UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VINÍCIUS CELLA CERIOTTI

Classificação de Sons Ambientais utilizando Redes Neurais Convolucionais para aplicações em *Hardwares* com Recursos Limitados

Monografia apresentada como requisito parcial para a obtenção do grau de Engenheiro Eletricista

Orientador: Prof. Dr. Tiago Oliveira Weber

Porto Alegre
2023

RESUMO

A classificação de sons ambientais tem encontrado aplicabilidade em diversos setores, como sistemas de vigilância de áudio, monitoramento inteligente, preservação ambiental e detecção de ruído em áreas urbanas. No contexto de cidades inteligentes, que buscam soluções tecnológicas para os desafios da urbanização, a classificação de sons se mostra especialmente relevante. Com a integração de técnicas de aprendizado de máquina, torna-se possível aprimorar o desempenho no reconhecimento de padrões acústicos, facilitando a implementação de soluções mais robustas e adaptativas em aplicações práticas. A presente pesquisa analisa 5 topologias de redes neurais convolucionais propostas em trabalhos relacionados (SBCNN, DMIX, STRIDED, 2DCNN e 1DCNN) a partir de três técnicas de extração de características (espectrograma em escala logarítmica, espectrograma mel e espectrograma mel em escala logarítmica) e com e sem a utilização de *Data Augmentation*. Os modelos foram validados a partir da utilização de técnica *10-fold cross validation*. A partir dos resultados obtidos, foram aplicadas as técnicas de poda computacional e quantização para redução do número de parâmetros e tamanho dos modelos. A topologia STRIDED obteve taxa de acerto média de $72,19\%$, com 149.892 parâmetros (80% de esparsidade) e tamanho médio de 217.498 *bytes* para a base de dados UrbanSound8K, enquanto que a topologia SBCNN obteve taxa de acerto de $73,48\%$, com 86.476 parâmetros e tamanho médio de 125.669 *bytes*.

Palavras-chave: Sons Ambientais, Redes Neurais Convolucionais, Espectrograma, *Data Augmentation*, Poda Computacional, Quantização.

ABSTRACT

Environmental sound classification has found applicability in several fields such as audio surveillance systems, smart monitoring, environmental conservation, and noise detection in urban areas. Sound classification is essential within the context of smart cities, which seek technological solutions to the challenges of urbanization. With the integration of machine learning techniques, enhancing performance in acoustic pattern recognition becomes possible, facilitating the implementation of more robust and adaptive solutions in practical applications. The current research examines 5 topologies of convolutional neural networks proposed in related works (SBCNN, DMIX, STRIDED, 2DCNN and 1DCNN) using three feature extraction techniques (log-scaled spectrogram, mel spectrogram, and log-scaled mel spectrogram), both with and without the use of Data Augmentation. The models were validated using the 10-fold cross validation technique. Based on the results, pruning and quantization techniques were applied to reduce the number of parameters and model sizes. The STRIDED topology achieved an average accuracy rate of 72.19%, with $149,892$ parameters (80% sparsity) and an average size of $217,498$ bytes for the UrbanSound8K database. Meanwhile, the SBCNN topology achieved an accuracy rate of 73.48%, with $86,476$ parameters and an average size of $125,669$ bytes.

Keywords: Environmental Sounds, Convolutional Neural Networks, Spectrogram, *Data Augmentation*, Pruning, Quantization.

LISTA DE ABREVIATURAS E SIGLAS

FT	*Fourier Transform*
FFT	*Fast Fourier Transform*
STFT	*Short-Time Fourier Transform*
DCT	*Discrete Cosine Tranform*
DFT	*Discrete Fourier Transform*
ESC	*Environmental Sound Classification*
ESR	*Event Sound Recognition*
ADC	*Analog to Digital Converter*
MFCC	*Mel Frequency Cepstral Coefficient*
ANN	*Artificial Neural Network*
MLP	*Multilayer Perceptron*
CPU	*Central Processing Unit*
GPU	*Graphics Processing Unit*
RAM	*Random Access Memory*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O estudo do comportamento de sons ambientais tem se intensificado com o crescimento das áreas urbanas. Um tópico que traz preocupação para as grandes metrópoles se trata da poluição sonora em que as pessoas são expostas diariamente. A poluição sonora pode trazer consequências diretas e cumulativas à saúde, podendo causar distúrbios no sono, perda auditiva e doenças cardiovasculares (JARIWALA et al., 2017). Cidades como Nova York, Dublin e Barcelona já possuem sistemas de monitoramento de ruídos a partir da implantação de sensores de áudio em pontos estratégicos da cidade, a fim de obter dados para análise e tomada de decisão (NORDBY, 2019).

Outro campo de aplicação que pode se beneficiar da utilização de sensores de áudio são os sistemas de vigilância a partir da integração com os sistemas de vídeo. Isso se justifica visto que câmeras de segurança padrão têm um campo de visão angular limitado, enquanto que os sensores de áudio podem ser omnidirecionais. Outro ponto é que eventos importantes para a vigilância, como gritos ou disparos de arma de fogo, são mais facilmente captados por sensores de áudio (CROCCO et al., 2014).

Após a aquisição do áudio nos dois cenários de aplicação de sensores supracitados, são necessárias técnicas de detecção e classificação das características dos áudios. Diante dessa necessidade, o uso de técnicas de aprendizado de máquina para a classificação dos sons tem sido amplamente empregado. Pesquisas como Piczak (2015a), Salamon e Bello (2017) e Zhang et al. (2018) têm mostrado que redes neurais convolucionais apresentam resultados significativos para tarefas de classificação de sons ambientais a partir da utilização de técnicas de extração de características com o intuito de obter informações relevantes dos áudios para o treinamento dos modelos.

Tendo em vista as observações apresentadas, a presente pesquisa visa realizar a classificação de sons ambientais a partir da aplicação de treinamento, poda computacional e quantização em topologias de redes neurais convolucionais presentes na literatura científica, com o intuito de reduzi-las em tamanho e complexidade para aplicações em dispositivos embarcados.

1.1 Objetivo Geral

O objetivo geral da presente pesquisa consiste no desenvolvimento de um sistema de classificação de sons ambientais a partir da utilização de Redes Neurais Convolucionais para análise da viabilidade da realização de classificação em dispositivos com limitação de *hardware*, a partir da aplicação de técnicas de otimização e quantização, para a base de dados UrbanSound8K.

1.2 Objetivos Específicos

- Revisão bibliográfica de trabalhos da área de classificação de sons;
- Estudo e análise da base de dados UrbanSound8K;
- Realizar treinamento das topologias de Redes Neurais Convolucionais selecionadas para base de dados UrbanSound8K desconsiderando, inicialmente, a limitação de recursos;
- Aplicar técnicas de poda computacional e quantização para redução de parâmetros e tamanho das topologias selecionadas.

1.3 Justificativa

Pesquisas na área de classificação de sons ambientais possuem uma extensa variedade de possíveis aplicações práticas, que justificam o investimento nesta área. Os tópicos citados abaixo exemplificam alguns dos campos que podem se beneficiar da classificação de sons, destacando a extensa amplitude e diversidade do assunto em discussão:

- Sistemas de vigilância: no campo da segurança, sistemas avançados de vigilância podem se beneficiar dessa tecnologia para detecção de ruídos anômalos, que indiquem atividades potencialmente suspeitas;
- Ruído urbano: no âmbito urbano, auxiliando na identificação e controle de poluição sonora em regiões residenciais e áreas industriais, por exemplo. A classificação de sons ambientais pode contribuir na identificação de fontes de ruídos, como o tráfego, aeroportos, obras e fábricas;
- Controle de qualidade: a fim de controlar a qualidade em processos industriais,

pode-se colaborar na identificação de anomalias ou problemas, tal como a detecção de ruídos anormais ou vibrações em equipamentos;

- Saúde e Medicina: na saúde, o reconhecimento de sons específicos, como os sons cardíacos e pulmonares, pode auxiliar no diagnóstico e monitoramento de condições médicas;
- Indústria automotiva: sistemas de detecção de ruídos e vibrações em veículos podem ser aprimorados com a classificação de sons ambientais, contribuindo para a identificação de problemas mecânicos precoces.

2 FUNDAMENTAÇÃO TEÓRICA

No presente capítulo são revisados os temas e estudos essenciais para a compreensão da pesquisa. Inicialmente é discutido o cenário de impactos dos sons ambientais em cidades inteligentes, seguido por uma revisão dos conceitos de processamento de sinais de áudio. Em seguida, são revisados os conceitos de aprendizado de máquina e redes neurais. Por fim, apresenta-se uma revisão dos trabalhos relacionados à classificação de sons ambientais sem restrição de recursos, e com restrição de recursos para implementação em sistemas embarcados.

2.1 Detecção e Classificação de Sons Ambientais

Uma das maiores tarefas de pesquisa em análise de sinais acústicos é o estudo da classificação de sons ambientais (ESC - do inglês, Environmental Sound Classification), também conhecido como reconhecimento de eventos sonoros (SER - do inglês, Event Sound Recognition). O propósito principal da análise computacional de sons é obter dados do áudio utilizando técnicas computacionais de identificação e reconhecimento. Na classificação, o objetivo é categorizar uma gravação de áudio em um conjunto de classes predefinidas. Na detecção, o objetivo é localizar no tempo as ocorrências de um tipo específico de som, seja encontrando cada instância da ocorrência do som ou encontrando todas as posições temporais que determinado som está ativo (VIRTANEN; PLUMBLEY; ELLIS, 2017) .

A classificação de sons ambientais geralmente envolve a taxonomia de dois grandes componentes básicos: a utilização das melhores características acústicas e a implementação de classificadores com melhores resultados. Normalmente, as características (*features*) de um áudio são extraídas separando o sinal de áudio considerado em *frames* com a utilização da técnica de janelamento. Um conjunto de *features* é extraído para cada *frame* e usado para treinamento e teste (MUSHTAQ; SU, 2020). Uma técnica bem conhecida, e que vem sendo amplamente aplicada para a classificação de eventos sonoros ambientais é o banco de filtros *Mel*, o qual será abordado na subseção 2.3.6.

A ESC tem sido utilizada em aplicações industriais e no contexto de cidades inteligentes, tais como sistemas de vigilância de áudio, aparelhos auditivos, monitoramento inteligente de estabelecimentos, preservação ambiental, detecção de ruído em áreas urbanas, cuidados de saúde e problemas médicos, entre outras aplicações. A seção 2.2

apresenta uma abordagem mais ampla da aplicação de ESC em cidades inteligentes.

2.2 Classificação de Sons Ambientais em Cidades Inteligentes

A rápida tendência de urbanização cria grandes oportunidades de inovação e desenvolvimento econômico, porém também cria problemas significativos relacionados ao impacto ambiental da atividade humana, às condições de infraestrutura, a dificuldade de policiar e proteger efetivamente os espaços públicos e potenciais reduções na saúde e qualidade de vida das pessoas. O conceito de cidades inteligentes surge a partir da tendência de alavancagem de sistemas e soluções tecnológicas para abordar problemas enfrentados pelas comunidades urbanas, a partir dos avanços dos sistemas de sensoriamento inteligente, conectividade generalizada e tratamento de dados para coletar, distribuir e analisar dados necessários para entender a situação do terreno, antecipar o comportamento futuro e conduzir uma ação eficaz (VIRTANEN; PLUMBLEY; ELLIS, 2017).

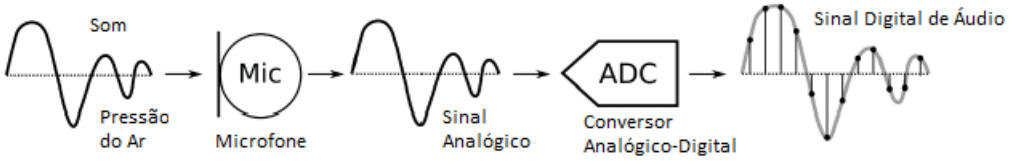
Um dos campos de aplicação de ESC em cidades inteligentes é a utilização de áudios para vigilância. Originalmente, os sistemas de vigilância eram operados por humanos que precisavam monitorar constantemente os fluxos de vídeo provenientes de um vasto número de câmeras necessárias para cobrir todas as áreas de interesse. Essa operação demanda um nível muito alto de atenção do operador, o que, a longo prazo, é um fator que não pode ser garantido. Diante disso, o desenvolvimento de tecnologias capazes de alertar os humanos sobre riscos potenciais tem aumentado. Exemplos de utilização de informações de áudio para fins de vigilância são apresentados em Crocco et al. (2016).

Outra aplicação importante da classificação de sons ambientais em cidades inteligentes refere-se ao monitoramento de poluição sonora, que é considerado uma das causas de maior impacto na qualidade de vida dos residentes de regiões urbanas, agravando quadros de estresse, hipertensão, perturbação do sono e perda de audição (VIRTANEN; PLUMBLEY; ELLIS, 2017). Para combater esse cenário, conforme Taber (2010), a maioria das grandes cidades têm buscado regular a geração de ruído em função do horário do dia e local, medindo o ruído em termos de nível de pressão sonora. A utilização de dispositivos sensores acústicos torna-se uma solução no contexto de monitoramento de ruído, uma vez que a análise do som pode contribuir para a identificação de fontes específicas de ruído e suas características (VIRTANEN; PLUMBLEY; ELLIS, 2017).

2.3 Sinais Digitais de Áudio

O som é uma variação física na pressão que se propaga através de um meio de transmissão ao longo do tempo. Para realizar a classificação de sinais de áudio, deve-se primeiro obter o formato digital do áudio. Conforme apresentado na Figura 2.1, o som é convertido em sinais elétricos analógicos através de um módulo microfone, passa por um processo de filtragem e, então, é digitalizado a partir da utilização de um Conversor Analógico-Digital (ADC - do inglês, *Analog to Digital Converter*) (CHRISTENSEN, 2019).

Figura 2.1: Conversão do som em representação digital.



Fonte: Adaptado de (NORDBY, 2019).

No processo de digitalização, o sinal é quantizado no tempo em uma determinada frequência de amostragem e a amplitude é quantizada em uma determinada profundidade de bits. Uma frequência de amostragem típica é de $44.1kHz$. Com esses parâmetros, a maioria das informações de um áudio, perceptíveis pelo ser humano, são capturadas. Entre as seções 2.3.1 e 2.3.4 serão apresentadas técnicas derivadas da transformada de Fourier que podem ser utilizadas para facilitar a extração de informações de sinais de áudio.

2.3.1 Transformada Discreta de Fourier

A Transformada Discreta de Fourier (DFT – do inglês, *Discrete Fourier Transform*) é utilizada para converter um sinal discreto no domínio do tempo para um sinal discreto no domínio da frequência. A DFT pode ser descrita conforme a Equação 2.1.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \tag{2.1}$$

sendo $X[k]$ o valor da DFT na frequência k, N o número de amostras processadas com n variando de 0 até $N-1$.

2.3.2 Transformada Rápida de Fourier

A Transformada Rápida de Fourier (FFT - do inglês, *Fast Fourier Transform*) é utilizada para contornar o problema de alto custo computacional atrelado à Transformada Discreta de Fourier, visto o número elevado de operações necessárias conforme o número de amostras aumenta. A partir da utilização do algoritmo de Cooley-Tukey, é possível otimizar o processo do cálculo de uma DFT com a utilização de uma FFT, visto que em uma DFT, necessita-se de N^2 operações de multiplicação e $N(N-1)$ operações de adição para a realização do cálculo, enquanto que em uma FFT o número de multiplicações é limitado em $Nlog_2N$ (COOLEY; TUKEY, 1965).

2.3.3 Transformada de Fourier de Tempo-Curto

Uma forma comumente utilizada para computar um espectrograma a partir de sinais de áudio é a utilização da Transformada de Fourier de Tempo-Curto (STFT - do inglês, *Short-Time Fourier Transform*) (SMITH, 2010). A STFT opera dividindo o áudio em pequenos pedaços consecutivos e computando a FFT para estimar o conteúdo de frequência para cada pedaço. Na prática, um dado espectro $X[k]$ é aproximado aplicando a DFT em um *frame* de comprimento N de um dado sinal $x[n]$ (SERIZEL et al., 2017). A f-*ésima* componente da DFT do t-*ésimo frame* de $x[n]$ é calculada conforme a Equação 2.2

$$X[t, f] = \sum_{k=0}^{N-1} w[k]x[tN + k]e^{-j2\pi kf/N} \tag{2.2}$$

onde $w[k]$ é a função de janelamento utilizada visando suavizar alguns dos efeitos da aproximação DFT e para reforçar a continuidade e a periodicidade na borda dos *frames* (SERIZEL et al., 2017).

2.3.4 Transformada Discreta de Cosseno

A Transformada Discreta de Cosseno (DCT - do inglês, *Discrete Fourier Transform*) é amplamente utilizada na compressão de imagem e áudio e é muito semelhante à transformadas de Fourier. A diferença é que uma DCT envolve apenas o uso de funções cosseno e coeficientes reais, enquanto as outras transformadas de Fourier utilizam fun-

ções seno e funções cosseno e exigem o uso de números complexos. Por esse motivo, as DCTs são mais simples de calcular (BERRY, 2012).

Tanto as transformadas de Fourier quanto as DCTs convertem dados de um domínio espacial em um domínio frequência. Ao comprimir sinais analógicos, muitas vezes as informações são descartadas para permitir uma compactação eficiente. Deve-se ter cuidado com quais informações em um sinal devem ser descartadas ao remover bits para compactar um sinal. A DCT ajuda neste processo, removendo os elementos de alta frequência de um sinal analógico de forma a não distorcer tanto o sinal a ponto de torná-lo irreconhecível (BERRY, 2012).

O objetivo básico quando utiliza-se uma DCT para processamento é transformar um sinal em um tipo de representação para outro. Por exemplo, transformar uma imagem, que caracteriza-se como um sinal bidimensional, em dados numéricos para que a informação da imagem exista em uma forma quantitativa e possa ser manipulada em uma compressão (RADHA; SHRUTHI, 2013).

2.3.5 Espectrograma

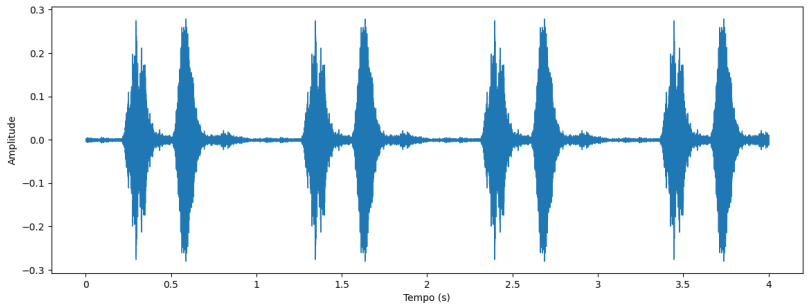
Um artifício muito utilizado na análise e representação de sinais de áudio é o espectrograma. Um espectrograma pode ser descrito como uma representação bidimensional que ilustra o contéudo de sinais de áudio, proporcionando informações relevantes sobre o comportamento do espectro no tempo. Graficamente, pode ser analisado a partir do tempo *versus* frequência, e o conteúdo do sinal é representado por uma codificação de cores (CHRISTENSEN, 2019).

Conforme apresentado na subseção 2.3.3, ao aplicar a STFT em um sinal de áudio, é possível gerar um espectrograma. Essa técnica segmenta o áudio em diversos fragmentos para analisar o conteúdo de frequência de cada um. Dado que a STFT fornece valores complexos que representam a fase e a magnitude de cada frequência, a construção do espectrograma envolve a elevação ao quadrado do valor absoluto da magnitude, descartando a informação da fase (GRIFFIN; LIM, 1984).

