

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**FERNANDO KELVIN DA SILVA SOARES**

**MÉTODO DE CLASSIFICAÇÃO DE FLUXO UTILIZANDO  
SENSORIAMENTO ACÚSTICO DISTRIBUÍDO E REDES NEURAIAS**

**CURITIBA**

**2022**

**FERNANDO KELVIN DA SILVA SOARES**

**MÉTODO DE CLASSIFICAÇÃO DE FLUXO UTILIZANDO  
SENSORIAMENTO ACÚSTICO DISTRIBUÍDO E REDES  
NEURAIS**

**Flow Classification Method Using Distributed Acoustic Sensing And  
Neural Networks**

Dissertação apresentado(a) como requisito para obtenção do título(grau) de Mestre em Engenharia Elétrica e Informática Industrial, do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof. Dr. Cicero Martelli  
Coorientador(a): Prof. Dr. Jean Carlos Cardozo da Silva

**CURITIBA**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Campus Curitiba**



FERNANDO KELVIN DA SILVA SOARES

**MÉTODO DE CLASSIFICAÇÃO DE FLUXO UTILIZANDO SENSORIAMENTO ACÚSTICO DISTRIBUÍDO E REDES NEURAIS**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Automação E Sistemas.

Data de aprovação: 21 de Outubro de 2022

Dr. Cicero Martelli, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Aleksander Sade Paterno, Doutorado - Fundação Universidade do Estado de Santa Catarina (Udesc)

Dr. Cesar Janeczko, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Daniel Rodrigues Pipa, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 21/10/2022.

Dedico este trabalho a minha esposa Barbara e  
ao meu filho Martin, por tornarem a minha vida  
mais feliz.

## **AGRADECIMENTOS**

Primeiramente agradeço a minha esposa, Barbara Prado Zerbato por me apoiar nas horas difíceis e incentivar o meu desenvolvimento pessoal e profissional.

A meus pais, Anselmo Soares e Soraia Batista da Silva por terem investido na minha educação.

Ao Professor Dr. Cicero Martelli por ter me orientado e incentivado, sempre acreditando no meu potencial e confiando no meu trabalho.

Aos queridos amigos, Guilherme Dutra e Emerson Silva pela parceria nessa jornada.

Ao colega, Guilherme Weber pela gentileza em ceder os dados utilizados neste trabalho e pela atenção com as minhas dúvidas.

E a todos aqueles que me estenderam a mão sem esperar nada em troca.

"You can't always get what you want  
But if you try sometimes, you just might find  
You just might find  
You get what you need!  
Oh, yeah!"  
(JAGGER, Mick; RICHARDS, Keith, 1969).

## RESUMO

SOARES, Fernando Kelvin da Silva. **Método de Classificação de Fluxo Utilizando Sensoriamento Acústico Distribuído e Redes Neurais**. 2022. 59 f. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2022.

O trabalho apresentado nesta dissertação de mestrado tem por objetivo propor uma técnica de classificação para diferenciar o tipo fluxo em uma tubulação de testes, nas seguintes classes: mono líquido, mono gás e multifásico (composto por líquido e gás). Os dados de fluxo foram coletados de experimentos com diferentes valores de velocidade fase de líquido e gás, utilizando um sensor acústico distribuído (*DAS - Distributed Acoustic Sensing*), do tipo  $\Phi$ -OTDR de detecção heteródina. A metodologia utilizada para a realização da tarefa de classificação baseia-se em uma técnica de transferência de aprendizado aplicada a redes neurais, onde um modelo de rede pré-treinado é utilizado para extração de vetores de associações (*embeddings*), os quais são utilizados como valores de entrada para o treinamento de uma rede de classificação customizada. Desta forma, para gerar as bases de treinamento e teste da rede customizada, os dados dos experimentos foram pré-processados, convertidos em arquivos de áudio e inferidos pelo modelo pré-treinado, gerando vetores de *embeddings* relacionados ao tipo de fluxo respectivo. Após o treinamento o classificador foi capaz de atingir uma acurácia acima de 95,0 % para os dados de treinamento e acima de 87,0 % para os dados de teste, mostrando que a metodologia utilizada se adapta bem a tarefa proposta e que possui potencial de ser aplicada em tarefas de classificação mais complexas, ampliando o número de classes de fluxo.

**Palavras-chave:** DAS. Classificação de áudio. Redes neurais. Transferência de aprendizado.

## ABSTRACT

SOARES, Fernando Kelvin da Silva. **Method for Flow Classification using Fiber-Optic Distributed Acoustic Sensing and Neural Networks**. 2022. 59 p. Dissertation (Master's Degree in Electrical and Computer Engineering) – Universidade Tecnológica Federal do Paraná. Curitiba, 2022.

The work presented in this master's dissertation proposes a classification technique to differentiate the flow type in a test tube, in one of the following classes: Only liquid, only gas and multiphase (composed by liquid and gas). The flow data were collected from experiments with different values of phase velocity for liquid and gas, using a  $\Phi$ -OTDR distributed acoustic sensor (DAS), with heterodyne detection. The methodology used to perform the classification task is based on a transfer learning technique applied to neural networks, where a pre-trained model is used for extracting a vector of embeddings, which are used as input values for the training of a customized classification network. To generate the customized network training and test datasets, the experiment data were pre-processed, converted into audio files, and inferred by the pre-trained model, generating embedding vectors related to the respective flow type. After training, the classifier was able to achieve an accuracy above 95,0 % for the training dataset and 87,0 % for the test datasets for each class, showing that the methodology used adapts well to the proposed task and that it has potential to be applied in more complex classification tasks extending the number of flow classes.

**Keywords:** DAS. Audio classifier. Neural networks. Transfer learning.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração de funcionamento de um sensor DAS. . . . .	14
Figura 2 – Perfis de escoamento na vertical. . . . .	16
Figura 3 – Exemplo de sinal de um sensor acústico distribuído monitorando uma tubulação com fluxo multifásico sendo representado nos eixos do espaço e do tempo. . . . .	17
Figura 4 – Espectro típico de espalhamento espontâneo de matéria de estado sólido. . .	19
Figura 5 – Representação do espalhamento Rayleigh. . . . .	21
Figura 6 – Diagrama simplificado do OTDR. . . . .	22
Figura 7 – Sinal obtido pelo retroespalhamento Rayleigh utilizando fonte de baixa coerência. . . . .	22
Figura 8 – Representação da arquitetura de uma MLP com 2 camadas escondidas. . . .	24
Figura 9 – Modelo não linear de um neurônio, denominado $k$ . . . . .	25
Figura 10 – Gráfico da função ReLu. . . . .	26
Figura 11 – Gráfico da função Sigmoide. . . . .	27
Figura 12 – Ilustração dos dois fluxos básicos de informação em uma rede MLP. . . . .	27
Figura 13 – Operação de correlação cruzada bidimensional. . . . .	29
Figura 14 – Cálculo de correlação cruzada com 2 canais de entrada. . . . .	30
Figura 15 – Diagrama de funcionamento de uma camada convolucional com entrada $4 \times 4 \times 2$ , utilizando dois kernels de $3 \times 3 \times 3$ , com Padding = 1 e Stride = 1. . . .	31
Figura 16 – Max-pooling com uma janela de $2 \times 2$ . As porções sombreadas são utilizadas para o cálculo do primeiro elemento de saída, da seguinte forma: $max(0, 1, 3, 4) = 4$ . . . . .	31
Figura 17 – Camadas da rede AlexNet. . . . .	32
Figura 18 – Esquema das instalações do laboratório NUEM. . . . .	35
Figura 19 – Sensor DAS $\Phi$ -OTDR desenvolvido pelo Laboratório de Tecnologia de Sistemas de Sensores UTFPR (LTSS). . . . .	36
Figura 20 – Cabo Furukawa OpticLan ABNT. . . . .	37
Figura 21 – Representação em superfície do sinal de um experimento multifásico. . . . .	39
Figura 22 – Fluxo processamento de uma entrada de áudio pela rede YAMNet. . . . .	40
Figura 23 – Arquitetura detalhada do Classificador de fluxo. . . . .	41
Figura 24 – Conversão de trechos de 1 m dos dados do DAS em arquivos de áudio. . . .	42
Figura 25 – Trecho do Colab Notebook apresentando parte do código de implementação do classificador customizado. . . . .	44
Figura 26 – Fluxo de implementação e treinamento do classificador customizado na plataforma Colab. . . . .	45
Figura 27 – Gráficos de evolução da precisão e erro durante o treinamento do classificador customizado com dados da seção horizontal da tubulação. . . . .	47
Figura 28 – Matrizes de confusão para as classificações realizadas pelos modelos em 30 épocas na base de testes contendo dados da seção horizontal da tubulação. . .	48
Figura 29 – Matrizes de confusão para a classificação dos dados de todos os experimentos referente a seção vertical do tubo utilizando modelo treinado com dados da seção horizontal. . . . .	49
Figura 30 – Perfis de escoamento de acordo com as velocidades de fase e orientação do fluxo. . . . .	50

Figura 31 – Gráficos de evolução da precisão e erro durante o treinamento do classificador customizado com dados das seções horizontal e vertical. . . . .	51
Figura 32 – Matrizes de confusão para as classificações realizadas pelos modelos em 30 épocas na base de testes contendo dados da seção horizontal e vertical da tubulação. . . . .	52
Figura 33 – Matrizes de confusão para da seção vertical da tubulação. À esquerda resultado com o modelo treinado apenas com os dados da seção horizontal. À direita resultado do modelo treinado com as seções horizontal e vertical. . .	53

## LISTA DE TABELAS

Tabela 1 – Relação dos experimentos utilizados para o desenvolvimento do trabalho. . .	38
Tabela 2 – Divisão dos experimentos entre as bases de treinamento e teste para um trecho de 9 m da seção horizontal da tubulação . . . . .	46
Tabela 3 – Divisão dos experimentos para a base balanceada de treinamento. . . . .	50

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

### SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CNN	Rede Neural Convolucional, do inglês <i>convolutional Neural network</i>
CRN	Ruído Rayleigh Coerente, do inglês <i>Coherent Rayleigh Noise</i>
DAS	Sensoriamento Acústico Distribuído <i>Distributed Acoustic Sensing</i>
FC	Completamente Conectada, do inglês <i>Fully Connected</i>
FUT	Fibra sob Teste, do inglês <i>Fiber Under Test</i>
GPU	Unidade de processamento Gráfico, do inglês <i>Graphics Processing Unit</i>
LTSS	Laboratório de Tecnologia de Sistemas de Sensores
MLP	Perceptron Multicamadas, do inglês <i>Multi-Layer Perceptron</i>
NUEM	Núcleo de Escoamento Multifásico
OTDR	Refletômetro Óptico no Domínio do Tempo, do inglês <i>Optical Time Domain Reflectometer</i>
PhD	Fotodetector, do inglês <i>Photodetector</i>
RTToF	Tempo de Voo de Ida e Volta, do inglês <i>Round Trip Time of Flight</i>
UTFPR	Universidade Tecnológica Federal do Paraná
$\Phi$ -OTDR	Refletômetro Óptico Sensível a Fase no Domínio do Tempo, do inglês <i>Phase-sensitive Optical Time Domain Reflectometer</i>

### ACRÔNIMOS

ReLu	Unidades Lineares Retificadas, do inglês <i>Rectified Linear Units</i>
Argmax	Argumento Máximo, do inglês, do inglês <i>Maximum Argument</i>

## LISTA DE SÍMBOLOS

### LETRAS LATINAS

$b$	<i>Bias</i> do Perceptron
$c$	Velocidade da luz no vácuo
$F$	Função de aproximação
$L_{coh}$	Comprimento de coerência da fonte
$n$	Índice de refração
$n_{eff}$	Índice de refração efetivo
$p$	Coefficiente fotoelástico
$k$	Constante de Boltzmann
$T_F$	Temperatura fictícia
$v$	Valor interno do Perceptron
$w$	Pesos da rede
$x$	Valor de entrada
$y$	Valor de saída

### LETRAS GREGAS

$\alpha_{Re}$	Coefficiente de espalhamento Rayleigh
$\beta_c$	Compressibilidade isotérmica
$\Delta_t$	Delta de tempo
$\lambda$	Comprimento de onda
$\omega_0$	Frequência da fonte
$\Phi$	Fase
$\varphi$	Função de ativação
$\pi$	Pi (constante matemática)

## SUMÁRIO

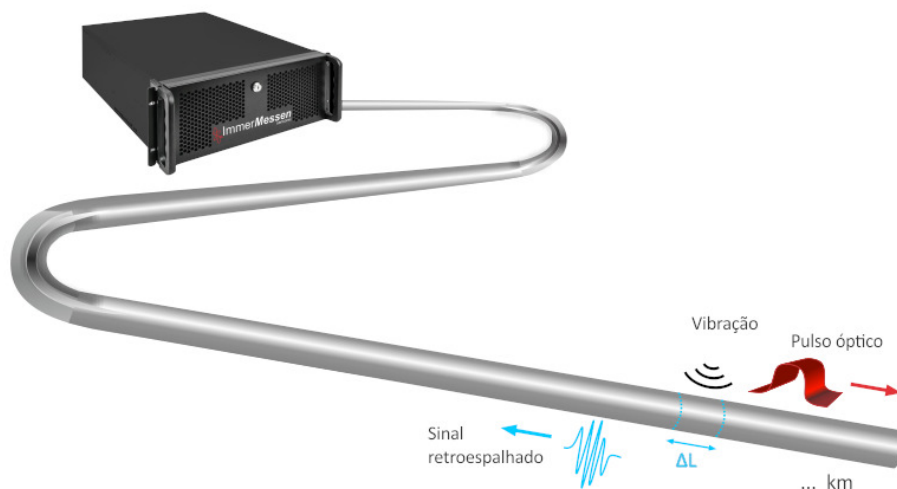
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>14</b>
1.1	MOTIVAÇÃO . . . . .	15
1.2	OBJETIVOS . . . . .	18
<b>2</b>	<b>REVISÃO DA LITERATURA</b> . . . . .	<b>19</b>
2.1	SENSORIAMENTO ACÚSTICO DISTRIBUÍDO . . . . .	19
2.1.1	Espalhamento Rayleigh . . . . .	20
2.1.2	OTDR e $\Phi$ -OTDR . . . . .	20
2.2	REDES NEURAIIS PROFUNDAS . . . . .	23
2.2.1	Redes Profundas de Alimentação Direta . . . . .	24
2.2.2	Redes Neurais Convolucionais . . . . .	28
2.2.3	Transferência de Aprendizado . . . . .	32
2.2.4	Ajuste fino . . . . .	33
2.2.5	Extração de Recursos . . . . .	33
<b>3</b>	<b>MATERIAIS E MÉTODOS</b> . . . . .	<b>35</b>
3.1	INSTALAÇÕES . . . . .	35
3.2	EXPERIMENTOS . . . . .	37
3.3	IMPLEMENTAÇÃO DA REDE NEURAL . . . . .	39
3.3.1	Conversão dos Dados em Áudio . . . . .	42
3.3.2	Treinamento da Rede Neural . . . . .	43
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b> . . . . .	<b>46</b>
<b>5</b>	<b>CONCLUSÕES E PERSPECTIVAS</b> . . . . .	<b>54</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>55</b>

## 1 INTRODUÇÃO

Sensores baseados em fibra óptica têm sido amplamente utilizados, tanto nas áreas de pesquisa científica quanto na indústria, dados os benefícios associados às suas características físicas, como: imunidade a interferências eletromagnéticas, isolamento elétrico, baixa atenuação de sinal, dimensões reduzidas e alta sensibilidade (OTHONOS, 1997; GIALLORENZI *et al.*, 1982). A variabilidade de técnicas de implementação, como é o caso das redes de Bragg, sensores de amplitude, interferômetros e sensores distribuídos (DAS - *Distributed Acoustic Sensing*) (LOPEZ-HIGUERA *et al.*, 2011), possibilita a utilização de sensores ópticos em diversas aplicações como: medição de temperatura (MARTELLI *et al.*, 2012), monitoramento de estruturas (HU *et al.*, 2021) e medição de velocidade de fluxo em fluidos (VAHABI *et al.*, 2020).

No caso dos sensores DAS, toda a extensão da fibra óptica é utilizada como elemento sensor, permitindo o monitoramento perturbações acústicas ao longo de grandes estruturas. Conforme apresentado na Figura 1, o princípio de funcionamento de um sensor DAS consiste no lançamento de um pulso através da fibra óptica e na leitura do sinal de retorno gerado pelo espalhamento da luz ao encontrar não-uniformidades espaciais das moléculas de sílica do núcleo da fibra (IMMERMESSEN, 2022). Perturbações mecânicas ao longo da fibra alteram a distribuição espacial destas moléculas localmente, de forma que o sinal retroespalhado também sofre alteração.

**Figura 1 – Ilustração de funcionamento de um sensor DAS.**



**Fonte: Adaptação de (IMMERMESSEN, 2022).**

Ao adquirir o sinal retroespalhado ao longo do tempo, é possível associar cada amostra a uma posição relativa na fibra através do cálculo do tempo de ida e volta do pulso relativo aquela posição, permitindo a construção de assinaturas espaciais do sinal a cada cilho de leitura (IMMERMESSEN, 2022). O funcionamento dos sensores DAS será explorado mais a fundo na seção 2.1.

Os recentes avanços na fabricação de lasers com alto comprimento de coerência e baixo ruído possibilitaram o aumento da precisão dos sensores DAS (FERNANDEZ-RUIZ *et al.*, 2018), podendo alcançar resoluções espaciais de aproximadamente 0.5 m utilizando a arquitetura  $\Phi$ -OTDR (BAO; CHEN, 2012). Contudo, o processamento e interpretação dos dados provenientes desses sensores pode ser complexo, tanto pelo volume de dados na ordem de centenas de megabytes para cada segundo de aquisição, quanto pela dificuldade de ressaltar características específicas do sinal para extrair as informações desejadas.

