Detector de motores a combustão por meio de análise sonora através de uma rede neural.

Matheus Felipe Sozza[1]*; Maurício Eloy[2]

[1] Robert Bosch GmbH. Engenheiro Eletricista. Rodovia Anhanguera, km 98 – Vila Boa Vista; 13065-900 Campinas, São Paulo, Brasil
[2] USP - ESALQ. MSc. em Matemática e MBA em Data Science e Analytics. Avenida Pádua Dias, 235 – Agronomia; 13418-900 Piracicaba, São Paulo, Brasil
*autor correspondente: matheussozza@hotmail.com

Detector de motores a combustão por meio de análise sonora através de uma rede neural.

Resumo

O atual cenário tecnológico da computação nos provê com grande capacidade de coleta e armazenamento de informações, seja de maneira local ou na nuvem, ao mesmo tempo em que a evolução dos computadores incrementou de maneira exponencial a capacidade de processamento intensivo de todo tipo de dado no geral. Ao aliar essas características com técnicas modernas de aprendizado de máquina temos a possibilidade de extrair de tais dados informações que, a priori, são humanamente imperceptíveis, porém agregam valor e ajudam a direcionar a análise da informação ali contida, no caso do presente trabalho, como um classificador de amostras sonoras. Para tal, foi utilizada uma base de dados sonoros e previamente rotulados disponível na internet (UrbanSound8k), sob a qual diversos trabalhos científicos já foram realizados. No caso do presente trabalho foi proposta uma análise de segmentos sonoros dos quais extraíram-se os espectrogramas de frequências que serviram como entradas para um treinamento supervisionado através de uma rede neural convolucional, com o objetivo de classificar se o segmento sonoro em questão possui o som de um motor a combustão em funcionamento ou não, sendo, portanto, uma classificação binária. A classificação apresentou performance interessante e contribuiu de maneira evidente para abrir portas para futuros trabalhos de reconhecimento e classificação de áudio voltados à indústria automotiva, seja na melhoria das emissões, controle de tráfego ou mesmo para diagnóstico automatizado das partes mecânicas.

Palavras-chave: Redes Neurais, Motores à combustão, Diagnóstico Veicular, Classificação.

Introdução

Motores a combustão são sistemas muito presentes no cotidiano dado sua extensa aplicabilidade e o fato de ser uma tecnologia há muito conhecida e cujo fenômeno base, a combustão, é dominada com robustez pela indústria.

Ao mesmo tempo, é sabido que apesar da relevância que esses dispositivos possuem, existem alguns pontos contra notáveis, como a poluição atmosférica e consequente potencialização do efeito estufa e aquecimento global, além dos riscos à saúde das pessoas expostas direta e prolongadamente aos poluentes emitidos pelo escapamento (Nikischer, 2020).

Fazendo uso dos padrões sonoros do motor é possível extrair informações a respeito da sua performance de operação, o que permite julgar se o dispositivo está em funcionamento, operando corretamente, e até mesmo diagnosticar falhas em partes específicas, como citado em Wu. Z. et al., (2022). Atualmente o diagnóstico de falhas é subjetivo e altamente correlacionado à expertise do técnico mecânico responsável pela avaliação e manutenção do veículo, o que resulta em baixo nível de sucesso, havendo então uma oportunidade de melhoria desse processo por meio de algoritmos inovadores (Kemalkar e Bairagi, 2016).

Outra possível utilização de dados sonoros de motores é aquela voltada à geração de estatísticas de tráfego e planejamento de demanda, ou até mesmo no controle de semáforos

baseado em situações especiais, como a presença de veículos de emergência (ambulâncias), como destacado em Analytics Vidhya (2022). Dados extraídos num contexto de Big-Data têm sido cada vez mais utilizados no âmbito das cidades inteligentes (*Smart-Cities*), auxiliando no planejamento e gestão do tráfego das zonas rural e urbana (Zhao, Y. et al., 2018).

Mohammadi e Al-Fuqaha (2018) enunciam que, apesar das altas capacidades de coleta e armazenamento de dados dos sistemas computacionais atuais, muito pouco se aproveita da informação ali contida, com métodos tradicionais baseados apenas em análise temporal, os quais negligenciam a presença de padrões valorosos contidos nos dados armazenados e que não visíveis diretamente numa análise por amostras. Por outro lado, o uso de redes neurais profundas (DNN ou *Deep Neural Networks*) se mostra uma alternativa válida e promissora para a extração de informações perspicazes do ponto de vista analítico.

Outro ponto enunciado por Mohammadi e Al-Fuqaha (2018) é o fato de que, uma vez que os dados são coletados e armazenados, é improvável que venham a ser reutilizados no futuro, encorajando-se o processamento imediato.

O presente trabalho visa demonstrar como é possível classificar amostras de sons em categorias definidas por meio do processamento intensivo e supervisionado dos dados através redes neurais convolucionais [CNNs], mesmo quando os padrões e métricas a se extrair não são observáveis ou perceptíveis a priori para um processamento através de algoritmos tradicionais e explicáveis, os quais se baseiam em regras pré-estabelecidas e modelagem baseada puramente em premissas matemáticas. (Bhatia, 2018).

Material e Métodos

Dados

Para o trabalho em questão foi utilizada a base de dados ‘UrbanSound8k’. Essa coletânea consiste em 8732 recortes sonoros de até 4 segundos de duração os quais já estão previamente rotulados em 10 categorias distintas, sendo elas: Ar-Condicionado, Buzina de Carro, Crianças brincando, Cães latindo, Perfuração, Motor em marcha lenta, Tiro de arma de fogo, Britadeira, Sirene e Música Urbana.

Todos os recortes de áudio foram gerados através de amostras disponíveis livremente na internet, já sendo pré-distribuídas em 10 diferentes pastas de maneira a facilitar a divisão dos dados para treino e teste quando aplicados em algoritmos de Machine-Learning [ML] e/ou Deep-Learning [DL]. É importante manter a distribuição original das 10 pastas, evitando a mistura e redivisão das amostras, pois dessa maneira evita-se que subamostras originárias de um mesmo arquivo de áudio sejam utilizadas tanto para o treino como para o teste e

validação do modelo, consequentemente gerando resultados artificialmente altos, como recomendado em Salamon et. al. (2014) para garantir a validades dos resultados. Os dados estão balanceados dentro de cada uma das dez pastas, isto é, todas as dez categorias rotuladas (Ar-Condicionado, Buzina, Crianças etc.) estão presentes de maneira bem distribuída.

Os recortes de áudio foram fornecidos em formato '.wav' e com taxa de amostragem de 44100 Hertz.

Pré-processamento – Segmentação e balanceamento de classes

De maneira a normalizar e preparar os recortes sonoros para o posterior processamento no algoritmo de ML, é importante adotar algumas premissas que serão mantidas durante todo o processo no que se refere à segmentação e amostragem dos recortes sonoros originais.

A duração das amostras padrão foi arbitrada em 476 milissegundos e 952 milissegundos, a depender do experimento ou rodada de execução (discorrido adiante). As amostras provenientes da base de dados 'Urbansound8k' cuja duração é menor que a duração padrão foram descartadas.

Dado o escopo do trabalho, a classificação dos dados se dá de maneira binária, isso é, o intuito é classificar apenas se há um motor a combustão em funcionamento ou não no recorte sonoro em análise, não sendo necessária a classificação em dez diferentes rótulos como originalmente presente na base de dados "UrbanSound8k".

Para tal, as dez categorias originais da base de dados foram reduzidas em apenas duas. A primeira categoria foi chamada de Motores a combustão [MC], onde foi considerado o rótulo original "Motor em Marcha Lenta" apenas; A segunda categoria foi chamada "Não-Motor a Combustão" [NMC], onde todos os demais rótulos foram agrupados.

Por consequência, gerou-se um notável desbalanceamento dos dados, onde 80 a 90% das amostras estiveram contidas na categoria NMC, e apenas 10 a 20% das amostras estiveram contidas na categoria MC. No que tange os algoritmos de ML e DL, assume-se que os dados das classes consideradas estão distribuídos razoavelmente de maneira similar e balanceada, o que muitas vezes não é verdade num contexto de vida real. Tem-se então um fator dificultador ao aprendizado do algoritmo que tende a apresentar um elevado viés em direção ao grupo majoritário, quando o interesse é realizar previsões de qualidade relativas à classe minoritária (Krawczyk, 2016).

Para atingir o objetivo de balancear as classes de maneira equilibrada, direcionou-se o foco desta etapa de pré-processamento à classe minoritária MC e aplicou-se uma série de

técnicas de maneira a sobreamostrar e criar artificialmente dados dentro da mesma, técnicas popularmente conhecidas como “oversampling” [OS] e “data-augmentation” [DA].

Durante a segmentação em amostras padrão da classe MC, aplicou-se um fator de “overlapping” de maneira a aumentar artificialmente o número de amostras contidas na classe em questão, como referenciado em Chen (2019) e Song (2021). Neste processo, realiza-se não apenas a segmentação da amostra original, mas existe uma sobreposição entre uma determinada subamostra e as respectivas subamostras antecessora e sucessora, de maneira que parte do conteúdo é compartilhado entre elas. O processo é mais bem ilustrado na Figura 1.

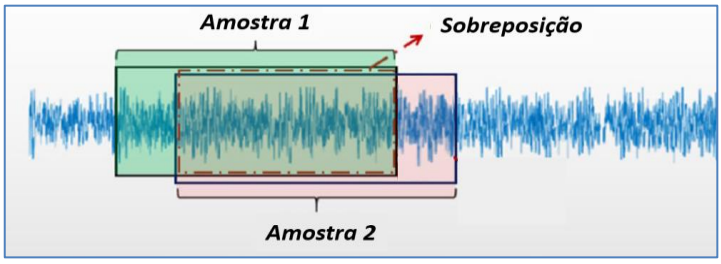


Figura 1. Demonstração gráfica do processo de sobreamostragem ou "oversampling" [OS] com sobreposição ou "overlapping"
Fonte: Song (2021) – traduzido e adaptado

Para uma dada amostra de áudio, é possível calcular o número de subamostras resultantes após a segmentação com sobreposição através da eq. (1):

$$N = \ 1 + \left[\frac{T - Ta}{Ta * (1 - S)}\right] \quad (1)$$

onde, T: é a duração da amostra original a ser segmentada; Ta: é a duração alvo das subamostras; S: é o fator de sobreposição que pode variar entre 0 (sem sobreposição) e 1 (sobreposição total); e N: é o número de subamostras resultantes, arredondado sempre para baixo.