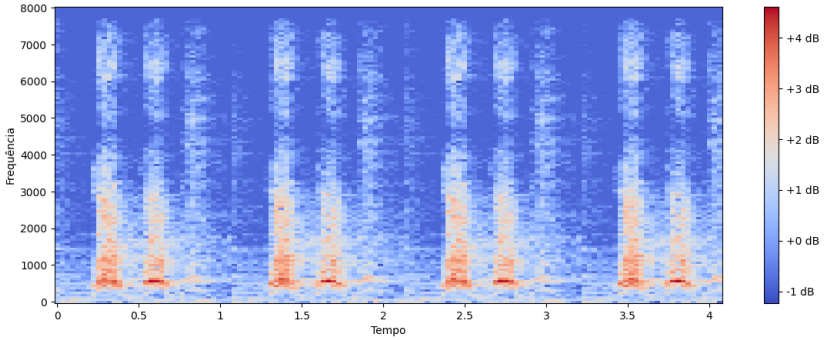
A Figura 2.2a apresenta o sinal de áudio em forma de onda, enquanto que a Figura 2.2b apresenta o sinal de áudio no formato de um espectrograma.

Figura 2.2: Gráfico de (a) Amplitude *versus* Tempo de um sinal de áudio e de (b) Frequência *versus* Tempo de um Espectrograma.

(a)



(b)



Fonte: O Autor.

2.3.6 Espectrograma em Escala *Mel*

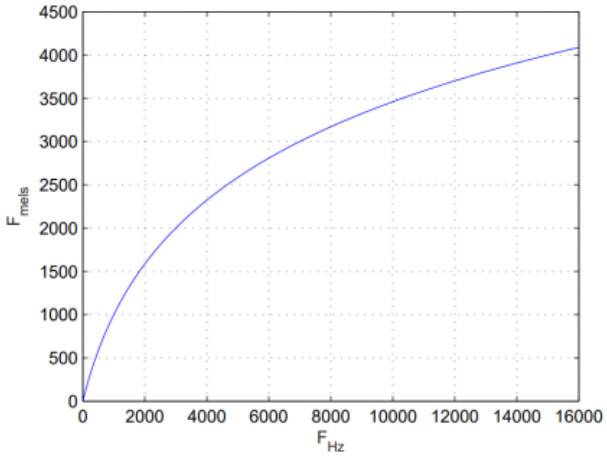
A escala mel, desenvolvida por (STEVENS; VOLKMANN; NEWMAN, 1937), propõe-se a representar a interpretação de tons de uma forma mais semelhante à audição humana, a qual identifica frequências de forma não linear, com alta precisão para bandas de baixa frequência e baixa precisão para bandas de alta frequência. A escala mel pode ser aproximada para uma forma logarítmica, a qual é apresentada na Equação 2.3

$$F_{mel} = 2595 log_{10}(1 + \frac{F_{Hz}}{700}) \tag{2.3}$$

sendo F_{mel} a frequência na escala mel e F_{Hz} a frequência medida em *Hertz*. O gráfico F_{mel} *versus* F_{Hz} é apresentado na Figura 2.3 .

Um espectrograma pode ser transformado para refletir frequências na escala mel através da aplicação de um banco de filtros triangulares. Estes filtros são distribuídos segundo a escala mel. O espectrograma resultante dessa transformação é denominado espectrograma mel. Cada filtro desse conjunto apresenta amplitude máxima de 1 em sua

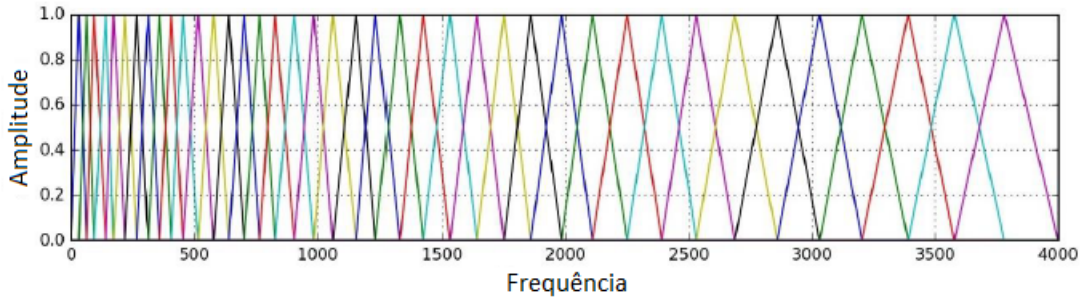
Figura 2.3: Gráfico F_{mel} *versus* F_{Hz}



Fonte: O Autor.

frequência central, diminuindo linearmente até zero nas frequências centrais dos filtros vizinhos (FAYEK, 2016). A Figura 2.4 apresenta uma representação de um banco de 40 filtros, ajustados conforme a escala de frequência mel.

Figura 2.4: Representação gráfica de Amplitude *versus* Frequência de um banco de 40 filtros conforme a escala mel



Fonte: Adaptado de (FAYEK, 2016).

2.3.7 Coeficientes Cepstrais de Frequência *Mel*

Os Coeficientes Cepstrais de Frequência *Mel* (MFCC - do inglês, *Mel Frequency Cepstral Coefficients*) são uma representação compacta de um sinal de áudio, amplamente utilizados para classificações realizadas em dispositivos com recursos limitados. Os MFCCs são uma forma eficiente de extrair características sonoras de espectrogramas mel. Ao utilizar os MFCCs para identificar sons ambientais, nota-se um bom desempenho com baixo custo computacional. Isso ocorre porque os MFCCs diminuem a dimensionalidade para um pequeno número de coeficientes com baixa correlação, resultando em uma

boa relação entre precisão e complexidade (PORTELO et al., 2009) (ANUSUYA; KATTI, 2011).

Os MFCCs são obtidos através da aplicação de uma DCT calculada a partir de um espectrograma mel. A Equação 2.4 apresenta o cálculo para obtenção desses coeficientes

$$c[n] = \sum_{m=0}^{M-1} log_{10}(s(m))cos(\frac{\pi(m-0.5)}{M}) \tag{2.4}$$

para $n = 0, 1, 2, ..., C-1$, onde $c[n]$ representa os coeficientes cepstrais, $s(m)$ representa o espectro mel, M é o número total de filtros triangulares mel e C é o número de MFCCs. De forma geral, são utilizados de 8 a 13 coeficientes MFCCs (RAO; VUPPALA, 2014).

2.4 Inteligência Artificial e Aprendizado de Máquina

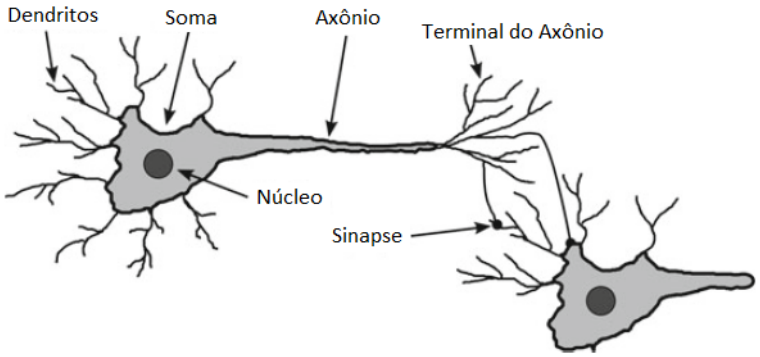
A inteligência artificial (IA - do inglês, *Artificial Intelligence*) é uma área de estudo que engloba técnicas de aprendizado de máquina e aprendizado profundo. Adicionalmente, também abrange estratégias que não se baseiam exclusivamente em processos de aprendizado. Por exemplo, os primeiros *softwares* de xadrez, que eram fundamentados em regras pré-estabelecidas por desenvolvedores, se enquadram no âmbito da inteligência artificial, porém não utilizavam princípios de aprendizado de máquina (CHOLLET, 2017).

Conforme mencionado, aprendizado de máquina é uma subárea da inteligência artificial que busca criar sistemas que consigam aprender de maneira autônoma a partir da utilização de métodos computacionais. Um sistema de aprendizado faz escolhas baseadas em conhecimentos adquiridos ao resolver desafios passados com êxito (MONARD; BARANAUSKAS, 2003).

2.5 Redes Neurais Artificiais

Uma rede neural artificial (RNA) é um grupo interconectado de unidades computacionais menores, chamadas neurônios, que tentam imitar o comportamento de neurônios biológicos (AGHDAM; HERAVI, 2017). A Figura 2.5 apresenta uma estrutura simplificada de um neurônio biológico.

Figura 2.5: Diagrama simplificado de um neurônio biológico.



Fonte: Adaptado de (AGHDAM; HERAVI, 2017)

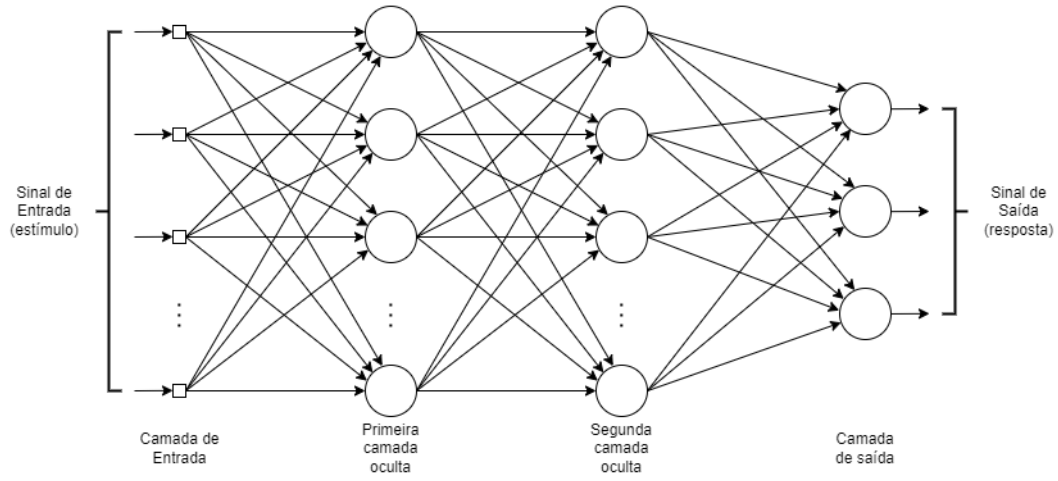
Um neurônio biológico é composto por quatro componentes essenciais, as quais são nominadas como dendritos, soma, axônio e núcleo. Os dendritos funcionam como entradas do neurônio, conectando-se a sensores ou a outros neurônios através das sinapses. Se as entradas ultrapassam um determinado limiar, provocam uma sequência de impulsos pelo axônio. Quando o sinal é gerado, o núcleo retorna à sua condição normal. Nesse estado de equilíbrio, a sequência de impulsos cessa e os sinais gerados são enviados a um outro neurônio pelos terminais do axônio. Por fim, as ligações sinápticas facilitam a transmissão de sinais entre os neurônios (AGHDAM; HERAVI, 2017).

2.5.1 Perceptron Multicamadas

O Perceptron Multicamadas (MLP, sigla em inglês para Multilayer Perceptron) tem a capacidade de ter múltiplas camadas de processamento. Diferente de um perceptron simples, que possui somente uma camada de entrada e saída e onde todos os cálculos são expostos ao usuário, o MLP inclui várias camadas de cálculo. Estas camadas adicionais são denominadas "camadas ocultas", pois os cálculos feitos nelas permanecem inacessíveis ao usuário (AGGARWAL, 2018).

A Figura 2.6 ilustra a configuração de uma rede neural multicamadas com conexão completa, onde um neurônio de uma camada liga-se aos neurônios da camada anterior. A rede em questão possui duas camadas ocultas e uma camada de saída.

Figura 2.6: Diagrama de uma rede neural multicamadas com duas camadas ocultas.



Fonte: Adaptado de (HAYKIN, 2003).

A rede MLP é classificada como uma rede *feedforward*. Essas redes se caracterizam por terem as saídas dos neurônios de uma camada ligando-se unicamente aos neurônios da camada subsequente. Dessa forma, a entrada se propaga, de forma progressiva, através da rede. De acordo com Haykin (2003), as redes MLP são versáteis e podem ser empregadas em uma variedade de desafios de alta complexidade, utilizando o treinamento supervisionado com o auxílio do algoritmo *backpropagation*. O algoritmo de *backpropagation* opera em dois estágios: a fase de avanço (*forward*) e a fase de retrocesso (*backward*).

2.5.2 Redes Neurais Convolucionais

A principal motivação na escolha da utilização de Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Networks*) na habilidade de desenvolver uma solução que minimize a quantidade de parâmetros, permitindo a utilização de uma rede mais profunda com uma quantidade muito menor de parâmetros (AGHDAM; HERAVI, 2017). Enquanto uma rede neural multicamadas padrão aprende a partir de padrões globais, como por exemplo, todos os *pixels* de uma imagem, uma rede convolucional aprende a partir de padrões locais, como por exemplo, bordas cinza em uma imagem e texturas (CHOLLET, 2017).

Na literatura científica, é possível encontrar arquiteturas variadas de CNNs. No entanto, a estrutura padrão é formada por três camadas fundamentais, as quais são no-

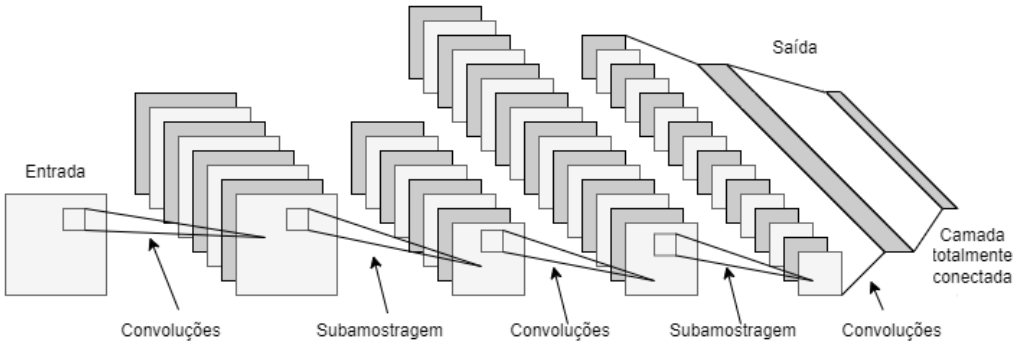
minadas como camada convolucional (*convolutional layer*), camada de *pooling* (pooling layer) e camada totalmente conectada (fully connected layer).

As camadas convolucionais capturam características distintas das entradas por meio de diversos filtros de convolução. Esses filtros reconhecem detalhes específicos dos dados de entrada, como a detecção de um rosto em uma imagem (CHOLLET, 2017).

A camada de *pooling* tem a finalidade de diminuir a resolução espacial dos mapas de características durante o aprendizado, o que é por vezes referido como subamostragem ou *downsampling*. Esse processo permite representar padrões variáveis em tamanho presentes nas imagens. O método de subamostragem é frequentemente referido como *stride*. Geralmente, operações de *pooling* são inseridas entre camadas convolucionais consecutivas. As operações padrão nesta camada são *max pooling*, o qual escolhe o valor mais alto de um campo específico, e *average pooling*, o qual determina a média dos valores desse campo. (BOUREAU; PONCE; LECUN, 2010).

As camadas totalmente conectadas são colocadas após as camadas convolucionais e de *pooling* e tem por finalidade interpretar as características de alto nível de abstração.

Figura 2.7: Estrutura de uma Rede Neural Convolucional.



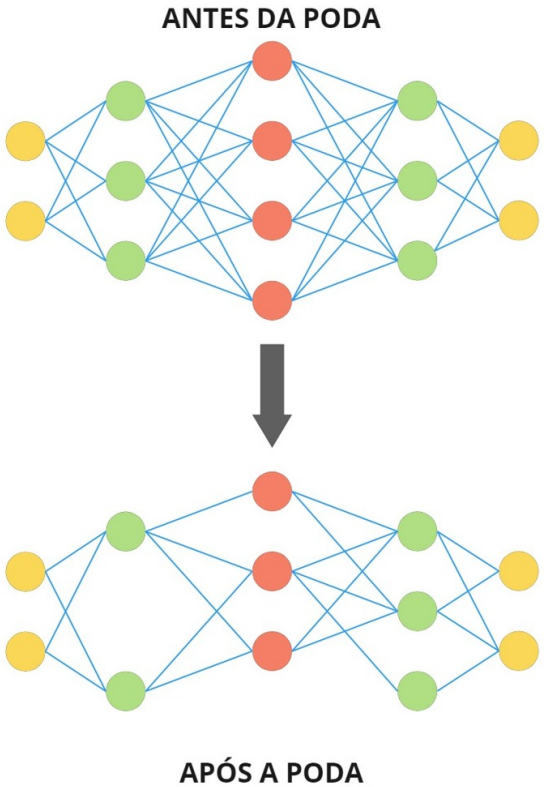
Fonte: Adaptado de (LECUN; KAVUKCUOGLU; FARABET, 2010).

Na Figura 2.7 é apresentada a estrutura de uma Rede Neural Convolucional com a entrada sendo uma imagem $2D$. Essa imagem passa por uma camada convolucional e, em seguida, por uma camada de *pooling*, onde é aplicada uma subamostragem nas características (*features*) mapeadas pela primeira camada convolucional. Na sequência, são aplicadas mais uma camada convolucional e de *pooling*, finalizando com uma camada totalmente conectada e uma camada de saída.

2.5.3 Poda Computacional

A técnica de poda (do inglês, *pruning*) computacional envole a remoção de trechos que não são necessários um modelo de aprendizado de máquina com o intuito de otimizá-lo. Em Redes Neurais Artificiais, esse processo ocorre através da remoção de sinapses entre neurônios, podendo resultar, inclusive, na remoção do próprio neurônio, caso não permaneça nenhuma ligação de outro neurônio com ele. A Figura 2.8 traz uma representação gráfica dessa técnica. Esse processo ajuda também a evitar o *overfitting* do modelo, removendo pesos no início do treinamento, não deixando que eles sejam reativados novamente.

Figura 2.8: Poda computacional com remoção de sinapses e neurônios.



Fonte: O Autor.

A poda busca induzir esparsidade nas matrizes de uma conexão de uma rede neural, reduzindo, assim, o número de parâmetros com valor diferente de zero no modelo. (ZHU; GUPTA, 2017).

2.5.4 Quantização

O método de quantização em aprendizado de máquina refere-se a uma técnica usada para reduzir a complexidade e o tamanho dos modelos, especialmente em cenários onde o poder computacional ou a capacidade de armazenamento são limitados. A ideia principal é representar os parâmetros do modelo com menos bits do que o usual, sem prejudicar a eficiência.

Quando os modelos de aprendizado de máquina são treinados, os pesos e parâmetros geralmente são armazenados em formato de ponto flutuante de 32 bits, o que permite uma alta precisão, mas também resulta em altos requisitos computacionais e de memória. A quantização reduz essa quantidade de bits necessários para representar os parâmetros do modelo, como através de inteiros de 8 bits. Isso reduz significativamente o tamanho do modelo e torna sua execução mais eficiente, podendo ter valores de latência até 4x menores quando comparados aos modelos com ponto flutuante de 32 bits (SIVAKUMAR et al., 2019).

2.6 Aprendizado de Máquina em Sistemas Embarcados

A implementação de técnicas de aprendizado de máquina é tipicamente realizada em servidores CPUs ou GPUs em nuvem, com alta capacidade de processamento e alto consumo de energia. Essa configuração acaba tornando inviável a utilização de dispositivos com recursos limitados por uma série de fatores, como problemas de latência, privacidade de dados e consumo de energia. Por esses motivos, busca-se constantemente a simplificação de modelos de aprendizado de máquina para possibilitar a implementação em *hardwares* limitados, para realização da fase de inferência do modelo, respeitando o limite de memória e capacidade de processamento do dispositivo, e diminuindo o tempo de execução e o consumo de energia (MOONS; BANKMAN; VERHELST, 2019).

