UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DANIEL PEIXOTO PINTO DA SILVA

ANÁLISE DE INTERPRETABILIDADE EM MODELO PROFUNDO PARA DETECÇÃO DE INSUFICIÊNCIA RESPIRATÓRIA: UM ESTUDO DE CASO PARA A COVID-19

MEDIANEIRA

2021

DANIEL PEIXOTO PINTO DA SILVA

ANÁLISE DE INTERPRETABILIDADE EM MODELO PROFUNDO PARA DETECÇÃO DE INSUFICIÊNCIA RESPIRATÓRIA: UM ESTUDO DE CASO PARA A COVID-19

Interpretability Analysis in Deep Model for Detection of Respiratory Failure: a Case Study for COVID-19

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof. Dr. Arnaldo Candido Junior.

Coorientador(a): Prof. Dr. Marcelo Finger.

MEDIANEIRA

2021

DANIEL PEIXOTO PINTO DA SILVA

ANÁLISE DE INTERPRETABILIDADE EM MODELO PROFUNDO PARA DETECÇÃO DE INSUFICIÊNCIA RESPIRATÓRIA: UM ESTUDO DE CASO PARA A COVID-19

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 06 dezembro 2021

_____

Arnaldo Candido Junior
Doutorado
Universidade Tecnológica Federal do Paraná – Campus Medianeira

_____

Jorge Aikes Junior
Mestrado
Universidade Tecnológica Federal do Paraná – Campus Medianeira

_____

Marcelo Finger
Doutorado
Universidade de São Paulo

_____

Pedro Luiz de Paula Filho
Doutorado
Universidade Tecnológica Federal do Paraná – Campus Medianeira

MEDIANEIRA

2021

RESUMO

A COVID-19 é uma doença que afetou todo o mundo, sendo declarada uma pandemia pela Organização Mundia da Saúde. Todos os métodos criados para a detecção dessa doença são custosos e requerem que o paciente esteja em condições específicas pra que o diagnostico seja correto. Assim, neste trabalho analisou-se o estado da arte em detecção de insuficiência respiratória decorrente de COVID-19 por ANNs para verificar vieses e trazer explicações sobre o resultado da classificação dessa ANN. Para isso foi feito teste de ablação ao incluir novas informações como entrada ao modelo, utilizou-se o algoritmo Grad-CAM para ressaltar qual partes do dado alimentado à ANN são mais importantes. Também, foram sintetizados áudios do produto entre um áudio original e o mapa de calor do Grad-CAM para permitir a análise sonora dos resultados. Além disso foi treinado uma PANN e também treinou-se a SpiraNet com as técnicas Mixup e SpecAugment para tentar superar o estado da art. Os resultados do teste de ablação mostraram uma grande importância da frequência fundamental da voz e do espectrograma de mel. Já os áudios sintetizados mostraram que silabas tônicas e palavras prolongadas são importantes para a classificação de pacientes neste trabalho. O experimento com aumento de dados não obteve resultados significativos. E por fim, foi superado o estado da arte com a PANN treinada obtendo uma acurácia de 94,44%.

Palavras-chave: Inteligência Artificial; Rede Neural Artificial; Aprendizado do computador.

ABSTRACT

COVID-19 is a disease that has affected the whole world, being declared a pandemic by the World Health Organization. All methods created to detect this disease are costly and human, which is available in specific conditions so that the diagnosis is correct. Thus, in this work, the state of the art in detecting respiratory failure due to COVID-19 by ANNs was analyzed to verify biases and provide explanations about the result of the classification of this ANN. For this, an ablation test was performed, including new information as input to the model, the Grad-CAM algorithm was used to highlight which parts of the data fed to the ANN are more important. Also, audios of the product between an original audio and the Grad-CAM heat map were synthesized to allow a sound analysis of the results. In addition, a PANN was trained and also Mixup and SpecAugment techniques were used on training of SpiraNet to overcome the state of art. The results of the ablation test showed a great importance of the fundamental frequency of the voice and the melspectrogram. The synthesized audios showed that stressed syllables and prolonged words are important for the classification of patients in this work. The data augmentation experiment did not obtain significant results. And finally, the state of the art was surpassed with the trained PANN obtaining an accuracy of 94,44%.

Keywords: Artificial Intelligence; Artificial Neural Network; Machine Learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANN	Artificial Neural Network
CAM	Class Activation Mapping
CNN	Convolutional neural network
COVID-19	Coronavirus disease 2019
F0	Frequência Fundamental da Voz
GAP	Global Average Pooling
Grad-CAM	Gradient-weight Class Activation Mapping
MLP	MultiLayer Perceptron
PANN	Pretrained Audio Neural Networks
ReLU	Rectified Linear Unit
SPIRA	Sistema de detecção Precoce de Insuficiência Respiratória por meio de análise de Áudio
XOU	Ou exclusivo

SUMÁRIO

1 INTRODUÇÃO

Em dezembro de 2019, uma nova espécie do vírus coronavírus foi descoberta, denominada SARS-CoV-2 causadora da doença Coronavirus disease 2019 (COVID-19). Essa variante provou-se mais contagiante e letal, por isso, espalhou-se mundialmente em um curto período de tempo. A COVID-19 foi declarada como uma pandemia pela Organização Mundial da Saúde (OMS), desse modo, fez com que os países criassem medidas restritivas severas como o *lockdown* (VENTURA *et al.*, 2021).

Entretanto, o Brasil é o país com mais mortes e contaminados da América Latina (LANCET, 2020), pois, a COVID-19, além do seu alto contágio, manifesta-se com os sintomas iniciais de um resfriado comum (ISER *et al.*, 2020). Isso auxiliou a propagação do vírus no país ao criar aglomerações em hospitais e postos de saúde, que poderiam ter sido evitadas caso a triagem dos pacientes fosse mais eficiente. Além disso, os métodos criados para efetuar o diagnóstico da doença são caros e precisam que o paciente esteja em condições específicas para que o resultado seja correto (MAGNO *et al.*, 2020).

A fim de otimizar a triagem dos pacientes, e possivelmente reduzir seus custos, surgiu o projeto *Sistema de detecção Precoce de Insuficiência Respiratória por meio de análise de Áudio*[1] (SPIRA). Ele visa a detecção automática do sintoma de insuficiência respiratória em estágios inciais do vírus. A detecção é realizada por uma Rede Neural Artificial (ANN – Artificial Neural Network), e isso, facilita a triagem de pacientes que precisam procurar atendimento médico-hospitalar.

Assim, no projeto SPIRA foi criada uma ANN por Casanova *et al.* (2021) que provou-se eficaz no conjunto de testes criado, porém, os dados utilizados para o treino e teste dessa ANN possuem ruídos devido a forma como a coleta foi realizada. Os áudios de pacientes portadores de COVID-19 foram coletados em ambiente hospitalar. Enquanto que os áudios do grupo de controle, foram coletados por voluntários via *web*, na plataforma do projeto.

Contudo, para a rede ser mais confiável e validar que ela não possui nenhum viés, é necessário realizar um estudo mais profundo nos critérios que foram importantes para a classificação do resultado. Porém, isso não é uma tarefa simples, pois, ANNs são modelos

[1] Plataforma do projeto SPIRA: `https://spira.ime.usp.br/coleta/`

"caixa preta" e dependem da utilização de métodos e algoritmos que consigam explicar seus parâmetros.

1.1 Objetivo Geral e Específicos

Esse trabalho tem como objetivo geral analisar o comportamento do modelo neural de Casanova *et al.* (2021) e investigar o seu comportamento. Os objetivos específicos são:

- Treinar modelos derivados da ANN de Casanova *et al.* (2021) com o objetivo de analisar o funcionamento interno desses modelos;
- Realizar teste de ablação para descobrir quais dados são mais importantes para a ANN;
- Aplicar o algoritmo Grad-CAM (SELVARAJU *et al.*, 2016b) para gerar mapas de calores para cada experimento do teste de ablação a fim de indicar as principais regiões usadas para tomada de decisão da ANN;
- Ressintetizar os áudios combinados com os mapas de calor obtidos;
- Utilizar transferência de aprendizado e técnicas de aumento de dados para tentar superar a acurácia do estado da arte;
- Trazer justificativas e hipóteses para o comportamento da ANN na detecção da COVID-19;
- Verificar se o ambiente de coleta influenciou nas decisões da ANN, visto que áudios hospitalares tem ruídos característicos.

1.2 Justificativa

Algoritmos descritivos de aprendizado de máquina (SKANSI, 2018), como árvores de decisão (GÉRON, 2019), conseguem ter seus resultados justificados sem a necessidade de um estudo profundo ou algum outro algoritmo que explique-os. Porém, em diversos problemas complexos é necessário o uso de ANNs, e, isso traz a impossibilidade de uma análise profunda dos meios que levaram o modelo a uma decisão desses modelos preditivos.

Dessa forma, a análise dos dados tornam-se subjetivas, porque, não há como deduzir

os critérios que a rede levou em consideração para a resposta. Assim, diversas vezes, os ajustes do modelo são feitos de forma heurística. Isso aumenta a quantidade de treinos necessários para chegar no resultado ótimo e dificulta a análise de enviesamento no modelo gerado.

Por isso, encontrar e utilizar um meio de analisar as ANNs é de extrema importância para criação de modelos ótimos. Já que isso permite encontrar vieses no conjunto de dados como ruídos predominantes em uma classe e fazer um *fine tunning* na rede mais apropriada.

1.3 Organização do Documento

Esse documento está organizado da seguinte forma. O Capítulo 2 apresenta os principais assuntos para entender o trabalho. O Capítulo 3, mostra os materiais e métodos que serão utilizados para realizar o trabalho. O Capítulo 4 apresenta os resultados obtidos após a conclusão de todos os experimentos. E por fim é realizada a conclusão do documento no Capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados conceitos sobre áudios, ANNs e processamento de áudios com ANNs, que serão utilizados para a realização do trabalho.

2.1 Áudio

Áudio é a representação eletrônica de um som, e, é com ele que é possível armazenar sons em dispositivos de armazenamento, e reproduzi-los com fidelidade (FONSECA, 2007). Já o som é a uma distorção na pressão do ar que pode ser perceptível pelos ouvidos humanos (DOWNEY, 2016). Porém, para armazenar o som é necessário convertê-lo em um sinal digital, já que, ele é uma onda continua e os dispositivos eletrônicos são discretos (digitais), o que demandaria um armazenamento com memória infinita para armazenar o som em sua totalidade. Essa conversão é feita utilizando métodos como amostragem e quantização (RUSSELL; NORVIG, 2013).

2.1.1 Propriedades

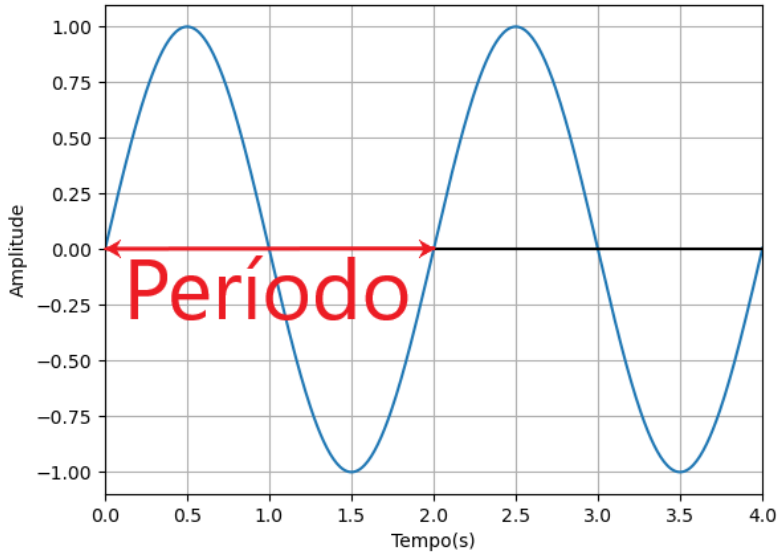
O áudio herda algumas propriedades do som por ser sua representação eletrônica. Suas propriedades mais importantes são a frequência, intensidade, taxa de amostragem e resolução. Cada propriedade descreve um aspecto importante do áudio, sendo assim, é possível a criação de sistemas inteligentes a partir delas.

A frequência é a grandeza que mede distorção do meio em um determinado período (T). Essa distorção gera uma onda que se irradia para fora da fonte de perturbação (BALLOU,

2008). Sua unidade de media é o Hertz (Hz) e também pode ser chamada de ciclos por segundo. Seu período é o tempo, em segundos, que a onda leva para completar um ciclo (HORNER *et al.*, 2005). A velocidade que essa onda oscila é calculada utilizando um segundo de referência e é definida pela Equação 1. A Figura 1 mostra uma onda senoide com frequência de $0,5Hz$ e com período de 2 segundos ($\frac{1}{0,5}$).

$$f = \frac{1}{T} \tag{1}$$

Figura 1 – Forma de onda de uma senoide de $0,5Hz$



Fonte: Autoria própria (2021)

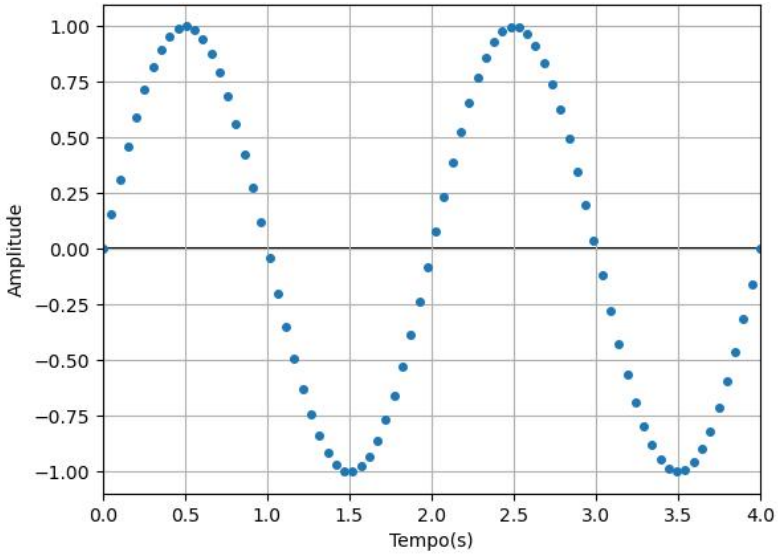
A intensidade do som, também conhecida como volume, define amplitude da onda. O som mais intenso é produzido por uma fonte que possui oscilações de maior amplitude e maiores máximos e mínimos de pressão (WINER, 2018). A unidade de medida padrão do volume é o decibel (dB). Porém, ele não é uma unidade absoluta como o Kelvin, mas sim, é uma comparação da amplitude do som com um valor de referência. Por exemplo, no caso da unidade Volt, tem-se como convenção que $0dBV = 1V$. Assim, para calcular a intensidade, utilizando o voltagem como referência, aplica-se a Equação 2, em que o E_2 é $1V$.

$$dB = 20\log\frac{E_1}{E_2} \tag{2}$$

Para converter som em áudio, os dispositivos de captura registram em instantes regulares a tensão produzida pelo som para conseguir criar uma representação aproximada da forma da onda analógica (WINER, 2018). Assim, a taxa de amostragem, ou *sample rate*, refere-se ao intervalo que a tensão será capturada, ela é medida em Hz. A taxa de amostragem padrão é

$44,1kHz$, já que, conforme o teorema de Nyquist a frequência captada pelo áudio será a metade da frequência de amostragem (FONSECA, 2007). Assim a frequência padrão irá resultar em gravações com no máximo $20kHz$, sendo essa, a maior frequência audível pelo ser humano. A Figura 2 apresenta uma senoide de $0,5Hz$ com taxa de amostragem de $20Hz$, significa que a cada um segundo serão armazenadas 20 amostras.

Figura 2 – Áudio de uma senoide de $0,5Hz$ com uma taxa de amostragem de $20Hz$



Fonte: Autoria própria (2021)

Após amostrar o áudio é necessário converter os valores para bits, para que os dispositivos digitais consigam processá-lo. Logo a resolução mede, em bits, a precisão para armazenar cada valor da amostra capturada (WINER, 2018). A resolução padrão é de 16 bits, mas também, utilizam-se outras como 24 bits (FONSECA, 2007). Dessa forma, quanto maior a fidelidade do áudio maior deverá ser a resolução e a taxa de amostragem. Enquanto a taxa de amostragem determina o tamanho do salto no eixo tempo entre uma amostra e outra, a taxa de bits determina o tamanho do salto no eixo amplitude.

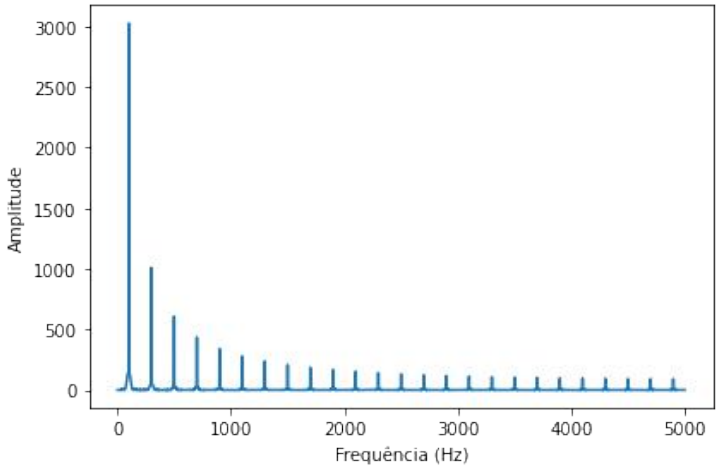
Essas propriedades permitem a extração de novas informações ao serem combinadas com transformadas ou escalas diferentes. Isso é mostrado nas seções sobre Transformada Discreta de Fourier e mel-espectrograma.