Em paralelo, os constantes avanços na área de redes neurais profundas (DNNs - *Deep Neural Networks*) têm apresentado alta acurácia e eficiência (MCLOUGHLIN *et al.*, 2015; CHEN *et al.*, 2014) em tarefas de classificação de dados de imagem e áudio. Tais resultados, fizeram com que esta tecnologia se espalhasse por diversas aplicações inclusive sendo utilizada no domínio dos sensores ópticos. Trabalhos recentes apresentam a utilização de redes neurais profundas aplicadas a classificação e detecção de perturbações causadas por pedestres, trens em movimento e ruídos ambientes ao longo de uma ferrovia, monitorados por sensores acústicos distribuídos do tipo  $\Phi$ -OTDR (WANG *et al.*, 2020).

Buscando explorar a junção dessas tecnologias, o trabalho apresentado nesta dissertação tem por objetivo propor uma metodologia de utilização de redes neurais para realizar a tarefa de classificação do tipo de fluxo, para uma base de dados pré-existente, proveniente de experimentos realizados em uma tubulação de testes, utilizando um sensor acústico distribuído.

## 1.1 MOTIVAÇÃO

Uma das áreas que mais se beneficia do sensoriamento distribuído utilizando fibra óptica é a indústria de petróleo e gás, uma vez que suas tubulações e estruturas são extensas e em muitos casos estão instaladas em ambientes hostis para sensores elétricos como plataformas de extração de petróleo em alto mar, por exemplo.

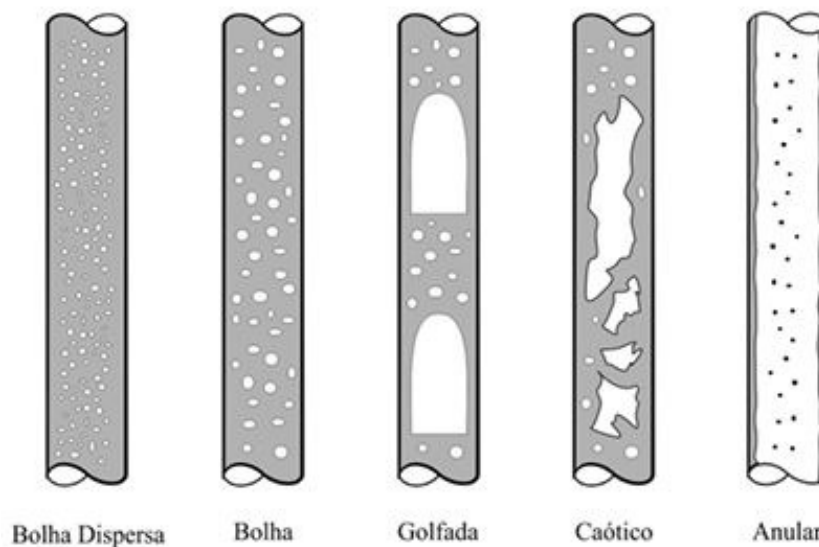
Neste trabalho, serão utilizados experimentos realizados em uma planta de testes para simulação de regimes de escoamento em tubulações horizontais e verticais. Os experimentos



foram realizados utilizando líquido e gás em regimes do tipo monofásico, onde tem-se apenas ou líquido (mono líquido) ou gás (mono gás) fluindo pela tubulação, e regimes do tipo multifásico, onde existe uma combinação de líquido e gás fluído pela tubulação.

É importante ressaltar que dentro da categoria de regime multifásico existem diversos perfis de escoamento dependendo da velocidade superficial de líquido e gás e da orientação da tubulação. A Figura 2 apresenta exemplos de perfis de escoamento para uma tubulação na vertical, onde o líquido é apresentado na cor cinza e o gás na cor branca. No entanto, para este trabalho não é explorada a diferenciação destes perfis de forma que todos os experimentos do tipo multifásico foram unidos em uma única classe.

**Figura 2 – Perfis de escoamento na vertical.**

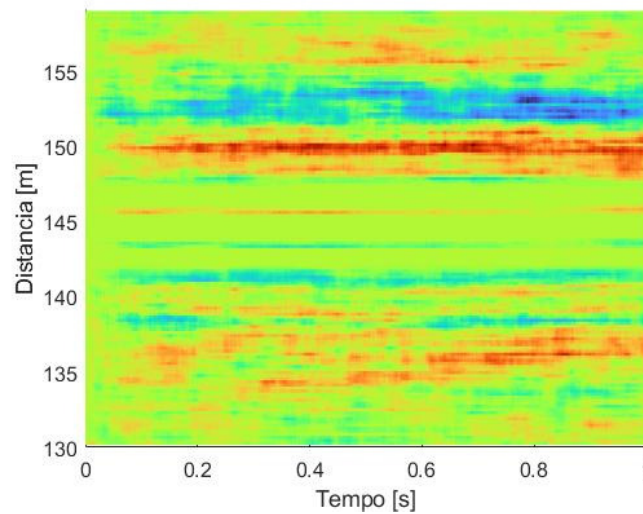


Fonte: (STUCKENBRUCK, 2022).

Uma forma comum de análise dos dados dos sensores DAS é a utilização de representações gráficas no domínio do tempo e da frequência. A Figura 3 apresenta um exemplo de representação gráfica do sinal de um sensor DAS no das dimensões do comprimento e do tempo referente a um fluxo do tipo multifásico em uma tubulação de testes. A intensidade das perturbações acústicas está mapeada em uma escala de cor utilizando o azul para representar os menores valores, o verde para valores intermediários e o vermelho para os maiores valores.

Dependendo da análise realizada, é comum que os dados sejam pré-processados utilizando técnicas como: filtros, derivadas, integrais e correlação. Este processamento busca ressaltar características e padrões nos gráficos que possam ser relacionados a uma grandeza física, como a deformação da estrutura, ou a ocorrência de eventos no sinal que possam ser associados a uma fonte de perturbação, como uma máquina trabalhando próxima a estrutura monitorada.

**Figura 3 – Exemplo de sinal de um sensor acústico distribuído monitorando uma tubulação com fluxo multifásico sendo representado nos eixos do espaço e do tempo.**



**Fonte: Autoria própria.**

Uma alternativa a representação gráfica de sinais de sensores é a transformação destes em uma representação audível através da técnica de Sonificação (NEES; WALKER, 2012). A conversão de sinais em formatos sonoros busca explorar a excelente capacidade da audição humana em reconhecer mudanças temporais e padrões (BREGMAN, 1990).

A sonificação pode ser baseada em eventos, onde variações do sinal de entrada são mapeadas na forma de variações dos atributos do áudio de saída. Sendo utilizada na geração de alarmes, notificações ou indicações de progresso, permitindo que o usuário possa estar concentrado visualmente em outras atividades (NEES; WALKER, 2012). Nos carros por exemplo, é comum que sensores de estacionamento reportem ao motorista a distâncias do veículo em relação aos obstáculos que o cercam através de avisos sonoros que aumentam a sua frequência e tonalidade a medida que a distância diminui.

Contudo, o conceito de sonificação pode ser expandido além da indicação de eventos, e pode ser utilizado também para explorar sinais de forma contínua através de uma conversão das oscilações do sinal em ondas sonoras de forma direta (CAMPO, 2007), método também conhecido como Audificação (*Audification*). Para que isso seja possível, em alguns casos é necessário que a frequência do sinal e a sua temporização sejam alterados de modo que se enquadrem dentro da faixa humana audível (NEES; WALKER, 2012). Por exemplo, a conversão de dados sísmicos em representações audíveis tem sido explorada para facilitar a classificação de eventos sísmológicos (DOMBOIS; BIRLINGHOVEN, 2002).

Motivado pela possibilidade de representação de sinais de forma audível, a metodologia proposta neste trabalho apresenta uma forma de sonificação dos dados de sensores DAS em arquivos de áudio e a implementação de uma rede neural de classificação de áudio capaz de diferenciar três tipos de fluxo em uma tubulação de testes: mono líquido, mono gás e multifásico. Utilizando como base um modelo de rede neural profunda para classificação de áudio, pré-treinado em um grande conjunto de dados.

## 1.2 OBJETIVOS

Este trabalho tem por objetivo geral apresentar uma técnica de conversão dos dados dos experimentos de escoamento coletados com um sensor acústicos distribuídos em uma representação audível. E implementar de uma rede neural profunda para a classificar os três tipos de fluxo: mono líquido, mono gás e multifásico, tendo como entrada arquivos de áudio. Para atingir o objetivo principal, as tarefas realizadas foram divididas em objetivos específicos, sendo eles:

- Analisar os dados dos experimentos tanto na forma visual quanto na forma auditiva;
- Converter os sinais do sensor DAS em arquivos de áudio;
- Utilizar um modelo de rede neural profunda para a extração de características do sinal de áudio e implementar um classificador de fluxo
- Avaliar o desempenho da classificação dos tipos de fluxo com a metodologia proposta.

## 2 REVISÃO DA LITERATURA

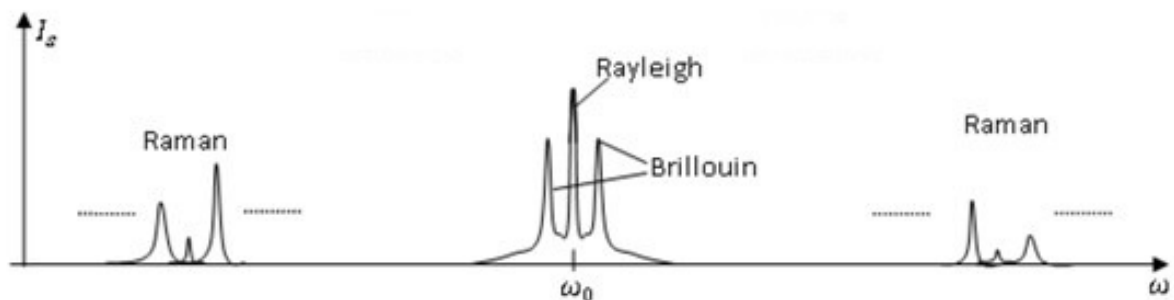
Este capítulo é dividido em duas seções e apresenta uma revisão da literatura visitando os fundamentos básicos para o trabalho apresentado por essa dissertação. A primeira seção realiza uma abordagem geral sobre sensores acústicos distribuídos, mostrando seu princípio de funcionamento dando enfoque as topologias OTDR e  $\Phi$ -OTDR. Já a segunda seção apresenta uma revisão sobre os princípios de funcionamento das redes neurais profundas do tipo MLP e CNN.

### 2.1 SENSORIAMENTO ACÚSTICO DISTRIBUÍDO

Os espalhamentos podem ser classificados em elástico e inelástico de acordo com a energia do fóton espalhado. No espalhamento elástico, como é o caso do espalhamento Rayleigh, o fóton espalhado possui a mesma energia do fóton incidente. Já no inelástico, a interação do fóton incidente com as partículas do meio gera um fóton espalhado de energia diferente, inferior ou superior ao fóton incidente, como acontece nos espalhamentos Brillouin e Raman (BOYD; MASTERS, 1960).

Na Figura 4 é possível observar que o espalhamento Rayleigh, por ser elástico, possui a mesma frequência da fonte  $\omega_0$ , já os espalhamentos Brillouin e Raman apresentam deslocamento no eixo da frequência tanto para a esquerda, frequências menores que a frequência da fonte, quanto para a direita, frequências maiores do que a frequência da fonte, sendo o espalhamento Raman o que apresenta os maiores desvios.

**Figura 4 – Espectro típico de espalhamento espontâneo de matéria de estado sólido.**



Fonte: Adaptado de Bao e Chen (2012).

Existem implementações de sensores do tipo DAS utilizando tanto os espalhamentos elásticos quanto os inelásticos. Contudo, neste capítulo serão apresentadas apenas as topologias OTDR e  $\Phi$ -OTDR que tem por princípio de funcionamento o espalhamento inelástico do tipo Rayleigh.

### 2.1.1 Espalhamento Rayleigh

O espalhamento Rayleigh recebe este nome em homenagem ao Lord Rayleigh, pioneiro na observação deste tipo de espalhamento em 1899. Este tipo de espalhamento ocorre quando o comprimento de onda incidente é muito menor que o tamanho das partículas do meio, sendo mais intenso quanto menor for o comprimento de onda (FARAHANI; GOGOLLA, 1999).

Em fibras ópticas fabricadas com sílica, o espalhamento Rayleigh é o mecanismo dominante de perdas entre os comprimentos de onda ultravioleta e infravermelho, sendo resultante das não homogeneidades relacionadas as flutuações aleatórias de densidade da fibra causadas pelo resfriamento do vidro durante seu processo de fabricação (SENIOR; JAMRO, 2009). O espalhamento ocorre em praticamente todas as direções, produzindo uma atenuação proporcional a  $1/\lambda^4$ , onde  $\lambda$  é o comprimento de onda óptico, podendo ser calculado pela Equação (1) (OLSHANSKY, 1979):

$$\alpha_{Re} = \frac{8\pi^3}{3\lambda^4} n^8 p^2 \beta_c k T_F \quad (1)$$

onde:  $\lambda$  é o comprimento de onda da luz no vácuo,  $n$  é o índice de refração da fibra,  $p$  é o coeficiente fotoelástico médio,  $\beta_c$  é a compressibilidade isotérmica na temperatura fictícia  $T_F$  e  $k$  é a constante de Boltzmann. A temperatura fictícia é definida como a temperatura em que o vidro pode atingir um estado de equilíbrio térmico e está intimamente relacionada com a temperatura de recozimento (SENIOR; JAMRO, 2009).

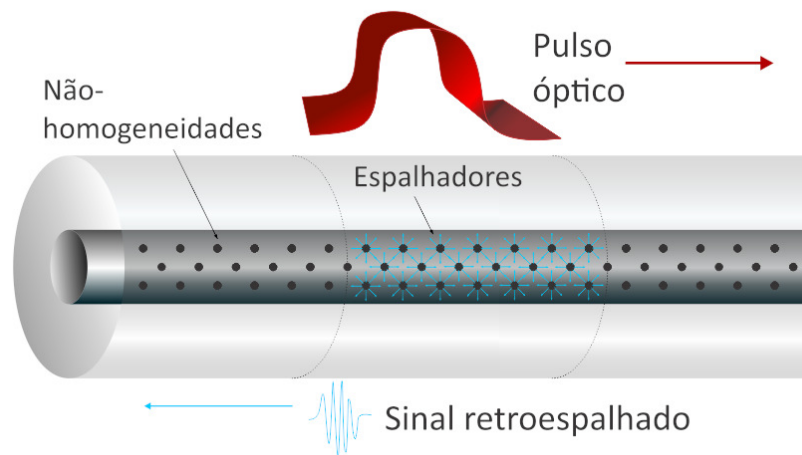
### 2.1.2 OTDR e $\Phi$ -OTDR

Proposta pela primeira vez em 1976 por Barnoski e Jensen a reflectometria óptica no domínio do tempo (OTDR – *Optical Time Domain Reflectometry*) tem por princípio o envio de um pulso ao longo da fibra óptica e medição da resposta deste pulso através da luz refletida pelo efeito de espalhamento, usualmente resultante do espalhamento Rayleigh.

Sistemas OTDR são muito eficientes na detecção da localização de rupturas e perdas ao longo da fibra óptica (AOYAMA *et al.*, 1981). A identificação da região correspondente ao espalhamento é calculada através do tempo de voo de ida e volta do pulso (RTToF - *Round Trip Time-of-Flight*) (PALMIERI; SCHENATO, 2013).

Na Figura 6 é apresentado um esquema básico de um OTDR. A luz do laser é enviada a um modulador responsável por gerar os pulsos ópticos que viajarão pela fibra sobre teste (FUT – *Fiber Under Test*). Conforme apresentado na Figura 5, ao se propagar pela fibra o pulso óptico é espalhado ao encontrar as não-homogeneidades do meio. A parcela do pulso que retorna no sentido contrário ao de propagação, chamada de sinal retroespalhado, é direcionado a um fotodetector (PhD – *Photodetector*).

**Figura 5 – Representação do espalhamento Rayleigh.**



**Fonte: immermessen (2022).**

A localização  $Z$  correspondente ao sinal retroespalhado pode ser calculada utilizando a Equação (2) da seguinte forma:

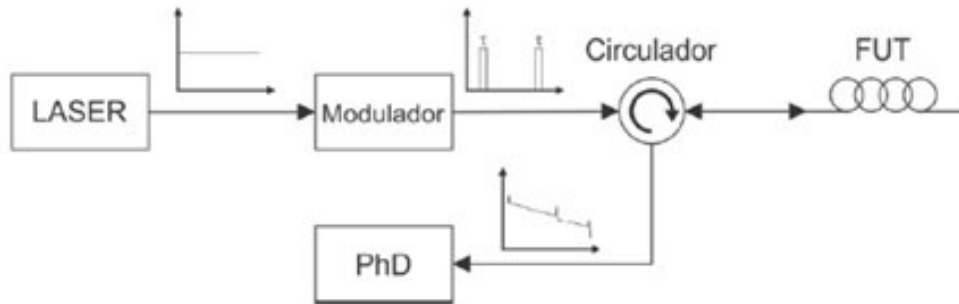
$$Z = \frac{\Delta t c}{2n_{eff}} \quad (2)$$

onde:  $\Delta t$  é o delta de tempo,  $c$  é a velocidade da luz e  $n_{eff}$  é o índice de refração efetivo da fibra óptica. O fator no denominador 2 representa o tempo de voo de emissão da luz e de seu espalhamento (BAO; CHEN, 2012).

Em sensores OTDR baseados no espalhamento Rayleigh a resolução espacial do sistema é definida pela largura do pulso, podendo ser calculada através da Equação (2). Por exemplo, um sistema com largura de pulso de 10 ns terá uma resolução de 1 m. Equipamentos comerciais usualmente recebem sua especificação de resoluções de leitura (read-out resolution), pela taxa

de leitura de seu digitalizador (BAO; CHEN, 2012), um equipamento com taxa de aquisição de 200 MHz possui um intervalo entre amostras de 1 ns correspondente a uma comprimento de 0.5m na fibra óptica.

