Um Estudo sobre Técnicas utilizadas para o Reconhecimento de Sons com o uso de Inteligência Artificial e Python

Jéfer Benedett Dörr
Universidade Estadual de Maringá - UEM
Maringá, Brasil
pg54802@uem.br

Linnyer Beatrys Ruiz Aylon
Universidade Estadual de Maringá - UEM
Maringá, Brasil
lbruiz@uem.br

***Abstract*—This article presents an introduction to concepts and practices to learn the basics of working with Sound Recognition. The necessary theoretical background, concepts and tools to work in a practical project of Pattern Recognition in audio using Artificial Intelligence with the Python programming language will be presented.**

***Resumo*—Este artigo apresenta uma introdução de conceitos e práticas para aprender o inicial de como trabalhar com Reconhecimento de Sons. Será apresentado o embasamento teórico necessário, os conceitos e as ferramentas para trabalhar na prática em um projeto prático de Reconhecimento de Padrões em áudio utilizando Inteligência Artificial com a linguagem de programação *Python*.**

***Palavras-chave*—Reconhecimento de Áudio; Identificação de Áudio; Inteligência Artificial; Aprendizagem de Máquina; *Machine Listening*.**

I. Introdução

O objetivo deste trabalho é apresentar um estudo bibliográfico sobre técnicas de processamento e análise de áudio, com o uso da linguagem de programação Python e técnicas de inteligência artificial. A análise de dados de áudio trata da compreensão dos sinais de áudio capturados por dispositivos digitais, com inúmeras aplicações seja na indústria, saúde, produtividade e cidades inteligentes - *Smart Cities*. As aplicações incluem análise de satisfação do cliente a partir de chamadas, auxílio no diagnóstico médico, monitoramento de paciente com tecnologias assistivas para pessoas idosas ou par amelhoria de qualidade de vida [1]–[8], análise de áudio para segurança pública ou planejamento na tomada de decisões contexto de cidades inteligentes [9]–[17], análise sonora de defeitos em teste de qualidade na indústria [18]–[20], monitoramento não invasivo de florestas [21]–[27] identificando conflitos com humanos e vida silvestre - *Human–wildlife conflict (HWC)* [28] (armas de fogo, motosserras, presença de animais perigosos...) [29] ou análise da vocalização de animais (ecoacústica [30]), acompanhando migração de pássaros, sons subaquáticos de baleias [31], identificando presenção de alguma espécie na recuperação de florestas, entre diversas outras aplicações. O *smart audio* [32], permite interpretar o som, *acoustic sensing*, produzindo dados relevantes para tomadas de decisão e pode ser aplicado ou trazer contribuições em diversos campos.

O reconhecimento sonoro por Inteligência Artificial - *Artificial Intelligence (AI)*, técnica que permite aos computadores imitar a inteligência humana utiliza a Aprendizagem de Máquina - *Machine Learning* - (ML) com o Aprendizado Profundo - *Deep Learning (DL)* para realizar o Reconhecimento de Padrões - *Pattern recognition - (PR)* seja detectando sons com aprendizado profundo (*Detecting Sounds with Deep Learning*) ou reconhecimento sons usando aprendizado de máquina (*Recognizing Sounds Using Machine Learning*) para poder realizar atividades como uma classificação de gênero musical, um agrupamento (segmentação) de sons por semelhança na Classificação de som ambiental (*Environmental Sound Classification*), a classificação de eventos específicos de áudio (*Audio Event Recognition* (AER)), a detecção e interpretação da voz no Reconhecimento Automático de Fala (*Automatic Speech Recognition*) (ASR), como o realizado nos assistentes virtuais tipo a *Alexa*, a *Siri*, *Cortana* ou o *Google Home*. Este material fornece uma introdução e atividades práticas guiadas dos conceitos básicos de uso de som na programação para extração de características de áudio, classificação e segmentação de som.

Em 2019 o reconhecimento de som era considerado uma tecnologia estratégica fundamental para *AI* [33]. Em 2020 o reconhecimento de som permaneceu um campo relativamente inexplorado por ser considerado difícil [34]. Em 2021, a *IBM* apontou o som como sendo uma nova fonte de dados para a indústria 4.0 [35] e a Gartner[1], respeitada empresa de

[1] https://www.gartner.com/en

consultoria, destacou a Internet das Coisas - *Internet of Things - IoT* como uma das tendências que definirão o futuro da TI [36].

Este trabalho tem como objetivo apresentar os conceitos necessários para trabalhar com interpretação do som na programação (arquivos de áudio, propriedades, ondas, espectrogramas, etc). Em seguida, são apresentados alguns *datasets* de áudio utilizados em trabalhos de reconhecimento de áudio e os conceitos de *AI* necessários para a realização destas tarefas. Unindo estes dois conhecimentos, são apresentados alguns modelos pré-treinados de reconhecimento de sons e, na sequência, as ferramentas de programação e bibliotecas de apoio para tarefas de *AI* e as para trabalhar com sons. Após a base teórica e arcabouço de ferraemntas, são demonstradas atividades práticas guiadas de forma incremental, iniciando em como importar áudio, executar áudio, gerar espectrogramas até atividades de básicas de reconhecimento de padrões em áudio. Com isso, este embasamento teórico inicial, juntamente com as práticas dirigidas disponibilizadas neste material, buscam mostrar um caminho para inicial em atividades de reconhecimento de som.

II. Referencial Teórico

Esta seção irá apresentar a teoria necessária para entender e permitir trabalhar com o reconhecimento de sons. A seguir, alguns conceitos e terminologias relacionados ao reconhecimento de sons com programação:

A. Inteligência Artificial

A *AI* é considerada uma das principais tecnologias disruptivas da atualidade. A *AI* não é um termo novo, foi cunhado em 1956 por John McCarthy [37] e, segundo ele a *AI* busca tornar as máquinas inteligentes imitando a inteligência humana.

Uma Rede Neural Artificial - *Artificial Neural Network (ANN)*, utilizada na *AI*, é uma modelagem matemática do processo de aprendizado baseada em neurônios, e pode ser aplicadas em aprendizado supervisionado ou não-supervisionado.

O Aprendizado Supervisionado é a tarefa de aprendizado de máquina que consiste em aprender uma função que mapeia uma entrada para uma saída com base em pares de entrada-saída de exemplo. Inferindo uma função a partir de dados de treinamento rotulados, oriundos de um conjunto de exemplos de treinamento.