2.1.2 Transformada Discreta de Fourier

A Transformada Discreta de Fourier é a aplicação da Transformada de Fourier original em dados discretos, sendo um deles o áudio amostrado. Ela é utilizada para converter o áudio no domínio do tempo para o domínio da frequência, decompondo o sinal em todas as senoides que o compõe (SMITH, 1999). A Transformada Discreta de Fourier é definida pela Equação 3, em que $X(k)$ é o k-ésimo coeficiente de Fourier, $x(n)$ denota para a n-ésima amostra do sinal, j é $\sqrt{-1}$, e N equivale o total de amostras do sinal. O resultado obtido aplicando-a sobre uma onda quadrada de $100Hz$ é retratado pela Figura 3, porém o resultado dela é espelhado, assim é necessário exibir $X/2$ frequências, devido ao teorema de Nyquist.

$$X(k) \stackrel{\Delta}{=} \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \qquad k = 0,1,2,\ldots,N-1 \tag{3}$$

Figura 3 – Frequências que compõem a onda quadrada de $100Hz$



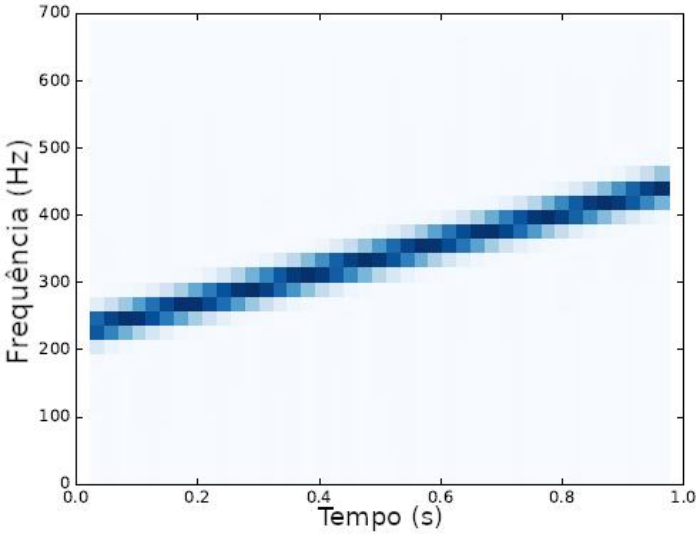
Fonte: Adaptado de Downey (2016)

Porém a Transformada Discreta de Fourier não é eficiente o suficiente para ser computada em tempo real, dessa forma, utiliza-se a Transformada Rápida de Fourier. Ela é a implementação otimizada da Transformada Discreta de Fourier, portanto ela irá produzir uma saída semelhante. A comparação de sua velocidade é de $N * \log(N)$ contra N^2 (ROCKMORE, 2000) da transformada discreta original.

2.1.3 Espectrograma

O espectrograma é uma maneira de representar o áudio. Essa representação é muito utilizada para processamento de áudio, análise de fala, e, até classificar comida a partir do sons produzidos durante a mastigação (KALANTARIAN *et al.*, 2015). Isso acontece porque com ele é possível inferir as frequências predominantes em um instante de tempo, pois, ele mostra o volume do áudio em função das suas frequências e tempo (WYSE, 2017), diferente da forma de onda que é apenas a amplitude no domínio do tempo. Porém, o espectrograma é tridimensional, dessa forma é necessário utilizar cores para representar a amplitude. A Figura 4 mostra um espectrograma de uma onda *chirp* (frequências que aumentam ou diminuem no decorrer do tempo), em que o azul representa as frequências presentes nessa onda e os tons são a amplitude de determinada frequência.

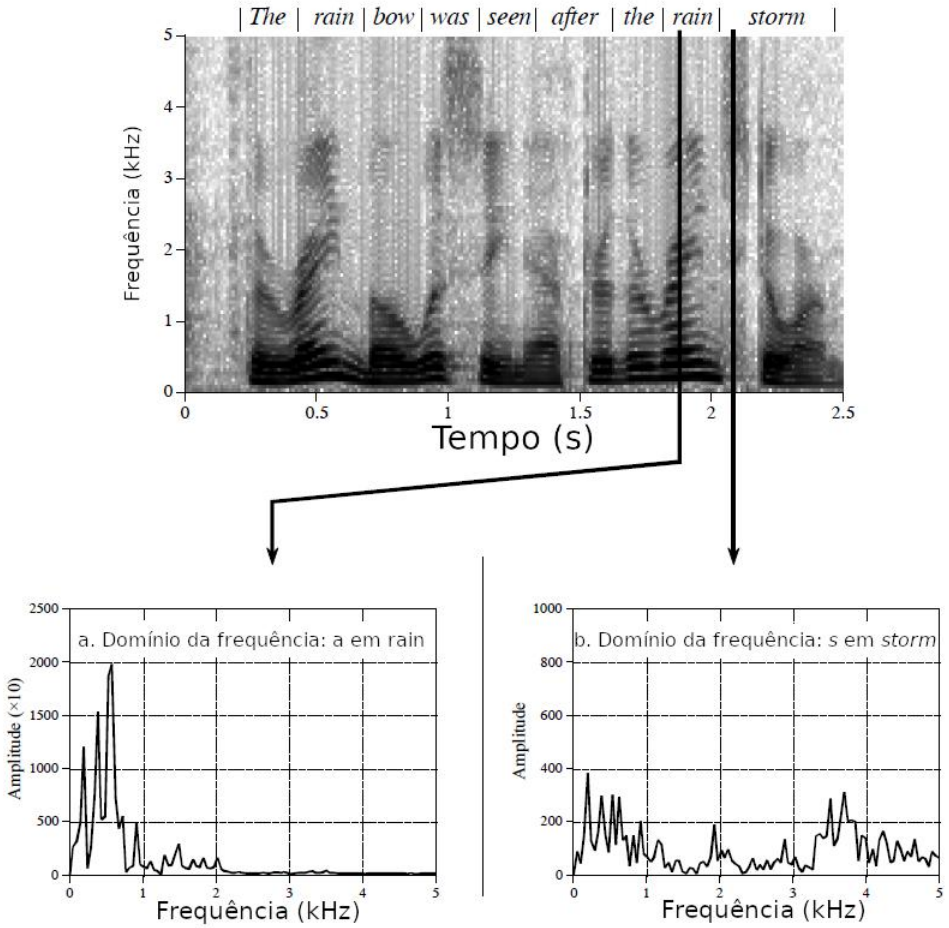
Figura 4 – Espectrograma de uma onda *chirp* de uma oitava de um segundo



Fonte: Adaptado de Downey (2016)

Essa representação também possibilita a identificação de padrões na voz, já que, permite identificar onde está cada palavra, ou sílaba, e sua frequência predominante (SMITH, 1999). Isso pode ser visto na Figura 5, em que o espectrograma é da gravação da frase "*The rainbow was seen after the rain storm*".

Figura 5 – Espectrograma de uma voz, mostrando o domínio da frequência dos fonemas /a/ (item (a)) e /s/ (item (b)) durante a fala das palavras "rain" e "storm", respectivamente



Fonte: Adaptado de Smith (1999)

2.1.4 Mel-espectrograma

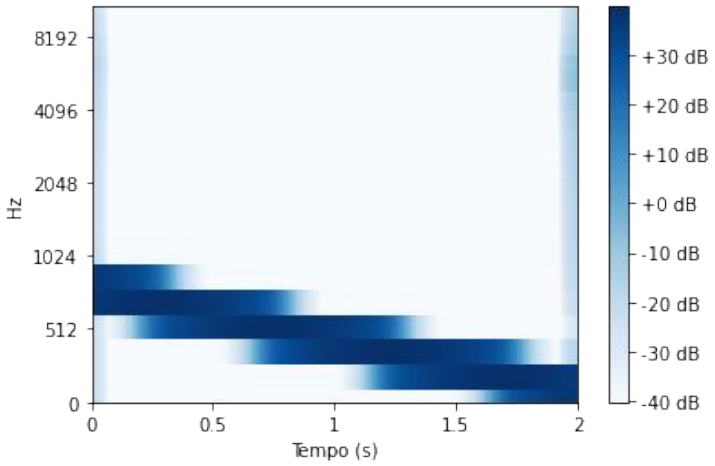
Outra maneira de representar o espectrograma é o mel-espectrograma, sua diferença é que no eixo *y* é mostrado a frequência na escala Mel (*m*) (GIBSON *et al.*, 2014). Ela é uma escala logarítmica, que tenta imitar como os humanos percebem o som, pois, a variação de frequências em frequências baixas são mais perceptíveis que em frequências altas.

Para obter-se o mel-espectrograma é necessário converter a menor e a maior frequências para escala Mel utilizando a Equação 4. Depois, escolhe-se a quantidade de bandas, costuma-se utilizar valores entre 20 e 40, para dividir igualmente o intervalo da escala. E por fim aplica-se um banco de filtros triangulares, sendo a quantidade total de filtros igual a de banda da escala, sobre o espectrograma convencional que resulta no mel-espectrograma. A Figura 6

apresenta o mel-espectrograma de uma chirp entre $880Hz$ e $220Hz$ com 20 intervalos de banda.

$$m = 2595\log_{10}(1+\frac{f}{700}) \tag{4}$$

Figura 6 – Exemplo de mel-espectrograma de uma chirp entre $880Hz$ e $220Hz$, com 20 intervalos de banda



Fonte: Autoria própria (2021)

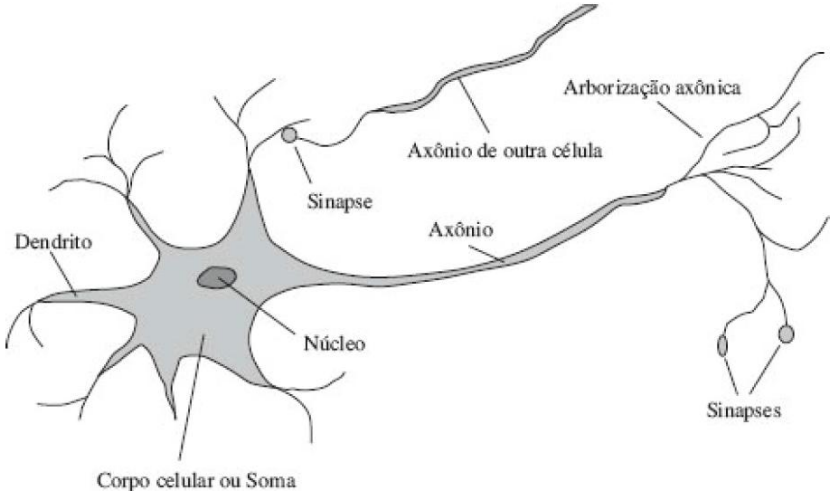
2.2 Redes neurais artificiais

Redes Neurais Artificias são algoritmos importantes de aprendizado de máquina. Elas são modelos computacionais inspirados no funcionamento do cérebro humano e seus neurônios. O Perceptron é um modelo de neurônio artificial que atua como unidade básica de uma ANN, ele é uma função matemática que representa um único neurônio biológico e seu funcionamento. Assim, as ANNs podem ser definidas como um conjunto de nós, neurônios artificiais, conectados (RUSSELL; NORVIG, 2013) que conseguem armazenar conhecimento experimental e disponibilizá-lo para uso (HAYKIN, 2010).

2.2.1 Neurônio biológico

O neurônio é a célula básica do cérebro, composto por um corpo e prolongamentos que se estendem a partir dele. O corpo, Soma, é responsável por processar as informações recebidas de outros neurônios. Os prolongamentos são divididos em dois: Dendritos e Axônios. O Dendrito é responsável por receber impulsos nervosos de outros neurônios, em contrapartida, o Axônio propaga os impulsos nervosos processados pelo Soma para outros neurônios (COPPIN, 2004). A Figura 7 representa um único neurônio.

Figura 7 – Neurônio Biológico



Fonte: (RUSSELL; NORVIG, 2013)

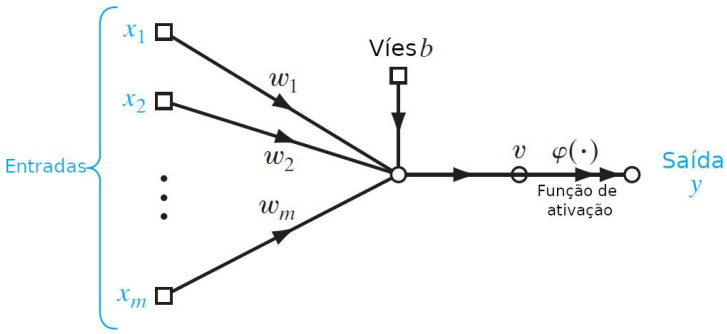
A informação flui através de um neurônio para outro, quando o primeiro está suficientemente excitado. Porém, entre eles não há uma ligação celular, no entanto, há uma fenda chamada sinapse. Ela é responsável pela transmissão do potencial de ação do Axônio para outro neurônio.

A plasticidade sináptica controla a eficácia da comunicação entre dois neurônios, assim, o cérebro consegue se adaptar a novas informações, já que, a força das sinapses é alterada pela plasticidade. Ela é dividida em plasticidade de curto prazo e longo prazo. Plasticidade de curto prazo refere-se a habilidade de fortalecer ou enfraquecer uma sinapse em um espaço de tempo pequeno. Enquanto que, a plasticidade de longo prazo fortalece a conexão por um período de tempo maior (HAYKIN, 2010).

2.2.2 Perceptron

O Perceptron foi desenvolvido por Rosenbaltt (1957), inspirado no neurônio artificial de McCulloch e Pitts (1943). Ele é a configuração mais simples de uma ANN, sendo chamado de rede Perceptron de uma única camada composta por apenas um nó. O Perceptron consiste em m entradas binárias x_i, pesos (w_i) correspondentes a cada entrada, e um viés (b) que irá indicar sua propensão de ativação. A Figura 8 representa uma rede Perceptron de uma única camada com uma função de ativação Φ, porém, a função de ativação proposta por Rosenbaltt (1957) é a função passo.

Figura 8 – Perceptron



Fonte: Adaptado de Haykin (2010)

O Perceptron realiza a soma do produto de todas as entradas pelos seus pesos e, após isso, é adicionado o viés, conforme a Equação 5. Essa equação pode ser simplificada para a Equação 6, ao transformar as entradas e os seus pesos em vetores, assim, é possível realizar o produto escalar entre eles. O resultado (v) é um valor real, contudo, ele não é a saída do Perceptron. A função de ativação, representada pela Equação 7, é a responsável por determinar a saída do Perceptron ao receber v, nesse caso sempre retornará 0 ou 1.

$$v = \sum_{i=1}^{m} w_i x_i + b \tag{5}$$

$$v = w \cdot x + b \tag{6}$$

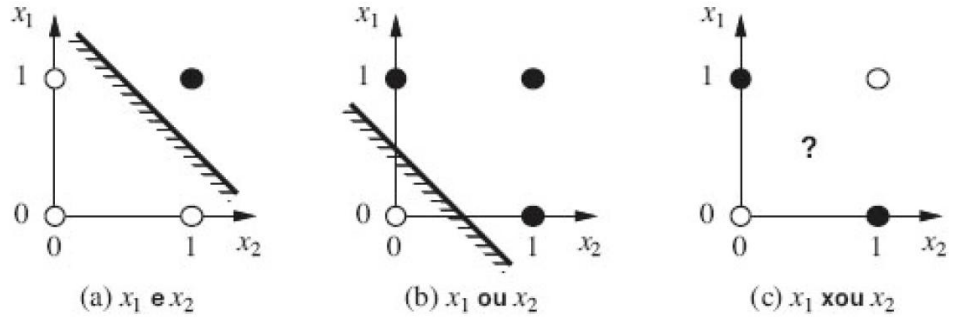
$$\text{saída} = \begin{cases} 1 & \text{se } w \cdot x + b > 0 \\ 0 & \text{se } w \cdot x + b \leq 0 \end{cases} \tag{7}$$

Assim o perceptron pode ser definido como um classificador binário capaz de

identificar classes linearmente separáveis (SILVA *et al.*, 2010). Isso implica que apenas uma pequena parcela dos problemas podem ser resolvidos com os Perceptrons clássicos (RUSSELL; NORVIG, 2013), logo, é necessário utilizar outros modelos de redes neurais artificias que consigam resolver problemas não linearmente separáveis, como o problema do ou exclusivo (XOU).

O XOU, é uma função lógica que recebe duas entradas binárias, e retorna 1 apenas quando as suas entradas são diferentes. Funções linearmente separáveis, conseguem ser desenhadas em um gráfico bidimensional e ter uma linha traçada entre suas classes. A Figura 9 apresenta três funções lógicas, E, OU, XOU, em que as saídas 1 são representadas por um circulo preto e 0 por um circulo branco para duas entradas apresentadas nos eixos $x1$ e $x2$. Assim, é possível observar que as funções E e OU têm suas classes separadas por uma reta e a XOU não. As duas primeiras são linearmente separáveis, já a última não é linearmente separável. Assim, ela não consegue ser processada por um Perceptron clássico, pois, não é possível traçar uma única reta que separe suas saídas em um gráfico bidimensional (COPPIN, 2004).

