processos químicos ou térmicos, incluindo despolimerização de materiais de embalagens plásticas, recuperação e purificação de monômeros virgens, que podem ser polimerizados novamente para a fabricação de novas embalagens plásticas primárias ou outros materiais (FORLIN; FARIA, 2002).

Por fim, há a reciclagem quaternária, que envolve a incineração de plásticos para obtenção de energia térmica, proporcionando a reciclagem com liberação de energia para a combustão ou pirólise de materiais plásticos utilizados como embalagens (FORLIN; FARIA,

2002). A pirólise é uma opção de conversão energética de materiais plásticos e envolve parte do componente térmico, derivado do petróleo bruto, que pode ser utilizado como fonte de energia, por meio de combustão, ou convertido em outros produtos, ou materiais (HIWATASHI, 1999, p. 32). A Figura 3.2 tem a um fluxograma ilustrativo do processo de reciclagens.

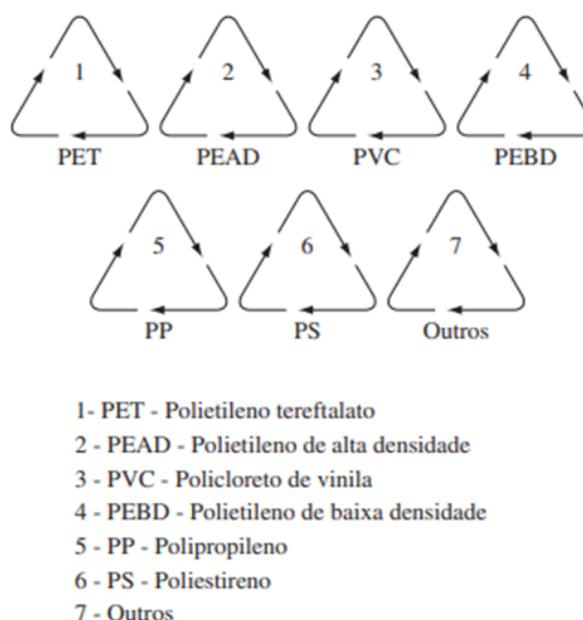


Fonte: Adaptado de HIWATASHI, 1999.

Para a realização da reciclagem é necessária a separação correta dos tipos de resinas e muitos produtos feitos de materiais plásticos têm uma identificação das resinas, geralmente um número de 1 a 7 em um triângulo de três setas, abaixo do qual está uma abreviação destinada a indicar o tipo específico de resina usado no produto, sendo: PET (polietileno tereftalato), PEAD (polietileno de alta densidade), PVC (policloreto de vinila), PEBD (polietileno de baixa densidade), PP (polipropileno), PS (poliestireno) e Outros. Esse código está geralmente localizado no fundo do recipiente ou no verso da embalagem que contém o produto. Os códigos de identificação são projetados para facilitar a reciclagem de recipientes de resíduos plásticos misturados com resíduos sólidos urbanos, pois facilitam a separação e uma posterior reciclagem e reavaliação. O sistema de número de identificação de resina foi introduzido em 1988 pela *Plastics Industry Association* - Sociedade das Indústrias de Plásticos (SPI) a pedido de recicladores para atender suas necessidades e simultaneamente fornecer aos fabricantes um sistema consistente, unificado e padronizado (COLTRO; GASPARINO; QUEIROZ, 2008).

O sistema de código de identificação de resinas SPI é a base da norma ABNT (Associação Brasileira de Normas Técnicas) NBR (Normas Brasileiras) 13230 de 1994 - "Símbolos para Reciclagem e Identificação de Materiais Plásticos", com uma versão mais recente de 2008. Os símbolos de identificação do material plástico são mostrados na Figura 3.3. Os sete materiais identificados pelo símbolo são os plásticos dominantes no mercado (COLTRO; GASPARINO; QUEIROZ, 2008).

Figura 3.3: Símbolos de identificação dos materiais plásticos segundo a norma ABNT NBR 13230.

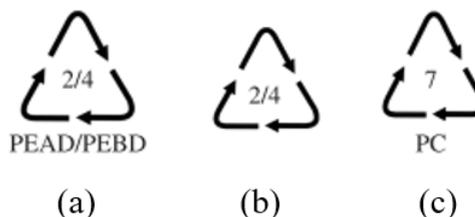


Fonte: COLTRO; GASPARINO; QUEIROZ, 2008.

Para produtos fabricados com polietileno linear de baixa densidade (PELBD), deve-se utilizar o mesmo símbolo do número 4 (PEBD), pois esses materiais podem ser reciclados no mesmo processo sem separação prévia (COLTRO; GASPARINO; QUEIROZ, 2008). Se o produto for feito de uma mistura de resina, co-extrusão e/ou co-injeção, devem ser identificados os dois principais componentes da mistura, como PEAD/PEBD abaixo do triângulo e 2/4 dentro desse triângulo, conforme Figura 3.4 (a) ou Figura 3.4 (b). O símbolo "7 - outros" é geralmente usado para produtos plásticos feitos de policarbonato, ABS (acrilonitrila butadieno estireno), poliamida, acrílico ou resinas diferentes e/ou resinas mistas e tem como sugestão o uso da sigla abaixo do símbolo também, conforme Figura 3.4 (c) (COLTRO; DUARTE, 2013).

O Quadro 3.1 mostra alguns exemplos de aplicações para as resinas e seus respectivos produtos após reciclagem.

Figura 3.4: Embalagens fabricadas por mistura de resinas: (a) mistura PEAD/PEBD com símbolo normalizado (b) mistura PEAD/PEBD com símbolo normalizado em embalagem com dimensões reduzidas (c) símbolo “7 – Outros” indicando qual resina utilizou.



Fonte: COLTRO; DUARTE, 2013.

Quadro 3.1: Exemplos de aplicações das resinas plásticas antes e após reciclagem.

| Símbolo | Resina                         | Aplicação  | Reciclagem  |
|---------|--------------------------------|--|---|
|         | Polietileno tereftalato        | Garrafas para refrigerante, água, óleo comestível, molho para salada, anti-séptico bucal, xampu                        | Fibra para carpete, tecido, vassoura, embalagem de produtos de limpeza, acessórios diversos |
|         | Polietileno de alta densidade  | Garrafas para iogurte, suco, leite, produtos de limpeza, potes para sorvete, frascos para xampu                        | Frascos para produtos de limpeza, óleo para motor, tubulação de esgoto, conduíte            |
|         | Policloreto de vinila          | Filmes estiráveis, berços para biscoitos, frascos para anti-séptico bucal, xampu, produtos de higiene pessoal, blister | Mangueira para jardim, tubulação de esgoto, cones de tráfego, cabos                         |
|         | Polietileno de baixa densidade | Plástico filme, embalagem flexível para leite, iogurte, saquinhos de compras, frascos squeezable                       | Envelopes, filmes, sacos, sacos para lixo, tubulação para irrigação                         |
|         | Polipropileno                  | Potes para margarina, sorvete, tampas, rótulos, copos descartáveis, embalagem para biscoitos, xampu                    | Caixas e cabos para bateria de carro, vassouras, escovas, funil para óleo, caixas, bandejas |
|         | Poliestireno                   | Copos descartáveis, pratos descartáveis, pote para iogurte, bandejas, embalagem para ovos, acolchoamento               | Placas para isolamento térmico, acessórios para escritório, bandejas                        |
|         | Outros                         | Embalagem multicamada para biscoitos e salgadinhos, mamadeiras, CD, DVD, utilidades domésticas                         | Madeira plástica, reciclagem energética   |

Fonte: Adaptado de COLTRO; GASPARINO; QUEIROZ, 2008.

Apesar dos critérios de identificação, apenas 80% das embalagens originais trazem símbolos de identificação da resina (COLTRO; GASPARINO; QUEIROZ, 2008). Porém, muitas dessas embalagens chegam despedaçadas nas cooperativas e a simbologia acaba se perdendo. Além disso, segundo COLTRO *et al.* (2008) em alguns casos, até 40% dos

materiais de embalagem foram identificados incorretamente, quando utilizado o método de identificação manual. Como resultado, o mercado brasileiro ainda possui desinformação sobre o tipo de material de embalagem plástica, incluindo a falta de símbolos de identificação, bem como a falta de informações sobre os símbolos corretos de identificação da resina, ambos prejudicando a cadeia de reciclagem de plástico. Por isso, é necessário o estudo de um método que seja confiável e fácil para ajudar na diminuição da taxa de erro na identificação dos materiais.

### 3.2 Processamento de sinais

O processamento de sinal é um método de análise que utiliza ferramentas matemáticas para realizar transformações ou extrair as informações transportadas pelo sinal (ALBUQUERQUE; ALBUQUERQUE, 2002). Do ponto de vista da engenharia, um sinal é uma função ou sequência usada para passar informações de uma fonte para um receptor (MATUCK, 2005). Sinais existem em uma variedade de situações cotidianas, como fala, música, áudio, fotos e vídeo, e podem ser definidos como uma função de uma ou mais variáveis independentes que transmitem informações sobre propriedades de fenômenos físicos, como tempo, distância, localização, temperatura e pressão (SARMENTO; REGIS; ARAÚJO, 2012).

A maioria dos sinais são gerados naturalmente, mas também podem ser sintetizados ou simulados para transportar informações. Os sinais podem ser criados por uma única fonte, como um sinal escalar, ou por múltiplas fontes, onde é obtido um vetor de sinais, também conhecido como sinal multicanal. A notação unidimensional (1D) é uma função de uma única variável independente, como a fala. Sinais de imagem, como fotografias, são bidimensionais (2D) com duas variáveis independentes. E os sinais multidimensionais (MD) são uma função de múltiplas variáveis, como sinais de vídeo preto e branco são tridimensionais (3D) (ALBUQUERQUE; ALBUQUERQUE, 2002).

Uma das operações mais utilizadas no processamento de sinais é a filtragem. Sua finalidade é permitir que certos componentes de frequência do sinal passem pelo sistema sem qualquer distorção e bloqueie outros componentes de frequência. Os sistemas que implementam essas operações são chamados de filtros. Dependendo da natureza da operação de filtragem, vários tipos de filtros podem ser definidos. Geralmente, a operação de filtragem de sinais analógicos e lineares é descrita pela integração de convolução, como a Equação (1) (ALBUQUERQUE; ALBUQUERQUE, 2002).

$$y(t) = \int_{-\infty}^{\infty} h(t - \xi)x(\xi)d\xi \quad (1)$$

onde  $x(t)$  é o sinal de entrada e  $y(t)$  a saída do filtro caracterizada pela resposta ao impulso  $h(t)$  (ALBUQUERQUE; ALBUQUERQUE, 2002).

Os filtros comumente usados são o filtro passa-baixa, que permite que todos os componentes de frequência passem pela frequência de corte ( $fc$ ) inferior e corte todas as frequências acima. O filtro passa-alta realiza a tarefa oposta, passando frequências acima de  $fc$  e rejeitando frequências mais baixas. O filtro passa-banda permite que frequências em uma faixa definida por  $fc1$  e  $fc2$  passem e rejeitem quaisquer frequências fora dessa faixa. E o filtro rejeita-banda realiza o oposto de um filtro passa-banda, rejeitando apenas as frequências definidas na faixa de  $fc1$  e  $fc2$  e permitindo que quaisquer outros componentes de frequência passem. Para melhor compreensão, a Figura 3.5 ilustra o sinal de entrada original e o comparativo com os respectivos filtros, onde se tem o limite mínimo da frequência de corte de 80 Hz e o máximo de 150 Hz (ALBUQUERQUE; ALBUQUERQUE, 2002).

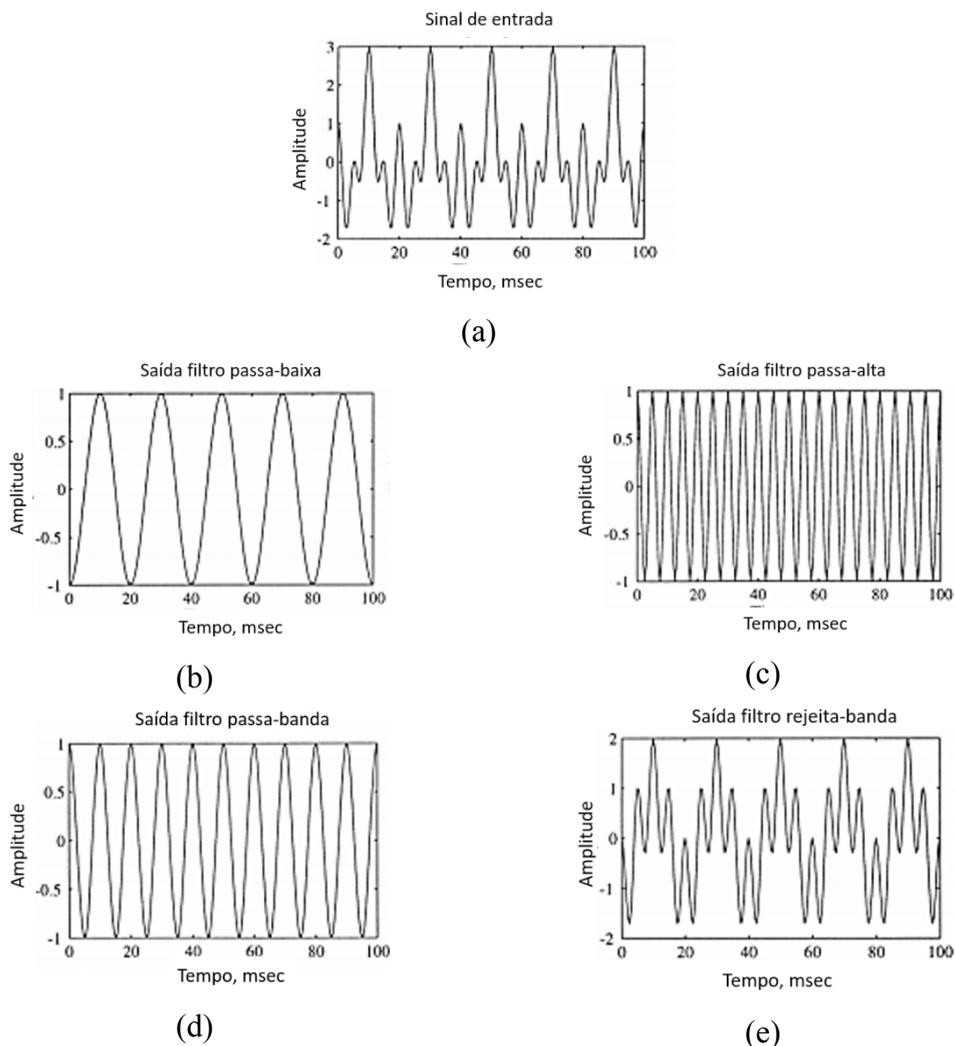
Para que qualquer dispositivo eletrônico funcione, os sinais analógicos devem ser processados em formato digital. A conversão é realizada por meio de uma interface especial, um conversor analógico-digital (ADC). Depois que o sinal é convertido para digital por análise de computador, essa informação ainda pode ser enviada de volta ao mundo real. Para isso, é utilizado um conversor digital-analógico (DAC), que consegue analisar a determinação da sequência e convertê-la em um sinal analógico. Para digitalizar um sinal analógico, são necessários basicamente três passos importantes: amostragem, quantização e codificação (SANTOS, 2015, p. 64).

A amostragem é a coleta de amostras de um sinal contínuo em instantes igualmente espaçados (SANTOS, 2015, p. 65). Conforme o teorema de amostragem, também conhecido como teorema de Nyquist, a frequência de amostragem ( $f_s$ ) deve ser maior ou igual a duas vezes a frequência mais alta do sinal ( $f_{max}$ ) conforme notação da Equação (2) (LIMA, 2022, p. 26).

$$f_s \geq 2 f_{max} \quad (2)$$

Por exemplo, um sinal com uma frequência máxima de 440 Hz é necessária uma taxa de amostragem de 880 amostras por segundo. Se a taxa de amostragem não satisfizer o teorema, ocorrerão efeitos de *aliasing*, sendo impossível restaurar o sinal original (PEREIRA, 2021, p. 19).

Figura 3.5: Comparativo dos filtros no sinal (a) sinal de entrada original (b) saída do sinal com o filtro passa-baixa com  $f_c$  de 80 Hz (c) saída do sinal com o filtro passa-alta com  $f_c$  de 150 Hz (d) saída do sinal com o filtro passa-banda (e) saída do sinal com o filtro rejeita-banda limite mínimo de 80 Hz e o máximo 150 Hz.



Fonte: Adaptado de ALBUQUERQUE; ALBUQUERQUE, 2002.

As amostras obtidas através do processo de amostragem podem quantificar as magnitudes, que pertencem a uma faixa contínua de valores possíveis, chamados níveis de quantização. Essa discretização de magnitude é geralmente definida em termos do número de bits. Assim, por exemplo, se o número de bits em um sinal quantizado for 8, o sinal terá 256

níveis. O número de níveis é sempre obtido fazendo  $2^n$  onde  $n$  é o número de bits utilizados. Cada estado representará uma faixa de valores de magnitude, onde cada valor discreto satisfaz uma faixa menor de valores contínuos para maior precisão. Um erro é adicionado quando o sinal original é arredondado para o nível de quantização e este erro é chamado de erro de quantização. Quanto mais estágios, menor a distância entre eles e menor o erro (SANTOS, 2015, p. 65).

A codificação é o processo final de conversão de analógico para digital. Os níveis podem ser codificados usando qualquer código binário. Ao codificar o sinal, ele produz um sinal digital que completa o processo de conversão (SANTOS, 2015, p. 66).

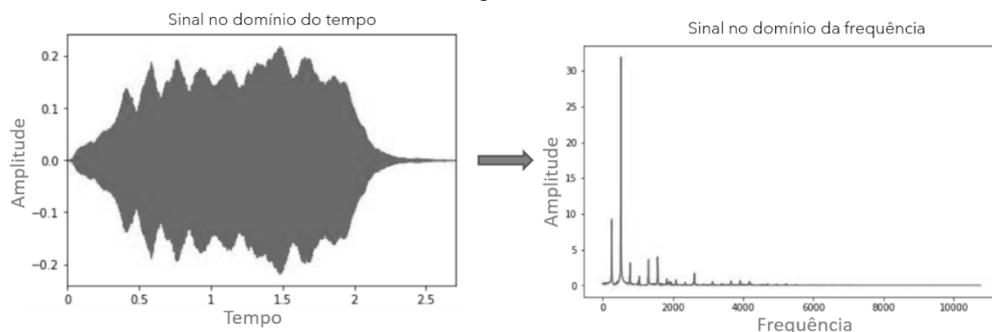
### 3.2.1. Transformada de Fourier

A transformada de Fourier é uma ferramenta útil para extrair informações globais de sinais de onda. A análise de Fourier transforma um sinal do seu domínio original para uma representação no domínio da frequência (COSTA; SOUZA; MIRANDA, 2019), podendo ser representada matematicamente pela Equação (3) e ilustrada pela Figura 3.6.

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i.2.\pi.f.t} dt \quad - \infty < f < \infty \quad (3)$$

onde,  $X(f)$  é a transformada de Fourier direta de  $x(t)$  (SANTOS, 2015, p. 67).

Figura 3.6: Sinal original no domínio do tempo e o sinal aplicado com a transformada de Fourier no domínio da frequência.



Fonte: Adaptado de de um vídeo do canal “Valerio Velardo – DEEP LEARNING The Sound of AI” do YouTube.

A análise de Fourier decompõe um sinal em seus componentes básicos de seno e cosseno. O objetivo da decomposição é facilitar a interpretação do sinal, pois o processamento do sinal original, dependendo da natureza, pode ser mais complexo. Um sinal senoidal na entrada do sistema tem a mesma saída, apenas amplitude e a fase de um sinal podem mudar, mas sua frequência e forma de onda não (SANTOS, 2015, p. 67).

Cada sinal periódico pode ser representado por uma soma de séries de Fourier. Com análise adequada, uma representação de Fourier de um sinal de duração finita pode ser obtida. Essa representação é chamada de transformada discreta de Fourier (DFT) (SCHMIDT, 2013). É amplamente utilizado no estudo da espectroscopia de sinais e determinado digitalmente com a ajuda de um computador digital e aplicado ao processamento digital de sinais, pois espalha a conceituação de tempo contínuo a tempo discreto (SANTOS, 2015, p. 67).

O cálculo da transformada discreta de Fourier considera o sinal amostrado nos domínios do tempo e da frequência. Admitindo-se um sinal discreto no tempo  $x(n)$  com duração finita igual a  $N$  amostras de dados, a DFT é definida como a Equação (4) (SANTOS, 2015, p. 67).

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn} \quad k = 0, 1, \dots, N - 1 \quad (4)$$

na qual  $W_N = e^{-j\frac{2\pi}{N}}$ .

Para pequenos valores de  $N$ , a resolução em si não consome muito tempo ou recursos. No entanto, para valores de  $N$  suficientemente grandes, a computação direta torna-se ineficiente não apenas porque leva muito tempo, mas também porque acumula os recursos necessários. A partir disso pode-se perceber que a ordem de computação direta impõe limitações às aplicações que utilizam DFT, principalmente em tempo real, pois para valores maiores que alguns  $N$ , a computação pode ser lenta e os recursos disponíveis podem ser muito limitados, e assim aparece a FFT, um algoritmo para calcular a DFT rapidamente, através da decomposição da matriz de transformada de Fourier discreta em um produto esparso. É importante ressaltar que ele elimina muitos cálculos repetidos, resultando em cálculos mais rápidos com custos operacionais mais baixos (SCHMIDT, 2013).

### 3.2.2. Coeficientes cepstrais de frequência mel (MFCC)

Os procedimentos de dados de áudio transmitido por uma rede de computadores possuem informações que precisam ser recebidas, interpretadas, identificadas e compreendidas no destino para que essas informações sejam transmitidas corretamente. Existem técnicas de extração das características de fala que permitem processar e apresentá-los a um sistema ou algoritmo em um formato mais tratável. Tais técnicas são utilizadas em sistemas que podem apresentar diversos objetivos relacionados ao reconhecimento de fala, palavras, linguagem, sentimentos e oradores (GOMES, 2018).

Uma das técnicas utilizadas para extrair essas informações é a coeficientes cepstrais de frequência mel – MFCC (do inglês, *Mel-frequency Cepstral Coefficients*) que têm sido utilizadas na análise de fala e tem recentemente recebido atenção em diversas análises de áudio, como música, canto de pássaros e sons em geral (LOUGHRAN et al., 2008).

Para a aplicação desse método, é necessária a escala Mel, cujo nome vem da palavra melodia. É uma unidade de medida de altura proposta por Stevens, Volkman e Newmann em 1937 para refletir com precisão como os humanos ouvem os tons musicais. Esta relação é calculada a partir de uma referência entre o mel e as frequências normais, especificando uma altura sonora percebida, tal que, 1.000 Hz corresponde a 1.000 mel (GOMES, 2018).

O primeiro passo na determinação do MFCC é a pré-ênfase, que processa o sinal passando por um filtro que enfatiza as frequências mais altas, aumentando a energia do sinal. Como o sinal não é estacionário, as amostras convertidas de analógico para digital precisam ser segmentadas e divididas em quadros de pequenos comprimentos em intervalos de 10 a 30 milissegundos (ms), apesar de que entre 10 e 13 ms seja geralmente considerado suficiente para codificar (BENBA; JILBAB; HAMMOUCH; 2016).

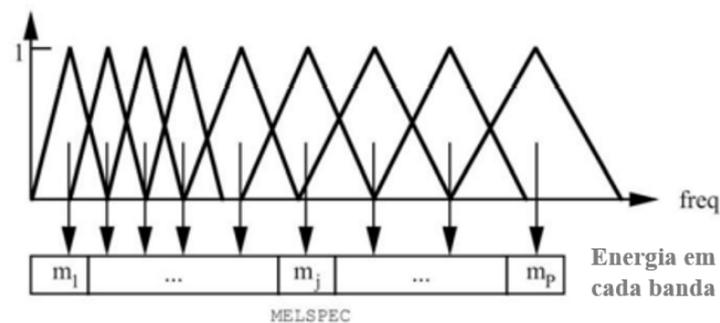
Os sinais são limitados no tempo; por isso, apenas um número limitado de amostras pode ser processado. Deste modo, a próxima etapa no processo, é a janela de quadro. O objetivo desta etapa é reduzir as descontinuidades do sinal e suavizar as extremidades o suficiente para unir com a origem. Isso é feito reduzindo o sinal a zero no início e no final de cada quadro usando uma janela Hamming, aplicando a Equação (5) às amostras de áudio, sendo  $S_n$ ,  $n = 1, \dots, N$  e  $N$  o número de pontos na janela de saída (BENBA; JILBAB; HAMMOUCH; 2016).

$$S'_n = \left\{ 0,54 - 0,46 \cos \cos \left( \frac{2\pi(n-1)}{N-1} \right) \right\} s_n \quad (5)$$

Em seguida, é aplicada a transformada rápida de Fourier, um algoritmo rápido para implementar a transformada discreta de Fourier definida em um conjunto de  $N$  amostras conforme a Equação (4) explicada no tópico 3.2.1, que transforma as  $N$  amostras de cada quadro do domínio do tempo para o domínio de frequência (BENBA; JILBAB; HAMMOUCH; 2016).

É necessária a aplicação dos filtros de mel, pois é uma escala de percepção que ajuda a simular o funcionamento do ouvido humano, correspondendo a uma melhor resolução em baixas frequências e menos em altas. Utiliza-se o banco de filtros triangulares que ajuda a capturar a energia em cada banda crítica e fornecer uma onda aproximada da forma de espectro, além de suavizar a estrutura harmônica (GOMES, 2018). A forma geral do banco de filtros está representada pela Figura 3.7, sendo linearmente espaçada e para calcular a escala Mel para uma determinada frequência ( $f$ ), utiliza-se a aproximação da Equação (6) (BENBA; JILBAB; HAMMOUCH; 2016).

Figura 3.7: Banco de filtros triangulares da escala Mel.



Fonte: Adaptado de BENBA; JILBAB; HAMMOUCH; 2016.

$$Mel(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (6)$$

Finalmente, o MFCC é calculado pela transformada discreta de cosseno (DCT) a partir da magnitude do banco de filtros logarítmicos ( $m_j$ ) usando a Equação (7), onde  $N$  é o número de canais no banco de filtros, e o resultado da transformação é denominado coeficientes cepstrais de frequência mel (BENBA; JILBAB; HAMMOUCH; 2016). O conjunto de coeficientes ( $c_i$ ) é chamado vetor acústico. Dessa forma, cada entrada é transformada em uma série de vetores (GOMES, 2018).

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^N m_j \cos \left( \frac{\pi i}{N} (j - 0,5) \right) \quad (7)$$

### 3.3 Aprendizado de Máquinas

O aprendizado de máquinas, do inglês *Machine Learning* é a parte do campo de pesquisa em IA que realiza análise de dados e visa automatizar análises e modelos estatísticos, conseguindo aprender com base em computadores, principalmente por dados, aprimorando-o por experiência anterior, capaz de reconhecer padrões e ajudar o ser humano a tomar decisões (FRAGOSO, 2022, p. 17; SILVA, 2022, p. 25; BRAGA, 2022, p. 234).

Prontamente, este campo está ligado à estatística, matemática e ciência da computação. Essas áreas de conhecimento possibilitam a criação de algoritmos e soluções que podem lidar com abundância de dados, ajudando a ciência a evoluir de tópicos simples para tópicos mais complexos (FRAGOSO, 2022, p. 17). Na engenharia química, as aplicações incluem controle de processos químicos e bioquímicos, otimização em tempo real, detecção de falhas e reconhecimento de padrões.

Os problemas de aprendizado de máquina são divididos em três subcampos principais: a classificação, regressão e agrupamento. A classificação é baseada na previsão da classe de uma determinada observação. Por exemplo, um classificador pega dados não observados de um paciente e o classifica como doente ou não. A regressão, semelhante à classificação, usa dados de entrada já observados (preditores) para prever uma resposta. A grande diferença é que, neste caso, tenta-se estimar um valor numérico ao invés de uma classificação de observações. Tendo como exemplo, um modelo que usa a idade e os anos de escolaridade de um indivíduo anteriormente não observado para tentar prever seu salário. Para isso, utiliza-se como base as idades, anos de estudo e salários de vários indivíduos observados anteriormente. Por fim, o agrupamento, visa agrupar as observações em grupos denominados “*clusters*”. Essas observações apresentam semelhanças dentro de seus agrupamentos e diferenças em relação aos demais agrupamentos formados. Ao contrário da classificação, a rotulagem de *cluster* não é realizada, portanto, não há grupos falsos ou corretos. Um exemplo é o agrupamento de fotos de animais semelhantes sem saber de antemão qual animal está sendo apresentado (SILVA, 2020, p. 25).

Existem quatro categorias de aprendizado de máquina: aprendizado supervisionado, não supervisionado, semisupervisionado e por reforço. No aprendizado supervisionado, os

dados fornecidos ao algoritmo para a etapa de treinamento são rotulados e, com base nisso, o algoritmo consegue previsões sobre dados não vistos. Além disso, é frequentemente aplicado a tarefas de classificação. Por outro lado, no aprendizado não supervisionado, os dados de treinamento fornecidos ao algoritmo não são rotulados, ou seja, o objetivo é tentar aprender individualmente, e ainda é comumente usado para agrupamento e redução de dimensionalidade. Algoritmos semi-supervisionados combinam técnicas supervisionadas e não supervisionadas, capazes de aprender usando uma pequena porção de dados rotulados e uma grande porção de dados não rotulados. Por fim, nos algoritmos de reforço, a máquina tenta aprender a melhor ação para decidir com base em uma situação específica onde há uma penalidade ou recompensa ao final de cada decisão (SILVA, 2022, p. 26).