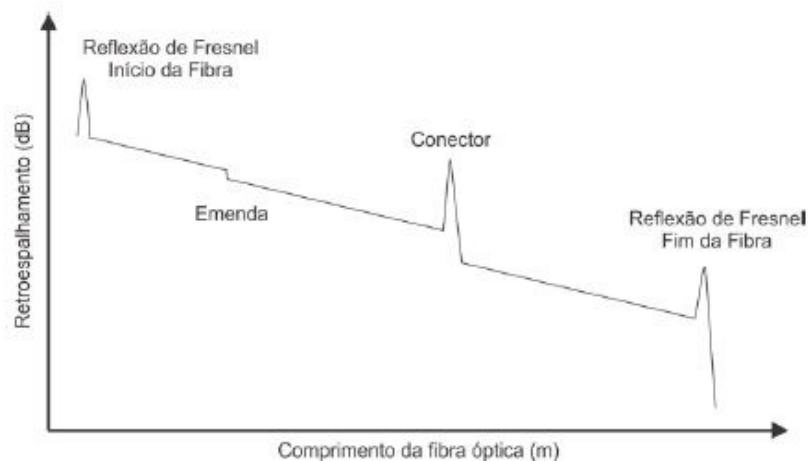
**Figura 6 – Diagrama simplificado do OTDR.**



**Fonte: Dutra (2018).**

A Figura 7 apresenta um exemplo de um perfil de resposta convencional de um OTDR, nela é possível observar a atenuação do sinal ao longo da fibra, causado principalmente pelo espalhamento Rayleigh, de aproximadamente 0,16 dB/km para o comprimento de onda de 1550 nm (AGRAWAL, 2000). O sinal também apresenta regiões com variações mais abruptas de intensidade, que normalmente são ocasionadas por curvaturas, emendas, conectores e terminações.

**Figura 7 – Sinal obtido pelo retroespalhamento Rayleigh utilizando fonte de baixa coerência.**



**Fonte: Dutra (2018).**

Além das atenuações do sinal explicadas anteriormente, o OTDR também possui uma resposta a perturbações acústicas. Estas perturbações afetam o agrupamento dos espalhadores do núcleo da fibra e podem ser de qualquer natureza, sendo as transversais e longitudinais as

de maior influência (DUTRA, 2018). As variações decorrentes das interferências construtivas e destrutivas que acontecem com maior frequência devido a coerência temporal da fonte são chamadas de Ruído Rayleigh Coerente (CRN - *Coherent Rayleigh Noise*) (RESHAK *et al.*, 2013). O CRN possibilita a geração das assinaturas da fibra e a observação de variações temporais no sinal retroespalhado decorrentes de vibrações acústicas.

Uma variação da tecnologia OTDR convencional é a reflectometria óptica no domínio do tempo sensível a fase conhecida como  $\Phi$ -OTDR. Neste modelo de implementação o sensor é projetado para determinar a variação de comprimento da fibra entre duas regiões de espalhamento (HARTOG, 2018), através da comparação do sinal retroespalhado por cada uma delas. Ao realizar este procedimento para vários pares, tem-se uma medida de deformação (*strain*) dinâmica em função do tempo e da posição ao longo da fibra. A distância entre as regiões de medição é chamada de comprimento de medida (*Gauge Length*).

Para que a comparação dos sinais retroespalhados seja precisa os sistemas  $\Phi$ -OTDR requerem um laser com baixa variação de frequência, largura de banda muito estreita (na ordem de kHz) e alta coerência temporal. A baixa variação na frequência é necessária pois a mudança de frequência do laser provoca flutuações entre os traços, ou seja, entre cada aquisição da assinatura da fibra óptica pelo  $\Phi$ -OTDR (JUAREZ; TAYLOR, 2005). Uma vez que frequências diferentes possuem velocidades de propagação diferentes, a largura do espectro óptico da fonte deve ser reduzida, quanto mais estreito o espectro maior será a coerência da fonte. Já a coerência temporal do laser diz respeito a capacidade do laser em manter a fase do sinal propagado em dois pontos distintos.

As técnicas de implementação para sensores  $\Phi$ -OTDR, podem ser divididas em dois domínios: detecção direta e detecção coerente. Na detecção direta, a comparação dos sinais retroespalhados ocorre no domínio óptico, enquanto, na detecção coerente, o sinal retroespalhado é misturado com um oscilador local resultando em um sinal elétrico de onde a fase é processada (HARTOG, 2018). As implementações de detecção coerente podem ser do tipo homódina, onde o oscilador local possui a mesma frequência do da fonte, ou heteródina, onde as frequências são diferentes.

## 2.2 REDES NEURAIIS PROFUNDAS

As redes neurais profundas (DNNs *Deep neural Networks*), que nos últimos anos demonstraram alta eficiência e acurácia na execução de tarefas como classificação, detecção e



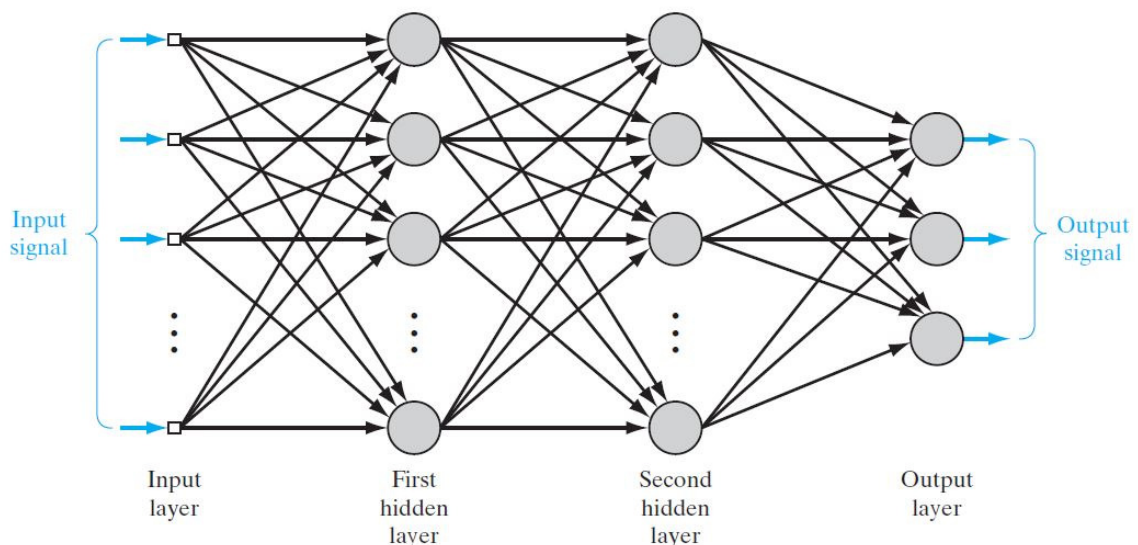
segmentação de objetos em imagens, tem se tornando cada dia mais populares e viáveis. Graças aos modelos disponibilizados com código aberto, aos bancos de dados públicos contendo milhares de arquivos pré-avaliados, avanço em ferramentas e interfaces de programação, bibliotecas voltadas ao desenvolvimento de redes neurais e o avanço e diminuição do custo das unidades de processamento gráfico (GPU – *Graphics Processing Units*) que permitiram o treinamento de modelos que muitas vezes possuem milhões de parâmetros, as redes neurais profundas têm sido utilizadas nas mais diversas áreas do conhecimento.

### 2.2.1 Redes Profundas de Alimentação Direta

As redes profundas de alimentação direta (*Deep Feed Forward Networks*), também conhecidas como *Multi-Layer Perceptron* (MLP), são constituídas por uma camada de nós de entrada (*input layer*), uma ou mais camadas escondidas contendo neurônios (*hidden layer*) e uma camada de saída (*output layer*), também composta por neurônios (XU *et al.*, 2003). E são ditas de alimentação direta (*feed forward*) pois a informação é processada em apenas um sentido sem realimentações (GOODFELLOW *et al.*, 2016).

A Figura 8 apresenta um exemplo de uma rede MLP com 2 camadas escondidas e uma camada de saída, em sua forma genérica onde as conexões entre as camadas são do tipo *fully connected*, ou seja, um neurônio (perceptron) em qualquer uma das camadas recebe o sinal de saída de todos os neurônios da camada anterior, e os sinais, ou valores de entrada, são propagados na rede da esquerda para a direita de uma camada para outra.

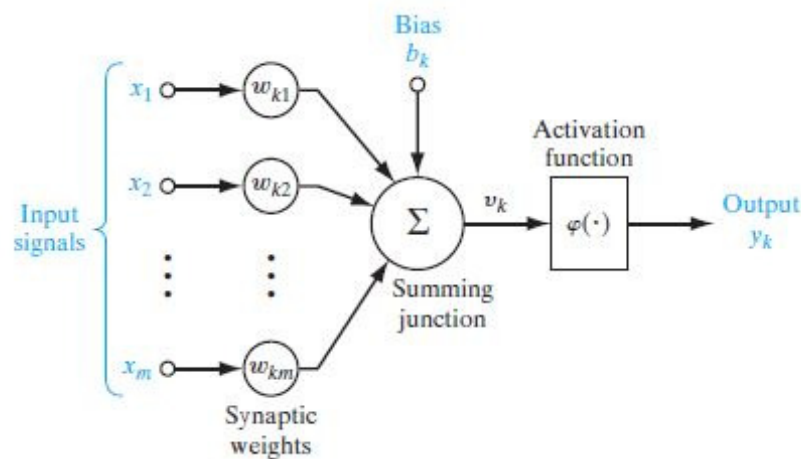
**Figura 8 – Representação da arquitetura de uma MLP com 2 camadas escondidas.**



**Fonte: Haykin (2009).**

Estas redes têm por elemento base o modelo matemático de um neurônio chamado de perceptron, proposto inicialmente por Rosenblatt (1958), com o objetivo de representar de forma probabilística o armazenamento e organização das informações no cérebro. Conforme apresentado na Figura 9 o valor de saída de cada neurônio é realizado através do somatório da multiplicação dos valores de entrada  $x$  pelos seus respectivos pesos  $w_k$  acrescido ao valor de constante  $b_k$  chamado de bias, gerando assim o valor  $v_k$ , que por sua vez é avaliado pela função de ativação (Activation function)  $\varphi$  gerando a saída  $y_k$ . Em termos matemáticos podemos modelar a resposta de um neurônio através das equações 3 e 4.

**Figura 9 – Modelo não linear de um neurônio, denominado  $k$ .**



Fonte: Haykin (2009).

$$v_k = \sum_{j=1}^m w_{kj} x_j \quad (3)$$

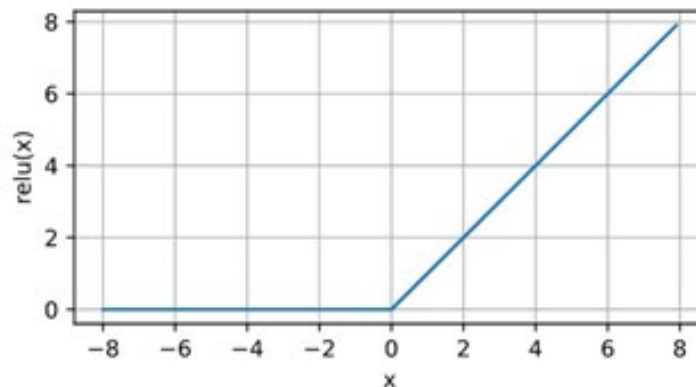
$$y_k = \varphi(v_k + b_k) \quad (4)$$

O *bias* ( $b_k$ ) tem a função de aumentar ou diminuir o valor de entrada que será passado à função de ativação, podendo ser positivo ou negativo, afetando assim o potencial de ativação (HAYKIN, 2009). As funções de ativação determinam o valor de saída do neurônio, permitindo um controle do fluxo de informação decidindo se ele será ativado ou não (ZHANG *et al.*, 2021). Ela também representa uma característica muito importante das redes neurais que é a inserção de não linearidades, as quais permitem a resolução de problemas que não são linearmente separáveis. Existem diversas funções de ativação as quais podem ser escolhidas de acordo com o tipo de saída ou comportamento desejado da rede neural, algumas das mais utilizadas são: *ReLU* e *Sigmoid*.

A função de unidades lineares retificadas (*ReLU - Rectified Linear Units*) está entre as mais populares, devido a facilidade de implementação. Conforme apresentado na Figura 10 e na Equação 5 a saída da função representa o valor máximo entre zero e a entrada  $x$ . Ou seja, a função de ativação mantém os valores positivos e zera os valores negativos, de forma que: ou valor de entrada de um neurônio é transmitido para a saída ou é anulado. Outro motivo em utilizar a função *ReLU* é a facilidade de computar sua derivada, pois quando a entrada é negativa o valor da derivada é 0 e para entradas positivas o valor da derivada é igual a 1. (ZHANG *et al.*, 2021).

$$ReLU(x) = \max(0, x) \quad (5)$$

**Figura 10 – Gráfico da função ReLu.**

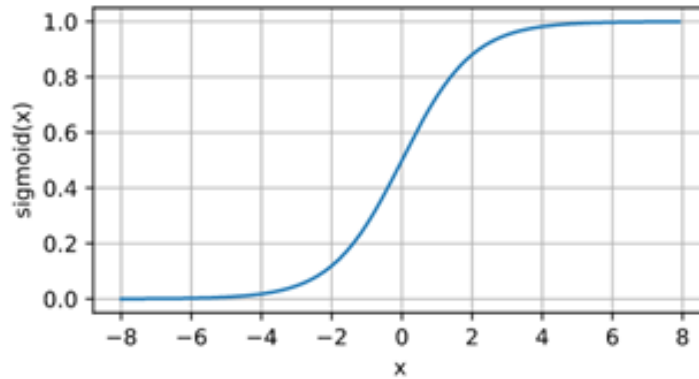


**Fonte: Zhang *et al.* (2021).**

Já função *Sigmoide* restringe seu argumento que está no domínio dos números reais, de -infinito a +infinito, entre os valores 0 e 1, conforme apresentado na Figura 11 e descrito pela Equação 6. A sua alta aderência nas aplicações se deve ao fato de ser uma função suave, que é capaz de ativar e desativar as saídas dos neurônios e é derivável. A função *Sigmoide* é amplamente utilizada como funções de ativação na camada de saída, quando deseja-se interpretar as saídas como probabilidades para problemas de classificação binária (ZHANG *et al.*, 2021). No entanto, usualmente nas camadas ocultas ela é substituída pela *ReLU*, por ser mais simples e facilitar o treinamento. Em muitos casos essa substituição está relacionada aos desafios que a função *Sigmoide* apresenta para a otimização (LECUN *et al.*, 1998), uma vez que seu gradiente desaparece para grandes argumentos positivos e negativos.

$$Sigmoide(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

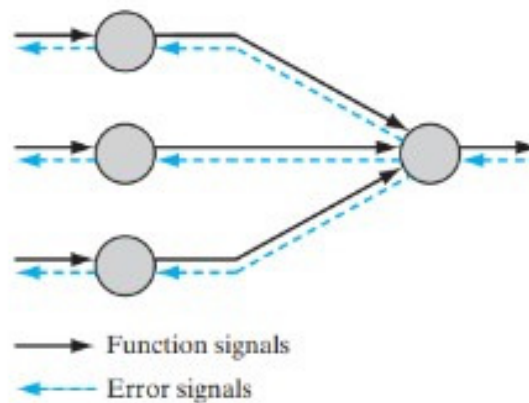
Figura 11 – Gráfico da função Sigmoide.



Fonte: Zhang *et al.* (2021).

Cada neurônio da rede possui dois fluxos de informação os quais contribuem em momentos diferentes para o funcionamento do sistema. Como podemos observar na Figura 12, o primeiro fluxo, da esquerda para a direita, é a passagem dos sinais processados para a próxima camada (*function signals*). E o outro fluxo segue o sentido contrário e consiste na propagação reversa do erro da saída (*error signals*) esperada com relação a saída obtida (HAYKIN, 2009).

Figura 12 – Ilustração dos dois fluxos básicos de informação em uma rede MLP.



Fonte: Zhang *et al.* (2021).

A propagação do erro no sentido contrário serve para a realização dos ajustes dos pesos da rede, através da técnica de treinamento conhecida como propagação reversa (*backpropagation*). A implementação do algoritmo de *backpropagation* permite que ocorra um mapeamento das relações entre entradas e saídas, de forma a desenvolver o aprendizado da rede com relação a resposta desejada da rede. O ajuste do pesos para o algoritmo de *backpropagation* é realizado através da implementação da descida do gradiente, que consiste em encontrar as derivadas parciais de uma função de aproximação  $F(w,x)$ , realizada para todos os elementos de peso ajustável  $w$  para um vetor e entradas  $x$  (HAYKIN, 2009).

O treinamento das redes MLP ocorre de forma supervisionada, ou seja, durante a execução, são apresentados à rede os conjuntos dados de entradas conhecidas (*dataset*)  $x$  e suas respectivas saídas desejadas  $y$ . Assim, pode-se calcular o erro das saídas obtidas com relação as desejadas e realizar o ajuste dos pesos da rede utilizando o *backpropagation*, buscando a minimizá-lo. (ZURADA, 1992).

Nas implementações mais comuns de problemas de classificação utilizando MLPs o número de neurônios da camada de saída é igual ao número de classes nas quais a base de dados foi dividida. Durante o treinamento a rede irá buscar maximizar o valor da saída respectiva a classe do dado de entrada. Uma vez que os valores das saídas da rede não estão restringidos a uma faixa de valores específica, faz-se necessário utilizar uma função de normalização que permita eleger a classe ganhadora. Para realizar esta normalização as funções mais utilizadas são *Argmax* e *Softmax*.

A função *Argmax* é utilizada para encontrar o maior valor na camada de saída da rede, para esta saída é atribuído o valor 1 e as demais o valor 0, sinalizando que esta é a classe prevista pela rede para os dados presentes na entrada. No entanto a função *Argmax* não reflete a noção de precisão ou nível de confiança na classe escolhida, para que seja possível tal análise a função *Softmax* pode ser utilizada.