Para dispositivos com recursos limitados, o principal caso de utilização para sistemas de aprendizado consiste em treinar o modelo em dispositivos com alta capacidade de processamento ou em nuvem e realizar a implementação do modelo no microcontrolador para realização da inferência. Existem diversas ferramentas dedicadas para conversão de modelos de aprendizado em estruturas que possam ser executadas em um microcontrolador. Abaixo, estão listadas algumas dessas ferramentas:

- CMSIS-NN: é uma biblioteca de baixo nível para microcontroladores da linha Cortex-M a qual implementa blocos básicos para construção de redes neurais, como redes neurais convolucionais $2D$.
- X-CUBE-AI: complemento do kit de desenvolvimento de software STM32CubeMX, que permite carregar modelos de treinamento implementados utilizando bibliotecas como Keras, Tensorflow, PyTorch e outros.
- uTensor: permite rodar uma quantidade reduzida de modelos da biblioteca TensorFlow em microcontroladores da linha ARM Cortex-M.
- Tensorflow Lite (TFLite): é uma versão "leve"do Tensorflow. Foi desenvolvido para atender aplicações que envolvem aprendizado de máquina em sistemas embarcados. Um modelo de aprendizado de máquima pode ser treinado utilizando a biblioteca Tensorflow, convertido para o formato *.tflite*, embarcado em um dispositivo que tenha suporte ao TFLite e a partir disso, pode-se realizar inferências com auxílio da biblioteca TFLite.

2.7 Base de Dados

Dentro do espectro de pesquisa de classificação de sons ambientais, existem bases de dados consistentes que são amplamente utilizadas para aplicação de técnicas de aprendizado. As subseções 2.7.1, 2.7.2 e 2.7.4 apresentarão as três bases mais difundidas na literatura científica.

2.7.1 ESC-50

A base de dados ESC-50 foi apresentada em Piczak (2015b) e possui 2000 gravações de áudio ambiental. Esse conjunto de dados consiste em gravações de 5 segundos com uma frequência de amostragem de $44.1kHz$ em 50 classes, com 40 exemplos por classe, organizadas em 5 categorias principais: "animais", "paisagens sonoras e sons da água", "humanos (sons sem fala)", "sons internos/domésticos"e "ruídos externos/urbanos".

2.7.2 ESC-10

A base de dados ESC-10 é um subconjunto da base de dados ESC-50. É dividida em 10 classes (totalizando 400 amostras) nominadas como: "latido de cachorro", "chuva", "ondas do mar", "bebê chorando", "tique-taque do relógio", "espirro de ser humano", "helicóptero", "motoserra", "galo"e "fogo crepitante". Ao todo, são 400 amostras de áudio com duração de 5 segundos cada amostra.

2.7.3 ESC-US

A base de dados ESC-US consiste em um conjunto de 250 mil amostras de áudio ambientais não rotuladas, com duração de 5 segundos cada áudio. Essa base de dados é adequada para pré-treinamento não supervisionado.

2.7.4 Urbansound8k

A base de dados Urbansound8K é apresentada em Salamon, Jacoby e Bello (2014a). Contém 8,75 horas de áudio de eventos sonoros encontrados em ambiente urbano, rotulados manualmente. É composta por 8732 trechos de áudio, de até 4 segundos cada, de sons urbanos divididos em dez classes, as quais são nominadas como: "ar condicionado", "buzina de carro", "crianças brincando", "latido de cachorro", "perfuração", "motor em marcha lenta", "tiro", "britadeira", "sirene"e "música de rua".

A base de dados UrbanSound8K foi dividida em 10 dobras (*folds*) pelo próprio autor da base de dados, o qual recomenda a utilização da técnica *10-fold cross validation* para treinamento e validação de modelos de aprendizado de máquina, sem realizar alterações na disposição de áudios e *folds* realizadas, uma vez que os áudios da base de dados em questão não são triviais e para permitir uma comparação adequada entre trabalhos científicos. Ainda, vale ressaltar que a base de dados UrbanSound8K foi a escolhida para a utilização na presente pesquisa.

2.8 Trabalhos Relacionados

Esta seção tem o propósito de analisar trabalhos relacionados com o tema de classificação de sons ambientais, e está segmentada em duas partes: uma que trata da classificação de sons ambientais por meio de técnicas de aprendizado de máquina sem restrições de *hardware*, e outra focada na categorização de sons do ambiente com técnicas de aprendizado de máquina implementadas em *hardwares* de capacidades restritas.

Para a análise, considerou-se as bases de dados apresentadas nas subseções 2.7.1, 2.7.2 e 2.7.4, por serem as mais difundidas na literatura quando o enfoque é classificação de sons ambientais.

2.8.1 Classificação de Sons Ambientais

Uma grande quantidade de trabalhos apresentaram modelos de aprendizado de máquina para classificação de sinais de áudio que alcançaram a performance do estado-da-arte, considerando as bases de dados ESC-50, ESC-10 e Urbansound8k. Entretanto, dentre esses trabalhos, poucos relataram completamente os tamanhos dos modelos utilizados e os requisitos de processamento e memória necessários.

Em Piczak (2015b), apresentou-se o detalhamento do desenvolvimento da base de dados ESC-50 e o respectivo subconjunto ESC-10, e foram utilizadas algumas técnicas de classificação, como *K-Nearest Neighbors* (KNN), *Random Forest* (RF) e *Support Vecotr Machine* (SVM), para classificação dessas bases de dados. A técnica que apresentou o melhor resultado foi a RF, com 72,7% de acerto para a base ESC-10 e 44.3% para a base ESC-50. O mesmo autor, em Piczak (2015a), utilizou a técnica CNN, obtendo 90.2% de taxa de acerto para a base ESC-10, 64.5% para a base ESC-50 e 73.7% para a base de dados Urbansound8k.

Os trabalhos apresentados na Tabela 2.1, com exceção de Piczak (2015b), utilizaram a técnica CNN como modelo base, variando as técnicas de extração de características dos áudios. Levou-se em consideração o melhor resultado obtido em cada pesquisa, independente das técnicas de extração de características adotadas.

Os melhores resultados registrados na literatura científica para a base de dados ESC-10 variam de 66.7% à 97.0%. A Tabela 2.1 apresenta alguns resultados, levando em consideração a taxa de acerto percentual e o modelo base de classificação adotado.

Tabela 2.1: Comparação de taxas de acerto de trabalhos relacionados com a base de dados ESC-10.

Pesquisa	**Modelo**	**Taxa de Acerto (%)**
Piczak (2015b)	KNN	66,7
Piczak (2015b)	RF	72,7
Piczak (2015b)	SVM	67,5
Piczak (2015a)	CNN	90,2
Boddapati et al. (2017)	AlexNet	86,0
Boddapati et al. (2017)	GoogLeNet	91,0
Zeng et al. (2017)	CNN	94,2
Tokozume et al. (2017)	EnvNet	86,8
Tokozume et al. (2018)	EnvNet2	88,8
Qin et al. (2018)	CNN	91,2
Balamurugan et al. (2019)	CNN	94,2
Mushtaq & Su (2020)	DCNN	94,9
Mohaimenuzzaman et al. (2021)	ACDNet	96,6
Lujie et al. (2022)	CNN	97,0

Fonte: O Autor.

As pesquisas que utilizaram a base de dados ESC-50 são mostradas na Tabela 2.2, de acordo com a taxa de acerto e o modelo base de classificação. Da mesma forma que foi apresentado na Tabela 2.1, com exceção das técnicas KNN, RF e SVM apresentadas por (PICZAK, 2015b), todas os outros trabalhos apresentam técnicas baseadas em CNN. Os melhores resultados obtidos para essa base de dados variam de 32.2% à 91.2%.

Tabela 2.2: Comparação de taxas de acerto de trabalhos relacionados com base de dados ESC-50.

Pesquisa	**Modelo**	**Taxa de Acerto (%)**
Piczak (2015b)	KNN	32.2
Piczak (2015b)	RF	44.3
Piczak (2015b)	SVM	39.6
Piczak (2015a)	CNN	64.5
Zeng et al. (2017)	CNN	86.5
Boddapati et al. (2017)	AlexNet	65
Boddapati et al. (2017)	GoogLeNet	73
Tokozume & Harada (2017)	EnvNet	66.4
Tokozume & Harada (2018)	EnvNet2	81.6
Qin et al. (2018)	CNN	91.2
Balamurugan et al. (2019)	CNN	84.0
Li et al. (2019)	CNN	86.2
Mushtaq & Su (2020)	DCNN	86.5
Mohaimenuzzaman et al. (2021)	ACDNet (CNN)	87.1
Lujie et al. (2022)	CNN	88.3

Fonte: O Autor.

Para a base de dados Urbansound8k, todos os trabalhos apresentados na Tabela 2.3 utilizaram CNN como modelo base, com taxas de acerto variando de 73.7% à 93.0%.

Tabela 2.3: Comparação de taxas de acerto de trabalhos relacionados com a base de dados Urbansound8k.

Pesquisa	**Modelo**	**Taxa de Acerto (%)**
Piczak CNN (2015a)	CNN	73,7
Boddapati et al. 2017)	AlexNet	92,0
Boddapati et al. (2017)	GoogLeNet	93,0
Salamon & Bello (2017)	SBCNN	79,0
Qin et al. (2018)	CNN	85,1
Zhang et al. (2018)	D-MIX (CNN)	83,7
Li et al. (2019)	CNN	83,4
Zhichao et al. (2019)	CNN	76,0
Mushtaq & Su (2020)	DCNN	86,5
Lujie et al. (2022)	CNN	83,5

Fonte: O Autor.

2.8.2 Classificação de Sons Ambientais em Sistemas Embarcados para as Bases de Dados ESC-50 e ESC-10

O número de trabalhos relacionados à classificação de sons ambientais reduz significativamente quando envolve implementação em sistemas embarcados. Isso se deve ao fato de que, para a implementação de modelos em *hardwares* limitados, é imprescindível a aplicação de técnicas de poda computacional e quantização, o que acaba, muitas vezes, impactando no desempenho do modelo.

Dentre os trabalhos analisados na subseção 2.8.1, o trabalho proposto por Peng et al. (2022), foi o que apresentou menor quantidade de parâmetros utilizados, totalizando $340k$ parâmetros para uma taxa de acerto de 97.0% na base de dados ESC-10. O modelo proposto por Kumari et al. (2019), nominado Edge-L^3, teve como objetivo principal reduzir os requisitos computacionais do modelo L^3-Net, proposto em Arandjelović e Zisserman (2017). Com a poda do modelo, reduziu-se o número de parâmetros de aproximadamente $4.7M$, para aproximadamente $275k$ parâmetros. Dessa forma, também reduziu-se a quantidade de memória Flash (não-volátil) requerida de $18MB$ para $0.814MB$.

Em Mohaimenuzzaman et al. (2021), além do modelo ACDNet, citado na subseção 2.8.1, foi apresentado o modelo Micro-ACDNet, após poda e quantização do modelo

ACDNet. O número de parâmetros utilizados reduziu de $4.74M$ para $0.131M$, enquanto a quantidade de memória não volátil requerida reduziu de $18MB$ para $0.5MB$, obtendo baixa diminuição na taxa de acerto, decaindo de 87.1% para 83.7% para a base de dados ESC-50, o que já pode se considerar plausível para uma aplicação em MCUs.

Já em Andreadis, Giambene e Zambon (2021a), implementou-se uma CNN em MCU de 32 bits da linha ARM-Cortex M4F, com $1MB$ de memória Flash e $256kB$ de memória SRAM. Comparou-se três técnicas de pré-processamento: Espectrograma linear, espectrograma mel e MFCC. O melhor resultado obtido, considerando a base de dados ESC-10, apresentou taxa de acerto de 74.2%, com a utilização da técnica MFCC e quantização para processamento em ponto fixo de 8 bits, apresentando tempo de inferência de $110ms$ e pico de memória RAM de $23.2kB$.

2.8.3 Classificação de Sons Ambientais em Sistemas Embarcados para a Base de Dados UrbanSound8K

Assim como para as bases de dados ESC-50 e ESC-10, o número de trabalhos envolvendo a base de dados UrbanSound8K é reduzido quando a abordagem é sistemas embarcados. Em Nordby (2019), foi proposto um modelo de CNN de $113,6k$ parâmetros para classificações em tempo real em um microcontrolador ARM Cortex-M4F. Foi utilizada a técnica de extração de características espectrograma mel em escala logarítmica e obteve-se uma taxa de acerto média de $70,9\%$. Foram utilizadas técnicas de *data augmentation*, como *time-stretching* e *pitch-shifting*. Como foi utilizada a técnica *10-fold cross validation*, foi escolhido um dos 10 modelos treinados para embarcar em hardware, o qual obteve $72,0\%$ de taxa de acerto no conjunto validação.

Em Shah, Tariq e Lee (2019), foi proposto um modelo de CNN para monitoramento de ruído urbano, o qual foi embarcado em uma *Raspberry Pi 4*. Para efeitos de validação do modelo, utilizou-se a base de dados UrbanSound8K, o qual obteve aproximadamente $74,0\%$ de taxa de acerto. Foram utilizadas técnicas de *data augmentation*, bem como diferentes técnicas de normalização para as características extraídas dos áudios. Espectrogramas lineares foram extraídos dos áudios para serem utilizados como dados de entrada da CNN proposta, a qual possui aproximadamente $241,5k$ parâmetros.

Em Vandendriessche et al. (2021), foram propostas duas topologias de CNN. A primeira consiste em uma CNN unidimensional de $102,9k$ parâmetros. A segunda consiste em uma CNN bidimensional baseada na topologia apresentada em Salamon e Bello

(2017), com algumas reduções de dimensionalidade, o qual resultou em um modelo base de aproximadamente $115,5k$ parâmetros. Para a CNN unidimensional, foi utilizada a combinação de técnicas de extração de características como espectrograma mel, MFCC e cromagrama. Já para a CNN bidimensional, foi utilizado espectrograma mel em escala logarítmica. Após treinados, os modelos foram submetidos à técnicas de poda computacional e quantização, para posterior utilização em hardwares com restrição de recursos. Foram utilizados os *hardwares Raspberry Pi 4B*, dois FPGAs *Xilinx* (*Pynq-z2* e *ZCU104*), o dispositivo *USB Coral TPU* e a placa de desenvolvimento *Coral*. Foi obtido taxa de acerto média de $63,88\%$ para o modelo base da CNN bidimensional e taxa de acerto média de $60,06\%$ para o modelo base da CNN unidimensional.

3 METODOLOGIA

O presente capítulo tem como objetivo apresentar a solução implementada para a classificação de sons ambientais a partir da utilização de redes neurais convolucionais e aplicação de técnicas de poda computacional e quantização, a fim de reduzir o número de parâmetros e o tamanho dos modelos.

3.1 Materiais e Ferramentas

3.1.1 Ambiente de Desenvolvimento

Como ambiente de desenvolvimento, optou-se por utilizar a plataforma *Google Colaboratory* uma vez que esta permite utilizar serviço de nuvem gratuito, e oferece recursos de *hardware*, tais como GPUs e TPUs, para a execução dos códigos desenvolvidos. Além disso, permite a utilização da linguagem Python, a qual será utilizada para implementação dos modelos de aprendizado de máquina. Ainda, oferece integração com as bibliotecas utilizadas para a implementação do projeto, as quais estão listadas a seguir:

- Plataforma *TensorFlow*, versão 2.12.0 – utilizada para implementação das redes neurais convolucionais;
- Biblioteca *Numpy*, versão 1.22.4 – utilizada para manipulações matemáticas;
- Biblioteca *Pandas*, versão 1.5.3 – utilizada para visualização e manipulação da base de dados;
- Biblioteca *Matplotlib*, versão 3.7.1 – utilizada para análise e visualização de dados;
- Biblioteca *librosa*, versão 0.10.0 – utilizada para manipulação e processamento de áudio;
- Biblioteca *scikit-learn*, versão 1.2.2 – utilizada para implementação da técnica de validação das topologias selecionadas.

3.1.2 Especificações de *Hardware*

O ambiente *Google Colaboratory* permite escolher entre a utilização de *CPU* ou *GPU* para execução dos testes. A fim de utilizar o máximo da capacidade da ferramenta,

escolheu-se a utilização da *GPU*. As especificações de *hardware* apresentadas na Tabela 3.1 foram utilizadas para o treinamento e validação dos modelos.

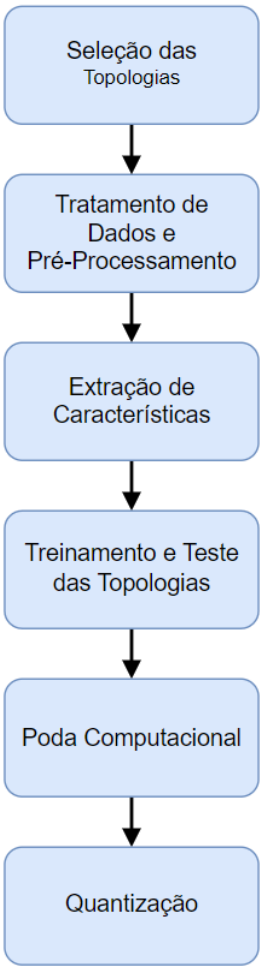
Tabela 3.1: Especificações de *hardware* utilizadas para treinamento e validação dos modelos no ambiente *Google Colaboratory*.

CPU	**Memória RAM**	**GPU**
Intel Xeon 2,0 GHz	12,7 GB	V100 - 16GB

3.2 Procedimento

O fluxograma correspondente à solução implementada está apresentado na Figura 3.1. Cada etapa que compõe o procedimento adotado será explanada nas subseções futuras.

Figura 3.1: Fluxograma correspondente à Solução Implementada.



Fonte: O Autor.

3.2.1 Seleção das Topologias

A seleção das topologias se deu a partir da análise dos trabalhos científicos apresentados nas subseções 2.8.1 e 2.8.2. Para efeitos comparativos, selecionou-se duas topologias apresentadas em pesquisas que realizaram a implementação apenas em hardware sem restrição de recursos. Para essas, escolheu-se os modelos com número de parâmetros inferior a 5 milhões e taxa de acerto superior a 65%. Ainda, selecionou-se 3 topologias de pesquisas que realizaram a implementação em hardwares sem e com restrição de recursos. Como critério de escolha, optou-se pelas topologias que apresentaram número de parâmetros inferior a 1 milhão (sem poda computacional), taxa de acerto superior a 55% e que apresentaram resultados de tempo de latência para os hardwares abordados. As cinco topologias selecionadas estão listadas nas subseções 3.2.1.1 à 3.2.1.5. A Tabela 3.2 apresenta o número de parâmetros para cada topologia selecionada.

Tabela 3.2: Número de parâmetros de cada topologia selecionada.

Pesquisa	**Topologia**	**Nº de Parâmetros**
Salamon & Bello (2017)	SBCNN	432.378
Zhang et al. (2018)	D-MIX	4.722.538
Nordby (2019)	STRIDED	749.460
Shah, Tariq & Lee (2019)	2DCNN	241.434
Vandendriessche et al. (2021)	1DCNN	88.210

Fonte: O Autor.

3.2.1.1 Topologia 1 - SBCNN

A topologia apresentada em Salamon e Bello (2017) é composta por uma CNN bidimensional, com os dados de entrada sendo uma matriz MxNxP onde M representa as características extraídas para cada janela de tempo N, e P representa o número de canais. O número total de parâmetros da topologia em questão é apresentado na Tabela 3.2 corresponde a dados de entrada no formato 128x128x1, o que é o caso da presente pesquisa. O detalhamento a respeito dos dados de entrada é apresentado na Subseção 3.2.3.

3.2.1.2 Topologia 2 - D-MIX

A topologia apresentada em Zhang et al. (2018) é uma CNN bidimensional e foi escolhida pelo fato de possuir um número elevado de parâmetros quando comparada com as outras topologias selecionadas. Dessa forma pode-se avaliar o impacto do aumento do número de parâmetros na performance do modelo. Vale ressaltar que o autor da topologia em questão realizou o treinamento sem e com a configuração *mixup*, a qual consiste em utilizar dois canais na entrada da CNN, onde cada canal é composto por um tipo de extração de características. No presente trabalho, considerou-se apenas a topologia sem configuração *mixup*, mantendo assim o formato de entrada de 128x128x1 para a CNN.