Figura 9 – Comparação entre as funções lógicas E, OU e XOU



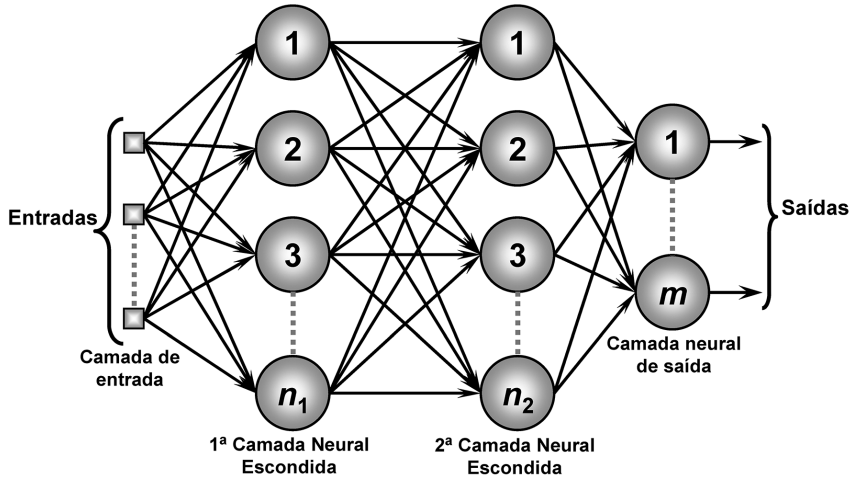
Fonte: Russell e Norvig (2013)

2.2.3 Perceptron multicamadas

As redes multicamadas conseguem resolver problemas não linearmente separáveis como o XOU. As Redes Perceptron de Múltiplas Camadas (MLP – MultiLayer Perceptron), é composta por diversos Perceptrons dispostos em três tipos de camadas: uma camada de entrada, uma camada de saída e uma ou mais camadas ocultas. Além disso, é necessário que

cada neurônio utilize uma função de ativação que seja diferenciável (HAYKIN, 2010). A forma mais simples de uma MLP é a rede Perceptron multicamadas *feedforward* e completamente conectada. A Figura 10 representa uma rede MLP *feedforward* e completamente conectada com duas camadas ocultas.

Figura 10 – Representação de uma MLP



Fonte: Adaptado de Silva *et al.* (2010)

Modelos *feedforward* representam uma ANN que o sinal de entrada é propagado da esquerda para direita, da camada de entrada para a camada de saída, passando por todas as camadas e sem criar ciclos. Uma ANN é completamente conectada quando cada neurônio está conectado, por um peso sináptico, com todos os neurônios da sua camada anterior (COPPIN, 2004).

O treinamento de uma rede MLP é feito ao ajustar os pesos e o viés de cada neurônio da ANN utilizando o processo de treinamento supervisionado (SILVA *et al.*, 2010). Assim, é necessário um algoritmo que consiga realizar esse treinamento. Para isso, é utilizado o algorítimo *backpropagation*. Esse algorítimo é dividido em duas fases: *forward* e *backward*. A fase *forward*, propaga os sinais da camada de entrada para as camadas posteriores até a camada de saída da ANN, assim, produzindo uma resposta finalizando a fase *forward*. Desse modo, inicia-se a fase *backward*, que calcula o erro de cada neurônio em relação a saída desejada, depois, atualiza todos os pesos e vieses dos neurônios da ANN baseado nos erros dos neurônios da camada de saída. Esse processo é repetido até o erro estar pequeno (SILVA *et al.*, 2010).

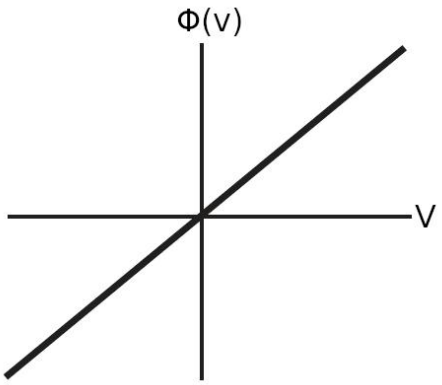
2.2.4 Função de ativação

A função de ativação determina a saída de um Perceptron com base no seu potencial de ativação. Ela é responsável por introduzir a não linearidade na saída de uma ANN (RUSSELL; NORVIG, 2013). Isso é importante porque a maioria dos problemas do mundo real não são lineares. E também, se uma MLP utilizar apenas funções de ativações lineares, ela terá poder computacional equivalente a uma rede de uma única camada (AGGARWAL, 2018). Nesse trabalho serão discutidas cinco funções de ativação, sendo elas: Linear, Sigmoide, Rectified Linear Unit (ReLU), Mish e Softmax.

A função de ativação Linear, representada pela Equação 8, é o tipo mais básico de função de ativação, pois, ao receber o valor v apenas o propaga, dessa forma gerando uma reta como apresenta a Figura 11. Ela é tipicamente utilizada na camada de saída de redes para problemas de regressão. Utilizá-la equivale a simplesmente usar a soma ponderada das entradas e do viés como o nível de ativação do neurônio (COPPIN, 2004).

$$\Phi(v) = v \tag{8}$$

Figura 11 – Gráfico da função Linear



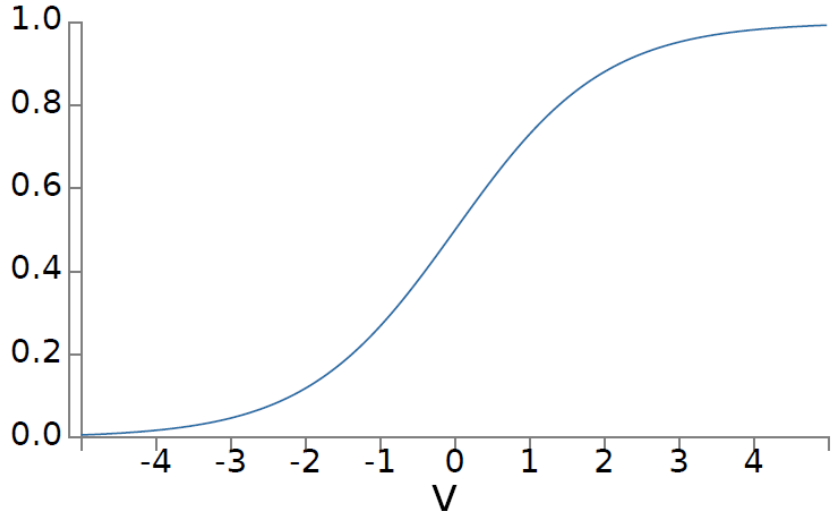
Fonte: Adaptado de Coppin (2004)

A função Sigmoide, representada pela Equação 9, já foi o tipo de função mais comum em ANNs (HAYKIN, 2010). Ela recebe o potencial de ativação do Perceptron e retorna um valor no intervalo de 0 a 1. A Figura 12 apresenta o gráfico da função Sigmoide, assim, é possível visualizar que ela é definida como uma função contínua, portanto, diferenciável em todos os pontos. Essa função torna-se útil para criar modelos que necessitam que suas saídas sejam valores probabilísticos, já que, sua saída pertence a um intervalo entre 0 a 1

(AGGARWAL, 2018). Ela é tipicamente utilizada em problemas da classificação binária na última camada da ANN. Essa função possui algumas desvantagens como a sua saída não ser centralizada em zero e estar mais sujeita a dissipação do gradiente (NWANKPA *et al.*, 2018). A dissipação do gradiente torna os gradientes da ANN muito pequenos, dessa forma, dificulta o seu treino.

$$\Phi(v) = \frac{1}{1+e^{-v}} \tag{9}$$

Figura 12 – Gráfico da função Sigmoide



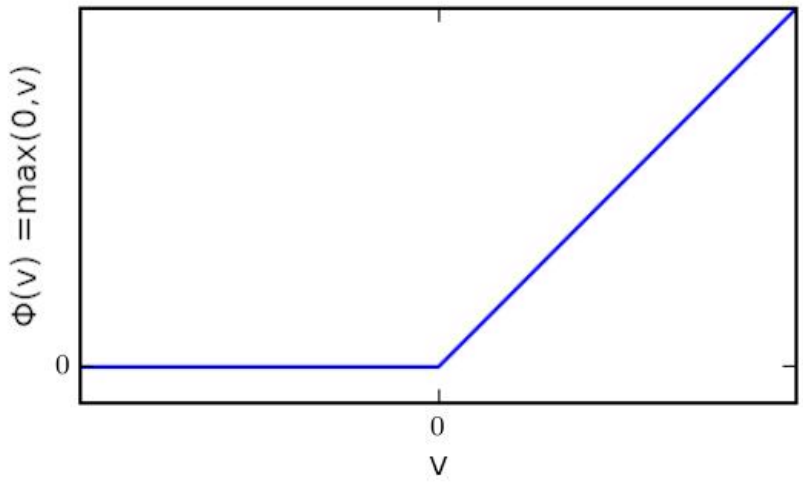
Fonte: Adaptado de Nielsen (2015)

A função ReLU, é uma função de ativação popular tipicamente utilizada em camadas convolucionais. Ela é representada pela Equação 10 e seu gráfico é definido pela Figura 13. Ela retorna 0 se receber qualquer entrada negativa, porém, para qualquer valor positivo v, ela retornará o próprio valor. Essa função resolve o problema da dissipação do gradiente porém trás um problema chamado "Dying ReLU" (PEDAMONTI, 2018), que, quando seu gradiente é zero, o nó deixa de influenciar a rede, assim diminuindo seu aprendizado.

$$\Phi(v) = \max(0, v) \tag{10}$$

A função Mish, está se popularizando como uma substituta da ReLU. Isso, deve-se ao fato dessa função de ativação demonstrar mais robustez durante o treinamento de ANNs (MISRA, 2019). Também, ela possui a característica de não ser monotônica o que permite preservar valores negativos, portanto, estabilizar o gradiente da ANN. A Mish é representada pela Equação 10 e seu gráfico é definido pela Figura 14.

Figura 13 – Gráfico da função ReLU



Fonte: Adaptado de Goodfellow *et al.* (2016)

$$\Phi(v) = \max(0, v) \tag{11}$$

A função Softmax, expressada pela Equação 12, é tipicamente utilizada para tarefas de classificação multi-classe (NWANKPA *et al.*, 2018). Essa função gera um vetor que representa as distribuições de probabilidade de uma lista de possíveis resultados, e o valor resultante da soma do vetor é 1. Assim, o resultado final da ANN é obtido ao selecionar a maior probabilidade do vetor gerado.

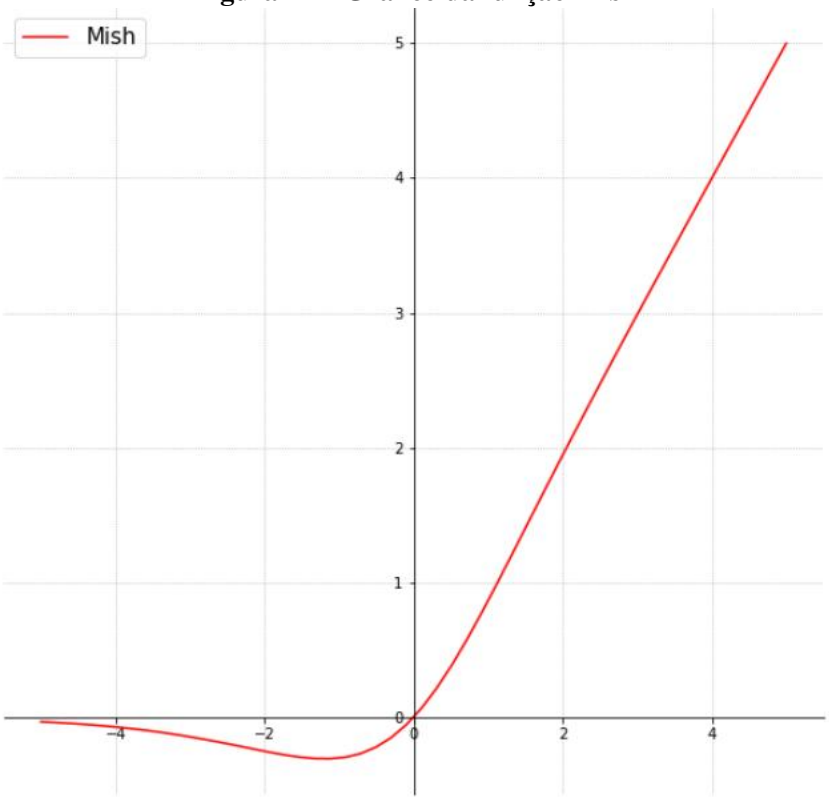
$$\Phi(v_i) = \frac{e^{v_i}}{\sum e^{v_i}} \tag{12}$$

2.2.5 Backpropagation

O algoritmo Backpropagation, como dito antes, é utilizado para treinar MLPs. Considerado como o principal componente no aprendizado de ANNs de acordo com Nielsen (2015). Ele busca minimizar o custo da ANN ajustando os pesos de cada neurônio. Isso é obtido ao executar duas etapas: *forward* e *backward*.

Primeira etapa, *forward*, recebe uma instância do conjunto de treinamento e a propaga

Figura 14 – Gráfico da função Mish



Fonte: Adaptado de Misra (2019)

para as camadas posteriores, sendo processada por cada neurônio, até a camada de saída gerando uma resposta. A aplicação desta fase visa apenas a obtenção de uma resposta sem realizar nenhuma alteração nos pesos e vieses atuais dos neurônios (SILVA *et al.*, 2010). Depois, a etapa *backward* inicia-se calculando a diferença entre a saída obtida com a desejada, obtendo-se o erro, que também pode ser chamado de custo. Assim, atualizam-se todos os pesos e vieses de cada neurônio a partir da camada de saída para as camadas ocultas anteriores baseado nos erros dos neurônios da camada de saída.

A atualização dos pesos (w_{ij}) e vieses (b_j), acontece com a utilização da técnica do Gradiente Descendente, que aplica a descida do gradiente para minimizar o custo da ANN. Dessa forma, a correção ocorre ao subtrair o valor atual de w_{ij} pela derivada parcial do custo multiplicada pela taxa de aprendizado (η), representado pela Equação 13, a qual, aplica-se também para o viés conforme a Equação 14, sendo elas adequadas para qualquer local da ANN. Em seguida, realiza-se a derivada parcial do custo em relação ao w_{ij} e em relação ao b_j, utilizando a regra da cadeia, já que eles estão compostos na função de custo, conforme a Equação 15 e a Equação 16 foram adaptadas de Haykin (2010), entretanto, essas duas equações só valem para os pesos e vieses conectados a camada de saída. No qual y é a saída do nó, C é o

custo da ANN e v é o potencial de ativação.

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial C}{\partial w_{ij}} \tag{13}$$

$$b_j \leftarrow b_j - \eta \frac{\partial C}{\partial b_j} \tag{14}$$

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial C}{\partial y_j} \frac{\partial y_j}{\partial v_j} \frac{\partial v_j}{\partial w_{ij}} \tag{15}$$

$$\frac{\partial C}{\partial b_j} = \frac{\partial C}{\partial y_j} \frac{\partial y_j}{\partial v_j} \frac{\partial v_j}{\partial b_j} \tag{16}$$

É possível adaptar essas equações fazendo a derivada parcial do custo em relação ao potencial de ativação do nó, sendo definido como δ conforme a Equação 17. Também o resultado de $\frac{\delta v_j}{\delta w_{ij}}$ é a entrada y_i do neurônio atual, e $\frac{\delta v_j}{\delta b_j}$ é 1. Contudo, essa simplificação só abrange os pesos e vieses que estão conectados na camada de saída, pois a atualização dos nós das camadas ocultas depende do resultado dos nós posteriores. Para nós ocultos, δ passa ser representado pela Equação 18 em que k é a camada posterior em relação a camada atual j.

$$\delta_j = \frac{\partial C}{\partial y_j} \Phi_j^{'}(v_j) \tag{17}$$

$$\delta_j = \Phi_j^{'}(v_j) \sum_k \delta_k w_{kj} \tag{18}$$

Assim a Equação 13 e Equação 14 passam a ser representadas, depois de serem simplificadas, pela Equação 19 e Equação 20. O processo de simplificação das equações é detalhado em Haykin (2010).

$$w_{ij} \leftarrow w_{ij} - \eta \delta_j y_i \tag{19}$$

$$b_j \leftarrow b_j - \eta \delta_j \tag{20}$$

2.2.6 Rede neural convolucional

Rede neural convolucional (CNN), é uma classe de ANN *feedforward*, com neurônios

localmente conectados, otimizada para entradas em grade (AGGARWAL, 2018). A maior parte de CNNs é destinada para o processamento de imagens, porém, elas podem ser utilizadas para o processamento de todos os tipos de dados temporais e espaciais (GOODFELLOW *et al.*, 2016). Dessa forma, é possível utilizar CNNs para o processar áudios. Essa ANN é dividida em três camadas, que podem ser replicadas para formar uma rede mais profunda, sendo elas: convolucional, ativação e *pooling*. A camada de ativação não será discutida, já que, é apenas a aplicação de uma função de ativação sobre os valores da estrutura em grade.