Ao utilizar a função *Softmax* os valores de saída são representados em um intervalo entre 0 e 1, sendo que a soma dos valores das saídas deve ser igual a 1. O cálculo dos valores de saída utilizando a função *Softmax* são realizados da seguinte forma (ZHANG *et al.*, 2021):

$$Softmax(y_k) = \frac{e^{y_k}}{\sum_j e^{y_j}} \quad (7)$$

onde o índice  $k$  é relativo a saída desejada e o  $j$  representa o índice de todas as saídas para realização do somatório.

## 2.2.2 Redes Neurais Convolucionais

As redes neurais convolucionais (CNNs - *Convolutional Neural Networks*) têm apresentado grande sucesso em tarefas de classificação de objetos em dados bidimensionais, especialmente trabalhando com imagens. O nome "rede neural convolucional" indica que a rede emprega uma operação de convolução no lugar da multiplicação matricial em pelo menos uma de suas camadas (GOODFELLOW *et al.*, 2016). CNNs tendem a ser computacionalmente eficientes,

tanto por possuírem menos parâmetros do que arquiteturas totalmente conectadas, quanto pela facilidade de paralelizar as operações de convolução entre os núcleos das GPUs (CHETLUR *et al.*, 2014).

A função de convolução calcula a sobreposição entre 2 vetores ou matrizes, aqui denotados por  $f$  e  $g$ . O processo consiste em inverter uma das matrizes e deslocá-la sobre a outra, de cima para baixo e da esquerda para a direita, calculando a soma dos produtos de suas posições sobrepostas. Para o caso de dados bidimensionais discretos, a função de convolução pode ser calculada de com a Equação (8) (ZHANG *et al.*, 2021):

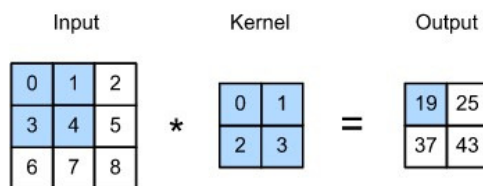
$$(f * g)(i, j) = \sum_a \sum_b f(a, b)g(i - a, j - b) \quad (8)$$

Onde  $(a, b)$  representam os índices da matriz  $f$  e  $(i - a, j - b)$  os índices da matriz  $g$ .

Imagens podem possuir um ou mais canais representados por matrizes bidimensionais. Estes canais contém os valores de cada pixel da imagem de acordo com uma característica. Em imagens monocromáticas cada pixel representa um valor de intensidade de cor, usualmente em uma escala de tons de cinza. Já em imagens coloridas do tipo RGB (*Red, Green and Blue*) existem três canais, um para o vermelho, outro para o verde e outro para o azul, cada um representando os valores de intensidade dos *pixels* da respectiva cor.

Independentemente do número de canais, o funcionamento de uma rede convolucional está baseado na utilização de núcleos (*kernels*). Como apresentado na Figura 13, cada camada convolucional correlaciona a entrada e o seu *kernel* e em seguida adiciona um valor de (*bias*). Também é possível notar que a saída do exemplo possui dimensões menores que a imagem de entrada, isso ocorre quando o número de linhas e colunas do *kernel* é maior que 1. Caso seja necessário manter as dimensões de entrada é possível utilizar uma técnica de adição de zeros ao redor das bordas da imagem de entrada, conhecida como *zero padding* (ZHANG *et al.*, 2021).

**Figura 13 – Operação de correlação cruzada bidimensional.**



Fonte: Zhang *et al.* (2021).

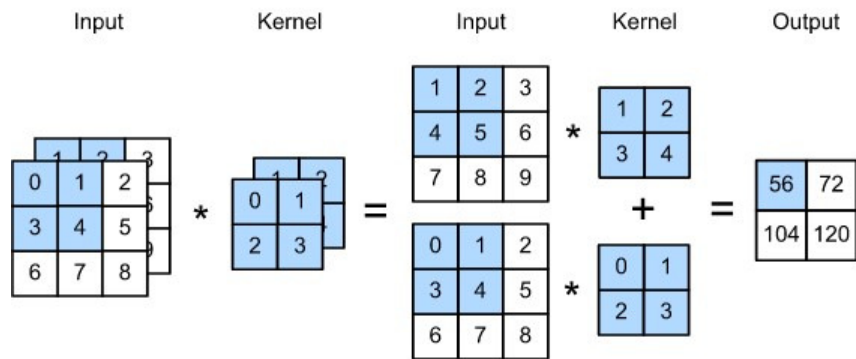
As porções sombreadas são utilizadas para o cálculo do primeiro elemento de saída:

$$0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$$

Como dito anteriormente, uma imagem pode possuir um ou mais canais. Sendo assim, o *kernel* a ser aplicado sobre essa imagem deve possuir o mesmo número de canais. Como podemos observar na Figura 14, para uma camada convolucional que tem uma entrada de 2 canais de tamanho  $3 \times 3 \times 2$  os seus *kernels* deverão ter a mesma quantidade de canais da entrada,  $2 \times 2 \times 3$ . Contudo, ao executar a convolução deste *kernel* com a entrada, a dimensão da saída resultante será de  $2 \times 2 \times 1$ , uma vez que o resultado das operações de convolução são somados gerando uma única saída. Neste exemplo o primeiro valor de saída (56) foi calculado da seguinte forma:

$$(1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 5) + (0 \times 0 + 1 \times 1 + 2 \times 3 + 3 \times 4) = 56$$

**Figura 14 – Cálculo de correlação cruzada com 2 canais de entrada.**

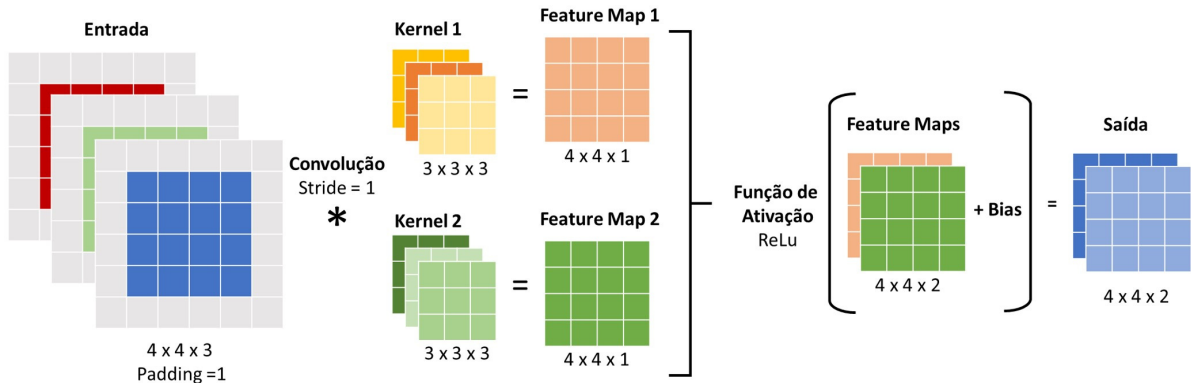


Fonte: Zhang *et al.* (2021).

Uma camada convolucional pode possuir mais de um *kernel*, resultando em uma saída com um número de canais igual ao número de *kernels*, o resultado da convolução da entrada e do *kernel* recebe o nome de mapa de características (*feature maps*) (ZHANG *et al.*, 2021), pois representa a extração de características através dos *kernels* que foram aprendidos durante o treinamento. Para gerar a saída que será passada para a próxima camada, os *feature maps* são somados a um valor de *bias* e avaliados por uma função de ativação, conforme apresentado na Figura 15.

Assim como nas MLPs, o treinamento das camadas convolucionais é realizado de forma supervisionada utilizando a técnica de *backpropagation*, em cada iteração o valor do erro entre a saída obtida e a saída desejada é calculado através de uma função de penalização. Na sequência, o ajuste dos valores para cada *kernel* e *bias* é realizado através do método de descida do gradiente, usualmente no início do treinamento ambos recebem valores aleatórios (GOODFELLOW *et al.*, 2016).

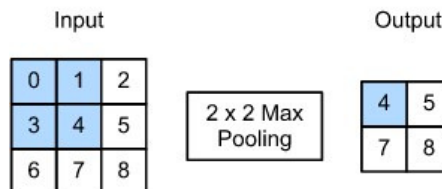
**Figura 15 – Diagrama de funcionamento de uma camada convolucional com entrada 4x4x2, utilizando dois kernels de 3x3x3, com Padding = 1 e Stride = 1.**



Fonte: Autoria própria.

Além de camadas que executam a operação de convolução é comum encontrar nas arquiteturas de CNNs camadas de agrupamento (*pooling*). A operação *pooling* é utilizada para reduzir as dimensões de uma matriz realizando uma subamostragem (*downsampling*) segundo um critério de agrupamento, resultando em imagens de menor resolução. Camadas de *pooling* servem para mitigar a sensibilidade das camadas convolucionais e para diminuir a carga computacional. (ZHANG *et al.*, 2021)

**Figura 16 – Max-pooling com uma janela de  $2 \times 2$ . As porções sombreadas são utilizadas para o cálculo do primeiro elemento de saída, da seguinte forma:  $\max(0, 1, 3, 4) = 4$ .**



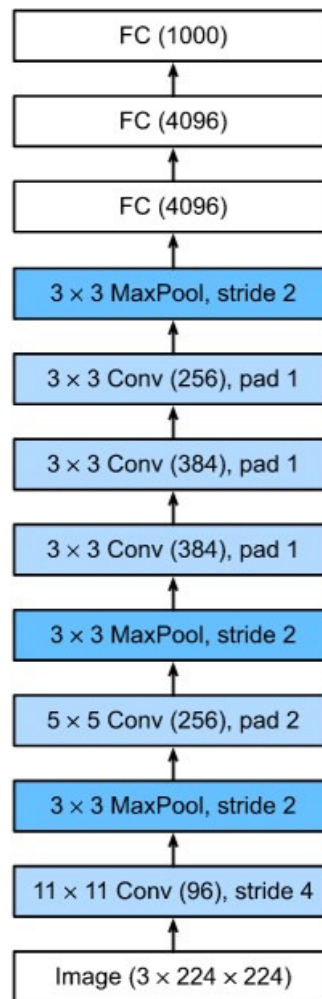
Fonte: Zhang *et al.* (2021).

De maneira similar a convolução, a operação de *pooling* é baseada em uma janela de tamanho  $m \times n$  que desliza sobre a matriz original de cima para baixo e da esquerda para a direita. Ao percorrer a imagem, os *pixels* desta janela são agrupados em um único *pixel* conforme um critério. Os métodos mais utilizados são: *Max-pooling*, apresentado na Figura 16, onde o *pixel* resultante é igual ao *pixel* de maior valor dentro da janela, e *Average pooling*, onde o *pixel* resultante é a média do valor dos *pixels* dentro da janela. O passo de deslocamento da janela de *pooling*, também conhecido como *stride*, pode ser ajustado para um valor maior do que 1, possibilitando reduzir ainda mais as dimensões da matriz de saída sem alterar o tamanho da janela de *pooling*.



Arquiteturas de CNNs utilizam uma combinação de camadas do tipo convolucionais, *polling* e *fully connected* (FC). Um exemplo é a arquitetura da rede AlexNet, apresentada na Figura 17. Proposta em 2012 por (KRIZHEVSKY *et al.*, 2012) a AlexNet quebrou paradigmas e foi a primeira a demonstrar o verdadeiro potencial das redes neurais profundas ao vencer com uma grande margem o desafio de reconhecimento visual em larga escala ImageNet (DENG *et al.*, 2009) em 2012(RUSSAKOVSKY *et al.*, 2013).

**Figura 17 – Camadas da rede AlexNet.**



**Fonte: Zhang *et al.* (2021).**

### 2.2.3 Transferência de Aprendizado

A transferência de aprendizado ou *transfer learning* é uma técnica onde aproveita-se de um modelo de rede neural profunda pré-treinado, para gerar um novo modelo adaptado a uma nova tarefa. Por exemplo, é possível utilizar um modelo que foi treinado em um grande banco de dados contendo milhares de imagens pertencentes a diversas classes, como: casas, animais,

peçoas, carros entre outros, e adaptá-lo para classificar espécies de pássaros (DONAHUE *et al.*, 2014).

A utilização da transferência de aprendizado reduz o tempo de treinamento e melhora o desempenho do modelo. Ela também é muito eficiente nos casos em que o conjunto de dados disponível é pequeno e não suficiente para treinar um modelo com milhares ou milhões de parâmetros. Dentre as técnicas mais utilizadas para realizar a transferência de aprendizado em redes neurais, destacam-se as técnicas de ajuste fino e extração de recursos (LI; HOIEM, 2016).

#### 2.2.4 Ajuste fino

Ajuste fino é um processo no qual um modelo pré-treinado é adaptado através da continuidade do seu treinamento utilizando o novo conjunto de dados, adaptando o tamanho de sua saída de acordo com o número de classes da nova tarefa (BARCELLOS *et al.*, 2019). Neste processo, podem ser ajustadas apenas algumas camadas do modelo de acordo com o nível de informação ou características aprendidas pela camada. Por exemplo, as camadas inferiores do modelo pré-treinado aprenderam a detectar bordas, linhas e padrões simples, enquanto as camadas mais altas do modelo aprenderam a detectar padrões mais complexos, como formas e contornos.

A escolha de quais camadas ajustar depende do tamanho do conjunto de dados. Tendo um grande conjunto de dados, é possível ajustar todas as camadas do modelo pré-treinado. No entanto, se o conjunto de dados for pequeno, eles serão capazes de ajustar apenas as camadas mais altas do modelo pré-treinado.

#### 2.2.5 Extração de Recursos

A extração de recursos é um processo no qual um modelo pré-treinado em uma grande base de dados é utilizado para extrair conjuntos de características de uma nova base de dados, possibilitando que estas sejam utilizadas no treinamento de um classificador customizado para uma nova tarefa (DONAHUE *et al.*, 2014). A extração de recursos é útil quando o conjunto de dados disponíveis para treinamento é pequeno.

Existem muitas maneiras de extrair recursos de dados, e a escolha do extrator de recursos dependerá da natureza dos dados e da tarefa em questão. No âmbito das redes neurais convolucionais, uma das formas mais comuns de extração de recursos é através de vetores de

associações. As associações de dados, também chamadas de *embeddings*, são expressas na forma de um vetor contínuo contendo valores que representam variáveis discretas de categorias aprendidas através dos dados de treinamento.

O modelo resultante desta técnica contém as camadas do classificador que foi treinado para a nova tarefa conectadas à camada de extração de *embeddings* do modelo pré-treinado. Uma vez que o modelo pré-treinado é apenas utilizado para extração do vetor de *embeddings*, sua arquitetura e pesos originais são mantidos.

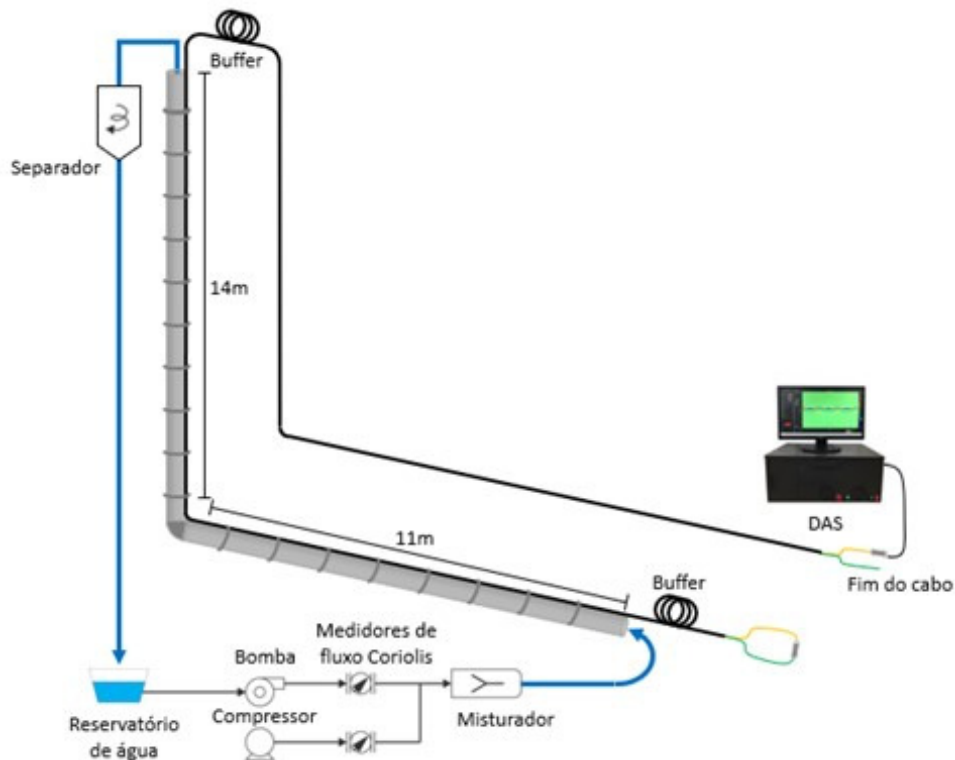
### 3 MATERIAIS E MÉTODOS

Neste capítulo serão apresentadas as instalações, equipamentos e métodos utilizados para a implementação da classificação dos sinais do sensor DAS. Inicialmente será apresentado o experimento utilizado para a aquisição dos dados e na sequência a metodologia de processamento e classificação.

#### 3.1 INSTALAÇÕES

Os dados utilizados neste trabalho foram capturados através de experimentos realizados no laboratório de escoamento do Núcleo de Escoamento Multifásico (NUEM), da UTFPR. As instalações do laboratório, representadas na Figura 18, oferecem uma tubulação de testes construída por uma seção vertical de 14 m e uma seção horizontal de 11 m, construída com tubos de acrílico com diâmetro interno de 26 mm, as duas seções são unidas por uma junção do tipo cotovelo de 90°.