3.2.1.3 Topologia 3 - STRIDED

A topologia proposta em Nordby (2019) é uma CNN bidimensional e foi escolhida pelo fato de ter sido implementada para ser embarcada em *hardware* com restrição de recursos. Escolheu-se a estrutura do modelo de CNN que obteve maior taxa de acerto na pesquisa em questão. É importante ressaltar que na pesquisa em questão não foram aplicadas técnicas de poda e quantização, e que os dados de entrada para CNN constituíam o formato 60x31x1 ($102,3k$ parâmetros), com a utilização de espectrogramas mel em escala logarítmica. Na presente pesquisa, optou-se por manter o formato de entrada fixo para todas as topologias. Dessa forma utilizou-se o formato 128x128x1, o que impacta no número de parâmetros do modelo, aumentando para aproximadamente $749,5k$. Entretanto, como serão aplicadas técnicas de poda e quantização, torna-se viável a utilização da topologia para efeitos de comparação.

3.2.1.4 Topologia 4 - 2D-CNN

A topologia apresentada em Shah, Tariq e Lee (2019) é uma CNN bidimensional e foi selecionada por possuir taxa de acerto satisfatória para um número de parâmetros consideravelmente pequeno, além de também ter sido projetada para aplicação em *hardware* com recursos limitados. O formato dos dados de entrada utilizados na pesquisa em questão também constituíam o formato 128x128x1.

3.2.1.5 Topologia 5 - 1DCNN

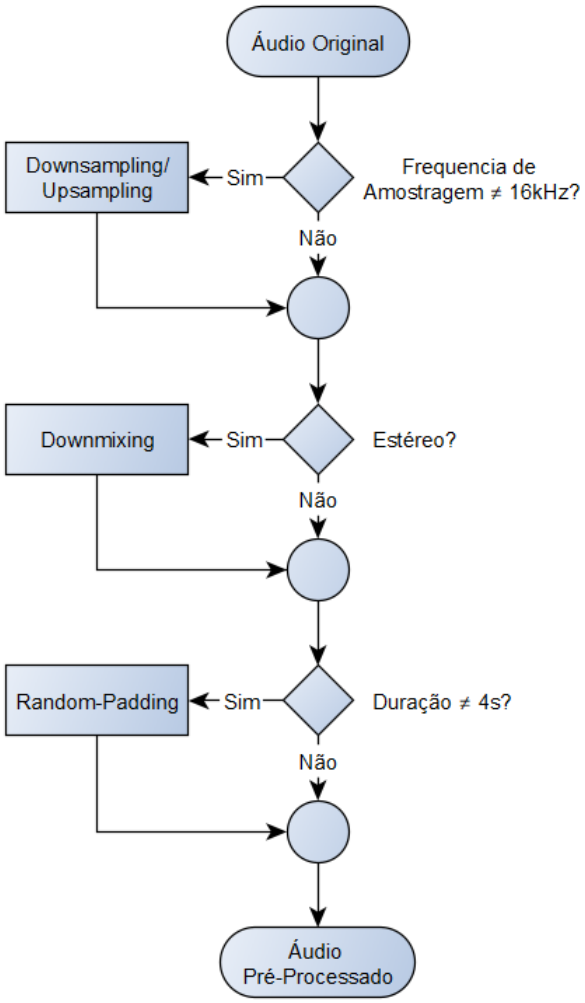
A topologia apresentada em Vandendriessche et al. (2021) é composta por uma CNN unidimensional. A estrutura da CNN é composta por uma camada de entrada que recebe vetores de dimensão Mx1, sendo M o número de recursos de cada extração de características. Essa topologia foi escolhida haja vista o número reduzido de parâmetros quando comparada às demais topologias selecionadas. Ainda, vale ressaltar que na pesquisa em que a topologia 1DCNN foi proposta, foi utilizada uma combinação de técnicas de extração de características, obtendo uma entrada no formato 193x1. Já na presente pesquisa, optou-se por utilizar o formato de entrada 128x1 para essa topologia.

3.2.2 Tratamento de Dados e Pré-Processamento

A base de dados *UrbanSound8K* possui 8732 amostras de áudio, todas em formato *.WAV*, e é segmentada em 10 classes distintas, conforme descrito na subseção 2.7.4. As amostras de áudio contidas nessa base de dados possuem diferentes configurações quanto à frequência de amostragem, duração do áudio e número de canais. A fim de manter a consistência da análise, se fez necessária a padronização dos dados a partir do pré-processamento das amostras de áudio.

A Figura 3.2 apresenta o fluxograma do pré-processamento das amostras de áudio para padronização dos dados a serem analisados. As subseções 3.2.2.1, 3.2.2.2 e 3.2.2.3 descrevem com maiores detalhes as técnicas aplicadas para o pré-processamento dos áudios.

Figura 3.2: Fluxograma de pré-processamento dos sinais de áudio.



Fonte: O Autor.

3.2.2.1 Downsampling e Upsampling

A base de dados *UrbanSound8K* possui amostras de áudio com diferentes frequências de amostragem, variando de $8kHz$ até $192kHz$. A frequência de $16kHz$ foi definida pelo autor da monografia como a frequência de amostragem padrão para todos os áudios, uma vez que tem-se como objetivo a redução do custo computacional do processo de extração de características e essa frequência de amostragem se prova apropriada para a classificação de sons ambientais (ANDREADIS; GIAMBENE; ZAMBON, 2021b). Dessa forma, para sinais de áudio com taxa de amostragem inferior a $16kHz$ aplicou-se a técnica de *Upsampling*, enquanto que para sinais de áudio com frequência de amostragem superior a $16kHz$ aplicou-se a técnica de *Downsampling*.

3.2.2.2 Downmixing

A base de dados utilizada consiste predominantemente em trechos de áudio com configuração estéreo, isto é, com dois canais. Para efeitos de otimização, empregou-se a técnica de *downmixing*, a qual combina os dois canais presentes nas amostras estéreo, convertendo-os em uma configuração mono, que é representada por um único canal.

3.2.2.3 Random-Padding

Para o problema de amostras de áudio com durações distintas, tomou-se como base a pesquisa feita no trabalho apresentado em (DONG et al., 2020), a qual aplica a técnica de *Random-Padding*. Essa técnica consiste em realizar o preenchimento das amostras de áudio a fim de alcançar a duração desejada. Para o presente trabalho, definiu-se a duração de 4 segundos para todas as amostras de áudio. A Figura 3.3 apresenta o pseudo-código da técnica em questão.

Em linhas gerais, para áudios com duração de até 2 segundos, a função repete o áudio original o número necessário de vezes (até o limite superior) para preencher a duração restante. Por outro lado, se a duração do áudio for maior do que 2 segundos e menor do que 4 segundos, a função seleciona aleatoriamente um segmento do áudio original com a duração restante para alcançar os 4 segundos definidos. Essa é uma alternativa à técnica *Zero-Padding*, a qual adiciona zeros para realizar o preenchimento dos sinais de áudio.

Figura 3.3: Pseudo-Código do método *Random-Padding*.

Algorithm 1 Random-Padding

Input: y_n:raw audio; $freq$:sampling frequency;
Output: Y_{pad}:after the random-padding of the raw audio;

$n \leftarrow len(y_n)$; $n_{all} \leftarrow 4 * freq$;
$n_{lack} \leftarrow n_{all} - y_n$; $t \leftarrow y_n/freq$;
if $t > 0$ **and** $t \leq 2$ **then**
    $k \leftarrow ceil(n_{lack}/n)$;
    $Y_{pad} \leftarrow copy\ the\ y_n\ k\ times$;
return $Y_{pad}[: n_{all}]$
else
    $point \leftarrow random.choice(1 : (y_n - n_{lack}))$;
    $Y_{pad} \leftarrow y_n[point : point + n_{lack}]$;
return Y_{pad}
end if

Fonte: Retirado de (DONG et al., 2020).

3.2.3 Extração de Características

As técnicas de extração de características selecionadas para serem aplicadas nos áudios pré-processados foram Espectrograma em escala logarítmica, Espectrograma Mel e Espectrograma Mel em escala logarítmica.

O número de pontos utilizados na STFT (*nfft*) foi definido como 512. Além disso, a quantidade de deslocamento ao analisar o sinal de áudio através de uma janela, também conhecido como *hop length*, foi igualmente definido como 512. Essas configurações, juntamente com a definição da frequência de amostragem igual a 16 kHz e duração de cada sinal de áudio igual a 4 segundos, resultam em uma janela de tempo (*window size*) de aproximadamente 32 milissegundos, sem sobreposição (*overlap*) entre janelas. Importante ressaltar que os parâmetros supracitados foram mantidos constantes para todas as técnicas de extração de características.

Após a extração de características, foi aplicada normalização nos dados extraídos. Foi utilizado o método de normalização *z-score*, o qual consiste em subtrair a média do conjunto de características (μ) por cada uma das características individualmente (x), e então dividir o resultado pelo desvio padrão correspondente ao conjunto de características (σ), conforme a Equação 3.1. Somou-se uma contante (ϵ) de valor 10^{-9} ao desvio padrão (σ) com o intuito de evitar divisões por zero. Isso resulta em uma matriz em que a média de todos os valores é igual a zero e o desvio padrão é igual a 1.

$$z = \frac{x - \mu}{\sigma + \epsilon} \tag{3.1}$$

3.2.3.1 Extração de Características para CNN Bidimensional

Um detalhamento dos parâmetros utilizados para aplicação de cada uma das técnicas de extração de características é apresentado a seguir:

- *Espectrograma em escala logarítmica*: a partir dos parâmetros de *hop length* e *nfft* definidos, foram obtidos espectrogramas com dimensões de 257x128. Posteriormente, foi aplicada uma escala logarítmica à amplitude do espectrograma. Para adequar esta representação ao formato desejado de 128x128 para entrada das redes neurais convolucionais, empregou-se redimensionamento por interpolação. Assim, a matriz resultante para cada áudio representa as 128 componentes da intensidade do espectrograma distribuídas ao longo das 128 janelas de tempo.

- *Espectrograma Mel*: foi definida a utilização de 128 bandas de filtros *mel*. Dessa forma, obteve-se uma matriz de tamanho 128x128 para cada espectrograma *mel*, onde estão representadas as 128 bandas de filtros *mel* para cada uma das 128 janelas de tempo analisadas nos sinais de áudio.
- *Espectrograma Mel em escala logarítmica*: também escolheu-se o número de 128 bandas de filtros *mel* para cada janela de tempo. Dessa forma, obteve-se uma matriz de tamanho 128x128 para cada espectrograma *mel*, onde estão representadas as 128 bandas de filtros *mel* para cada uma das 128 janelas de tempo. Adicionalmente, aplicou-se o logaritmo aos componentes das bandas de filtros mel correspondentes às 128 janelas de tempo, resultando em espectrogramas mel em escala logarítmica.

3.2.3.2 Extração de Características para CNN Unidimensional

Como uma das topologias selecionadas consiste em uma CNN unidimensional, foi necessário realizar redimensionamento nas características extraídas. Para isso, calculou-se a média da matriz transposta de formato 128x128 para cada uma das três técnicas de extração de características. Dessa forma, calculou-se a média das intensidades de cada característica extraída ao longo do tempo. Assim, obteve-se formato 128x1 para cada amostra de áudio.

3.2.4 *Data Augmentation*

A fim de otimizar o desempenho das topologias selecionadas e aumentar o número de amostras de áudio da base de dados, utilizou-se técnicas de *Data Augmentation*. Vale destacar que as técnicas de *Data Augmentation* foram aplicadas diretamente no áudio, anteriormente à aplicação das técnicas de extração de características. As técnicas e os fatores escolhidos para a aplicação de cada técnica tomaram como base a pesquisa realizada em (SALAMON; BELLO, 2017). Dessa forma, selecionou-se as técnicas de *Pitch Shifting* e *Time Stretching*. A Tabela 3.3 apresenta as técnicas de *data augmentation* e os respectivos fatores utilizados.

A técnica de *Pitch Shifting* é utilizada a fim de aumentar e diminuir o tom das amostras de áudio enquanto a duração do áudio é inalterada, sendo -1 e 1 representações de fatores que diminuem e aumentam um tom no áudio original, respectivamente. Os fatores selecionados para a aplicação dessa técnica foram -2 e 2. Já a técnica de *Time*

Stretching é utilizada a fim de acelerar e desacelerar uma amostra de áudio enquanto o tom do áudio é inalterado, sendo 0.5 e 1.5 fatores que representam a desaceleração e aceleração do áudio original, respectivamente. Os fatores selecionados para a aplicação da técnica de *Time Stretch* foram 0.81 e 1.23.

A aplicação das técnicas de *Data Augmentation* resultou no aumento em 4 vezes da base de dados original, visto que utilizou-se 2 técnicas e 2 fatores para cada técnica, e que a aplicação de um fator resulta em uma nova representação para cada amostra de áudio da base de dados. Assim, após a aplicação de *Data Augmentation*, obteve-se 34.928 novas amostras, resultando em um conjunto de dados de 43.660 amostras. Vale ressaltar que as amostras geradas a partir da técnica de *Data Augmentation* foram adicionadas à *fold* do áudio original correspondente, mantendo assim a proporção de amostras de áudio por *fold*. As topologias selecionadas foram treinadas com e sem *Data Augmentation*.

Tabela 3.3: Técnica de *data augmentation* e respectivos fatores aplicados.

Técnica	**Fatores**
Pitch Shifting	(-2, 2)
Time Stretching	(0.81, 1.23)

Fonte: O Autor.

3.2.5 Combinação de Fatores para Treinamento

Para uma análise mais robusta dos fatores em estudo, foram formuladas combinações considerando as cinco topologias pré-selecionadas, as três técnicas de extração de características, bem como a variação da base de dados com e sem *data augmentation*. Assim, o procedimento resultou em um total de 30 combinações distintas. Os fatores utilizados para combinação estão descritos na Tabela 3.4.

Tabela 3.4: Fatores utilizados para criação das 30 combinações distintas.

Topologias
SBCNN
STRIDED
D-MIX
2DCNN
1DCNN

Extração de Características
Espectrograma em Escala Logarítmica
Espectrograma Mel
Espectrograma Mel em Escala Logarítmica

Data Augmentation
Sim
Não

3.2.6 Definição de Hiperparâmetros

A definição dos hiperparâmetros para as topologias selecionadas foi realizada a fim de manter a avaliação do impacto dos fatores apresentados na Subseção 3.2.5. Assim, optou-se por manter o número de épocas, função de perda e otimizador fixos para todas as topologias. Utilizou-se número de épocas igual a 100, *batch size* igual a 128 e Entropia Cruzada Categórica (do inglês, *Categorical Cross Entropy*) como função de perda, uma vez que é comumente utilizada para problemas multiclasse. Por ser computacionalmente eficiente e requerer menos memória quando comparado a outros métodos de otimização estocásticos (KINGMA; BA, 2017), selecionou-se o otimizador *Adam*, com *learning rate* igual a 0.001.

3.2.7 Técnica de Validação

Para validação de cada combinação adotou-se a técnica de *10-fold cross-validation*, recomendada pelo autor da base de dados selecionada com o intuito de viabilizar a comparação de resultados obtidos com trabalhos relacionados que utilizaram a mesma base

de dados e técnica de validação (SALAMON; JACOBY; BELLO, 2014b). Cabe salientar que a base de dados $UrbanSound8K$ é originalmente dividida em 10 *folds*, e o autor enfatiza a importância de se manter a configuração original das *folds* para a correta aplicação do *10-fold cross-validation*.

Para cada uma das 10 iterações da técnica, selecionou-se uma *fold* distinta para compor o conjunto de teste, enquanto as nove *folds* restantes foram utilizadas para o conjunto de treinamento. Assim, ao final das 10 iterações, cada *fold* foi utilizada uma única vez como conjunto de teste. Vale ressaltar que, no processo de treinamento de cada modelo, optou-se por utilizar o mesmo procedimento apresentado em (SALAMON; BELLO, 2017), o qual consiste em utilizar uma das nove *folds* do conjunto de treinamento como conjunto de validação a fim de identificar a época de treinamento que obteve os melhores parâmetros do modelo ao realizar o treinamento com as 8 *folds* restantes. Dessa forma, para cada combinação descrita na subseção 3.2.5, aplicou-se o treinamento por validação cruzada. Após a conclusão das execuções, os resultados das 10 *folds* foram agregados através da aplicação da média aritmética.

Vale enfatizar que as amostras de áudio geradas a partir da técnica de *Data Augmentation* não foram incluídas nos conjuntos de validação. Isto é, em cada iteração da validação cruzada, a *fold* destinada à validação não continha os áudios gerados por aumento de dados. Dessa forma, assegurou-se que as topologias treinadas, com ou sem *Data Augmentation*, foram avaliadas com o mesmo conjunto de validação.

3.2.8 Métricas de Avaliação

Para a avaliação dos modelos treinados, foram utilizadas as métricas taxa de acerto percentual e os respectivos desvios padrão, número de parâmetros, *F1-Score*, *Recall* e matriz de confusão. Todas as métricas foram calculadas levando em consideração a média dos resultados obtidos nas 10 iterações de *10-fold* para cada uma das 30 combinações apresentadas na Seção 3.2.5. Ainda, considerou-se o número de parâmetros dos modelos a fim de estabelecer uma relação entre o custo computacional e a performance dos modelos. A matriz de confusão foi aplicada apenas para as duas topologias de melhor desempenho.

3.2.9 Teste de Hipóteses

Para investigar o impacto no desempenho a partir da utilização das diferentes topologias, técnicas de extração de características e *data augmentation*, foi utilizado o teste não paramétrico de *Mann-Whitney-Wilcoxon*, ou *U-Test* (MANN; WHITNEY, 1947). O teste de hipótese foi aplicado a partir da combinação 2 a 2 das 30 variações apresentadas na Seção 3.2.5. Para essa abordagem, utilizou-se como dados para análise a taxa de acerto obtida em cada iteração da técnica *10-fold cross validation* aplicada para cada uma das 30 combinações. Assim, foi possível verificar a ocorrência ou não de diferença estatisticamente significativa na aplicação de uma determinada técnica em detrimento de outra.

3.2.10 Poda Computacional

Para aplicação da poda computacional, utilizou-se a técnica de esparsidade constante com o auxílio da biblioteca *TensorFlow Model Optimization*. A biblioteca em questão permite duas abordagens para aplicação da esparsidade constante: a partir de um modelo não treinado e a partir de um modelo já treinado. Para a presente pesquisa, optou-se em aplicar a poda no modelo já treinado, processo conhecido como *fine-tune*. Para isso, definiu-se o número reduzido de 10 épocas, apenas para a aplicação da poda, e *learning rate* de 10^{-5}. Os seguintes níveis de esparsidade foram definidos pelo autor da monografia: 30%, 50%, 70%, 80% e 90%. A partir disso, selecionou-se o nível de esparsidade que resultou na maior taxa de acerto dos modelos.

3.2.11 Quantização

A técnica de Quantização de Intervalo Dinâmico (do inglês, *Dynamic Range Quantization*) foi empregada para a quantização do modelo após poda computacional. Esta técnica converte os pesos que possuem formato de ponto flutuante (*float*) de 32 *bits* para um formato de inteiro *int* de 8 *bits*, reduzindo significativamente a complexidade e o tamanho do modelo. A partir da aplicação dessa técnica, é possível reduzir o tamanho do modelo em até quatro vezes. (SIVAKUMAR et al., 2019).