Já o Aprendizado Não Supervisionado é um tipo de aprendizado de máquina onde não é fornecido nenhum rótulo pré-atribuído ou pontuação para os dados de treinamento do algoritmo e, por similaridade de características, o algoritmo irá aprender e agrupar os dados.

O *PR* busca identificar algum padrão conhecido, as principais técnicas de reconhecimento de padrões em áudio são:

- *Template Matching* - Correspondência de modelo - combina as características dos dados com um modelo pré-gravado e definido.
- *Structural / Syntactic* - Estrutural / Sintático - esta forma de reconhecimento de padrão de áudio ajuda a definir relações entre diferentes elementos e componentes, como classes de áudio. Essa técnica de reconhecimento de som envolve aprendizado de máquina semi supervisionado.
- *Statistica* - Estatística - é usada para identificar o local onde um dado específico pertence e envolve aprendizado de máquina supervisionado.

1) Aprendizagem de Máquina: O *ML* é um subcampo da *AI* [38] e busca tornar as máquinas inteligentes utilizando métodos estatísticos que possibilitem as máquinas aprenderem a partir de dados [39]. São exemplos de alguns algoritmos de *ML* utilizados no contexto de reconhecimento de sons: *K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB),* e *Isolation Forest*. No *ML*, atuam as *ANN* com múltiplas camadas que assimilam tarefas e realizam reconhecimentos a partir de dados. Estas *ANNs*, utilizadas na *AI* foram descritas em 1943 como sendo estruturas de raciocínio artificiais baseadas em modelos matemáticos que simulariam o sistema nervoso humano [40].

Para compreender mais sobre *ML*, é necessário conhecer mais alguns conceitos envolvidos:

- Em *ML*, *Overfitting* ou sobreajuste, é um termo usado em estatística para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados, mas se mostra ineficaz para prever novos resultados. Ocorre quando a acurácia no conjunto de treinamento é maior do que no conjunto de testes, ou seja, o modelo está treinado em excesso para as amostras de treino e não consegue prever usando um novo conjunto de amostras.
- Para os treinamentos, uma *epoch* ou época, é quando todo o *dataset* passa por ciclo completo de treinamento na *ANN*. Este ciclo pode conter resultados melhores ou não que na época anterior.
- O *batch size* é o número total de amostras de treinamento presentes em um único lote.
- O *number of iterations*, ou número de iterações, é o número de lotes necessários para completar uma época[2].
- A *Cross Entropy (CE)* ou Entropia Cruzada, é um método de otimização baseado em uma abordagem geral de Monte-Carlo para otimização combinatória e de amostragem de importância. Permite que a rede avalie

[2]Por exemplo, no caso hipotético de 1000 amostras de treinamento e o tamanho do lote definido como 500, seriam necessárias 2 iterações para completar 1 época

pequenos erros e os elimine. Pode ajudar a reduzir o *overfitting*.

- A *Confusion Matrix* ou Matriz de Confusão é uma tabela com duas linhas e duas colunas que relata o número de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos.
- Os métodos iterativos de otimização, *Gradient Descent* ou Gradiente Descendente e *Stochastic gradient descent* ou Gradiente Estocástica Descendente onde, a Gradiente Descendente é um método iterativo de otimização utilizada nos pesos que são atualizados incrementalmente após cada época para buscar o melhor resultado e a Descida Gradiente Estocástica Descendente é um método iterativo para otimizar uma função objetivo.
- *Region of Interest - ROI Pooling* ou Região de Interesse, é a extração e normalização de características de região de interesse. No caso de um espectrograma, um intervalo de tempo ou frequência em uma forma de onda.

2) Aprendizagem Profunda: O *DL* utiliza redes neurais artificiais com várias camadas de abstração, para aplicar *PR* e realizar classificação amparada por conjuntos de dados [41]. Uma forma de prover *DL* é com Redes Neurais Convolucionais - *Convolutional Neural Network (CNN)*. A combinação de camadas convolucionais, realizando a extração de características, e um perceptron multicamadas, realizando a operação de reconhecimento de acordo com os resultados da convolução [42]. O *DL* é a técnica que irá utilizar as *CNN* para prover *ML* ao sistema.

B. Conceitos de Áudio

A forma como a audição humana capta as frequências de som é conhecida como tom. Um som de alta frequência tem um tom mais alto, mais agudo do que um som de baixa frequência. A percepção dos sons de frequências mais baixas é melhor do que a das frequências mais altas.

Uma frequência de 200 *Hertz (Hz)* é o dobro de 100 *Hz*, enquanto a frequência de 10100 *Hz* é apenas 1% mais alta do que a frequência de 10000 *Hz* e, possivelmente não seria notada a diferença entre uma e a outra. A audição humana ouve em uma escala logarítmica ao invés de uma escala linear. Por causa disto, em 1937, Stevens, Volkmann e Newmann propuseram uma unidade de altura chamada *PITCH* [43] tal que distâncias iguais na altura soassem igualmente distantes para o ouvinte. Esta escala foi chamada de escala de mel e, uma operação matemática é realizada nas frequências para convertê-las para esta escala.

Um sinal sonoro é produzido por variações na pressão do ar. Pode-se medir a intensidade das variações de pressão e traçar essas medições ao longo do tempo. Os sinais sonoros costumam se repetir em intervalos regulares para que cada onda tenha a mesma forma. A altura mostra a intensidade do som e é conhecida como amplitude. O tempo que leva para o sinal completar uma onda completa é o período. O número de ondas feitas pelo sinal em um segundo é chamado de frequência, a unidade de frequência é o *Hz*. A Figura 1 mostra uma onda sonora.

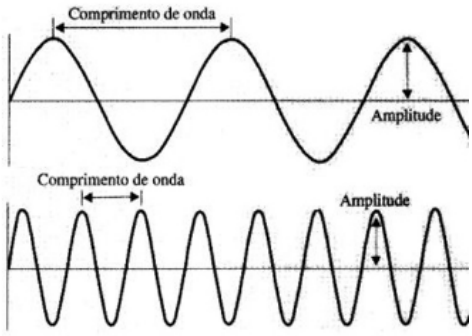

Fig. 1. Amplitude e Comprimento da onda.Fonte: [44]

Para digitalizar uma onda sonora, o sinal é transformado em uma série de números medindo a amplitude do som em intervalos fixos de tempo. As ondas sonoras são digitalizadas por amostragem em intervalos discretos conhecidos como taxa de amostragem. Uma frequência de 44.100 *Hz*, significa que as amostras foram obtidas 44.100 vezes por segundo. Frequência comum em arquivos de áudio digital utilizando o formato *.wav*. Cada uma destas amostras é a amplitude da onda em um determinado intervalo de tempo, e a quantidade de *bits* determina a qualidade sonora, normalmente 16 bits, o que significa que uma amostra pode variar a amplitude 65.536 posições ou seja, (2^6).