**Figura 18 – Esquema das instalações do laboratório NUEM.**

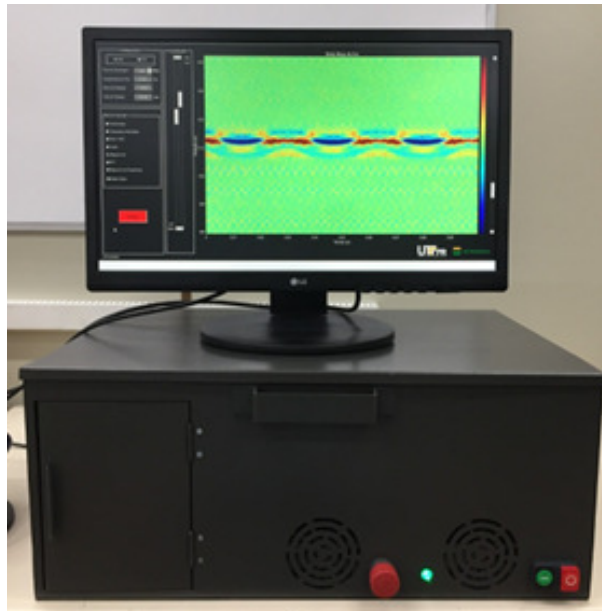


Fonte: Autoria própria.

O fluxo dos líquidos e gases é gerado através de uma bomba que extrai a água de um reservatório e um compressor. Ambos permitem o controle da velocidade de fase do líquido ( $J_l$ ) e gás ( $J_g$ ) de forma independente as quais podem ser estimadas através de dois medidores de fluxo de Coriolis, os quais medem a vazão mássica para líquido e gás. Na sequência os fluxos de líquido e gás passam por um misturador e entram na tubulação de forma ascendente. Ao chegar ao final da tubulação, na parte mais alta da instalação, existe um separador para remoção do ar antes de retornar à água para o reservatório.

O monitoramento da tubulação foi realizado utilizando um sistema de sensoriamento acústico distribuído baseado na tecnologia  $\Phi$ -OTDR de detecção heteródina, desenvolvido no Laboratório de Tecnologia de Sistemas de Sensores UTFPR (LTSS) dentro dos projetos de pesquisa em conjunto com a Petrobras 2015/00083-9 e 2017/00475-0. Conforme apresentado na Figura 19, o sistema é composto por um gabinete e um monitor. O gabinete provê as conexões ópticas e elétricas e comporta os circuitos ópticos, sistema de amostragem e unidade de processamento. Já o monitor é utilizado para apresentar a interface com usuário pela qual é possível realizar a configuração de parâmetros de aquisição do sistema como: largura de pulso do sensor e comprimento de medida e visualizar as aquisições dos experimentos realizados.

**Figura 19 – Sensor DAS  $\Phi$ -OTDR desenvolvido pelo Laboratório de Tecnologia de Sistemas de Sensores UTFPR (LTSS).**



**Fonte: Autoria própria**

Como elemento sensor foi utilizado um cabo de fibra ótica de inha OpticLan ABNT da Furukawa, similar ao apresentado na Figura 20, contendo duas fibras monomodo imersas em gel

e protegidas por um tubo termoplástico. O cabo foi acoplado nas seções vertical e horizontal da tubulação utilizando cintas de nylon. Conforme apresentado anteriormente na Figura 18, em uma das extremidades do cabo as duas fibras são conectadas possibilitando que ambas sejam utilizadas como elemento sensor. Na outra extremidade, próxima ao equipamento, uma das fibras é conectada ao gabinete e a outra representa a terminação do elemento sensor.

**Figura 20 – Cabo Furukawa OpticLan ABNT.**



**Fonte: Furukawa (2022)**

### 3.2 EXPERIMENTOS

Os experimentos utilizados neste trabalho foram fornecidos pelos integrantes Laboratório de Tecnologia de Sistemas de Sensores UTFPR (LTSS). Os dados foram coletados dentro dos projetos de pesquisa em conjunto com a Petrobras 2015/00083-9 e 2017/00475-0, em atividades anteriores a execução do trabalho apresentado nesta dissertação e sem o intuito específico de serem utilizados para o treinamento de redes neurais.

A execução dos experimentos foi feita com a tubulação instrumentada conforme apresentado na seção 3.1 utilizando variações da velocidade superficial de líquido  $J_l$  e gás  $J_g$  e diferentes valores de taxa de repetição, largura de pulso e duração. Conforme apresentado na Tabela 1, os experimentos contendo apenas líquido fluindo pela tubulação são denominados de Mono Líquido, os que contém apenas gás são chamados Mono Gás e aqueles com a combinação de líquido e gás são chamados de Multifásico. Os dados foram adquiridos para todo o comprimento da fibra de testes utilizando taxas de repetição de 10 kHz e 40 kHz, largura de pulso de 10 ns e 30 ns e *Gauge Length* de 1,5 vezes o comprimento ocupado pelo pulso na fibra.

**Tabela 1 – Relação dos experimentos utilizados para o desenvolvimento do trabalho.**

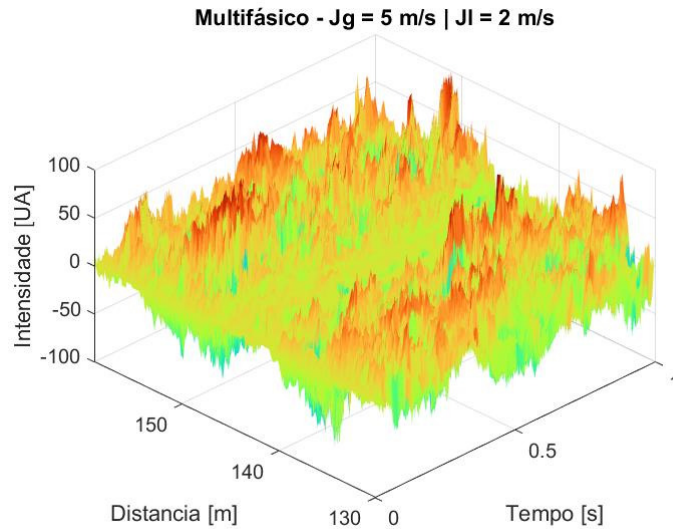
Experimento	Tipo de Fluxo	$Jl$	$Jg$	Taxa de Rep.	Larg. de pulso	Duração
		[m/s]	[m/s]	[Hz]	[ns]	[s]
1	Mono Gás	0,00	20,00	40000	10	5
2	Mono Gás	0,00	5,00	10000	30	10
3	Mono Gás	0,00	10,00	10000	30	10
4	Mono Gás	0,00	15,00	10000	30	10
5	Mono Gás	0,00	20,00	10000	30	10
6	Mono Gás	0,00	25,00	10000	30	10
7	Mono Líquido	1,00	0,00	40000	10	5
8	Mono Líquido	2,00	0,00	40000	10	5
9	Mono Líquido	0,50	0,00	10000	30	22
10	Mono Líquido	0,75	0,00	10000	30	22
11	Mono Líquido	1,00	0,00	10000	30	22
12	Mono Líquido	1,25	0,00	10000	30	22
13	Mono Líquido	1,50	0,00	10000	30	22
14	Mono Líquido	1,75	0,00	10000	30	22
15	Mono Líquido	2,00	0,00	10000	30	10
16	Mono Líquido	2,25	0,00	10000	30	10
17	Mono Líquido	2,50	0,00	10000	30	10
18	Mono Líquido	2,75	0,00	10000	30	10
19	Mono Líquido	3,00	0,00	10000	30	10
20	Multifásico	1,00	1,00	40000	10	5
21	Multifásico	1,00	1,00	40000	10	5
22	Multifásico	2,00	5,00	40000	10	5
23	Multifásico	1,00	9,50	40000	10	5
24	Multifásico	2,00	1,00	5000	10	20
25	Multifásico	1,00	3,00	5000	10	20
26	Multifásico	2,00	3,00	5000	10	20
27	Multifásico	1,00	5,00	5000	10	20

**Fonte: Autoria própria.**

Os dados capturados foram entregues na forma de dois vetores e uma matriz. Um dos vetores representa a distância relativa da leitura ao longo da fibra, expresso em metros, e o outro representa o tempo de repetição relativo a cada ciclo de leitura para todo o comprimento da fibra, expresso em segundos. Já a matriz contém os valores de variação de intensidade de fase em função da distância e do tempo de repetição do ciclo de leitura. A Figura 21 mostra um exemplo de uma representação dos dados na forma de um gráfico de superfície para um experimento com velocidade de fase do líquido ( $Jl$ ) igual a 5 m/s e velocidade de fase do gás ( $Jg$ ) igual a 2 m/s.

Além da representação da intensidade da variação de fase do sinal no plano distância-tempo, existem outras visualizações que podem ser exploradas, tais como a utilização da transformada de Fourier para expressar as intensidades do sinal em diferentes frequências tanto na dimensão espacial quanto temporal.

**Figura 21 – Representação em superfície do sinal de um experimento multifásico.**



### 3.3 IMPLEMENTAÇÃO DA REDE NEURAL

A rede neural de classificação de fluxo foi implementada utilizando a técnica de transferência de aprendizado através da extração de recursos, onde, como explicado anteriormente, um modelo pré-treinado é utilizado para extrair vetores de recursos dos dados da base de interesse. Estes vetores são então utilizados para treinar um classificador customizado para a tarefa desejada. Como base para o desenvolvimento do classificador de fluxo um modelo pré-treinado da rede profunda de classificação de áudio YAMNet foi utilizado na extração e recursos dos dados.

Desenvolvida por Hershey *et al.* (2017), a YAMNet é uma rede neural profunda, implementada com base na arquitetura de convolução separável em profundidade MobileNetV1 proposta por Howard *et al.* (2017), capaz de classificar eventos de áudio em 521 categorias da ontologia conjunto de dados AudioSet.

O conjunto de dados AudioSet foi produzido com aproximadamente dois milhões de trechos de áudios de 10 s extraídos de vídeos do YouTube e anotados manualmente segundo uma ontologia composta por diversas camadas. A primeira camada por exemplo, divide os sons nas categorias: sons humanos, sons de objetos, sons de animais, sons naturais, música, sons de fontes ambíguas e sons ambientes e de fundo (GEMMEKE *et al.*, 2017).

As entradas de áudio para a rede YAMNet devem ser matrizes unidimensionais contendo os dados de áudio do tipo mono, normalizados entre os limites +1 e -1 e amostrados a uma taxa de 16 kHz. Internamente o modelo transforma os dados de áudio em um espectrograma na escala



mel de 125 Hz a 7500 Hz com 64 bandas no eixo vertical.

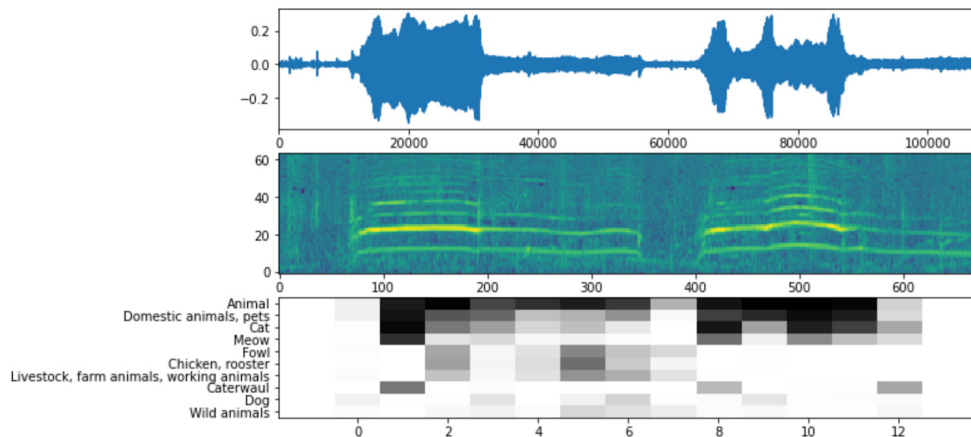
A escala mel busca reproduzir a distância entre tons que a audição humana geralmente percebe ser equidistante um do outro. O nome mel deriva de melodia e indica que a escala é baseada na comparação entre os tons (SIEGEL, 1965). Na escala mel o espectrograma é representado em um conjunto de bandas a partir de uma frequência base. Estas bandas possuem uma frequência central e uma largura que diminui pela metade à medida que a frequência aumenta. A conversão de um valor de frequência linear para a escala mel pode ser realizado através da seguinte fórmula:

$$f_{mel} = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right) \quad (9)$$

Onde  $f$  representa o valor da frequência linear e  $f_{mel}$  a frequência mel equivalente.

Após a geração do espectrograma mel, são extraídos dele quadros de 0,96 s de duração com 50% de sobreposição ao quadro anterior, ou seja, cada quadro é adquirido utilizando uma janela de 96s se deslocando a um passo de 0,48 s. Cada quadro tem dimensão 96x64 relativos as 96 amostras de 10 ms do eixo horizontal e as 64 bandas da escala mel do eixo vertical. Estes quadros são então passados para uma rede MobileNetV1 produzindo um vetor de *embeddings* de dimensão 1024 ao final das camadas convolucionais, este vetor é então passado para as camadas de classificação a qual possui uma saída com 512 valores relativos à pontuação de cada uma das classes.

**Figura 22 – Fluxo processamento de uma entrada de áudio pela rede YAMNet.**



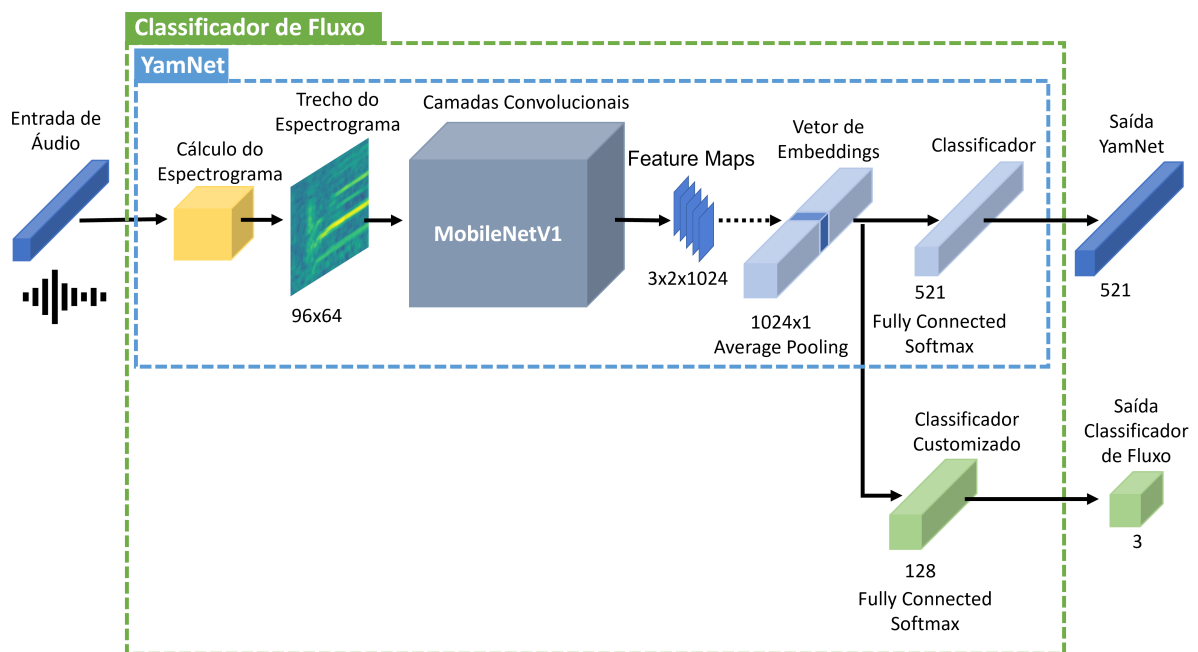
**Fonte: (MARTINS, 2021).**

A Figura 22 apresenta um exemplo de conversão de um sinal de áudio em um espectrograma mel e as 10 classes com maior pontuação geral previstas pela YAMNet. Na parte superior da figura é apresentado o gráfico de um sinal de áudio no domínio do tempo discreto, amostrado a uma frequência de 16 kHz e com duração de 6,73 s (117.680 amostras). A parte central apresenta o espectrograma na escala mel, relativo ao áudio de entrada, do qual foram retirados quadros de 0,96 s a cada 0,48 s. Na parte inferior é apresentado um gráfico em escala de cinza representado os scores obtidos para as 10 classes com maior pontuação geral, para cada trecho classificado.

Além de fornecer como saída o vetor de previsões para as 521 classes de áudio, a YAMNet também disponibiliza como saídas o seu vetor de *embeddings* e o espectrograma mel utilizados. Sendo assim, os dados dos experimentos foram convertidos em arquivos de áudio e submetidos a rede YAMNet para extração dos vetores de *embeddings*.

O classificador customizado foi implementado contendo uma camada de entrada de tamanho 1024, mesmo tamanho do vetor de *embeddings* da rede YAMNet, uma camada fully connected de tamanho 128 com função de ativação *Softmax* e uma camada de saída de tamanho 3, para representar os valores de previsão da rede para cada classe de fluxo. A arquitetura final do modelo de classificação de fluxo contendo tanto a YAMNet quanto a nova rede de classificação é apresentada na Figura 23.