Em relação aos algoritmos utilizados no aprendizado supervisionado, especialmente nas tarefas de classificação, existem algoritmos como regressão logística, redes neurais, árvores de decisão, *K-Nearest Neighbors* (KNN), *Support Vector Machines* (SVM), entre outros. Os algoritmos utilizam diferentes métodos de aprendizado, portanto os resultados variam de um algoritmo para outro (SILVA, 2022, p. 29).

Neste trabalho o foco é na classificação e no aprendizado de máquina supervisionado utilizando algoritmos de rede neurais artificiais para tratar do problema estudado.

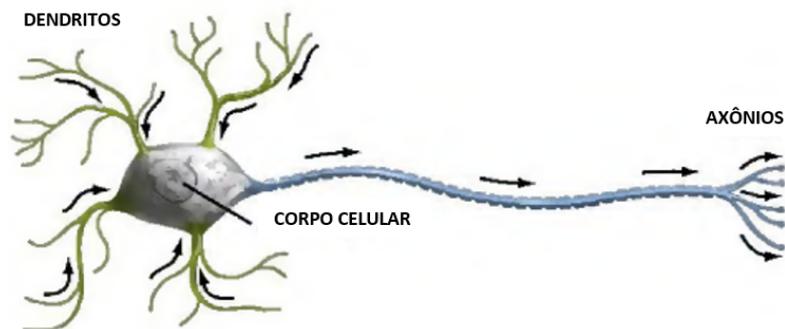
### 3.3.1 Redes Neurais Artificiais (RNA)

O comportamento das redes neurais artificiais se reflete no comportamento do cérebro humano, que contém bilhões de estruturas de processamento simples, chamada neurônios, cada uma das quais está conectada a centenas de outras estruturas por sinapses (OLIVEIRA, 2020, p. 7). Os neurônios são compostos basicamente do corpo celular, dendritos e axônios. Os dendritos têm a função de receber informações de outros neurônios e transmiti-los ao corpo celular, essas informações são processadas e geram novos impulsos, transmitidos para outros neurônios, que passam pelo axônio atingindo o dendrito do neurônio seguinte, conforme a ilustração da Figura 3.8. Esses neurônios se unem formando as redes neurais através das sinapses, que atuam como válvulas que controlam a transmissão dos impulsos e o fluxo de informações entre os neurônios (BENTES, 2018, p. 18).

As redes neurais artificiais são como essas estruturas, funcionando de maneira semelhante e visam modelar uma função que responda a um determinado problema. Para simular as respostas que ocorrem no cérebro humano, os neurônios artificiais recebem múltiplas entradas, processadas para produzir saídas, alimentadas por funções de ativação. O

resultado dessa função é chamado nível de ativação, sendo a saída do neurônio. Os pesos são semelhantes às sinapses dos neurônios biológicos e são responsáveis por ponderar o valor da entrada, e adicionar o desvio para configurar a operação linear (OLIVEIRA, 2020, p. 7).

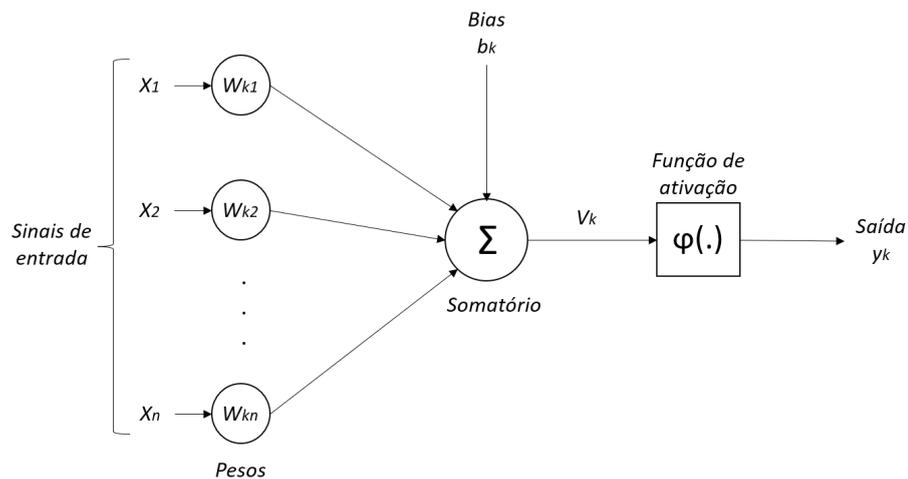
Figura 3.8: Estrutura de uma rede neural biológica.



Fonte: Adaptado de BENTES, 2018.

A Figura 3.9 é uma representação de um modelo de neurônio artificial apresentado em 1940 por McCulloch e Pitts.

Figura 3.9: Representação do modelo neural artificial.



Fonte: Adaptado de MCCULLOCH; PITTS, 1943.

As entradas dos neurônios contêm pesos ligados ( $w_{k1}, w_{k2}, \dots, w_{kn}$ ) com  $n$  entradas e a saída desses neurônios é um somatório ( $u_k$ ) das entradas ( $X_i$ ) ponderada com os pesos ( $w$ ) que pode ser descrita pela Equação (8) (OLIVEIRA, 2020, p. 8).

$$u_k = \sum_{i=1}^n w_{ki} X_i \quad (8)$$

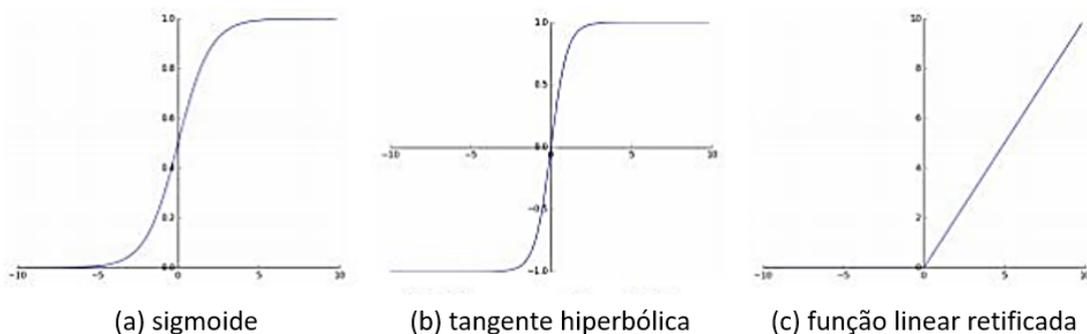
Os bias ( $b_k$ ) são adicionados ao somatório dos neurônios de entrada, conforme Equação (9) e submetido a função de ativação ( $\varphi$ ) como a limite limiar que é uma função degrau que descreve uma operação parecida com a Equação (10) e o  $v_k$  é a saída da função de ativação do neurônio (OLIVEIRA, 2020, p. 8).

$$v_k = u_k + b_k \quad (9)$$

$$\varphi(v_k) = \begin{cases} 1, & \text{para } v_k \geq 0 \\ 0, & \text{para } v_k < 0 \end{cases} \quad (10)$$

Existem outras funções de ativação comuns, que diferem do limite limiar, como a sigmoide, tangente hiperbólica e a função linear retificada (ReLU), que estão demonstradas na Figura 3.10, respectivamente.

Figura 3.10: Exemplo de funções de ativação.



Fonte: Adaptado de ARAÚJO, 2019.

A função sigmoide tem um valor entre 0 e 1 sendo usada no modelo para prever probabilidades como saída. Sua curva tem formato de “S”, conforme mostra a Figura 3.10 (a), que pode ser representada matematicamente pela Equação (11) (ARAÚJO, 2019, p. 18).

$$\varphi(x) = \frac{1}{1+e^{-x}} \quad (11)$$

A função de ativação da tangente hiperbólica (Tanh) é semelhante à função sigmoide, mas é simétrica em relação à origem, portanto varia entre os valores -1 a 1 e pode ser expressa matematicamente pela Equação (12) e sua curva é representada pela Figura 3.10 (b) (ARAÚJO, 2019, p. 18).

$$\tanh \tanh(x) = \frac{(e^x - e^{-x})}{e^x + e^{-x}} \quad (12)$$

A função ReLU foi introduzida pela primeira vez em redes dinâmicas em 2000 por Hahnloser e outros pesquisadores (HAHNLOSER *et al.*, 2000). E em 2011, o uso dessa função como uma não linearidade mostrou treinar redes neurais supervisionadas profundas sem a necessidade de um pré-treinamento não supervisionado. Em comparação com as outras duas funções mencionadas anteriormente, essa permite um treinamento mais rápido e eficiente de arquiteturas mais profundas com grandes conjuntos de dados complexos e não utiliza expoentes, apenas realiza comparações, adições e multiplicações (ARAÚJO, 2019 p. 19).

Essa função, é definida apenas para a parte positiva de seu parâmetro, ou seja, sempre que tiver um valor negativo em sua entrada, o valor será convertido para zero e o neurônio não será ativado, apenas uma parte, tornando a rede dispersa. Sua curva está representada pela Figura 3.10 (c) e pode ser escrita matematicamente pela Equação (13) (ARAÚJO, 2019, p. 19).

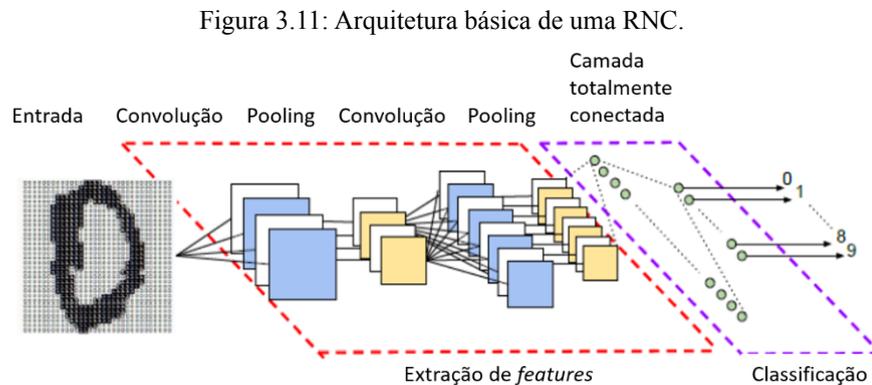
$$f(x) = \max(0, x) \quad (13)$$

O aprendizado de uma RNA ocorre por um processo iterativo de ajuste de pesos sinápticos, conhecido como treinamento. A aprendizagem ocorre apenas quando a rede neural atinge uma solução generalizada para um determinado problema. Em resumo, treinar uma rede é ajustar sua matriz de peso sináptico de tal forma que o vetor de saída corresponda ao valor esperado de cada vetor entrada (FLECK *et al.*, 2016).

Durante a fase de aprendizado, a rede neural artificial extrai informações relevantes dos padrões de informação que lhe são apresentados, resultando em sua própria representação do problema. Nesta etapa são ajustados os parâmetros da rede, caracterizados pelos pesos das conexões entre as unidades de processamento. Ao final da fase de treinamento, a rede terá adquirido conhecimento sobre o problema e armazenado as informações em seus parâmetros (FLECK *et al.*, 2016).

### 3.3.1.1 Rede neural convolucional

A rede neural convolucional, foi proposta pelo pesquisador francês Yann LeCun em 1998 e mostrou ser muito eficiente para resolução de problemas de classificação e viável em comparação a outros modelos para essa solução (LeCun *et al.*, 1998). As RNC requerem uma abundância de dados rotulados para extrair suas características, conhecida como *features*. Para isso, uma arquitetura básica da rede contém camadas convolucionais, *pooling* e camada totalmente conectada, conforme ilustra a Figura 3.11 (RODRIGUES, 2018, p. 23).



Fonte: Adaptado de PENHA, 2018.

Nas camadas de convolução são realizados os cálculos entre a entrada e o kernel de convolução que gera um mapa de características das entradas e passa a função de ativação para obter a saída da camada convolucional, que seguem a Equação (14) (LU *et al.*, 2020).

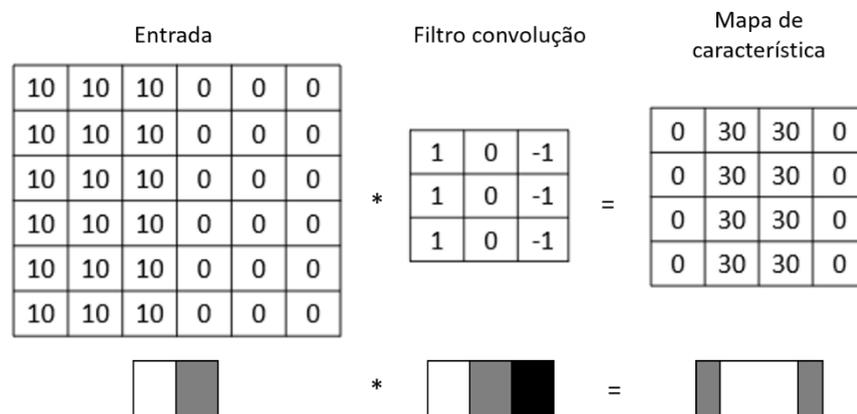
$$y_j^l = f(b_j^l + \sum_{i \in M_j} conv(w_{ij}^{l-1}, x_i^{l-1})) \quad (14)$$

onde,  $y_j^l$  é a saída do neurônio na camada  $l$ ,  $f$  é uma função não linear,  $b_j^l$  é a tendência escalar do neurônio na camada  $l$ ,  $M_j$  representa uma seleção de mapas de entrada,  $w_{ij}^{l-1}$  são os pesos do *kernel* do neurônio na camada  $l-1$  e  $x_i^{l-1}$  é a saída do neurônio  $l-1$  (LU *et al*, 2020).

As camadas convolucionais são responsáveis por extrair recursos da entrada. Esse processo de extração é realizado por um filtro de convolução de tamanhos reduzidos, onde o filtro percorre a largura, altura e profundidade, chamadas dimensões dos dados de entrada, realizando operações de convolução nos dados (RODRIGUES, 2018, p. 24).

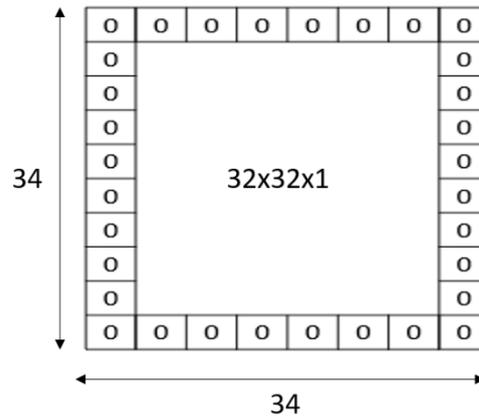
Cada filtro tem um conjunto de pesos, portanto, filtros diferentes aprendem diferentes categorias de recursos de entrada. A Figura 3.12 é um exemplo de detecção de borda vertical com uma entrada de tamanho 6x6, com valores em tons de cinza, um tamanho de passo (chamado *stride*) de 1, que indica como o filtro avança pelas camadas e a dimensão do filtro é igual a 3 (PENHA, 2018, p. 18).

Figura 3.12: Extração das características com filtro detecção de borda.



Fonte: Adaptado de PENHA, 2018.

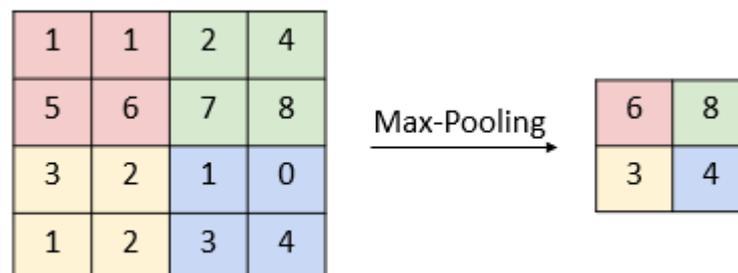
Além do *stride*, outra técnica muito utilizada em camadas convolucionais é o *padding*, que adiciona zeros ao redor das bordas da imagem para que as características contidas próximas às bordas possam ser extraídas. A Figura 3.13 é um exemplo de aplicação de preenchimento de tamanho 1, uma borda extra com valor 0 (PENHA, 2018, p. 19).

Figura 3.13: Exemplo utilização de *padding*.

Fonte: Adaptado de PENHA, 2018.

À medida que a entrada na rede progride, a estrutura de aprendizado de filtros se torna cada vez mais complexa, ou seja, quanto mais filtros convolucionais, mais recursos extraímos da entrada, mas isso tem um custo de memória e processamento, requerendo um equilíbrio ao definir arquitetura (RODRIGUES, 2018, p. 24). Geralmente após a camada convolucional, usa-se uma *pooling*, que consiste basicamente em uma camada projetada para reduzir o tamanho dos dados de entrada, para que as camadas convolucionais subsequentes recebam outra forma de representação dos dados, permitindo que a rede aprenda diferentes modos, evitando assim o *overfitting* (RODRIGUES, 2018, p. 24).

Há várias técnicas para executar a camada *pooling*, como *max pooling*, que reduz subpartes dos dados originais pelo maior valor encontrado nessas sub regiões, reduzindo com isso o tamanho da imagem por um fator de filtro  $m \times n$ . A *average pooling*, que utiliza o valor da média ou *L2- pooling*, sendo a normal do conjunto (RODRIGUES, 2018, p. 24). A mais utilizada é a *max pooling*, que pode ser vista em um exemplo na Figura 3.14 em uma representação de mapa de característica, com o uso de filtros do tamanho  $2 \times 2$  e *stride* igual a 2 (PENHA, 2018, p. 21).

Figura 3.14: Exemplo camada *max-pooling*.

Fonte: Adaptado de PENHA, 2018.

As camadas totalmente conectadas estão geralmente localizadas no final da rede. Nessas camadas, as características extraídas nas camadas convolucionais anteriores são utilizadas para a classificação da saída da rede, geralmente utilizando uma função *softmax*, que normaliza a saída da camada totalmente conectada. A saída desta camada consiste em números positivos que somam 1, que podem então ser usados como probabilidades na camada de classificação. A função de ativação da unidade de saída é obtida pela Equação (15) (PENHA, 2018, p. 22).

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \quad (15)$$

Na camada de classificação, a função de treinamento toma o valor da *softmax* e usa a função de entropia cruzada para atribuir cada entrada a uma das  $n$  classes mutuamente exclusivas, o que pode ser determinado pela Equação (16) (PENHA, 2018, p. 23). As funções de perda, são para calcular a quantidade de dados previstos na saída que difere dos dados reais. Para problemas de classificação, é utilizado a função de Perda de Entropia Cruzada, quando tiver duas ou mais classes fornecidos como números inteiros (Keras).

$$E(w) = \frac{1}{n} \sum_{i=1}^n [t_i \cdot \log \log(y_i) + (1 - t_i) \cdot \log(1 - y_i)] \quad (16)$$

onde,  $w$  é a matriz de pesos,  $y_i$  o resultado deduzido,  $t_i$  o resultado correto e  $n$  o número total de itens de treinamento (PENHA, 2018, p. 23).

Como as redes neurais convolucionais são frequentemente usadas para aprendizado supervisionado, elas precisam ser treinadas com excesso de dados para poder generalizar melhor para outras entradas às quais nunca foram expostas. Esse treinamento é realizado passando os dados de treinamento pela rede várias vezes, onde o número dessas iterações é um dos parâmetros a serem determinados (PENHA, 2018, p. 23).

O aprendizado da rede, consiste em duas etapas, a *Forwardpass* constitui em passar a entrada para todos os neurônios necessários para chegar à saída, que realizarão suas ações típicas, considerando os pesos aprendidos (ou inicializados aleatoriamente). Nesta etapa, a

entrada é propagada formalmente por uma função específica de cada camada da rede na forma  $f_n (...f_2 (f_1(f_0(x)) ...)$  onde  $f$  é a função de ativação da camada e  $n$  se refere ao número de camadas de rede (PENHA, 2018, p. 24).

A etapa de retropropagação, realizada por otimização, executada através de um método denominado gradiente descendente estocástico (do inglês, *Stochastic Gradient Descent - SGD*), que visa minimizar a função de perda alterando os pesos na direção de maiores inclinações do gradiente da função. Essa otimização é realizada em lotes, subconjuntos da entrada, divididos assim devido à limitada capacidade de memória das GPUs, os dispositivos mais utilizados para treinamento de redes neurais convolucionais atualmente (PENHA, 2018, p. 24).

Para a avaliação do modelo para problemas de classificação, normalmente utiliza-se métrica Accuracy, que calcula a frequência em que as previsões são iguais aos rótulos. O  $\hat{y}_i$  é o valor previsto da amostra e  $y_i$  o valor real equivalente, onde a fração de previsões corretas em amostras ( $n_{samples}$ ) é dada pela Equação (17) (Scikit-Learn).

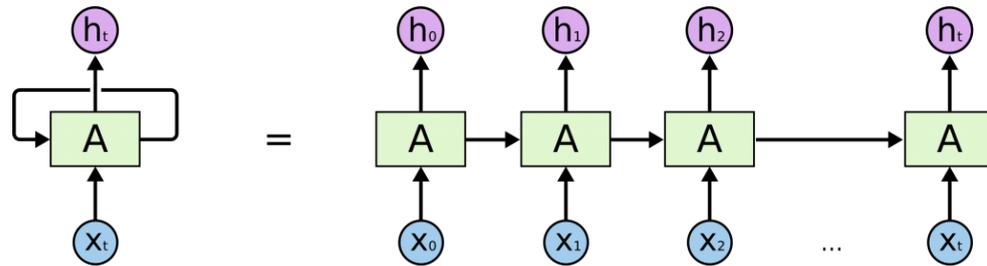
$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(y_i = \hat{y}_i) \quad (17)$$

### 3.3.1.2 Redes neurais recorrentes

Uma rede neural recorrente (do inglês, *Recurrent Neural Network*) é uma rede específica para problemas de dados sequenciais. Eles podem ser pensados como múltiplas réplicas da mesma rede, cada uma entregando mensagens ao próximo. As RNNs possuem um mecanismo recorrente, chamado *looping*, que atua como um caminho que permite que as informações fluam de uma rede para outra (JULIANI, 2019, p. 17).

A função deste mecanismo *looping* é mostrada na Figura 3.15, onde a entrada é representada por  $X_t$ , a rede neural por  $A$  e o valor de saída é representado por  $h_t$ . A ideia de conectar informações anteriores visa entender o quadro atual com base nas informações anteriores. Quando isso acontece, há uma relação entre as duas entradas sequenciais. No entanto, surgem problemas quando a diferença de intervalo entre uma entrada e outra é muito grande, ficando difícil acessar informações em passados distantes. (JULIANI, 2019, p. 17).

Figura 3.15: Funcionamento de uma RNN.



Fonte: JULIANI, 2019.

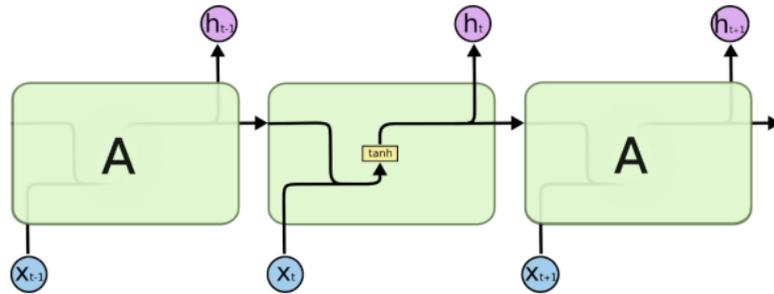
Redes neurais recorrentes comuns usam *feedback* para armazenar representações de eventos recentes da entrada, chamada memória de curto prazo (do inglês - *short-term memory*). Esse recurso é útil para muitas aplicações que não exigem influência do estado anterior. Existem algoritmos e arquiteturas de rede mais complexos que tentam aprender melhor esses termos em redes recorrentes comuns, mas o processo de treinamento é muito complexo e lento e pode não funcionar bem (OLIVEIRA, 2020, p. 10).

### 3.3.1.2.1 Redes neurais recorrentes LSTM

Para resolver o problema de memória de curto prazo eficientemente, as redes neurais recorrentes *Long Short Term Memory* (LSTM), originalmente propostas por Hochreiter e Schmidhuber em 1997, apresentam bom desempenho em uma ampla variedade de problemas, são abrangentemente utilizadas hoje e capazes de aprender dependências (HOCHREITER; SCHIMIDHUBER, 1997). Esta categoria de rede recorrente consegue reter informações de muitas entradas anteriores (memória de longo prazo), mantendo os estados recentes mais relevantes (PEREIRA, 2017, p. 18).

Todas as redes neurais têm a forma de cadeias de módulos repetitivos. Em uma RNA padrão, esse módulo possui uma estrutura muito simples, como uma camada com função Tanh, conforme mostrado na Figura 3.16 (PEREIRA, 2017, p. 18).

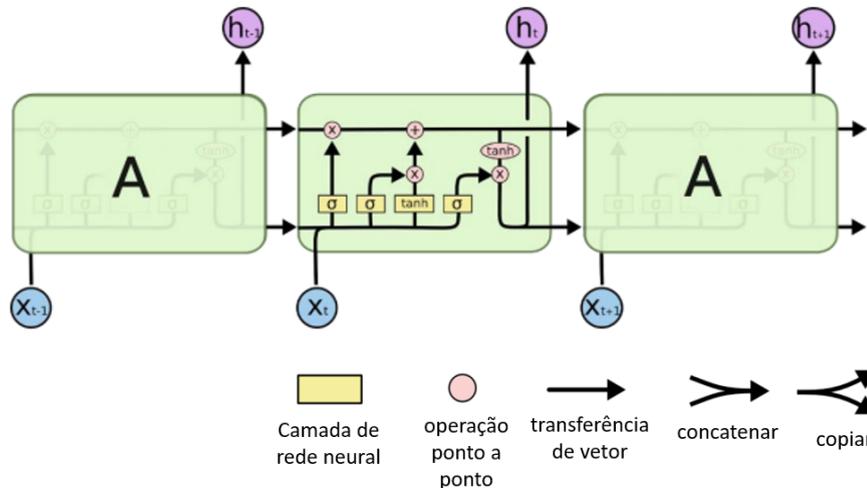
Figura 3.16: Módulo de uma RNA comum com apenas uma camada.



Fonte: PEREIRA, 2017

As redes LSTM também possuem essa estrutura encadeada, mas os módulos de repetição têm uma estrutura diferente. Em vez de apenas uma camada de redes neurais, existem quatro, e elas interagem de maneiras muito específicas. A Figura 3.17 visualiza esse conceito (PEREIRA, 2017, p. 19).

Figura 3.17: Módulo da rede LSTM com quatro camadas.



Fonte: PEREIRA, 2017

A ideia principal da rede LSTM é criar uma representação do estado da célula, que corresponde à linha horizontal na parte superior do gráfico. Esse estado de célula percorre toda a cadeia de células com apenas um pequeno número de interações lineares, de modo que as informações podem fluir sem muita mudança (PEREIRA, 2017, p. 19).

Em um dado instante de tempo  $t$ , a unidade LSTM toma como entrada o instante de informação atualmente inserido na rede, identificado como  $x_t$ , o estado oculto (*hidden state*)  $h_{t-1}$  e o estado unitário (*cell state*)  $C_{t-1}$ , ambos estados do tempo  $t - 1$ . As saídas da célula são o

estado atual imediato da célula  $C_t$ , o estado oculto e a saída de informação  $h_t$ . Se a unidade pertence à última camada da rede,  $h_t$  é entendido como a saída final, e se a camada estiver na rede,  $h_t$  será  $h_{t-1}$  para a próxima camada da rede (OLIVEIRA, 2020, p. 12).

Além da entrada e saída, uma célula LSTM internamente consiste em uma combinação de funções de ativação, adições e produtos. Essas operações internas das células são chamadas de *gates*. São uma maneira de fazer com que as informações fluam através de uma rede neural. Eles consistem em camadas sigmóides ( $\sigma$ ) de redes neurais e multiplicações pontuais. A camada sigmoide produz números de 0 a 1, ou seja, descreve quanto de cada componente deve passar pela porta. Quanto maior o valor, mais informações são passadas pela camada. Esta rede possui *gates* para proteger e controlar o estado da célula (OLIVEIRA, 2020, p. 12).