Assim sendo, arbitrando-se o tamanho das subamostras e o fator de sobreposição, é possível prever o quanto o conjunto de dados aumentará com base na técnica de segmentação com sobreposição. O fator de sobreposição deve ser escolhido de maneira a balancear as classes de maneira igualitária.

Pré-processamento – Data Augmentation

Uma vez que os dados resultantes da segmentação com sobreposição contêm informações idênticas e repetidas (pois é uma técnica de sobreamostragem), aumenta-se então a tendência ao sobreajuste ou “overfitting” do nosso modelo devido ao aprendizado reforçado de um mesmo tipo de padrão repetidamente observado, conforme mencionado em Fernandez (2018). Essa situação é caracterizada por uma alta performance de classificação obtida nos dados de treino em oposição à uma baixa performance obtida durante a etapa de teste do modelo, o que indica que o algoritmo é pouco confiável ao ser aplicado em dados nunca vistos.

Para evitar tal problema, duas técnicas de “Data Augmentation” são aplicadas, com o intuito de criar subamostras sintéticas baseadas naquelas previamente existentes: Compressão ou Dilatação no Tempo e Adição de Ruído Branco Gaussiano.

A Compressão ou Dilatação no Tempo é realizada de maneira simples e direta, onde uma determinada amostra de áudio é comprimida ou dilatada, de maneira a ser executada de maneira mais rápida ou mais lenta, a exemplo do que é realizado em Zhou (2022). Dado que esse processo altera a duração da amostra, a aplicação dele é realizada em coordenação com a etapa de segmentação e sobreposição, de maneira a manter inalterada a duração das subamostras resultantes, estando de acordo com a duração padrão previamente arbitrada. Por exemplo, um áudio de 30 segundos dilatado com um fator de 2 teria uma nova duração de 60 segundos; do contrário, um áudio de 30 segundos comprimido com um fator de 0.5 teria uma nova duração de 15 segundos (Zhou, 2022).

Já a adição de ruído branco Gaussiano é realizada por meio da adição à amostra original de um sinal aleatório contendo intensidade uniforme em diferentes frequências, possuindo uma densidade espectral de potência constante. Uma vez que a mesma amostra dá origem a diferentes subamostras artificiais com diferentes níveis de ruído aleatório, espera-se que o modelo seja mais robusto ao se deparar com tal tipo de ruído em um futuro dado de entrada, o que melhora sua capacidade de generalização e performance como um todo, evitando um sobreajuste aos dados de treino. Como citado por Bishop (1995), em algumas circunstâncias é esperada uma melhora significativa na performance de generalização, pois treinar o modelo com ruído é equivalente à uma forma de regularização dos dados na qual um termo extra é adicionado à função de erro.

Processamento – Extração de "features" ou informações

Para treinar um modelo estatístico ou de ML, é necessário primeiramente extrair informações úteis do sinal de áudio. As "features" ou dados são basicamente descrições do sinal sonoro que podem ser fornecidas como entradas ao modelo a ser treinado de maneira a obter um sistema de processamento inteligente. Algumas aplicações do tipo incluem sistemas classificadores de áudio, reconhecedores de voz ou fala, rotuladores de música automáticos, removedores de ruído, dentre outros (Devopedia, 2021).

De acordo com Devopedia (2021), no que se refere ao escopo temporal dos dados de áudio, podemos dividi-los em:

- Instantâneos: Como sugerido, são os dados que nos fornecem informação instantânea sobre o sinal de áudio, isso é, consideram apenas recortes curtos do sinal, na faixa dos milissegundos (por exemplo, 10 milissegundos).
- Segmentados: São os dados calculados baseados em segmentos de áudio, na faixa de segundos.
- Globais: São os dados calculados com base no sinal sonoro inteiro, sem especificação de tempo ou duração.

Já relativo ao tipo de dado sonoro, ainda de acordo com Devopedia (2021), podemos classificá-los em:

- Domínio do Tempo: São aqueles os quais são extraídos diretamente das formas de onda no domínio do tempo do sinal de áudio em questão. Estão contidos nessa categoria métricas como "Zero-Crossing Rate", Envelope de Amplitude e Energia Média RMS.
- Domínio da Frequência: São as métricas focadas nos componentes de frequência do sinal sonoro, obtidas por meio de uma conversão do domínio do tempo para o domínio da frequência por meio da transformada rápida de Fourier [TRF]. Podemos exemplificar essa categoria através de métricas como Razão de Energia por Frequência, Centróide Espectral e Fluxo Espectral.
- Representação Tempo-Frequência: São as métricas que combinam tanto aspectos no domínio do tempo quanto aspectos no domínio da frequência. Esse tipo de representação é obtido por meio da Transformada de Fourier de Tempo Curto [TFTC]. Os espectrogramas simples e espectrogramas em frequência Mel são exemplos dessa categoria.

Considerando o caráter segmentado das amostras resultantes da etapa de pré-processamento, com tempos de duração padrão, a extração das informações dos recortes de áudio se deu por meio do cálculo de seus espectrogramas. A utilização de tal abordagem é particularmente útil porque transforma o sinal sonoro unidimensional (domínio do tempo) em uma imagem bidimensional (domínio do tempo e da frequência), o que auxilia no processo de separação do ruído de fundo e demais componentes indesejados do sinal sonoro principal que é alvo da identificação e classificação, como citado em Wu Z. et al (2022), tornando o desafio de processar esses dados algo similar ao processamento de imagens. A Figura 2 demonstra os resultados obtidos ao se converter um sinal no domínio do tempo para um espectrograma de tempo-frequência:

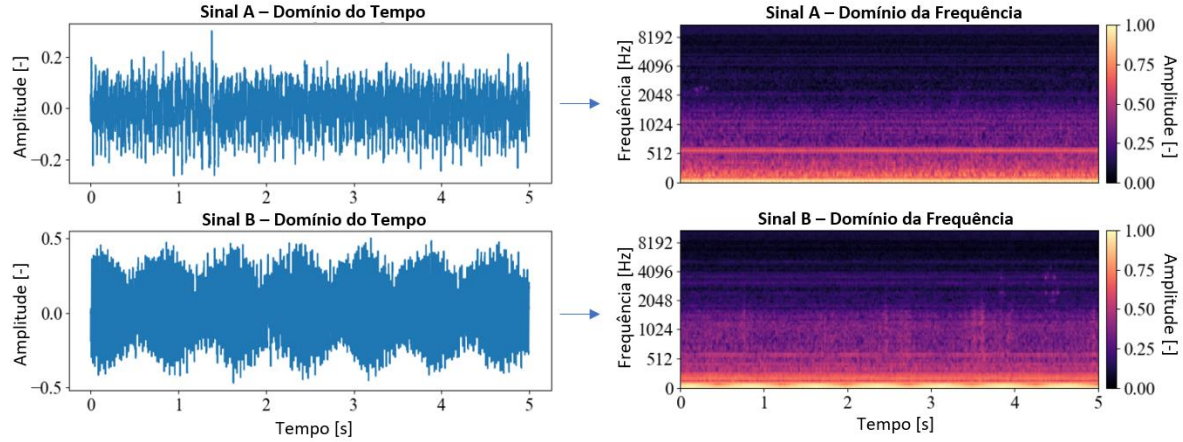


Figura 2. Representação de um mesmo sinal no domínio do tempo (esq.) e por meio de um espectrograma tempo-frequência (dir.)
Fonte: Ciric, et al. (2021) – traduzido e adaptado.

Os espectrogramas foram calculados de maneira a representarem o sinal sonoro por meio de seus coeficientes cepstrais na escala Mel, do inglês “Mel-Frequency Cepstral Coefficients” [MFCC].

Antes de contextualizar o que são os MFCC, é importante entender a definição do que é a escala Mel. Em resumo, tal escala é uma unidade subjetiva de medida da altura do som proposta por Steven et al. (1937), cujo propósito é refletir de maneira quantitativa como os humanos ouvem e percebem os tons musicais, isto é, a característica sonora que varia do grave ao agudo. Essa escala é conversível para uma escala padrão (com frequências medidas em Hertz) e o ponto de referência entra a escala Mel e a escala habitual é definido atribuindo um tom perceptual de 1.000 mels a um tom de 1.000Hz, 40 dB acima do limiar do ouvinte. A escala possui um espaçamento linear para frequências abaixo de 1000Hz, e um espaçamento logarítmico para frequências acima de 1000Hz (Hasan et al., 2004). A conversão de Hertz para Mels é dada pela eq. (2) e demonstrada na Figura 3:

$$mel = 2595 * log10(1 + \frac{f}{700}) \quad (2)$$

onde, f: é a frequência em Hertz; mel: é o resultado na escala Mel.

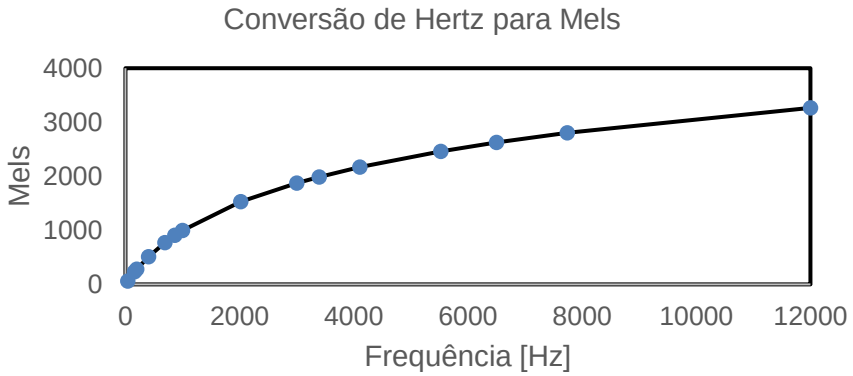


Figura 3. Demonstração gráfica da relação entre escala de frequências convencional e escala Mel
Fonte: Dados originais da pesquisa

Já os MFCC são coeficientes que coletivamente compõem um cepstro[1] de frequência Mel, ou do inglês “Mel-Frequency Cepstrum” [MFC], o qual é uma representação de tempo curto do espectro de potências de um sinal sonoro. Seu cálculo é baseado na transformada discreta de cosseno do espectro de potências em escala logarítmica de um sinal sonoro representado na escala de frequências Mel. O cálculo dos MFCC é dado pelas equações abaixo, como citado em Sahidullah (2011):

Seja $[s_{original}]_{Ntot}$ um sinal discreto no tempo contendo N_{tot} amostras (duração padrão), representando um sinal de áudio.