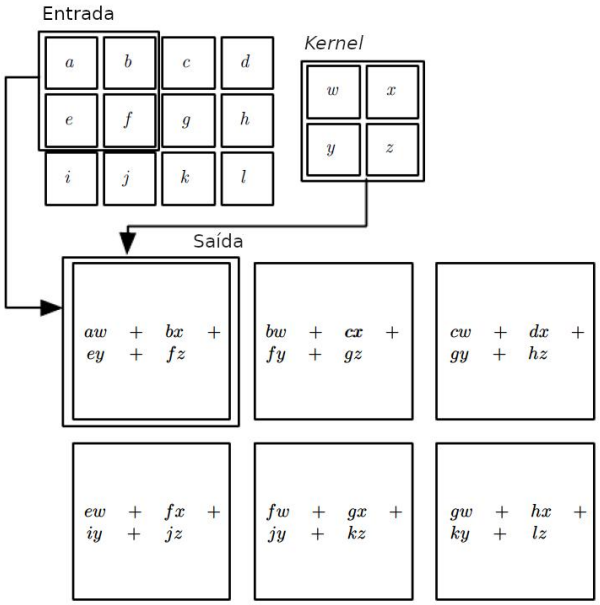
Camada de convolução é responsável por extrair características de sua entrada. Ela é uma estrutura em grade tridimensional ($L_e \times B_e \times d$), no qual L_e representa a altura da entrada, B_e a largura e d a profundidade. No contexto de imagens, a profundidade é utilizada como a quantidade de canais de cor que a imagem possui (ZHANG *et al.*, 2021).

O processo de extração de características depende de três hiperparâmetros: *kernel*, *padding* e *stride*. Assim realiza-se o produto escalar entre uma porção da entrada e o *kernel*, produzindo um mapa de características após ele convolver sobre toda a imagem conforme mostra a Figura 15. *Kernels* são estruturas tridimensionais ($L_k \times B_k \times d$), compostas por parâmetros treináveis (ERTEL, 2017) menores que a entrada e geralmente de dimensões ímpares (AGGARWAL, 2018). L_k representa a altura do *kernel*, B_k a largura e a profundidade é igual a da entrada que está sendo processada.

Para convolver os *kernels* pela entrada é utilizado o hiperparâmetro *stride* (S), ele é responsável em definir quantas colunas serão deslocadas a cada produto escalar. É comum o uso do valor 1 para esse hiperparâmetro, sendo implícito nas definições da camada. O mapa de características resultante da convolução da entrada pelo *kernel*, possui dimensões de acordo com as Equações $L_f = ((L_e - L_k)/S + 1)$ e $B_f = ((B_e - B_k)/S + 1)$, dessa forma, ao utilizar uma imagem em tons de cinza $28 \times 28 \times 1$ e um *kernel* $5 \times 5 \times 1$ obtém-se um mapa de características $24 \times 24 \times 1$.

Percebe-se que ao realizar a convolução a entrada perde informações nas bordas pelo fato de ter suas dimensões reduzidas, isso pode influenciar na tomada de decisão da CNN. Desse modo, utiliza-se o *padding* (P) que preenche a borda com valores para evitar a perca de informação excessiva nas bordas da entrada da camada. Comumente utiliza-se o *zero-padding*, que preenche a borda com zeros, para o mapa de características manter as mesmas dimensões da entrada (QUINTANILHA, 2017). A dimensão do mapa de características gerado, ao combinar os três hiperparâmetros, pode ser calculada utilizando a Equação 21 e a Equação 22. Ao aplicar os mesmos valores utilizados anteriormente, porém, com a adição de um *zero-padding* igual a 2, obtém-se um mapa de características de dimensões $28 \times 28 \times 1$ sendo essa dimensão igual a

Figura 15 – Exemplo de convolução de uma entrada 3x4 e um *kernel* 2x2, resultando em um mapa características 2x3



Fonte: Adaptado de Goodfellow *et al.* (2016)

entrada.

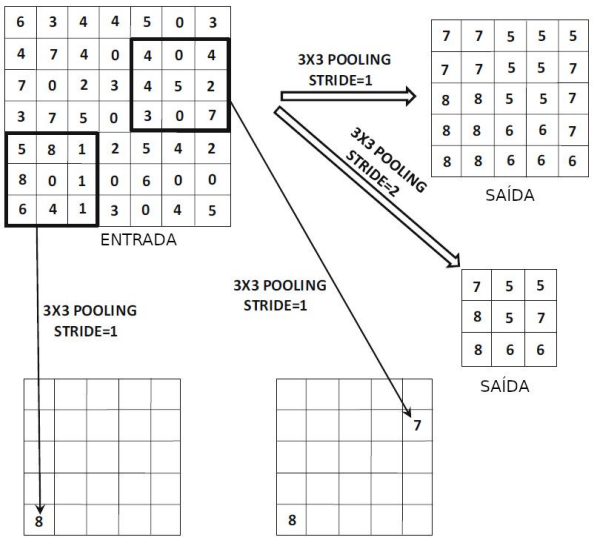
$$L_f = \left(\frac{L_e + 2P - L_k}{S}\right) + 1 \tag{21}$$

$$B_f = \left(\frac{B_e + 2P - B_k}{S}\right) + 1 \tag{22}$$

Camada de *pooling* reduz a dimensionalidade de cada mapa de características obtido após a camada de convolução, isso diminui o risco da ANN memorizar o conjunto de dados (*overfitting*) e o custo computacional da rede (GÉRON, 2019), pois, ela não utiliza parâmetros. É necessário definir um *stride* e um tamanho de *kernel* para essa camada. Ela aplica o *kernel* sobre a imagem, convolvendo-o a cada operação, que retorna um valor condensado.

A Figura 16 mostra a aplicação de duas configurações de *max-pooling*, um método comum de *pooling* (NIELSEN, 2015), sobre um mapa de características 7×7 tendo como resultado uma saída, com dimensões menores que a entrada, respectiva para cada configuração. Esse método retorna o valor máximo presente em um *kernel*, sendo esse o motivo pelo qual ele recebe a palavra “max” em seu nome. Para descobrir as novas dimensões do mapa de características utiliza-se as $L_f = ((L_e - L_k)/S + 1)$ e $B_f = ((B_e - B_k)/S + 1)$ similar ao processo

Figura 16 – Exemplo de pooling com uma entrada 7×7 e aplicação do *max-pooling* com diferentes configurações



Fonte: Adaptado de Aggarwal (2018)

de convolução sobre um *kernel*.

2.2.7 Gradient-weight Class Activation Mapping

ANNs são utilizadas para resolver diversos problemas complexos, porém, para áreas do conhecimento mais críticas, como medicina, é fundamental explicar o resultado gerado pela ANN (VELLIDO, 2020). Por isso, surgiu a necessidade de interpretar modelos de *machine learning*, porém, muitas das maneiras pensadas surgem ao custo da precisão do modelo (SELVARAJU *et al.*, 2016b; ZHOU *et al.*, 2015).

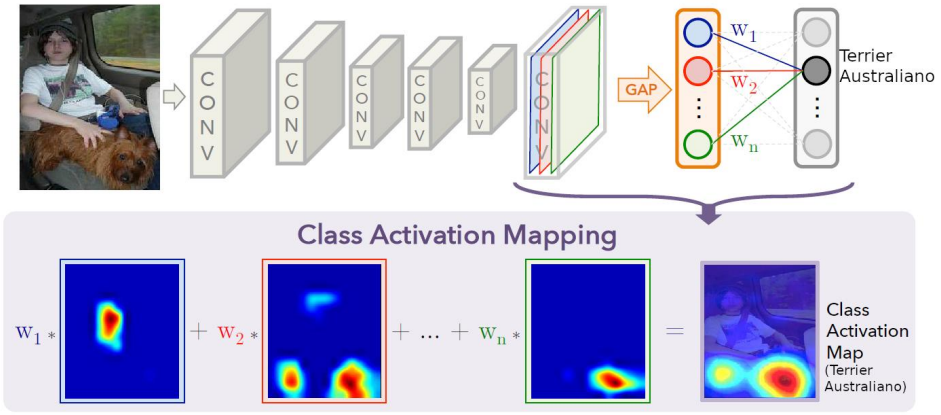
A fim de conseguir explicar os resultados de CNNs foi proposto por Zhou *et al.* (2015) a técnica chamada *Class Activation Mapping* (CAM), que é capaz de localizar regiões mais importantes em imagens para a classificação resultante do modelo. Entretanto, é necessário alterar o modelo e retreiná-lo para essa técnica. Pois, é fundamental que após a última camada convolucional tenha uma camada de *Global Average Pooling* (GAP) seguida por uma camada densa conectada a saída. Essa alteração afeta como é calculada a pontuação (y) de determinada classe (c) antes da aplicação da função de ativação da saída, o novo cálculo é representado pela Equação 23. Isso permite que os pesos (w_k) da camada densa em relação a uma classe c sejam projetados sobre os mapas de características (A^k) da última camada convolucional. O que só

é possível pelo fato da ANN utilizar um GAP conectado a ultima camada densa, pois, isso cria uma estrutura de conexão simples. Logo, é gerado k projeções de importância da saída que podem ser somadas para gerar um mapa de calor de dimensão igual aos mapas de características da última camada convolucional, conforme a Equação 24 e apresentado pela Figura 17.

$$y^c = \sum_k w_k^c \overbrace{\frac{1}{Z}\sum_i \sum_j A_{ij}^k}^{\text{Global Avarage Pooling sobre } A^k} \tag{23}$$

$$CAM^c = \sum_k w_k^c A^k \tag{24}$$

Figura 17 – Estrutura do processo do CAM



Fonte: Adaptado de Zhou *et al.* (2015)

O CAM traz consigo o problema da troca de precisão por interpretabilidade, já que é necessário alterar a ANN e retreinar-lá. Por essa razão, foi criada a técnica *Gradient-weight Class Activation Mapping* (Grad-CAM) que consegue tornar o modelo interpretável sem a necessidade de retreinar ou modificar a CNN.

O Grad-CAM, diferente de seu antecessor, pode ser aplicado sobre uma grande quantidade de arquiteturas de CNN, pois, não é necessário realizar nenhuma alteração sobre a ANN original. A principal diferença entre esses algoritmos está no valor que é projetado sobre os mapas de características. No Grad-CAM os valores que são multiplicados por A^k deixam de ser os pesos da camada densa e tornam-se os pesos de importância (a) calculados com ajuda do Backpropagation. Esses a são obtidos ao calcular o gradiente da pontuação (y) para classe c em respeito a A^k, e depois, aplicar o GAP sobre esses valores transformando-os em um único

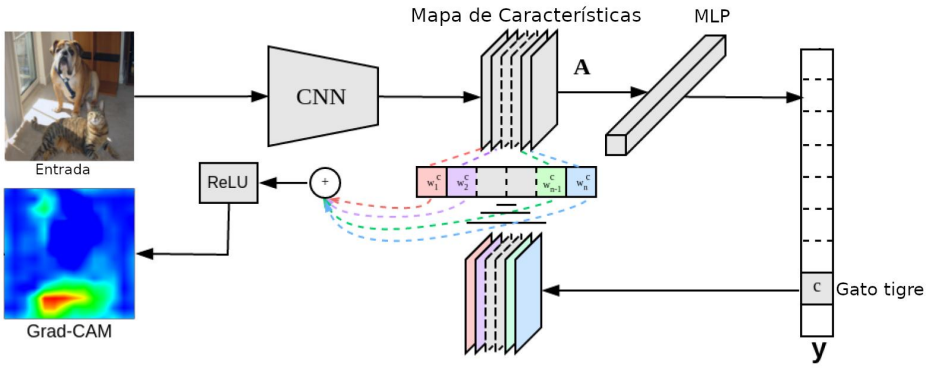
número, conforme a Equação 25.

$$a_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{GAP}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{Gradientes}} \tag{25}$$

E por fim, para obter o mapa de calor em relação a classe $Grad-CAM^c$ é realizada a soma entre todas as multiplicações de cada a_k^c com o mapa de características A^k. Também, a fim de otimizar o resultado e exibir apenas características que influenciaram positivamente o resultado é aplicada a função de ativação ReLU, pois, ela transformará qualquer valor negativo em zero, demonstrado na Equação 26. O processo da rede pode ser visualizado na Figura 18 e a combinação entre o Grad-CAM com a imagem pode ser vista na Figura 19.

$$\text{Grad-CAM}^c = ReLU(\sum_k a_k^c A^k) \tag{26}$$

Figura 18 – Estrutura do processo do Grad-CAM

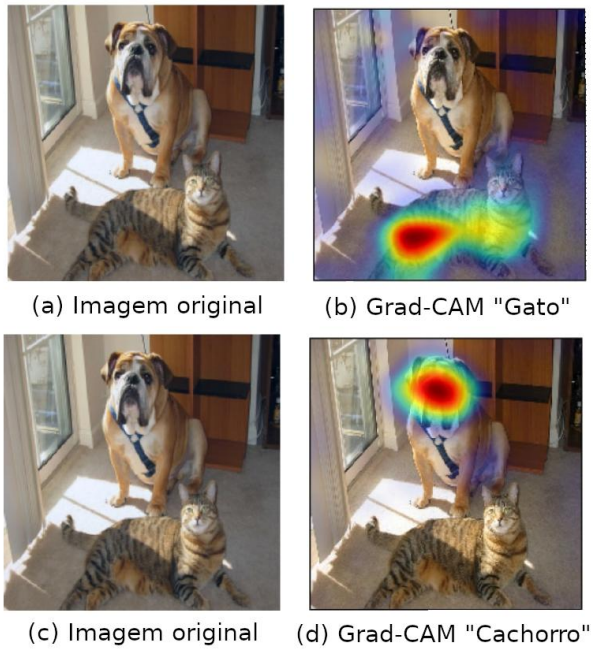


Fonte: Adaptado de Selvaraju *et al.* (2016a)

2.3 Processamento de Áudio utilizando Redes Neurais Artificiais

A literatura sobre ANNs tem em sua maioria o foco em visão computacional e processamento de linguagem natural, porém, o processamento de áudio com ANNs tem crescido com o passar do tempo. Isso decorre do fato de que muitas vezes o processamento de

Figura 19 – Figura original(a,c). Figura original combinado com saída do Grad-CAM ao selecionar classe gato (b). Figura original combinado com saída do Grad-CAM ao selecionar classe cachorro (d)



Fonte: Adaptado de Selvaraju *et al.* (2016b)

áudio depende de conhecimentos avançados na área para solucionar os problemas e ajustar os modelos. Porém, é possível processar áudios com CNNs ao transformá-los em espectrogramas e suas variantes (MACCAGNO *et al.*, 2021), assim, consegue-se mais facilidade e a possibilidade de aplicar diversas técnicas sobre os áudios como transferência de aprendizado e aumento de dados.

2.3.1 Transferência de Aprendizado com Pretrained Audio Neural Networks

É fato que o treino de ANNs profundas é custoso, bem como, o aumento da complexidade dos problemas que elas devem resolver. Dessa forma, é necessário encontrar meios de otimizar o aprendizado desses modelos complexos, logo, criou-se técnicas para transferir conhecimento entre modelos (TORREY; SHAVLIK, 2010). Elas consistem em utilizar o conhecimento aprendido por um modelo sobre um conjunto de dados e passá-lo para outra ANN a fim de facilitar seu treino, já que, isso reduz a quantidade de dados necessários

(ZHUANG *et al.*, 2020). Assim, o novo modelo irá utilizar as características aprendidas pela primeira ANN.

A transferência de aprendizado provou-se essencial para as áreas de visão computacional (GOPALAKRISHNAN *et al.*, 2017) e processamento de língua natural (RUDER *et al.*, 2019). Por isso, foi proposto por Kong *et al.* (2020) uma técnica para processamento de áudio. Ela é chamada de *Pretrained Audio Neural Networks* (PANN) que são modelos que foram treinados no conjunto de dados AudioSet, que possui cerca de 1,9 milhão de áudios totalizando mais de 5000 horas e 527 classes.

Foi provado que PANNs podem ser utilizadas como uma técnica de transferência de aprendizado para uma grande variedade de tarefas. Também podem ser utilizadas para fazer o ajuste fino de modelos que não possuem uma grande quantidade de áudios (KONG *et al.*, 2020). Isso pode ser de grande importância para processar áudios no português brasileiro, já que, há poucos conjuntos de dados abertos disponibilizados.

2.3.2 Aumento de Dados

Aumento de dados é uma técnica muito utilizada para treino de ANNs, principalmente quando não há dados suficientes para conseguir um resultado ideal. Isso ocorre porque técnica consegue gerar novas instâncias de treinamento sem a necessidade de novas coletas (PEREZ; WANG, 2017). Para cada área de estudo existem técnicas de aumento de dados específicas, e, no processamento de dados duas mostraram-se interessantes: Mixup e SpecAugment.