O *forget gate* é o primeiro componente de um neurônio LSTM e é responsável pela capacidade da rede de esquecer informações retidas por estados passados. A Figura 3.18 (a) destaca o portão de esquecimento na célula LSTM. Pode-se ver que as duas entradas  $h_{t-1}$  e  $x_t$  da unidade estão conectadas neste ponto e passam pela função de ativação  $\sigma$ . A Equação (18) mostra o processamento realizado, onde  $x_t$  é a entrada da célula no tempo atual ( $t$ ),  $h_{t-1}$  é o estado oculto no tempo passado,  $W_f$  e  $b_f$  são os pesos e bias da entrada do *gate* de esquecimento,  $\sigma$  denota a função de ativação sigmoide e saída  $f_t$  (OLIVEIRA, 2020, p. 12).

$$f_t = \sigma (W_f [h_{t-1}, x_t] + b_f) \quad (18)$$

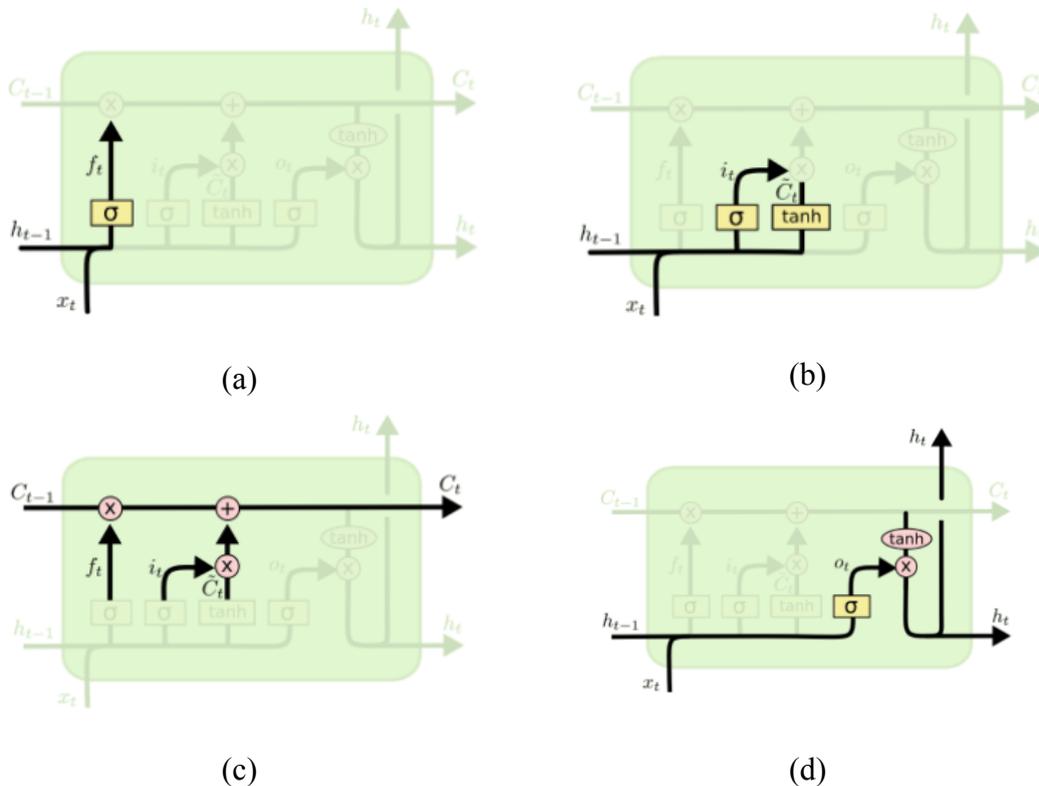
O *input gate* ( $i_t$ ) e o *cell gate* ( $\tilde{C}$ ) são responsáveis por decidir quanta informação de entrada nova é alimentada na rede, ressaltadas na Figura 3.18 (b) em uma célula LSTM. A combinação das entradas  $h_{t-1}$  e  $x_t$  é passada por duas funções de ativação diferentes  $\sigma$  e  $\tanh$  (tangente hiperbólica). O sistema de equações (19) mostra o processamento feito pelas portas de entrada e célula (OLIVEIRA, 2020, p. 13).

$$i_t = \sigma (W_i [h_{t-1}, x_t] + b_i) \quad \tilde{C} = \tanh \tanh (W_c [h_{t-1}, x_t] + b_c) \quad (19)$$

Uma atualização de célula (*cell update*) corresponde a um conjunto de operações realizadas para obter o estado da célula, propagado da célula correspondente para o próximo momento no tempo. O *cell update* é responsável por atualizar o valor do *cell state*, e sua

função é carregar as informações geradas pela operação interna da célula sem participar da saída (OLIVEIRA, 2020, p. 14).

Figura 3.18: Representação da célula LSTM (a) *forget gate* (b) *input gate* (c) *cell gate* e (d) *output gate*.



Fonte: [JUNIOR, 2019.](#)

A Figura 3.18 (c) destaca o posicionamento das ações executadas em uma atualização de célula. Observa-se que há uma multiplicação entre o *cell state* no momento  $C_{t-1}$  com a saída do *forget gate*  $f_t$ . Somando esse valor ao resultado obtido do produto entre o *input gate*  $i_t$  e o *cell gate*  $\tilde{C}_t$ . A Equação (20) mostra o processamento da conclusão da atualização da célula (OLIVEIRA, 2020, p. 15).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (20)$$

O símbolo \* denota o produto Hadamard, também conhecido como multiplicação por elementos. O resultado dessas operações produz o estado da célula  $C_t$ , que servirá como uma das entradas para a próxima célula LSTM (OLIVEIRA, 2020, p. 15).

Finalmente, o *output gate* é responsável por processar parte das informações que serão emitidas como  $h_t$  da rede. Como mencionado anteriormente, o *hidden state* é um estado que carrega informações de momentos anteriores que serão propagadas na rede para momentos posteriores. A Figura 3.18 (d) mostra a localização da porta de saída e o estado oculto na célula LSTM (OLIVEIRA, 2020, p. 15).

O *output gate* processa informações da entrada  $x_t$  e do *hidden state* no último momento  $h_{t-1}$ . Desta forma, o estado oculto é atualmente gerado pela combinação do *output gate* e o *cell state*, e o estado da célula também é transformado pela função de ativação. O sistema de equações (21) mostram o processamento feito pelo *output gate* ( $o_t$ ) e o *hidden gate* ( $h_t$ ) é obtido (OLIVEIRA, 2020, p. 15).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad h_t = o_t * \tanh(C_t) \quad (21)$$

A composição interna dessas redes torna o estado interno da célula responsável por propagar informações sobre estados passados através da rede, não necessitando, portanto, de *feedback* explícito, sendo estruturalmente diferente das RNAs comuns. Essa estrutura de unidades individuais as aproxima das redes de multicamadas, pois a informação flui diretamente da entrada da rede para a saída (OLIVEIRA, 2020, p. 16).

## Capítulo 4 - METODOLOGIA

Este capítulo descreve a metodologia utilizada para a gravação dos áudios das amostras e classificação dos sete tipos de resinas plásticas recicláveis por meio de técnicas de inteligência artificial. O trabalho está dividido em três principais etapas: amostragem, preparação dos sinais de áudio e classificação.

Todos os algoritmos realizados na pesquisa foram feitos no *Google Colaboratory (Colab)*, um ambiente interativo que permite escrever e executar códigos em Python. É bastante utilizado pela comunidade de aprendizado de máquina, podendo importar um conjunto de dados, como imagens e áudios, treinar o classificador e avaliar o modelo. Os *notebooks Colab* executam os códigos nos servidores em nuvem da Google, aproveitando o *hardware*, como GPU's (Unidade de Processamento Gráfico) e TPU's (*Tensor Processing Unit*), o qual é a base para o processamento dos recursos de aprendizagem profunda, independente do poder de processamento da máquina utilizada, tudo é realizado pelo navegador (COLAB).

### 4.1 Amostragem

No presente trabalho, foram coletadas amostras de embalagens plásticas oriundas de resíduos urbanos de vários tamanhos e das sete categorias de resina, denominadas como PET, PEAD, PVC, PEBD, PP, PS e Outros. Alguns exemplos de materiais coletados, são garrafas, potes, frascos, sacolas, plástico filme, copos e embalagens de alimentos.

Foram separadas 100 amostras por categoria, somando 700 amostras, higienizadas e cortadas em aproximadamente 10 centímetro de comprimento e a largura foi utilizada as dos próprios materiais. Foram separadas em grupos de cerca de 100 gramas e colocadas em bags de pano de forma compactada, conforme Figura 4.1 para que não houvesse interferência no som. Havia um bag para cada categoria de resina, para poderem ser amassadas separadamente.

Outras 700 amostras foram utilizadas peças inteiras, uma forma facilitada de ser aplicada nas cooperativas, sem a utilização do bag de pano, como mostra Figura 4.2. O total de amostras foi de 1.400. Para um melhor entendimento, ao longo deste trabalho, serão adotadas como caso 1 para as amostras cortadas e colocadas em conjunto no bag de pano e caso 2 para as amostras inteiras. As medidas e peso das amostras escolhidos foram para que

coubessem na morsa e ter certeza que todas as amostras que estavam dentro do bag fossem amassadas.

Figura 4.1: Amostras cortadas e amassadas em conjunto para caso 1 (a) Amostras de PET cortadas em aproximadamente 10 centímetro de comprimento e largura dos materiais (b) Amostras cortadas dentro do saco de pano.



(a)

(b)

Fonte: Autora, 2022.

Figura 4.2: Amostras inteiras das respectivas categorias: PET, PEAD, PVC, PEBD, PP, PS e Outros para caso 2.



Fonte: Autora, 2022.

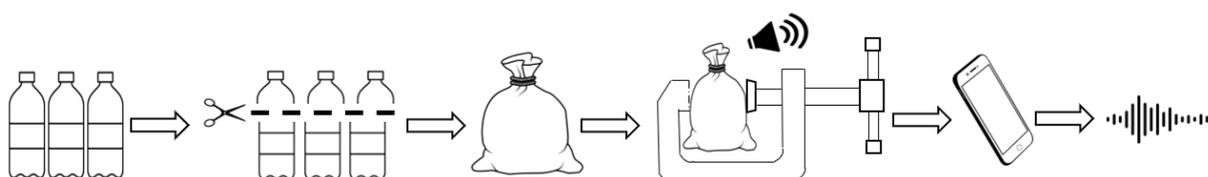
A validação do modelo, foi realizado a captura do áudio de 20 amostras de cada categoria de resina, totalizando 140 amostras. O processo de amassamento utilizando foi o mesmo para o caso 1. Colocando as amostras dentro do bag de pano e com o auxílio da morsa, mas amassando de forma mais lenta, para analisar se as características dos modelos neurais se mantinham. Essa etapa, foi denominado caso 3.

A gravação do som do plástico sendo amassado, utilizou-se o gravador de um smartphone do modelo Iphone 12, que contém microfone integrado. A gravação foi realizada para o caso 1, caso 2 e caso 3. Os plásticos foram amassados em um local que tinha menos ruídos externos possíveis para que não houvesse interferência na gravação. Seguiu-se um padrão de duração de 3 segundos, pois é um tempo suficiente para identificação e o processo de esmagamento é realizado rapidamente quando feito nas cooperativas de reciclagem. Um

esquema representativo do processo de captura do áudio realizado para o caso 1 e caso 3 pode ser analisado pela Figura 4.3 e para caso 2 pela Figura 4.4.

O esquema representativo do processo de captura de áudio para o caso 1 e caso 3, tem as amostras inteiras e higienizadas, em seguida elas são cortadas e colocadas no bag de pano. O bag de pano foi colocado no meio da morsa e com a válvula prensaram-se as amostras manualmente. Enquanto ocorria o esmagamento, com o auxílio do smartphone foi capturado o som gerado, onde se obteve o sinal do áudio. Esse processo foi repetido com todas as amostras das sete categorias de resina.

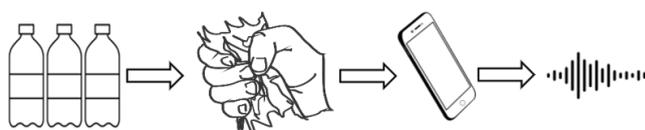
Figura 4.3: Esquema representativo do processo de captura do áudio do caso 1 e caso 3.



Fonte: Autora, 2022.

Para a captura do áudio para o caso 2, foi mais simples, pois foram utilizadas as amostras inteiras e higienizadas, amassadas com as mãos mesmo, por ser apenas uma embalagem de cada vez. O som foi capturado pelo auxílio do smartphone, gerando o sinal de áudio. Esse processo foi realizado para todas as amostras das diferentes categorias.

Figura 4.4: Esquema representativo do processo de captura de áudio para o caso 2.



Fonte: Autora, 2022.

#### 4.1.1 Sistema de aquisição de dados

A gravação dos áudios, teve uma taxa de amostragem de 44.100 Hz, um valor padrão utilizado na maioria dos codificadores (CAVALEIRO, 2015, p. 16). Para montagem de um banco de dados equivalente para classificação, realizaram-se essas gravações em 100 amostras de plásticos de cada categoria nos ambos casos. Ficando assim com um banco de dados de 1.400 arquivos de áudio. Os sinais foram passados para um computador com um

processador Core i3 de 8<sup>th</sup> geração com uma memória RAM (*Random Access Memory*) interna de 4GB e um sistema operacional de 64 bits.

Os arquivos de áudio foram convertidos do formato mp4 para o formato WAV (*Waveform Audio File Format*), onde não há perda da qualidade, reproduzindo tudo que foi gravado (AMARAL, 2019, p. 37), formato aceitável nas bibliotecas Python. Dividiram-se esses arquivos em sete pastas diferentes, cada uma representando um tipo de resina plástica. O banco de dados para o caso 1 e caso 2 podem ser visualizados no repositório da Unicamp pelo link: <https://doi.org/10.25824/redu/UGMPJR>.

## 4.2. Preparação dos sinais de áudio

É necessária uma análise dos sinais para se conhecer o tipo de dados, como a forma de onda original, aplicação da transformada de Fourier e uma técnica para extração das características de áudio para a identificação. A técnica empregada foi a MFCC, que mostrou ser eficiente e solucionável para a preparação dos dados de áudio das resinas plásticas.

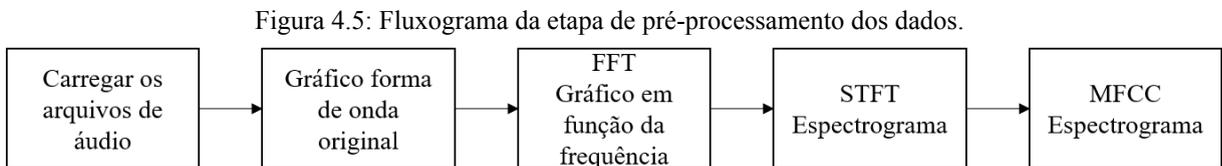
Nesta etapa foi desenvolvida uma primeira etapa do código, chamado de pré-processamento, mostrando a forma de onda original e em seguida aplicou-se a transformada de Fourier para análise da frequência, conforme a seção 3.2.1. Como há muito dados, seria inviável realizar a plotagem desses gráficos em todas as gravações, então analisou-se uma amostra por categoria. A amostra analisada, foi escolhida aleatoriamente. Analisaram-se também espectrogramas e após aplicou-se a técnica de extração das características. A segunda etapa do código, foi o processamento, onde se utilizou diretamente a técnica MFCC para todos os arquivos de áudio das sete categorias em ambos os casos, onde as características dos áudios foram armazenadas em arquivos separados.

### 4.2.1. Pré-processamento

Realizou-se uma análise dos arquivos de áudio de cada categoria de resina para os casos 1 e 2, para a visualização do gráfico da forma de onda original, e depois aplicou-se a FFT que transforma o sinal de áudio, que estava em função do tempo, para função da frequência. Nesta etapa, inseriu-se um novo arquivo com as amostras do caso 3, amassadas de forma mais lenta para observar se as características da FFT se mantinham e utilizadas posteriormente no conjunto de teste da rede neural artificial.

No código, primeiramente, carregou-se o arquivo de áudio e plotou-se a forma de onda original das sete resinas e aplicou-se a FFT para a visualização do gráfico em função da frequência com uma taxa de amostragem de 44.100 Hz. É necessário também, o entendimento de como as frequências estão contribuindo para o som em geral ao longo do tempo e por isso aplicou-se a transformada de Fourier de tempo curto (STFT), que na sua forma discreta pode ser definido como a Equação (4), que está na seção 3.2.1 e plotou-se um espectrograma que fornece informações sobre a amplitude em função do tempo.

Em seguida, aplicou-se a técnica MFCC, dividindo o sinal em segmentos. A resposta da amplitude fornece a potência do espectro do sinal de áudio que contém muitas informações desnecessárias. Portanto, o banco de filtros MFCC é usado para estimar a energia presente em cada região de frequência. A escala de Mel correlaciona a frequência percebida na área com sua frequência de medição original. Essa conversão foi feita utilizando a Equação (6), conforme a seção 3.2.2 (BENBA; JILBAB; HAMMOUCH; 2016). As características extraídas dos áudios compõem uma matriz, contendo os vetores com os coeficientes MFCC. O fluxograma da etapa de pré-processamento dos dados pode ser visualizado pela Figura 4.5.



Fonte: Autora, 2023.

O programa foi desenvolvido em algoritmos em linguagem Python, conforme o *script* no Ap. A, que segue as seguintes funções:

Importação das bibliotecas Python:

- *matplotlib*: para criação e plotagem de gráficos e figuras;
- *librosa*: para análise de áudio;
- *librosa.display*: para visualização dos dados como espectrogramas que é construído a partir da biblioteca *matplotlib*;
- *numpy*: utilizada aqui, para a operação matemática da transformada de Fourier.

2. Carregamento de um arquivo de áudio;

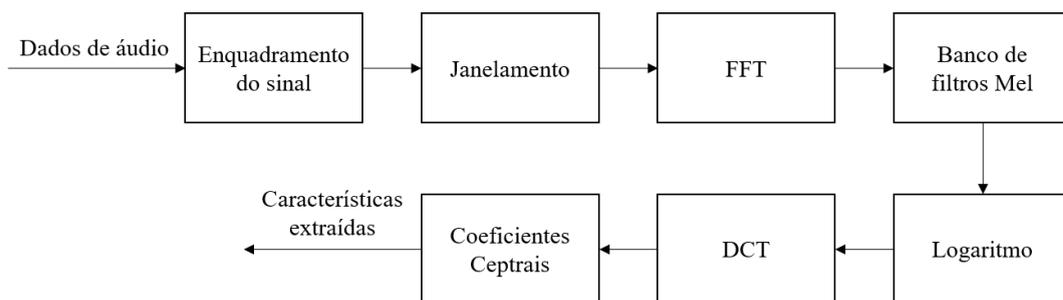
3. Visualização do gráfico da forma de onda original;

4. Aplicação da Transformada Rápida de Fourier (FFT);
5. Aplicação da Transformada de Fourier de tempo curto (STFT);
6. Extração das características MFCC's;
7. Plotagem do espectrograma.

#### 4.2.2. Processamento

Esta seção explica como foi realizado o programa de processamento dos sinais, para todos os casos, onde se extraíram as características dos áudios. Aplicou-se a técnica MFCC, conforme explicado na revisão de literatura, no item 3.2.2 e no fluxograma da Figura 4.6 para a extração das características, utilizadas como entradas na rede neural artificial.

Figura 4.6: Fluxograma da etapa de processamento dos dados.



Fonte: Autora, 2023.

Armazenou-se em arquivos adjacentes no formato JSON (*JavaScript Object Notation*), um formato compacto, aberto e independente de padrão para troca de dados fácil e rápida entre sistemas. Esse formato armazena informações estruturadas e usado principalmente para transferência de dados. As vantagens de usar arquivos neste formato são que a sintaxe é simples e fácil de aprender e interpretar; o tamanho dos arquivos resultantes é muito menor em comparação com outras linguagens intermediárias, como XML (*Extensible Markup Language*). Os tipos de dados que podem ser armazenados são *strings*, números, booleanos e nulos (JSON - encoder and decoder). Neste gerado para a pesquisa, contém informações como os gêneros, que são os tipos de resinas plásticas recicláveis (PET, PEAD, PVC, PEBD, PP, PS e Outros), os vetores MFCC para cada segmento (como exemplo: [-338.15258789][49.42590332][-41.34419632]), utilizados como entrada nas redes e os *labels* (rótulos) os quais são as categorias pertencentes. Tendo como índice 0 – PET, índice 1 –

PEAD, índice 2 – PVC, índice 3 – PEBD, índice 4 – PP, índice 5 – PS e índice 6 – Outros. Extraí as características dos dados e gerei arquivos para o caso 1 (amassadas em conjunto), caso 2 (por peça).

O número de segmentos é necessário, pois no aprendizado profundo necessita-se de muitos dados, na técnica MFCC divide-se cada dado em segmentos, para que ao invés de salvar como entrada, salvar como segmentos. Os vetores MFCC extraídos são os próprios dados de treinamento, teste e validação que foram utilizados na rede neural artificial para a classificação e os *labels* são os resultados esperados na classificação.

O programa desenvolvido para o processamento dos dados apresenta as seguintes funções e pode ser visualizado no Ap. B.

#### 1. Importação das bibliotecas Python:

- *os*: para manipulação de arquivos;
- *librosa*: para análise de áudio;
- *math*: manipulação de funções matemáticas estabelecido pelo padrão C;
- *json*: para o armazenamento das características extraídas do coeficiente MFCC.

#### 2. Carregamento dos arquivos de áudio;

#### 3. Construção de um dicionário para armazenar os dados;

#### 4. Percorrer todas as pastas e salvar os *labels*;

#### 5. Processamento dos segmentos MFCC e armazenamento dos dados;

#### 6. Salvar as informações na pasta adjacente.

Com o programa, geraram-se os arquivos JSON que foram salvos para a utilização na etapa de classificação.

### 4.3. Classificação

Nessa seção será mostrado como foram desenvolvidos os programas para a classificação dos áudios de plástico, utilizando modelos como RNC e Redes neurais recorrentes LSTM e os parâmetros de desempenho utilizados para a avaliação dos modelos neurais.

### 4.3.1. Redes Neurais Convolucionais (RNC)

Como na rede neural convolucional normalmente se utilizam imagens, as quais são uma matriz de pixels, as entradas que se utilizaram para a classificação do áudio são os espectrogramas, que podem ser interpretados como uma imagem. Nos espectrogramas, o tempo, nos diferentes intervalos, e a frequência são como índice  $x$  e  $y$  para os pixels de uma imagem e a amplitude podem ser encaminhados para o valor associado a cada pixel. Assim, o espectrograma é uma matriz bidimensional que é comparável a uma imagem. E para haver a forma de dados corretas, configura-se a técnica de extração MFCC, com o número de segmentos, o comprimento do salto e o número total de amostras do arquivo de áudio, obtendo uma forma de dado com janela, sendo a divisão do número total de amostra com o comprimento do salto; o número de segmentos e uma terceira dimensão que é a profundidade, dado igual a 1. Com essa estrutura todos os dados podem ser comparados a imagens em uma escala de cinza (WYSE, 2017). Isso é necessário, pois é o formato que o *tensorflow* do Python aceita para o modelo neural convolucional.

Foi construído um algoritmo para o modelo de Rede Neural Convolucional, conforme Ap. C, que segue as seguintes funções:

#### 1. Importação das bibliotecas Python:

- *json*: para abrir e ler os arquivos adjacentes;
- *numpy*: para executar operações matemáticas;
- *sklearn.model\_selection*: para a divisão das matrizes em subconjuntos aleatórios para treinamento e teste;
- *tensorflow.keras*: para utilização da técnica de aprendizado profundo, onde contém os modelos neurais;
- *matplotlib.pyplot*: para criação e plotagem de gráficos e figuras;
- *sklearn.metrics*: ferramenta para avaliar o modelo neural;
- *itertools*: para iterar e retornar funções.

#### 2. Carregamento do conjunto de dados;

#### 3. Divisão dos dados em conjuntos de treinamento, teste e validação;

#### 4. Construção da arquitetura da rede neural;

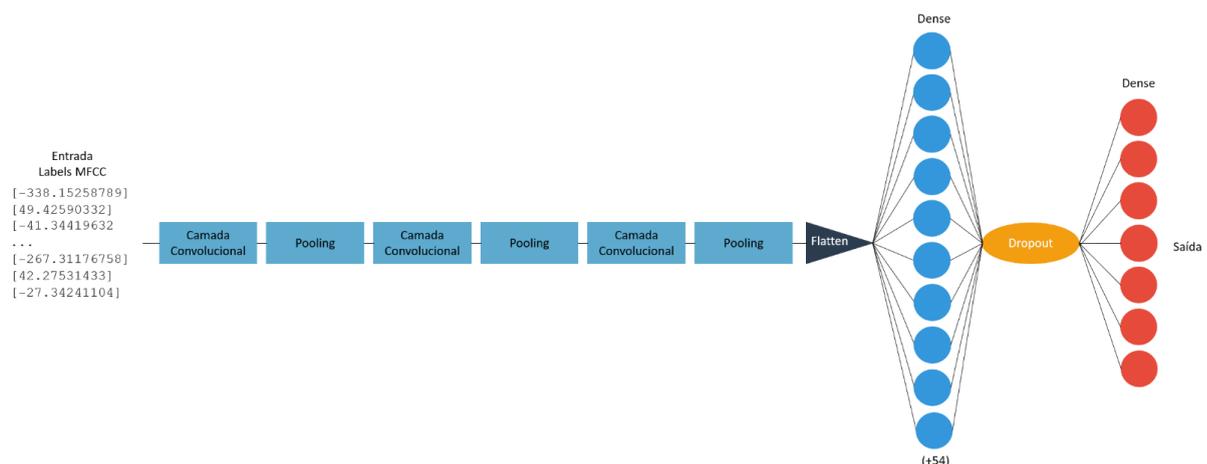
#### 5. Otimizador;

6. Treinamento da rede;
7. Gráfico de precisão e erro para validação do treinamento;
8. Avaliação da rede no conjunto de teste;
9. Realização de uma previsão de uma amostra;
10. Plotagem da matriz de confusão.

Carregou-se o conjunto de dados dos arquivos JSON do caso 1 e caso 2, um de cada vez, que contém as características extraídas. No item 3, a divisão dos dados de conjunto ficaram 80% para o treinamento da rede e 20% para realizar o teste, e desses últimos, 20% para a validação. Essa divisão é válida, pois não desprezam o erro de aproximação e a maximização do erro é a menor dentre as existentes e os melhores resultados são alcançados utilizando essa divisão (GHOLAMY; KREINOVICH; KASHELEVA, 2018).

A arquitetura da RNC utilizada pode ser visualizada pelo esquema ilustrativo que está na Figura 4.7, onde se tem como entrada os labels MFCC e em seguida as camadas convolucionais, pooling, flatten, dense (com 64 neurônios), dropout e dense (sendo a camada final com 7 neurônios a qual são as categorias classificadas).

Figura 4.7: Esquema ilustrativa da arquitetura da RNC utilizada na pesquisa.



Fonte: Autora, 2022.

A construção da arquitetura da rede neural (item 4), foi composta por 3 camadas convolucionais, contendo: a quantidade filtros, o tamanho desses filtros e a função de ativação, A função utilizada foi a ReLU (unidade linear retificada), que retorna 0 para todos os valores negativos. Na camada convolutacional, contém uma camada *Pooling max*, que irá

reduzir o tamanho dos parâmetros de entrada, tendo como informações o tamanho da grade, dos passos e o tipo de preenchimento utilizado foi o *same*, que preenche com valores de zero e também uma camada de normalização que padroniza as ativações, acelera o treinamento, obtendo assim um modelo mais confiável (CLAPPIS, 2019).

Em seguida é colocada uma camada *Flatten*, onde a matriz obtida pela camada anterior de convolução e agrupamento é dimensionada em uma matriz linear unidimensional. Esta etapa é a preparação para entrar na camada principal da rede neural. Utiliza-se a camada *Dense* para implementar uma rede neural totalmente conectada, a dimensão de saída é o número de neurônios, a função de ativação é a mesma utilizada no treinamento. Na camada *Dropout*, a regularização é realizada para desligar aleatoriamente alguns neurônios e suas conexões apenas durante o treinamento. Durante a previsão, todos os neurônios permanecem ativos. A razão para isso é evitar *overfitting* durante o treinamento, informando a porcentagem de abandono (CLAPPIS, 2019).

Finalmente, tem-se outra camada *Dense*, onde a dimensão deve ser igual ao número de categorias a serem classificadas usando a função de ativação *softmax*, onde transforma os valores de cada neurônio que se tem no final da rede neural artificial em probabilidades, para ocorrer a classificação. O cálculo é realizado com a Equação (15) do item 3.3.1.1 e no final o valor de probabilidade tem que somar 1.

O otimizador (item 5) utilizado foi o *Adam*, um método estocástico, com eficiência computacional, com baixos requisitos de memória, redimensionamento da constante de gradiente, ideal para problemas de quantidade de dados ou parâmetros grandes. A taxa de aprendizagem utilizada foi de 0,0001 que é o padrão utilizado para o otimizador (KINGMA; BA, 2014). A função perda utilizada foi *SparseCategoricalCrossentropy* que calcula a perda de entropia cruzada entre rótulos e previsões, que pode ser determinada através da Equação (16) do item 3.3.1.1 (TensorFlow). A métrica para avaliação do modelo foi a Accuracy, calculada pela Equação (17).

Para o treinamento da rede (item 6) utilizam-se os dados de entrada e saída separados para o treinamento e validação, precisando informar dois parâmetros, como o tamanho do lote igual a 32 e o número de épocas, a qual executou o treinamento, fez-se um teste com os valores 30 e 100 interações, para comparação de qual obteve melhor desempenho.

Nos itens 7 e 8, avalia-se o treinamento e teste da rede através de um gráfico de acurácia e perda e depois é informada a porcentagem de acurácia do conjunto de teste. Após, realiza-se uma previsão de uma amostra (item 9) não vista no treinamento, comparando se a classe prevista corresponde com a real. A resposta é dada em índice e cada um representa uma

categoria de resina plástica, numerados de 0 a 6. Tendo como índice 0 – PET, índice 1 – PEAD, índice 2 – PVC, índice 3 – PEBD, índice 4 – PP, índice 5 – PS e índice 6 – Outros. A matriz de confusão, último item, informará a quantidade de dados previsto corretamente para cada categoria e quais foram confundidas, utilizando os dados de teste e baseado no cálculo da *Accuracy*.

#### 4.3.2. Redes Neurais Recorrentes LSTM

Como os áudios são dados com série temporal, coletados em intervalos de tempo específicos, implementou-se uma rede neural recorrente, utilizada para dados que a ordem é importante, podendo entender o contexto e processar esses dados mantendo a sequência. Para resolver o problema de memória de curto prazo, utilizou-se a rede neural recorrente LSTM, redes de memória de longo prazo, que permite aprender padrões de prazos difusos. Para o modelo de Rede Neural Recorrente, também foi construído um algoritmo em linguagem Python, conforme Ap. D, que segue as seguintes funções:

##### 1. Importação das bibliotecas Python:

- *json*: para abrir e ler os arquivos adjacentes;
- *numpy*: para executar operações matemáticas;
- *sklearn.model\_selection*: para a divisão das matrizes em subconjuntos aleatórios para treinamento e teste;
- *tensorflow.keras*: para utilização da técnica de aprendizado profundo, onde contém os modelos neurais;
- *matplotlib.pyplot*: para criação e plotagem de gráficos e figuras;
- *sklearn.metrics*: ferramenta para avaliar o modelo neural;
- *itertools*: para iterar e retornar funções.