No processamento de sinais, a amostragem é a redução de um sinal contínuo em uma série de valores discretos. A frequência ou taxa de amostragem é o número de amostras obtidas em um determinado período de tempo. Quanto mais alta frequência de amostragem, menos perda de informação e maior gasto computacional. Para baixas taxas de amostragem, maior perda de informação, mas mais rápidas e baratas de calcular.

O sinal sonoro capturado como uma forma de onda pode ser interpretado, modificado e analisado. Para ser analisado, o som é transformado em imagem.

1) A imagem do som: A *Fast Fourier Transform (FFT)* é um método de medição importante na ciência da medição de áudio e acústica. A *FFT* é utilizada para converter um sinal em componentes espectrais individuais e fornece informações de frequência sobre o sinal convertendo o sinal do domínio do tempo no domínio da frequência. O resultado é denominado espectro, por exemplo, um espectro do som de um ar condi-

cionado gerado a partir do *dataset urban8k* e disponibilizados nos *colabs* de práticas gerados neste estudo é mostrado na Figura 2.

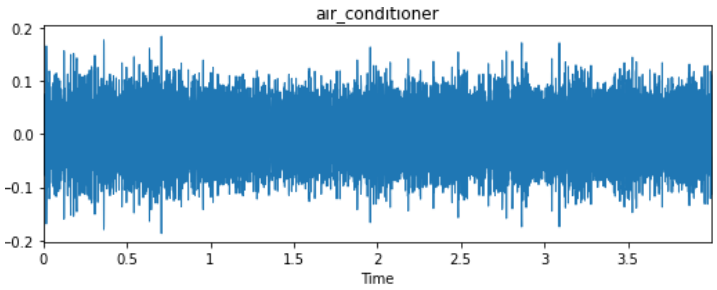

Fig. 2. Espectro do som. Fonte: o autor

A *FFT* converte um sinal de suas frequências componentes, mas perde toda a informação do tempo. Para resolver este problema, a *Short-time Fourier transform (STFT)* divide o sinal em janelas de tempo e executa uma *FFT* em cada janela, preservando as informações tempo. O primeiro a propor este enfoque foi Dennis Gabor, em 1946 [45], com a chamada transformada de Gabor, tamb[em chamada de transformada de Fourier de Janela Deslizante, a *STFT*.

O *STFT* é calculado em segmentos de janela sobrepostos do sinal, desta forma é obtido o espectrograma. O espectrograma pode ser definido como um gráfico que mostra a intensidade por meio do escurecimento ou coloração da região, as faixas de frequência no eixo vertical e o tempo no eixo horizontal. Sendo as estrias horizontais denominadas de harmônicas. Um espectrograma é a impressão digital de um som, e a partir dele, pode se analisar o som como se fosse uma imagem. Um espectrograma é uma representação visual do espectro de frequências de um sinal variando no tempo. A Figura 3 mostra um espectrograma *STFT* de ar condicionado do gerado a partir *dataset urban8k* e disponibilizados nos *colabs* das práticas dirigidas.

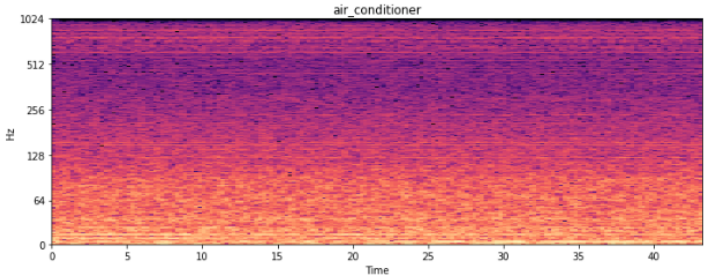

Fig. 3. Espectrograma *STFT* do som de um ar condicionado do *dataset urban8k*. Fonte: *colab* do autor

Para gerar este espectrograma, foi utilizada a *FFT* [46] e a *mel scale*, que é uma escala de tons com distâncias iguais entre um e outro tom. Um *Mel spectrogram* é a representação visual do som na escala *mel* [47].

O *Mel Frequency Cepstral Coefficients (MFCC)* é um espectrograma com um conjunto menor de características do áudio, é baseado nas características extraídas do áudio pela *CNN* [48]. O *MFCC* também é chamado de espectro do espectro, é o espectrograma gerado com as características extraídas do espectrograma original do áudio. Geralmente é gerado com as características extraídas do espectrograma original do áudio por uma *CNN*, com estes dados, é gerado o *MFCC*, como o mostrado na Figura 4, obtido das características do espetrograma de um ar condicionado gerado a partir do *dataset urban8k* e disponibilizados nas práticas da trilha de aprendizagem.

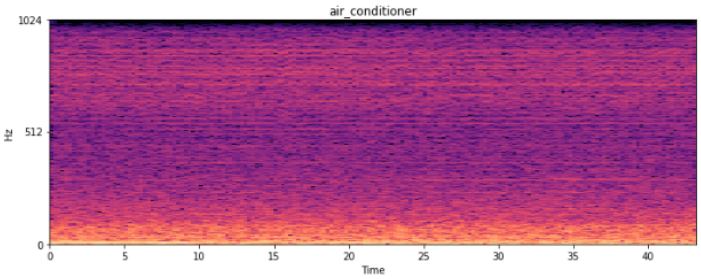

Fig. 4. *MFCC* do som de um ar condicionado do *dataset urban8k*. Fonte: *colab* do autor.

C. Datasets de áudio

Datasets são conjuntos de dados. Esta seção irá apresentar alguns dos *datasets* utilizados em artigos e estudos, provendo algumas informações sobre eles para poder contextualizar e compreender melhor quando eles são citados.

Os *datasets* possuem *labels*, etiquetas, em arquivos separados como um *txt*, *json* ou *csv* para identificar o que cada arquivo representa. Alguns *datasets* ainda separam os conjuntos de dados em conjuntos de treinamento e testes. A proporção costuma ser 80% e 20%, respectivamente.

1) MIMII (sound): Conjunto de dados *MIMII*[3] possui um tamanho aproximado de 100 *Gigabytes (GB)* e é disponibilizado sob a licença livre *Creative Commons*. O conjunto de sons *MIMII* tem objetivo de investigação e inspeção de máquinas industriais com defeito [49].