Mixup consiste em fazer uma combinação entre duas instâncias aleatórias (x_i e x_j) do conjunto de treino e suas classes (y_i e y_j) para gerar uma nova instância ($\tilde{x}, \tilde{y}$) que é gerada a partir das Equações 27 e 28, em que $\lambda \in [0,1]$ é gerado a partir da Distribuição Beta (WEISSTEIN, 2003). Em virtude disso, ela se torna uma técnica genérica para qualquer tarefa.

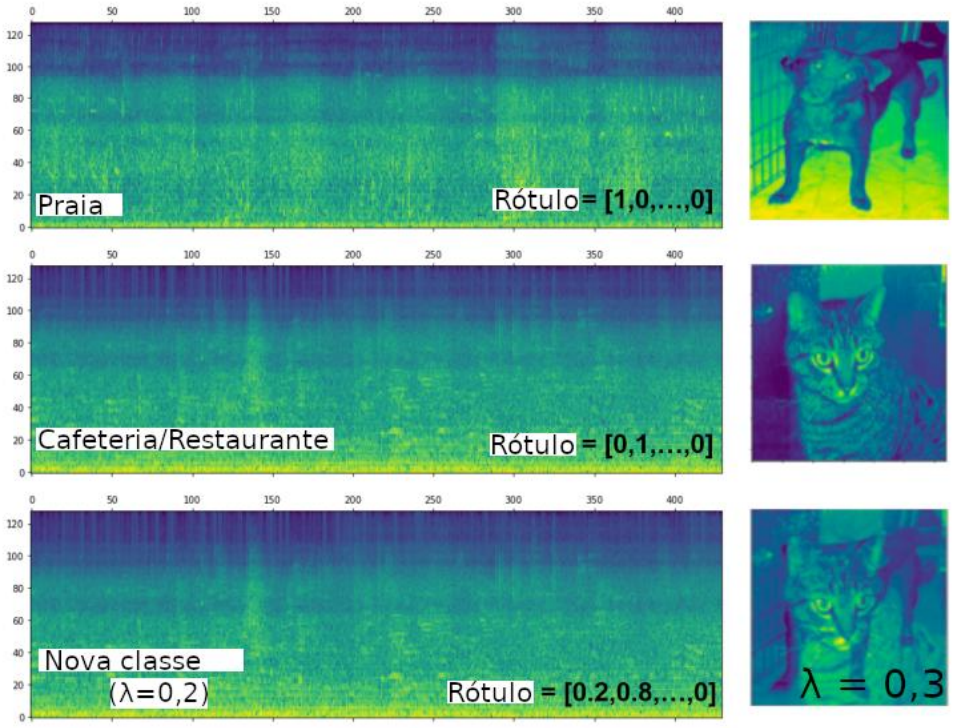
$$\tilde{x} = \lambda x_i + (1-\lambda)x_j \tag{27}$$

$$\tilde{y} = \lambda y_i + (1-\lambda)y_j \tag{28}$$

Essa técnica foi proposta por Zhang *et al.* (2017) para ser uma nova técnica de aumento de dados, pois, diferente de técnicas como rotação, recorte e inversão horizontal que são técnicas mais comuns em imagens. Essa técnica pode ser utilizada em diversas tarefas, inclusive,

processamento de áudios (XU *et al.*, 2018) como demonstra a Figura 20, já que ela é uma técnica genérica. Além disso, os autores provaram que a Mixup consegue melhorar a generalização das ANNS.

Figura 20 – Exemplo de aumento de dados com Mixup em espectrogramas (esquerda) e imagens de animais (direita)

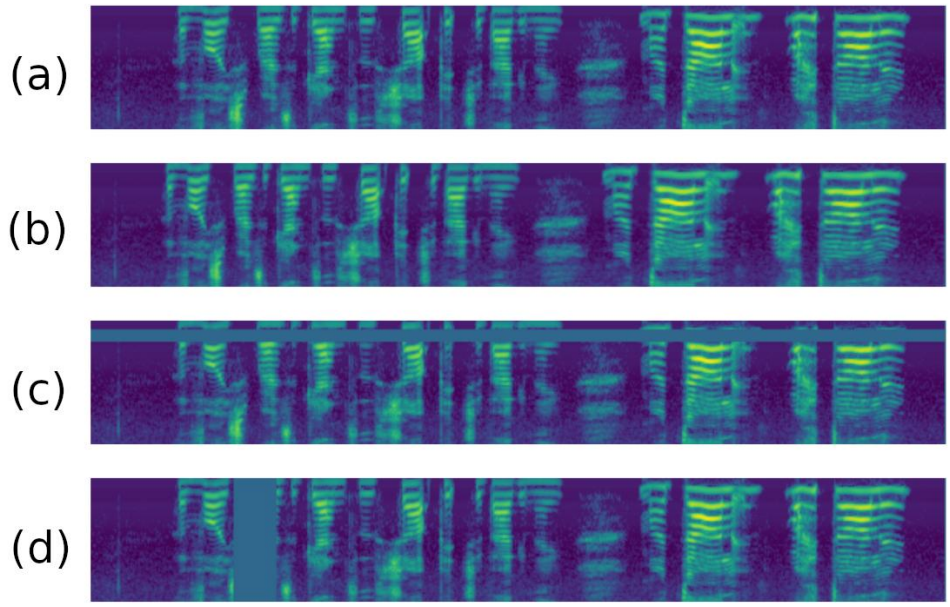


Fonte: Adaptado de Xu *et al.* (2018) e Kulakov (2020)

Diferente do Mixup, o SpecAugment, proposto por Park *et al.* (2019), é focado no aumento de dados em espectrogramas,pois, essa técnica foi pensada para tarefas de Reconhecimento automático de voz, que, é uma tarefa que se beneficia com o aumento da quantidade de dados para seu treinamento. Essa técnica realiza o aumento de dados sobre o espectrograma ao distorcê-lo na dimensão do tempo, mascarando partes dos canais de frequência consecutivos e mascarando blocos no tempo. Isso pode ser visualizado na Figura 21.

A máscara de frequência é feita sobre f canais de mel consecutivos $[f_0, f_0 + f)$, em que f é escolhido a partir de uma distribuição uniforme de 0 a F (que é um parâmetro da máscara), enquanto, f_0 é obtido a partir de $[0, v - f)$ no qual v é o número de canais de frequência de mel. Já a máscara temporal é realizada sobre t espaços de tempo $[t_0, t_0 + t)$, essa máscara segue os mesmos cálculos da máscara anterior, apenas alterando as variáveis.

Figura 21 – (a) Espectrograma original. (b) Espectrograma com distorção temporal. (c) Espectrograma com máscara de frequência. (d) Espectrograma com máscara de tempo



Fonte: Adaptado de Park *et al.* (2019)

2.3.3 Estado da arte

Casanova *et al.* (2021) propuseram a hipótese de que ANNs são capazes de detectar o sintoma de insuficiência respiratória causada pela COVID-19. Esse sintoma é um fator essencial para a hospitalização do paciente, pois, caso ele esteja presente o paciente será internado.

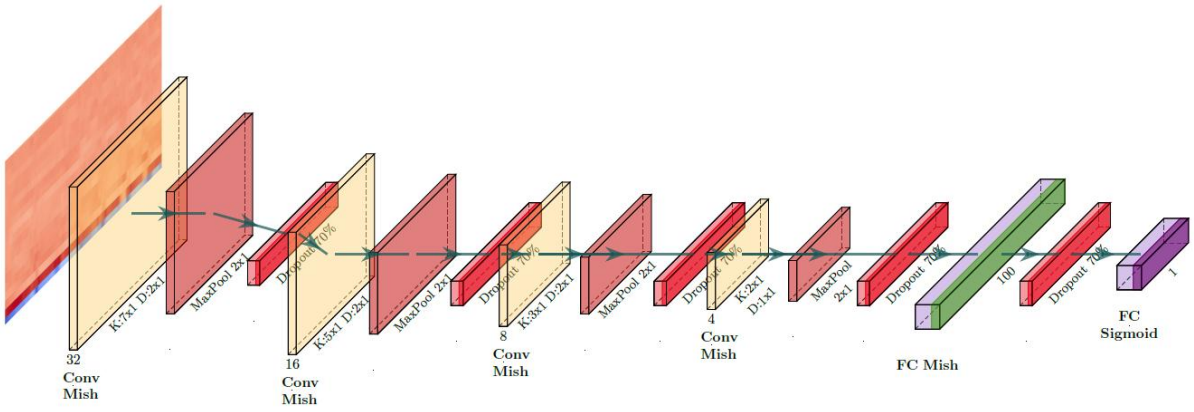
A fim de treinar uma ANN foi criado um conjunto de dados separado em pacientes e controles. Pacientes eram pessoas que estavam internadas com COVID-19 e que apresentavam uma oxigenação sanguínea abaixo de 92% (insuficiência respiratória). Já os controles, foram pessoas saudáveis que doaram suas vozes através da plataforma do projeto SPIRA. Os dois grupos foram requisitados a falar três frases diferentes, contudo, apenas uma foi escolhida para compor o conjunto de dados [1]. Esse conjunto de dados apresentou uma grande quantidade de ruídos, pois, não foram gravados em estúdio mas sim em hospitais (pacientes) e por pessoas sem equipamento de estúdio (controle). Isso provou-se um desafio que foi atacado com a utilização da adição de ruídos coletados, antes de cada sessão de gravação, nos hospitais para evitar o vieses na ANN.

Nesse trabalho foi proposto uma CNN que recebia os áudios do conjunto de dados na representação de Cepstrum de Frequência de Mel, já que, foi a abordagem que se mostrou mais

[1] "O amor ao próximo ajuda a enfrentar o coronavírus com a força que a gente precisa."

adequada nos testes preliminares do trabalho. Outra característica importante dessa ANN é que foi utilizada uma grande quantidade de regularização para evitar o sobreajuste, sendo elas a utilização da função de ativação Mish, *dropout*, decaimento de peso de $0,01$ e normalização de grupo (ZHOU *et al.*, 2015). A ANN final é mostrada pela Figura 22 em que K é tamanho do filtro, D é o tamanho da dilatação e a camada densa é FC.

Figura 22 – ANN SpiraNet



Fonte: Casanova *et al.* (2021)

O resultado final foi obtido a partir de diversos experimentos, sendo o melhor o que obteve uma precisão de $91,66\%$. Isso provou a hipótese inicial e também que a voz humana pode ser utilizada como um biomarcador para detecção de COVID-19. E também, outra contribuição importante foi a criação de seu conjunto de dados, já que ele foi disponibilizado para a comunidade.

3 MATERIAIS E MÉTODOS

Neste capítulo serão descritos os materiais e métodos aplicados para atingir os objetivos desse trabalho.

3.1 Materiais

Nesta seção serão apresentados os hardwares e softwares utilizados. Primeiro são apresentados os computadores que foram usados para esse trabalho, e depois, os softwares e as bases de dados que possibilitaram o treinamento das ANNs.

As especificações dos computadores são mostradas na Tabela 1. O Computador 1 foi utilizado para treinar as ANNs, já que, possui duas placas de vídeo e cada placa dispõe de 2.944 *CUDA cores*. Já o Computador 2 foi utilizado para realizar testes utilizando o Grad-CAM e gerar áudios a partir da combinação do áudio original e o mapa de calor fornecido pelo Grad-cam.

Tabela 1 – Especificações dos computadores

Especificações	Computador 1	Computador 2
Processador	Intel(R) Xeon(R) Silver 4210R	Intel Core i7-7700HQ
Placa de vídeo	2x GeForce RTX 2080 SUPER	Nvidia GeForce GTX 1050 Ti
Memória RAM	62GB	16 GB
Sistema Operacional	Ubuntu	Windows 10

Fonte: Autoria própria (2021)

No projeto foi utilizada a linguagem de programação Python. Ela é uma linguagem de alto nível, multi-paradigma, de tipagem dinâmica, que possui diversas bibliotecas e *frameworks* para processamento de áudio e treinamento de ANNs. As bibliotecas que foram escolhidas são:

Pandas[1], Matplotlib[2], NumPy[3], Librosa[4] e PyTorch[5].

- Pandas: é uma biblioteca de manipulação e análise de dados de alta performance em diversos formatos e estruturas de dados, nesse trabalho ela foi utilizada para manipular os CSVs que fazem parte do conjunto de dados escolhidos;
- Matplotlib é utilizada para geração de gráficos 2D, projetada para o usuário ser capaz de gerá-los em apenas algumas linhas de código, em vista disso foi escolhida para gerar as representações dos áudios (espectrogramas e espectrogramas de mel) e também para salvar os resultados do Grad-CAM;
- Numpy é uma biblioteca desenvolvida para computação científica, que permite a manipulação de *arrays* de forma eficiente, e integra ao Python diversas funções matemáticas como a transformada discreta de Fourier. Foi escolhida pois algumas bibliotecas nesse trabalho, como o MatplotLib, a utilizam e para tratamento de dados em formato de *array*;
- Librosa é uma ferramenta para análise de áudios que prove diversas implementações para a área de recuperação de informações musicais. Foi escolhida com o objetivo de obter funções para a reconstrução de áudios;
- Pythorch é uma biblioteca de aprendizado de máquina que permite a criação e treinamento de ANNs utilizando a linguagem Python. Também, possibilita a escrita dessas ANNs de maneira simples e permite a utilização de placas de vídeo para o treinamento delas (PASZKE *et al.*, 2019). Por esse motivos foi escolhida como biblioteca para a construção e treinamento do modelo desse trabalho;
- Torchaudio é uma biblioteca que se integra ao Pytorch, provendo diversas funções para a utilização do Pytorch em áudios. Ela foi utilizada para gerar representações e manipulações nos áudios que compõe o conjunto de dados de treino e teste das ANNs.

O conjunto de dados que foi utilizado neste trabalho é o produzido pelo projeto SPIRA (CASANOVA *et al.*, 2021). Ele possui informações de idade, sexo, oxigenação sanguínea e gravação de áudios feitas por pessoas saudáveis (controle) e de pessoas internadas diagnosticadas com COVID-19. Porém, a oxigenação sanguínea foi apenas utilizada para a seleção dos pacientes, pois, para para ser um paciente válido era necessário que o nível de oxigenação sanguínea fosse inferior de 92%. Além disso, observou-se diversos problemas presentes nas amostras, especialmente a presença da voz do coletor, o que levou na remoção

[1]Disponíel em: `https://pandas.pydata.org/`
[2]Disponíel em: `https://matplotlib.org/`
[3]Disponíel em: `https://numpy.org/`
[4]Disponíel em: `https://librosa.org/`
[5]Disponíel em: `https://pytorch.org/`

de amostras resultando em um conjunto de dados com 432 instâncias.

As coletas restantes foram separadas em 292 para treino, 32 para validação e 108 para testes, totalizando $1,22$ horas de áudio. Os conjuntos de teste e validação foram balanceados por classe, e, o conjunto de validação também foi balanceado por sexo. A Tabela tal apresenta os dados do conjunto de dados final.

Tabela 2 – Informações do conjunto de dados

Conjuntos	Controle	Paciente	Quantidade de áudios	Duração total (s)
Treino	143	149	292	3110
Validação	16	16	32	296
Teste	54	54	108	983

Fonte: Adaptado de Casanova *et al.* (2021)

É importante salientar que esse conjunto de dados pode conter diferentes sotaques. As coletar de voz dos controles foi feita por uma plataforma *web* disponibilizada na Internet e as vozes dos pacientes foram coletadas em dois hospitais universitários diferentes que recebem pessoas de todos os estados brasileiros. Apesar disso, os resultados tendem a não sofrer viés de sotaque porque a maioria dos sotaques de controle e pacientes são de São Paulo.

3.2 Métodos

Nessa seção são abordados os métodos que serão aplicados durante a realização do trabalho.

3.2.1 Janelamento e inserção de ruídos

Os experimentos deste trabalho utilizam duas técnicas que provaram-se essenciais em Casanova *et al.* (2021), sendo elas o janelamento e a inserção de ruídos. Essas técnicas servem como aumento de dados e conseguiram reduzir o sobreajuste das ANNs treinadas para aquele trabalho.

O janelamento consiste em criar diversos áudios a partir do original com a mesma duração variando apenas o início de cada áudio, o inicio varia um segundo a cada áudio (janela) criado. A duração máxima dessas janelas é de 4 segundos, pois, essa é a duração do menor áudio do conjunto de dados. Também, o janelamento é utilizado para normalizar o tamanho (duração) da entrada para a ANN, assim, evitando problemas de dimensionalidade durante o processamento dos dados.

Já a inserção de ruídos, serve para incluir ruídos hospitalares em todas as instâncias. Isso evita que a ANN aprenda que áudios da classe paciente possuem ruídos semelhantes. Por isso, para o teste de ablação foram utilizados três ruídos para cada classe, enquanto, os experimentos de aumento de dados e transferência de aprendizado utilizaram três ruídos para pacientes e quatro para controles.