##### 2. Carregamento do conjunto de dados;

##### 3. Divisão dos dados em conjuntos de treinamento, teste e validação;

##### 4. Construção da arquitetura da rede neural;

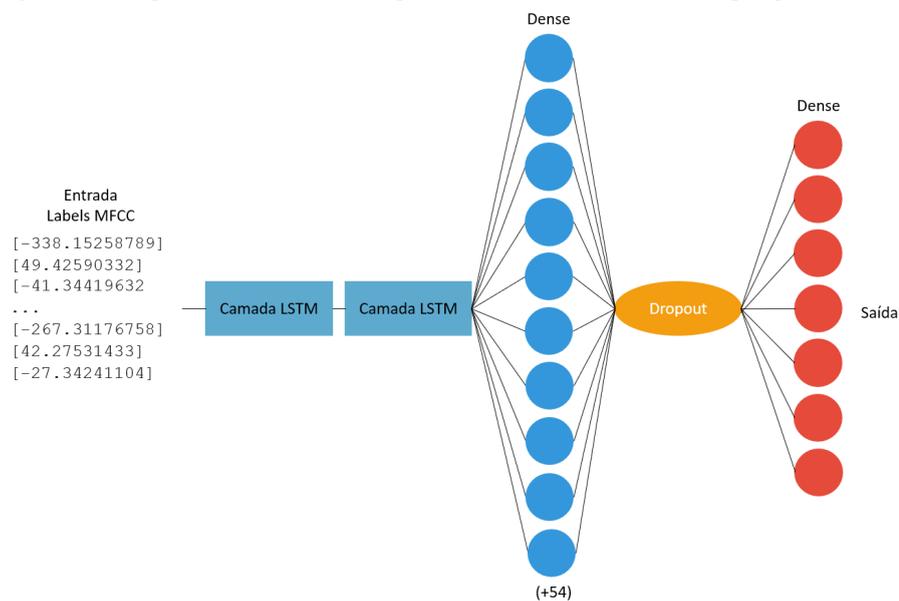
##### 5. Otimizador;

##### 6. Treinamento da rede;

7. Gráfico de precisão e erro para validação do treinamento;
8. Avaliação da rede no conjunto de teste;
9. Realização de uma previsão de uma amostra;
10. Plotagem da matriz de confusão.

Por possuir retroalimentação, as redes neurais recorrentes podem processar não apenas pontos de dados únicos, mas sequencias de dados inteiras. Na construção do modelo da rede LSTM, tem apenas duas dimensões, a primeira é o número de passos de tempo e a segunda os coeficientes reais MFCC que foram extraídos quando se processaram os dados. A diferença do modelo da rede neural recorrente do tipo LSTM para a rede neural convolucional na parte de códigos é somente a arquitetura, o esquema ilustrativo da arquitetura pode ser visto na Figura 4.8, que tem como entrada os labels MFCC, camadas LSTM, camada dense (com 64 neurônios), dropout e por último a dense com 7 neurônios sendo as categorias.

Figura 4.8: Esquema ilustrativa da arquitetura da LSTM utilizada na pesquisa.



Fonte: Autora, 2022.

A arquitetura foi constituída por duas camadas LSTM, sendo a primeira contendo parâmetros como o número de neurônios, a forma de entrada e uma sequência de retorno, que retorna a última saída da rede. A segunda camada LSTM é igual à modelada, indicando somente o número de neurônios. Em sequência tem uma camada *Dense*, informando a dimensão de saída e a função de ativação, também utilizada a ReLU e uma camada *Dropout*

indicando a porcentagem de abandono, a fim de evitar o *overfitting* durante o treinamento. E por fim, mais uma camada *Dense*, indicando a dimensão que dever ser igual ao número de categorias e a função de ativação *softmax*. Os itens 5 ao 10 da rede neural recorrente LSTM foram construídos iguais ao modelo do programa da rede neural convolucional.

#### 4.3.3. Validação

Para uma última avaliação do desempenho dos modelos, acrescentou-se dados somente no conjunto de teste, e gravadas 20 amostras amassadas de forma mais lenta, iguais aos utilizados no pré-processamento na etapa da FFT, denominadas caso 3, conforme citado no item 4.2.1. O objetivo é avaliar se os modelos para as duas redes (RNC e LSTM) se mantêm utilizando dados de sequências temporais diferentes das utilizadas anteriormente, no treinamento.

Esse teste foi pensando para o dia-a-dia das cooperativas de reciclagem, onde, não haverá uma pessoa especializada para retreinar a rede com frequência. E como há vários trabalhadores, dificilmente amassarão as amostras de plásticos em uma mesma velocidade. Então, o propósito do teste de “amassar lentamente” é para simular uma condição desfavorável e muito provável de ocorrer no cotidiano das cooperativas.

## Capítulo 5 – RESULTADOS E DISCUSSÕES

Este capítulo descreve os resultados da aquisição dos dados e uma comparação entre as redes neurais através de uma discussão.

### 5.1 Pré-processamento

Primeiramente, carregaram-se os arquivos de áudio de cada categoria, obtendo-se um sinal que é uma matriz *numpy* (multidimensional) e plotou-se o gráfico da forma de onda original em função do tempo e amplitude. Estes gráficos podem ser visualizados na Figura 5.1 que correspondem as categorias PET, PEAD, PVC, PEBD, PP, PS e Outros, para o caso 1 (amassados em conjunto) e caso 2 (amassados por peça).

Figura 5.1: Forma de onda original do caso 1 (amassados em conjunto) e do caso 2 (amassados por peça) para todas as categorias.

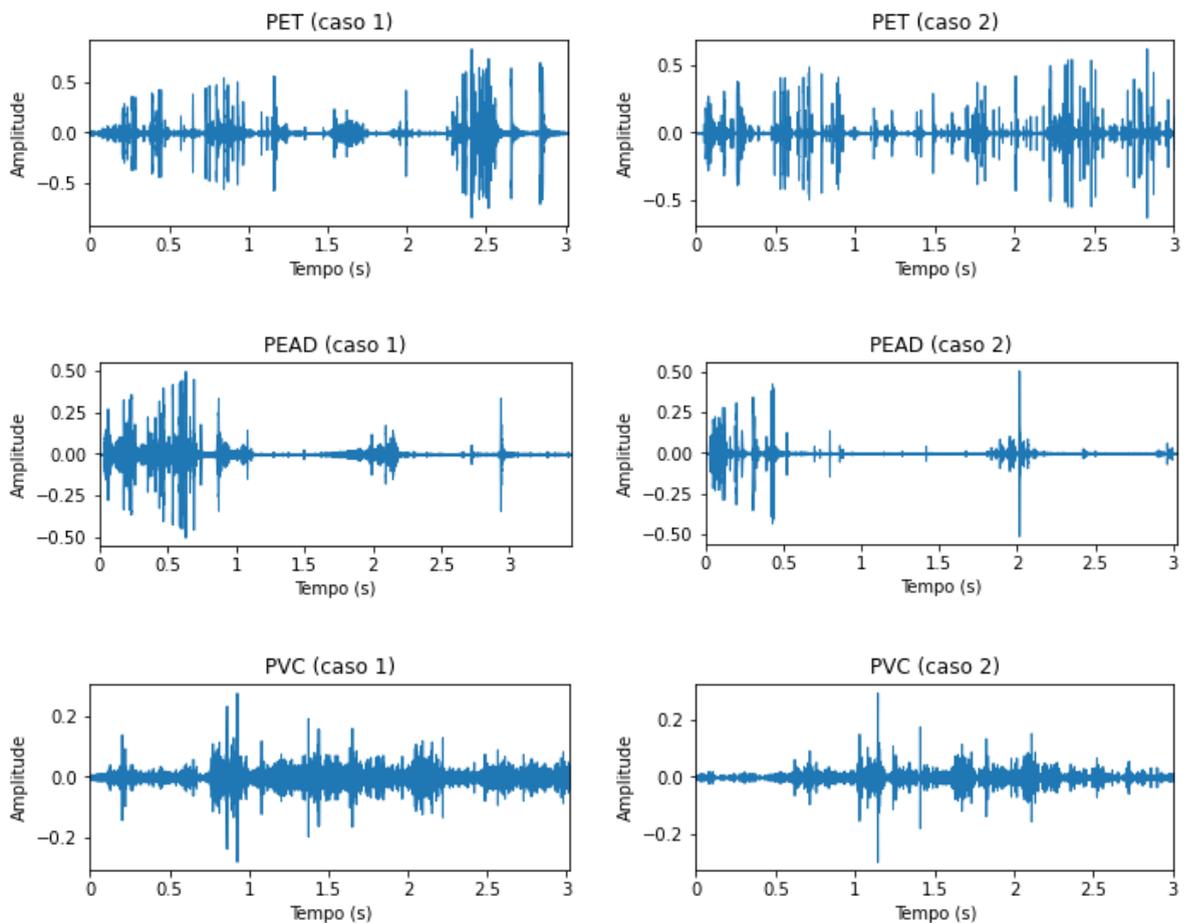
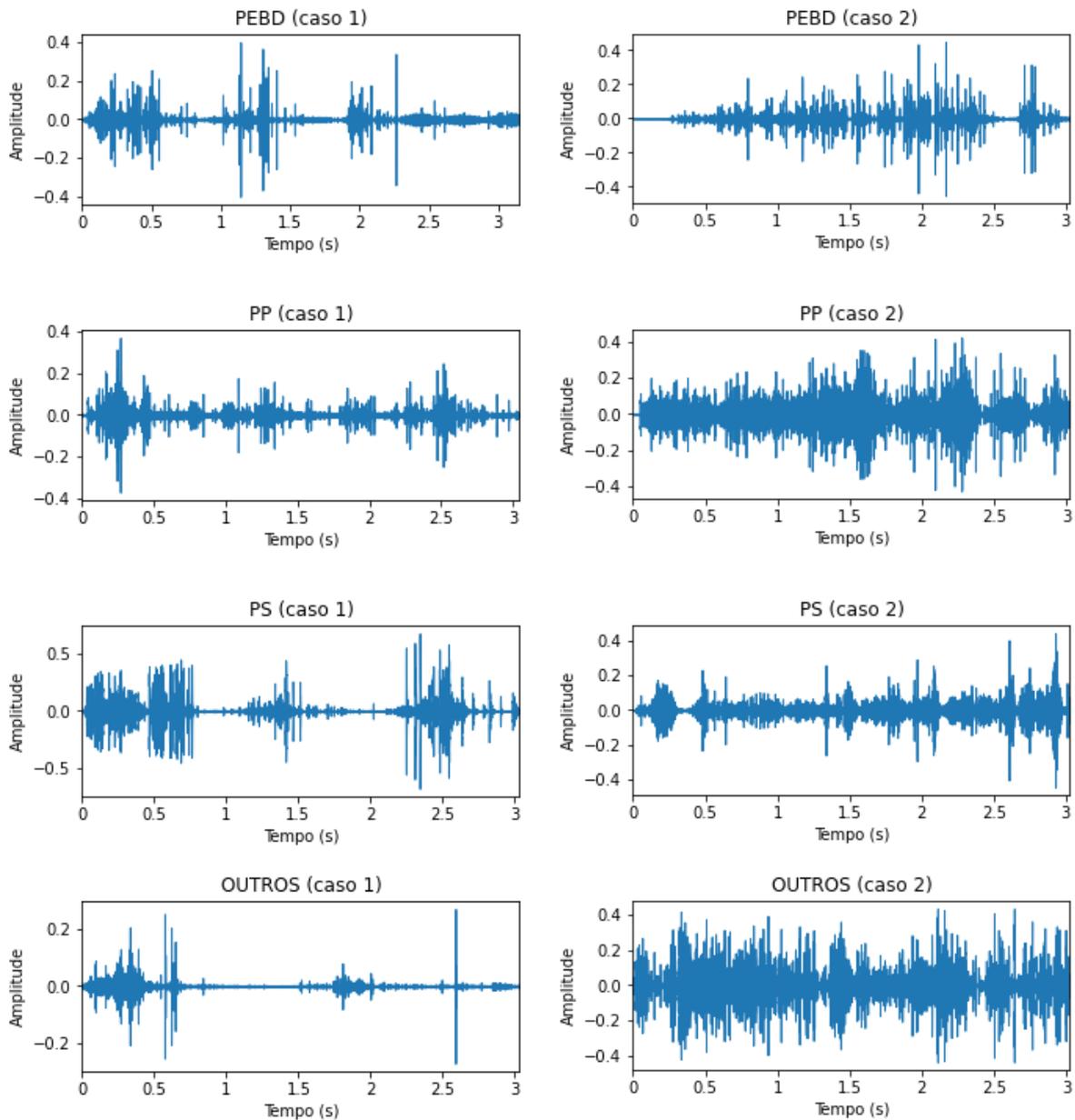


Figura 5.1: Forma de onda original do caso 1 (amassados em conjunto) e do caso 2 (amassados por peça) para todas as categorias.



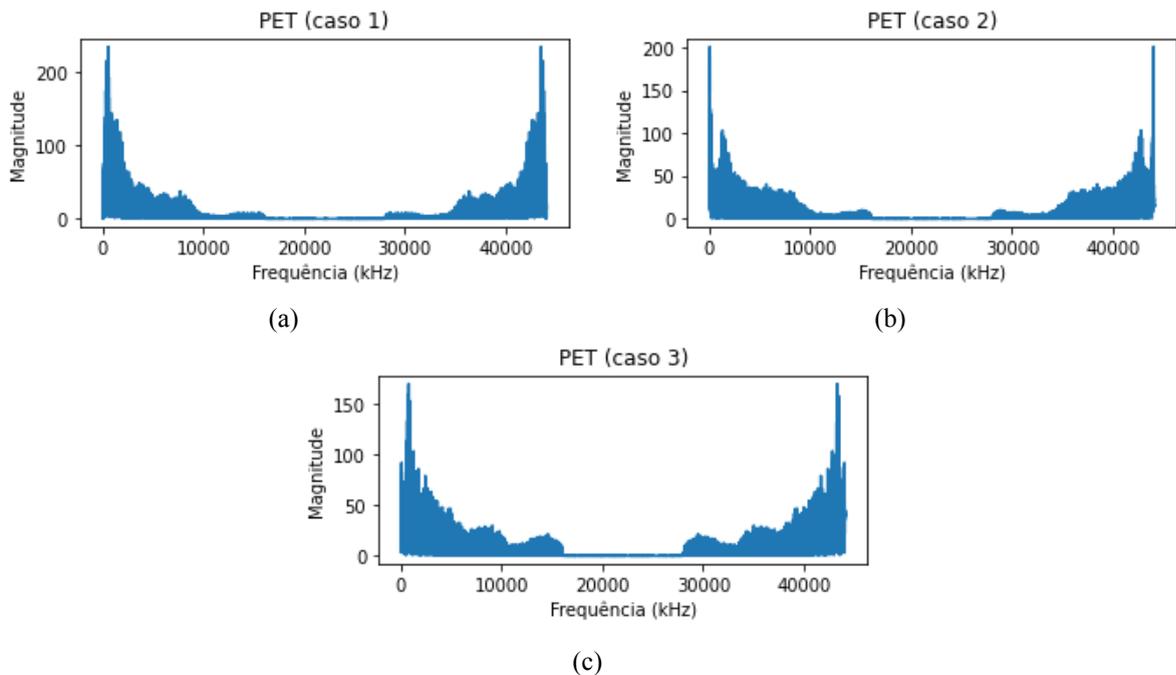
Fonte: Autora, 2022.

Observa-se que as formas de ondas originais dos áudios gravados durante o processo de deformação dos plásticos se diferenciam para cada tipo de resina. E fazendo um comparativo entre os dois casos (1 e 2), há mudanças nas amplitudes da onda sonora. Essa diferença ocorre por conta da intensidade, entendida também como sendo o “volume do som”.

Para conversão do sinal em componentes espectrais individuais e fornecimento de informações da frequência, realizou-se o cálculo da FFT para todas as categorias e para os casos 1 e 2. Nesta etapa, inseriu-se os dados do caso 3 – amostras amassadas em conjunto, porém com menor velocidade no esmagamento - conforme explicado no item 4.3.3.

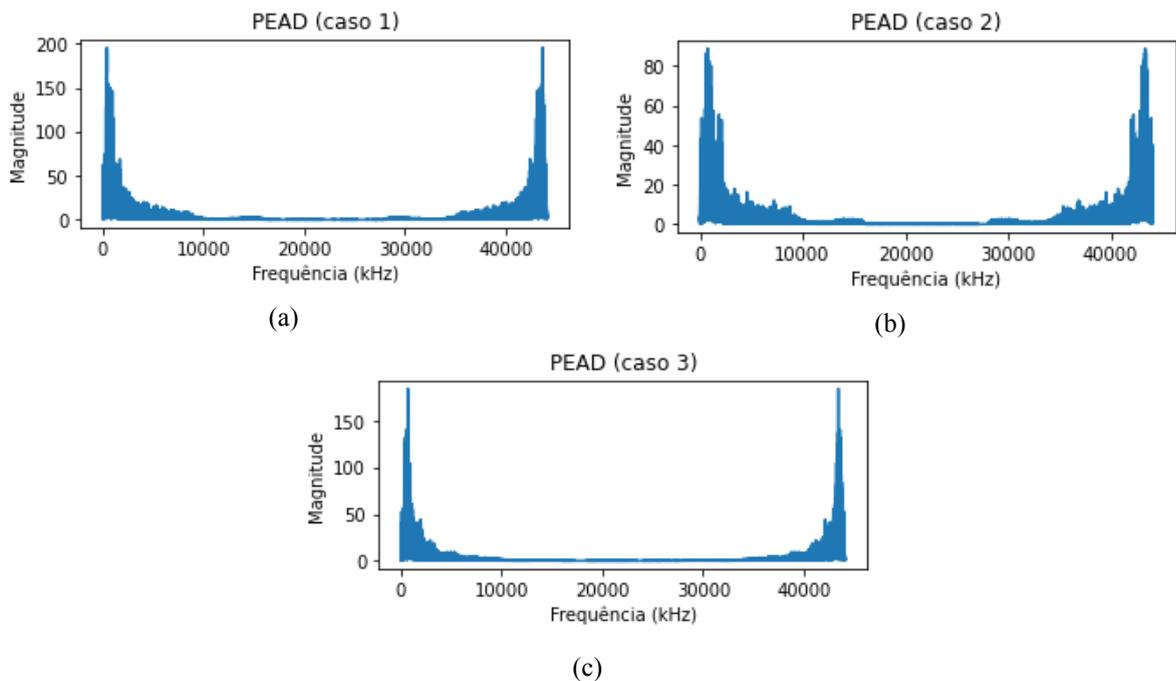
Os gráficos podem ser analisados na Figura 5.2 para PET, Figura 5.3 para PEAD, sendo (a) para o caso 1, (b) para o caso 2 e (c) para o caso 3. Os gráficos das resinas restantes podem ser vistos no Ap. E.

Figura 5.2: Gráfico da transformada rápida de Fourier da categoria PET (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta,



Fonte: Autora, 2022.

Figura 5.3: Gráfico da transformada rápida de Fourier da categoria PEAD (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.



Fonte: Autora, 2022.

Observa-se que os gráficos da transformada rápida de Fourier em função da frequência são simétricos, uma propriedade da Transformada de Fourier, dado pelo teorema de Nyquist (PEREIRA, 2021, p. 19; LIMA, 2022, p. 26). O sinal convertido em frequência foi processado, conforme a seção 4.2.2, com a taxa de amostragem 44.100 Hz.

A categoria PVC para o caso 3, amostras amassadas de forma mais lenta, foi o que obteve maiores diferenças em relação ao caso 1 e 2, por conta da magnitude do sinal, que tem a ver com o volume da amostra. Como foram amassadas lentamente, o barulho que a amostra fez ao amassar foi menor que para os outros casos.

O problema é que o espectro de potência é um instantâneo estático, ou seja, dura somente instantes e depois para em todo o som, considerado a média da energia, que se refere a magnitude dos diferentes períodos de frequência. Portanto, é necessário entender como as frequências estão contribuindo para o sinal em geral ao longo do tempo, e para solucionar, realiza-se a STFT que é um método que pega uma “janela” que desliza ao longo da série temporal e realiza a DFT no segmento dependente do tempo, obtendo-se espectrogramas que fornecem informações sobre a amplitude (cor) em função da frequência e do tempo. Os gráficos para o caso 1 e caso 2 dos setes categorias podem ser vistos pela Figura 5.4.

Figura 5.4: Espectrograma STFT tempo x frequência x amplitude para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).

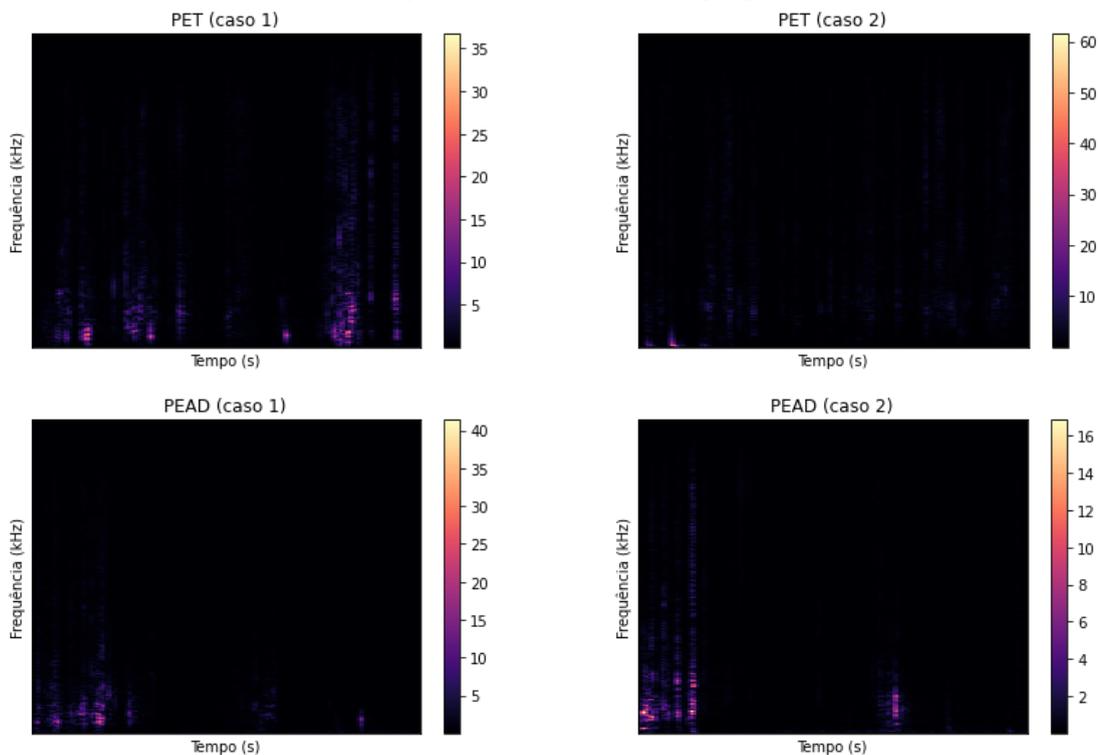


Figura 5.4: Espectrograma STFT tempo x frequência x amplitude para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).

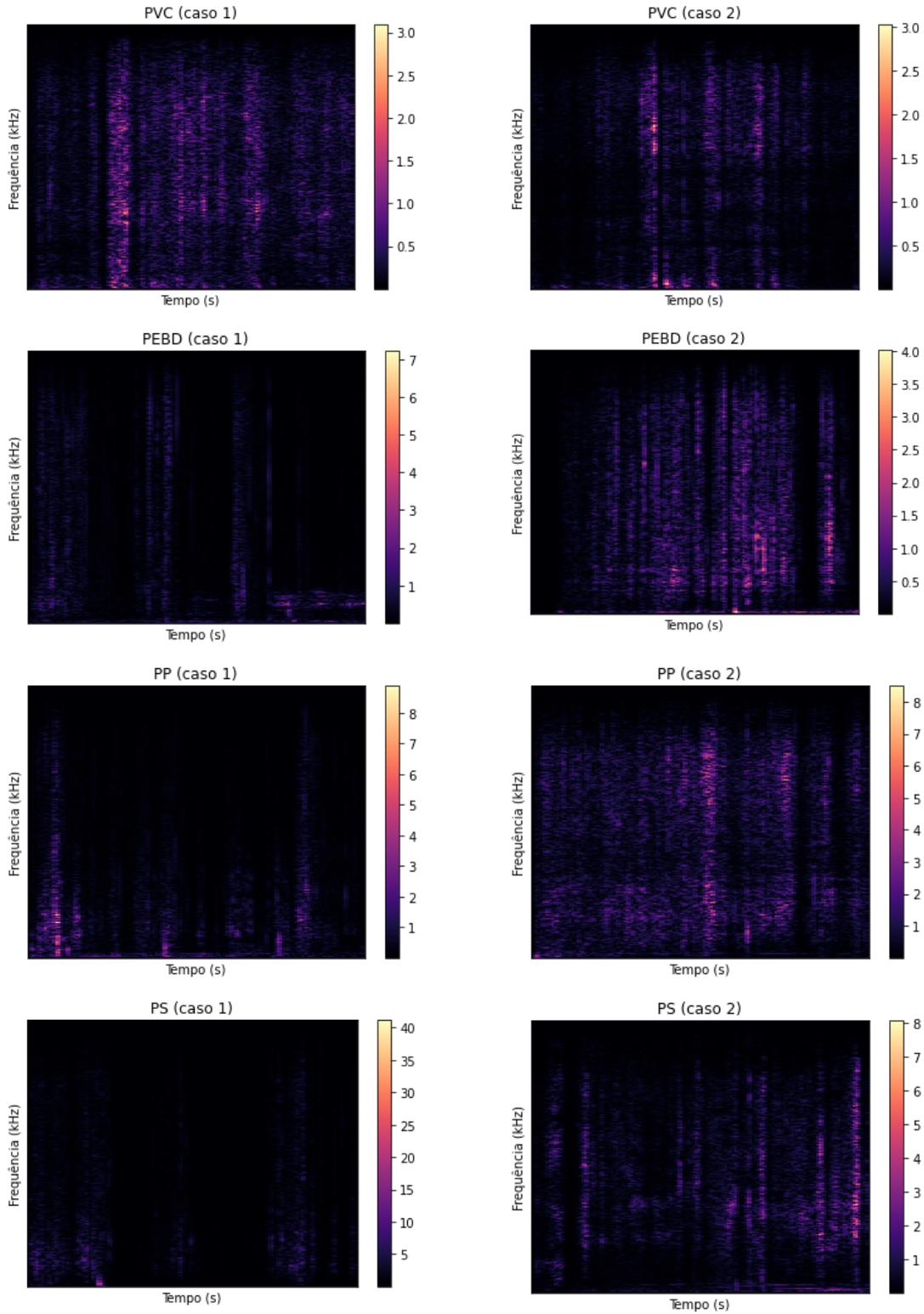
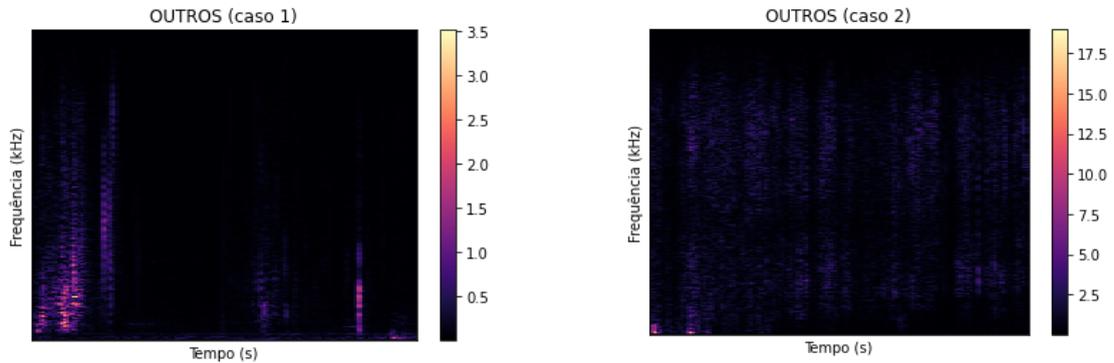


Figura 5.4: Espectrograma STFT tempo x frequência x amplitude para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).



Fonte: Autora, 2022

Nos espectrogramas de STFT consegue-se visualizar dados que se assemelham com um mapa de calor, contendo o eixo x, eixo y e um terceiro item referente a cor. Pode-se perceber também certas explosões semelhantes de energia nas frequências mais baixas. Há uma maneira de suavizar essa amplitude e visualizar de forma mais agradável, um logarítmico, chamado de log espectrograma, onde converte a amplitude do espectro original para decibéis (dB), que indica o volume do som, se está alto ou baixo para o ouvido humano. A Figura 5.5 mostra os gráficos em função do tempo, frequência e dB para as sete categorias de resinas, para o caso 1 e caso 2.

Figura 5.5: Log espectrograma tempo x frequência x dB para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).

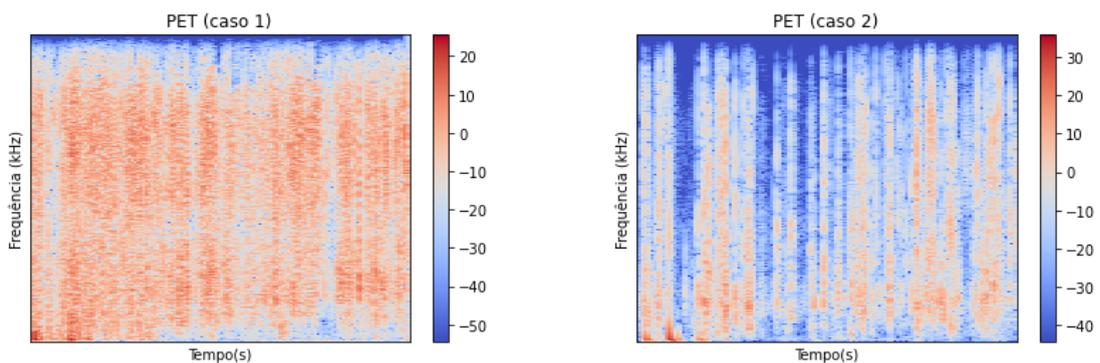


Figura 5.5: Log espectrograma tempo x frequência x dB para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).