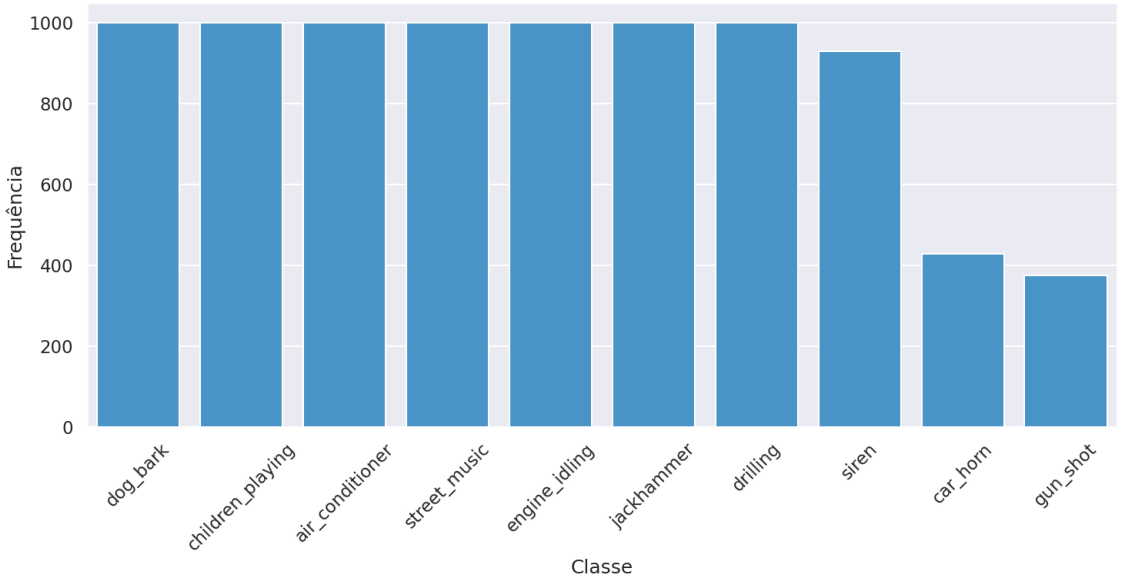
4 ANÁLISE DE RESULTADOS

No presente capítulo serão realizadas as análises dos resultados obtidos a partir da aplicação dos procedimentos discorridos no capítulo anterior.

4.1 Análise Estatística da Base de Dados *UrbanSound8K*

Conforme previamente apresentado na subseção 2.7.4, a base de dados *UrbanSound8K* é composta por 8732 amostras de áudio subdivididas em *10 folds*, e rotuladas em 10 classes de sons ambientais, os quais são nominados como: "dog bark"(1000), "children playing"(1000), "air conditioner"(1000), "street music"(1000), "engine idling"(1000), "jackhammer"(1000), "drilling"(1000), "siren"(929), "car horn"(429) e "gun shot"(374). A Figura 4.1 apresenta o gráfico de frequência de classes da base de dados em questão.

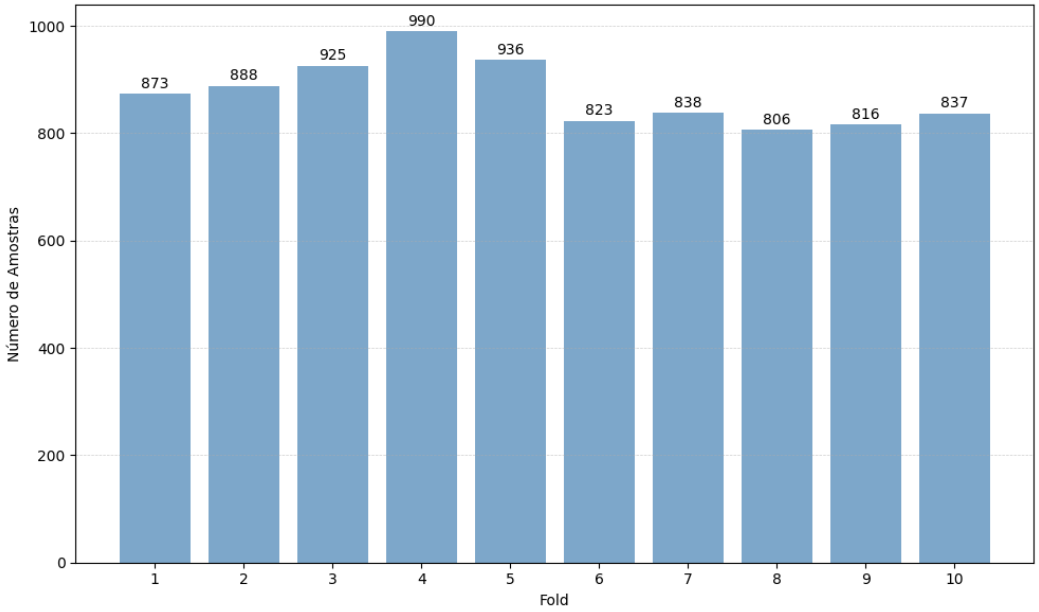
Figura 4.1: Gráfico de frequência de classes da base de dados *UrbanSound8K*.



Fonte: O Autor.

O gráfico da Figura 4.2 apresenta a distribuição de amostras de áudio em cada *fold* da base de dados *UrbanSound8K*, em que é possível verificar um desbalanceamento na divisão de amostras de áudio, sendo a *fold* 4 a que possui o maior número de amostras (990, enquanto que a *fold* 8 apresenta o menor número de amostras (806).

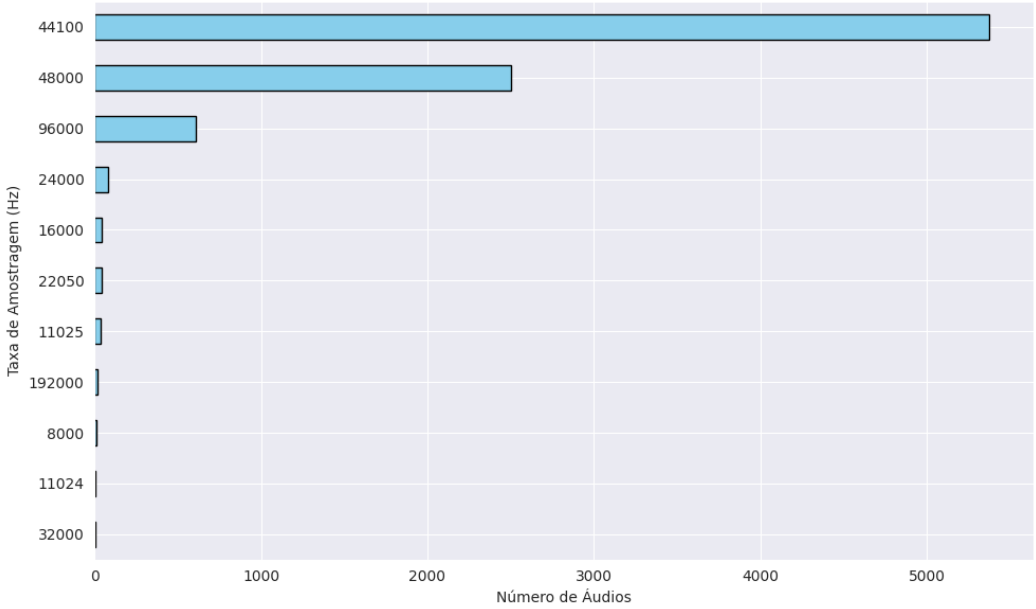
Figura 4.2: Gráfico de frequência de amostras de áudio por *fold* da base de dados *UrbanSound8K*.



Fonte: O Autor.

As amostras de áudio da base de dados *UrbanSound8K* possuem diferentes taxas de amostragem: $44,1kHz$ (5370), $48kHz$ (2502), $96kHz$ (610), $24kHz$ (82), $16kHz$ (45), $22,05kHz$ (44), $11,025kHz$ (39), $192kHz$ (17), $8kHz$ (12), $11,024kHz$ (7) e $32kHz$ (4). A Figura 4.3 apresenta o gráfico de frequência das taxas de amostragem por classe.

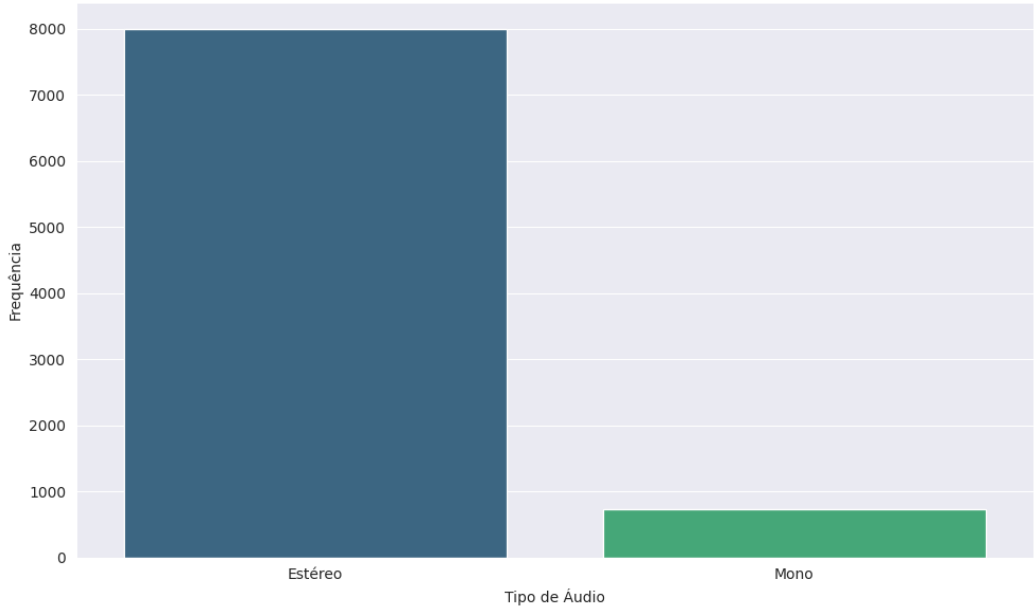
Figura 4.3: Gráfico de frequência das taxas de amostragem por classe da base de dados *UrbanSound8K*.



Fonte: O Autor.

A Figura 4.4 mostra a frequência de áudios estéreo e mono da base de dados selecionada. É possível notar que 8000 amostras possuem configuração estéreo, correspondendo a mais 90% das amostras de áudio da base de dados. Por outro lado, 732 amostras de áudio possuem configuração mono.

Figura 4.4: Gráfico de frequência de áudios estéreo e mono da base de dados *UrbanSound8K*.



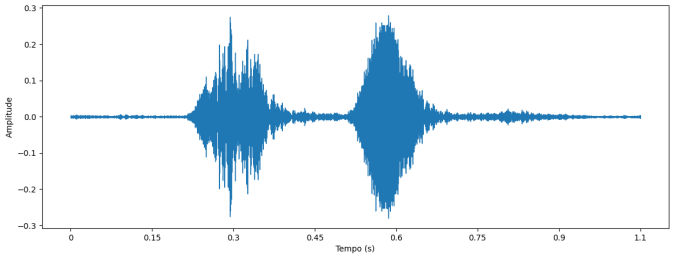
Fonte: O Autor.

4.2 Pré-processamento das Amostras de Áudio

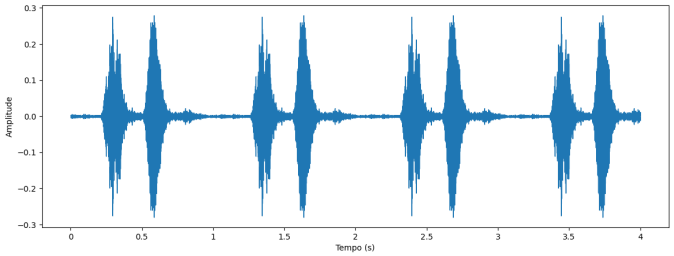
A Figura 4.5 apresenta uma amostra de áudio pertencente à classe "dog bark"(latido de cachorro) em sua forma original ($1,1$ segundos de duração), após aplicação da técnica de *Random Padding* para preenchimento do áudio até 4 segundos de duração e após as técnicas de *data augmentation* conhecidas como *time-stretching* e *pitch-shifting*.

Figura 4.5: Visualização do (a) áudio original, do (b) áudio após aplicação de *Random Padding*, do (c) áudio após aplicação da técnica de *data augmentation* conhecida como *pitch-shifting* com fator -2 e do (d) áudio após aplicação da técnica de *data augmentation* nominada *time-stretching* com fator $1,23$.

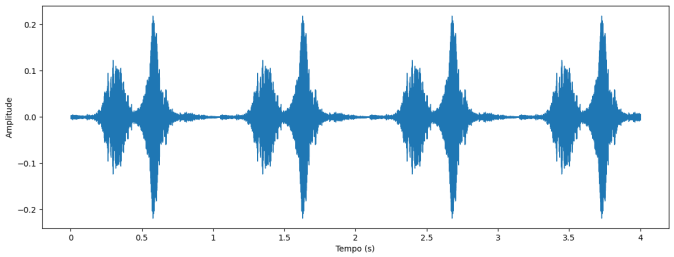
(a)



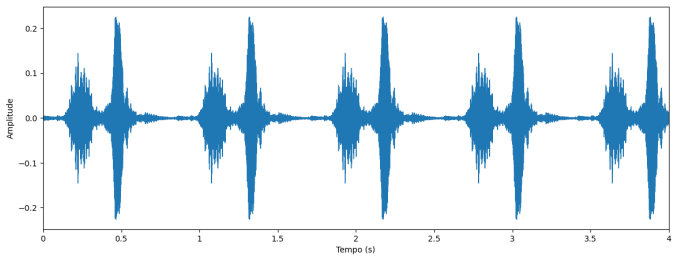
(b)



(c)



(d)

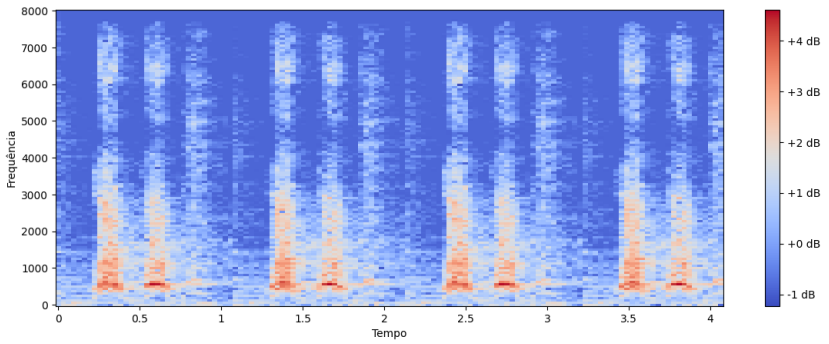


Fonte: O Autor.

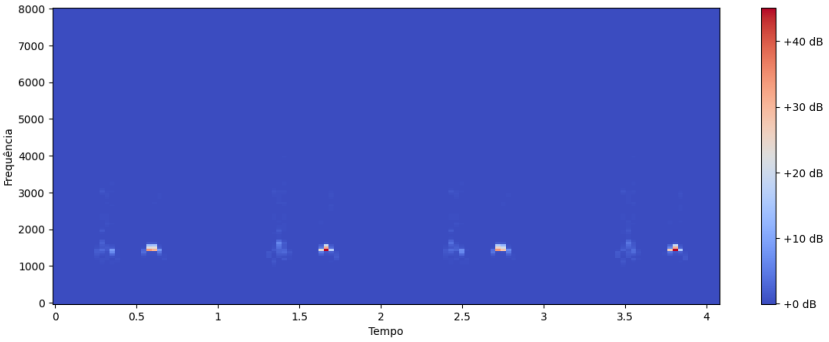
A Figura 4.6 apresenta os espectrogramas da mesma amostra de áudio apresentada na Figura 4.5, obtidos a partir da aplicação das técnicas de extração de características espectrograma em escala logarítmica, espectrograma mel e espectrograma mel em escala logarítmica, respectivamente.

Figura 4.6: Representação gráfica de um (a) espectrograma em escala logarítmica, de um (b) espectrograma mel e de um (c) espectrograma mel em escala logarítmica.

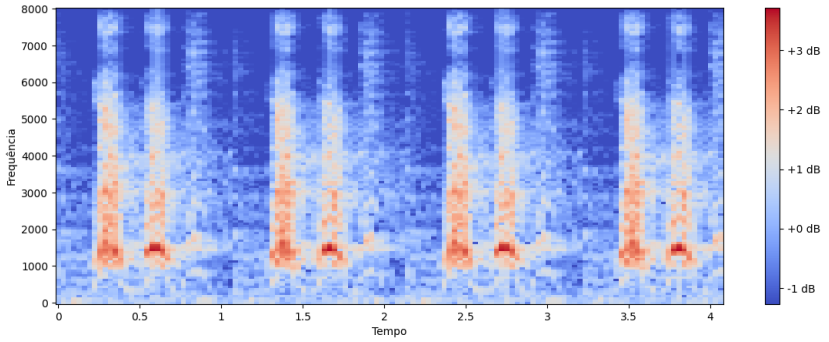
(a)



(b)



(c)



Fonte: O Autor.

4.3 Análise de Resultados dos Modelos de CNN

As Tabelas 4.1, 4.2, 4.3, 4.4 e 4.5 apresentam as taxas de acerto médias e os respectivos desvios padrão obtidos após treinamento e validação das 30 combinações. Nas tabelas em questão, utilizou-se a nomenclatura *"Spec"* para espectrograma em escala logarítmica, *"Mel"* para espectrograma mel e *"LogMel"* para espectrograma mel em escala logarítmica.

A partir da análise da Tabela 4.1 pode-se verificar que a combinação que obteve a maior taxa de acerto média foi com a utilização de espectrograma mel em escala logarítmica com *data augmentation* $(73,20\%)$, enquanto que a que obteve pior desempenho foi com a utilização de espectrograma mel sem data augmentation $(64,58\%)$.

Tabela 4.1: Resultados obtidos para a topologia SBCNN.

Extração de Características	*Data Augmentation*	Taxa de Acerto Média (%)	Desvio Padrão
Spec	Não	68,08	0,0627
	Sim	72,14	0,0552
Mel	Não	**64,58**	0,0530
	Sim	67,28	0,0452
LogMel	Não	72,51	0,0491
	Sim	**73,20**	0,0427

Fonte: O Autor.

A partir da análise da Tabela 4.2 verifica-se que, para a topologia *STRIDED*, a combinação que obteve a maior taxa de acerto média foi com a utilização de espectrograma mel em escala logarítmica com *data augmentation* $(72,95\%)$. Já o a pior combinação, assim como analisado para o modelo *SBCNN*, foi com a utilização de espectrograma mel sem *data augmentation* $(65,07\%)$.

Tabela 4.2: Resultados obtidos para a topologia STRIDED.

Extração de Características	*Data Augmentation*	Taxa de Acerto Média (%)	Desvio Padrão
Spec	Não	68,10	0,0524
	Sim	71,20	0,0439
Mel	Não	**65,07**	0,0455
	Sim	69,06	0,0607
LogMel	Não	69,94	0,0520
	Sim	**72,95**	0,0483

Fonte: O Autor.

A Tabela 4.3 apresenta os resultados obtidos a partir do treinamento da topologia *2DCNN*. É possível afirmar que a combinação que obteve a maior taxa de acerto média foi com a utilização de espectrograma mel em escala logarítmica com *data augmentation* $(68,98\%)$. Já a combinação com menor taxa de acerto média foi com a utilização de espectrograma mel sem *data augmentation* $(61,17\%)$.

Tabela 4.3: Resultados obtidos para a topologia 2DCNN.

Extração de Características	*Data Augmentation*	Taxa de Acerto Média (%)	Desvio Padrão
Spec	Não	61,98	0,0360
	Sim	68,94	0,0407
Mel	Não	**61,17**	0,0490
	Sim	68,51	0,0519
LogMel	Não	64,77	0,0314
	Sim	**68,98**	0,0328

Fonte: O Autor.