3.2.2 Teste de ablação

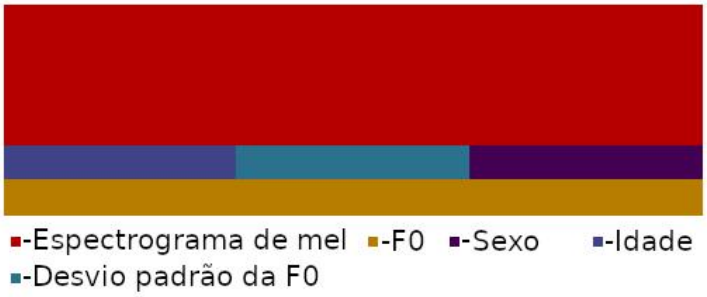
Ablação é um termo utilizado em diversas áreas do conhecimento com significados distintos. Em Inteligência Artificial esse termo representa a remoção total ou parcial de um componente do sistema. Isso é realizado para verificar se há relações entre esse componente e o todo (ERMAN; LESSER, 1990). Essa relação pode ser percebida pela mudança de performance entre os testes que contem todo sistema e testes que contem parte deste sistema.

A ANN de Casanova *et al.* (2021) provou-se capaz de detectar COVID-19 ao conseguir uma precisão de $91,6\%$ sobre o conjunto de teste do seu trabalho. Porém, não foram testadas outras representações de áudio ou tipos de dados para verificar investigar vieses causados pelo ruído hospitalar. Dessa forma, foi realizado um teste de ablação sobre idade, sexo (masculino e feminino), frequência fundamental da voz (F0) e desvio padrão da F0. Também foi testado o espectrograma de mel como representação dos áudios alimentados à ANN, pois ele retém mais informações espaciais que o Coeficientes Mel-Cepstrais.

São propostos três experimentos para o teste de ablação. No Experimento 1 foram criadas imagens que contém todos os dados e o espectrograma de mel, representada pela Figura 23. Porém, é necessário transformar os dados escalares (idade, gênero e desvio padrão da F0) em matrizes, e, transformar o vetor de F0 em uma sequência de barras na imagem. Isso deve-se ao fato de ser utilizada uma CNN e que o objetivo é gerar mapas de calor indicando quais entradas são mais importantes para a classificação. Assim, após o experimento com a

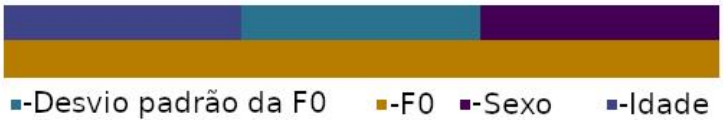
imagem completa, foi feito um experimento com uma imagem que só contém os dados escalares (Experimento 2), representado pela Figura 24, e um experimento com apenas o espectrograma de mel (Experimento 3) representado pela Figura 25.

Figura 23 – Exemplo de entrada para o Experimento 1



Fonte: Autoria própria (2021)

Figura 24 – Exemplo de entrada para o Experimento 2



Fonte: Autoria própria (2021)

Figura 25 – Exemplo de entrada para o Experimento 3

Fonte: Autoria própria (2021)

3.2.3 Mapas de calor gerados a partir do Grad-CAM

O Grad-CAM foi utilizado para gerar mapas de calor após o teste de ablação, pois, além da acurácia da ANN sobre o conjunto de teste é interessante saber quais das informações

foram mais importante para a classificação de cada resultado. Isso pode trazer explicações para a detecção de COVID-19 utilizando ANNs e também aferir algum possível viés durante o treinamento, já que o conjunto de dados possui diversos ruídos hospitalares que podem ter sido considerados pela rede durante sua tomada de decisão.

A análise de viés na rede foi realizada sobre os mapas de calor referentes ao terceiro experimento do teste de ablação, pois, é nele que a ANN irá analisar apenas o áudio durante o seu treinamento. Já nos outros dois experimentos foi possível identificar o peso de cada dado, assim, definir os dados essenciais para a detecção de COVID-19 utilizando ANNs.

3.2.4 Síntese de áudios

A fim de facilitar a análise dos mapas de calor do Experimento 3 do teste de ablação, foi realizada a síntese de áudios (A_s) baseados nesses resultados. Pois, isso permite a diferenciação sonora entre posições do áudio que os pesos da ANN foram mais influenciados.

Para isso, primeiro é feito o produto de Hadamard (multiplicação elemento a elemento) entre o espectrograma de mel original em decibel (A_o) e o mapa de calor (M) conforme mostra a Equação 29. Logo após será realizado alguns processos para converter esse espectrograma de mel em decibéis para um espectrograma linear. Por fim, utiliza-se a biblioteca Librosa e sua função *istft*[6] para converter o espectrograma linear e a fase do áudio em um novo áudio. Esse áudio sintetizado possibilitará a análise do mapa de calor gerado pelo Grad-CAM. O processo é apresentado pela Figura 26.

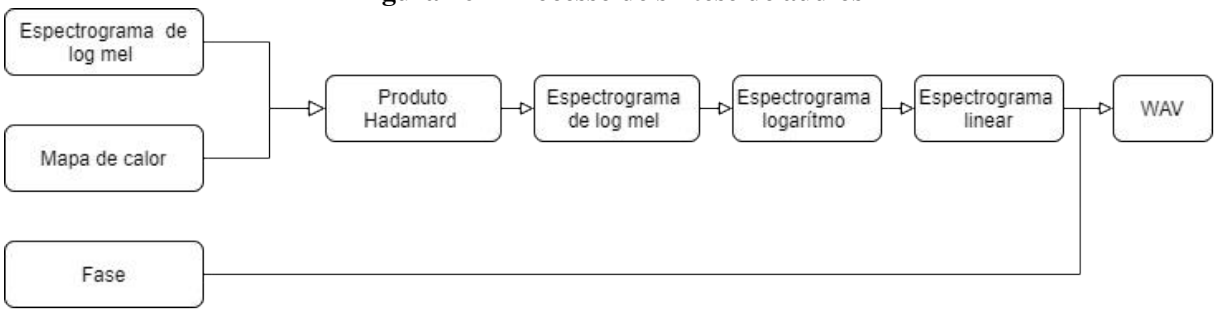
$$A_s = A_o \cdot M \tag{29}$$

3.2.5 Treino da PANN

Nos experimentos realizados no projeto SPIRA, percebeu-se que as redes baseadas em espectrogramas possuem desempenho menor que redes baseadas coeficientes cepstrais de mel. A fim de contornar esse deficit foi treinada a PANN CNN14 originada em Kong *et al.* (2020) por ser uma arquitetura simples e por ser pouco profunda. Visto que, PANNs provaram-se

[6] `https://librosa.org/doc/main/generated/librosa.istft.html`

Figura 26 – Processo de síntese de áudios



Fonte: Autoria própria (2021)

uma boa opção para a realização de transferência de aprendizado, e, por isso foi utilizada uma com intuito de saber se é possível aumentar a acurácia sobre o conjunto de teste ao utilizar o espectrograma de mel como representação dos áudios. Isso aumentará a qualidade dos mapas de calor gerado pelo Grad-CAM, já que, ele não troca performance por interpretabilidade.

3.2.6 Aumento da acurácia dos modelos com aumento de dados

Foi utilizado também as técnicas Mixup e SpecAugment a fim de melhorar os resultados de Casanova *et al.* (2021) ao treinar a SpiraNet e as PANNs, pois, o conjunto de dados possui poucos dados e a utilização de aumento de dados pode ser um fator crucial para a melhora da performance do modelo.

4 RESULTADOS E DISCUSSÃO

Neste capítulo serão apresentados os resultados dos experimentos e a discussão sobre eles. Cada resultado apresentado foi obtido após 1.000 épocas de treinamento e testados sobre o conjunto de dados do projeto SPIRA que contém no total 108 instâncias (54 para cada classe).

4.1 Experimento com Espectrograma, F0 e Dados Escalares

O Experimento 1 obteve maior acurácia entre os experimentos do teste de ablação. A ANN treinada para ele conseguiu a acurácia de $81,48\%$ e suas respostas estão apresentadas na Tabela 3. Logo percebe-se que a combinação entre o espectrograma de mel e os outros dados é uma abordagem interessante para a detecção de COVID-19.

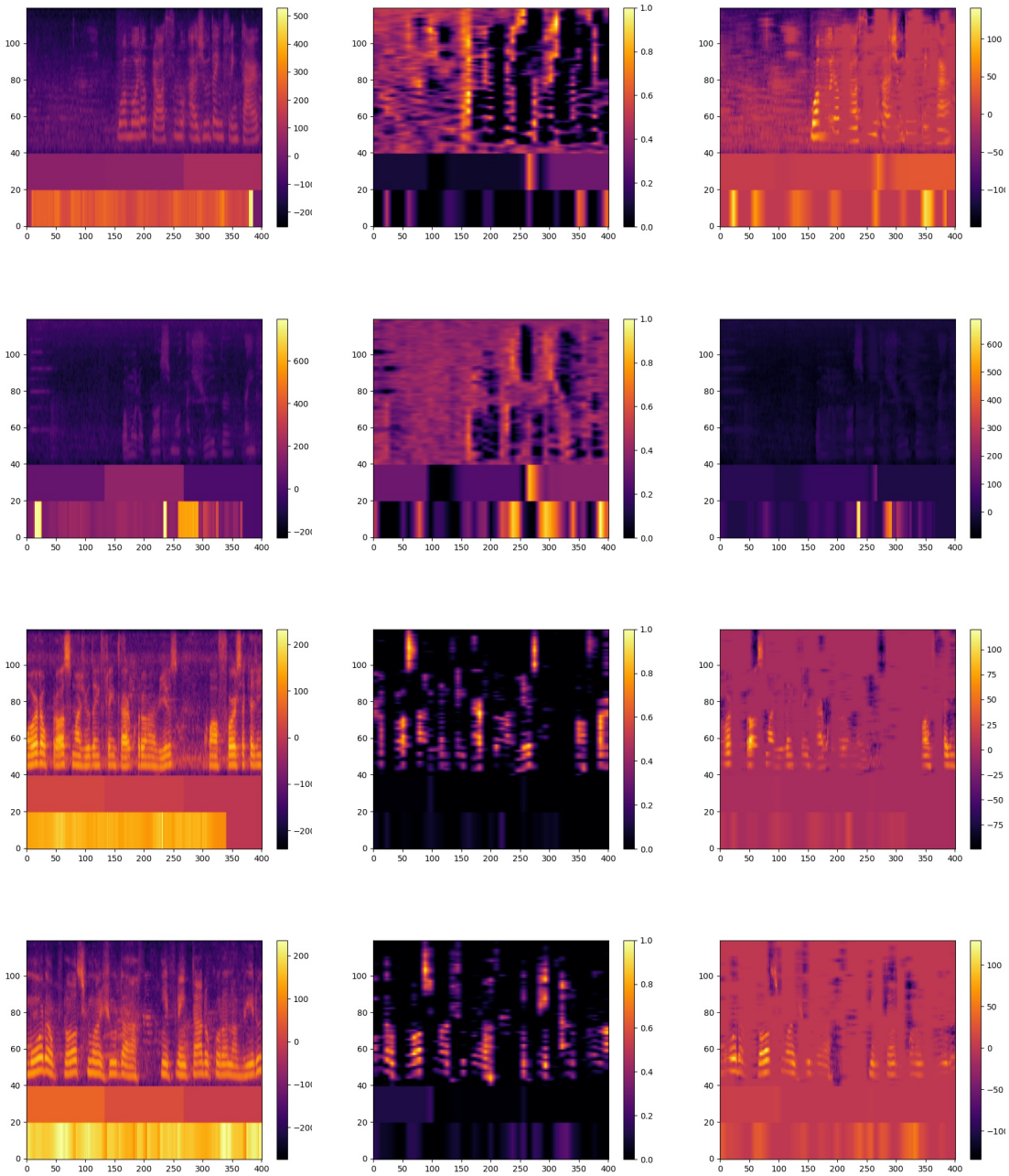
Tabela 3 – Resultado do Experimento 1

verdadeiro positivo	verdadeiro negativo	falso positivo	falso negativo
44	44	10	10

Fonte: Autoria própria (2021)

A Figura 27 traz quatro instâncias do conjunto de teste (esquerda) junto com seus respectivos mapas de calor (centro) e produtos de Hadamard (direita). As duas primeiras linhas dessa figura são de instâncias de controle classificadas corretamente. Nos seus mapas de calor é possível perceber que a ANN deu mais atenção para partes sem a voz humana no espectrograma, $F0$ e sexo. Entretanto, as duas últimas linhas são instâncias de pacientes classificadas corretamente, a ANN deu mais relevância para a voz humana. Já o desvio padrão da f0 e a idade são os dados que menos influenciam para a decisão desta ANN, pois, só esporadicamente são levados em consideração para as duas classes.

Figura 27 – Entrada original (esquerda); Mapa de Calor (centro); Produto (direita)



Fonte: Autoria própria (2021)

4.2 Experimento com F0 e Dados Escalares

O Experimento 2 foi realizado para investigar se a arquitetura utilizada era capaz de detectar COVID-19 sem o espectrograma e também investigar qual a importância do

espectrograma de mel para a tarefa, já que foi mostrado que, com a voz (no formato de coeficientes de mel-cepstrais), é possível detectar essa doença (CASANOVA *et al.*, 2021). A ANN treinada obteve acurácia de $68,51\%$ sendo o menor valor obtido em todos os experimentos do teste de ablação, e suas previsões são apresentadas pela Tabela 4. Isso demonstra a importância que a representação do áudio em formatos adequados (MFCCs, espectrogramas) traz para a detecção de COVID-19 ao utilizar o conjunto de dados do projeto SPIRA.

A Figura 28 apresenta quatro instâncias classificadas pela ANN, em que as duas primeiras linhas foram classificadas corretamente como controle e as duas últimas linhas classificadas corretamente como pacientes. A partir da figura percebe-se que a ANN treinada demonstrou atenção especial para o sexo, idade e a $F0$ durante a classificação de instâncias como controle. Esse comportamento repete-se para todas as instâncias que foram classificadas como controle. Já a atenção da ANN para os dados que foram classificados como paciente foi direcionada a $F0$. Isso é visualizado nas duas últimas linhas da Figura 28. Por outro lado o desvio padrão da $F0$ possui pouca importância para este experimento, visto que, ele não aparece em nenhum mapa de calor.

Tabela 4 – Resultado do Experimento 2

verdadeiro positivo	verdadeiro negativo	falso positivo	falso negativo
36	38	16	18

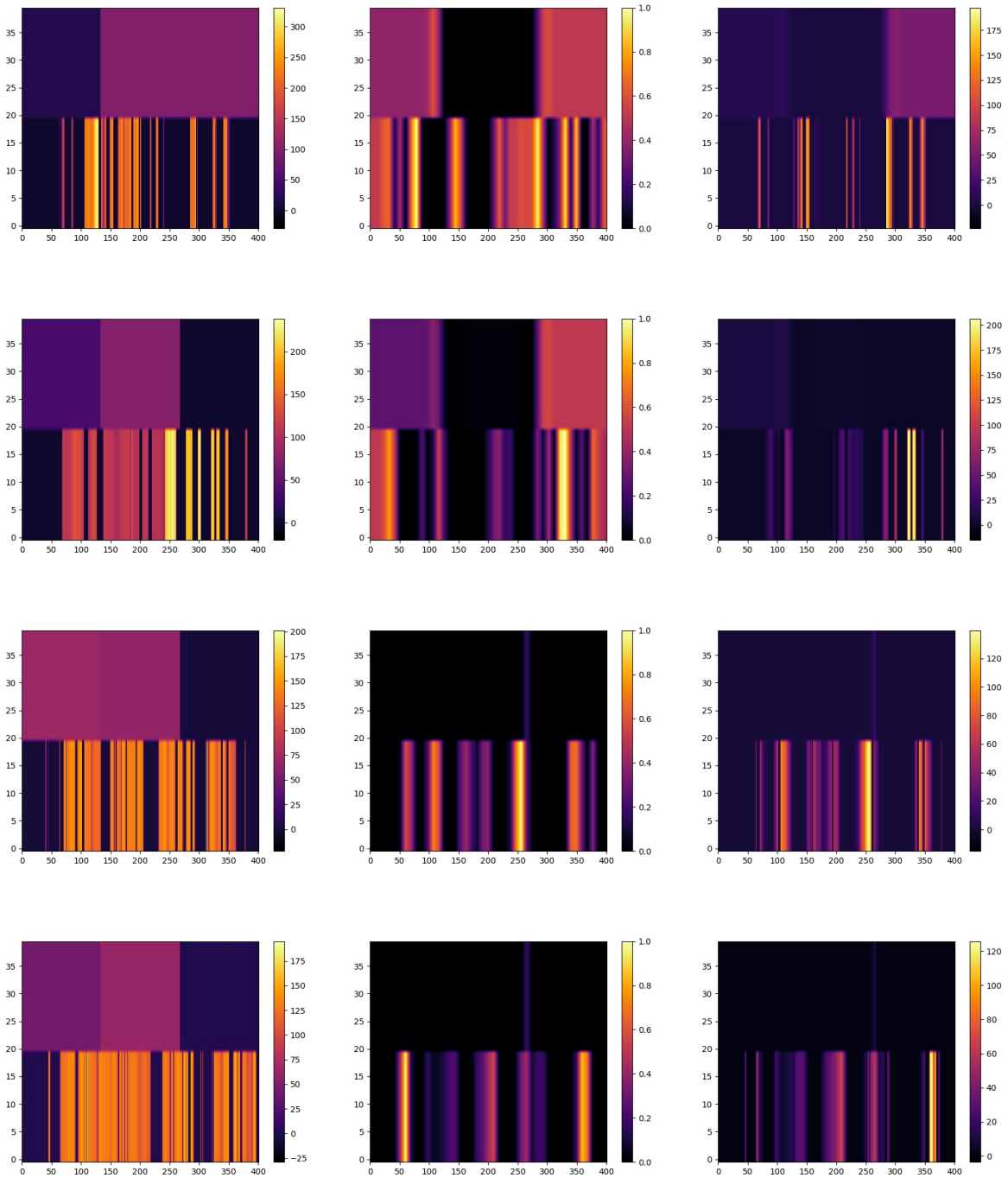
Fonte: Autoria própria (2021)

4.3 Experimento com Espectrograma de Mel

Para analisar a importância do espectrograma de mel e também verificar possíveis vieses no treinamento da ANN foi realizado o Experimento 3. Nele obteve-se a segunda maior acurácia do teste de ablação sendo de $79,62\%$, e as classificações da ANN são apresentadas pela Tabela 5. Esse resultado salienta ainda mais importância do espectrograma para este trabalho, visto que, há apenas $1,86\%$ de diferença entre as acurácias deste experimento e o Experimento 1.