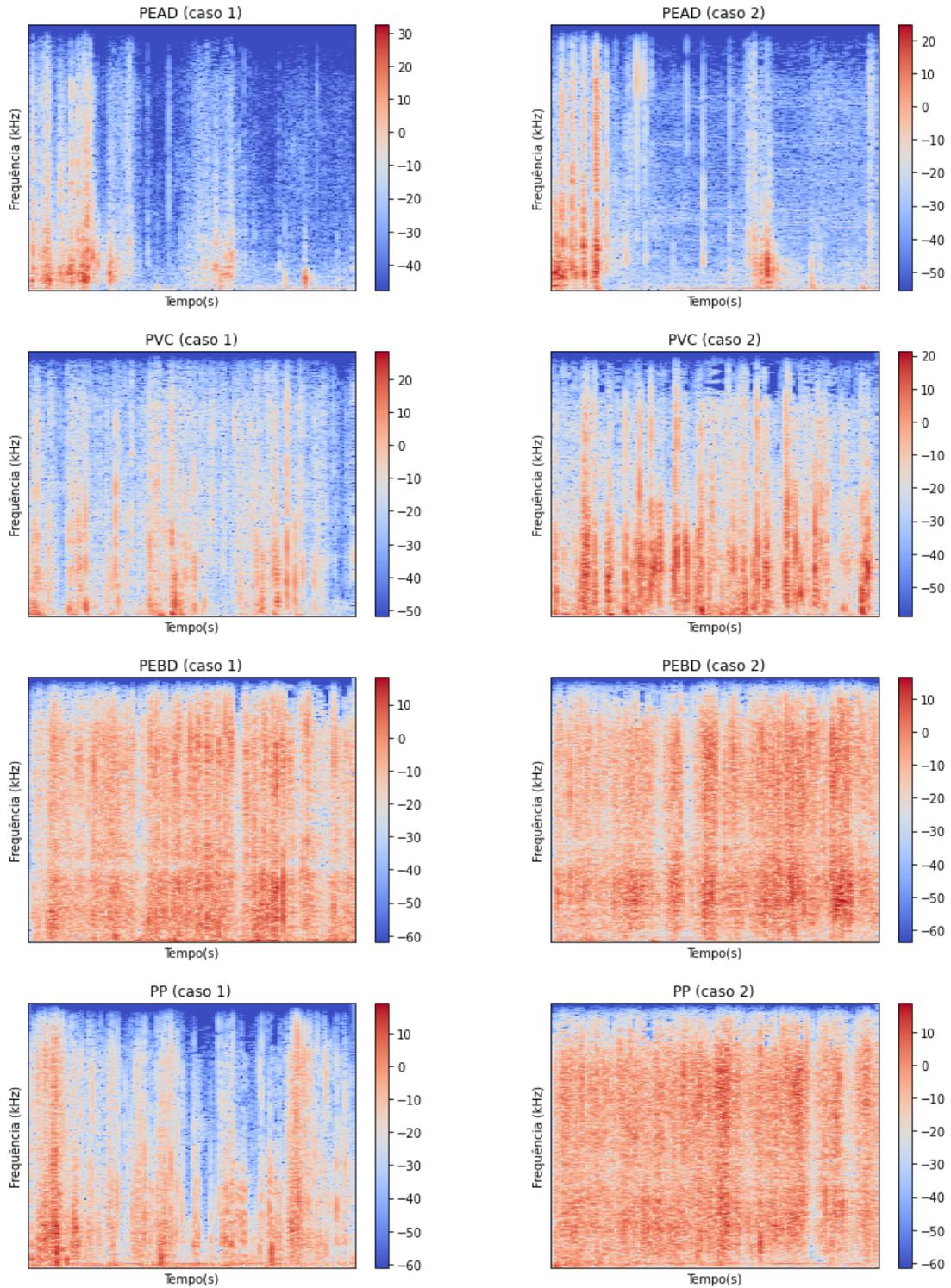
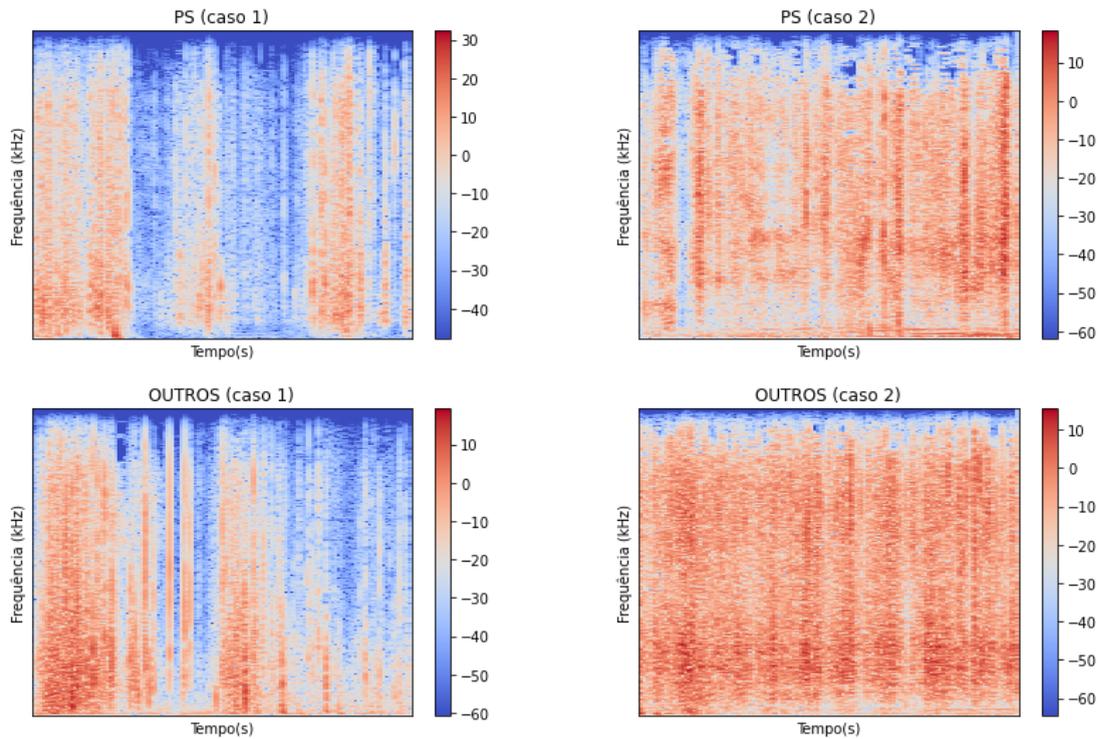


Figura 5.5: Log espectrograma tempo x frequência x dB para todas categorias e caso 1 (amassados em conjunto) e caso 2 (amassados por peça).



Fonte: Autora, 2022.

Observa-se que o espectrograma fica mais fácil de entender, a barra colorida representa dB, é a intensidade do som, quanto maior a onda original (Figura 5.1), mais alto o volume. Um valor dB positivo indica uma razão (ganho) maior que 1, valores negativos indicam uma razão menor que 1 (decaimento) e um valor de 0 significa que não haver mudança de sinal. A vermelho nos gráficos tem intensidades maiores e o azul menores intensidades. À medida que os valores vão ficando negativos, significa que o som fica menos perceptível ao ouvido humano.

Em relação à diferença nas frequências entre o caso 1 e caso 2, é por conta que o processo de amassamento das amostras ter sido diferente. Para o caso 1, as amostras para cada categoria de resina foram amassadas em conjunto dentro do bag de pano com auxílio de uma morsa manual, sendo padronizadas. Enquanto para o caso 2, as amostras foram amassadas inteiras, sem o bag de pano e nem com o auxílio da morsa, fazendo com que as amostras divergissem.

Para a aplicação do MFCC é necessário calcular a FFT e dividiu-se em segmentos. O uso de cerca de 20 coeficientes MFCC é comum na Automatic Speech Recognition (ASR), embora 13 coeficientes sejam frequentemente considerados suficientes para a codificação e foi eficiente para os dados de áudio nessa pesquisa. Não se observou diferença de

desempenho entre 20 e 13 segmentos, por isso, decidiu-se manter 13 segmentos (MCFEE *et al.*, 2015). A Figura 5.6 mostra os espectrogramas da técnica de extração das características, para o caso 1 e caso 2 das sete resinas plásticas.

Figura 5.6: Espectrograma MFCC das sete categorias para caso 1 (amassadas em conjunto) e caso 2 (amassadas por peça).

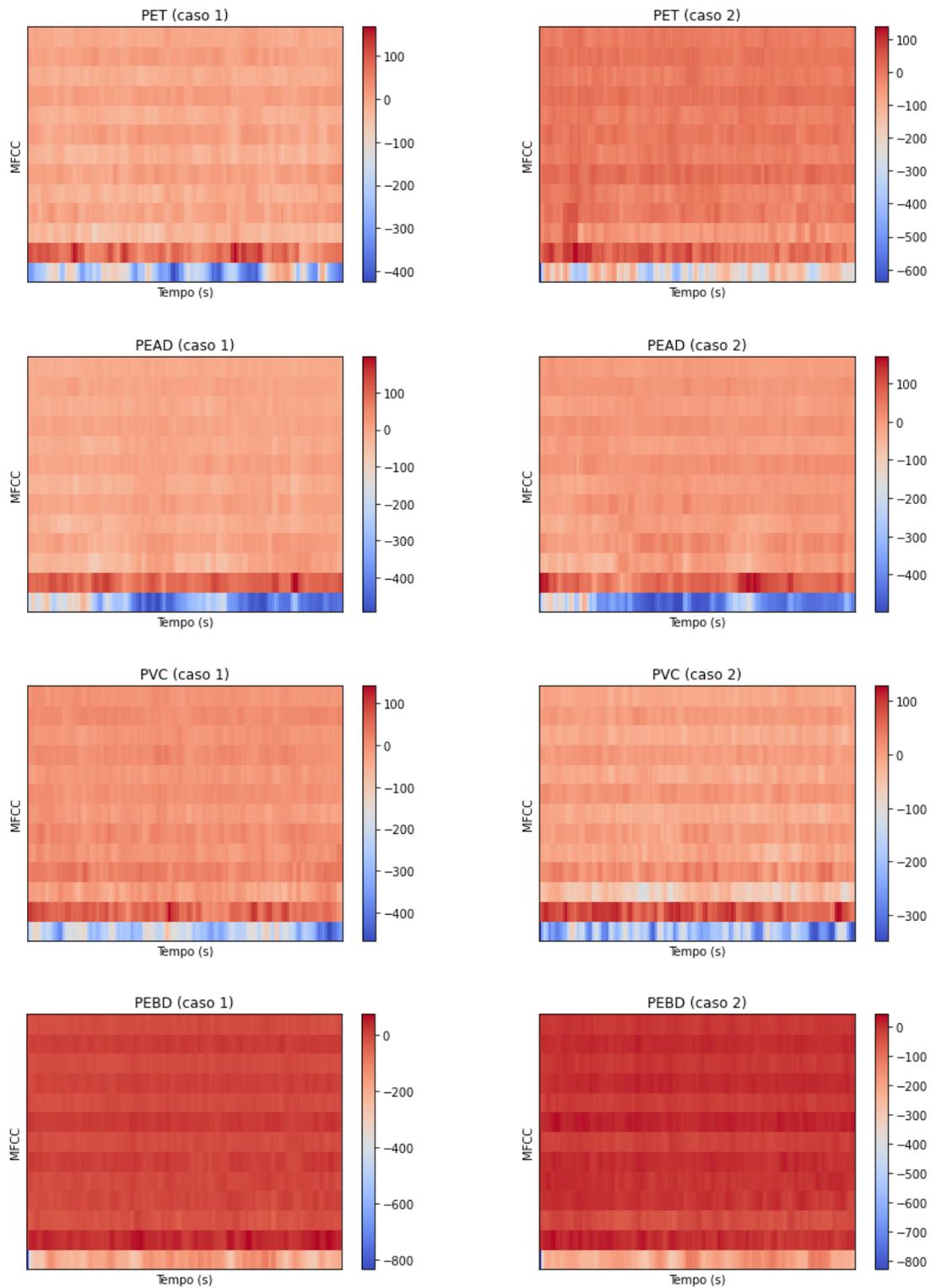
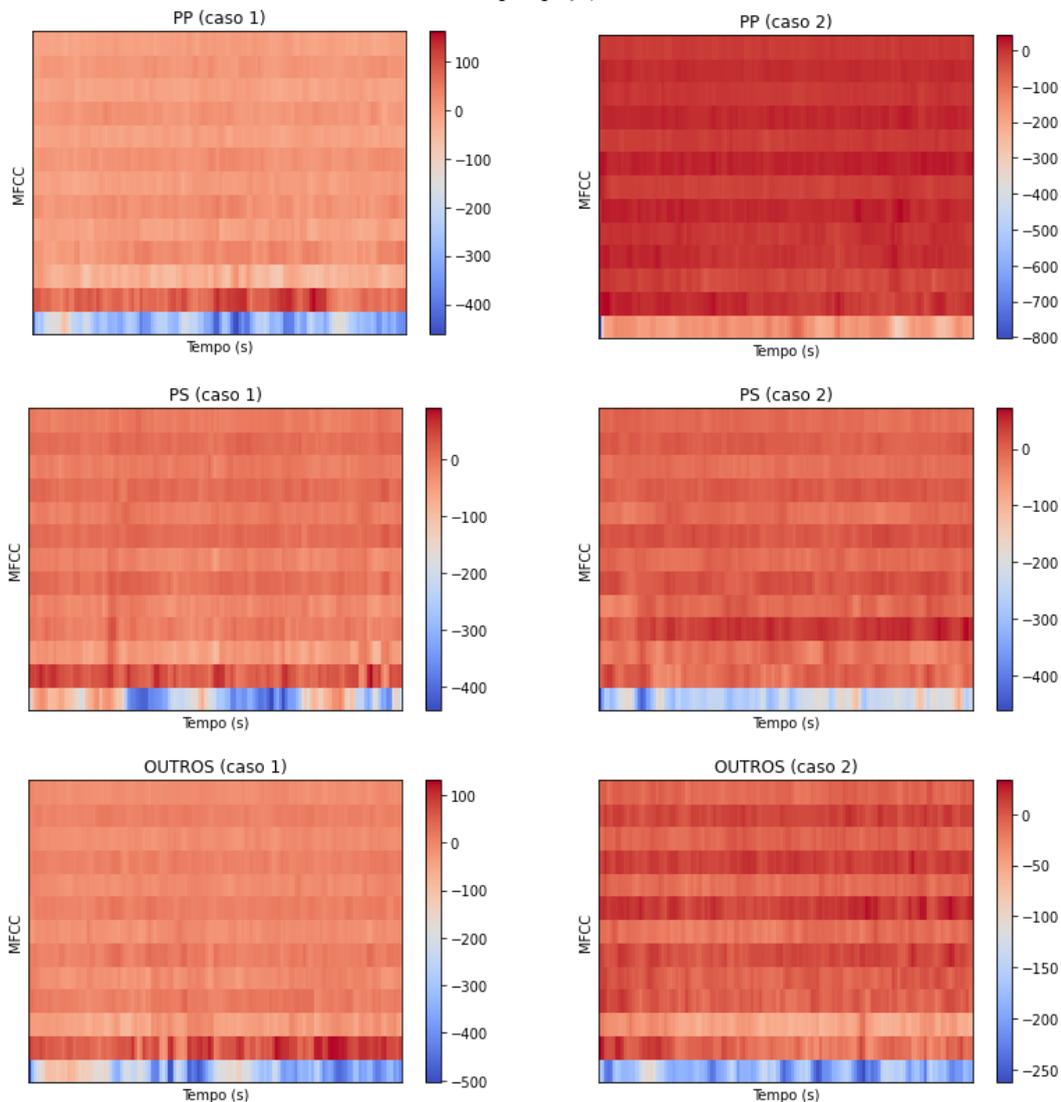


Figura 5.6: Espectrograma MFCC das sete categorias para caso 1 (amassadas em conjunto) e caso 2 (amassadas por peça).



Fonte: Autora, 2022.

Pelo espectrograma do coeficiente MFCC consegue-se observar no eixo y os intervalos com os 13 segmentos e no eixo x tem o tempo que se pode analisar como os diferentes coeficientes MFCC evoluem ao longo dele. Olhando os valores dos coeficientes MFCC obtidos, o valor negativo recebido é porque o coeficiente 0 é produzido tomando o valor logarítmico da energia, conforme a Equação (6), conforme a seção 3.2.2. Ao carregar sinais com código construído a partir da biblioteca *librosa* do Python, os valores são normalizados de -1 a 1 e, geralmente, a energia do quadro é menor que 1. Então, realiza o logaritmo de um valor menor que 1, obtendo assim números negativos. Com esses resultados e gráficos consegue-se ter um entendimento melhor do banco de dados de áudio trabalhado e o uso da técnica de extração.

## 5.2 Processamento

Nesta etapa foi utilizado o código para fazer a extração das características por meio da técnica MFCC, onde se gerou um arquivo adjacente para caso 1 e caso 2 que podem ser visualizados pelo repositório da Unicamp no link: <https://doi.org/10.25824/redu/UGMPJR>.

Os arquivos contêm os nomes das resinas plásticas (PET, PEAD, PVC, PEBD, PP, PS e Outros) os quais são os gêneros, classes ou categorias da classificação. As características extraídas, são os vetores MFCC com os 13 segmentos, utilizados como entrada nas redes neurais e os rótulos pertencentes a cada classe, numeradas de 0 a 6, sendo 0 pertencente a PET, 1 pertencente a PEAD, 2 pertencente a PVC, 3 a PEBD, 4 a PP, 5 a PS e 6 pertencente a categoria Outros.

## 5.3 Classificação

Para a classificação dos áudios dos plásticos, utilizaram-se técnicas de inteligência artificial, a aprendizagem de máquina, como as redes neurais artificiais. Neste trabalho, utilizou-se dois tipos de rede, a primeira chamada de RNC e a segunda de rede neurais recorrentes LSTM

Ao longo desta seção serão apresentados os resultados obtidos na classificação por ambas redes neurais artificiais para o banco de dados das amostras do caso 1 (amassadas em conjunto) e caso 2 (amassadas por peça) e a validação com as amostras do caso 3 (amassadas de forma mais lenta).

### 5.3.1 Rede Neural Convolutacional

Os valores dos parâmetros utilizados em cada camada da rede RNC podem ser vistos na Tabela 5.1.

Para o treinamento, utilizaram-se os dados do primeiro arquivo adjacente JSON para o caso 1 e o segundo arquivo para o caso 2 gerados na parte de processamentos, contendo os tipos de resinas plásticas recicláveis, os vetores MFCC para cada segmento e as classes pertencentes, que podem ser acessados no link na seção 5.2.

Tabela 5.1: Valores dos parâmetros da arquitetura da rede neural convolucional.

| Arquitetura             | Parâmetros          | Valores dos parâmetros |      |
|-------------------------|---------------------|------------------------|------|
| 1ª camada convolucional | Conv2D              | Filtros                | 32   |
|                         |                     | Tamanho do Kernel      | 3x3  |
|                         |                     | Função de ativação     | ReLU |
|                         | MaxPoo 12D          | Tamanho do Kernel      | 3x3  |
|                         |                     | Passos                 | 2x2  |
|                         |                     | Tipo de preenchimento  | same |
| 2ª camada convolucional | Conv2D              | Filtros                | 32   |
|                         |                     | Tamanho do Kernel      | 3x3  |
|                         |                     | Função de ativação     | ReLU |
|                         | MaxPoo 12D          | Tamanho do kernel      | 3x3  |
|                         |                     | Passos                 | 2x2  |
|                         |                     | Tipo de preenchimento  | same |
| 2ª camada convolucional | Conv2D              | Filtros                | 32   |
|                         |                     | Tamanho do Kernel      | 2x2  |
|                         |                     | Função de ativação     | ReLU |
|                         | MaxPoo 12D          | Tamanho do Kernel      | 3x3  |
|                         |                     | Passos                 | 2x2  |
|                         |                     | Tipo de preenchimento  | same |
| 1ª camada <i>Dense</i>  | Número de neurônios | 64                     |      |
|                         | Função de ativação  | ReLU                   |      |
| Camada <i>Dropout</i>   | % de abandono       | 30                     |      |
| 2ª camada <i>Dense</i>  | Número de neurônios | 7                      |      |
|                         | Função de ativação  | <i>Softmax</i>         |      |

Fonte: Autora, 2022.

O otimizador traz um resumo dos parâmetros gerados para a rede, com a quantidade de 22.791 parâmetros. Para uma melhor análise da configuração da rede, foi realizado três vezes o treinamento da rede para ambos os casos, com 30 e 100 interações, obtendo os resultados da perda Equação (16) e acurácia Equação (17) do treinamento, validação e teste que estão listados na Tabela 5.2.

As linhas e colunas da Tabela 5.2 que estão em negrito, são os que apresentaram melhores resultados para o treinamento da rede. Para ambos os casos com 100 interações foram o que apresentaram melhores resultados. Para o caso 1, o segundo treinamento foi o que obteve melhor desempenho, com uma acurácia do treinamento de 96% e de teste 84% e conseqüentemente com menores perdas. Já para o caso 2, o primeiro treinamento foi o que teve melhores resultados, atingindo uma acurácia no treino e teste de 96% e 82%, respectivamente. O tempo de treinamento foi de 3 minutos e 23 segundos para o caso 1 e de 3 minutos e 18 segundos para o caso 2, considerado rápido para o número de interações realizadas.

Tabela 5.2: Resultado do treinamento e teste da rede neural convolucional.

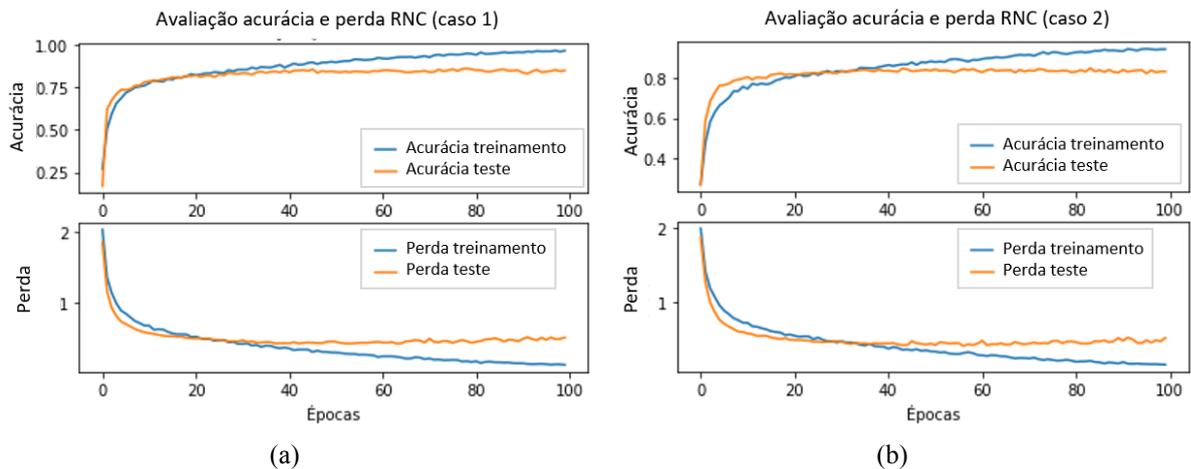
| Nº<br>treinamento | Caso     | Épocas     | Treinamento   |               | Validação     |               | Teste         |               |
|-------------------|----------|------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                   |          |            | perda         | acurácia      | perda         | acurácia      | perda         | acurácia      |
| <b>1</b>          | 1        | 30         | 0,4183        | 0,8636        | 0,5464        | 0,8136        | 0,5107        | 0,8310        |
|                   |          | 100        | 0,1452        | 0,9493        | 0,7208        | 0,8177        | 0,7282        | 0,8155        |
|                   | <b>2</b> | 30         | 0,4391        | 0,8480        | 0,5435        | 0,815         | 0,5479        | 0,8098        |
|                   |          | <b>100</b> | <b>0,1447</b> | <b>0,9551</b> | <b>0,5453</b> | <b>0,8449</b> | <b>0,6985</b> | <b>0,8163</b> |
| <b>2</b>          | <b>1</b> | 30         | 0,3873        | 0,8625        | 0,5801        | 0,8122        | 0,5146        | 0,8196        |
|                   |          | <b>100</b> | <b>0,1360</b> | <b>0,9616</b> | <b>0,7481</b> | <b>0,8150</b> | <b>0,6637</b> | <b>0,8408</b> |
|                   | 2        | 30         | 0,4449        | 0,8459        | 0,5777        | 0,7959        | 0,5544        | 0,8090        |
|                   |          | 100        | 0,1241        | 0,9629        | 0,6831        | 0,8340        | 0,8042        | 0,8049        |
| 3                 | 1        | 30         | 0,3891        | 0,8721        | 0,5767        | 0,8014        | 0,5322        | 0,8122        |
|                   |          | 100        | 0,1484        | 0,9527        | 0,5375        | 0,8503        | 0,5892        | 0,8278        |
|                   | 2        | 30         | 0,4288        | 0,8486        | 0,5627        | 0,7932        | 0,5604        | 0,7927        |
|                   |          | 100        | 0,1315        | 0,9633        | 0,7076        | 0,8245        | 0,7866        | 0,8057        |

Fonte: Autora, 2022.

Com isso, plotou-se um gráfico para observar melhor o desempenho da acurácia e da perda do modelo neural durante o treinamento e teste, que podem ser vistos pela Figura 5.7 (a) para o caso 1 e (b) para o caso 2 dos que atingiram melhores resultados. A acurácia do treinamento e do teste de ambos os casos foi aumentando conforme a redução da perda,

mostrando um bom desempenho do modelo. E percebe-se que a partir de 40 interações, a curva começou a divergir, podendo levar a uma ocorrência de *overfitting*. Por isso, pode-se recomendar um número de interações menor, entre 30 e 40, mesmo a acurácia atingindo uma menor porcentagem.

Figura 5.7: Avaliação da acurácia e perda do treinamento e teste da rede neural convolucional. (a) caso 1 – amostras amassadas em conjunto (b) caso 2 – amostras amassadas por peça.



Fonte: Autora, 2022.

Para avaliar se o modelo está classificando corretamente os dados, realizou-se um teste, onde se inseriu um conjunto de dados de áudio aleatório, não visto no treinamento e analisa-se o índice previsto era o esperado, conforme explicado no item 4.3.1. Foram realizados 10 testes para cada caso, que podem ser visualizados pela Tabela 5.3.

No caso 1, em todos os testes realizados, foram previstas corretamente todas as amostras. Para o caso 2, nos testes 2, 7, 8 e 10 foram previstas categorias diferentes das esperadas. Isso pode ter ocorrido devido às amostras terem sido amassadas individualmente, onde não se teve uma padronização nas amostras, dificultando o acerto.

Tabela 5.3: Índice esperado e índice previsto do conjunto de teste da rede neural convolucional.

| Caso | Teste | Número amostra | Índice/categoria esperado | Índice/categoria prevista |
|------|-------|----------------|---------------------------|---------------------------|
| 1    | 1     | 10             | 4 - PP                    | 4 - PP                    |
|      | 2     | 15             | 2 - PVC                   | 2 - PVC                   |
|      | 3     | 20             | 1 - PEAD                  | 1 - PEAD                  |
|      | 4     | 25             | 5 - PS                    | 5 - PS                    |
|      | 5     | 30             | 1 - PEAD                  | 1 - PEAD                  |
|      | 6     | 35             | 5 - PS                    | 5 - PS                    |
|      | 7     | 40             | 3 - PEBD                  | 3 - PEBD                  |
|      | 8     | 45             | 5 - PS                    | 5 - PS                    |

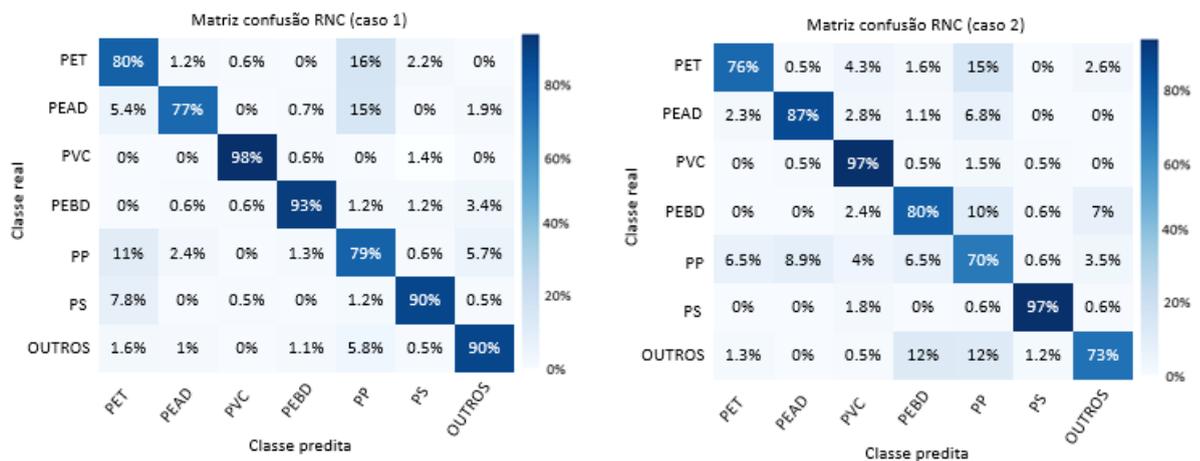
Tabela 5.3: Índice esperado e índice previsto do conjunto de teste da rede neural convolucional.

|   |    |    |            |            |
|---|----|----|------------|------------|
|   | 9  | 50 | 4 - PP     | 4 - PP     |
|   | 10 | 55 | 5 - PS     | 5 - PS     |
|   | 1  | 10 | 0 - PET    | 0 - PET    |
|   | 2  | 15 | 3 - PEBD   | 5 - PS     |
|   | 3  | 20 | 4 - PP     | 4 - PP     |
| 2 | 4  | 25 | 1 - PEAD   | 1 - PEAD   |
|   | 5  | 30 | 4 - PP     | 4 - PP     |
|   | 6  | 35 | 2 - PVC    | 2 - PVC    |
|   | 7  | 40 | 3 - PEBD   | 6 - Outros |
|   | 8  | 45 | 6 - Outros | 3 - PEBD   |
|   | 9  | 50 | 2 - PVC    | 2 - PVC    |
|   | 10 | 55 | 6 - Outros | 4 - PP     |

Fonte: Autora, 2022.