Os resultados obtidos a partir da utilização da topologia *D-MIX* estão apresentados na Tabela 4.4. Pode-se verificar que a combinação que obteve a maior taxa de acerto média foi com a utilização de espectrograma mel em escala logarítmica com *data augmentation* $(72,89\%)$. A combinação que obteve a menor taxa de acerto percentual média foi com a utilização de espectrograma mel sem *data augmentation* $(68,11\%)$.

Tabela 4.4: Resultados obtidos para a topologia D-MIX.

Extração de Características	*Data Augmentation*	Taxa de Acerto Média (%)	Desvio Padrão
Spec	Não	70,81	0,0504
	Sim	71,60	0,0520
Mel	Não	**68,11**	0,0587
	Sim	70,42	0,0570
LogMel	Não	72,60	0,0472
	Sim	**72,89**	0,0499

Fonte: O Autor.

A Tabela 4.5 apresenta os resultados obtidos a partir do treinamento da topologia *1DCNN*. Observa-se que a maior taxa de acerto média é decorrente da utilização de espectrograma linear com *data augmentation*, diferenciando-se do padrão dos demais modelos, os quais obtiveram maior taxa de acerto com a utilização da técnica espectrograma mel em escala logarítmica. Já a menor taxa de acerto média foi obtida com a utilização de espectrograma mel sem *data augmentation*, o que segue o padrão das demais topologias.

Tabela 4.5: Resultados obtidos para a topologia 1DCNN.

Extração de Características	*Data Augmentation*	Taxa de Acerto Média (%)	Desvio Padrão
Spec	Não	54,43	0,0717
	Sim	**57,83**	0,0569
Mel	Não	**52,04**	0,0450
	Sim	53,05	0,0427
LogMel	Não	53,50	0,0458
	Sim	55,69	0,0389

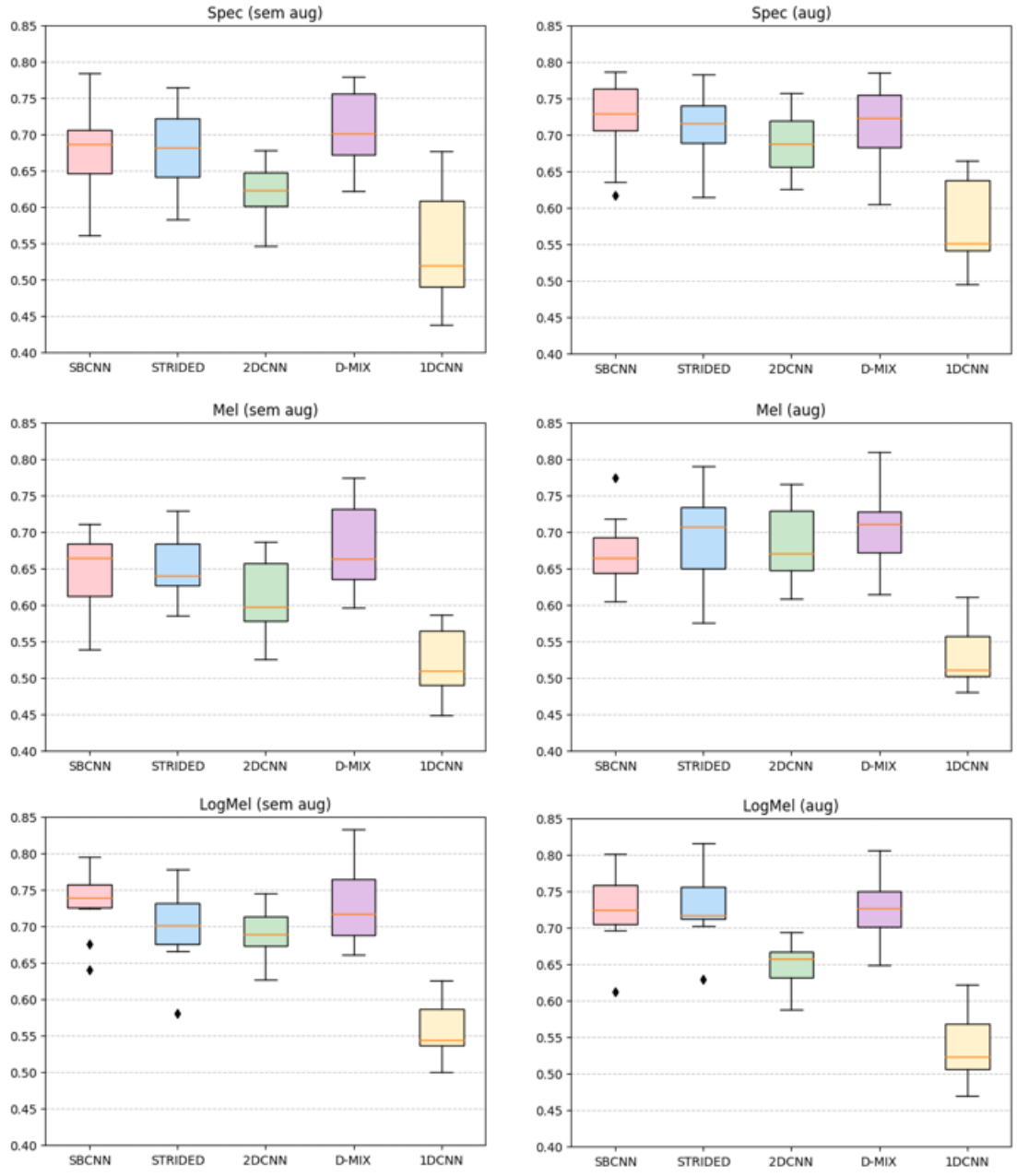
Fonte: O Autor.

A partir da análise das Tabelas 4.1, 4.2, 4.3, 4.4 e 4.5, pode-se verificar que, em linhas gerais, a combinação da técnica de extração de características espectrograma mel em escala logarítmica e a presença de *data augmentation* obtiveram as maiores taxas de acerto média para todos as topologias, com exceção da topologia *1DCNN*. Por outro lado, para todas as topologias, a combinação da técnica de extração espectrograma mel sem a utilização de *data augmentation* obteve a menor taxa de acerto média. Vale ressaltar que se observa um alto desvio padrão associado a todas as taxas de acerto médias. Esse comportamento era esperado, visto que o autor da base de dados justifica a recomendação

do uso da técnica *10-fold cross validation* pelo fato de que a separação das *folds* foi realizada de uma forma não trivial. Ou seja, os modelos tendem a obter pontuações mais altas quando treinados nas *folds* 1 à 9 e validados na *fold* 10 do que quando treinados nas *folds* 2 à 10 e validados na *fold* 1 (URBANSOUND8K, 2016).

A fim de apresentar os resultados obtidos de uma forma mais visual, criou-se gráficos *box plot*, os quais estão apresentados na Figura 4.7. Os resultados foram subdivididos em 6 gráficos, cada um correspondendo à uma técnica de extração de característica com a presença ou ausência de *data augmentation*. As topologias estão dispostas ao longo do eixo horizontal e a taxa de acerto média ao longo do eixo vertical.

Figura 4.7: Gráficos *Box Plot* das taxas de acerto referentes aos treinamentos das 30 combinações realizadas.



Fonte: O Autor.

A partir da análise da Figura 4.7, fica evidente que, em todas as combinações nas quais a topologia *1DCNN* é utilizada, o desempenho é consideravelmente inferior às combinações que empregam as outras topologias. Esse é um resultado esperado, uma vez que, dentre as 5 topologias utilizadas na presente pesquisa, esta é a que apresenta o menor número de parâmetros ($\approx 88,2k$). Além disso, é a única que possui uma estrutura unidimensional de CNN, com dados de entrada que representam a média de energia de cada banda de frequência ao longo de todo o áudio, sem levar em consideração a variação

temporal.

Para verificar se a suposição de que a topologia *1DCNN* impacta significativamente nas taxas de acerto médias quando comparada às demais topologias, aplicou-se o teste não paramétrico de *Mann-Whitney Wilcoxon*, respeitando as variáveis técnica de extração de características e presença ou ausência de *data augmentation*. Dessa forma, comparou-se apenas a influência da topologia nos resultados. Essa abordagem gerou 24 comparações. O nível de significância adotado para todas as análises foi de $0,05$.

Mediante a execução do teste de *Mann-Whitney Wilcoxon*, observou-se a rejeição da hipótese nula em todas as combinações analisadas, uma vez que o *p-valor* de cada combinação apresentou valor abaixo do nível de significância especificado. Os resultados estão apresentados na Tabela 4.6. Este resultado evidencia uma diferença estatisticamente significativa nas taxas médias de acerto para os modelos treinados com a topologia *1DCNN* em comparação com as demais topologias avaliadas. Dessa forma, optou-se por não considerar as combinações que possuem a topologia *1DCNN* nos demais testes realizados.

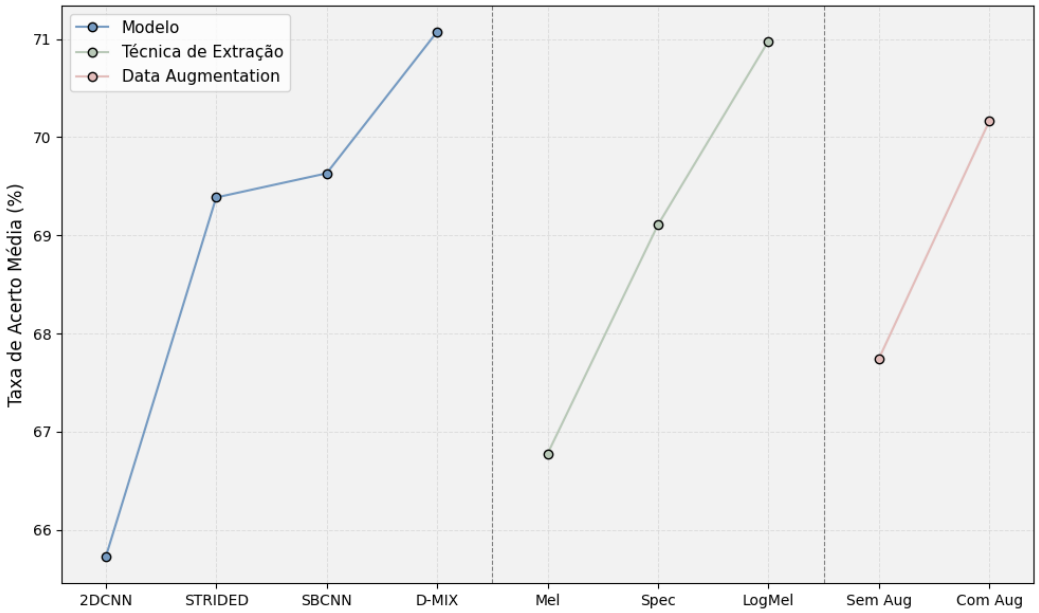
Tabela 4.6: Resultados do teste de Mann-Whitney-Wilcoxon para a combinação da topologia 1DCNN com os demais modelos, respeitando os demais fatores.

Extração/Aug	Modelo	Valor-P	H0 Rejeitada
Mel Aug	SBCNN	0,00025	Sim
	STRIDED	0,00033	Sim
	2DCNN	0,00025	Sim
	DMIX	0,00018	Sim
Mel No Aug	SBCNN	0,00058	Sim
	STRIDED	0,00025	Sim
	2DCNN	0,0022	Sim
	DMIX	0,00018	Sim
LogMel Aug	SBCNN	0,00025	Sim
	STRIDED	0,00018	Sim
	2DCNN	0,00044	Sim
	DMIX	0,00018	Sim
LogMel No Aug	SBCNN	0,00018	Sim
	STRIDED	0,00044	Sim
	2DCNN	0,00018	Sim
	DMIX	0,00018	Sim
Spec Aug	SBCNN	0,0013	Sim
	STRIDED	0,00058	Sim
	2DCNN	0,0010	Sim
	DMIX	0,00058	Sim
Spec No Aug	SBCNN	0,0017	Sim
	STRIDED	0,0013	Sim
	2DCNN	0,017	Sim
	DMIX	0,00044	Sim

Para dar sequência à analise dos fatores, utilizou-se um gráfico de efeitos principais para assertividade, o qual está apresentado na A Figura 4.8. Nota-se que o modelo que obteve as mais altas taxas de acerto foi o DMIX, enquanto que o que apresentou os resultados menos expressivos foi o modelo 2DCNN. Ainda, verificou-se uma diferença de, aproximadamente, 4 pontos percentuais entre a topologia 2DCNN e a topologia subsequente (STRIDED). Quanto às técnicas de extração de características, nota-se uma

superioridade da técnica espectrograma mel em escala logarítmica nas taxas de acerto médias. A presença de *Data Augementation* também apresentou resultados de taxa de acerto média superiores aos sem *Data Augmentation*.

Figura 4.8: Gráfico de efeitos principais para assertividade.



Fonte: O Autor.

A partir da análise do gráfico apresentado na Figura 4.8, decidiu-se aplicar o teste de *Mann-Whitney-Wilcoxon* a fim de verificar a existência ou não de diferença estatisticamente significativa entre o modelo 2DCNN quando comparado aos demais. Dessa forma, comparou-se apenas a influência do fator "topologia"nos resultados, respeitando as variáveis técnica de extração de características e presença ou ausência de *Data Augmentation*. A Tabela 4.7 apresenta os resultados do teste realizado, com os respectivos valores P. É possível notar que houve rejeição de hipótese nula para todas as combinações de espectrograma mel em escala logarítmica com *Data Augmentation* e para todas as combinações de espectrograma em escala logarítmica sem *Data Augmentation*. Dessa forma, decidiu-se desconsiderar as combinações que possuem a topologia 2DCNN e a extração espectrograma mel.

Tabela 4.7: Resultados do teste de Mann-Whitney-Wilcoxon para a combinação da topologia 2DCNN com os demais modelos, respeitando os demais fatores.

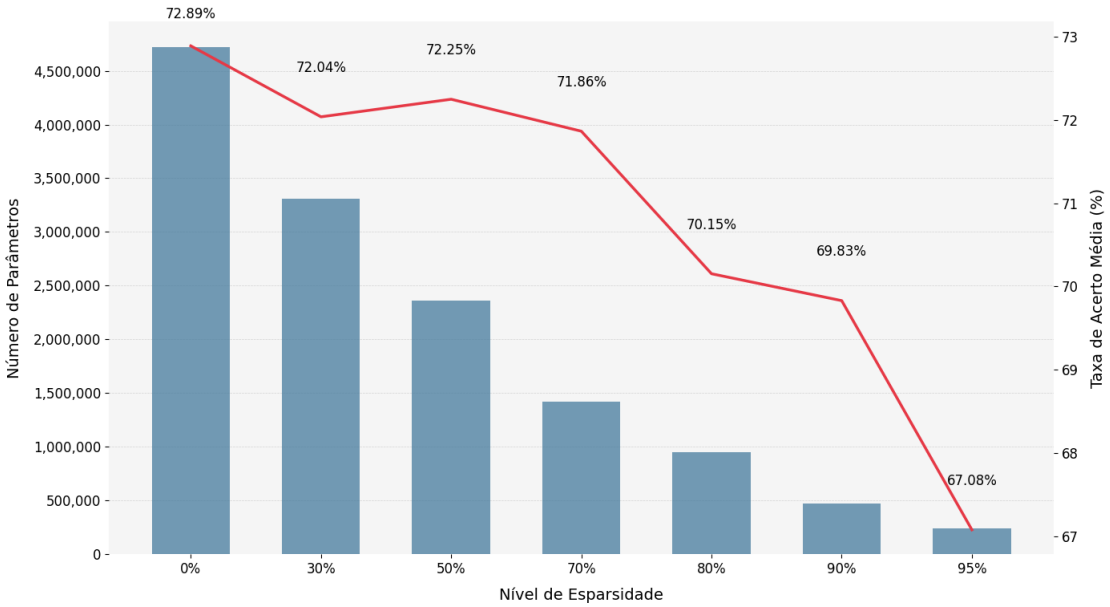
Extração/Aug	Modelo	Valor-P	H0 Rejeitada
Mel Aug	SBCNN	0,73373	Não
	STRIDED	0,85011	Não
	DMIX	0,47268	Não
Mel No Aug	SBCNN	0,14047	Não
	STRIDED	0,16197	Não
	DMIX	0,02575	Sim
LogMel Aug	SBCNN	0,00171	Sim
	STRIDED	0,00131	Sim
	DMIX	0,00361	Sim
LogMel No Aug	SBCNN	0,00211	Sim
	STRIDED	0,44952	Não
	DMIX	0,12122	Não
Spec Aug	SBCNN	0,21229	Não
	STRIDED	0,24132	Não
	DMIX	0,24132	Não
Spec No Aug	SBCNN	0,01726	Sim
	STRIDED	0,01726	Sim
	DMIX	0,00171	Sim

Ainda, aplicou-se o teste de *Mann-Whitney-Wilcoxon* para verificar a existência de diferença estatisticamente significativa com a utilização de "*Data Augmentation*". Após a aplicação do teste, notou-se que não houve rejeição de hipótese nula. O mesmo teste foi aplicado para para verificar a existência de diferença estatisticamente significativa entre o espectrograma em escala logarítmica e o espectrograma mel em escala logarítmica. Em nenhuma das combinações houve rejeição de hipótese nula. No entanto, a partir da análise do gráfico de box plots e do gráfico de efeitos principais da Figura 4.8, e para prosseguir e simplificar os testes desta monografia, decidiu-se seguir com o espectrograma mel em escala logarítmica e a presença de *Data Augmentation*. Essa decisão visou simplificar as etapas futuras de testes, bem como selecionar os fatores que obtiveram melhores resultados a partir das análises previamente realizadas, mesmo com as rejeições de hipótese nula explicitadas.

4.4 Aplicação de Poda Computacional nas Topologias Selecionadas

Com base na análise apresentada na Seção 4.3, os modelos *D-MIX*, *STRIDED* e *SBCNN* foram submetidos à poda computacional, a técnica de espectrograma mel em escala logarítmica e *data augmentation*. Dessa forma, aplicou-se poda computacional com níveis de esparsidade 30%, 50%, 70%, 80%, 90% e 95%. Os resultados obtidos para a topologia *D-MIX* estão apresentados na Tabela 4.9

Figura 4.9: Gráfico de Nível de Esparsidade *versus* Número de Parâmetros e Taxa de Acerto Média para a topologia *D-MIX*.

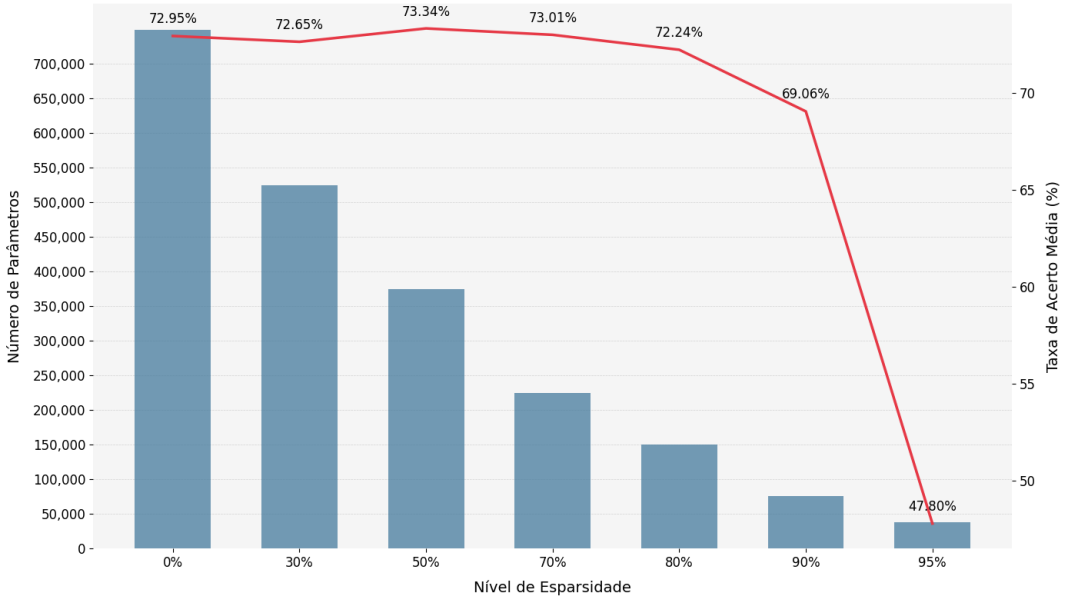


Fonte: O Autor.