Considere agora a divisão de $S_{original}$ em T segmentos de tamanho N, tal que o produto T x N resulte no tamanho original N_{tot}. Tal divisão resulta no sinal segmentado $[s]_{TxN}$, conforme mostrado na Figura 4:

[1] O cepstro ou cepstrum é uma operação matemática que consiste em extrair a Transformada de Fourier do espectro do sinal na forma logarítmica. O nome “cepstrum” deriva de inverter a ordem das primeiras quatro sílabas de “spectrum”.

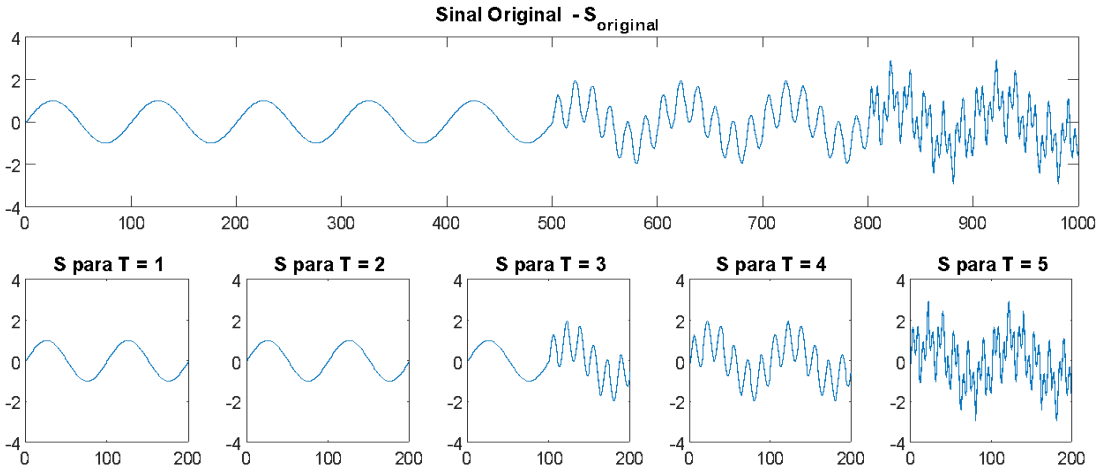


Figura 4. Demonstração da segmentação de um sinal de N_{tot}(1000) amostras em T(5) segmentos de N(200) amostras.
Fonte: Dados originais da pesquisa

Para o trabalho em questão, arbitrou-se que os segmentos conterão 512 amostras de áudios originalmente amostrados em 44100Hz, isso é, os segmentos que comporão o espectrograma e posteriormente o MFCC têm duração de 512/44100 = 11,6 milissegundos.

Etapa 1, Janelamento, descrita pela eq. (3):

$$[s_w]_{TxN} = [s]_{TxN} \; o \; [w]_{TxN} \qquad (3)$$

onde, T é o número de segmentos de tamanho N no qual o sinal original foi segmentado; S é a matriz de tamanho T x N contendo o sinal original após segmentação; W é a matriz de tamanho T x N contendo a mesma função de janelamento em todas suas linhas; O operador ‘o’ denota a multiplicação termo-a-termo entre as matrizes; e S_w é o sinal resultante após o janelamento, como exemplificado na Figura 5:

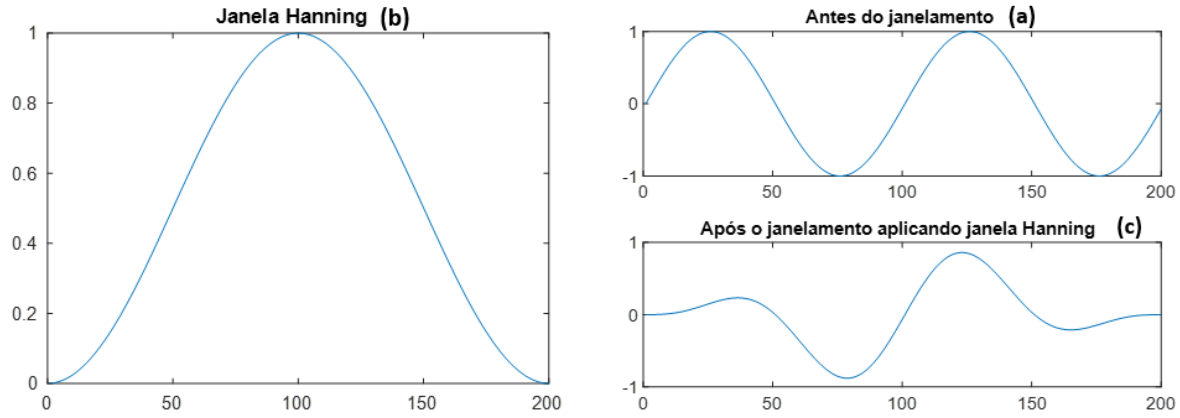


Figura 5. Exemplo de janelamento do sinal original (a) aplicando-se uma janela Hanning (b) e resultando no sinal pós-janelamento (c)
Fonte: Dados originais da pesquisa

Etapa 2, preenchimento com zeros ou “Zero-Padding”, descrita pela eq. (4):

$$[s_{zp}]_{TxM} = [s_w]_{TxN}\ [I\ \ O]_{NxM} \quad (4)$$

onde, T é o número de segmentos de tamanho N no qual o sinal original foi segmentado; M é um escalar (recomendável potência de 2) maior do que N que define o tamanho final do preenchimento com zeros; S_w é a matriz resultante do janelamento; I é a matriz identidade de tamanho N x N; O é a matriz nula de tamanho N x (M-N); e S_{zp} é o sinal resultante após o preenchimento com zeros; Essa etapa é opcional, e recomenda-se que o sinal resultante seja múltiplo de 2 devido à otimização de alguns algoritmos para tal caso. O resultado é mostrado na Figura 6:

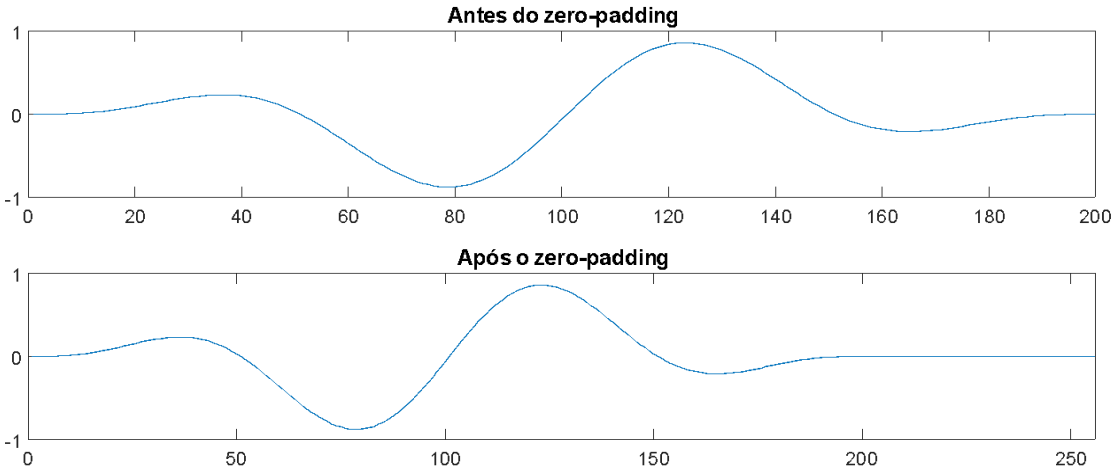


Figura 6. Sinal antes (acima) e após (abaixo) o processo de “Zero-Padding”, sendo estendido de 200 para 256 amostras
Fonte: Dados originais da pesquisa

Etapa 3, cálculos dos coeficientes discretos de Fourier, descritos pela eq. (5):

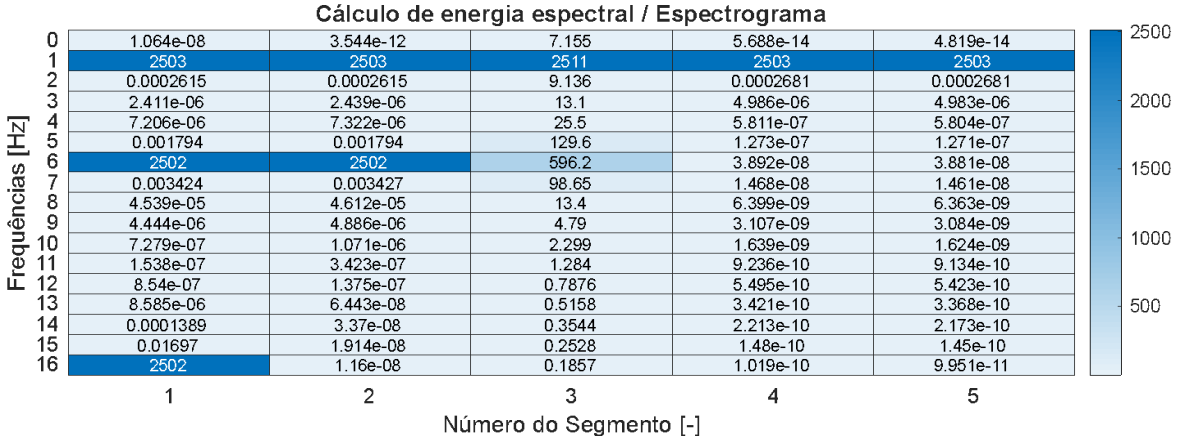
$$[\Omega]_{TxM/2} = [s_{zp}]_{TxM} \ [W]_{MxM/2} \quad (5)$$

onde, T é o número de segmentos do áudio original; M é um escalar maior do que N que define o tamanho final do preenchimento com zeros; S_{zp} é o sinal resultante do preenchimento com zeros; W é a matriz de fatores de rotação, do inglês “twiddle factor matrix”; e Ω é a matriz contendo os coeficientes discretos de Fourier resultantes.