2) UrbanSound8k: O conjunto de dados *UrbanSound8k*[4], de aproximadamente 6 *GB*, contém 8732 trechos de som

[3]https://zenodo.org/record/3384388#.YVTucLzMKCg
[4]https://urbansounddataset.weebly.com/urbansound8k.html

rotulados, com duração de até 4 segundos, de sons urbanos divididos em 10 classes, sendo elas: *air_conditioner, car_horn, children_playing, dog_bark, drill, enginge_idling, gun_shot, stone_crusher, sirene* e *street_music* [50].

3) ESC-50: O conjunto *ESC-50*[5] é uma coleção de arquivos de som, disponibilizada em um arquivo compactado de aproximadamente 600 *Megabytes* (MB), anotadas, de 2.000 clipes curtos, compreendendo 50 classes. Sendo 40 clipes por classe, com duração de aproximadamente 5 segundos por clipe. Existe uma variação com menos classes que é o *ESC-10*, sendo 10 classes e uma variação chamada *ESC-US* que contém 250.000 sons não rotulados extraídos de gravações disponíveis através do projeto Freesound[6] [51].

4) Xeno-canto: Xeno-canto é um site[7] dedicado a compartilhar sons de pássaros de todo o Mundo e, um *dataset* e também um projeto colaborativo de compartilhamento com intenção de compartilhar o som do canto dos pássaros para colaborar com a conservação deles.

5) GTZan: O conjunto de dados *GTZan*[8] tem tamanho de 1.2 *GB* e consiste em 1000 faixas de áudio, sendo cada uma com 30 segundos de duração. Contém 10 gêneros, sendo cada um representado por 100 faixas. Todas as faixas são arquivos de áudio Mono de 16 bits e 22050 *Hz* no formato *.wav* [52].

6) Speech Commands: O conjunto de dados *Speech Commands*[9], Comandos de Voz, de tamanho 2.37 *GB*, conta com mais de 105.000 arquivos de áudio *.wav* de pessoas falando trinta palavras diferentes divididas em 12 classes. Esses dados foram coletados pelo *Google* e divulgados sob uma licença *Creative Commons*[10] [53] .

7) AudioSet: O *dataset AudioSet*[11], sons das coisas, é uma coleção de clipes de áudio de 10 segundos extraídos de vídeos do *YouTube*[12]. A identificação, anotação, dos áudios foi feita por humanos em 2.084.320 áudios divididos em 527 classes [54].

8) FSDKaggle2018: Freesound Dataset Kaggle 2018, ou *FSDKaggle2018*, é um *dataset* de áudio que contém 11.073 arquivos de áudio anotados com 41 classes ontológicas. O conjunto *FSDKaggle2018* é voltado para detecção e classificação de sons, *Detection and Classification of Acoustic Scenes and Events (DCASE)*.

[5]https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YDEPUT
[6]https://freesound.org/
[7]https://www.xeno-canto.org/
[8]http://opihi.cs.uvic.ca/sound/genres.tar.gz
[9]https://www.tensorflow.org/datasets/catalog/speech_commands
[10]https://creativecommons.org/licenses/by/4.0/deed.pt_BR
[11]https://research.google.com/audioset
[12]https://www.youtube.com/

D. Modelos Pré Treinados

Existe a possibilidade de serem utilizados Modelos de *ML* Pré-Treinados para trabalhar com reconhecimento de áudio. Serão apresentados alguns modelos de *ML* pré-treinados utilizados no reconhecimento de sons.

1) Yet another Audio Mobilenet Network: Yet another Audio Mobilenet Network - YAMNet[13] é um classificador pré-treinado que utiliza uma rede de convolução profunda, Mobilenet_v1 [55], e consegue prever 521 eventos em áudios de 16 *Kilo-Hertz (KHz)*.

2) EnvNet-V2: O modelo *EnvNet-V2*[14] é uma rede neural mais profunda que a sua versão anterior *EnvNet*. A *EnvNet-V2* é utilizada no reconhecimento de sons com mais acerto que humanos [56].

3) Resnet18: ResNet-18 é uma rede neural convolucional com 18 camadas de profundidade para classificação de imagens pré-treinada com mais de um milhão de imagens do banco de dados *ImageNet*[15]. É também utilizada para reconhecimento de sons via reconhecimento dos espectrogramas.

4) ESResNet: ESResNet é uma *CNN* para classificação de sons do ambiente baseada no domínio de imagem, o espectrograma. Os espectrogramas são gerados com *STFT* e avaliados em *cross-domain pre-training* usando *Residual Neural Network (ResNet)*, *Siamese Neural Network* e outras [57].

5) Self-Supervised Pitch Estimation : O modelo *Self-Supervised Pitch Estimation (SPICE)* é um codificador convolucional que mapeia um som de 16.000 *Hz*, mono, para uma nota musical, tom. O tom de um som é uma medida qualitativa da frequência do som. Um som com um tom alto tem uma frequência mais alta do que um som baixo [58].

6) vggish: O modelo pré-treinado *vggish*[16] é implementado em *TensorFlowKeras* e, foi treinado com o *YouTube-8M dataset* [59] para realizar a detecção de eventos de áudio.

E. Classificação de Áudio

Classificação de Áudio - *Audio Classification*, significa aprender diversas classes de sons e depois classificar um som em alguma destas classes.

Separação ou Segmentação de Áudio - *Audio Separation and Segmentation* é separar ou isolar algum som específico. Por exemplo, no meio de outros ruídos, separar a voz de uma pessoa específica, ou no meio de um musica, separar o som apenas de um instrumento[17]. Por exemplo, ouvir e isolar o som de um

[13]https://tfhub.dev/google/yamnet/1
[14]https://github.com/mohaimenz/EnvNet-V2
[15]http://www.image-net.org
[16]https://tfhub.dev/google/vggish/1
[17]http://sound-of-pixels.csail.mit.edu/

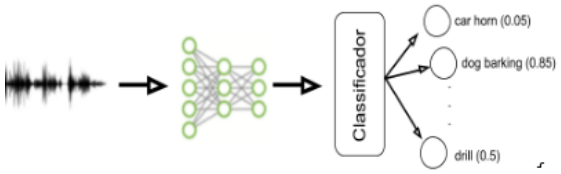

Fig. 5. Classificação de áudio. Fonte: o autor.

coração humano ou da respiração para identificar anomalias, como em [60] ou [61].

Classificação de Gênero musical - *Music Genre Classification and Tagging* é identificar e classificar uma música baseada no áudio, com o que ele se parece, rock, jazz, vocal feminino, música alegre, entre outra possíveis classes.