Analisando o gráfico da Figura 4.9 verifica-se a formação de uma curva descendente na taxa de acerto média com o aumento do nível de esparsidade. Ainda, é possível observar uma queda mais acentuada da taxa de acerto a partir da esparsidade *80%*.

Os resultados após aplicação de poda computacional para a topologia *STRIDED* é mostrada na Figura 4.10. É possível inferir que a taxa de acerto média se manteve estável até a esparsidade de 80%, havendo uma queda abrupta entre os níveis de esparsidade 90% e 95%.

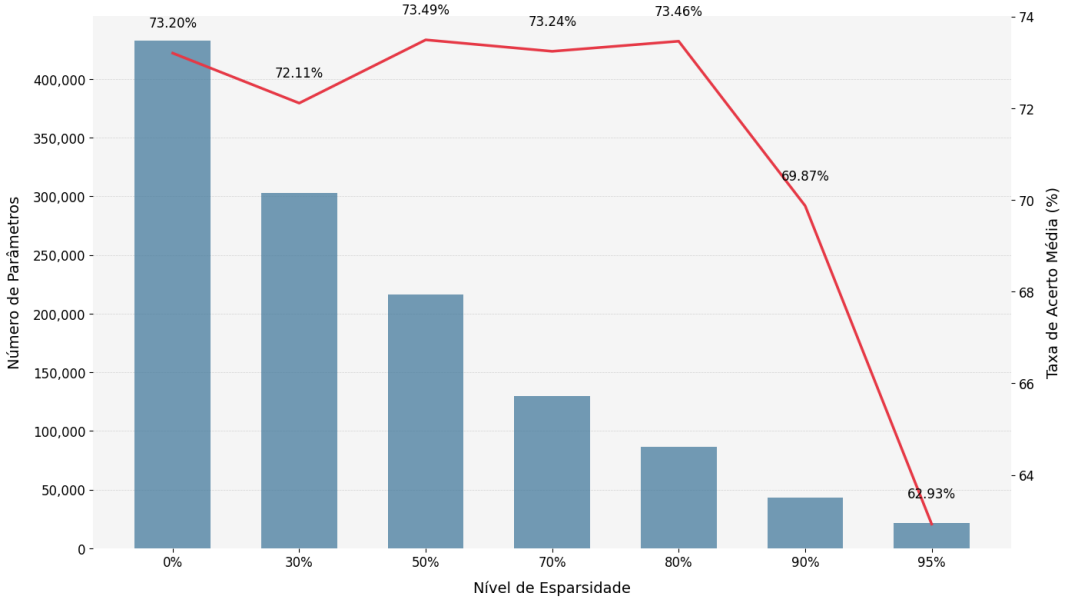
Figura 4.10: Gráfico de Nível de Esparsidade *versus* Número de Parâmetros e Taxa de Acerto Média para a topologia *STRIDED*.



Fonte: O Autor.

Os resultados obtidos para os níveis de esparsidade especificados para a topologia *SBCNN* são mostrados no gráfico da Figura 4.11. É possível observar uma constância nas taxas de acerto médias até o nível de esparsidade 80%. Para as esparsidades 90% e 95% nota-se uma tendência de queda mais acentuada.

Figura 4.11: Gráfico de Nível de Esparsidade *versus* Número de Parâmetros e Taxa de Acerto Média para a topologia *SBCNN*.



Fonte: O Autor.

Em linhas gerais, nota-se que as três topologias apresentaram queda mais significativa na taxa de acerto média a partir de 80% de esparsidade. Além disso, para a esparsidade de 80% as topologias SBCNN $(73,46\%)$ e STRIDED $(72,24\%)$ se sobressaíram à topologia D-MIX $(70,15\%)$. Levando em consideração o número de parâmetros isso fica ainda mais evidente, visto que a topologia D-MIX é a que possui maior número de parâmetros entre as analisadas. Dessa forma, decidiu-se seguir com as topologias SBCNN e STRIDED com nível de esparsidade de 80% para os testes subsequentes.

4.5 Quantização

Após a definição das duas topologias, realizou-se a quantização e a comparação das taxas de acerto média para os modelos original (sem poda), podadado com 80% de esparsidade e o modelo quantizado.

Os resultados obtidos do tamanho em Bytes e taxa de acerto média para a topologia STRIDED podem ser observados nas Tabelas 4.8 e 4.9, respectivamente.

Tabela 4.8: Tamanho resultante, em Bytes, dos Modelos Original, Podado e Quantizado para a Topologia STRIDED.

Modelo	**Tamanho (Bytes)**	**Desvio Padrão**
Original	2792467	32959
80% de Esparsidade	867906	9242
Quantizado	217498	11178

Fonte: O Autor.

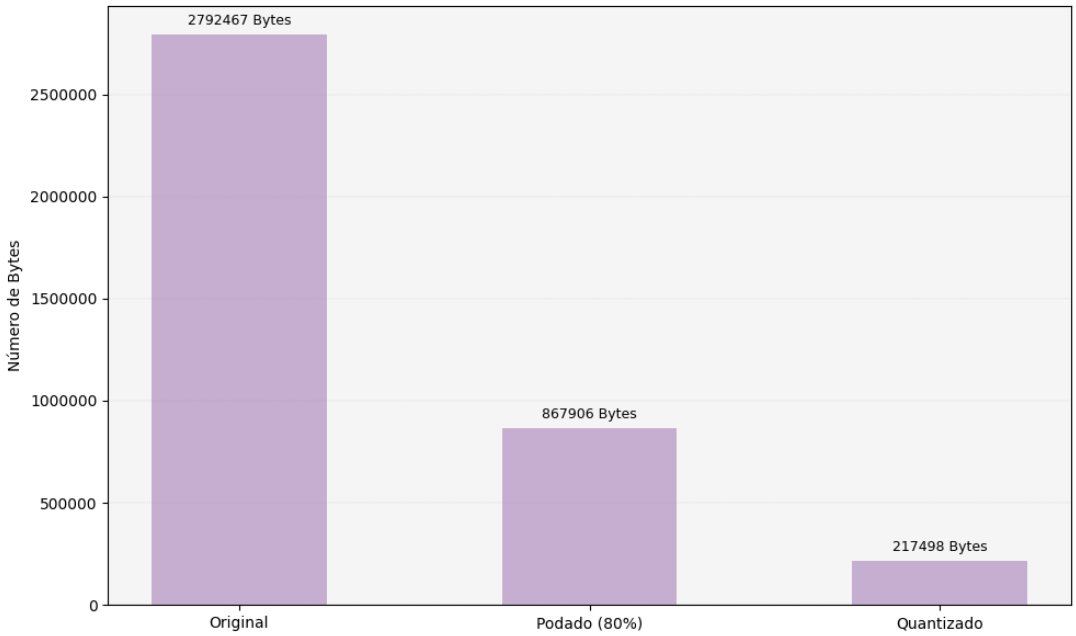
Tabela 4.9: Taxa de Acerto Média dos Modelos Original, Podado e Quantizado para a Topologia STRIDED.

Modelo	**Taxa de Acerto (%)**	**Desvio Padrão**
Original	72,95	0,0483
80% de Esparsidade	72,24	0,0484
Quantizado	72,19	0,0492

Fonte: O Autor.

O gráfico da Figura 4.12 apresenta de forma visual as relações de tamanho, em Bytes, dos modelos original, podado e quantizado para a topologia STRIDED. Ao comparar o modelo original ao modelo podado, observa-se uma diminuição de $3,22$ vezes no tamanho, enquanto que a taxa de acerto média variou de $72,95\%$ para $72,24\%$. Quando comparado o modelo podado ao modelo quantizado, observa-se uma redução no tamanho de $3,99$ vezes e uma redução na taxa de acerto média de $72,24\%$ para $72,19\%$. Ainda, ao comparar o modelo original ao modelo quantizado, nota-se uma diminuição no tamanho de $12,84$ vezes e uma diminuição na taxa de acerto de $72,95\%$ para $72,19\%$. Dessa forma, é possível inferir que há uma variação mínima na taxa de acerto média ao passo que há uma diminuição significativa no tamanho dos modelos, comprovando a efetividade da poda computacional e quantização para a topologia STRIDED.

Figura 4.12: Gráfico de Tamanho em Bytes *versus* Modelo para a Topologia *STRIDED*.



Fonte: O Autor.

Os resultados obtidos para a topologia SBCNN podem ser observados nas Tabelas 4.10 e 4.11. Conforme observado na análise da topologia STRIDED, também observa-se um desvio padrão considerável para o número de Bytes nos modelos original, podado e quantizado da topologia SBCNN. Ainda, é notável a diferença no tamamho dos modelos quando comparados aos modelos da topologia STRIDED, uma vez que a topologia STRIDED original possui aproximadamente $749,5$ mil parâmetros, enquanto que a topologia SBCNN original possui aproximadamente $432,4$ mil parâmetros.

Tabela 4.10: Tamanho resultante, em Bytes, dos Modelos Original, Podado e Quantizado, para a Topologia SBCNN.

Modelo	**Tamanho (Bytes)**	**Desvio Padrão**
Original	1612749	9171
80% de Esparsidade	513392	1879
Quantizado	125669	1588

Fonte: O Autor.

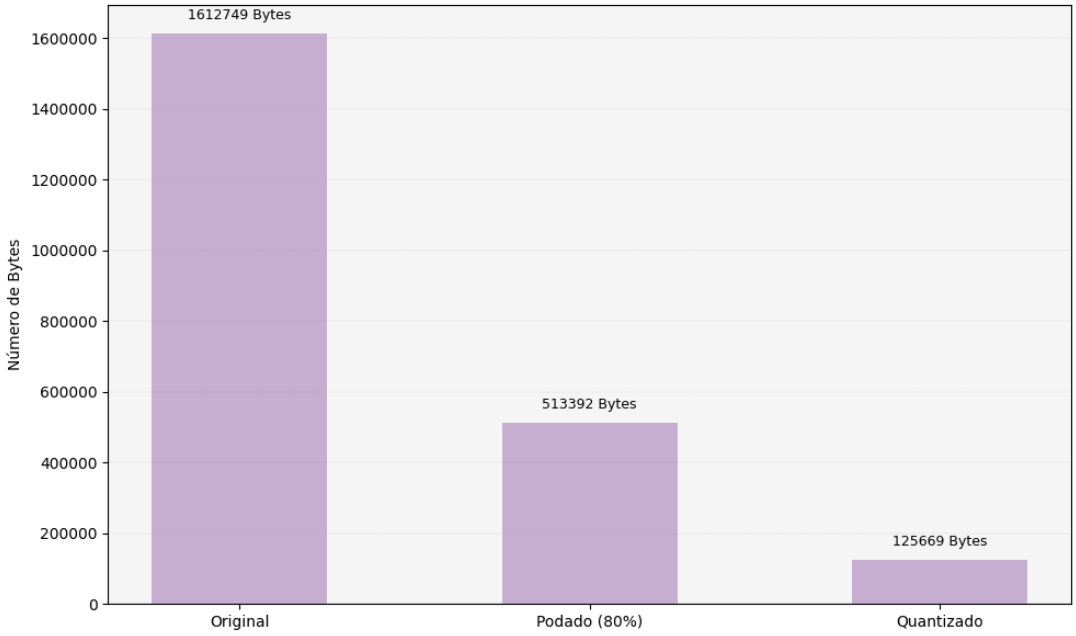
Tabela 4.11: Taxa de Acerto Média dos Modelos Original, Podado e Quantizado para a Topologia SBCNN.

Modelo	**Taxa de Acerto (%)**	**Desvio Padrão**
Original	73,20	0,0427
80% de Esparsidade	73,46	0,0473
Quantizado	73,48	0,0479

Fonte: O Autor.

A Figura 4.13 apresenta o gráfico de relação de taxa de acerto média e tamanho do modelo em Bytes para a topologia SBCNN. Ao comparar o modelo original ao modelo podado, observa-se uma diminuição de $3,14$ vezes no tamanho, enquanto que a taxa de acerto média variou de $73,20\%$ para $73,46\%$. Quando comparado o modelo podado ao modelo quantizado, observa-se uma redução no tamanho de $3,52$ vezes e uma variação na taxa de acerto média de $73,46\%$ para $73,48\%$. Ainda, ao comparar o modelo original ao modelo quantizado, nota-se uma diminuição no tamanho de $12,83$ vezes e uma variação na taxa de acerto de $73,20\%$ para $73,48\%$. Dessa forma, é possível inferir que, mesmo com a aplicação de técnicas de otimização de parâmetros, como poda computacional e quantização, o modelo manteve praticamente a mesma taxa de acerto média ao passo que obteve uma diminuição significativa no tamanho dos modelos para a topologia SBCNN.

Figura 4.13: Gráfico de Tamanho em Bytes *versus* Modelo para a Topologia SBCNN.



Fonte: O Autor.

4.6 Análise das Métricas Obtidas para as Topologias Quantizadas

A presente seção apresenta uma análise das métricas *Precision*, *Recall* e *F1-Score*, bem como as matrizes de confusão agregadas dos modelos quantizados das topologias STRIDED e SBCNN.

Tabela 4.12: Tabela de Métricas com *Precision*, *Recall* e *F1-Score* para o modelo quantizado da topologia STRIDED.

Classe	**Precision**	**Recall**	**F1-score**	**Suporte**
Air Conditioner	0.56	0.41	0.48	1000
Car Horn	0.85	0.88	0.86	429
Children Playing	0.72	0.84	0.78	1000
Dog Bark	0.80	0.84	0.82	1000
Drilling	0.65	0.70	0.68	1000
Engine Idling	0.59	0.63	0.61	1000
Gun Shot	0.95	0.95	0.95	374
Jackhammer	0.62	0.57	0.59	1000
Siren	0.88	0.80	0.84	929
Street Music	0.77	0.82	0.80	1000

Fonte: O Autor.

A partir da análise da Tabela 4.12, é possível inferir que o modelo quantizado da topologia STRIDED se destacou em identificar a classe "gun shot"(disparo), visto que foram obtidos valores de *precision* e *recall* consideravelmente altos, seguido pelas classes "car horn"(buzina de carro) e "siren"(sirene). Também é notável que o modelo demonstrou maior dificuldade nas previsões das classes "air conditioner"(ar condicionado) e "jackhammer"(britadeira).

Tabela 4.13: Tabela de Métricas com *Precison*, *Recall* e *F1-Score* para o modelo quantizado da topologia SBCNN.

Classe	**Precision**	**Recall**	**F1-Score**	**Suporte**
Air Conditioner	0.63	0.47	0.54	1000
Car Horn	0.89	0.87	0.88	429
Children Playing	0.75	0.82	0.79	1000
Dog Bark	0.76	0.87	0.81	1000
Drilling	0.74	0.70	0.72	1000
Engine Idling	0.61	0.61	0.61	1000
Gun Shot	0.86	0.95	0.90	374
Jackhammer	0.57	0.63	0.60	1000
Siren	0.91	0.84	0.87	929
Street Music	0.78	0.79	0.79	1000

Fonte: O Autor.

A Tabela 4.13 apresenta as métricas *Precision*, *Recall* e *F1-Score* para o modelo quantizado da topologia SBCNN. É possível notar que, assim como para o modelo STRIDED quantizado, o modelo SBCNN quantizado se destacou em identificar as classes "disparo", "buzina de carro"e "sirene". Da mesma forma, se mostrou com maiores dificuldades para prevêr as classes "ar condicionado"e "britadeira".

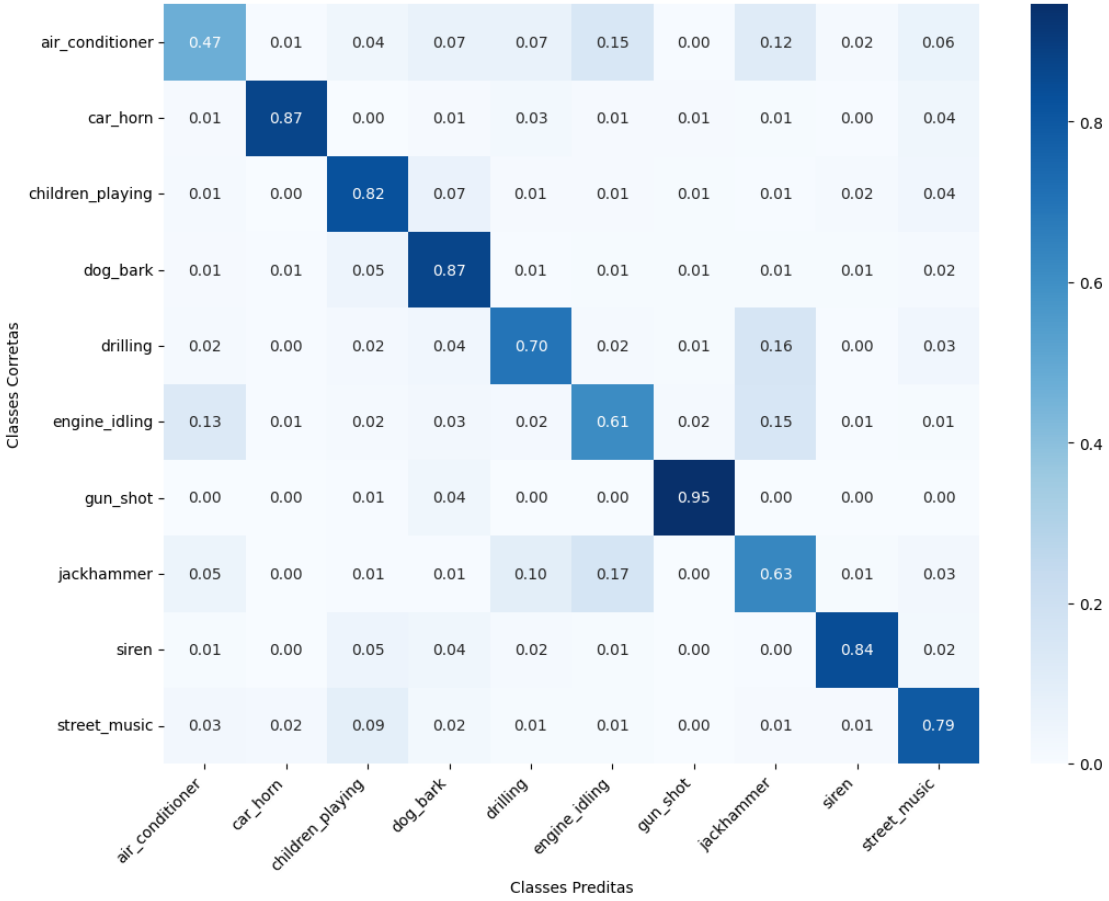
A partir da análise das Tabelas 4.12 e 4.13, é possível notar uma similaridade na identificação das classes para os dois modelos em questão. Isso também fica evidenciado ao visualizar as matrizes de confusão agregadas de cada topologia, apresentadas nas Figuras 4.14 e 4.15.

Figura 4.14: Matriz de Confusão Agregada para o Modelo Quantizado da Topologia STRIDED.

Fonte: O Autor.