Além disso, os mapas de calor gerados pelo Grad-CAM revelam que esta ANN utiliza especialmente a voz humana para gerar um resultado, em que são apresentados na Figura 29. Essa figura traz quatro amostras, classificadas corretamente, de áudio original (esquerda) junto

Figura 28 – Espectrograma original (esquerda); Mapa de Calor (centro); Produto (direita)



Fonte: Autoria própria (2021)

com seu mapa de calor (centro) e produto (direita), em que as duas primeiras linhas são de controles e as duas últimas são de pacientes. Isso fortalece a hipótese de que a ANN não possui viés em ruídos característicos gerados pelos locais de coleta. Contudo, pela complexidade de análise de um espectrograma de mel não é possível identificar diferenças entre instâncias dos

Tabela 5 – Resultado do Experimento 3

verdadeiro positivo	verdadeiro negativo	falso positivo	falso negativo
37	49	5	17

Fonte: Autoria própria (2021)

controles e pacientes, diferente dos outros dois experimentos do teste de ablação.

4.4 Ressíntese dos Áudios

Para facilitar a análise dos resultados obtidos no Experimento 3 foram transformados os produtos, no formato de espectrograma de mel, em áudios. Essa abordagem permite descobrir se a ANN direciona sua atenção para palavras especificas ou para fonemas. Também, é possível investigar a existência de vieses e qual viés afeta mais a decisão da ANN.

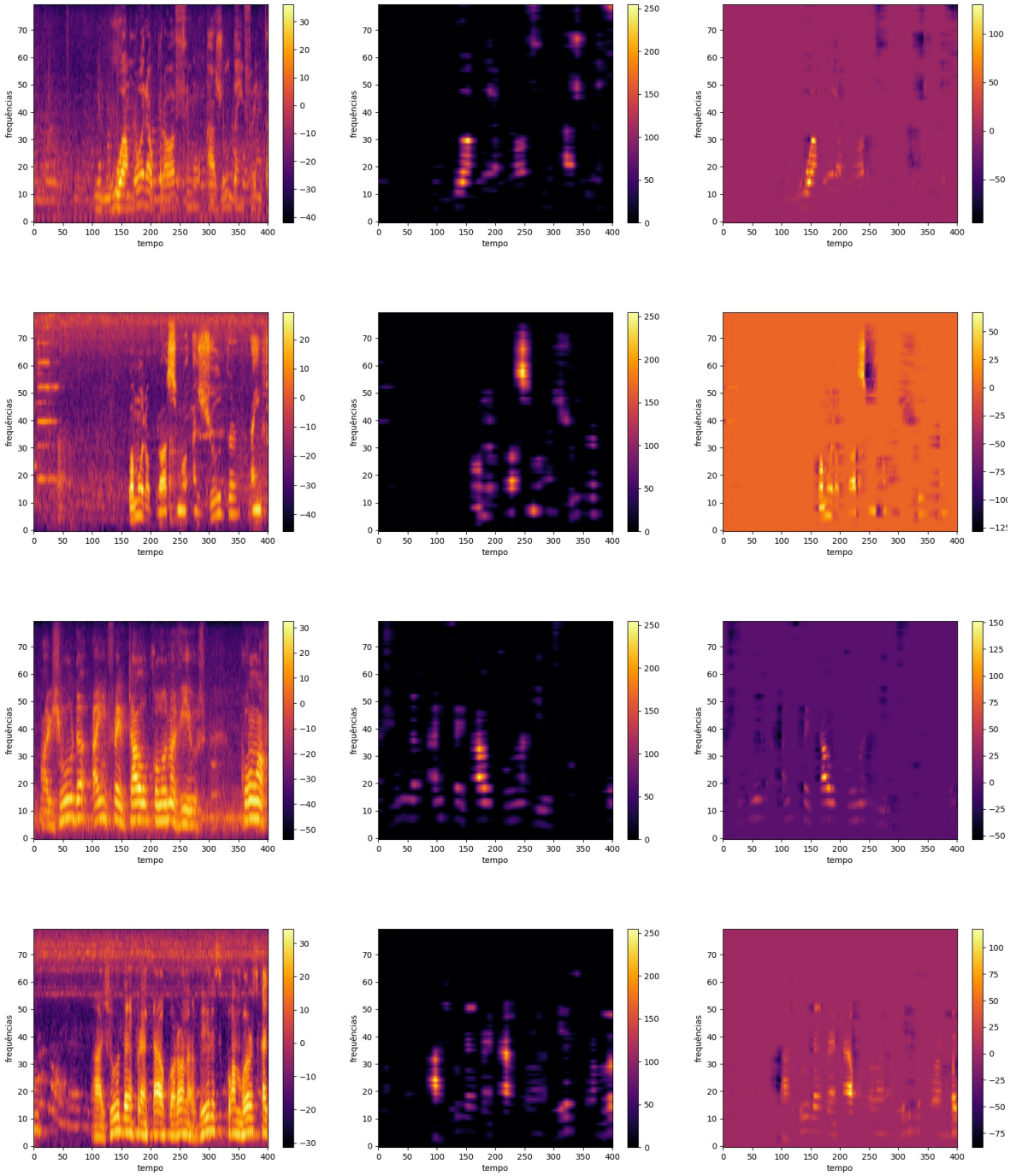
Após a síntese e análise dos áudios percebeu-se que a atenção da ANN é variante entre as duas classes. Para a classificação de pacientes a atenção da ANN foi direcionada para as sílabas tônicas e palavras longas, conforme é apresentado na Figura 30. Entretanto, para a classificação dos controles a ANN demonstrou interesse para silabas átonas, preposições e palavras pronunciadas juntas conforme é apresentado pelas Figura 31.

Além disso, percebe-se uma grande redução do volume entre o áudio original e o produto indicando que muitas frequências foram ignoradas ou que a ANN atribuiu pouca importância a elas, como mostra a Figura 32. E a partir dos resultados obtidos também é possível reforçar que a ANN não baseia-se em ruídos provenientes dos ambientes de coleta, visto que, a voz humana sempre está presente e não é obstruída por ruídos.

4.5 Experimento com Transferência de Aprendizado

A fim de conseguir uma ANN alimentada por espectrogramas de mel que consiga superar a acurácia do estado da arte foi trainada a PANN CNN14 para a tarefa deste trabalho. Visto que, em testes preliminares de Casanova *et al.* (2021) não obteve-se bons resultados ao

Figura 29 – Espectrograma original (esquerda); Mapa de Calor (centro); Produto (direita)

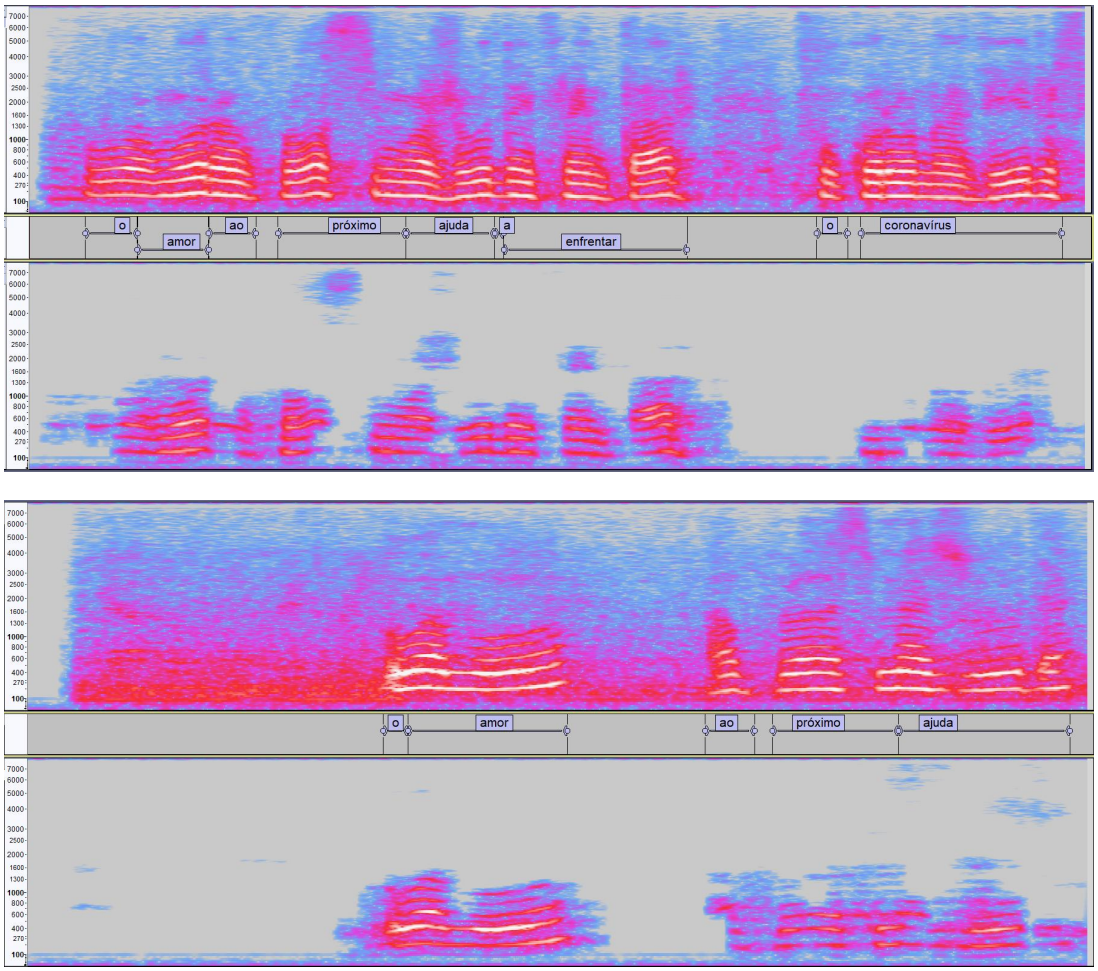


Fonte: Autoria própria (2021)

utilizar arquiteturas alimentadas por espectrogramas, também, isso pode ser reforçado pelo Experimento 3 que obteve $79,62\%$ de acurácia.

Após o treinamento da PANN obteve-se um acurácia de $94,44\%$ e suas predições são apresentadas pela Tabela 6. Além disso, essa ANN conseguiu superar as ANNs treinadas para o

Figura 30 – Amostra de áudios sintetizados classificados como paciente. Original (cima) produto (baixo)



Fonte: Autoria própria (2021)

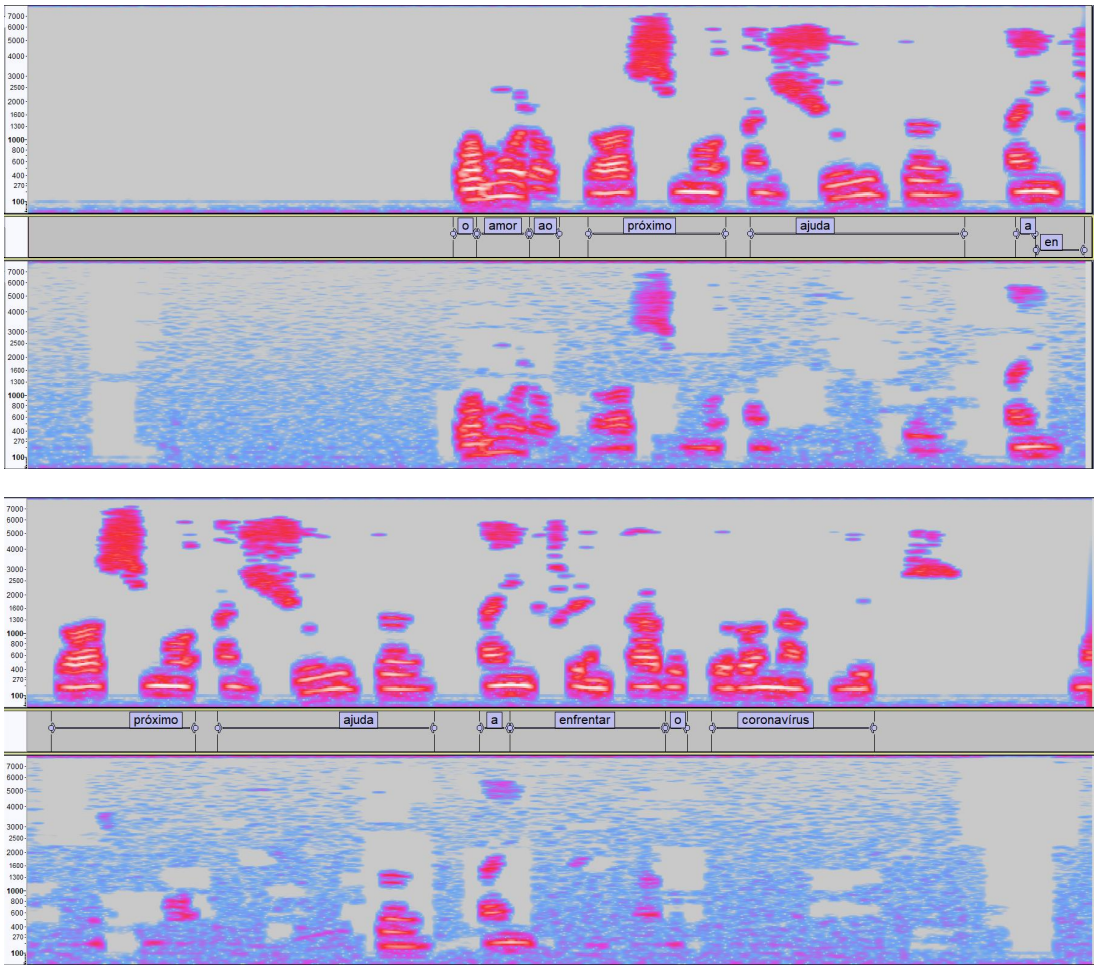
teste de ablação e também os resultados do estado da arte. Isso mostra a importância de utilizar transferência de aprendizado para tarefas complexas ou que possuem poucos dados igual a deste trabalho. Pois, quanto maior a quantidade de dados melhor será a generalização das ANNs para as tarefas.

Tabela 6 – Resultado do Experimento com Transferência de Aprendizado

verdadeiro positivo	verdadeiro negativo	falso positivo	falso negativo
51	51	3	3

Fonte: Autoria própria (2021)

Figura 31 – Amostra de áudios sintetizados classificados como controle. Original (cima) produto (baixo)



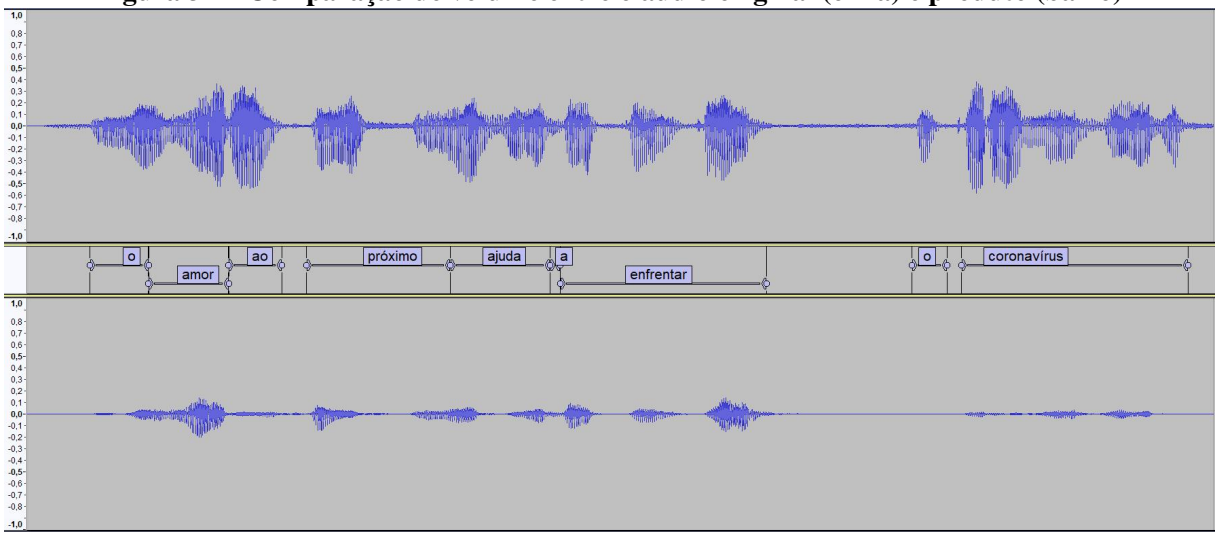
Fonte: Autoria própria (2021)

4.6 Experimento com Mixup e SpecAugment

O aumento de dados, como a transferência de aprendizado, é uma técnica utilizada para facilitar a convergência em tarefas complexas ou que possuam poucos dados. Por isso, neste trabalho as técnicas Mixup e SpecAugment foram utilizadas sobre o conjunto de dados.