Outra métrica de avaliar a classificação do modelo neural, é pelo gráfico matriz confusão, construído a partir dos vetores do conjunto de teste, onde se tem a classe real no eixo x e a classe predita no eixo y. Os números que aparecem em cada célula indicam a porcentagem de vetores naquela situação, compatível com a intensidade da cor. A Figura 5.8 mostra o gráfico para o caso 1 e caso 2.

Figura 5.8: Matriz confusão da RNC para (a) caso 1 – amassadas em conjunto e (b) caso 2 – amassadas por peça.



(a)

(b)

Fonte: Autora, 2022.

Pelo gráfico matriz confusão da Figura 5.8, observa-se que nas células diagonais a intensidade da cor é mais forte, indicando a quantidade de previsões dos vetores de teste que estão corretos. No caso 1, a categoria PVC atingiu a maior porcentagem de 98% e PEAD foi a que teve menos previsões corretas e consequentemente menor porcentagem que foi 77%. As

outras categorias ficaram entre 80 e 90%. Para o caso 2, as categorias PVC e PS atingiram 97% das predições. A categoria PP atingiu a menor porcentagem com 70%, tendo mais dificuldade em classificar do que as outras categorias e o restante teve entre 76 e 87% de acerto. De modo geral, as categorias ficaram com uma porcentagem de acurácia consideradas aceitáveis na predição dos dados de teste.

Em um comparativo entre os dois casos para o modelo da rede neural convolucional, o caso 1 foi o que teve um desempenho melhor, obtendo a porcentagem de acurácia do teste maior e classificou corretamente todas as novas amostras que não foram vistas no treinamento. Na matriz confusão, as categorias PET, PVC, PEBD, PP e Outros tiveram uma maior porcentagem na predição. Enquanto o caso 2, obteve 4 classificações de novos dados erradas e na matriz confusão, apenas as categorias PEAD e PS tiveram uma melhor predição.

A vantagem do caso 1 em relação ao caso 2, se deve a ação de como as amostras foram amassadas. O caso 1, foi amassado em conjunto para cada categoria em um bag de pano, mantendo assim um padrão das amostras. Onde os sinais de áudio de uma mesma categoria ficaram mais parecidos, conseguindo assim classificar melhor a maioria das resinas plásticas.

### 5.3.2 Rede Neural Recorrente LSTM

Os valores dos parâmetros utilizados em cada camada da rede LSTM pode ser visualizada na Tabela 5.4.

Tabela 5.4: Valores dos parâmetros da arquitetura da rede neural recorrente do tipo LSTM.

| Arquitetura            | Parâmetros          | Valores dos parâmetros |
|------------------------|---------------------|------------------------|
| 1ª camada LSTM         | Número de neurônios | 64                     |
| 1ª camada LSTM         | Número de neurônios | 64                     |
| 1ª camada <i>Dense</i> | Número de neurônios | 64                     |
|                        | Função de ativação  | ReLU                   |
| Camada <i>Dropout</i>  | % de abandono       | 30                     |
| 2ª camada <i>Dense</i> | Número de neurônios | 7                      |

Tabela 5.4 Valores dos parâmetros da arquitetura da rede neural recorrente do tipo LSTM.

| Função de ativação | <i>Softmax</i> |
|--------------------|----------------|
|--------------------|----------------|

Fonte: Autora, 2022.

Utilizou-se os mesmos dados do modelo da rede neural convolucional, disponibilizados no link da seção 4.2.2. Os arquivos adjacentes, do caso 1 e caso 2, contêm os tipos de resinas plásticas recicláveis, os vetores MFCC's para cada segmento e as classes pertencentes para o treinamento da rede.

A quantidade total de parâmetros para o modelo LSTM ficou em 57.607. Sendo uma quantidade maior que a da RNC. Foi realizado três vezes o treinamento para esta rede também, obtendo os resultados da perda Equação (16) e acurácia Equação (17) do treinamento, validação e teste que estão listados na Tabela 5.5.

Tabela 5.5: Resultado do treinamento e teste da rede neural recorrente LSTM.

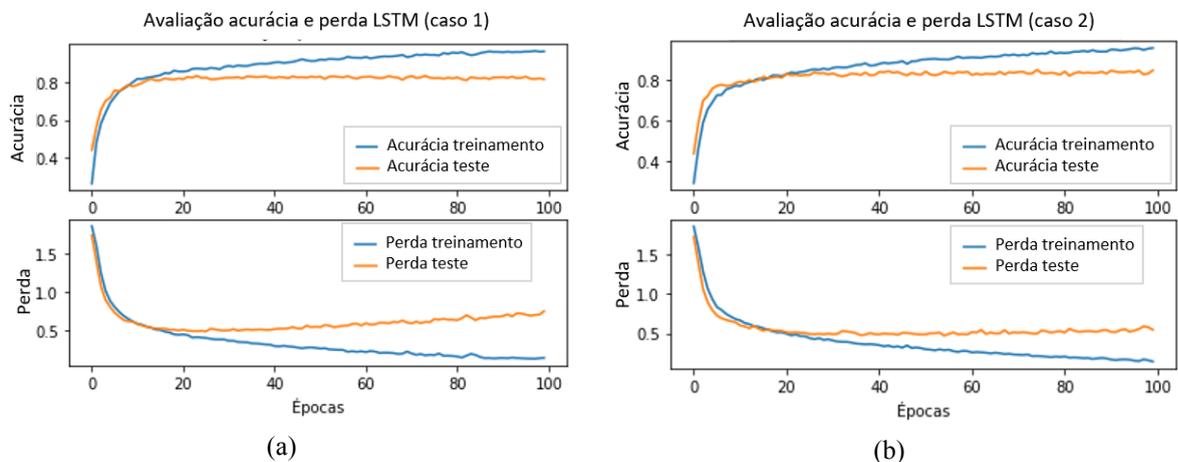
| Nº treinamento | Caso | Épocas     | Treinamento   |               | Validação     |               | Teste         |               |
|----------------|------|------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                |      |            | perda         | acurácia      | perda         | acurácia      | perda         | acurácia      |
| 1              | 1    | 30         | 0,3978        | 0,8605        | 0,4043        | 0,8517        | 0,4363        | 0,8302        |
|                |      | 100        | 0,1281        | 0,9595        | 0,4829        | 0,8422        | 0,5195        | 0,8433        |
|                | 2    | 30         | 0,4461        | 0,8452        | 0,5344        | 0,8027        | 0,4872        | 0,822         |
|                |      | 100        | 0,1857        | 0,9361        | 0,5046        | 0,8367        | 0,4490        | 0,8424        |
| 2              | 1    | 30         | 0,3745        | 0,8690        | 0,4648        | 0,8367        | 0,4499        | 0,8343        |
|                |      | <b>100</b> | <b>0,1043</b> | <b>0,9680</b> | <b>0,4961</b> | <b>0,8503</b> | <b>0,4856</b> | <b>0,8669</b> |
|                | 2    | 30         | 0,4495        | 0,8367        | 0,5392        | 0,8014        | 0,4508        | 0,8457        |
|                |      | 100        | 0,1549        | 0,9503        | 0,5862        | 0,8272        | 0,5805        | 0,8278        |
| 3              | 1    | 30         | 0,3971        | 0,8622        | 0,4212        | 0,8381        | 0,4551        | 0,8376        |
|                |      | 100        | 0,1268        | 0,9626        | 0,4866        | 0,8476        | 0,4295        | 0,8563        |
|                | 2    | 30         | 0,4425        | 0,8384        | 0,4533        | 0,8422        | 0,4907        | 0,8253        |
|                |      | <b>100</b> | <b>0,1556</b> | <b>0,9456</b> | <b>0,5172</b> | <b>0,834</b>  | <b>0,4713</b> | <b>0,8531</b> |

Fonte: Autora, 2022.

As linhas e colunas em negrito na Tabela 5.5 foram os que apresentaram melhores resultados durante o treinamento e teste da rede. Para o caso 1, o segundo treinamento foi o que obteve maior acurácia durante o treino e teste, ficando com 97% e 87%, com um tempo de treinamento de 7 minutos e 20 segundos. E para o caso 2, a maior porcentagem de acurácia foi de 95% no treinamento e 85% no teste, no terceiro e último treinamento, com um tempo de 8 minutos e 26 segundos. Ambos os casos, os melhores resultados foram atingidos com interações de 100 épocas.

Os gráficos da Figura 5.9 mostram o desempenho da acurácia e da perda do modelo neural recorrente LSTM durante o treinamento e teste que atingiram um melhor desempenho, sendo (a) para o caso 1 e (b) para o caso 2. A acurácia foi aumentando e a perda diminuindo no treinamento e teste, o que também aconteceu com a rede neural convolucional e que pode ser considerado um bom modelo tanto para o caso 1 como para o caso 2. Percebe-se que a partir de 20 interações, a curva começou a desconiliar, afastando-se a curva uma da outra, podendo levar a um overfitting. Por isso, igual analisado para a rede RNC, prefere-se uma interação menor, mesmo levando a resultados de acurácia menores também.

Figura 5.9: Avaliação da precisão e erro do treinamento e teste da rede neural recorrente LSTM. (a) caso 1 – amostras amassadas em conjunto (b) caso 2 – amostras amassadas por peça.



Fonte: Autora, 2022.

Realizou-se os 10 testes para cada caso com os dados não visto no treinamento, para avaliação do modelo, se está classificando corretamente os dados conforme os índices, que correspondem com cada resina plástica, como explicado no item 4.3.1. Os resultados estão listados na Tabela 5.6.

Tabela 5.6: Índice esperado e índice previsto da rede neural recorrente LSTM.

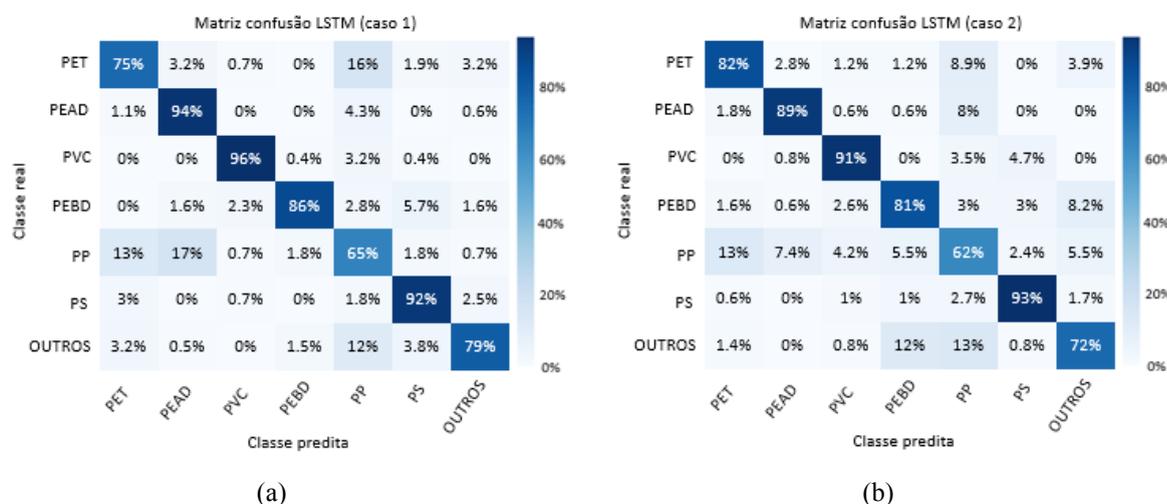
| Caso | Teste | Número amostra | Índice/categoria esperado | Índice/categoria prevista |
|------|-------|----------------|---------------------------|---------------------------|
| 1    | 1     | 10             | 2 - PVC                   | 2 - PVC                   |
|      | 2     | 15             | 4- PP                     | 4 - PP                    |
|      | 3     | 20             | 1- PEAD                   | 1 - PEAD                  |
|      | 4     | 25             | 4- PP                     | 4 - PP                    |
|      | 5     | 30             | 4- PP                     | 4 - PP                    |
|      | 6     | 35             | 3 - PEBD                  | 3 - PEBD                  |
|      | 7     | 40             | 3 - PEBD                  | 3 - PEBD                  |
|      | 8     | 45             | 3 - PEBD                  | 3 - PEBD                  |
|      | 9     | 50             | 0 - PET                   | 4 - PP                    |
|      | 10    | 55             | 6 - Outros                | 4 - PP                    |
| 2    | 1     | 10             | 5 - PS                    | 5 - PS                    |
|      | 2     | 15             | 2 - PVC                   | 2 - PVC                   |
|      | 3     | 20             | 2 - PVC                   | 2 - PVC                   |
|      | 4     | 25             | 2 - PVC                   | 4 - PP                    |
|      | 5     | 30             | 0 - PET                   | 3 - PVC                   |
|      | 6     | 35             | 4 - PP                    | 1 - PEAD                  |
|      | 7     | 40             | 4 - PP                    | 4 - PP                    |
|      | 8     | 45             | 2 - PVC                   | 2 - PVC                   |
|      | 9     | 50             | 4 - PP                    | 4- PP                     |
|      | 10    | 55             | 5 - PS                    | 5 - PS                    |

Fonte: Autora, 2022.

Para o modelo da rede neural recorrente, no caso 1, somente dois testes tiveram previsão do índice diferente do esperado, o 9 e 10. No teste 2, os testes 4, 5 e 6 tiveram previsões diferentes. Como há muitas categorias a serem classificadas e algumas amostras de áudios podem ter semelhanças, pode acontecer das previsões não serem as esperadas. Mas de uma maneira geral, conseguiu prever a maioria dos índices corretamente.

Plotou-se o gráfico matriz confusão para a rede LSTM também, pois é outra forma de avaliação do modelo e consegue-se visualizar as classes previstas corretamente e quais foram confundidas pela rede utilizando os dados de teste. A Figura 5.10 mostra o gráfico para o caso 1 e caso 2.

Figura 5.10: Matriz confusão da rede neural recorrente LSTM para (a) caso 1 – amassadas em conjunto e (b) caso 2 – amassadas por peça.



Fonte: Autora, 2022.

As células diagonais têm intensidade da cor mais forte, indicando a quantidade de previsões dos vetores de teste que estão corretos. No caso 1, a categoria PVC atingiu a maior porcentagem de 96% e PP foi a que obteve menor porcentagem ficando com 65% e desse modo foi a categoria de menos previsões corretas. As outras categorias ficaram entre 79 e 94%. No caso 2, a categoria PS atingiu 93% das previsões, a maior porcentagem para esse grupo de amostras e a categoria PP também atingiu a menor porcentagem com 62%, tendo mais dificuldade em classificar do que as outras categorias e o restante teve entre 72 e 91% de acerto. Genericamente, as categorias atingiram uma porcentagem de acurácia consideradas aceitáveis para predição dos dados de teste.

Fazendo o mesmo comparativo entre os dois casos para a rede neural recorrente LSTM. O caso 1 também obteve melhor desempenho, ficando com a porcentagem de acurácia do teste maior, classificando novas amostras com menos erros e na matriz confusão teve 5 categorias de resinas com porcentagem de previsões maiores, sendo elas, PEAD, PVC, PEBD, PP e Outros. E no caso 2, apenas as categorias PET e PS tiveram melhor desempenho na porcentagem da matriz confusão.

Conferindo as duas redes neurais, ambas tiveram 5 categorias que se destacaram na matriz confusão. A diferença foi que a rede RNC classificou melhor a categoria PET e a rede LSTM a categoria PEAD. Em relação à acurácia do teste, a rede neural recorrente LSTM teve vantagem, pela veracidade que essa rede trabalha melhor com dados que são de séries temporais.

### 5.3.3 Validação

Com o banco de dados das amostras amassadas lentamente para cada categoria de resina plástica, conforme explicado no item 4.3.3. , realizou-se um teste inserindo esses dados de áudio no modelo da rede neural convolucional e da rede neural recorrente LSTM que tiveram os melhores desempenhos, foram o treinamento 2 do caso 1 para ambas as redes, que atingiram em seus testes originais uma acurácia de 84% para RNC e 87% para LSTM.

Essas amostras, amassadas lentamente, foram colocadas nos modelos das redes neurais somente no conjunto de teste, para uma provável condição que ocorre no dia-a-dia das cooperativas de reciclagem. Assim, pode-se analisar o desempenho das redes neurais artificiais com dados de séries temporais diferentes das que foram treinadas. O resultado da acurácia e da perda para esses modelos, podem ser vistas pela Tabela 5.8.

Tabela 5.7: Resultado do treinamento e teste das redes neurais para o caso 3.

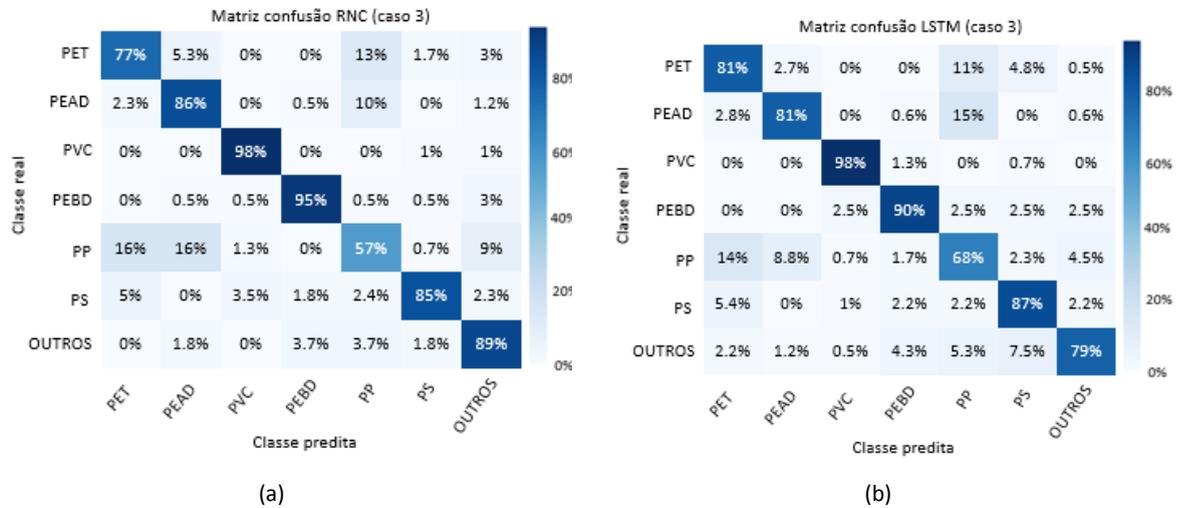
| Modelo | Treinamento |          | Validação |          | Teste |          |
|--------|-------------|----------|-----------|----------|-------|----------|
|        | perda       | acurácia | perda     | acurácia | perda | acurácia |
| RNC    | 0,1360      | 0,9616   | 0,7481    | 0,8150   | 1,213 | 0,7286   |
| LSTM   | 0,1043      | 0,9680   | 0,4961    | 0,8503   | 1,401 | 0,7020   |

Fonte: Autora, 2022.

A acurácia do teste para a rede RNC ficou em 73% e para a rede LSTM em 70%, tendo um declínio no desempenho de ambas as redes, além das perdas também terem aumentado em comparação com os resultados dos testes anteriores. Analisou-se também a matriz confusão, que pode ser vista pela Figura 5.11, sendo (a) para RNC e (b) para LSTM.

As categorias PVC e PEBD foram a que alcançaram um melhor desempenho, conseqüentemente menos confusões ao classificar e na categoria PP teve dificuldades ao identificar na matriz confusão para ambas as redes. Ao comparar os resultados dos testes realizados anteriormente, a porcentagem de acerto nas predições caiu para o caso 3. Esse decaimento, ocorre porque as novas amostras não foram treinadas pela rede e somente inseridas no conjunto de teste. Uma condição muito provável de ocorrer no cotidiano das cooperativas, pois pode haver diferenças nas forças dos funcionários que irão amassar as amostras, ocasionando uma situação desfavorável e não vista pela rede.

Figura 5.11: Matriz confusão do conjunto de teste do caso 3 (amassadas de forma mais lenta) para (a) RNC e (b) LSTM.



Fonte: Autora, 2022.

Com essas circunstâncias, uma rede recorrente LSTM tem dificuldades ao prever situações em que não seguem a sequência de tempo visto anteriormente e o modelo da RNC pode ser mais favorável para prever amostras que se encaixam a esses tipos de imprevistos.

### 5.3.4 Resumo dos resultados

Com todos os resultados obtidos na pesquisa, pode-se realizar uma comparação dos melhores desempenhos entre o caso 1 (amostras em conjuntos submetidas no bag de pano e morsa), caso 2 (amostras inteiras) e caso 3 (amostras amassadas de forma mais lenta no bag de pano e morsa). E também uma analogia entre os modelos da RNC e LSTM. A Tabela 5.9 mostra o resumo do treinamento e teste dos modelos neurais nas situações de esmagamento.

Tabela 5.8: Resumo dos resultados dos casos do treinamento e teste das redes neurais.

| Modelos neurais      | RNC    |        |        | LSTM   |        |        |
|----------------------|--------|--------|--------|--------|--------|--------|
|                      | Caso 1 | Caso 2 | Caso 3 | Caso 1 | Caso 2 | Caso 3 |
| Acurácia treinamento | 0,9616 | 0,9551 | 0,9616 | 0,9680 | 0,9456 | 0,9680 |
| Acurácia teste       | 0,8408 | 0,8163 | 0,7286 | 0,8669 | 0,8531 | 0,7020 |

Fonte: Autora, 2022.

A Acurácia do treinamento dos três casos ficaram boas para os dois modelos neurais. A rede LSTM foi a que teve melhor desempenho, conseguiu atingir 97% de acurácia no caso

1, por ser uma rede que trabalha bem com dados de séries temporais. Assim, os dados do caso 3, foram inseridos no conjunto de teste deste modelo, por isso a mesma porcentagem do treinamento para o caso 1 e caso 3.

Em relação ao teste, o caso 3, teve uma queda considerável na acurácia. O interessante é que se observa que nessas condições, onde as amostras estão com série temporal diferentes das do treinamento, a rede LSTM teve mais dificuldades, enquanto a rede RNC teve um melhor resultado, podendo ser preferível nessas circunstâncias. Por mais que o valor da acurácia tenha ficado menor no teste, ainda é considerado boa ao compararmos os 40% da classificação manual, realizadas hoje nas cooperativas.

Fazendo uma análise da matriz confusão pela Tabela 5.10, onde tem as resinas plásticas que obtiveram a maior e menor porcentagem de acerto em relação aos dados de teste do caso 1, caso 2 e caso 3 para o modelo neural RNC e LSTM.

Tabela 5.9: Maiores e menores porcentagens de acerto da matriz confusão.

| Rede              | RNC         |                 |            | LSTM       |           |            |
|-------------------|-------------|-----------------|------------|------------|-----------|------------|
| Caso              | 1           | 2               | 3          | 1          | 2         | 3          |
| Maior porcentagem | PVC<br>98%  | PVC e PS<br>97% | PVC<br>98% | PVC<br>96% | PS<br>93% | PVC<br>98% |
| Menor porcentagem | PEAD<br>77% | PP<br>70%       | PP<br>57%  | PP<br>65%  | PP<br>62% | PP<br>68%  |

Fonte: Autora, 2022.

A resina PVC foi a que se destacou, obtendo a maior porcentagem de amassamento e nos modelos neurais. Destaque-se que para o caso 2, onde as amostras foram amassadas inteiras, sem o auxílio do bag de pano e da morsa, a categoria de resina PS teve maior porcentagem de acerto nas redes neurais, empatando com PVC no modelo RNC. Das categorias que tiveram a menor porcentagem de acerto na matriz confusão foi PP. Somente para o caso 1, que as amostras foram amassadas em conjunto dentro do bag de pano e com auxílio da morsa, do modelo RNC, a categoria PEAD ficou com menor porcentagem.

Com isso, têm-se uma semelhança da categoria com maior e menor porcentagem de acerto na matriz confusão nos casos de amassamento realizados e nos modelos neurais, dando uma maior confiabilidade ao programa de classificação de resinas plásticas para reciclagem.

## Capítulo 6 - CONCLUSÕES

Com os resultados da pesquisa e testes realizados, ambas as redes utilizadas (RNC e LSTM) conseguiram classificar as resinas plásticas de reciclagem para o caso 1 - amostras amassadas em conjunto dentro do bag de pano e com auxílio da morsa e caso 2 – amostras amassadas por peça sem o bag de pano e sem morsa, obtendo boas porcentagens de acerto nas predições da matriz confusão, principalmente na categoria PVC classificando, respectivamente, de 96% a 98% das amostras corretamente. Como há bastante categorias para serem classificadas e algumas têm características semelhantes, fez com que os modelos neurais tivessem mais dificuldades em algumas resinas. Esse problema pode ser resolvido utilizando apenas as resinas mais consumidas, como PET, PEAD, PP e PEBD, facilitando a classificação.

Os melhores modelos das redes foram alcançados utilizando uma interação de 100 épocas, porém o gráfico de desempenho mostra que a partir de 40 interações as curvas começam divergir, podendo levar a uma ocorrência de *overfitting*, por isso, é preferível a utilização de um número menor de interações, entre 30 e 40 épocas.

O caso que as amostras foram amassadas em conjunto dentro do bag de pano e com a morsa, foi o que atingiu melhores valores para ambas as redes, por conta que teve padronização ao amassar essas amostras, fazendo com que os sinais de áudio de cada categoria de resina ficassem mais parecidos, conseguindo um melhor desempenho na maioria das resinas plásticas.

A rede LSTM, teve um treinamento mais eficiente do que a RNC, conseguindo uma acurácia do teste de 87% e conseqüentemente uma menor perda. Essa vantagem, se dá por conta da rede recorrente ser para dados com sequência temporais, dados de áudio.

Ao colocar dados no conjunto de teste, amassados de uma forma mais lenta, o desempenho da rede LSTM caiu, pois essas novas amostras não têm a mesma sequência que os dados treinados pela rede anteriormente.

E como os trabalhadores nas cooperativas provavelmente não irão amassar os bags com os pedaços de plástico na mesma velocidade e não haverá uma pessoa apta e disponível todo o tempo para poder retrainar a rede neural com novos dados, o caso que as amostras foram amassadas lentamente, foi uma condição desfavorável e muito provável de ocorrer no dia-a-dia da cooperativa. Assim, mesmo a rede LSTM ser considerada adequada para a situação ideal, quando se coloca uma condição real, a RNC pode ser uma opção melhor por não precisar da sequência do áudio para realizar a classificação. Isto é, mesmo que o

amassamento das amostras difira do realizado na coleta para a base de dados do treinamento, a RNC ainda consegue acertar numa taxa aceitável, enquanto a LSTM tende a errar mais nessas situações por depender da sequência.

Em geral, o desempenho da técnica de rede neural artificial foi considerado boa, os resultados obtidos ficaram superiores aos 40% do método utilizado hoje pelas cooperativas, forma manual. Assim, pode-se levar essa informação e tecnologia para as cooperativas de reciclagem e quem se interessar, onde se coloca como trabalho futuro o desenvolvimento de um aplicativo, que tenha como base o algoritmo desenvolvido nesse trabalho.

Portanto, com essa tecnologia, os cooperados poderão ter acesso facilitado, sem a necessidade de nenhum equipamento dispendioso e específico, apenas a utilização de um smartphone, tornando o método popular. Os funcionários das cooperativas, irão utilizar o programa após terem feito uma pré-seleção das resinas, para que o processo fique mais confiável e haja menos desperdício. As amostras podem ser recicladas com qualidade e não serem jogadas em aterros. E como o modelo já está treinado, o processo de coleta, amassamento e classificação das amostras se torna rápido, não tomando muito tempo. Com a disponibilização dos dados, torna-se possível, também, um trabalho futuro com novos testes, utilizando métodos de identificação que difere do estudado nesta pesquisa, como o de redução de dimensionalidade.

## REFERÊNCIAS

Abiplastic. **Índice de plástico reciclado pós-consumo cresceu em 2019, segundo estudo encomendado pelo picplast.** Disponível em:

<http://www.abiplast.org.br/noticias/indice-de-plastico-reciclado-pos-consumo-cresceu-em-2019-segundo-estudo-encomendado-pelo-picplast/>. Acesso em: 8 de mar. 2022.

Abiplastic. **Índices de reciclagem mecânica de plástico pós-consumo no Brasil. 2021.**

Disponível em: [http://www.abiplast.org.br/publicacoes/pesquisa\\_reciclagem\\_picplast/](http://www.abiplast.org.br/publicacoes/pesquisa_reciclagem_picplast/). Acesso em: 4 jan 2023.