Etapa 4, cálculo do espectro de potências, descrito pela eq. (6):

$$[\Theta]_{TxM/2} = [\Omega]_{Tx\frac{M}{2}} \ o \ [\Omega^*]_{Tx\frac{M}{2}} \quad (6)$$

onde, T é o número de segmentos do áudio original; M é um escalar maior do que N que define o tamanho final do preenchimento com zeros; Ω é a matriz contendo os coeficientes discretos de Fourier; Ω^* é o conjugado de Ω; e Θ é o espectro de potências resultante, conforme a Figura 7:



	1	2	3	4	5
0	1.064e-08	3.544e-12	7.155	5.688e-14	4.819e-14
1	2503	2503	2511	2503	2503
2	0.0002615	0.0002615	9.136	0.0002681	0.0002681
3	2.411e-06	2.439e-06	13.1	4.986e-06	4.983e-06
4	7.206e-06	7.322e-06	25.5	5.811e-07	5.804e-07
5	0.001794	0.001794	129.6	1.273e-07	1.271e-07
6	2502	2502	596.2	3.892e-08	3.881e-08
7	0.003424	0.003427	98.65	1.468e-08	1.461e-08
8	4.539e-05	4.612e-05	13.4	6.399e-09	6.363e-09
9	4.444e-06	4.886e-06	4.79	3.107e-09	3.084e-09
10	7.279e-07	1.071e-06	2.299	1.639e-09	1.624e-09
11	1.538e-07	3.423e-07	1.284	9.236e-10	9.134e-10
12	8.54e-07	1.375e-07	0.7876	5.495e-10	5.423e-10
13	8.585e-06	6.443e-08	0.5158	3.421e-10	3.368e-10
14	0.0001389	3.37e-08	0.3544	2.213e-10	2.173e-10
15	0.01697	1.914e-08	0.2528	1.48e-10	1.45e-10
16	2502	1.16e-08	0.1857	1.019e-10	9.951e-11

Figura 7. Espectrograma de potências resultante para o sinal do exemplo, segmentado em T(5) segmentos
Fonte: Dados originais da pesquisa

Etapa 5, filtragem e computação do logaritmo da energia, descrito pela eq. (7):

$$[\Psi]_{Txp} = \left[[\Theta]_{Tx\frac{M}{2}} * \ [\Lambda]_{\frac{M}{2}xp} \right] \quad (7)$$

onde, T é o número de segmentos do áudio original; M é um escalar maior do que N que define o tamanho final do preenchimento com zeros; Θ é o espectro de potências resultante; Λ é a matriz contendo um banco de p-filtros espaçados em escala Mel; e Ψ é o espectro resultante, filtrado e logarítmico, das potências do sinal original. Para o trabalho aqui apresentado fixamos a utilização de 25 filtros (p=25).

Etapa 6, cálculo da transformada de cosseno e obtenção dos MFCC, descrito pela eq. (8):

$$[x]_{Txp} = [\Psi]_{Txp} * [D]_{pxp} \quad (8)$$

onde, T é o número de segmentos do áudio original; p é o número de filtros do banco de filtros; D é a matriz p-dimensional da transformada de cosseno; e x é a matriz resultante contendo os MFCC, que se apresentam de forma similar ao espectrograma (bidimensional).

A escolha dos MFCC como informação principal para o treinamento e predição via algoritmo se dá pelo fato de ser uma grandeza utilizada com frequência em problemas que envolvem o reconhecimento, classificação e até mesmo predição com base em sinais de áudio, indo desde propostas como a identificação de pessoas através da voz desenvolvida por Hasan et al. (2004) até mesmo em aplicações focadas em sistemas mecânicos como a identificação e diagnóstico de defeitos em motores automotivos estudada por Kemalkar e Bairagi (2016).

Processamento – Rede Neural Convolucional

A utilização de redes neurais convolucionais, do inglês “Convolutional Neural Network” [CNN] é reconhecidamente um método efetivo para abordagens de processamento e classificação de sinais sonoros ambientais (J. Salamon, J. P. Bello, 2016), à exemplo do que é fornecido pelo banco de sons UrbanSound8k utilizado no trabalho aqui apresentado.

Para processar as amostras de áudio, foi utilizada uma topologia de CNN similar àquela utilizada por Piczak (2015) em um trabalho similar, apresentando resultados de boa performance.

Em resumo, a topologia utilizada inicia-se com duas camadas idênticas compostas por uma operação de convolução e uma operação de “pooling”. A parte convolucional contém um kernel de dimensão 3x3 e 80 filtros ou “feature maps”, enquanto o “pooling” é feito por meio de um kernel de dimensão 2x2 que captura o máximo valor, técnica mais conhecida pelo seu termo inglês “MaxPooling2D”. Com a aplicação do “pooling” espera-se uma melhora da

resposta do algoritmo a translações e pequenas variações nos dados de entrada (Piczak, 2015).

Após cada uma das camadas de convolução e “pooling”, tem-se também um “dropout” de 20%, resultando na remoção de parte dos valores calculados pelas unidades da CNN, com o objetivo de melhorar a capacidade de generalização da rede (Piczak, 2015).

Terminada a parte convolucional e de manipulação dos dados ainda em forma bidimensional, a matriz resultante passa novamente por uma camada de “pooling”, porém agora com o intuito de planificar os dados ali contidos, isso é, transformar a matriz bidimensional em um arranjo linear.

Por fim, o arranjo linear obtido na etapa anterior será fornecido como entrada a uma rede neural composta por duas camadas planas ou “flat layers” totalmente interconectados. A primeira camada possui 512 perceptrons, enquanto a segunda camada possui apenas 2 perceptrons, responsáveis pela classificação binária nas categorias MC e NMC conforme anteriormente definido.

A Figura 8 ilustra graficamente toda a descrição textual da CNN acima especificada:

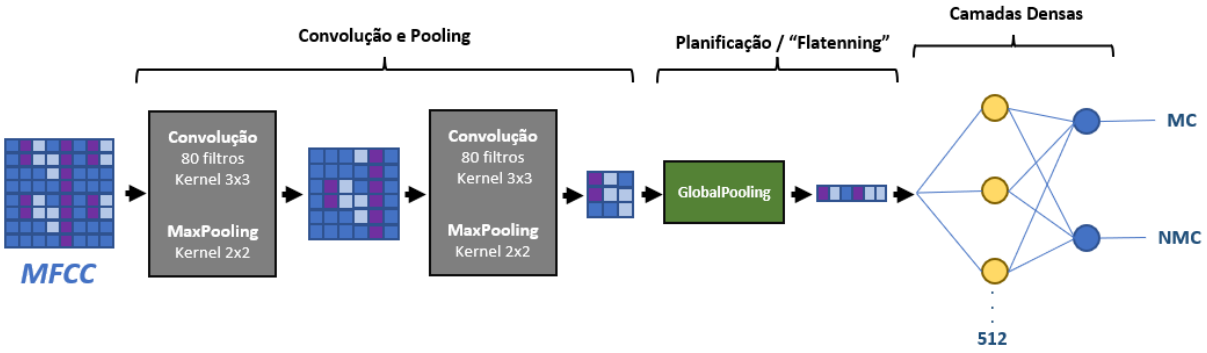


Figura 8. Representação simplificada da topologia da CNN utilizada para classificação dos sinais sonoros
Fonte: Dados originais da pesquisa

Processamento – Função de ativação

Ainda sob influência da topologia proposta por Piczak (2015), todas as camadas intermediárias possuem como função de ativação o ReLU ou “Rectified Linear Unit”, com exceção da camada final que realiza a classificação binária MC / NMC, a qual é ativada por meio da função de ativação Softmax.

O trabalho executado em Zhang e Zou (2017) propõe uma alternativa ao uso do ReLU nas camadas intermediárias, com sua substituição pela função de ativação LeakyReLU, evitando a perda de 50% da informação (quadrante esquerdo ou negativo) que ocorre quando

utilizamos apenas a função ReLU, devido à nulidade de ativação nessa região, conforme a Figura 9 demonstra:

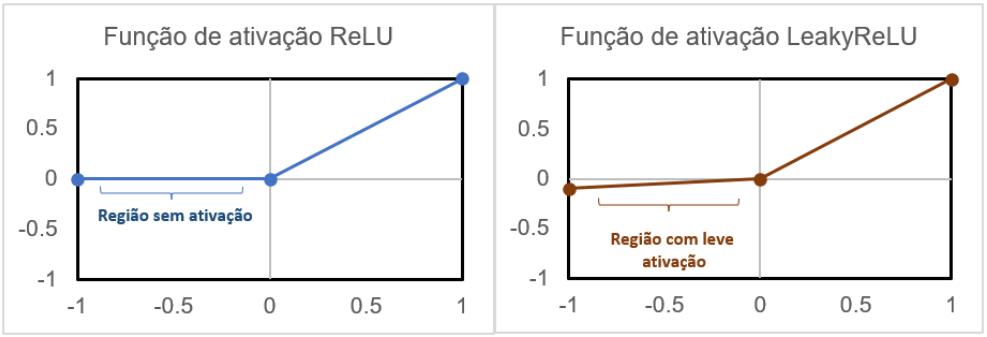


Figura 9. Comparativo gráfico das funções ReLU e LeakyReLU
Fonte: Dados originais da pesquisa

A função de ativação LeakyReLU é definida pela pela eq. (9):

$$f(x) = \{x\ se\ x > 0\ \alpha x\ se\ x < 0 \quad (9)$$

onde x é a variável independente; f(x) é o resultado da ativação; α é o fator de compressão para a zona negativa, nos casos em que x<0.

Tal variação de função de ativação foi explorada no presente trabalho, de modo a verificar se os resultados obtidos em Zhang e Zou (2017) se repetem, com uma melhoria nas métricas de classificação ao adotar-se a função LeakyReLU em detrimento da tradicional ReLU, permitindo que maior parte da informação seja processada sem perda e otimizando o aproveitamento entre o que é fornecido como entrada à rede neural e a esparsidade da informação durante o processamento. Para os experimentos executados no presente trabalho fixou-se o parâmetro α em 0.1 ou 10%.