**Figura 23 – Arquitetura detalhada do Classificador de fluxo.**



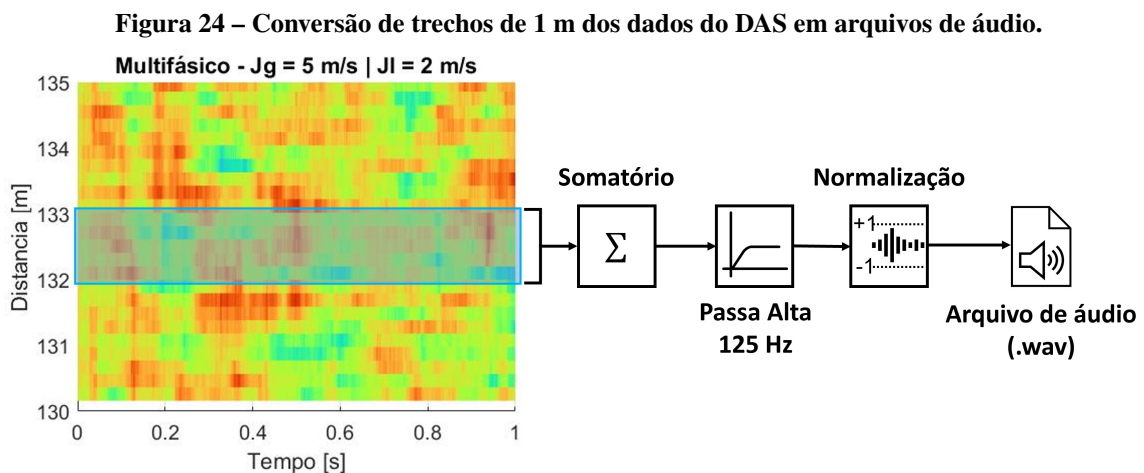
**Fonte: Autoria própria.**

### 3.3.1 Conversão dos Dados em Áudio

A preparação dos conjuntos de dados de treinamento e teste utilizados no treinamento e avaliação do classificador de fluxo foi realizada no software MatLab. Primeiramente, para facilitar o processamento dos dados, o tamanho dos arquivos foi reduzido. Selecionando apenas os trechos da fibra que estavam acoplados às seções horizontal e vertical do tubo e truncando o tempo de duração em 5 s.

Na sequência, os dados de cada trecho de interesse foram convertidos em arquivos de áudio, utilizando a metodologia apresentada na Figura 24. Primeiramente os dados foram divididos em trechos de 1 m ao longo do eixo da distância e então somados para cada ciclo de leitura ao longo do tempo gerando um vetor unidimensional. A escolha da distância de 1 m foi tomada com base na menor largura de pulso utilizada de 10 ns, que ocupa 1,01 m de comprimento ao se propagar pela fibra, conforme a Equação 2.

Em seguida, o vetor unidimensional passa por um filtro passa-alta de segunda ordem do tipo Butterworth, com uma frequência de corte igual a 125 Hz, com o objetivo de eliminar as componentes contínuas do sinal também de realçar a faixa de frequência utilizada pela rede YAMNet na geração do espectrograma mel, que vai de 125 Hz a 7500 Hz, conforme apresentado na seção 3.3. Por fim, os valores do vetor filtrado são normalizados entre o intervalo  $-1$  e  $+1$ , antes de serem salvos em um arquivo de áudio.



Fonte: Autoria própria.

Para serem utilizados pelo processo de treinamento, os arquivos de áudio foram separados em conjuntos de dados juntamente com um arquivo de metadados no formato CSV. O arquivo de metadados é utilizado para auxiliar na divisão das bases de treinamento e testes, contendo

uma lista relacionando os nomes dos arquivos de áudio, as suas classes e a qual conjunto os dados pertencem, treinamento ou teste.

### 3.3.2 Treinamento da Rede Neural

O treinamento da rede neural foi feito baseado em um tutorial de implementação de transferência de aprendizado para classificação de áudio com a rede YAMNet (disponível em: [https://www.tensorflow.org/tutorials/audio/transfer\\_learning\\_audio](https://www.tensorflow.org/tutorials/audio/transfer_learning_audio)). Nele é fornecido um arquivo exemplo, na forma de um bloco de anotações do tipo Colab Notebook, contendo as instruções e o código Python para a implementação do modelo. A execução do bloco de anotações é realizada através da plataforma web Google Colaboratory (2022), também conhecida como Colab.

A plataforma Colab disponibiliza um ambiente remoto de desenvolvimento (*runtime*) no qual é possível se conectar através do navegador e realizar a execução de programas na linguagem Python escritos em um bloco de anotações do tipo Colab Notebook. Os Colab Notebooks são baseados no formato Jupyter Notebook, estes blocos de anotações consistem em arquivos que combinam trechos informativos contendo texto, imagens e equações com trechos de código Python (JUPYTER, 2022), sendo muito úteis para organizar fluxos de trabalho alternando entre explicação e código.

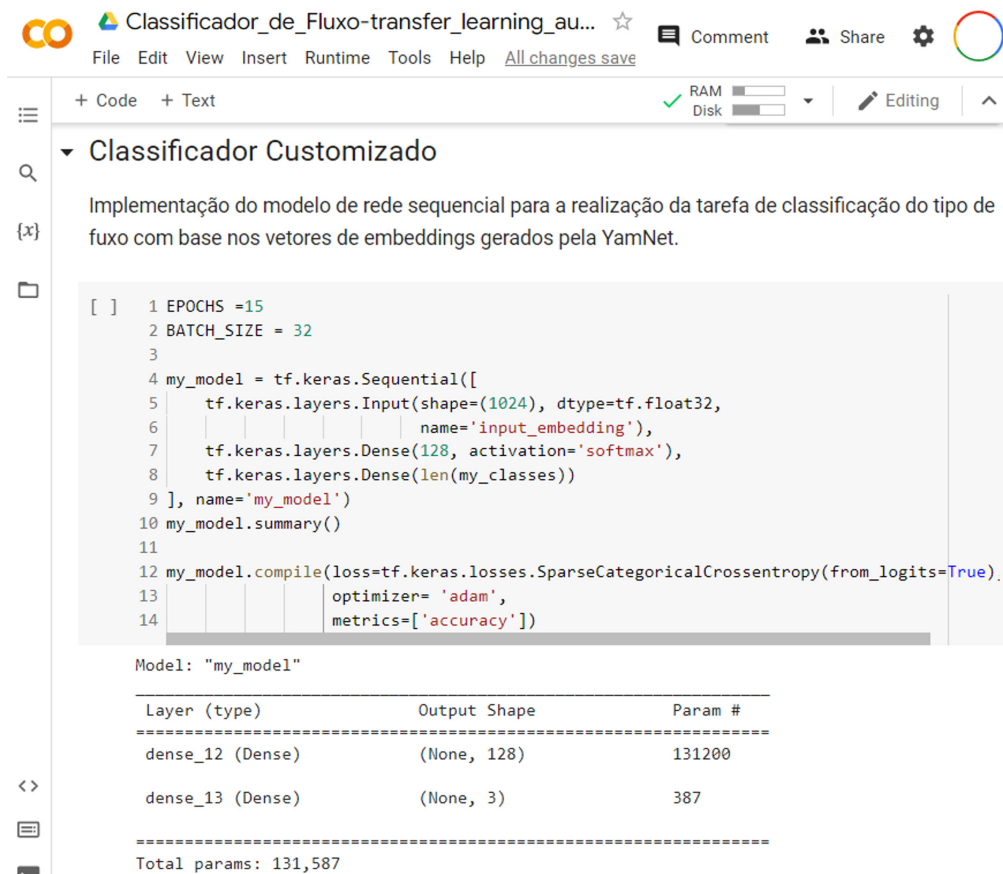
Os trechos de código podem ser executados individualmente, facilitando o processo de ajuste em partes do programa sem a necessidade de rodar todo o código para validar uma modificação. Para exemplificar, é apresentado na Figura 25 uma janela do ambiente de Colab contendo parte do código de implementação do classificador customizado.

Dentre as vantagens de se utilizar a plataforma Colab na implementação de modelos de redes neurais estão: a facilidade de conexão com o serviço de armazenamento de arquivos Google Drive, que permite abrir e executar os Colab Notebooks que estejam salvos em uma conta e o mapeamento de pastas do para dentro do ambiente de desenvolvimento e a possibilidade de utilizar recursos de hardware do tipo GPU para realizar o treinamento e execução dos modelos.

Desta forma, para que pudesse ser utilizada pelo ambiente de desenvolvimento na plataforma Colab, a base de dados contendo os arquivos de áudio e os arquivos CSV com os metadados foram salvos em uma diretório dentro de uma conta do Google Drive. Tendo as bases de dados disponíveis, o classificador foi implementado, treinado e avaliado seguindo o fluxo apresentado no diagrama em blocos da Figura 26.

Nas primeiras etapas as dependências são instaladas, as bibliotecas do Python, como tensorflow, NumPy e Pandas, são importadas e o mapeamento do diretório do Google Drive contendo os arquivos de áudio é realizado. Na sequência, o modelo pré-treinado da rede YAMNet é baixado do seu repositório no GitHub (disponível em: <https://github.com/tensorflow/models/tree/master/research/audioset/yamnet>).

**Figura 25 – Trecho do Colab Notebook apresentando parte do código de implementação do classificador customizado.**



```
[ ] 1 EPOCHS =15
2 BATCH_SIZE = 32
3
4 my_model = tf.keras.Sequential([
5     tf.keras.layers.Input(shape=(1024), dtype=tf.float32,
6         name='input_embedding'),
7     tf.keras.layers.Dense(128, activation='softmax'),
8     tf.keras.layers.Dense(len(my_classes))
9 ], name='my_model')
10 my_model.summary()
11
12 my_model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
13                 optimizer='adam',
14                 metrics=['accuracy'])
```

Model: "my\_model"

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 128)	131200
dense_13 (Dense)	(None, 3)	387

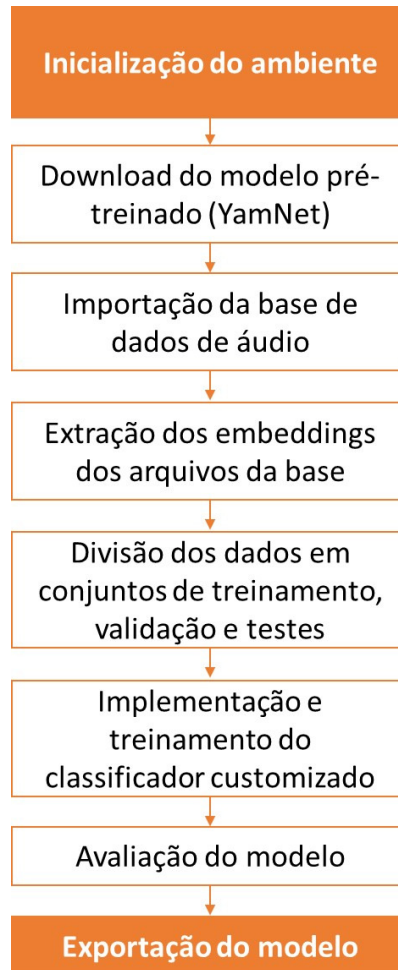
-----  
Total params: 131,587

**Fonte: Autoria própria.**

Em seguida, os arquivos de áudio são importados e processados pela rede YAMNet para obtenção dos vetores de *embeddings*. Após a obtenção dos vetores de características os dados da base de testes são separados em 3 conjuntos, treinamento, validação e testes. Os dados de validação representam uma parcela da base de testes que é utilizada para monitorar o avanço da precisão e erro da rede durante as etapas de treinamento, não sendo utilizados para ajustar os pesos da rede. Já os dados de teste não são utilizados de nenhuma forma durante o processo de treinamento e servem para avaliar a capacidade de generalização da rede para um conjunto desconhecido.

Após a separação das bases de dados, o treinamento da rede é realizado utilizando um otimizador de pesos do tipo Adam, com função de perda do tipo entropia categórica cruzada esparsa (*Sparse Categorical Cross-entropy*). Por fim são gerados gráficos de evolução de precisão e erro resultantes do treinamento e as matrizes de confusão para avaliar o desempenho da rede na base de testes.

**Figura 26 – Fluxo de implementação e treinamento do classificador customizado na plataforma Colab.**



**Fonte: Autoria própria.**

## 4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados de treinamento do modelo e o seu desempenho para a tarefa de classificação do tipo de fluxo. Apresentando desafios encontrados no processo e as soluções propostas.

O primeiro ciclo de treinamento e teste da rede foi realizado utilizando apenas dados de uma seção de 9 m da região horizontal da tubulação. Os dados dos 27 experimentos foram divididos em 81 % para treinamento e 19 % para testes, buscando maximizar a quantidade de dados disponíveis para treinamento da rede, conforme apresentado na Tabela 2. Os dados de testes foram selecionados visando a utilização de experimentos com velocidades superficiais diferentes das presentes na base de treinamento.

Conforme apresentado na subseção 3.3.1 e na subseção 3.3.2, os arquivos de áudio são gerados à partir de trechos de 1 m ao longo da fibra com duração de 5 s dos quais são extraídos vetores de *embeddings* a cada 0,48 s. Ou seja, para cada arquivo de áudio são gerados 10 vetores de *embeddings*. Para esta base inicial foram gerados 243 arquivos dos quais foram extraídos 2430 vetores.

**Tabela 2 – Divisão dos experimentos entre as bases de treinamento e teste para um trecho de 9 m da seção horizontal da tubulação**

Tipo de fluxo	Base	Experimentos [ $Jl$ [m/s]; $Jg$ [m/s]]	Quantidade Arquivos de Áudio
Mono Gás	Treinamento	1[0; 20], 2[0; 5], 3[0; 10], 5[0; 20], 6[0; 25]	45
	Teste	4[0; 15]	9
Mono Líquido	Treinamento	7[1; 0], 8[2; 0], 9[0.5; 0], 11[1; 0], 12[1.25; 0], 13[1.5; 0], 15[2; 0], 16[2.25; 0], 17[2.5; 0], 18[2.75; 0], 19[3; 0]	99
	Teste	10[0.75; 0], 14[1.75; 0]	18
Multifásico	Treinamento	20[1; 1], 21[1; 1], 22[2; 5], 23[1; 9.5], 26[2; 3], 27[1; 5]	54
	Teste	24[2; 1], 25[1; 3]	18
<b>Percentagem de treinamento</b>			81%
<b>Percentagem de testes</b>			19%
<b>Total de arquivos de áudio</b>			243
<b>Total de vetores de embeddings</b>			2430

**Fonte: Autoria própria.**

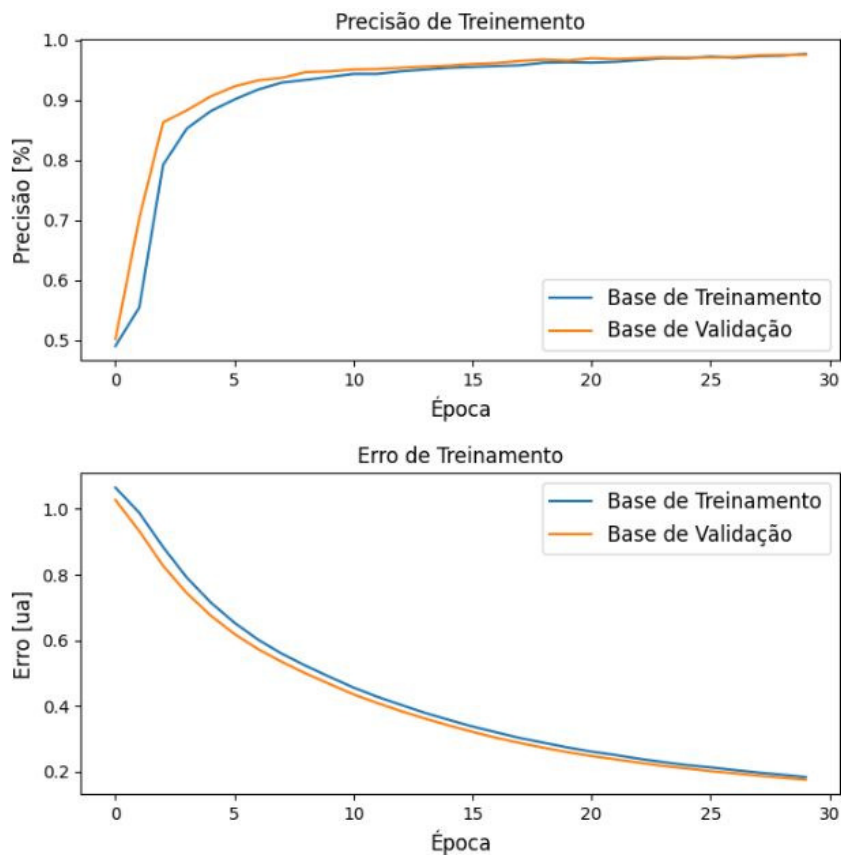
No primeiro treinamento, o classificador customizado foi treinado durante 30 épocas, com a base de treinamento separada em 70% para treinamento e 30% para validação, com um

tamanho de bateladas (*batch size*) igual a 32, utilizando otimizador do tipo Adam e função de perda do tipo entropia categórica cruzada esparsa para o cálculo do erro.

No processo de treinamento, uma época consiste em um ciclo completo de ajuste dos pesos do modelo levando em consideração todos os dados da base de treinamento. Já o *batch size* define o tamanho dos conjuntos de dados utilizados durante as iterações do ajuste dos valores dos pesos da rede dentro da execução de uma época.

Os gráficos de evolução da precisão e erro de saída do modelo ao longo das épocas de treinamento são apresentados na Figura 27. No gráfico superior nota-se que comportamento do aprendizado da rede é exponencial, crescendo de maneira acentuada até a época 10. Também é possível notar que o modelo aprende rapidamente as características da base de dados e atinge uma precisão de 97,55% de acerto na classificação dos dados de treinamento e validação, ao final das 30 épocas. Visto que as bases de treinamento e validação foram extraídas dos mesmos experimentos, pela similaridade dos dados é esperado que a precisão de ambas converta para o mesmo valor.