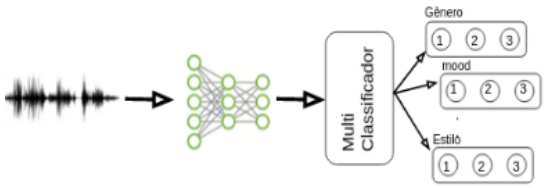

Fig. 6. Multi Classificação de áudio. Fonte: o autor.

Reconhecimento de voz - *Voice Recognition* é identificar o sentimento, o sexo ou a pessoa específica que está falando. Utilizado nos assistentes pessoais para saber quem é o usuário.

Fala para texto ou texto para fala - *Speech to Text and Text to Speech* Utilizado ou para transformar a fala em texto para executar comandos ou o texto em fala para utilizar em sintetizador de voz.

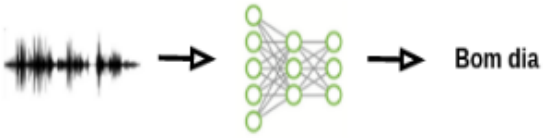

Fig. 7. Reconhecimento de fala. Fonte: o autor.

III. Tecnologias utilizadas

Apresentação das tecnologias e ferramentas para a implementação de projetos com o uso de som será apresentado para conhecer alguns *softwares* que darão apoio na implementação destas atividades. Inicialmente, o ambiente de desenvolvimento - *Integrated Development Environment (IDE)* será o *Google Colaboratory* ou apenas *Colab*[18]. O motivo da escolha é por dar suporte a todo o arcabouço de ferramentas necessárias e ser prático de utilizar, permitir documentar dentro do código e compartilhar facilmente o ambiente.

A. Python e bibliotecas de apoio

A linguagem de programação base será o *Python*, com apoio de outras bibliotecas complementares que serão utilizadas para extrair características, criar modelos e trabalhar com sons. O *Python* é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991 [62]. O *Python* é muito utilizado para trabalhar com *AI* e imagens. Como complementos serão utilizadas as bibliotecas:

- *NumPy*[19], uma das bibliotecas de computação numérica utilizada em processamento de imagens, finanças, bioinformática e outras.
- *SciPy*[20], uma biblioteca para computação científica, inclui suporte para álgebra linear, matrizes esparsas, estruturas de dados espaciais, estatísticas e outros.
- *Matplotlib*[21], uma biblioteca para criação de gráficos e visualizações de dados.
- *Scikit-learn*[22], uma biblioteca de *ML* de código aberto. Permite realizar classificação, regressão, clusterização, seleção de modelos e outras atividades de aprendizagem de máquina.
- *TensorFlow*[23], uma biblioteca de código aberto para *ML* aplicável a uma ampla variedade de tarefas. É um sistema para criação e treinamento de redes neurais para detectar e decifrar padrões e correlações, análogo à forma como humanos aprendem e raciocinam.
- *Keras*[24], uma *Application Programming Interface - API* de rede neural de código aberto que roda em cima do *TensorFlow* para permitir experimentação rápida com redes neurais profundas e, se concentra em ser fácil de usar, modular e extensível [63], [64]. O *Keras* implementa o "Otimizador Adam", *optimizer = tf.keras.optimizers.Adam*, um método de descida de gradiente estocástico baseado na estimação adaptativa de momentos de primeira e segunda ordem. Faz com que a taxa de aprendizado seja variável para cada parâmetro garantindo convergir, mesmo se a amostra de entrada não for linearmente separável, para um mínimo da função de erro. É usado, entre outros, para evitar o *overfitting*.

[18]https://colab.research.google.com/

[19]https://numpy.org/
[20]https://scipy.org/
[21]https://matplotlib.org/
[22]https://scikit-learn.org/stable/
[23]https://www.tensorflow.org/
[24]https://keras.io/

B. Bibliotecas de Áudio

O *pipeline* típico de processamento de áudio envolve a extração de características acústicas relevantes para a tarefa em questão, seguido por esquemas de tomada de decisão que envolvem detecção, classificação e fusão de conhecimento. O *Python* disponibiliza bibliotecas úteis que tornam a tarefa de manipulação de áudio mais fácil. A seguir, algumas destas bibliotecas:

- *IPython*[25], provê um conjunto de ferramentas interativas para manipulação de som. Um destas ferramentas é o método *IPython.display.Audio*, que é utilizado para reproduzir áudio direto no *frontend* do *Jupyter Notebook* [26] ou *colab*;
- *pydub*[27] é uma biblioteca simples de manipulação de áudio, capaz de reproduzir, obter informações, aumentar ou reduzir volume, unir áudios, cortar ou exportar áudios;
- a biblioteca *librosa*[28], provê funcionalidades para trabalhar com áudio [65], [66] e será uma das mais importantes;
- *pyAudioAnalysis*[29] uma biblioteca de código aberto que oferece uma ampla gama de procedimentos de análise de áudio, incluindo: extração de características de áudio, classificação de sinais de áudio, segmentação supervisionada e não supervisionada e visualização de conteúdo. [67].

C. Data Augmentation

Data Augmentation ou aumento de dados, é uma técnica utilizada para incrementar a diversidade de um *dataset* de forma artificial. É utilizada principalmente quando não se têm dados suficientes. Na área de imagens, o que é feito é rotacionar, escalonar, modificar cores, iluminação, ruídos nas imagens que já são identificadas, mas transformar imagens do espectrograma com rotação, seja horizontal ou vertical iria alterar o som que este espectrograma representa. Os métodos para aumentar os espectrogramas seriam:

- máscara de frequência - máscara aleatoriamente uma faixa de frequências consecutivas adicionando barras horizontais no espectrograma (método *tfio.audio.freq_mask*);
- Máscara de tempo - semelhante às máscaras de frequência, mas são selecionados aleatoriamente as faixas de tempo do espectrograma usando barras verticais (método *tfio.audio.time_mask*);
- *Time Stretch* ou aumento de tempo, randomicamente, aumentar ou diminuir a velocidade do áudio;
- *Pitch Shift* — modificar aleatoriamente as pausas do som, aumentando ou diminuindo;
- *Time Shift* — mudança no tempo: Deslocamento randômico do áudio para direita ou esquerda. Para sons contínuos (que se repetem), não faz diferença deslocar um pouco para um ou outro lado;
- *Add Noise* — adição de ruídos: adicionar ruídos aleatórios ao som existente;

O pacote *tensorflow-io* fornece aumentos de espectrograma avançados, como os *Frequency*, *Time Masking* e outros.