Métricas de Performance

Uma vez treinado e testado, é importante que tenhamos métricas bem definidas e coerentes com o experimento realizado, de modo a poder realizar comparações de performance válidas e avaliar criticamente a qualidade do modelo construído. Uma métrica muito popular para avaliar modelos de classificação é a matriz de confusão, que é basicamente uma tabela comparativa do que foi classificado em comparação ao que foi observado. No presente caso, onde temos um problema de classificação binária, temos uma matriz de confusão nos moldes do que é mostrado na Figura 10:

		Observação	
		NMC	**MC**
Classificação	**NMC**	Verdadeiro Negativo (VN)	Falso Negativo (FN)
	MC	Falso Positivo (FP)	Verdadeiro Positivo (VP)

Figura 10. Matriz de confusão
Fonte: Dados originais da pesquisa

Deseja-se que as observações MC sejam classificadas corretamente como MC, ao passo em que as observações NMC sejam também classificadas corretamente como NMC, isto é, deseja-se que boa parte ou a totalidade dos valores se situe na diagonal principal da matriz. Todos e quaisquer valores na diagonal secundária são resultantes de uma classificação errônea. Com base nessas premissas, podemos extrair as seguintes métricas[2]:

A Acurácia, por meio da eq. (10), que é uma métrica generalista e que representa a performance geral do modelo:

$$acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{10}$$

A Precisão, por meio da eq. (11), que mede a quantidade de identificações corretas da categoria alvo (no caso, MC) em relação a todas as identificações realizadas para tal categoria:

$$precisão = \frac{VP}{VP + FP} \tag{11}$$

A Sensibilidade, por meio da eq. (12), definida como a proporção de previsões corretas da categoria alvo (MC) em relação ao número total de observações da tal categoria:

$$sensibilidade = \frac{VP}{VP + FN} \tag{12}$$

E, por fim, a Especificidade, por meio da eq. (13), que mede a proporção de previsões negativas (categoria NMC) que foram realizadas corretamente:

[2] As métricas são apresentadas em Kuhn e Johnson (2013), Boehmke e Greenwell (2019).

$$especificidade = \frac{VN}{VN + FP} \quad (13)$$

Dessas quatro métricas, podemos extrair o F1 Score através de eq. (14), o qual consiste na média harmônica entre a precisão e a sensibilidade:

$$F1 = 2 * \frac{precisão * sensibilidade}{precisão + sensibilidade} \quad (14)$$

E o índice J de Youden através de eq. (15), o qual varia entre -1 e 1, onde -1 indica que todas as amostras foram classificadas erroneamente, 0 indica um classificador inútil ou aleatório e 1 indica um classificador perfeito, sem falsos positivos ou falsos negativos (Youden, 1950):

$$J = sensibilidade + especificidade - 1 \quad (15)$$

Resultados e Discussão

Conforme anteriormente descrito, o modelo foi treinado e testado em quatro diferentes cenários, e os resultados comparados entre si. As variações propostas para as diferentes rodadas de treinamento são:

Experimento 1: Modelo treinado com **ReLU** e segmentos de **476ms**.
Experimento 2: Modelo treinado com **LeakyReLU** e segmentos de **476ms**.
Experimento 3: Modelo treinado com **ReLU** e segmentos de **952ms**.
Experimento 4: Modelo treinado com **LeakyReLU** e segmentos de **952ms**.

Os valores específicos de 476 e 952 milissegundos são obtidos com base no critério de otimização baseado em potências de 2 (vide eq. 4). Dado que os recortes de áudio são amostrados em 44100Hz e foram calculados os coeficientes de Fourier em segmentos de 512 (2^9) amostras, temos que a multiplicação de 512 amostras a 44100Hz por 41 e 82 (T segmentos) respectivamente resulta nas durações de 476 e 952 milissegundos, que são as durações padrão arbitradas para os recortes de áudio originados do banco de sons UrbanSound8k.

Para o balanceamento das classes MC e NMC para os experimentos 1 e 2 (476ms) as amostras da categoria MC passaram pelos processos de OS e DA, tal que cada amostra original deu origem a 5 amostras sintéticas. Já o processo de segmentação em amostras padrão foi feito aplicando-se um fator S de sobreposição de 40% nas amostras da categoria MC (conforme eq. 1) tal que se atingiu um balanceamento satisfatório, mostrado na Figura 11. As amostras da categoria NMC foram segmentadas nos tamanhos padrão, porém sem qualquer sobreposição ou aplicação de técnicas de DA e OS.

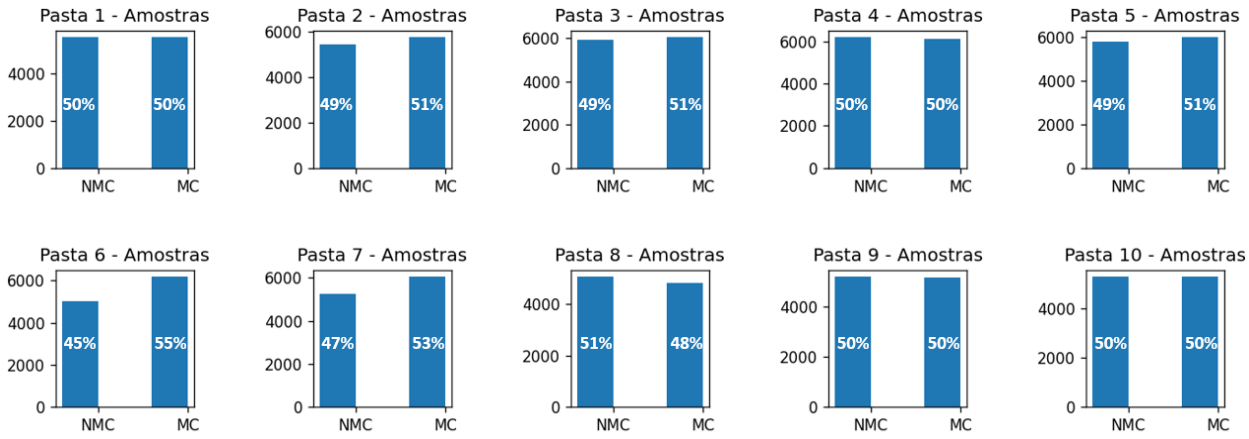


Figura 11. Balanceamento de classes para a amostragem em 476ms
Fonte: Resultados originais da pesquisa

O processo de balanceamento das classes MC e NMC para os experimentos 3 e 4 (952ms) seguiu exatamente as mesmas premissas e parametrizações daquele realizado para os experimentos 1 e 2, resultando em um balanceamento novamente satisfatório, o qual é graficamente demonstrado na Figura 12.

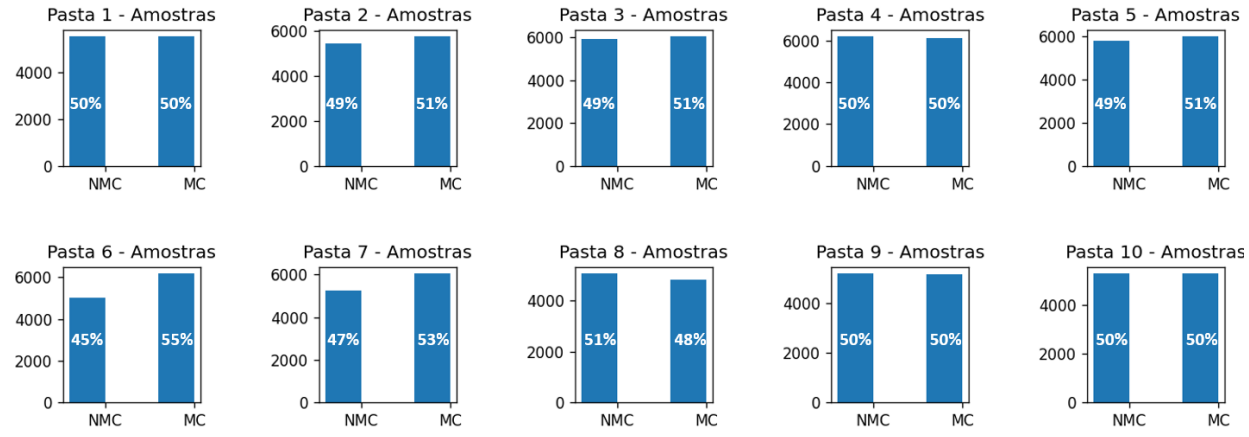


Figura 12. Balanceamento de classes para a amostragem em 952ms
Fonte: Resultados originais da pesquisa

Para cada um dos experimentos o modelo foi treinado dez vezes com a realização de validações cruzadas, onde 9 das pastas do banco de sons UrbanSound8k foram utilizadas para o treinamento, e a pasta restante para o teste. Na primeira rodada de treinamento (#1) utilizou-se a pasta 1 para o teste e as pastas 2 a 10 para o treinamento do modelo, progredindo sequencialmente até a última rodada (#10), onde utilizou-se a pasta 10 para o teste e as pastas 1 a 9 para o treino. Cada rodada de treinamento foi realizada com 20 épocas e ao fim do treinamento os pesos da CNN foram restaurados para a época que apresentou a maior acurácia nos dados de teste, de maneira a obtermos o melhor modelo que generalize bem para os dados de teste sem “overfit” aos dados de treino, conforme ilustrado na Figura 13.