Contudo, a acurácia da ANN treinada foi a menor entre todos os testes sendo de $65,74$ e suas predições são apresentadas pela Tabela 7. Essa tabela traz sugestão de que a ANN sofreu subajuste durante o seu treinamento, uma vez que, suas previsões são em maioria da classe paciente.

Figura 32 – Comparação de volume entre o áudio original (cima) e produto (baixo)



Fonte: Autoria própria (2021)

Tabela 7 – Resultado do Experimento com Mixup e SpecAugment

verdadeiro positivo	verdadeiro negativo	falso positivo	falso negativo
49	22	32	5

Fonte: Autoria própria (2021)

4.7 Discussão

Neste capítulo foram apresentados os resultados obtidos após o treinamento das ANNs e a síntese dos produtos. Esses resultados trouxeram explicações sobre o funcionamento de uma ANN ao ser treinada para tarefa de detecção de COVID-19.

O Experimento 1 foi o experimento com maior acurácia no teste de ablação. Com ele foi possível verificar que a $F0$, sexo e a voz humana foram importantes durante o processo de decisão para essa ANN. Enquanto, a idade e o desvio padrão da $F0$ não influenciaram tanto as decisões da ANN, visto que, esses dados são levados em consideração em apenas algumas instâncias de controle e paciente. Entretanto, ao analisar os mapas de calor gerados pelo Grad-CAM percebe-se um possível viés de silêncio ou ruído durante a classificação de controles. Contudo, a ANN pode estar utilizando a pausa dos controles como uma característica importante, visto que, é suposto que pacientes façam mais pausas do que controles por causa da insuficiência respiratória.

O Experimento 2 obteve a menor acurácia no teste de ablação e a segunda menor

acurácia entre todos os experimentos. Esse resultado pode ter sido influenciado pelo fato do conjunto de dados ser pequeno e também pela falta de normalização dos dados. Isso deve-se a dificuldade de encontrar uma maneira de normalizar os dados (idade, sexo, $F0$ e desvio padrão da $F0$) sem que nenhum perca sua importância para a ANN. Contudo, através dos mapas de calor gerados percebeu-se que a ANN direcionou sua atenção para o sexo, idade e $F0$ para a distinção das classes. Observou-se que a ANN também utilizou a $F0$ para a distinção de classes similar ao Experimento 1. Logo, este experimento é relevante pois mostrou a importância da representação do áudio e a possibilidade de sexo, idade e $F0$ serem utilizados para detecção de COVID-19.

O Experimento 3 obteve a segunda maior acurácia do teste de ablação, $79,62\%$. Nele foi possível verificar que, a partir dos mapas de calor, o espectrograma de mel é capaz de conter diversas informações importantes para a detecção de COVID-19 e também percebeu-se que esta ANN não está com vieses, pois, os mapas de calor salientam principalmente partes que contém a voz.

Para validar o resultado obtido no Experimento 3 foi realizada a síntese de áudios a partir dos produtos resultantes desse experimento. Após a análise dos áudios provou-se que a ANN não está com viés em ruídos característicos, pois não é possível ouvir a presença desses ruídos nos áudios gerados. Além disso, percebe-se que a ANN demonstrou interesse nas sílabas tônicas e em pronúncias prolongadas para a classificação de pacientes. Isso pode indicar que a ANN está diferenciando as classes a partir do sintoma de insuficiência respiratória, pois pessoas que possuem esse sintoma costumam ter uma fala prolongada e também fazem mais esforço durante a pronunciação de palavras. Porém, para validar essa hipótese é necessário entregar esses áudios para profissionais da fonoaudiologia e linguística, já que, eles possuem as técnicas necessárias para avaliar os áudios profundamente.

O experimento com transferência de aprendizado e experimento com Mixup e SpecAugment foram realizados com o intuito de aumentar a acurácia do estado da arte. Porém, foi apenas possível ultrapassar a acurácia do estado da arte com PANN provando que transferência de aprendizado é uma abordagem boa para tarefas complexas ou tarefas que possuem poucos dados. Em comparação com estado da arte essa ANN conseguiu acertar três instâncias a mais no conjunto de teste, isso é um resultado significativo já que em um teste real seriam três pessoas a mais quer teriam suas triagens agilizadas. Também, esse experimento mostra um grande potencial para ser utilizado em conjunto com o Grad-CAM, já que esse algoritmo não troca acurácia por interpretabilidade. Entretanto, o experimento de aumento de dados obteve a menor acurácia entre os experimentos e apresentou subajuste. Isso pode ter sido influenciado pelas técnicas de aumento de dados utilizadas visto que o SpecAugment remove

porções do áudio, então, pode ter sido removido partes do áudio necessárias para a tarefa, como visto na síntese dos áudio em que a rede utilizou sílabas tônicas e atonas para classificação de áudios. Por outro lado, a técnica Mixup pode ter criado uma distorção nos áudios em que a ANN não conseguiu extrair informações suficientes para diferenciar as classes da tarefa.

Em suma, apesar do experimento com aumento de dados não ter sido tão bem sucedido os outros apresentaram resultados compatíveis ou superiores ao estado da arte. Esses trouxeram expilações para como as ANNs se comportam para a tarefa de detecção de COVID-19, confirmaram que não há viés causado por ruídos característicos e também o experimento com PANN superou a acurácia do estado da arte.

5 CONCLUSÃO

Neste trabalho explorou-se a análise de ANNs para detecção de COVID-19 utilizando o algoritmo Grad-CAM e a tentativa de superar o estado da arte com técnicas de aumento de dados e transferência de aprendizado.

Todos os objetivos propostos neste trabalhos foram concluídos. Assim, com a síntese de produtos foi possível descobrir que sílabas tônicas e pronúncias prolongadas são características importantes para a classificação de pessoas com a COVID-19. Além disso, esses áudios comprovaram que os ruídos não influenciaram as decisões da ANN no Experimento 3. Também, a frequência fundamental da voz pode exercer um papel importante, porém, é necessário ajustar as ANNs que utilizaram esse dado para obter resultados melhores. E, foi possível superar o estado da arte ao obter-se $94,44\%$ de acurácia utilizando uma PANN, e isso significa que em um teste real três pessoas a mais teriam suas triagens agilizadas.

5.1 Trabalhos futuros

Para possíveis trabalhos futuros baseados neste, indica-se:

- Refazer experimentos deste trabalho com um conjunto de dados maior;
- Solicitar que profissionais da área da linguística e fonoaudiologia analisem os áudios sintetizados (produtos);
- Refazer experimento com aumento de dados utilizando uma técnica por vez e também alterar hiperparâmetros;
- Treinar ANNs sem o uso de janelamento para verificar qualidade dos produtos;
- Realizar teste de significância sobre os Experimentos 1 e 3;
- Utilizar o Grad-CAM para realçar partes das entradas menos influenciaram para a classificação e depois sintetizar áudios com esses resultados;
- Sintetizar áudios a partir do experimento de transferência de aprendizado.

REFERÊNCIAS

AGGARWAL, C. **Neural Networks and Deep Learning: A Textbook**. Gewerbestrasse 11, 6330 Cham, Switzerland: Springer International Publishing, 2018. ISBN 9783319944630.

BALLOU, G. **Handbook for sound engineers**. 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA: Elsevier, 2008.

CASANOVA, E.; GRIS, L.; CAMARGO, A.; SILVA, D. da; GAZZOLA, M.; SABINO, E.; LEVIN, A.; JR, A. C.; ALUISIO, S.; FINGER, M. Deep learning against COVID-19: Respiratory insufficiency detection in Brazilian Portuguese speech. In: **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**. Online: Association for Computational Linguistics, 2021. p. 625–633. Disponível em: <https://aclanthology.org/2021.findings-acl.55>. Acesso em: 25 de outubro de 2021.

COPPIN, B. **Artificial intelligence illuminated**. Sudbury, MA, United States: Jones & Bartlett Learning, 2004.

DOWNEY, A. B. **Think DSP: Digital Signal Processing in Python**. 1st. ed. Needham, Massachusetts, United States: O'Reilly Media, Inc., 2016. ISBN 1491938455.

ERMAN, L. D.; LESSER, V. R. The hearsay-ii speech understanding system: A tutorial. **Readings in Speech Recognition**, Morgan Kaufmann, p. 235–245, 1990.

ERTEL, W. **Introduction to Artificial Intelligence**. 2nd. ed. Gewerbestrasse 11, 6330 Cham, Switzerland: Springer Publishing Company, Incorporated, 2017. ISBN 9783319584874.

FONSECA, N. **Introdução À Engenharia De Som**. R. D. Estefânia, Lisboa, Portugal: FCA, 2007.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 1005 Gravenstein Highway North, Sebastopol, CA 95472, United States: O'Reilly Media, 2019.

GIBSON, J.; SEGBROECK, M. V.; NARAYANAN, S. Comparing time-frequency representations for directional derivative features. **Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH**, 09 2014.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. 1 Broadway, Cambridge, MA 02142, United States: MIT Press, 2016. `http://www.deeplearningbook.org`.

GOPALAKRISHNAN, K.; KHAITAN, S. K.; CHOUDHARY, A.; AGRAWAL, A. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. **Construction and building materials**, Elsevier, v. 157, p. 322–330, 2017.

HAYKIN, S. **Neural Networks and Learning Machines, 3/E**. Upper Saddle River, New Jersey 07458, United States: Pearson Education, 2010.

HORNER, M.; HALLIDAY, S.; BLYTH, S.; ADAMS, R.; WHEATON, S. **The Free High School Science Texts: A Textbook for High School Students Studying Physics.** Rondebosch 7701, South Africa: Free High School Science Texts, 2005.

ISER, B. P. M.; SLIVA, I.; RAYMUNDO, V. T.; POLETO, M. B.; SCHUELTER-TREVISOL, F.; BOBINSKI, F. Definição de caso suspeito da covid-19: uma revisão narrativa dos sinais e sintomas mais frequentes entre os casos confirmados. **Epidemiologia e Serviços de Saúde**, SciELO Brasil, v. 29, 2020.

KALANTARIAN, H.; ALSHURAFA, N.; POURHOMAYOUN, M.; SARIN, S.; LE, T.; SARRAFZADEH, M. Spectrogram-based audio classification of nutrition intake. **2014 IEEE Healthcare Innovation Conference, HIC 2014**, p. 161–164, 02 2015.

KONG, Q.; CAO, Y.; IQBAL, T.; WANG, Y.; WANG, W.; PLUMBLEY, M. D. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, IEEE, v. 28, p. 2880–2894, 2020.

KULAKOV, A. **2 reasons to use MixUp Augmentation when training your Deep Learning models**. 2020. Disponível em: <https://towardsdatascience.com/2-reasons-to-use-mixup-when-training-yor-deep-learning-models-58728f15c559>. Acesso em: 25 de outubro de 2021.

LANCET, T. Covid-19 in brazil:"so what?". **Lancet (London, England)**, Elsevier, v. 395, n. 10235, p. 1461, 2020.

MACCAGNO, A.; MASTROPIETRO, A.; MAZZIOTTA, U.; SCARPINITI, M.; LEE, Y.-C.; UNCINI, A. A cnn approach for audio classification in construction sites. In: **Progresses in Artificial Intelligence and Neural Systems**. 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore: Springer, 2021. p. 371–381.

MAGNO, L.; ROSSI, T. A.; MENDONÇA-LIMA, F. W. d.; SANTOS, C. C. d.; CAMPOS, G. B.; MARQUES, L. M.; PEREIRA, M.; PRADO, N. M. d. B. L.; DOURADO, I. Desafios e propostas para ampliação da testagem e diagnóstico para covid-19 no brasil. **Ciencia & saude coletiva**, SciELO Brasil, v. 25, p. 3355–3364, 2020.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MISRA, D. Mish: A self regularized non-monotonic neural activation function. **CoRR**, abs/1908.08681, 2019.

NIELSEN, M. A. **Neural networks and deep learning**. San Francisco, CA, USA: Determination press, 2015.

NWANKPA, C.; IJOMAH, W.; GACHAGAN, A.; MARSHALL, S. **Activation Functions: Comparison of trends in Practice and Research for Deep Learning**. 2018.

PARK, D. S.; CHAN, W.; ZHANG, Y.; CHIU, C.-C.; ZOPH, B.; CUBUK, E. D.; LE, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. **arXiv preprint arXiv:1904.08779**, 2019.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KÖPF, A.; YANG, E. Z.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. **CoRR**, abs/1912.01703, 2019. Disponível em: <http://arxiv.org/abs/1912.01703>. Acesso em: 25 de outubro de 2021.

PEDAMONTI, D. **Comparison of non-linear activation functions for deep neural networks on MNIST classification task**. 2018.

PEREZ, L.; WANG, J. The effectiveness of data augmentation in image classification using deep learning. **arXiv preprint arXiv:1712.04621**, 2017.

QUINTANILHA, I. M. **End-to-end speech recognition applied to brazilian portuguese using deep learning**. Tese (Doutorado) — MSc dissertation, PEE/COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, 2017.

ROCKMORE, D. N. The fft: an algorithm the whole family can use. **Computing in Science & Engineering**, IEEE, v. 2, n. 1, p. 60–64, 2000.

ROSENBALTT, F. The perceptron–a perciving and recognizing automation. **Report 85-460-1 Cornell Aeronautical Laboratory, Ithaca, Tech. Rep.**, 1957.

RUDER, S.; PETERS, M. E.; SWAYAMDIPTA, S.; WOLF, T. Transfer learning in natural language processing. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 15–18. Disponível em: <https://aclanthology.org/N19-5004>. Acesso em: 25 de outubro de 2021.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. Rio de Janeiro, Brasil: Elsevier, 2013.

SELVARAJU, R. R.; DAS, A.; VEDANTAM, R.; COGSWELL, M.; PARIKH, D.; BATRA, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. **CoRR**, abs/1610.02391, 2016. Disponível em: <http://arxiv.org/abs/1610.02391>. Acesso em: 25 de outubro de 2021.

SELVARAJU, R. R.; DAS, A.; VEDANTAM, R.; COGSWELL, M.; PARIKH, D.; BATRA, D. Grad-cam: Why did you say that? **arXiv preprint arXiv:1611.07450**, 2016.

SILVA, I. N. d.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. Av. Diógenes Ribeiro de Lima, 3294 05083-010, São Paulo, Brasil: Artliber Editora, 2010.

SKANSI, S. **Introduction to Deep Learning: from logical calculus to artificial intelligence**. Gewerbestrasse 11, 6330 Cham, Switzerland: Springer, 2018.

SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. USA: California Technical Publishing, 1999. ISBN 0966017668.

TORREY, L.; SHAVLIK, J. Transfer learning. In: **Handbook of research on machine learning applications and trends: algorithms, methods, and techniques**. 701 E. Chocolate Avenue, Suite 200 Hershey PA, United States: IGI global, 2010. p. 242–264.

VELLIDO, A. The importance of interpretability and visualization in machine learning for applications in medicine and health care. **Neural computing and applications**, Springer, v. 32, n. 24, p. 18069–18083, 2020.

VENTURA, D. d. F. L.; AITH, F. M. A.; RACHED, D. H. A emergência do novo coronavírus e a "lei de quarentena" no brasil. **Revista Direito e Práxis**, SciELO Brasil, v. 12, p. 102–138, 2021.

WEISSTEIN, E. W. Beta distribution. **https://mathworld. wolfram. com/**, Wolfram Research, Inc., 2003.

WINER, E. **The audio expert: everything you need to know about audio**. 711 Third Avenue, New York, NY 10017, United States: Focal Press, 2018.

WYSE, L. Audio spectrogram representations for processing with convolutional neural networks. 06 2017.

XU, K.; FENG, D.; MI, H.; ZHU, B.; WANG, D.; ZHANG, L.; CAI, H.; LIU, S. Mixup-based acoustic scene classification using multi-channel convolutional neural network. **CoRR**, abs/1805.07319, 2018. Disponível em: <http://arxiv.org/abs/1805.07319>. Acesso em: 25 de outubro de 2021.

ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. Dive into deep learning. **arXiv preprint arXiv:2106.11342**, 2021.

ZHANG, H.; CISSE, M.; DAUPHIN, Y. N.; LOPEZ-PAZ, D. mixup: Beyond empirical risk minimization. **arXiv preprint arXiv:1710.09412**, 2017.

ZHOU, B.; KHOSLA, A.; LAPEDRIZA, À.; OLIVA, A.; TORRALBA, A. Learning deep features for discriminative localization. **CoRR**, abs/1512.04150, 2015. Disponível em: <http://arxiv.org/abs/1512.04150>. Acesso em: 25 de outubro de 2021.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. **Proceedings of the IEEE**, IEEE, v. 109, n. 1, p. 43–76, 2020.