IV. Prática

A proposta desse trabalho, é inserir os conceitos necessários para trabalhar com som e mostrar um exemplo prático após a contextualização para utilizar as ferramentas e ver o resultado. Para o reconhecimento e classificação de sons, basicamente a metodologia utilizada é uma *CNN* para realizar a extração de características, *features*, baseada na imagem do espectrograma do som [68].

O *colab* nomeado como ”Iniciando com áudio no python”, disponível *link*[30], mostra como importar as bibliotecas necessárias, importar um arquivo de áudio, criar som, reproduzir, gerar o gráfico de sinais, gerar o espectrograma. Neste trecho é mostrado como instalar e importar a biblioteca necessária, foi carregado um som aleatório disponível online, exibidos alguns dados dele e executado o áudio para ouvir diretamente no ambiente de programação *online* do *colab*.

```
!pip install librosa #instalando a biblioteca
import librosa #importando

#download cat sound (exemplo)
!curl -O https://storage.googleapis.com/audioset/miaow_16k.wav
x, sr = librosa.load('/content/miaow_16k.wav')
#Exibir o tamanho do array do audio e taxa de amostragem
print(x.shape, sr) #(148422, 22050)

#Playing Audio: executar o audio direto no colab
import IPython.display as ipd
ipd.Audio('/content/miaow_16k.wav')
```

Listing 1. Básico do som no *Python*

Este trecho de código irá plotar o sinal como o mostrado na Figura 2.

```
#Visualizing Audio
%matplotlib inline
import matplotlib.pyplot as plt
import librosa.display

#Plotar a onda do audio com librosa.display.waveplot:
plt.figure(figsize=(14, 5))
```

[25]https://ipython.org/
[26]https://jupyter.org/
[27]http://pydub.com/
[28]https://librosa.org/
[29]https://github.com/tyiannak/pyAudioAnalysis
[30]https://bityli.com/PcstRoS

```
    librosa.display.waveplot(x, sr=sr)
    plt.title('Signal');
    plt.xlabel('Time (samples)');
    plt.ylabel('Amplitude');
```

Listing 2. Espectro do som - Sinais

Este trecho de código irá plotar o espectrograma como o mostrado na Figura 3.

```
    #mostrar o espectrograma com a librosa.display.
    specshow:
    X = librosa.stft(x)
    Xdb = librosa.amplitude_to_db(abs(X))
    plt.figure(figsize=(14, 5))
    librosa.display.specshow(Xdb, sr=sr, x_axis='time
    ', y_axis='hz')
```

Listing 3. Espectrograma

Este outro *colab* chamado "exemplo prático *YAMNet*" disponível no *link*[31], mostra como importar o modelo *YAMNet*, apresentado anteriormente, e o utiliza para tentar classificar um som, neste caso, um miado de gato que foi importado do *dataset audioset*.

```
    #carregando o modelo yamnet no Python:
    import tensorflow\_hub as hub
    model=hub.load('https://tfhub.dev/google/yamnet/1
    ')
```

Listing 4. utilizando a rede YAMNet para classificar um áudio

A Figura 8 mostra o sinal do som, o espectrograma do miado e uma matriz com o *score* obtido pelo som em cada uma das classes possíveis.

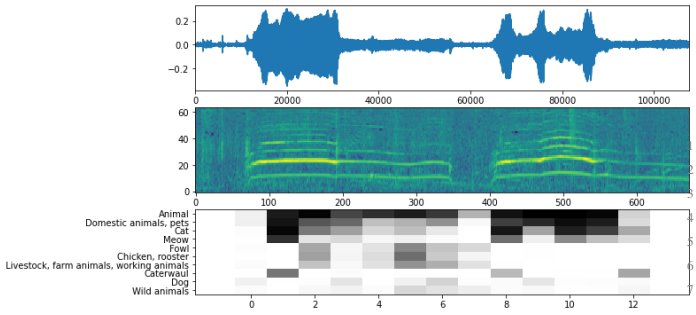

Fig. 8. Resultado *YAMNet*. Fonte: *colab* do autor

Neste exemplo, o miado foi classificado apenas como sendo som da classe "animal". O modelo não conseguiu ser específico o suficiente para chegar na classe "animal doméstico" que já seria amis próxima nem na classe "gato" que seria a mais específica. Para mais detalhes deste projeto, pode ser acessado o *colab* e acompanhado passo à passo todo o processo documentado.

[31]https://bityli.com/sdeTXGW

Com objetivo de melhorar este resultado, o *colab* do link[32] utilizou a rede pré treinada *YAMNet* e o *dataset ESC-50* sendo que a *YAMNet* foi utilizada apenas como um extrator de características e do *dataset ESC-50*, foram utilizados apenas os sons de gatos e cachorros.

Então, foi criado um Modelo Sequential com uma camada escondida e com apenas duas classes resultado, que classifica o som como sendo de cães ou gatos a partir do arquivo de som avaliado.

```
#criando modelo sequencial com 2 classes
my_model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(1024), dtype=tf.
    float32, name='input_embedding'),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(len(my_classes))], name='
    my_model')
my_model.summary()

#definindo a loss function
my_model.compile(loss=tf.keras.losses.
    SparseCategoricalCrossentropy(from_logits=True),
                optimizer="adam",
                metrics=['accuracy'])

#parar treinamento quando parar de melhorar
callback = tf.keras.callbacks.EarlyStopping(monitor='
    loss', patience=3,   restore_best_weights=True)

#treinando
history = my_model.fit(train_ds,
                       epochs=20,
                       validation_data=val_ds,
                       callbacks=callback)
```

Listing 5. Melhorando a classificação

Após criada e treinada a rede com sons apenas de gatos e cachorros para serem separados em duas classes, testar os parâmetros da rede resultante. A perda foi de 0,34 e a acurácia de 0,88.

```
#verificando \textit{overfitting}
loss, accuracy = my_model.evaluate(test_ds)
print("Loss: ", loss) # 0.3498896062374115
print("Accuracy: ", accuracy) #0.831250011920929

#testando o modelo
scores, embeddings, spectrogram = yamnet_model(
    testing_wav_data)
result = my_model(embeddings).numpy()

inferred_class = my_classes[result.mean(axis=0).
    argmax()]
print(f'The main sound is: {inferred_class}') # cat
```

Listing 6. Testando

Utilizando as características extraídas pela *YAMNet* e utilizando como entrada na rede específica treinada para avaliar apenas cães e gatos, o miado foi classificado corretamente na classe "gato".