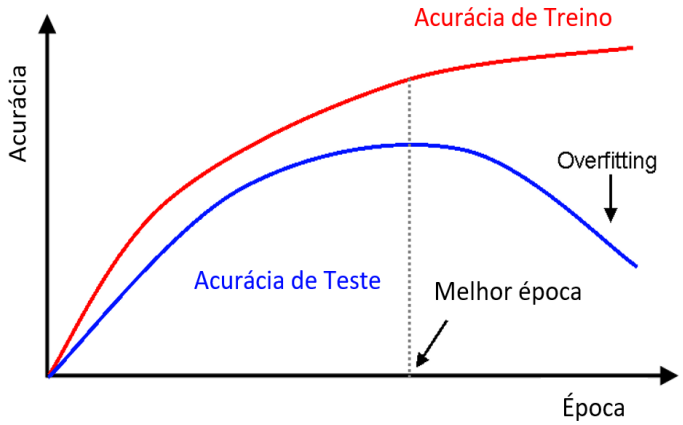


Figura 13. Regularização dos pesos de acordo com a época de melhor acurácia de teste
Fonte: Data Science Academy (2022) – Traduzido e Adaptado

Os resultados após a realização do Experimento 1 são mostrados na Tabela 1:

Tabela 1. Resultados dos testes de validação cruzada para o Experimento 1

Métrica	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Acurácia (%)	88,5	82,3	85,6	82,1	85,0	85,1	83,3	87,5	96,4	87,5
Precisão (%)	92,1	81,0	91,6	90,3	93,5	86,9	88,7	88,2	97,0	86,8
Sensibilidade (%)	84,1	85,5	78,7	84,8	75,9	86,1	78,9	85,8	95,6	88,3
Especificidade (%)	92,8	78,8	92,6	79,5	94,5	84,0	88,4	89,1	97,1	86,6
F1-Score (%)	87,9	83,2	84,6	82,5	83,8	86,5	83,5	87,0	96,3	87,6
Youden J (-)	0,77	0,64	0,71	0,64	0,70	0,70	0,67	0,75	0,93	0,75

Fonte: Resultados originais da pesquisa

A progressão das acurácias durante o treino e o teste para o Experimento 1 é mostrada na Figura 14:

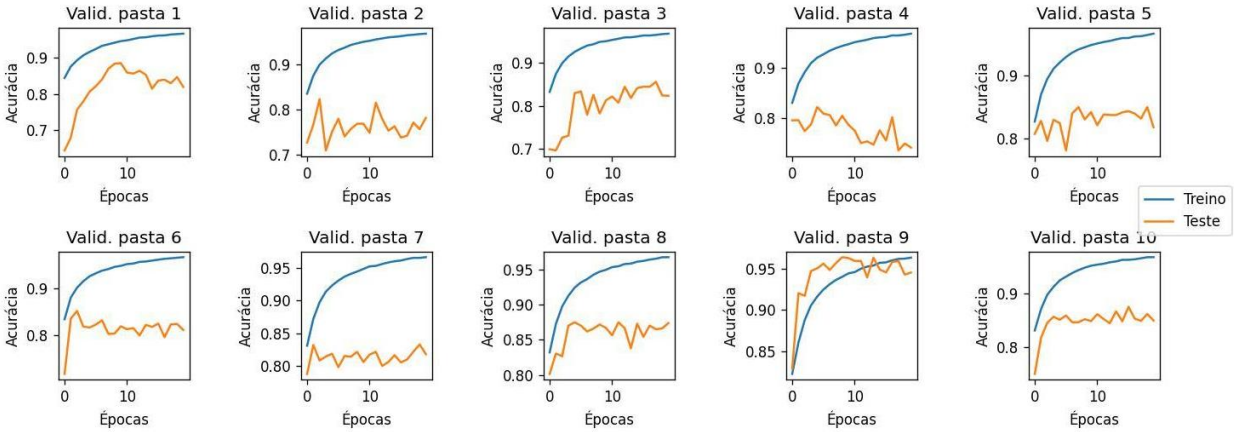


Figura 14. Progressão da acurácia de treino e teste para o Experimento 1
Fonte: Resultados originais da pesquisa

Por fim, as matrizes de confusão normalizadas para o Experimento 1 são mostradas na Figura 15:

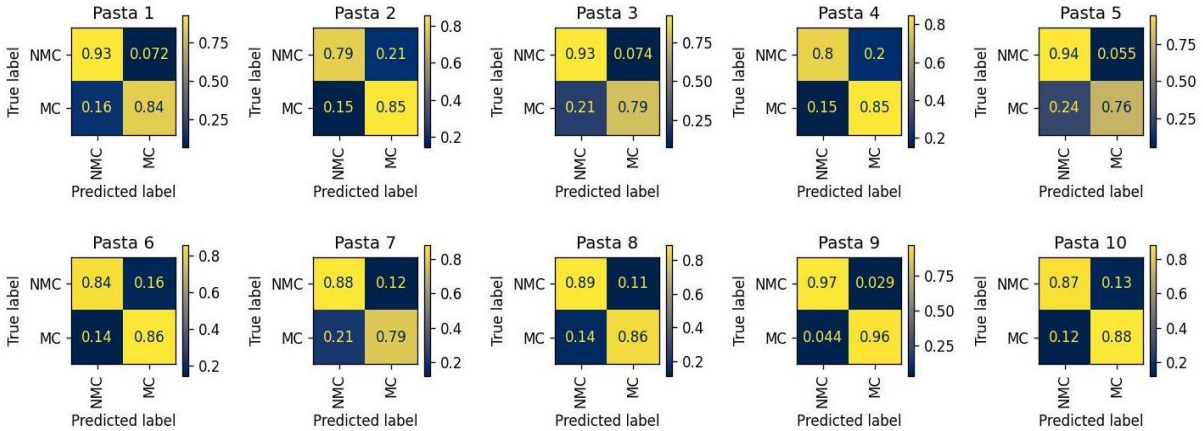


Figura 15. Matrizes de confusão normalizadas para o Experimento 1
Fonte: Resultados originais da pesquisa

Os resultados após a realização do Experimento 2 são mostrados na Tabela 2:

Tabela 2. Resultados dos testes de validação cruzada para o Experimento 2

Métrica	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Acurácia (%)	88,1	82,9	85,4	83,4	85,4	83,5	81,3	87,4	96,9	86,8
Precisão (%)	90,1	82,1	86,8	84,2	93,4	90,2	85,4	89,5	97,2	85,2
Sensibilidade (%)	85,4	85,3	83,7	81,9	76,8	78,7	78,7	84,0	96,6	89,0
Especificidade (%)	90,7	80,3	87,1	84,8	94,3	89,5	84,4	90,7	97,2	84,5
F1-Score (%)	87,7	83,6	85,2	83,0	84,3	84,0	81,9	86,7	96,9	87,1
Youden J (-)	0,76	0,66	0,71	0,67	0,71	0,68	0,63	0,75	0,94	0,74

Fonte: Resultados originais da pesquisa

A progressão das acurácias durante o treino e o teste para o Experimento 2 é mostrada na Figura 16:

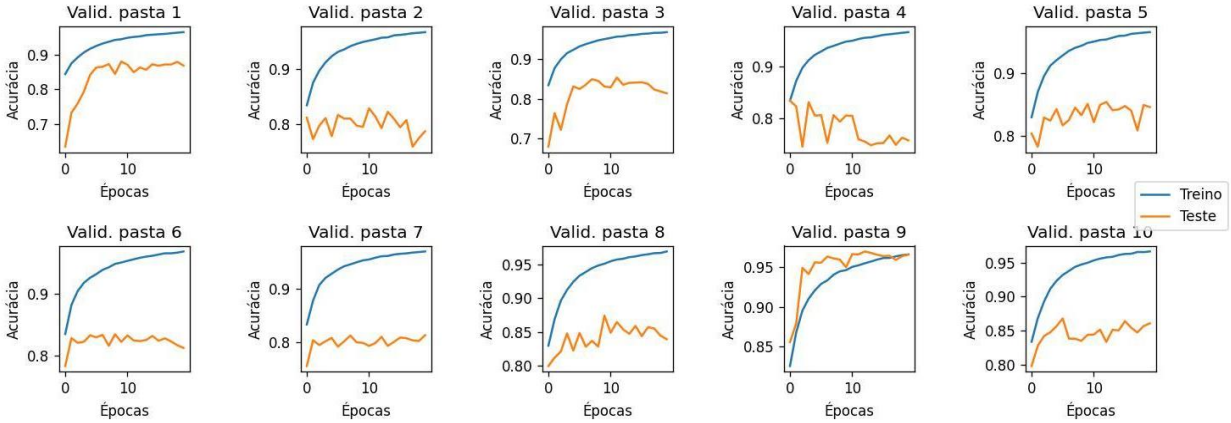


Figura 16. Progressão da acurácia de treino e teste para o Experimento 2
Fonte: Resultados originais da pesquisa

Por fim, as matrizes de confusão normalizadas para o Experimento 2 são mostradas na Figura 17:

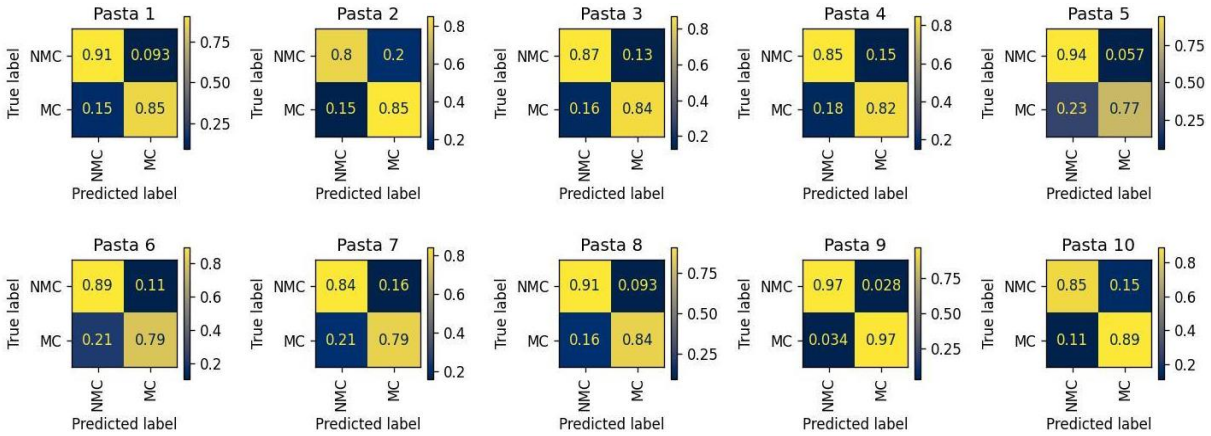


Figura 17. Matrizes de confusão normalizadas para o Experimento 2
Fonte: Resultados originais da pesquisa

Os resultados após a realização do Experimento 3 são mostrados na Tabela 3:

Tabela 3. Resultados dos testes de validação cruzada para o Experimento 3

Métrica	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Acurácia (%)	84,0	82,3	86,4	84,5	86,6	85,3	83,7	88,5	96,2	87,0
Precisão (%)	83,9	80,1	88,0	87,5	89,0	93,4	82,4	89,6	87,6	81,6
Sensibilidade (%)	82,8	86,0	83,5	79,1	83,2	78,3	87,5	85,5	94,5	94,4
Especificidade (%)	85,0	78,7	89,1	89,5	90,0	93,6	79,7	91,2	97,8	80,0
F1-Score (%)	83,3	82,9	85,7	83,1	86,0	85,2	84,8	87,5	96,0	87,5
Youden J (-)	0,68	0,65	0,73	0,69	0,73	0,72	0,67	0,77	0,92	0,74

Fonte: Resultados originais da pesquisa

A progressão das acurácias durante o treino e o teste para o Experimento 3 é mostrada na Figura 18:

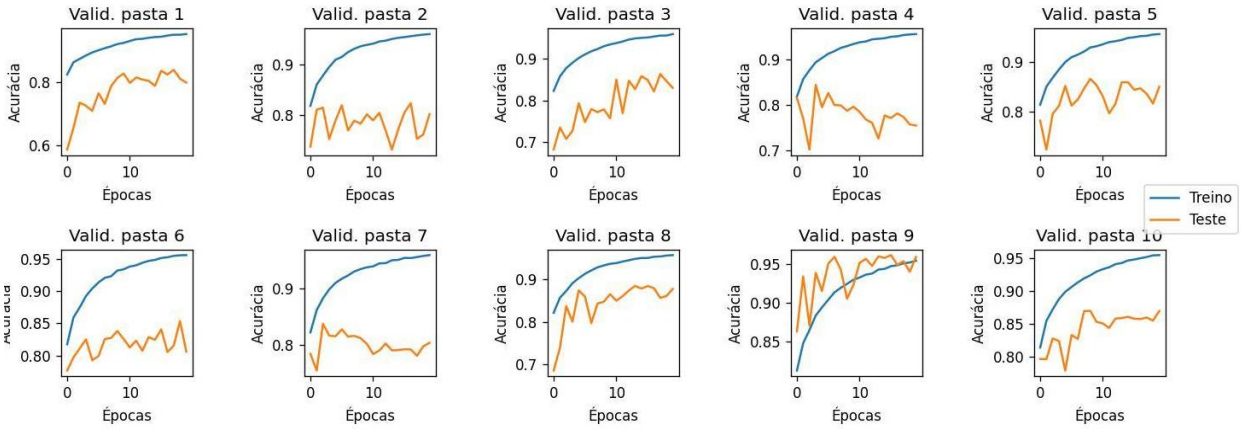


Figura 18. Progressão da acurácia de treino e teste para o Experimento 3
Fonte: Resultados originais da pesquisa

Por fim, as matrizes de confusão normalizadas para o Experimento 3 são mostradas na Figura 19:

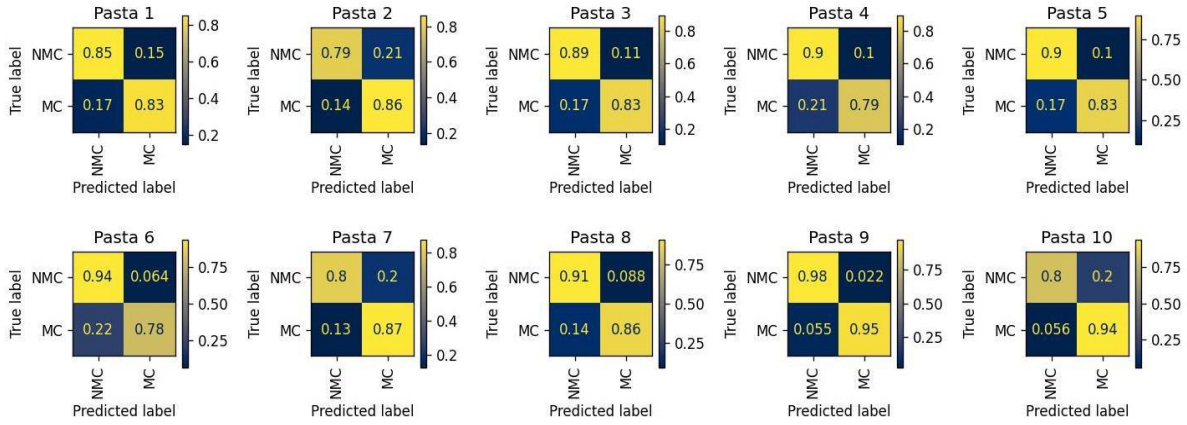


Figura 19. Matrizes de confusão normalizadas para o Experimento 3
Fonte: Resultados originais da pesquisa

Os resultados após a realização do Experimento 4 são mostrados na Tabela 4:

Tabela 4. Resultados dos testes de validação cruzada para o Experimento 4

Métrica	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Acurácia (%)	83.3	79.8	84.2	83.4	84.8	85.0	83.8	88.6	96.4	90.2
Precisão (%)	82.4	75.8	86.8	84.6	86.5	83.7	85.2	88.7	95.6	88.0
Sensibilidade (%)	83.3	87.5	79.8	80.0	82.0	89.4	83.4	86.8	97.1	92.4
Especificidade (%)	83.3	72.1	88.4	86.5	87.6	79.8	84.3	90.2	95.8	88.2
F1-Score (%)	82.8	81.2	83.1	82.2	84.2	86.5	84.3	87.7	96.3	90.2
Youden J (-)	0.67	0.60	0.68	0.66	0.70	0.69	0.68	0.77	0.93	0.81

Fonte: Resultados originais da pesquisa

A progressão das acurácias durante o treino e o teste para o Experimento 4 é mostrada na Figura 20:

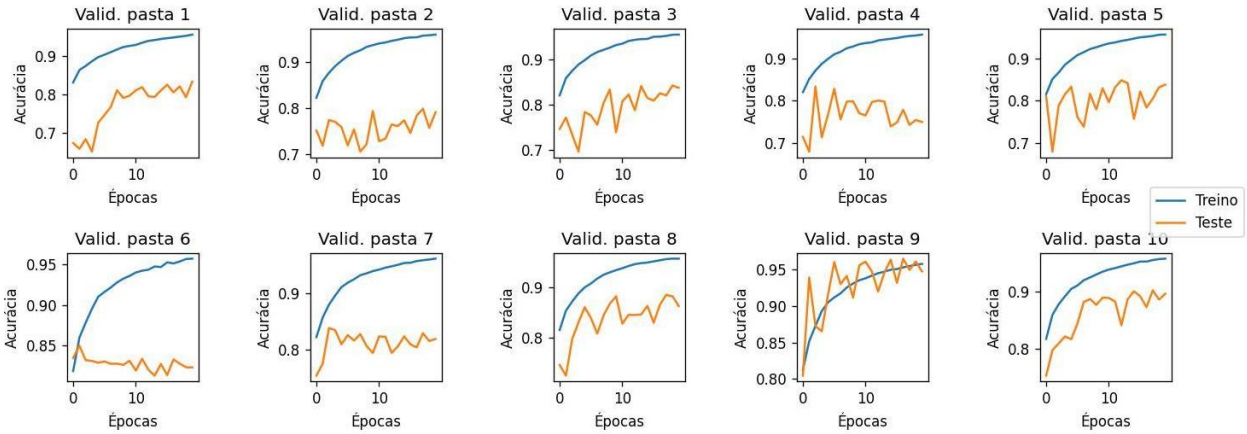


Figura 20. Progressão da acurácia de treino e teste para o Experimento 4
Fonte: Resultados originais da pesquisa

Por fim, as matrizes de confusão normalizadas para o Experimento 4 são mostradas na Figura 21:

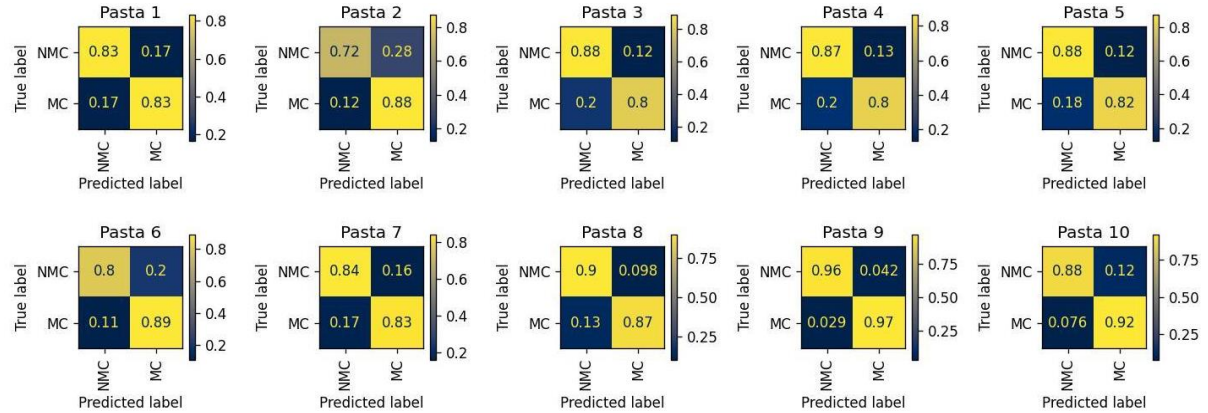


Figura 21. Matrizes de confusão normalizadas para o Experimento 4
Fonte: Resultados originais da pesquisa

Finalizados os quatro experimentos propostos, objetivou-se então analisar se há entre eles alguma diferença estatisticamente relevante que justifique a utilização de diferentes tamanhos de segmentos ou mesmo a aplicação da função de ativação de tipo LeakyReLU para obter melhor performance. Para tal, dado o número pequeno de amostras por experimento (apenas 10) utilizamos primeiramente o teste de Shapiro-Wilk, conforme descrito em Fávero e Belfiore (2017), de maneira a verificar se os indicadores J de Youden e F1-Score de todos os experimentos se distribuem de maneira normal. Os resultados obtidos são mostrados na Tabela 5.