**Figura 27 – Gráficos de evolução da precisão e erro durante o treinamento do classificador customizado com dados da seção horizontal da tubulação.**

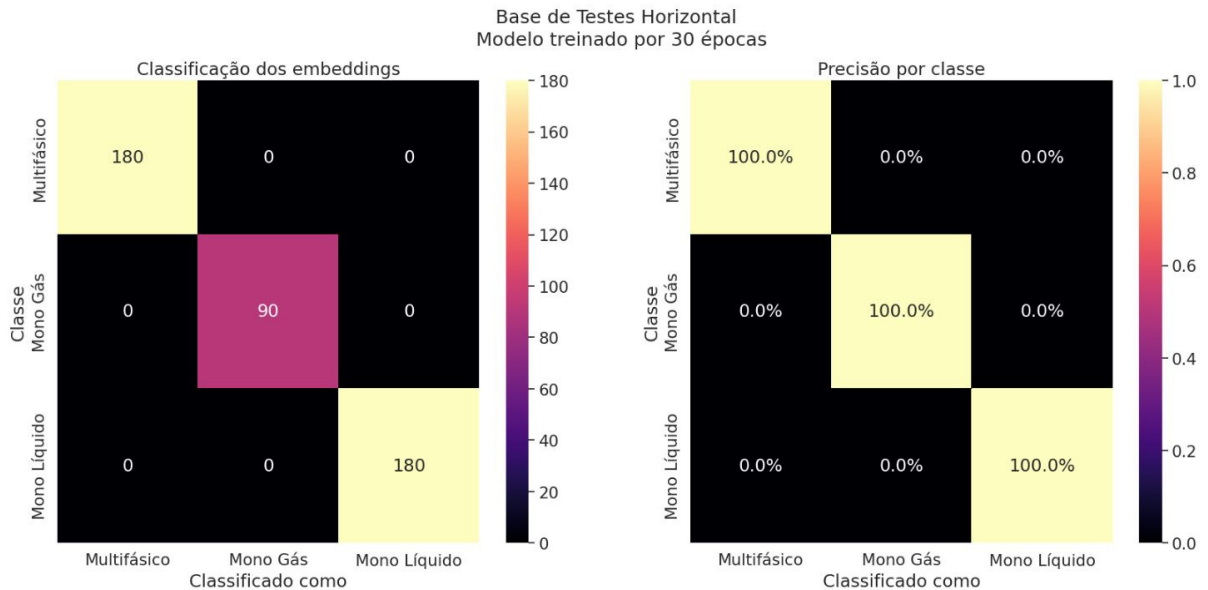


Fonte: Autoria própria.



A matriz de confusão para a classificação dos dados da base de testes é apresentada na Figura 28. Neste caso, o modelo foi capaz de classificar corretamente 100 % dos dados para as 3 classes. Lembrando que a base de testes utilizada para essa avaliação, apresentada anteriormente na Tabela 2, contempla apenas dados de 5 experimentos.

**Figura 28 – Matrizes de confusão para as classificações realizadas pelos modelos em 30 épocas na base de testes contendo dados da seção horizontal da tubulação.**

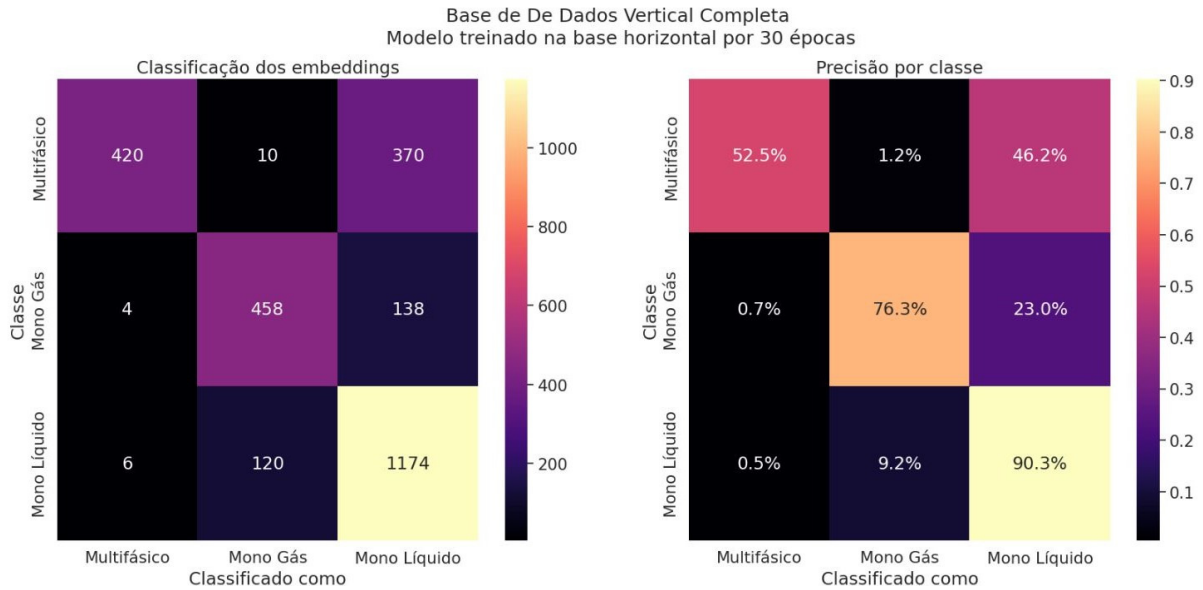


Fonte: Autoria própria.

Motivado pelo bom desempenho do modelo na base de testes da região horizontal da tubulação, uma nova análise foi realizada levando em conta a parte vertical, buscando avaliar a capacidade de generalização do modelo. Para isso os um anova base de testes foi criada utilizando dados de todos os 27 experimentos extraídos de um trecho de 10 m da parte vertical da tubulação, resultando em 2700 vetores de *embeddings*.

Novamente um a matriz de confusão foi gerada para avaliar a capacidade de classificação do modelo para os dados da região vertical, os resultados são apresentados na Figura 29. Na matriz da esquerda temos as classificações realizadas para os vetores de *embeddings* extraídos dos arquivos de áudio e na matriz da direita uma representação da precisão por classe. A classe Mono Líquido apresenta o melhor desempenho, com uma precisão de 90,3%. Para a classe Mono Gás a precisão é de 76,3%, sendo impactada de maneira mais significativa por 23,0% dos casos que foram classificados como Mono Líquido. Já a classe Multifásico teve o pior resultado, com precisão de 52,5% e 46,2% dos casos sendo classificados como Mono Líquido. A precisão geral foi de 76,0%.

**Figura 29 – Matrizes de confusão para a classificação dos dados de todos os experimentos referente a seção vertical do tubo utilizando modelo treinado com dados da seção horizontal.**



Os resultados obtidos na base de dados da região vertical indicam que o modelo treinado com dados da seção horizontal possui uma tendência de classificação dos dados da seção vertical como sendo da classe Mono Líquido. Um dos fatores que podem contribuir para essa tendência é o desbalanceamento da base de treinamento, pois 48 % dos experimentos desta são do tipo Mono Líquido.

Buscando verificar o impacto do desbalanceamento, uma nova base de dados foi criada de forma balanceada, contendo 5 experimentos de cada tipo de fluxo na base de treinamento, utilizando o mesmo trecho de 9 m da seção horizontal. Os experimentos que já faziam parte da base de teste foram mantidos e aqueles que saíram da base de treinamento foram incluídos na base de teste conforme apresentado na Tabela 3.

Após extrair os vetores de *embeddings* para a classe balanceada, um novo treinamento do modelo foi conduzido utilizando as mesmas características do treinamento anterior. Todavia, o desempenho deste no treinamento para os dados da seção vertical se manteve similar. Apresentando uma precisão de 56,9% para a classe Multifásico, 72,8% para a classe Mono gás e 88,5% para a classe Mono Líquido. Indicando que balanceamento dos dados de treinamento não resultou em uma melhoria significativa na capacidade do modelo classificar corretamente os experimentos do tipo Multifásico.

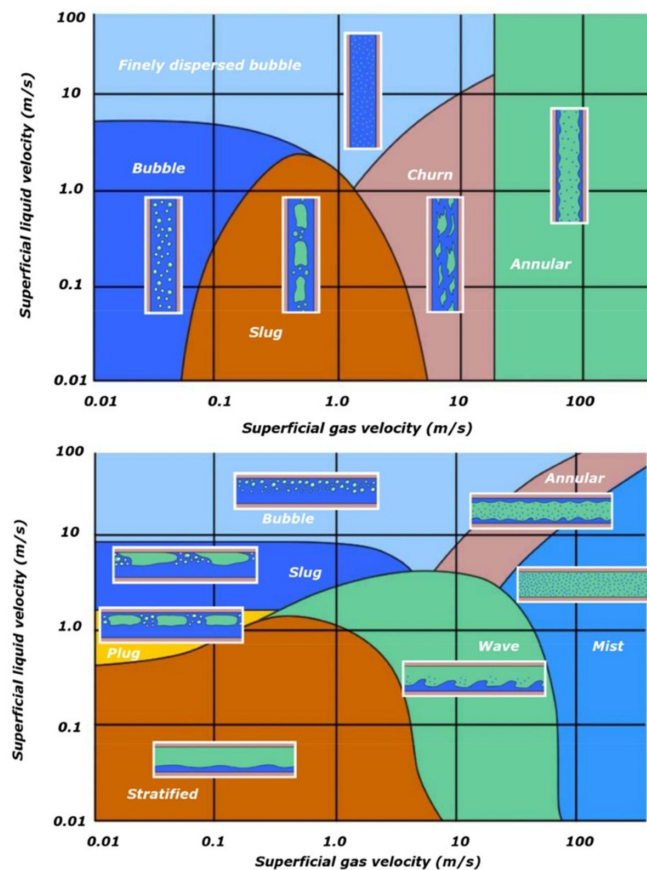
Tabela 3 – Divisão dos experimentos para a base balanceada de treinamento.

Tipo de fluxo	Base	Experimentos [ $Jl$ [m/s]; $Jg$ [m/s]]
Mono Gás	Treinamento	1[0; 20], 2[0; 5], 3[0; 10], 5[0; 20], 6[0; 25]
	Teste	4[0; 15]
Mono Líquido	Treinamento	8[2; 0], 9[0.5; 0], 11[1; 0], 13[1.5; 0], 19[3; 0]
	Teste	7[1; 0], 10[0.75; 0], 12[1.25; 0], 14[1.75; 0], 15[2; 0], 16[2.25; 0], 17[2.5; 0], 18[2.75; 0]
Multifásico	Treinamento	21[1; 1], 22[2; 5], 23[1; 9.5], 26[2; 3], 27[1; 5]
	Teste	20[1; 1], 24[2; 1], 25[1; 3]

Fonte: Autoria própria.

Uma vez verificado que o balanceamento não apresentou impacto significativo na classificação, o passo seguinte foi incluir na base de treinamento dados da seção vertical. Conforme apresentado na Figura 30, para a mesma combinação de velocidade de líquido e gás é possível que perfil de escoamento seja diferente para as orientações vertical e horizontal, gerando diferentes perturbações acústicas.

Figura 30 – Perfis de escoamento de acordo com as velocidades de fase e orientação do fluxo.



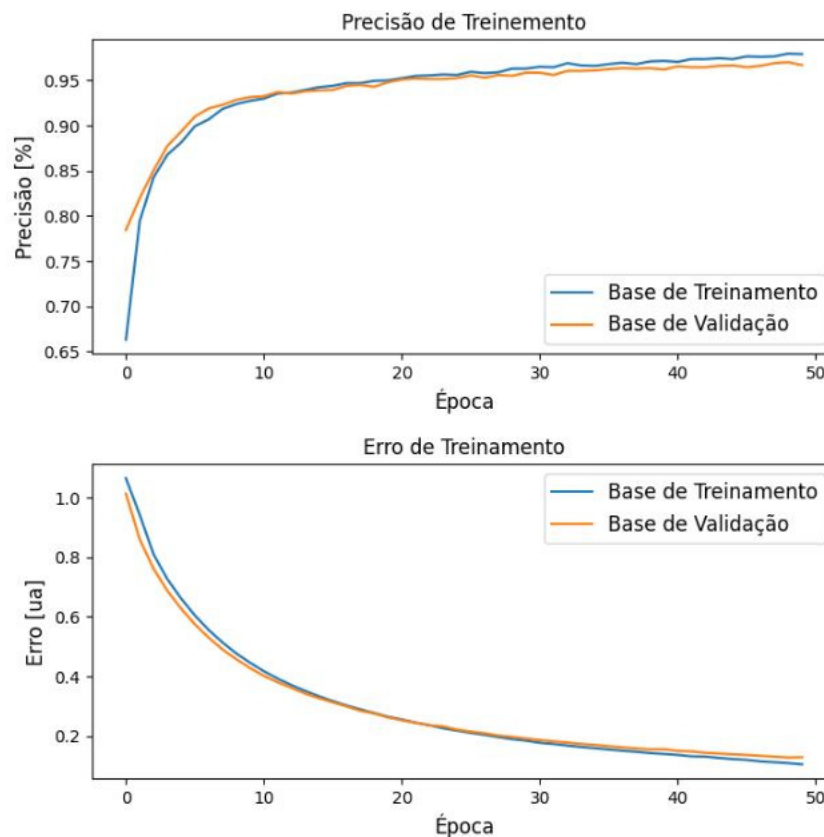
Fonte: (WANG *et al.*, 2019).

Desta forma, uma terceira base de dados foi criada incluindo os dados da seção horizontal e vertical de forma balanceada, seguindo a mesma divisão de experimentos proposta na Tabela 3. O treinamento do modelo foi conduzido novamente utilizando as mesmas configurações dos anteriores, porém dado o aumento do conjunto de dados o número de épocas foi alterado para 50.

Os resultados do treinamento com dados de ambas as seções a tubulação são apresentados na Figura 31. Nota-se que novamente o modelo apresenta um bom desempenho geral atingindo valores acima de 95% de precisão para a classificação dos dados das bases de teste e validação.

No entanto, logo após a época 20 a curva de precisão da base de treinamento começa a se destacar ligeiramente da curva da base de validação, indicando que a rede começou a se sobre-ajustar aos dados da base de treinamento, fenômeno também conhecido como *overfitting*. Desta forma optou-se por treinar o modelo novamente reduzindo o número de épocas para 20, buscando a melhor performance do classificador para os dados de teste.

**Figura 31 – Gráficos de evolução da precisão e erro durante o treinamento do classificador customizado com dados das seções horizontal e vertical.**



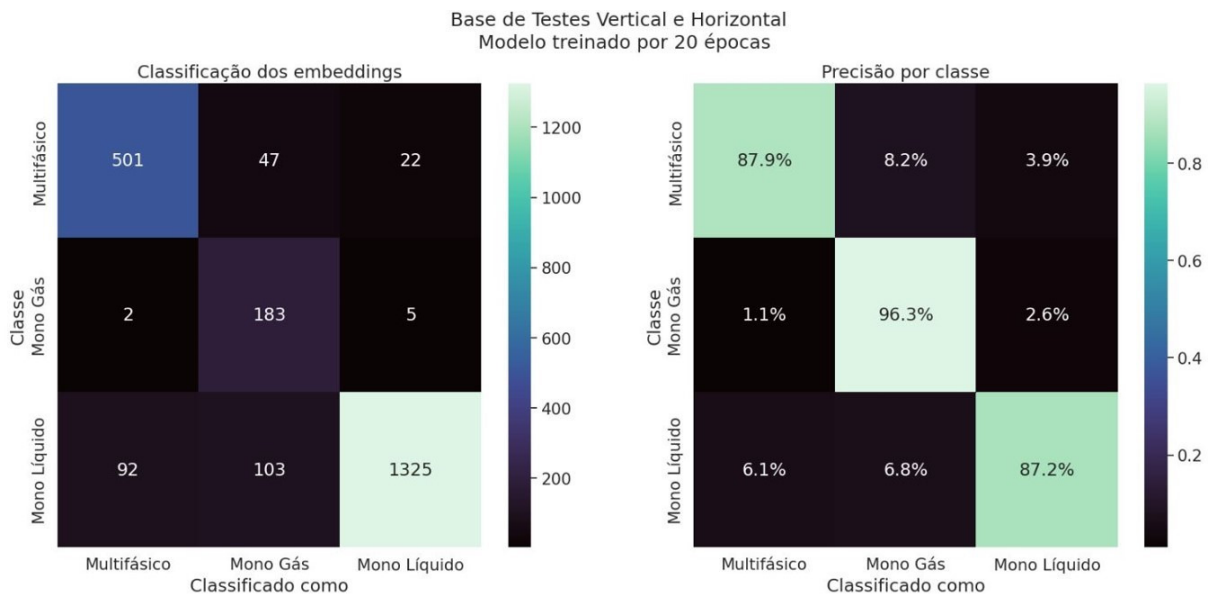
Fonte: Autoria própria.

O desempenho do modelo para a base de testes contendo dados da seção horizontal e vertical da tubulação é apresentado na Figura 32. Observando a matriz da esquerda é possível observar que o modelo foi capaz de classificar corretamente 87,9 % dos dados da classe Multifásico, 96,3 % dos dados da classe Mono Gás e 87,2 % dos dados da classe Mono Líquido. Sendo que a geral foi de 88,7 %.

Os resultados obtidos mostram que a inclusão dos dados da seção vertical melhorou a capacidade do modelo para a classe Multifásico, independente da orientação do fluxo sem prejudicar significativamente a classificação das outras classes.

Estes resultados também mostram a capacidade de generalização do modelo. Pois, mesmo sendo treinado com apenas 15 experimentos (5 de cada classe) na base de treinamento, o modelo foi capaz de classificar com precisão acima de 87 % todas as classes para 22 experimentos da base de testes, tanto para dados da região vertical quanto horizontal.