AHMAD, S.R. **A New Technology for Automatic Identification and Sorting of Plastics for Recycling.** Environmental Technology. v. 25, n. 10, p. 1143-1149, 2010. DOI

[10.1080/09593332508618380](https://doi.org/10.1080/09593332508618380). Disponível em: <https://www.tandfonline.com/loi/tent20>.

Acesso em: 20 maio 2021.

ALBUQUERQUE, Marcio Portes; ALBUQUERQUE, Marcelo Portes. **Processamento Digital de Sinais.** 2002. Disponível em:

<https://docplayer.com.br/114410388-Processamento-digital-de-sinais.html>. Acesso em: 2 maio 2021.

AMARAL, Luis Fernando. **O uso de podcast como ferramenta na educação:** criação do podcast conversa de corredor na escola Senac Franca. Orientador(a): Juliana Mara Flores Bicalho. 2019. 48 p. TCC (Especialização em Mídias na Educação) – Especialização em Mídias na Educação, Universidade Federal de São João Del-Rei, 2019. Disponível em:

<http://dspace.nead.ufsj.edu.br/trabalhospublicos/bitstream/handle/123456789/446/O%20USO%20DE%20PODCAST%20COMO%20FERRAMENTA%20NA%20EDUCA%c3%87%c3%83O%20CRIA%c3%87%c3%83O%20DO%20PODCAST%20CONVERSA%20DE%20CORREDOR%20NA%20ESCOLA%20SENAC%20FRANCA%20-%20Luis%20Fernando%20Amaral.pdf?sequence=1&isAllowed=y>. Acesso em: 6 jun. 2022.

ARAÚJO, Aline Moura. **Classificação e detecção de pessoas em ambientes não controlados utilizando redes neurais convolucionais.** Orientador: Leonardo Vidal Batista. 2019. 47 p. Monografia (Bacharel em Ciências da Computação) - Ciência da Computação, Universidade Federal da Paraíba, 2019. Disponível em:

[https://repositorio.ufpb.br/jspui/handle/123456789/16098?locale=pt\\_BR](https://repositorio.ufpb.br/jspui/handle/123456789/16098?locale=pt_BR). Acesso em: 20 abr. 2022.

BENBA, Achraf; JILBAB, Abdelilah; HAMMOUCH, Ahmed. **Analysis of multiple types of voice recordings in cepstral domain using MFCC for discriminating between patients with Parkinson's disease and healthy people.** Int J Speech Technol, p. 449–456, 2016. DOI

10.1007/s10772-016-9338-4. Disponível em:

<https://link.springer.com/article/10.1007/s10772-016-9338-4>. Acesso em: 15 abr. 2021.

BENTES, B. A. H. F. **Aplicação de Redes Neurais Artificiais para estimativa de volume de árvores em pé em uma floresta ombrófila densa na Amazônia oriental**. Orientador: Dr. Phil. José Natalino Macedo Silva. 2018. 60 p. Dissertação (Mestrado em Ciências Florestais) - Ciências Florestais, Universidade Federal Rural da Amazônia, 2018. Disponível em:

<http://repositorio.ufra.edu.br/jspui/bitstream/123456789/435/1/Aplica%C3%A7%C3%A3o%20de%20Redes%20Neurais%20Artificiais.....pdf>. Acesso em: 20 abr. 2022.

BORGES, G. P. *et al.* **Methods and systems to identify and manage recyclable materials**. US 9,108,797 B1. Depósito: 9 set. 2014. Concessão: 18 ago. 2015. Disponível em:

<https://www.freepatentsonline.com/y2016/0069840.html>. Acesso em: 25 mar. 2021.

BRAGA, Daniel L. S. **Pesquisas e inovações nacionais em Engenharia, Ciências Agrárias, Exatas e da Terra**. Florianópolis, SC. Instituto Scientia. 2022. Disponível em:

[https://web.archive.org/web/20220517025052id\\_/https://institutoscientia.com/wp-content/uploads/2022/04/Livro-Engenharia-Agrarias-Exatas.pdf#page=232](https://web.archive.org/web/20220517025052id_/https://institutoscientia.com/wp-content/uploads/2022/04/Livro-Engenharia-Agrarias-Exatas.pdf#page=232). Acesso em: 13 jul. 2022.

CANCIAM, César Augusto. **Identificação de resinas termoplásticas pelo teste de chama**. Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia, p. 99-106, 2020. DOI 10.22533/at.ed.45121190110. Disponível em:

[https://www.researchgate.net/publication/348634481\\_Identificacao\\_de\\_Resinas\\_Termoplasticas\\_pelo\\_Testes\\_de\\_Chama](https://www.researchgate.net/publication/348634481_Identificacao_de_Resinas_Termoplasticas_pelo_Testes_de_Chama). Acesso em: 05 jul. 2022.

CAVALEIRO, João Tiago A. S. **Tecnologias de Comunicação Multimédia**. 2015. 30 p. Dissertação (Mestrado) – Multimédia, Universidade do Porto, 2015. Disponível em:

[https://web.fe.up.pt/~cavaleiro/docs/TCM\\_MP3\\_2014-15\\_trabalho\\_Tiago\\_Cavaleiro.pdf](https://web.fe.up.pt/~cavaleiro/docs/TCM_MP3_2014-15_trabalho_Tiago_Cavaleiro.pdf). Acesso em: 6 jun. 2022.

CLAPPIS, Alan Melo. **Uma introdução as redes neurais convolucionais utilizando o Keras**. Medium. 12 jul. 2019. Disponível em:

<https://medium.com/data-hackers/uma-introdu%C3%A7%C3%A3o-as-redes-neurais-convolucionais-utilizando-o-keras-41ee8dce033e>. Acesso em: 10 ago. 2021.

COLAB. **Conheça o Colab**. Disponível em:

[https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt-BR#scrollTo=Nma\\_JWh-W-IF](https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt-BR#scrollTo=Nma_JWh-W-IF). Acesso em: 15 set. 2021.

COLTRO, Leda; DUARTE, Leda C. **Reciclagem de Embalagens Plásticas Flexíveis: Contribuição da Identificação Correta**. Polímeros. v. 23, n. 1, p. 128-134, 2013. DOI

<http://dx.doi.org/10.1590/S0104-14282013005000008>. Disponível em:

<https://revistapolimeros.org.br/journal/polimeros/article/doi/10.1590/S0104-1428201300500008>. Acesso em: 21 maio 2021.

COLTRO, Leda; GASPARINO, Bruno F.; QUEIROZ, Guilherme de C. **Reciclagem de Materiais Plásticos: A Importância da Identificação Correta**. *Polímero: Ciências e tecnologia*. v. 18, n. 2, p. 119-125, 2008. DOI

<http://dx.doi.org/10.1590/S0104-14282008000200008>. Disponível em:

<https://revistapolimeros.org.br/journal/polimeros/article/doi/10.1590/S0104-1428200800020008>. Acesso em: 21 maio 2021.

COSTA, Reynerth Pereira; SOUZA, José Luiz da Silva; MIRANDA, Francisco Otávio.

**Transformada Wavelet de Fourier aplicada a análise de sinais ruidosos**. *Brazilian Journal of Development*, v. 5, ed. 6, p. 6509-6527, 2019. Disponível em:

<https://www.brazilianjournals.com/index.php/BRJD/article/view/1881>. Acesso em: 10 maio 2021.

DEEP LEARNING (for Audio) with Python: Course Overview. Valerio Valardo, 2020. 1 vídeo (8 min). Disponível em:

<https://www.youtube.com/watch?v=fMqL5vckiU0&list=PL-wATfeyAMNrtbkCNsLcpoAyB BRJZVInf&index=1>. Acesso em: 16 de jun. 2021.

DHARMANA, M. M. et al. **Plastic segregation machine using audio signal analysis**.

Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology, p. 982-985, 2019. DOI <https://doi.org/10.1109/ICECA.2019.8821842>.

Disponível em: <https://ieeexplore.ieee.org/document/8821842>. Acesso em: 25 mar. 2021.

FERREIRA, D. H. L.; KINTSCHNER, F. E.; SUGAHARA, C. R. **Sistema de controle de produção e de estoque aplicado às cooperativas de reciclagem**. *Revista Tecnologia e Sociedade*, v. 18, n. 52, p. 151-165, 2022. Disponível em: <https://periodicos.utfpr.edu.br/rts>.

Acesso em: 05 jul. 2022.

FLECK, Leandro *et al.* **Redes Neurais Artificiais: Princípios Básicos**. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, ed. 13, p. 47-57, 2016. Disponível em:

[https://d1wqtxts1xzle7.cloudfront.net/57830283/4330-15577-1-PB-with-cover-page-v2.pdf?Expires=1654193390&Signature=ZoohKoZaiteurn5Bmkqtr6rxbA-oNEXnX~mlhPDyKYRspt74KvCsxSTUsHMgq21HmaiKR5SWcZEsj-mi4CrTAtEk-V67wbZFK17NgWyk5KGdRILKBmQEzcpjiMfiGd1gP6bgP~GDLK-FA6dfdbY~cSIc9JV2g2t6kTQhQJtNcoXMhb61NE2wHh9zHF6YHH6nsLouguAliPDayfitMRXWvj7DCG-TxwEnCgKcPI9nOnqQch2llUVS70y~ku8FPPrXwca5bgxuvpXeO0lt9xLnNAbrUYwc1M3VlbnQB40rszKke~Uqq-1QwOpkHJ6bDQx1yFqo4EtUSwW~M71G2fP7xA\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/57830283/4330-15577-1-PB-with-cover-page-v2.pdf?Expires=1654193390&Signature=ZoohKoZaiteurn5Bmkqtr6rxbA-oNEXnX~mlhPDyKYRspt74KvCsxSTUsHMgq21HmaiKR5SWcZEsj-mi4CrTAtEk-V67wbZFK17NgWyk5KGdRILKBmQEzcpjiMfiGd1gP6bgP~GDLK-FA6dfdbY~cSIc9JV2g2t6kTQhQJtNcoXMhb61NE2wHh9zHF6YHH6nsLouguAliPDayfitMRXWvj7DCG-TxwEnCgKcPI9nOnqQch2llUVS70y~ku8FPPrXwca5bgxuvpXeO0lt9xLnNAbrUYwc1M3VlbnQB40rszKke~Uqq-1QwOpkHJ6bDQx1yFqo4EtUSwW~M71G2fP7xA_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA). Acesso em:

2 maio 2021.

FORLIN, Flávio J.; FARIA, José de Assis F. **Considerações Sobre a Reciclagem de Embalagens Plásticas**. Polímeros: Ciência e Tecnologia. v. 12, n. 1, p. 1-10, 2002. DOI <https://doi.org/10.1590/S0104-14282002000100006>. Disponível em: <https://www.scielo.br/j/po/a/YNNvN9nLDV8rS5ffJp9rF4Q/abstract/?lang=pt>. Acesso em: 25 fev. 2022.

FRAGOSO, Rodrigo Andrade. **Avaliação do uso de modelos de aprendizado de máquina estatístico para previsão de diagnóstico para doenças infecciosas**. Orientador: Prof. Dr. Reinaldo A. C. Bianchi. 2022. 89 f. Dissertação (Mestrado) – Engenharia Elétrica, Centro Universitário FEI, São Bernardo do Campo, SP, 2022. Disponível em: <https://repositorio.fei.edu.br/bitstream/FEI/4489/1/fulltext.pdf>. Acesso em: 13 jul. 2022.

Gaia Social. **Programa ViraSer**. Disponível em: <https://giasocial.org.br/programa-viraser/>. Acesso em: 28 mai. 2021.

GEYER, Roland; JAMBECK, Jenna R.; LAW, Kara Lavender. **Production, use, and fate of all plastics ever made**. Science Advances, 2017. DOI <https://doi.org/10.1126/sciadv.1700782>. Disponível em: <https://www.science.org/doi/10.1126/sciadv.1700782>. Acesso em: 23 abr. 2021.

GHOLAMY, Afshin; KREINOVICH, Vladik; KOSHELEVA, Olga. **Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation**. Department Technical Reports (CS), 1209. Disponível em: [https://scholarworks.utep.edu/cs\\_techrep/1209/](https://scholarworks.utep.edu/cs_techrep/1209/). Acesso em: 4 de ago. 2022.

GOMES, Leonardo Manhães. **Técnicas de Extração de Características da Fala**. 2018. Disponível em: [http://www.midiacom.uff.br/debora/images/disciplinas/2018-2/smm/trabalhos/Artigo\\_MFCC.pdf](http://www.midiacom.uff.br/debora/images/disciplinas/2018-2/smm/trabalhos/Artigo_MFCC.pdf). Acesso em: 15 abr. 2021.

GONÇALVES, Ana Paula. **Desafios para redução das embalagens plásticas de alimentos pós-consumo nos rejeitos da coleta seletiva: um estudo de caso em uma cooperativa no município de Assis – SP**. Orientador: Prof. Dr. Valdir Schalch. 2022. 174 f. Dissertação (Mestrado) – Curso de Ciências, Universidade de São Paulo, São Carlos, 2022. Disponível em: <https://www.teses.usp.br/teses/disponiveis/18/18138/tde-20062022-104036/en.php>. Acesso em: 05 jul. 2022.

HAHNLOSER *et al.* **Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit**. Nature, v. 405, p. 947-951, 2000. DOI 10.1038/35016072. Disponível em: <https://www.nature.com/articles/35016072>. Acesso em: 20 dez. 2022.

HIWATASHI, Erica. **O processo de reciclagem dos resíduos sólidos**. Orientador: Prof. Dr. Luís Felipe M. Nascimento. 1999. 150 f. Dissertação (Mestrado em Administração) – Programa de pós-graduação em Administração, Núcleo de gestão da inovação tecnológica, 1999. Disponível em: <https://lume.ufrgs.br/handle/10183/29128>. Acesso em: 23 fev. 2022.

HOCHREITER, Sepp; SCHMIDHUBER, Jurgen. **Long Short-Term Memory**. Neural Computation, v. 9, ed. 8, p. 1735-1780, 1997. Disponível em: <https://www.bioinf.jku.at/publications/older/2604.pdf>. Acesso em: 21 dez. 2022.

JSON - **encoder and decoder**. Disponível em: <https://docs.python.org/3/library/json.html>. Acesso em: 18 mai. 2022.

JULIANI, Jeferson da Silva. **Gerando letras musicais utilizando uma rede neural recorrente LSTM - Long Short-Term Memory**. Orientador: Prof. Dr. Alexandre Matos Arruda. 2019. 47 p. TCC (Bacharel em Engenharia de Software) - Engenharia de Software, Universidade Federal do Ceará, 2019. Disponível em: [https://repositorio.ufc.br/bitstream/riufc/49519/1/2019\\_tcc\\_jsjuliani.pdf](https://repositorio.ufc.br/bitstream/riufc/49519/1/2019_tcc_jsjuliani.pdf). Acesso em: 25 abr. 2022.

JUNIOR, José R. F. **Redes Neurais Recorrentes — LSTM**. 2019. Disponível em: <https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6>. Acesso em: 27 abr. 2022.

KINGMA, Diederik P.; BA, Jimmy Lei. **Adam: A Method for Stochastic Optimization**. San Diego, v. 9, 2014. DOI 10.48550/arXiv.1412.6980. Disponível em: <https://arxiv.org/abs/1412.6980>. Acesso em: 20 mai. 2022.

KIPPER, Liane Mählmann; MÄHLMANN, Cláudia Mendes; RODRÍGUEZ, Adriane Lawisch. **Ações estratégicas sistêmicas visando à integração da cadeia produtiva e de reciclagem de plásticos**. v. IX, ed. IV, Revista Produção, 2009. DOI <http://dx.doi.org/10.14488/1676-1901.v9i4.348>. Disponível em: [https://www.researchgate.net/publication/307755874\\_bAcoes\\_estrategicas\\_sistemicas\\_visando\\_a\\_integracao\\_da\\_cadeia\\_produtiva\\_e\\_de\\_reciclagem\\_de\\_plasticosb](https://www.researchgate.net/publication/307755874_bAcoes_estrategicas_sistemicas_visando_a_integracao_da_cadeia_produtiva_e_de_reciclagem_de_plasticosb). Acesso em: 21 abr. 2021.

LeCun, Yann *et al.* **Gradient Based Learning Applied to Document Recognition**. Proc. of the IEEE, 1998. Disponível em: [http://vision.stanford.edu/cs598\\_spring07/papers/Lecun98.pdf](http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf). Acesso em: 20 dez. 2022.

LIMA, Renata Rodrigues. **Controle da flexão e extensão do cotovelo via FES em malha-fechada utilizando controladores RST**. Orientador: Prof. Dr. Fabricio Gonzalez Nogueira. 2022. 107 p. TCC (Bacharel em Engenharia Elétrica) - Engenharia Elétrica, Centro

de Tecnologia da Universidade Federal, 2022. Disponível em:  
<https://repositorio.ufc.br/handle/riufc/65857>. Acesso em: 2 jun. 2022.

LOUGHRAN, Róisín *et al.* **The Use of Mel-frequency Cepstral Coefficients in Musical Instrument Identification**. International Computer Music Conference, 2008. DOI 10.1.1.474.8056. Disponível em:  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.474.8056>. Acesso em: 14 abr. 2022.

LU, Gang *et al.* **One-dimensional convolutional neural networks for acoustic waste sorting**. Journal of Cleaner Production, v. 271, 2020. DOI  
<https://doi.org/10.1016/j.jclepro.2020.122393>. Disponível em:  
<https://www.sciencedirect.com/science/article/abs/pii/S0959652620324409>. Acesso em: 25 maio 2022.

MATUCK, Gustavo Ravanhani. **Processamento de sinais de voz na identificação de padrões comportamentais por redes neurais artificiais**. Orientador: Dr. José Demisio Simões da Silva. 2005. 62 p. Iniciação Científica (Graduação Departamento de Aeronáutica) - Instituto Nacional de pesquisas espaciais, 2005. Disponível em:  
<http://mtc-m21c.sid.inpe.br/col/sid.inpe.br/mtc-m21c/2020/08.03.18.54/doc/processamento%20sinais.pdf?metadatarepository=sid.inpe.br/mtc-m21c/2020/08.03.18.54.18&mirror=urllib.net/www/2017/11.22.19.04.03>. Acesso em: 10 mar. 2022.

MCCULLOCH, Warren S.; PITTS, Walter. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics, v.5, 1943.

MCFFEE, Brian *et al.* **librosa: Audio and Music Signal Analysis in Python**. Proc. of the 14<sup>th</sup> Python in Science Conf, p. 18-24, 2015. Disponível em:  
[https://conference.scipy.org/proceedings/scipy2015/pdfs/brian\\_mcftee.pdf](https://conference.scipy.org/proceedings/scipy2015/pdfs/brian_mcftee.pdf). Acesso em: 18 mai. 2022.

OLIVEIRA, Emerson Vilar. **Análise de desempenho de método baseado em Rede LSTM para classificação de falhas em um processo de controle de nível**. Orientador: Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira. 2020. 78 p. Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Universidade Federal do Rio Grande do Norte, 2020. Disponível em:  
<https://repositorio.ufrn.br/jspui/handle/123456789/30254#:~:text=Emerson%20Vilar%20de.-,An%C3%A1lise%20de%20desempenho%20de%20m%C3%A9todo%20baseado%20em%20rede%20LSTM%20para.do%20Norte%2C%20Natal%2C%202020>. Acesso em: 25 abr. 2022.

PENHA, Deyvison de Paiva. **Rede Neural Convolutacional aplicada à identificação de equipamentos residenciais para sistemas de monitoramento não-intrusivo de carga**. Orientador: Dra. Adriana Rosa Garcez Castro. 2018. 71 p. Dissertação (Mestrado em

Engenharia Elétrica) - Engenharia Elétrica, Universidade Federal do Pará, 2018. Disponível em:

[https://ppgee.propesp.ufpa.br/ARQUIVOS/dissertacoes/DM%2017\\_2018%20Deyvison%20de%20Paiva%20Penha.pdf](https://ppgee.propesp.ufpa.br/ARQUIVOS/dissertacoes/DM%2017_2018%20Deyvison%20de%20Paiva%20Penha.pdf). Acesso em: 23 jun. 2021.

PEREIRA, Mateus dos Santos. **Identificação de emoções em sinais de voz com base no operador de energia de Teager Aprimorado**. Orientador: Prof. Dr. Rodrigo Capobanco Guido. 2021. 48 p. Monografia (Graduação em Ciências da Computação) - Ciências da Computação, Universidade Estadual Paulista, 2021. Disponível em:

<https://repositorio.unesp.br/handle/11449/216062>. Acesso em: 2 jun. 2022.

PEREIRA, Matheus de Mattos. **Aprendizado profundo: Redes LSTM**. Orientador: Prof. Ms Anderson Bessa da Costa. 2017. 40 p. TCC (Bacharel em Sistemas de Informação) - Sistemas de Informação, Universidade Federal da Grande Dourados, 2017. Disponível em:

<https://repositorio.ufgd.edu.br/jspui/handle/prefix/2887>. Acesso em: 27 abr. 2022.

PEREIRA, R. C. M.; DANTAS, D. **Polímeros: Organização, Usabilidade e Produtos**. 2010. Disponível em:

[https://www.researchgate.net/publication/360936939\\_Polimeros\\_Organizacao\\_Usabilidade\\_e\\_Produtos?enrichId=rgreq-19c6fee8a24c2a8bf4dfd48471cdf6fe-XXX&enrichSource=Y292ZXJQYWdlOzM2MDkzNjkzOTtBUzoxMTYxMjc1MDczNjYyOTc2QDE2NTM4ODA5NjEwMjI%3D&el=1\\_x\\_2&\\_esc=publicationCoverPdf](https://www.researchgate.net/publication/360936939_Polimeros_Organizacao_Usabilidade_e_Produtos?enrichId=rgreq-19c6fee8a24c2a8bf4dfd48471cdf6fe-XXX&enrichSource=Y292ZXJQYWdlOzM2MDkzNjkzOTtBUzoxMTYxMjc1MDczNjYyOTc2QDE2NTM4ODA5NjEwMjI%3D&el=1_x_2&_esc=publicationCoverPdf). Acesso em: 05 jul. 2022.

RODRIGUES, Diego Alves. **Deep Learning e Redes Neurais Convolucionais:**

Reconhecimento automático de caracteres em placas de licenciamento automotivo.

Orientador: Prof. Dr. Leonardo Vidal Batista. 2018. Monografia (Bacharel em Ciência da Computação) - Ciência da Computação, Universidade Federal da Paraíba, 2018. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/15606/1/DAR20052019.pdf>.

Acesso em: 25 abr. 2022.

ROH, Seok-Beom; OH, Sung-Kwun. **Identification of black plastics based on Fuzzy RBF neural networks:** focused on data preprocessing techniques through Fourier transform infrared radiation. *Transcriptions on industrial informatics*, v. 14, ed. 5, p. 1802-1813, 2018.

DOI <https://doi.org/10.1109/TII.2017.2771254>. Disponível em:

<https://ieeexplore.ieee.org/document/8100918>. Acesso em: 23 mar. 2021.

SÁNCHEZ, J. C. *et al.* **Design of an automated system for the classification of plastic and glass bottles using sound**. *Séneca Repositorio institucional*, 2022. Disponível em:

<https://repositorio.uniandes.edu.co/handle/1992/56422>. Acesso em: 25 maio 2022.

SANTOS, Amélia S. F.; AGNELLI, José Augusto M.; MANRICH Sati. **Tendências e desafios da reciclagem de embalagens plásticas**. *Polímeros: Ciência e Tecnologia*, v. 14, n. 5, p. 307-312, 2004. DOI 10.1590/S0104-14282004000500006. Disponível em:

<https://www.scielo.br/j/po/a/pygZmYqm3yhzqVTzhwXvrNb/abstract/?lang=pt>. Acesso em: 05 jul. 2022.

SANTOS, Rejane Basbosa. **Deteção e localização de vazamentos em tubulações de gás de baixa pressão por meio de sensores acústicos e processamentos matemático de sinais**. Orientador: Ana Maria Frattini Fileti. 2015. 168 p. Tese (Doutorado em Engenharia Química) - Faculdade de Engenharia Química, Universidade Estadual de Campinas, 2015.

SARMENTO, Camila Freitas; REGIS, Danilo; ARAÚJO, Wagner. **Remoção adaptativa de ruído em sinais de áudio**. Congresso Norte Nordeste de Pesquisa e Inovação, 2012. Disponível em: <https://propi.iftto.edu.br/ocs/index.php/connepi/vii/paper/viewFile/2986/2540>. Acesso em: 24 mar. 2021.

SCAVINO, Edgar *et al.* **Application of automated image analysis to the identification and extraction of recyclable plastic bottles**. Journal of Zhejiang University SCIENCE A, v. 10, ed. 6, p. 794-799, 2009. DOI <http://dx.doi.org/10.1631/jzus.A0820788>. Disponível em: [https://www.researchgate.net/publication/227303235\\_Application\\_of\\_automated\\_image\\_analysis\\_to\\_the\\_identification\\_and\\_extraction\\_of\\_recyclable\\_plastic\\_bottles](https://www.researchgate.net/publication/227303235_Application_of_automated_image_analysis_to_the_identification_and_extraction_of_recyclable_plastic_bottles). Acesso em: 26 mar. 2021.

SCHMIDT, Ana Lucía. **FFT: Transformada Rápida de Fourier**. 2013. Disponível em: <http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-Schmidt%20Ana%20Luc%C3%ADa.pdf>. Acesso em: 4 maio 2021.

Scikit-Learn. Metrics and scoring: quantifying the quality of predictions. Disponível em: [https://scikit-learn.org/stable/modules/model\\_evaluation.html#accuracy-score](https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score). Acesso em: 20 set. 2021.

SILVA, Adilane Ribeiro da. Uma visão geral sobre *Machine Learning* – Classificação. **Data Science**. 13 ago. 2020. Disponível em: <https://operdata.com.br/blog/uma-visao-geral-sobre-machine-learning/>. Acesso em: 13 jul. 2022.

SILVA, Leandro Marcos da. **Classificação de domínios recém-registrados por meio de DNS passivo aplicando técnicas de aprendizado de máquina**. Orientador: Prof. Dr. Adriano Mauro Cansian. 2022. 77 f. TCC (Bacharel em Ciência da Computação) - Departamento de Ciências de Computação e Estatística do Instituto de Biociências Letras e Ciências Exatas, Universidade Estadual Paulista “Júlio de Mesquita Filho”, São José do Rio Preto, SP, 2022. Disponível em: <https://repositorio.unesp.br/handle/11449/216174?locale-attribute=en>. Acesso em: 13 jul. 2022.

TensorFlow. tf.keras.losses.SparseCategoricalCrossentropy. Disponível em: [https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy). Acesso em: 15 set. 2021.

TESSARINI, Letícia; FILETI, Ana Maria Frattini. **Banco de dados de áudio:** classificação de resinas plásticas para reciclagem usando processamento de sinais acústico e redes neurais artificiais. Repositório de Dados de Pesquisa Unicamp, 2022. Disponível em: <https://redu.unicamp.br/dataset.xhtml?persistentId=doi:10.25824/redu/UGMPJR>.

WAHAB, D. A. *et al.* **Development of a prototype automated sorting system for plastic recycling.** American journal of applied sciences, v. 3, ed. 7, p. 1924-1928, 2006. DOI <https://doi.org/10.3844/ajassp.2006.1924.1928>. Disponível em: [https://www.researchgate.net/publication/26439892\\_Development\\_of\\_a\\_Prototype\\_Automated\\_Sorting\\_System\\_for\\_Plastic\\_Recycling?enrichId=rgreq-3513092220cc494064f8ea141b23c661-XXX&enrichSource=Y292ZXJQYWdlOzI2NDM5ODkyO0FTOjEwMzY3NTY0NTEzNjkwMUAxNDAxNzI5NjA4Mjc0&el=1\\_x\\_2&\\_esc=publicationCoverPdf](https://www.researchgate.net/publication/26439892_Development_of_a_Prototype_Automated_Sorting_System_for_Plastic_Recycling?enrichId=rgreq-3513092220cc494064f8ea141b23c661-XXX&enrichSource=Y292ZXJQYWdlOzI2NDM5ODkyO0FTOjEwMzY3NTY0NTEzNjkwMUAxNDAxNzI5NjA4Mjc0&el=1_x_2&_esc=publicationCoverPdf). Acesso em: 26 mar. 2021.

WYSE, Lonce. **Audio spectrogram representations for processing with Convolutional Neural Networks.** Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN, Anchorage, v. 1, n. 1, p. 37-41, 2017. DOI 10.48550/arXiv.1706.09559. Disponível em: <https://arxiv.org/abs/1706.09559>. Acesso em: 23 mai. 2022.