Ao analisar a matriz de confusão agregada para o modelo quantizado da topologia STRIDED, apresentada na Figura 4.14, é possível notar que o modelo obteve maior dificuldade para classificar corretamente a classe "air conditioner"(ar condicionado), com 41% de taxa de acerto média. Ainda, verifica-se que essa classe foi confundida predominantemente com as classes "engine idling"(motor em marcha lenta) em 18% das predições e "drilling"(perfuração) em 14% das predições para a classe "ar condicionado". Já a classe em que o modelo melhor classificou foi "gun shot"(disparo), com 95% de taxa de acerto média. Vale ressaltar que a classe "disparo"obteve um bom desempenho mesmo sendo a classe com menor número de amostras na base de dados utilizada (374).

Figura 4.15: Matriz de Confusão Agregada para o Modelo Quantizado da Topologia SBCNN.



Fonte: O Autor.

A matriz de confusão agregada do modelo quantizado para a topologia SBCNN é apresentada na Figura 4.15. É possível inferir que, assim como para a topologia STRIDED, a classe com menor número de predições corretas foi "ar condicionado", com 47% de taxa de acerto média. Ainda, observa-se que essa classe foi predominantemente confundida com a classe "motor em marcha lenta", fato também observado para a topologia STRIDED. A classe com maior número de predições corretas para essa topologia foi "disparo"com 95% de taxa de acerto média, seguindo o mesmo padrão da topologia STRIDED.

A partir da análise das matrizes de confusão agregadas, é possível notar um padrão nas predições corretas de cada classe, bem como nas classes preditas de forma equivocada para as duas topologias, sendo a classe ar condicionado a com menor percentual de predições corretas e a classe "disparo"com o maior percentual de predições corretas.

4.7 Comparação dos Resultados Obtidos com Trabalhos Relacionados

A Tabela 4.14 apresenta uma comparação da taxa de acerto média entre trabalhos científicos relacionados e os resultados obtidos na presente pesquisa.

Tabela 4.14: Tabela comparativa da taxa de acerto média entre trabalhos científicos e os resultados obtidos.

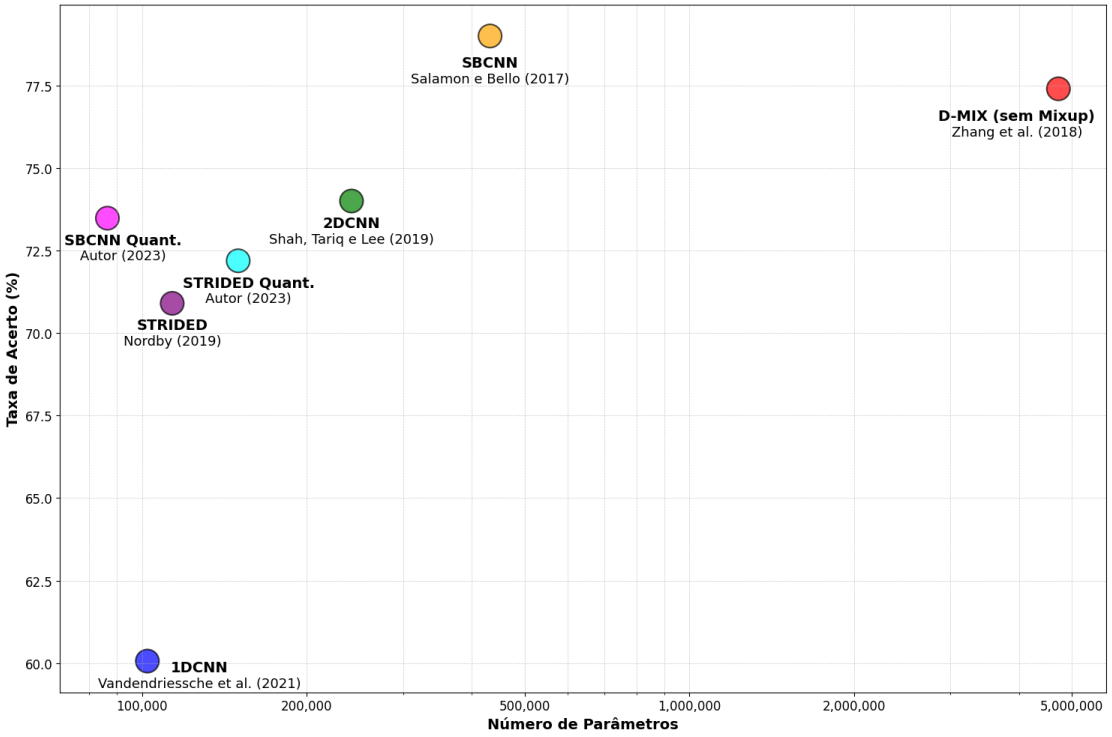
Modelo	**Características**	**Pesquisa**	**Taxa de Acerto (%)**
PiczakCNN	Sem Aug.	Piczak (2015)	72,70
SBCNN	Sem Aug.	Salamon & Bello (2017)	73,00
SBCNN	Com Aug.	Salamon & Bello (2017)	79,00
EnvNet-v2	Sem Aug.	Tokozume e Harada (2017)	78,30
D-CNN	Com Aug.	Zhang, Zou e Shi (2017)	81,90
D-MIX	Sem Mixup	Zhang et al. (2018)	77,40
D-MIX	Com Mixup + Aug.	Zhang et al. (2018)	83,70
STRIDED	Com Aug.	Nordby (2019)	70,90
2DCNN	Com Aug.	Shah, Tariq e Lee (2019)	74,00
1DCNN	Sem Aug.	Vandendriessche et al (2021)	60,06
STRIDED	Com Aug + Quant.	Autor (2023)	72,19
SBCNN	Com Aug + Quant.	Autor (2023)	73,48

Fonte: O Autor.

A partir da análise da Tabela 4.14 é possível notar que o modelo com quantização da topologia STRIDED obteve taxa de acerto maior do que a topologia original apresentada em Nordby (2019). Vale ressaltar que esse resultado pode estar associado ao tamanho dos dados de entrada, visto que na presente pesquisa utilizou-se espectrograma mel em escala logarítmica com formato de entrada 128x128, enquanto que em Nordby (2019) foi utilizada a mesma técnica de extração de características, porém com formato de entrada de 60x31.

Quanto ao modelo quantizado da topologia SBCNN, quando comparado ao modelo original proposto em Salamon e Bello (2017), observa-se um desempenho similar com a abordagem sem *data augmentation* e um desempenho abaixo em relação à abordagem com *data augmentation*.

Figura 4.16: Gráfico Comparativo da Taxa de Acerto *versus* Número de Parâmetros de Trabalhos Relacionados e dos Resultados Obtidos na Presente Pesquisa.



Fonte: O Autor.

O gráfico de taxa de acerto *versus* número de parâmetros da Figura 4.16 apresenta uma comparação entre as topologias de trabalhos científicos relacionados e a presente pesquisa. Vale ressaltar que foram consideradas no gráfico apenas as duas topologias que foram submetidas ao processo de quantização nesta pesquisa (SBCNN e STRIDED), uma vez que foram as topologias que melhor performaram com a combinação espectrograma mel em escala logarítmica e *Data Augmentation* para uma taxa de esparsidade de 80%.

Como observado no gráfico da Figura 4.16, o modelo STRIDED quantizado, obtido na presente pesquisa, possui número de parâmetros maior do que o modelo STRIDED original, apresentado em Nordby (2019). Conforme discutido anteriormente, isso se deve à diferença no formato dos dados de entrada da CNN. Entretanto, o resultado pode ser considerado satisfatório, uma vez que houve aumento na taxa de acerto média (de $70,90\%$ para $72,19\%$), apesar no aumento do número de parâmetros (de $102,3k$ para $149,9k$).

Ainda no gráfico da Figura 4.16, percebe-se diminuição significativa na taxa de acerto média e número de parâmetros do modelo SBCNN apresentado em Salamon e Bello (2017) em comparação como o modelo SBCNN quantizado, obtido na presente pesquisa. Enquanto que o modelo original possui aproximadamente $432,4k$ parâmetros e

obteve $77,40\%$ de taxa de acerto média, o modelo quantizado obtido na presente pesquisa possui aproximadamente $86,5k$ parâmetros e uma taxa de acerto média de $73,48\%$.

Outro fator a ser ressaltado é a comparação do número de parâmetros da topologia STRIDED apresentada em Nordby (2019) e o modelo SBCNN quantizado. É possível notar número de parâmetros inferior e taxa de acerto média superior quando comparado ao modelo STRIDED original, o que o torna um candidato à implementação em *hardwares* limitados. Vale ressaltar que para um modelo ser implementável em *hardwares* com restrições de recursos, outros fatores devem ser levados em conta além do tamanho do modelo, como memória RAM disponível no dispositivo embarcado e tempo de latência das inferências.

5 CONCLUSÕES

A presente pesquisa apresentou uma análise comparativa do desempenho de diferentes combinações de topologias de redes neurais convolucionais, técnicas de extração de características e utilização de *Data Augmentation*. Foi aplicada poda computacional e quantização nas topologias selecionadas a fim de verificar o impacto de tais técnicas de otimização na taxa de acerto média, no número de parâmetros e no tamanho dos modelos. Após treinamento dos modelos, verificou-se que a técnica de extração de características espectrograma mel em escala logarítmica obteve desempenho superior para todas as topologias de rede neural convolucional quando comparada às demais técnicas de extração. Também, observou-se que a utilização de *Data Augmentation* obteve, em linhas gerais, resultados superiores aos modelos treinados sem *Data Augmentation*.

Os melhores resultados foram obtidos com as topologias STRIDED e SBCNN. Após aplicação de poda computacional com nível de esparsidade de 80% e quantização, verificou-se que os modelos mantiveram as taxas de acerto média praticamente constantes, ao passo que obtiveram redução significativa no número de parâmetros e tamanho do modelo. A topologia STRIDED quantizada obteve taxa de acerto média de $72,19\%$ com desvio padrão de $4,92$, número de parâmetros de $149.892k$ e tamanho de 217.498 *bytes*. Já a topologia SBCNN quantizada obteve taxa de acerto média $73,48\%$ com desvio padrão de $4,79$, número de parâmetros de $86.746k$ e tamanho de 125.669 *bytes*.

Ao comparar a topologia SBCNN quantizada com o modelo original apresentado em Salamon e Bello (2017), observou-se diminuição na taxa de acerto média de $79,00\%$ para $73,46\%$ e diminuição no número de parâmetros de 432.378 para 86.746. Ao comparar a topologia STRIDED quantizada com o modelo original apresentado em Nordby (2019), é possível inferir que houve aumento na taxa de acerto média de $70,90\%$ para $72,19\%$ ao passo que também houve aumento no número de parâmetros de 102.290 para $149.892k$, visto que o formato dos dados de entrada utilizado na presente pesquisa foi consideravelmente maior do que o apresentado em Nordby (2019).

Por fim, conclui-se que a partir dos resultados obtidos, nota-se resultados satisfatórios de taxa de acerto e tamanho em *bytes* para aplicações em sistemas embarcados. Entretanto, vale ressaltar que ao realizar inferência em hardware a partir de modelos de aprendizado de máquina, outros fatores devem ser levados em consideração, tais como memória RAM, processamento e tempo de latência, o que não foi feito na presente pesquisa.

5.1 Trabalhos Futuros

Os resultados expostos nesta monografia sugerem oportunidades de aprimoramento em algumas abordagens realizadas. Podem ser exploradas diferentes técnicas de extração de características, estratégias de pré-processamento das amostras de áudio, técnicas de aprendizado de máquina, técnicas de *Data Augmentation* e formatos dos dados de entrada dos modelos.

Outra abordagem interessante é a implementação em *hardware* dos modelos quantizados para verificação do tempo de latência, consumo de memória RAM, tempo de processamento para extração das características dos áudios e potência consumida. Ainda, a inferência pode ser feita para áudios capturados em tempo real, por meio de módulo microfone. Outra possível abordagem é a comparação de desempenho em diferentes plataformas de *hardware*, tais como microprocessadores, microcontroladores e FPGAs.

REFERÊNCIAS

AGGARWAL, C. **Neural Networks and Deep Learning: A Textbook**. [S.l.]: Springer, 2018. ISBN 9783319944647.

AGHDAM, H.; HERAVI, E. **Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification**. [S.l.: s.n.], 2017. ISBN 978-3-319-57549-0.

ANDREADIS, A.; GIAMBENE, G.; ZAMBON, R. Convolutional neural networks for audio classification on ultra low power iot devices. In: **2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)**. [S.l.: s.n.], 2021. p. 1–6.

ANDREADIS, A.; GIAMBENE, G.; ZAMBON, R. Convolutional neural networks for audio classification on ultra low power iot devices. In: **2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)**. [S.l.: s.n.], 2021. p. 1–6.

ANUSUYA, M. A.; KATTI, S. K. Front end analysis of speech recognition: a review. **International Journal of Speech Technology**, v. 14, p. 99–145, 2011.

ARANDJELOVIć, R.; ZISSERMAN, A. **Look, Listen and Learn**. arXiv, 2017. Disponível em: <https://arxiv.org/abs/1705.08168>.

BERRY, N. **Discrete Cosine Transformations**. 2012. Disponível em: <http://datagenetics.com/blog/november32012/index.html>.

BOUREAU, Y.-L.; PONCE, J.; LECUN, Y. A theoretical analysis of feature pooling in visual recognition. In: **ICML**. [S.l.: s.n.], 2010.

CHOLLET, F. **Deep Learning with Python**. 1st. ed. USA: Manning Publications Co., 2017. ISBN 1617294438.

CHRISTENSEN, M. **Introduction to Audio Processing**. [S.l.]: Springer, 2019. ISBN 9783030117825.

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex fourier series. **Mathematics of Computation**, v. 19, p. 297–301, 1965.

CROCCO, M. et al. **Audio Surveillance: a Systematic Review**. 2014.

CROCCO, M. et al. Audio surveillance. **ACM Computing Surveys (CSUR)**, v. 48, p. 1 – 46, 2016.

DONG, X. et al. Environment sound event classification with a two-stream convolutional neural network. **IEEE Access**, v. 8, p. 125714–125721, 2020.

FAYEK, H. **Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between.** 2016. Disponível em: <https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html>.

GRIFFIN, D.; LIM, J. Signal estimation from modified short-time fourier transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 2, p. 236–243, 1984.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. 2. ed. [S.l.]: Bookman Editora, 2003. ISBN 9788577800865.

JARIWALA, H. et al. "noise pollution and human health: A review ". In: . [S.l.: s.n.], 2017.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2017.

KUMARI, S. et al. Edgel^3: Compressing l^3-net for mote scale urban noise monitoring. In: **2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)**. [S.l.: s.n.], 2019. p. 877–884.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: **Proceedings of 2010 IEEE International Symposium on Circuits and Systems**. [S.l.: s.n.], 2010. p. 253–256.

MANN, H. B.; WHITNEY, D. R. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 18, n. 1, p. 50 – 60, 1947. Disponível em: <https://doi.org/10.1214/aoms/1177730491>.

MOHAIMENUZZAMAN, M. et al. **Environmental Sound Classification on the Edge: A Pipeline for Deep Acoustic Networks on Extremely Resource-Constrained Devices**. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2103.03483>.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: **Sistemas Inteligentes: Fundamentos e Aplicações**. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168.

MOONS, B.; BANKMAN, D.; VERHELST, M. Embedded deep neural networks: Algorithms, architectures and circuits for always-on neural network processing. In: ____. [S.l.: s.n.], 2019. p. 1–31. ISBN 978-3-319-99222-8.

MUSHTAQ, Z.; SU, S.-F. Efficient classification of environmental sounds through multiple features aggregation and data enhancement techniques for spectrogram images. **Symmetry**, v. 12, n. 11, 2020. ISSN 2073-8994. Disponível em: <https://www.mdpi.com/2073-8994/12/11/1822>.

NORDBY, J. O. Environmental sound classification on microcontrollers using convolutional neural networks. In: . [s.n.], 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:202777894>.

PENG, L. et al. Ulsed: An ultra-lightweight sed model for iot devices. **Journal of Parallel and Distributed Computing**, v. 166, p. 104–110, 2022. ISSN 0743-7315. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0743731522000867>.

PICZAK, K. J. Environmental sound classification with convolutional neural networks. In: **2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)**. [S.l.: s.n.], 2015. p. 1–6.

PICZAK, K. J. ESC: Dataset for Environmental Sound Classification. In: **Proceedings of the 23rd Annual ACM Conference on Multimedia**. ACM Press, 2015. p. 1015–1018. ISBN 978-1-4503-3459-4. Disponível em: <http://dl.acm.org/citation.cfm?doid=2733373.2806390>.

PORTELO, J. et al. Non-speech audio event detection. In: **2009 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.: s.n.], 2009. p. 1973–1976.

RADHA, K.; SHRUTHI, G. Jpeg encoder using discrete cosine transform & inverse discrete cosine transform. **IOSR Journal of Electronics and Communication Engineering**, v. 5, p. 51–56, 2013.

RAO, K.; VUPPALA, A. **Speech Processing in Mobile Environments**. [S.l.]: Springer International Publishing, 2014. (SpringerBriefs in Speech Technology). ISBN 9783319031163.

SALAMON, J.; BELLO, J. P. Deep convolutional neural networks and data augmentation for environmental sound classification. **IEEE Signal Processing Letters**, v. 24, n. 3, p. 279–283, 2017.

SALAMON, J.; JACOBY, C.; BELLO, J. P. A dataset and taxonomy for urban sound research. In: **22nd ACM International Conference on Multimedia (ACM-MM'14)**. Orlando, FL, USA: [s.n.], 2014. p. 1041–1044.

SALAMON, J.; JACOBY, C.; BELLO, J. P. A dataset and taxonomy for urban sound research. In: **Proceedings of the 22nd ACM International Conference on Multimedia**. New York, NY, USA: Association for Computing Machinery, 2014. (MM '14), p. 1041–1044. ISBN 9781450330633. Disponível em: <https://doi.org/10.1145/2647868.2655045>.

SERIZEL, R. et al. Acoustic Features for Environmental Sound Analysis. In: **Computational Analysis of Sound Scenes and Events**. Springer International Publishing AG, 2017. p. 71–101. Disponível em: <https://hal.archives-ouvertes.fr/hal-01575619>.

SHAH, S. K.; TARIQ, Z.; LEE, Y. Iot based urban noise monitoring in deep learning using historical reports. In: **2019 IEEE International Conference on Big Data (Big Data)**. [S.l.: s.n.], 2019. p. 4179–4184.

SIVAKUMAR, S. et al. **TensorFlow Model Optimization Toolkit — Post-Training Integer Quantization**. 2019. Disponível em: <https://blog.tensorflow.org/2019/06/tensorflow-integer-quantization.html>. Acesso em: 01/08/2023.

SMITH, J. O. **Physical Audio Signal Processing**. [S.l.]: `http://-ccrma.stanford.edu/~jos/pasp/`, 2010. Online book.

STEVENS, S.; VOLKMANN, J.; NEWMAN, E. A scale for the measurement of a psychological magnitude: Loudness. **Psychological Review**, v. 43, n. 5, p. 405–416, 1937.

TABER, R. **Technology for a Quieter America**. National Academies Press, 2010. ISBN 9780309156325. Disponível em: <https://books.google.com.br/books?id=0StkAgAAQBAJ>.

URBANSOUND8K. 2016. <https://urbansounddataset.weebly.com/urbansound8k.html>. Acessado em: 01/09/2023.

VANDENDRIESSCHE, J. et al. Environmental sound recognition on embedded systems: From fpgas to tpus. **Electronics**, v. 10, n. 21, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/21/2622>.

VIRTANEN, T.; PLUMBLEY, M. D.; ELLIS, D. **Computational Analysis of Sound Scenes and Events**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2017. ISBN 3319634496.

ZHANG, Z. et al. **Deep Convolutional Neural Network with Mixup for Environmental Sound Classification**. 2018.

ZHU, M.; GUPTA, S. To prune, or not to prune: exploring the efficacy of pruning for model compression. **arXiv preprint arXiv:1710.01878**, 2017.