[32]https://bityli.com/agmPseg

V. Conclusão

Este artigo teve como objetivo realizar um levantamento bibliográfico para conhecer os conceitos fundamentais, técnicas utilizadas para o reconhecimento de sons com *AI* e a linguagem de programação *Python*. Introduziu desde a parte teórica inicial do som at[e o reconhecimento destes sons por uma *AI*, apresentando o arcabouço necessário e em seguida mostrou exemplos práticos. Todos os códigos-fontes podem ser acessados e executados no ambiente interativo do *google colab* que permite uma aprendizagem dirigida. A contribuição esperada foi a de prover conhecimento inicial para trabalhos de reconhecimento de sons e ir avançando na pesquisa do som como fonte de dados, com o objetivo de extrair informações do som captado para compreender padrões (*audio sensing*). A análise de som é uma tarefa desafiadora e associada a diversas aplicações. O objetivo final é compreender os sons para compreeder o ambiente e gerar informações úteis para viabilizadar tomadas de decisão baseadas em dados. Com isto, contribuindo para a área de pesquisa do som específico que esta sendo analisado.

Agradecimentos

Este trabalho tem apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Referências

[1] Y. Alsouda, S. Pllana, and A. Kurti, "Iot-based urban noise identification using machine learning: performance of svm, knn, bagging, and random forest," in *Proceedings of the international conference on omni-layer intelligent systems*, 2019, pp. 62–67.

[2] L. Lhoest, M. Lamrini, J. Vandendriessche, N. Wouters, B. da Silva, M. Y. Chkouri, and A. Touhafi, "Mosaic: A classical machine learning multi-classifier based approach against deep learning classifiers for embedded sound classification," *Applied Sciences*, vol. 11, no. 18, p. 8394, 2021.

[3] B. da Silva, A. W Happi, A. Braeken, and A. Touhafi, "Evaluation of classical machine learning techniques towards urban sound recognition on embedded systems," *Applied Sciences*, vol. 9, no. 18, p. 3885, 2019.

[4] Y. Alsouda, S. Pllana, and A. Kurti, "A machine learning driven iot solution for noise classification in smart cities," 2018.

[5] J. Segura-Garcia, S. Felici-Castell, J. J. Perez-Solano, M. Cobos, and J. M. Navarro, "Low-cost alternatives for urban noise nuisance monitoring using wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 836–844, 2014.

[6] J. Ye, T. Kobayashi, and T. Higuchi, "Smart audio sensor on anomaly respiration detection using flac features," in *2012 IEEE Sensors Applications Symposium Proceedings*. IEEE, 2012, pp. 1–5.

[7] A. A. Mahmoud, I. N. A. Alawadh, G. Latif, and J. Alghazo, "Smart nursery for smart cities: Infant sound classification based on novel features and support vector classifier," in *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*, 2020, pp. 47–52.

[8] S. K. Shah, Z. Tariq, and Y. Lee, "Iot based urban noise monitoring in deep learning using historical reports," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 4179–4184.

[9] R. E. Hall, B. Bowerman, J. Braverman, J. Taylor, H. Todosow, and U. Von Wimmersperg, "The vision of a smart city," Brookhaven National Lab., Upton, NY (US), Tech. Rep., 2000.

[10] J. P. Bello, C. Mydlarz, and J. Salamon, "Sound analysis in smart cities," in *Computational Analysis of Sound Scenes and Events*. Springer, 2018, pp. 373–397.

[11] J. Svatos and J. Holub, "Smart acoustic sensor," in *2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI)*. IEEE, 2019, pp. 161–165.

[12] Q. Mei, M. Gül, and M. Boay, "Indirect health monitoring of bridges using mel-frequency cepstral coefficients and principal component analysis," *Mechanical Systems and Signal Processing*, vol. 119, pp. 523–546, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327018306678

[13] R. K. Gunupudi, M. Nimmala, N. Gugulothu, and S. R. Gali, "Clapp: A self constructing feature clustering approach for anomaly detection," *Future Generation Computer Systems*, vol. 74, pp. 417–429, 2017.

[14] A. R. Hilal, A. Sayedelahl, A. Tabibiazar, M. S. Kamel, and O. A. Basir, "A distributed sensor management for large-scale iot indoor acoustic surveillance," *Future Generation Computer Systems*, vol. 86, pp. 1170–1184, 2018.

[15] Z. Tariq, S. K. Shah, and Y. Lee, "Speech emotion detection using iot based deep learning for health care," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 4191–4196.

[16] J. Chin, A. Tisan, V. Callaghan, and D. Chik, "Smart-object-based reasoning system for indoor acoustic profiling of elderly inhabitants," *Electronics*, vol. 10, no. 12, p. 1433, 2021.

[17] A. W. Ramadhan, A. Wijayanto, and H. Oktavianto, "Implementation of audio event recognition for the elderly home support using convolutional neural networks," in *2020 International Electronics Symposium (IES)*. IEEE, 2020, pp. 91–95.

[18] L. Gantert, M. Sammarco, M. Detyniecki, M. Elias, and M. Campista, "A supervised approach for corrective maintenance using spectral features from industrial sounds," in *IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021.

[19] R. Müller, F. Ritz, S. Illium, and C. Linnhoff-Popien, "Acoustic anomaly detection for machine sounds based on image transfer learning," in *ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, vol. 2. SciTePress, 2021, pp. 49–56.

[20] J. Sikora, R. Wagnerová, L. Landryová, J. Šíma, and S. Wrona, "Influence of environmental noise on quality control of hvac devices based on convolutional neural network," *Applied Sciences*, vol. 11, p. 7484, 8 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7484

[21] E. Browning, R. Gibb, P. Glover-Kapfer, and K. E. Jones, "Passive acoustic monitoring in ecology and conservation." *WWF Conservation Tecnology Series*, 2017.

[22] S. S. Sethi, R. M. Ewers, N. S. Jones, C. D. L. Orme, and L. Picinali, "Robust, real-time and autonomous monitoring of ecosystems with an open, low-cost, networked device," *Methods in Ecology and Evolution*, vol. 9, no. 12, pp. 2383–2387, 2018.

[23] B. Krause and A. Farina, "Using ecoacoustic methods to survey the impacts of climate change on biodiversity," *Biological conservation*, vol. 195, pp. 245–254, 2016.