APÊNDICE A – Programa pré-processamento dos dados de áudio para taxa de amostragem de 44.100 Hz em Python desenvolvido no ambiente Google Colab.

```
import librosa, librosa.display
import matplotlib.pyplot as plt
import numpy as np

file= "/content/drive/MyDrive/audios/banco_de_dados/PET/pet_1.wav"
signal, sr = librosa.load(file, sr=44100)
plt.subplots(figsize=(5, 2))
#Visualizar a forma de onda
librosa.display.waveplot(signal, sr=sr)
plt.title("PET (caso 1)")
plt.xlabel("Tempo (s)")
plt.ylabel("Amplitude")
plt.show()

fft = np.fft.fft(signal)

magnitude = np.abs(fft)
frequency = np.linspace(0, sr, len(magnitude))
#(intervalo de sequencia= 0Hz, taxa de amostragem (sr), nº de valores u
niformemente espaçados = o comprimento da magnitude)
plt.subplots(figsize=(5, 2))
#Considera somente a primeira parte do sinal
left_frequency = frequency
left_magnitude = magnitude

plt.plot(left_frequency, left_magnitude)
plt.title("PET (caso 1)")
plt.xlabel("Frequência (kHz)")
plt.ylabel("Magnitude")
plt.show()

n_fft = 2048 #número de amostras
hop_length = 512 #comprimento de salto

stft = librosa.core.stft(signal, hop_length=hop_length, n_fft=n_fft)
spectrogram = np.abs(stft)
#Plotagem do espectrograma

librosa.display.specshow(spectrogram, sr=sr, hop_length=hop_length)
plt.title("PET (caso 1)")
plt.xlabel("Tempo (s)")
plt.ylabel("Frequência (kHz)")
```

```
plt.colorbar() #Amplitude
plt.show()

#Plotagem do log espectrograma

log_spectrogram = librosa.amplitude_to_db(spectrogram)

librosa.display.specshow(log_spectrogram, sr=sr, hop_length=hop_length)
plt.title("PET (caso 1)")
plt.xlabel("Tempo (s)")
plt.ylabel("Frequência (kHz)")
plt.colorbar() #Amplitude
plt.show()

#Extrair as características do MFCC
MFCCs = librosa.feature.mfcc(signal, n_fft=n_fft, hop_length=hop_length
, n_mfcc=13)

#Plotar o espectrograma MFCC
librosa.display.specshow(MFCCs, sr=sr, hop_length=hop_length)
plt.title("PET (caso 1)")
plt.xlabel("Tempo (s)")
plt.ylabel("MFCC")
plt.colorbar()
plt.show()
```

Link para acessar o notebook do código no Google Colab:  
[https://colab.research.google.com/drive/1QfOeaBx1sR2vQGEUXxPMM3eFDE\\_COTII?usp=sharing](https://colab.research.google.com/drive/1QfOeaBx1sR2vQGEUXxPMM3eFDE_COTII?usp=sharing)

## APÊNDICE B – Programa processamento dos sinais de áudio utilizando MFCC.

```

import os
import librosa
import math
import json

DATASET_PATH = '/content/drive/MyDrive/audios/banco_de_dados_2'
JSON_PATH = "data_11.json"

SAMPLE_RATE = 22050 #taxa de amostragem
DURATION = 3 #segundos
SAMPLES_PER_TRACK = SAMPLE_RATE * DURATION

def save_mfcc(dataset_path, json_path, n_mfcc=13, n_fft=2048, hop_length=512, num_segments=7):

    #Construir um dicionário para armazenar os dados
    data = {
        "mapping": [],
        "mfcc": [],
        "labels": []
    }

    num_sample_per_segment = int(SAMPLES_PER_TRACK / num_segments)
    expected_num_mfcc_vectors_per_segment = math.ceil(num_sample_per_segment / hop_length)

    #Percorrer todas as classes
    for i, (dirpath, dirnames, filenames) in enumerate(os.walk(dataset_path)):

        # Percorrer todas as pastas das classes
        if dirpath is not dataset_path:

            # Salvar os labels
            dirpath_components = dirpath.split("/") # Ex.: classe/pet
            semantic_label = dirpath_components[-1]
            # Anexar mapping
            data["mapping"].append(semantic_label)
            # Imprimir para avisar que está sendo processado cada classe
            print("\nProcessing {}".format(semantic_label))

            # Processar arquivos de uma classe específica
            for f in filenames:

                # Carregar o arquivo de áudio

```

```

file_path = os.path.join(dirpath, f)
signal, sr = librosa.load(file_path, sr=SAMPLE_RATE)

# Processar segmentos extraíndo MFCC e armazenar os dados
for s in range(num_segments):
    start_sample = num_sample_per_segment * s
    finish_sample = start_sample + num_sample_per_segment

    # Obter os MFCC's
    mfcc = librosa.feature.mfcc(signal[start_sample:finish_sample
],
                                sr=sr,
                                n_fft=n_fft,
                                n_mfcc=n_mfcc,
                                hop_length=hop_length)

    mfcc = mfcc.T

    # Armazenar MFCC para o segmento se tiver o esperado
    if len(mfcc) == expected_num_mfcc_vectors_per_segment:
        data["mfcc"].append(mfcc.tolist())
        data["labels"].append(i-1)
        #Imprimir o nome do arquivo, o segmento e o formato é o cam
inho da pasta e especificar qual o segmento
        print("{}, segment:{}".format(file_path, s+1))

# Salvar
with open(json_path, "w") as fp:
    json.dump(data, fp, indent=4)

# Executar
if __name__ == "__main__":
    save_mfcc(DATASET_PATH, JSON_PATH, num_segments=7) #nun_segments = nú
mero de classes

```

Link para acessar o notebook do código no Google Colab:

<https://colab.research.google.com/drive/10aLBNqM30rad0KXvNq3kcbXncnG5DkwW?usp=sharing>

## APÊNDICE C - Programa de treinamento do modelo de rede neural convolucional em Python desenvolvido no ambiente Google Colab.

```

from google.colab import drive
drive.mount('/content/drive')

import json
import numpy as np
from sklearn.model_selection import train_test_split
import tensorflow.keras as keras
import matplotlib.pyplot as plt
#Matriz confusão
from sklearn.metrics import confusion_matrix
import itertools

DATA_PATH = "/content/drive/MyDrive/Pesquisa/Data/Teste 1 (Junto)/dataj_44.json"

#Definir uma função para abrir e ler os dados
def load_data(data_path):

    """Carrega conjunto de dados de treinamento do arquivo json

        : param data_path (str): Caminho para o arquivo json contendo
        dados
        : return X (ndarray): Entradas
        : return Y (ndarray): Saídas
    """

    with open(data_path, "r") as fp:
        data = json.load(fp)

    X = np.array(data["mfcc"])
    y = np.array(data["labels"])
    return X, y

#Criar uma função para a preparação dos dados
def prepare_datasets(test_size, validation_size):

    #Carregar os dados
    X, y = load_data(DATA_PATH)

    #Divisão dos dados em treinamento e teste
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
=validation_size)

    #Divisão dos dados em treinamento e validação

```

```

X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, test_size=validation_size)

#Matriz 3D -> (130,13,1)
X_train = X_train[..., np.newaxis] # matriz 4d -> (num_samples, 130, 13, 1)
X_validation = X_validation[..., np.newaxis]
X_test = X_test[..., np.newaxis]

return X_train, X_validation, X_test, y_train, y_validation, y_test

#Criar um função para o modelo
def build_model(input_shape):
    #Criar um modelo
    model = keras.Sequential()

    #1ª camada convolucional
    model.add(keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    # (qtd filtros, (tamanho da grade kernel), função de ativação, especificar a forma de entrada correta)
    # camada poolin máx -> irá reduzir a amostra da entrada
    model.add(keras.layers.MaxPool2D((3, 3), strides=(2, 2), padding='same'))
    # ((tamanho da grade), passos, tipo de preenchimento)
    # camada de normalização -> padroniza as ativações, acelera o treinamento, modelo mais confiável
    model.add(keras.layers.BatchNormalization())

    #2ª camada convolucional
    model.add(keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    model.add(keras.layers.MaxPool2D((3, 3), strides=(2, 2), padding='same'))
    model.add(keras.layers.BatchNormalization())

    #3ª camada convolucional
    model.add(keras.layers.Conv2D(32, (2, 2), activation='relu', input_shape=input_shape))
    model.add(keras.layers.MaxPool2D((2, 2), strides=(2, 2), padding='same'))
    model.add(keras.layers.BatchNormalization())

    #Nivelar a saída das camadas conv. e ajustar a camada densa
    model.add(keras.layers.Flatten())
    # camada densa
    model.add(keras.layers.Dense(64, activation='relu'))
    # (qtd de neurônios, função de ativação)
    # para evitar o overfitting

```

```

    model.add(keras.layers.Dropout(0.3)) # probabilidade de abandono de
30%

    #Camada de saída
    model.add(keras.layers.Dense(7, activation='softmax'))
    # a quantidade neurônio vai ser a quantidade de categorias que tenh
o
    # softmax -> cria uma probabilidade e distribui para neurônio de ca
da categoria, índice de segurança com o valor mais alto

    return model

#Definir a função de previsão
def predict(model, X, y):
    #Inserindo um novo eixo e copiando o resto
    X = X[np.newaxis, ...]

    # prediction = [[0.1, 0.2, ...]] 2D
    prediction = model.predict(X) # X -> (1, 130, 13, 1)

    #Extrair o índice com o valor máx.
    predicted_index = np.argmax(prediction, axis=1) #1D
    print("Índice esperado: {}, Índice previsto: {}".format(y, predicte
d_index))

#Definir uma função para a plotagem da precisão e erro
def plot_history(history):

    fig, axs = plt.subplots(2)

    # criar subplot de precisão
    axs[0].plot(history.history["accuracy"], label="Acurácia treinament
o")
    axs[0].plot(history.history["val_accuracy"], label="Acurácia teste"
)
    axs[0].set_ylabel("Acurácia")
    axs[0].legend(loc="lower right")
    axs[0].set_title("Avaliação acurácia e perda")

    # criar subplot de erro
    axs[1].plot(history.history["loss"], label="Perda treinamento")
    axs[1].plot(history.history["val_loss"], label="Perda teste")
    axs[1].set_ylabel("Perda")
    axs[1].set_xlabel("Épocas")
    axs[1].legend(loc="upper right")

    plt.show()
if __name__ == "__main__":

```

```

X_train, X_validation, X_test, y_train, y_validation, y_test = prep
are_datasets(0.20, 0.20)

input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]
)
model = build_model(input_shape)

optimizer = keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.summary()

history = model.fit(X_train, y_train, validation_data=(X_validation
, y_validation), batch_size=32, epochs=30)

plot_history(history)

test_error, test_accuracy = model.evaluate(X_test, y_test, verbose=
1)
#Imprimir o resultado
print("A precisão no conjunto de teste é: {}".format(test_accuracy)
)

X = X_test[50]
y = y_test[50]

predict(model, X, y)

#Predição de modelo
predictions = model.predict(x=X_test, batch_size=10, verbose=0)
rounded_predictions = np.argmax(predictions, axis=-1)

cm = confusion_matrix(y_test, rounded_predictions, normalize='true')
acc = accuracy_score(y_test, rounded_predictions)

plt.subplots(figsize=(18, 18))
#Definição da função matriz de confusão
def plot_cm (cm, classes, normalize=True, title='Matriz de confusão ',
cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))

```

```
plt.xticks(tick_marks,classes,rotation=45)
plt.yticks(tick_marks,classes)

thresh = cm.max()/7.
for i,j in itertools.product(range(cm.shape[0]),range(cm.shape[1]))
:
    plt.text(j,i,cm[i,j],horizontalalignment='center',color='black'
if cm[i,j]>thresh else 'black')

plt.tight_layout()
plt.ylabel('Classe real')
plt.xlabel('Classe predita')

cm_plot_labels = ['pet', 'pead', 'pvc', 'pebd', 'pp', 'ps', 'outros']
plot_cm(cm=cm,classes=cm_plot_labels,title='Matriz de confusão')
```

Link para acessar o notebook do código no Google Colab:  
<https://colab.research.google.com/drive/1j4OLqsrGf2B6WksOHvZGQV8Qbn5Jxn0f?usp=sharing>

APÊNDICE D – Programa para o modelo rede neural recorrente do tipo LSTM em Python desenvolvido no ambiente Google Colab.

```

from google.colab import drive
drive.mount('/content/drive')

import json
import numpy as np
from sklearn.model_selection import train_test_split
import tensorflow.keras as keras
import matplotlib.pyplot as plt
#Matriz confusão
from sklearn.metrics import confusion_matrix
import itertools

DATA_PATH = "/content/drive/MyDrive/Pesquisa/Data/Teste 1 (Junto)/dataj_44.json"

#Definir uma função para abrir e ler os dados
def load_data(data_path):

    """Carrega conjunto de dados de treinamento do arquivo json

        : param data_path (str): Caminho para o arquivo json contendo
        dados
        : return X (ndarray): Entradas
        : return Y (ndarray): Saídas
    """

    with open(data_path, "r") as fp:
        data = json.load(fp)

    X = np.array(data["mfcc"])
    y = np.array(data["labels"])
    return X, y

#Criar uma função para a preparação dos dados
def prepare_datasets(test_size, validation_size):

    #Carregar os dados
    X, y = load_data(DATA_PATH)

    #Divisão dos dados em treinamento e teste
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =test_size)

    #Divisão dos dados em treinamento e validação

```

```

X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, test_size=validation_size)

return X_train, X_validation, X_test, y_train, y_validation, y_test

#Criar uma função para o modelo
def build_model(input_shape):
    #Criar um modelo
    model = keras.Sequential()

    # Duas camadas LSTM
    model.add(keras.layers.LSTM(64, input_shape=input_shape, return_sequences=True))
    # (n° de unidades da camada, forma de entrada, sequencia de retorno que tem que ser verdadeiro)
    model.add(keras.layers.LSTM(64))
    # a segunda camada é igual a modelada, passa somente o n° de unidades

    # Camada densa
    model.add(keras.layers.Dense(64, activation='relu'))
    model.add(keras.layers.Dropout(0.3))
    # camada de exclusão (Dropout) para evitar o overfitting

    #Camada de saída
    model.add(keras.layers.Dense(7, activation='softmax'))
    # a quantidade neurônio vai ser a quantidade de categorias que tem o
    # softmax -> cria uma probabilidade e distribui para neurônio de cada categoria, índice de segurança com o valor mais alto

    return model

#Definir a função de previsão
def predict(model, X, y):

    X = X[np.newaxis, ...]
    prediction = model.predict(X)

    #Extrair o índice com o valor máx.
    predicted_index = np.argmax(prediction, axis=1)
    print("Índice esperado: {}, Índice previsto: {}".format(y, predicted_index))

#Definir uma função para a plotagem da precisão e erro
def plot_history(history):

    fig, axs = plt.subplots(2)

```

```

# criar subplot de precisão
axs[0].plot(history.history["accuracy"], label="Acurácia treinamento")
axs[0].plot(history.history["val_accuracy"], label="Acurácia teste")
)
axs[0].set_ylabel("Acurácia")
axs[0].legend(loc="lower right")
axs[0].set_title("Avaliação acurácia e perda")

# criar subplot de erro
axs[1].plot(history.history["loss"], label="Perda treinamento")
axs[1].plot(history.history["val_loss"], label="Perda teste")
axs[1].set_ylabel("Perda")
axs[1].set_xlabel("Épocas")
axs[1].legend(loc="upper right")

plt.show()

if __name__ == "__main__":

    X_train, X_validation, X_test, y_train, y_validation, y_test = prepare_datasets(0.20, 0.20)

    input_shape = (X_train.shape[1], X_train.shape[2])
    model = build_model(input_shape)

    optimizer = keras.optimizers.Adam(learning_rate=0.0001)
    model.compile(optimizer=optimizer,
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])

    model.summary()

    history = model.fit(X_train, y_train, validation_data=(X_validation,
y_validation), batch_size=32, epochs=30)

    plot_history(history)

    test_error, test_accuracy = model.evaluate(X_test, y_test, verbose=1)
    print("A precisão no conjunto de teste é: {}".format(test_accuracy))

)

X = X_test[50]
y = y_test[50]

```

```

predict(model, X, y)

#Predição de modelo
predictions = model.predict(x=X_test,batch_size=10,verbose=0)
rounded_predictions = np.argmax(predictions,axis=-1)

cm = confusion_matrix(y_test, rounded_predictions, normalize='true')
acc = accuracy_score(y_test, rounded_predictions)

plt.subplots(figsize=(18, 18))
#Definição da função matriz de confusão
def plot_cm (cm, classes, normalize=True, title='Matriz de confusão ',
cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks,classes,rotation=45)
    plt.yticks(tick_marks,classes)

    thresh = cm.max()/7.
    for i,j in itertools.product(range(cm.shape[0]),range(cm.shape[1]))
:
        plt.text(j,i,cm[i,j],horizontalalignment='center',color='black'
if cm[i,j]>thresh else 'black')

    plt.tight_layout()
    plt.ylabel('Classe real')
    plt.xlabel('Classe predita')

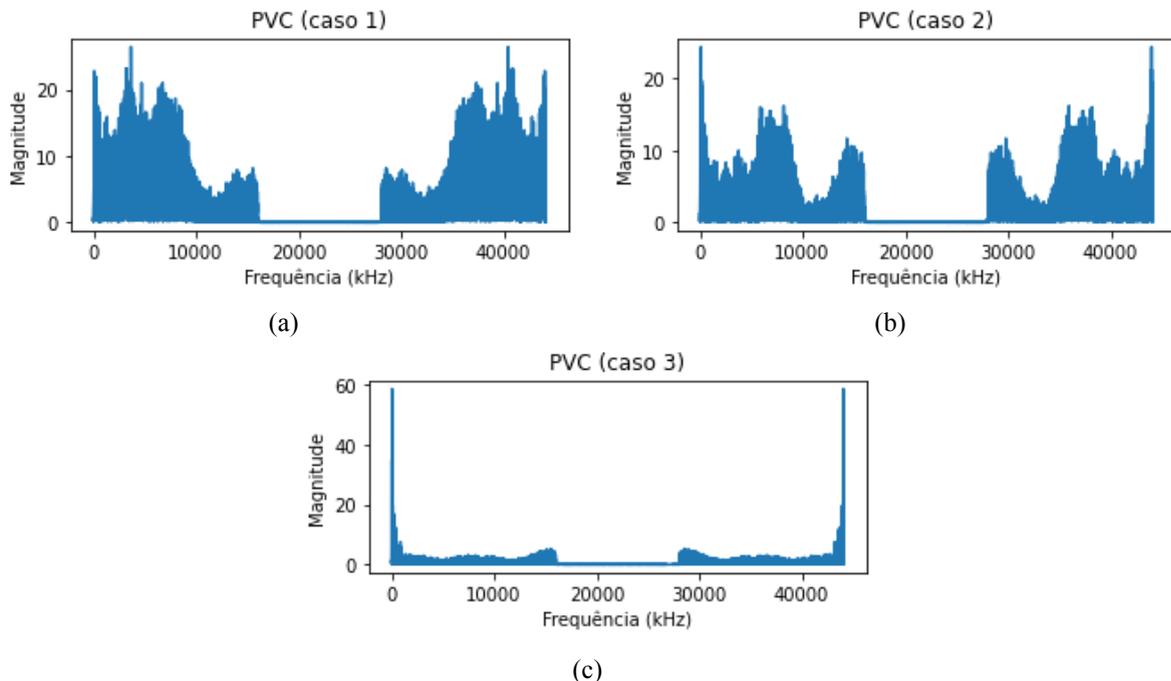
cm_plot_labels = ['pet', 'pead', 'pvc', 'pebd', 'pp', 'ps', 'outros']
plot_cm(cm=cm,classes=cm_plot_labels,title='Matriz de confusão')

```

Link para acessar o notebook do código no Google Colab:  
<https://colab.research.google.com/drive/1yEHQ4VkpS8LNakwmmJAsqk4vBT8mLtZY?usp=sharing>

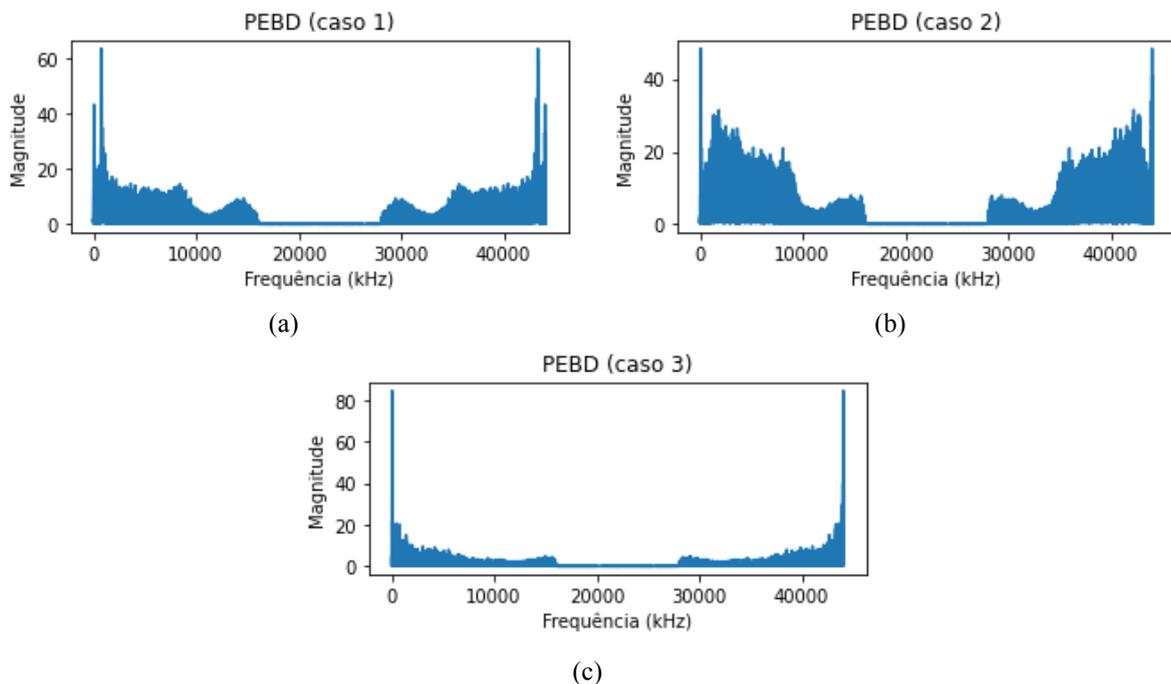
APÊNDICE E – Restante do gráfico da transformada de Fourier para as categorias PVC, PEBD, PP, PS e Outros.

Figura 0.1: Gráfico da transformada rápida de Fourier da categoria PVC (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.



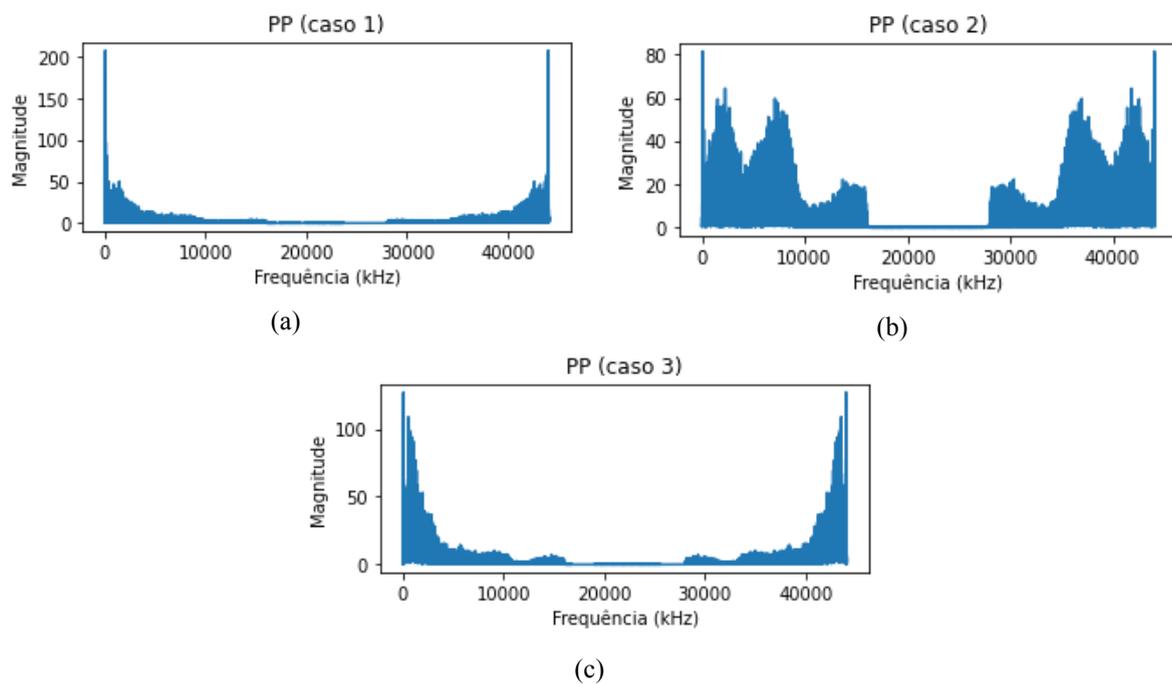
Fonte: Autora, 2022.

Figura 0.2: Gráfico da transformada rápida de Fourier da categoria PEBD (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.



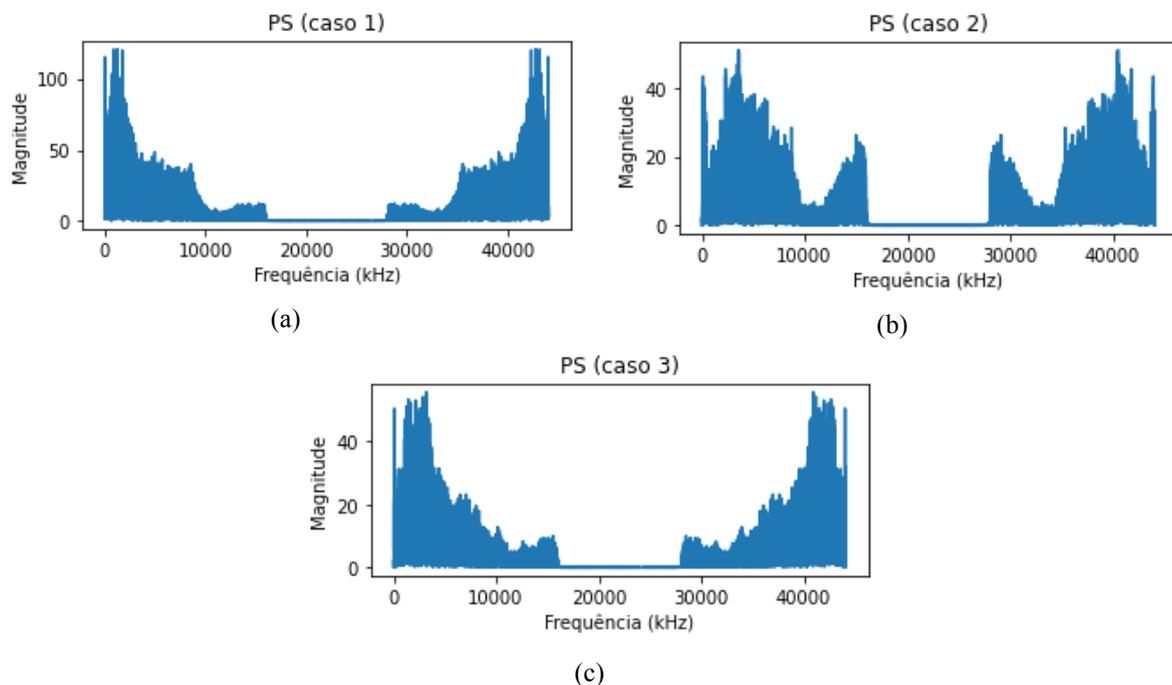
Fonte: Autora, 2022.

Figura 0.3: Gráfico da transformada rápida de Fourier da categoria PP (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.



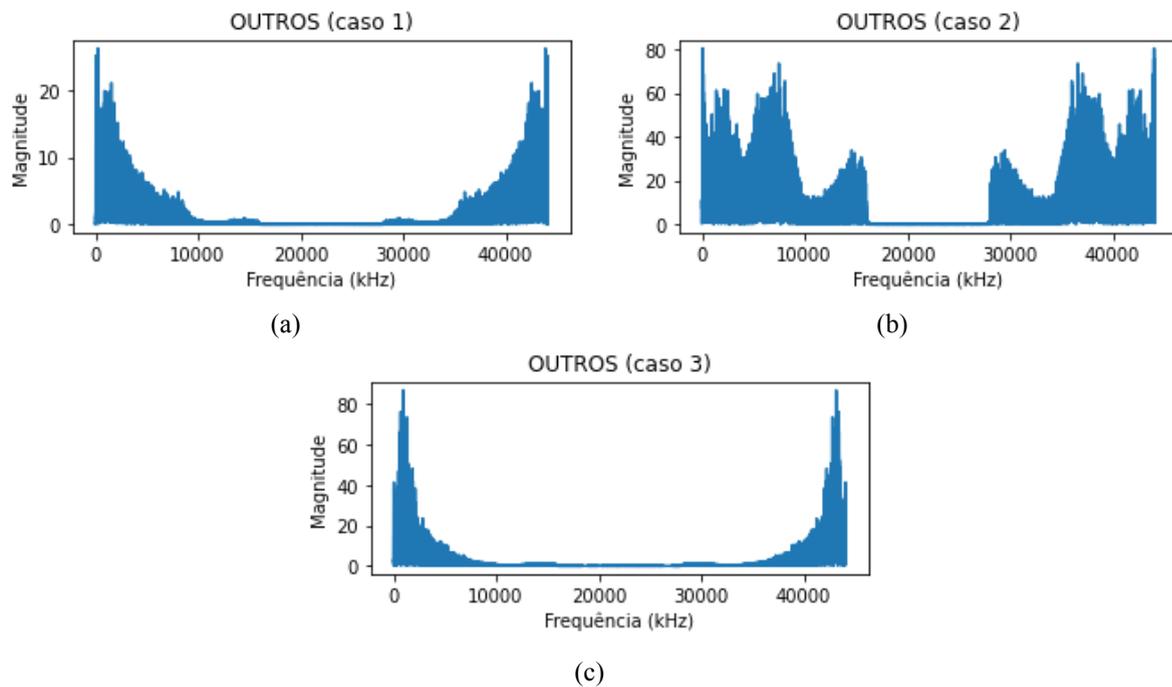
Fonte: Autora, 2022.

Figura 0.4: Gráfico da transformada rápida de Fourier da categoria PS (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.



Fonte: Autora, 2022.

Figura 0.5: Gráfico da transformada rápida de Fourier da categoria OUTROS (a) caso 1 – amassadas em conjunto (b) caso 2 – amassadas por peça (c) caso 3 – amassadas de forma mais lenta.



(c)  
Fonte: Autora, 2022.