Tabela 5. Resultados dos testes de normalidade para o J de Youden e F1-Score

Métrica	P-Valor Exp#1	P-Valor Exp#2	P-Valor Exp#3	P-Valor Exp#4
F1-Score (%)	0.0128	0.0076	0.0058	0.0622
Youden J (-)	0.0339	0.0190	0.0157	0.0565

Fonte: Resultados originais da pesquisa

Com base nos P-Valores obtidos, podemos afirmar que para um nível de significância de 1% os valores em questão se distribuem de maneira normalizada, não rejeitando-se, portanto, a hipótese nula. Então, procedemos com uma análise do tipo ANOVA, também descrita em Fávero e Belfiore (2017), de modo a analisar uma possível diferença estatística entre os quatro experimentos nos indicadores J de Youden e F1-Score. Os resultados são apresentados na Tabela 6:

Tabela 6. Resultados dos testes de ANOVA para o J de Youden e F1-Score

Métrica	P-Valor ANOVA
F1-Score (%)	0.9935
Youden J (-)	0.9957

Fonte: Resultados originais da pesquisa

Mais uma vez, foi possível concluir que para um intervalo de confiança de 99% que as quatro amostras provenientes dos quatros experimentos são estatisticamente iguais, não havendo, portanto, nenhum ganho mesurável quando se alterou a função de ativação e/ou o tamanho do segmento das amostras de áudio. Para finalizar a análise, na Figura 22 é possível ter um panorama visual de que, de fato, as amostras são estatisticamente indiferentes.

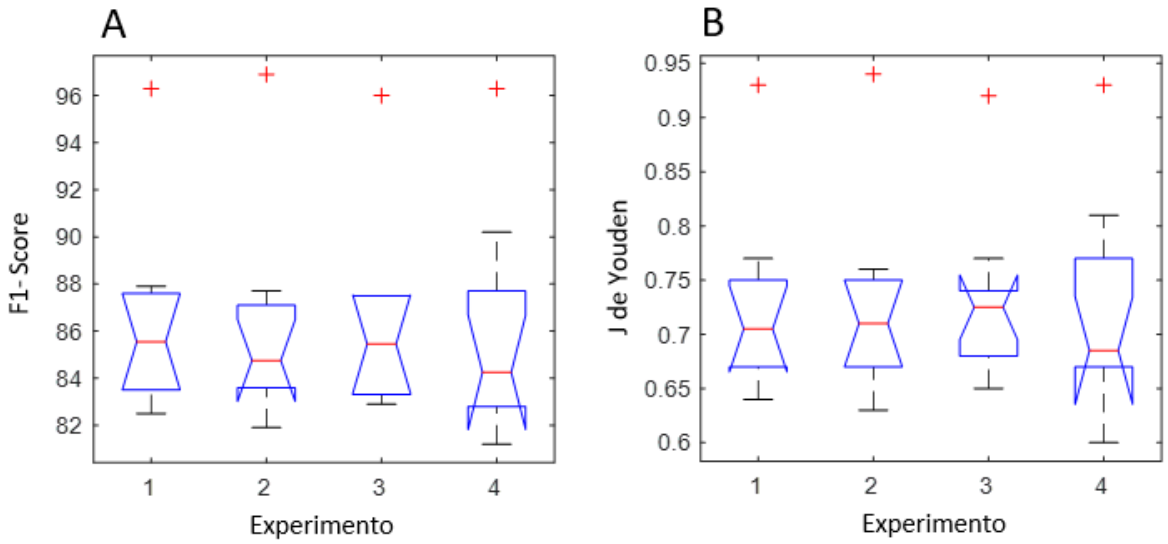


Figura 22. Gráfico do tipo “box-plot” contendo a média e variância para F1-Score (A) e J de Youden (B) em cada um dos experimentos executados
Fonte: Resultados originais da pesquisa

Conclusões

O propósito principal do trabalho aqui proposto consistiu em avaliar a viabilidade de se implementar, através de redes neurais, um classificador de sons capaz de identificar a presença ou não de um motor a combustão em funcionamento dentro de recortes sonoros, por meio da análise de seus espectrogramas em função do tempo. No que tange à essa finalidade, o trabalho apresentou bons resultados que se refletem nos altos índices de F1-Score e J de Youden, com valores absolutos sempre numa média de 85% em suas respectivas escalas. Desse modo, seria possível avançar nos estudos e trabalhos para prototipar aplicações que dependam de tal processamento sonoro, abrindo caminho para outras derivações de um classificador de tal tipo.

Como propósito secundário, avaliou-se o impacto tanto do tamanho do segmento sonora para análise (476ms e 952ms) como da função de ativação (ReLU e LeakyReLU 10%), repetindo-se a etapa de treino, teste e validação para cada uma das quatro combinações possíveis de variação desses parâmetros, no que foram chamados ‘quatro experimentos’. Essa etapa resultou que, estatisticamente, não há diferença significante entre os quatro experimentos e, portanto, a alteração dos parâmetros não melhorou ou piorou performance global do modelo, possuindo um impacto indiferente. Estudos posteriores são encorajados com diferentes variações destes e de outros parâmetros, e só não foram aqui executados devido ao alto custo computacional para repetir cada um dos experimentos, o que demandaria uma janela de tempo mais longa para a conclusão do trabalho.

Agradecimento

Agradeço primeiramente a Deus pela oportunidade de realizar estudos de pós-graduação. Agradeço também a minha família, especialmente aos meus pais Marco e Ioná e minha esposa Jennifer, que são os maiores incentivadores do meu progresso intelectual.

Referências

Analytics Vidhya. 2022. Vehicle Sound Classification Using Deep Learning. Disponível em <https://www.analyticsvidhya.com/blog/2022/01/vehicle-sound-classification-using-deep-learning>. Acesso em 3 de outubro de 2022.

Bishop, C. 1995. Training with Noise is Equivalent to Tikhonov Regularization. Neural Computation, vol.7, no.1: 108-116.

Bhatia, R. 2018. How do Machine Learning algorithms differ from traditional algorithms? Analytics India Magazine. Disponível em: <https://analyticsindiamag.com/how-do-machine-learning-algorithms-differ-from-traditional-algorithms>. Acesso em 22 de dezembro de 2022.

Boehmke, B.; Greenwell, B. 2019. Hands-on machine learning with R. Chapman and Hall/CRC, New York, NY, USA.

Ciric, D; Peric, Z; Nikolic, J; Vucic, N. 2021. Audio Signal Mapping into Spectrogram-Based Images for Deep Learning Applications. 2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH), 2021, Sarajevo, Bosnia e Herzegovina: 1-6.

Chen, H; Hu, N; Cheng, Z. 2019. A deep convolutional neural network based fusion method of two-direction vibration signal data for health state identification of planetary gearboxes. Measurement, Volume 146. 268-278.

Data Science Academy [DSA]. 2022. Deep Learning Book. Disponível em <https://www.deeplearningbook.com.br>. Acesso em 29 de dezembro de 2022.

Devopedia. 2021. Audio Feature Extraction. Disponível em: <https://devopedia.org/audio-feature-extraction>. Acesso em 26 de dezembro de 2022.

Fávero, L.P.; Belfiore, P. 2017. Manual de Análise de Dados - Estatística e Modelagem Multivariada com Excel®, SPSS® e Stata®. 1ed. Elsevier Editora, Rio de Janeiro, RJ, Brasil.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., Herrera, F. 2018. Learning from imbalanced data sets. Springer, Berlin, Alemanha.

Hasan, R; Jamil, M; Rabbani, G; Rahman, S. 2004. Speaker identification using Mel Frequency Cepstral Coefficients. 3rd International Conference on Electrical & Computer Engineering (ICECE 2004), 2004, Dhaka, Bangladesh.

Kemalkar, A.K.; Bairagi, V.k.. 2016. Engine Fault Diagnosis Using Sound Analysis. International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT): 943-946

Kuhn, M., e Johnson, K. 2013. Applied predictive modeling. Springer, New York, NY, USA.

Krawczyk, Bartosz. 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence 5 (4): 221-232.

Mohammadi, M.; Al-Fuqaha, A. 2018. Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges. IEEE Communications Magazine, vol. 56, no. 2: 94-101.

Nickischer, A. 2020. Environmental Impacts of Internal Combustion Engines and Electric Battery Vehicles. D.U. Quark, Volume #4 (Issue #2): 21-31.

Piczak, K. 2015. Environmental sound classification with convolutional neural networks. 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 2015: 1-6.

Sahidullah, M; Goutam Saha, G. 2011. Design, analysis and experimental evaluation of block-based transformation in MFCC computation for speaker recognition. Speech Communication, Volume 54, Issue 4, 2012: 543-565.

Salamon, J; Bello, J. 2016. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279-283, March 2017.

Salamon, J; Jacoby, C; Bello, J. 2014. A Dataset and Taxonomy for Urban Sound Research. 22nd ACM International Conference on Multimedia, 2014, Orlando, Flórida, Estados Unidos.

Steven, S; Volkmann, J; Newman, E. 1937. A scale for the measurement of the psychological magnitude pitch. Journal of the Acoustical Society of America, 8, 185–190.

Song, X; Cong, Y; Song, Y. 2021. A bearing fault diagnosis model based on CNN with wide convolution kernels. Journal of Ambient Intelligence and Humanized Computing 13: 4041–4056.

Tang, S.; Yuan, S.; Zhu, Y. 2020. Data Preprocessing Techniques in Convolutional Neural Network based on Fault Diagnosis towards Rotating Machinery. *IEEE Access*, vol. 8: 149487-149496.

Wu, Z.; Wan, Z.; Ge, D. et al. 2022. Car engine sounds recognition based on deformable feature map residual network. Sci Rep 12, 2744 (2022).

Youden, W. 1950. Index for rating diagnostic tests. Cancer, 3: 32-35.

Zhang, X; Zou, Y. 2017. Dilated Convolution Neural Network with LeakyReLU for Environmental Sound Classification. 2017 22nd International Conference on Digital Signal Processing (DSP), 2017: 1-5

Zhou, G; Chen, Y; Chien, C. 2022. On the analysis of data augmentation methods for spectral imaged based heart sound classification using convolutional neural networks. BMC Medical Informatics and Decision Making 22, 226 (2022).

Zhao, Y; Zhang, H; An, L; et al. 2018. Improving the approaches of traffic demand forecasting in the big data era. Cities, Volume 82: 19-26.