[24] A. J. Fairbrass, M. Firman, C. Williams, G. J. Brostow, H. Titheridge, and K. E. Jones, "Citynet—deep learning tools for urban ecoacoustic assessment," *Methods in ecology and evolution*, vol. 10, no. 2, pp. 186–197, 2019.

[25] A. Farina and S. H. Gage, *Ecoacoustics: The ecological role of sounds*. John Wiley & Sons, 2017.

[26] A. Zgank, "Bee swarm activity acoustic classification for an iot-based farm service," *Sensors*, vol. 20, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/21

[27] D. Vasconcelos, M. S. Yin, F. Wetjen, A. Herbst, T. Ziemer, A. Förster, T. Barkowsky, N. Nunes, and P. Haddawy, "Counting mosquitoes in the wild: An internet of things approach," in *Proceedings of the Conference on Information Technology for Social Good*, 2021, pp. 43–48.

[28] S. Thangavel and C. S. Shokkalingam, "The iot based embedded system for the detection and discrimination of animals to avoid human–wildlife

conflict," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–17, 2021.

[29] Y. B. Ouattara, T. A. Kobea, G. Baudoin, J.-M. Laheurte *et al.*, "Knn and svm classification for chainsaw identification in the forest areas," *International journal of advanced computer science and applications (IJACSA)*, vol. 10, no. 12, 2019.

[30] B. Holgate, R. Maggini, and S. Fuller, "Mapping ecoacoustic hot spots and moments of biodiversity to inform conservation and urban planning," *Ecological Indicators*, vol. 126, p. 107627, 2021.

[31] C. C. Constantinou, E. Michaelides, I. Alexopoulos, T. Pieri, S. Neophytou, I. Kyriakides, E. Abdi, J. Reodica, and D. R. Hayes, "Modeling the operating characteristics of iot for underwater sound classification," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021, pp. 1016–1022.

[32] M. Antonini, M. Vecchio, F. Antonelli, P. Ducange, and C. Perera, "Smart audio sensors in the internet of things edge for anomaly detection," *IEEE Access*, vol. 6, pp. 67 594–67 610, 2018.

[33] idceurope.com, "Why is sound recognition a key strategic technology for artificial intelligence," Dezembro 2019. [Online]. Available: https://blog-idceurope.com/sound-recognition-as-a-key-strategic-technology-for-artificial-intelligence/

[34] hellofuture, "Ai breaks the sound barrier: Sound recognition remains a relatively unexplored field of artificial intelligence," Dezembro 2020. [Online]. Available: https://hellofuture.orange.com/en/ai-breaks-the-sound-barrier/

[35] IBM, "Sound as a new data source for industry 4.0," DEZEMBRO 2021. [Online]. Available: HTTPS://WWW.IBM.COM/BLOGS/SERVICES/2021/05/03/SOUND-AS-A-NEW-DATA-SOURCE-FOR-INDUSTRY-4-0/

[36] gartner, "Gartner top strategic tech trends for 2021: Gartner's new ebook highlights trends, like internet of things (iot) edge cloud, that will define the future of it." Dezembro 2020. [Online]. Available: https://www.gartner.com/en/information-technology/trends/top-strategic-technology-trends-iot-gb-pd?utm_source=google&utm_medium=cpc&utm_campaign=RM_NA_2020_ITTRND_CPC_LG1_2021-TSTT-GB-PD&utm_adgroup=117569667994&utm_term=2Biot&ad=486441735318&matchtype=b&gclid=CjwKCAjwqeWKBhBFEiwABo_XBjyXYzulfy55ofo3v_sj1dJHytBLvCQl__0OANsZb6j4M8EaQhiZdhoCDgkQAvD_BwE

[37] J. McCarthy, "Artificial intelligence, logic and formalizing common sense," in *Philosophical logic and artificial intelligence*. Springer, 1989, pp. 161–190.

[38] T. Mitchell, *Machine learning*. McGraw hill Burr Ridge, 1997.

[39] X.-D. Zhang, "Machine learning," in *A Matrix Algebra Approach to Artificial Intelligence*. Springer, 2020, pp. 223–440.

[40] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[41] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[42] M. V. Valueva, N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020.

[43] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.

[44] M. N. VIEIRA. (2004) Acústica - princípios da produção e análise da voz. [Online]. Available: http://www.cefala.org/fonologia/acustica

[45] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.

[46] P. Heckbert, "Fourier transforms and the fast fourier transform (fft) algorithm," *Computer Graphics*, vol. 2, pp. 15–463, 1995.

[47] S. A. Majeed, H. Husain, S. A. Samad, and T. F. Idbeaa, "Mel frequency cepstral coefficients (mfcc) feature extraction enhancement in the application of speech recognition: a comparison study," *Journal of theoretical and applied information technology*, vol. 79, no. 1, p. 38, 2015.

[48] S. Li, H. Kim, S. Lee, J. C. Gallagher, D. Kim, S. Park, and E. T. Matson, "Convolutional neural networks for analyzing unmanned aerial vehicles sound," in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, 2018, pp. 862–866.

[49] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," *arXiv preprint arXiv:1909.09347*, 2019.

[50] A. A. Rahman and J. Angel Arul Jothi, "Classification of urbansound8k: A study using convolutional neural network and multiple data augmentation techniques," in *International Conference on Soft Computing and its Engineering Applications*. Springer, 2020, pp. 52–64.

[51] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[52] B. L. Sturm, "The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use," *arXiv preprint arXiv:1306.1461*, 2013.

[53] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[54] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[55] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[56] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.

[57] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Esresnet: Environmental sound classification based on visual domain models," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4933–4940.

[58] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "Spice: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.

[59] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[60] S. O. Folorunso, E. Ogbuju, and F. Oladipo, "Artificial intelligence and the control of covid-19: A review of machine and deep learning approaches," *Artificial Intelligence for COVID-19*, pp. 167–185, 2021.

[61] X. Huai, S. Kitada, D. Choi, P. Siriaraya, N. Kuwahara, and T. Ashihara, "Heart sound recognition technology based on convolutional neural network," *Informatics for Health and Social Care*, pp. 1–13, 2021.

[62] "Python," 2021. [Online]. Available: https://www.python.org/

[63] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," 2017.

[64] "Tensorflow," 2021. [Online]. Available: https://www.tensorflow.org/

[65] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[66] "librosa," 2021. [Online]. Available: https://librosa.org/doc/latest/index.htm

[67] T. Giannakopoulos, "pyaudioanalysis: An open-source python library for audio signal analysis," *PloS one*, vol. 10, no. 12, p. e0144610, 2015.

[68] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*. IEEE, 2015, pp. 1–6.