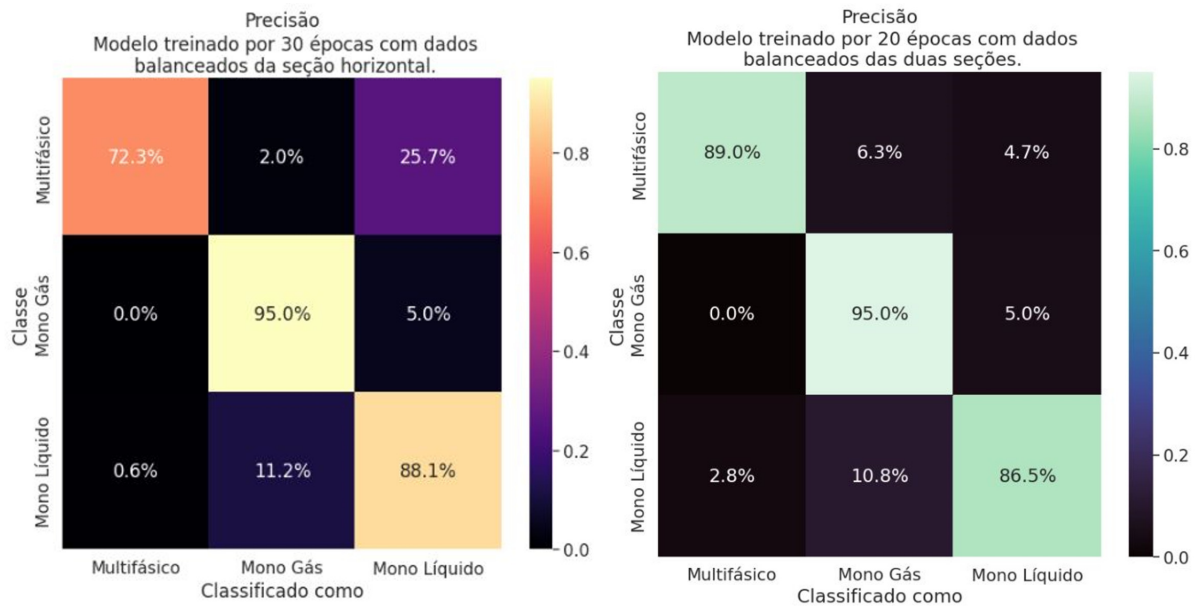
**Figura 32 – Matrizes de confusão para as classificações realizadas pelos modelos em 30 épocas na base de testes contendo dados da seção horizontal e vertical da tubulação.**



Fonte: Autoria própria.

Finalmente uma última avaliação foi conduzida para comparar o desempenho da versão modelo treinado apenas dados balanceados da seção horizontal com o modelo treinado com os dados balanceados de ambas as seções. Para que a comparação fosse correta, a análise foi feita utilizando apenas os dados de teste da seção horizontal. As matrizes de confusão são apresentadas na Figura 33. Os resultados obtidos reafirmam as conclusões anteriores mostrando que a inclusão dos dados da seção vertical melhora a capacidade de generalização do modelo.

**Figura 33 – Matrizes de confusão para da seção vertical da tubulação. À esquerda resultado com o modelo treinado apenas com os dados da seção horizontal. À direita resultado do modelo treinado com as seções horizontal e vertical.**



**Fonte: Autoria própria.**

Durante o processo de implementação do classificador de fluxo, foram testadas variações dos hiperparâmetros ligados ao modelo e ao processo de treinamento, alterando valores de número de camadas, tamanho das camadas e função de ativação, número de épocas, função de penalização e algoritmo de otimização dos pesos. Porém, em todos os testes o comportamento e desempenho da rede foram similares ou inferiores ao apresentado neste capítulo. Sendo assim, optou-se por não apresentar os resultados para estas variações.

No entanto esta invariabilidade ou insensibilidade do modelo aos ajustes nos hiperparâmetros revela dois pontos importantes. O primeiro é de que a metodologia proposta e tipo de modelo utilizado estão bem adaptados ao problema, pois mesmo com configurações diferentes o modelo manteve sua performance. E o segundo é que a melhoria da capacidade de generalização do modelo está ligada a melhoria da base de dados através da inclusão de novos experimentos com uma maior dispersão dos valores de velocidade de fase e diferentes orientações do fluxo.

## 5 CONCLUSÕES E PERSPECTIVAS

Os resultados obtidos mostram que metodologia de implementação do classificador de fluxo foi capaz de obter bons resultados, atingindo uma precisão acima de 87,0 % para as 3 classes de fluxo. Desta forma, é possível concluir que a técnica de sonificação proposta para a conversão dos dados do sensor DAS em arquivos de áudio foi capaz de gerar informações suficientes para o treinamento da rede e para a classificação dos experimentos.

Contudo, tanto a metodologia de sonificação quanto a rede neural YAMNet, utilizada para a extração de características (*embeddings*) dos arquivos de áudio, limitam a exploração dos dados do DAS na faixa de frequências de 125 Hz a 7500 Hz, gerando o descarte das informações fora desta faixa. Trabalhos futuros podem explorar técnicas de conversão em áudio que possibilitem a exploração das faixas desconsideradas, através da reamostragem do sinal gerado pelo DAS.

Além disso, a técnica de transferência de aprendizado escolhida mostra a flexibilidade de utilização de um modelo pré-treinados como extrator de recursos para o treinamento um classificador customizado, reduzindo o tempo de treinamento e possibilitando uma boa performance mesmo com um número reduzido de dados de treinamento. Ao generalizar a classificação para as regiões horizontal e vertical, foi possível observar que a melhoria da classificação dos dados está ligada ao aumento da diversidade de experimentos na base de treinamento.

Dado o bom desempenho para a tarefa de classificação em 3 tipos de fluxo, trabalhos futuros podem explorar a utilização desta metodologia para classificar diferentes perfis de escoamento dentro da classe multifásico. Ou até mesmo, a detecção de outros eventos na tubulação como fechamento e abertura de válvulas, rachaduras e intervenções de manutenção.

Por fim, uma vez que a metodologia proposta está relacionada a tecnologia do sensor e não ao elemento monitorado, a sua utilização pode ser estendida para a classificação de eventos acústicos em outras estruturas.

## REFERÊNCIAS

AGRAWAL, Gp. **Nonlinear Fiber Optics**. 3. ed. [S.l.]: Academic Press, 2000. ISSN: 0935-9648. ISBN 978-0-12-397023-7.

AOYAMA, K.; NAKAGAWA, K.; ITOH, T. Optical time domain reflectometry in a single-mode fiber. **IEEE Journal of Quantum Electronics**, v. 17, n. 6, p. 862–868, 1981.

BAO, X.; CHEN, L. Recent progress in distributed fiber optic sensors. **Sensors (Basel)**, v. 12, n. 7, p. 8601–8639, 2012. ISSN 1424-8220. ISBN: 1424-8220 (Electronic)\r1424-8220 (Linking).

BARCELLOS, William; SHITARA, Nicolas; FERRAZ, Carolina; QUEIROGA, Raissa; SAITO, Jose; GONZAGA, Adilson. Evaluation of fine tuning and feature extraction methods in biometric periocular recognition. *In: Anais do XV Workshop de Visão Computacional*. Porto Alegre, RS, Brasil: SBC, 2019. p. 43–48. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/wvc/article/view/7626>.

BOYD, Robert W.; MASTERS, Barry R. Nonlinear optics, third edition. **Journal of biomedical optics**, v. 14, p. 029902, 1960. ISSN 10833668. ISBN: 9780123694706.

BREGMAN, Albert S. **Auditory Scene Analysis: The Perceptual Organization of Sound**. The MIT Press, 1990. ISBN 9780262269209. Disponível em: <https://doi.org/10.7551/mitpress/1486.001.0001>.

CAMPO, Alberto. A data sonification design space map. 01 2007.

CHEN, Yushi; LIN, Zhouhan; ZHAO, Xing; WANG, Gang; GU, Yanfeng. Deep learning-based classification of hyperspectral data. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 7, n. 6, p. 2094–2107, 2014.

CHETLUR, Sharan; WOOLLEY, Cliff; VANDERMERSCH, Philippe; COHEN, Jonathan; TRAN, John; CATANZARO, Bryan; SHELHAMER, Evan. cudnn: Efficient primitives for deep learning. **CoRR**, abs/1410.0759, 2014. Disponível em: <http://arxiv.org/abs/1410.0759>.

COLABORATORY, Google. **Welcome To Colaboratory**. 2022. Disponível em: <https://colab.research.google.com/>.

DENG, Jia; SOCHER, R.; FEI-FEI, Li; DONG, Wei; LI, Kai; LI, Li-Jia. Imagenet: A large-scale hierarchical image database. *In: 2009 IEEE Conference on Computer*



**Vision and Pattern Recognition(CVPR)**. [s.n.], 2009. v. 00, p. 248–255. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5206848/>.

DOMBOIS, Florian; BIRLINGHOVEN, Schloss. Auditory seismology on free oscillations, focal mechanisms, explosions and synthetic seismograms. *In: . [S.l.: s.n.]*, 2002.

DONAHUE, Jeff; JIA, Yangqing; VINYALS, Oriol; HOFFMAN, Judy; ZHANG, Ning; TZENG, Eric; DARRELL, Trevor. Decaf: A deep convolutional activation feature for generic visual recognition. *In: XING, Eric P.; JEBARA, Tony (Ed.). **Proceedings of the 31st International Conference on Machine Learning***. Beijing, China: PMLR, 2014. (Proceedings of Machine Learning Research, 1), p. 647–655. Disponível em: <https://proceedings.mlr.press/v32/donahue14.html>.

DUTRA, Guilherme. **Development of advanced optical sensing techniques**. 2018. 134 p. Tese (dissertation) — Federal University of Technology, PR. Curitiba, 2018.

FARAHANI, M.A.; GOGOLLA, T. Spontaneous raman scattering in optical fibers with modulated probe light for distributed temperature raman remote sensing. **Journal of Lightwave Technology**, v. 17, n. 8, p. 1379–1391, 1999.

FERNANDEZ-RUIZ, María R.; PASTOR-GRAELLS, Juan; MARTINS, Hugo F.; GARCIA-RUIZ, Andres; MARTIN-LOPEZ, Sonia; GONZALEZ-HERRAEZ, Miguel. > 10 db snr enhancement in distributed acoustic sensors through first order phase noise cancellation. *In: **2018 Optical Fiber Communications Conference and Exposition (OFC)***. [S.l.: s.n.], 2018. p. 1–3.

FURUKAWA. **Cabo OPTIC-LAN - ABNT**. 2022. Disponível em: <https://www.furukawalatam.com/pt-br/catalogo-de-produtos-detalhes/optic-lan---abnt>.

GEMMEKE, Jort F.; ELLIS, Daniel P. W.; FREEDMAN, Dylan; JANSEN, Aren; LAWRENCE, Wade; MOORE, R. Channing; PLAKAL, Manoj; RITTER, Marvin. Audio set: An ontology and human-labeled dataset for audio events. *In: **Proc. IEEE ICASSP 2017***. New Orleans, LA: [s.n.], 2017.

GIALLORENZI, T.G.; BUCARO, J.A.; DANDRIDGE, A.; SIGEL, G.H.; COLE, J.H.; RASHLEIGH, S.C.; PRIEST, R.G. Optical fiber sensor technology. **IEEE Transactions on Microwave Theory and Techniques**, v. 30, n. 4, p. 472–511, 1982.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HARTOG, A.H. **An Introduction to Distributed Optical Fibre Sensors**. Taylor & Francis Group, 2018. (Fiber Optic Sensors). ISBN 978-1-138-08269-4. Disponível em: <https://books.google.com.br/books?id=aVmWnQAACAAJ>.

HAYKIN, Simon S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009.

HERSHEY, Shawn; CHAUDHURI, Sourish; ELLIS, Daniel P. W.; GEMMEKE, Jort F.; JANSEN, Aren; MOORE, Channing; PLAKAL, Manoj; PLATT, Devin; SAUROUS, Rif A.; SEYBOLD, Bryan; SLANEY, Malcolm; WEISS, Ron; WILSON, Kevin. Cnn architectures for large-scale audio classification. *In: International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [s.n.], 2017. Disponível em: <https://arxiv.org/abs/1609.09430>.

HOWARD, Andrew G.; ZHU, Menglong; CHEN, Bo; KALENICHENKO, Dmitry; WANG, Weijun; WEYAND, Tobias; ANDREETTO, Marco; ADAM, Hartwig. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **CoRR**, abs/1704.04861, 2017. Disponível em: <http://arxiv.org/abs/1704.04861>.

HU, Die; TIAN, Bin; LI, Hao; FAN, Cunzheng; LIU, Tao; HE, Tao; LIU, Yijie; YAN, Zhijun; SUN, Qizhen. Intelligent structure monitoring for tunnel steel loop based on distributed acoustic sensing. *In: 2021 Conference on Lasers and Electro-Optics (CLEO)*. [S.l.: s.n.], 2021. p. 1–2.

IMMERMESSEN. **datas-dois-new**. 2022. Disponível em: <https://immermessen.com/das.php>.

JUAREZ, J. C.; TAYLOR, H. F. Distributed fiber optic intrusion sensor system. **OFC/NFOEC Technical Digest. Optical Fiber Communication Conference, 2005.**, v. 4, n. 6, p. 2081–2087, 2005. ISSN 0733-8724. ISBN: 1-55752-783-0.

JUPYTER. **Project Jupyter's origins and governance**. 2022. Disponível em: <https://jupyter.org/>.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. *In: PEREIRA, F.; BURGESS, C. J.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

LECUN, Yann; BOTTOU, Leon; ORR, Genevieve B.; MÜLLER, Klaus Robert. Efficient BackProp. *In: ORR, Genevieve B.; MÜLLER, Klaus-Robert (Ed.). Neural Networks: Tricks of the Trade*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 9–50. ISBN 978-3-540-49430-0. Disponível em: [https://doi.org/10.1007/3-540-49430-8\\_2](https://doi.org/10.1007/3-540-49430-8_2).

LI, Zhizhong; HOIEM, Derek. Learning without forgetting. **CoRR**, abs/1606.09282, 2016. Disponível em: <http://arxiv.org/abs/1606.09282>.

LOPEZ-HIGUERA, José Miguel; COBO, Luis Rodriguez; INCERA, Antonio Quintela; COBO, Adolfo. Fiber optic sensors in structural health monitoring. **Journal of Lightwave Technology**, v. 29, n. 4, p. 587–608, 2011.

MARTELLI, Cicero; SILVA, Erlon Vagner da; SOUZA, Kleiton de Moraes; MEZZADRI, Felipe; SOMENZI, Jonas; CRESPI, Marcos; KALINOWSKI, Hypolito José; SILVA, Jean Carlos Cordoza da. Temperature sensing in a 175MW power generator. *In*: LIAO, Yanbiao; JIN, Wei; SAMPSON, David D.; YAMAUCHI, Ryozi; CHUNG, Youngjoo; NAKAMURA, Kentaro; RAO, Yunjiang (Ed.). **OFS2012 22nd International Conference on Optical Fiber Sensors**. SPIE, 2012. v. 8421, p. 84212F. Disponível em: <https://doi.org/10.1117/12.975308>.

MARTINS, Luiz Gustavo. **Transfer Learning for Audio Data with YAMNet**. 2021. Disponível em: <https://blog.tensorflow.org/2021/03/transfer-learning-for-audio-data-with-yamnet.html>.

MCCLOUGHLIN, Ian; ZHANG, Haomin; XIE, Zhipeng; SONG, Yan; XIAO, Wei. Robust sound event classification using deep neural networks. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, v. 23, n. 3, p. 540–552, 2015.

NEES, Michael; WALKER, Bruce. Theory of sonification. *In*: \_\_\_\_\_. [*S.l.*: *s.n.*], 2012. p. 9–39. ISBN 978-3-8325-2819-5.

OLSHANSKY, R. Propagation in glass optical waveguides. **Reviews of Modern Physics**, v. 51, n. 2, p. 341–367, 1979. ISSN 00346861. ISBN: 0034-6861.

OTHONOS, Andreas. Fiber bragg gratings. **Review of Scientific Instruments**, v. 68, n. 12, p. 4309–4341, 1997. Disponível em: <https://doi.org/10.1063/1.1148392>.

RESHAK, A. H.; SHAHIMIN, M. M.; MURAD, S. A. Z.; AZIZAN, S. Simulation of Brillouin and Rayleigh scattering in distributed fibre optic for temperature and strain sensing application. **Sensors and Actuators, A: Physical**, v. 190, p. 191–196, 2013. ISSN 09244247.

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65 6, p. 386–408, 1958.

RUSSAKOVSKY, Olga; DENG, Jia; HUANG, Zhiheng; BERG, Alexander C.; FEI-FEI, Li. Detecting avocados to zucchinis: What have we done, and where are we going? *In*: **2013 IEEE International Conference on Computer Vision**. [*S.l.*: *s.n.*], 2013. p. 2064–2071.

SENIOR, John M.; JAMRO, M. Yousif. **Optical Fiber Communications: Principles and Practice**. [*S.l.*]: Pearson Education Limited, 2009. ISSN: 0030-3909. ISBN 0-13-032681-X.

SIEGEL, Robert J. A replication of the mel scale of pitch. **The American Journal of Psychology**, University of Illinois Press, v. 78, n. 4, p. 615–620, 1965. ISSN 00029556. Disponível em: <http://www.jstor.org/stable/1420924>.

STUCKENBRUCK, Sidney. Mec2346 - escoamento multifásico. In: \_\_\_\_\_. [s.n.], 2022. Disponível em: <http://multifasico.usuarios.rdc.puc-rio.br/>.

VAHABI, Nafiseh; WILLMAN, Eero; BAGHSIAHI, Hadi; SELVIAH, David R. Fluid flow velocity measurement in active wells using fiber optic distributed acoustic sensors. **IEEE Sensors Journal**, v. 20, n. 19, p. 11499–11507, 2020.

WANG, Qiang; JIA, Xiaodong; WANG, Mi. Bubble mapping: Three-dimensional visualisation of gas-liquid flow regimes using electrical tomography. **Measurement Science and Technology**, v. 30, 02 2019.

WANG, Zhaoyong; LU, Bin; YE, Qing; CAI, Haiwen. Recent progress in distributed fiber acoustic sensing with -otdr. **Sensors**, v. 20, n. 22, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/22/6594>.

XU, K.; XIE, M.; TANG, L. C.; HO, S. L. Application of neural networks in forecasting engine systems reliability. **Applied Soft Computing**, v. 2, n. 4, p. 255–268, 2003. ISSN 1568-4946. Publisher: Elsevier BV.

ZHANG, Aston; LIPTON, Zachary C.; LI, Mu; SMOLA, Alexander J. Dive into deep learning. **arXiv preprint arXiv:2106.11342**, 2021.

ZURADA, J.M. **Introduction to Artificial Neural Systems**. West, 1992. ISBN 978-0-314-93391-1. Disponível em: <https://books.google.com.br/books?id=wLFQAAAAMAAJ>.