



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

GABRYELLE SOARES HERCULANO DE ALMEIDA

**RECONHECIMENTO DE INSTRUMENTOS MÚSICAIS
UTILIZANDO MACHINE LEARNING: UM ESTUDO DE CASO**

CAMPINA GRANDE - PB

2023

GABRYELLE SOARES HERCULANO DE ALMEIDA

**RECONHECIMENTO DE INSTRUMENTOS MUSICAIS
UTILIZANDO MACHINE LEARNING: UM ESTUDO DE CASO**

**Trabalho de Conclusão de Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

Orientador: Leandro Balby Marinho

CAMPINA GRANDE - PB

2023

GABRYELLE SOARES HERCULANO DE ALMEIDA

**RECONHECIMENTO DE INSTRUMENTOS MUSICAIS
UTILIZANDO MACHINE LEARNING: UM ESTUDO DE CASO**

**Trabalho de Conclusão de Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande como requisito parcial para a
obtenção do título de Bacharela em
Ciência da Computação.**

BANCA EXAMINADORA:

Leandro Balby Marinho

Orientador – UASC/CEEI/UFCG

Marcelo Alves de Barros

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 28 de junho de 2023.

CAMPINA GRANDE - PB

RESUMO

A área de Recuperação de Informações Musicais (Music Information Retrieval ou MIR) engloba uma variedade de tópicos, incluindo a transcrição musical, a separação de fontes sonoras, o reconhecimento de instrumentos e/ou gêneros musicais. Um exemplo prático de um desses campos é o Spotify, que utiliza sistemas de recomendação capazes de aprender o padrão de conteúdo reproduzido e sugere aos usuários músicas similares. No entanto, o reconhecimento de instrumentos ainda pode ser desafiador de acordo com o conjunto de dados utilizado, dificultando o reconhecimento de padrões. Nesse contexto, essa pesquisa tem como objetivo treinar um modelo capaz de detectar e identificar instrumentos, além de avaliá-lo em diferentes conjuntos de dados amplamente conhecidos na área de MIR. Para isso, foram utilizados áudios do OpenMIC-2018 no treinamento e os modelos foram avaliados em três conjuntos de dados, sendo estes: MTG-Jamendo, NSynth e áudios de apresentações ao vivo com instrumentos separados utilizando o Demucs. A acurácia será um dos critérios utilizados para avaliar o desempenho do modelo. Ao abordar essa problemática, espera-se contribuir para avanços na área de MIR, permitindo recomendações musicais mais personalizadas por meio do aprimoramento da precisão em sistemas de recomendação. Além disso, deseja-se fornecer insights para a comunidade de MIR, auxiliando na análise musical e em campos relacionados, a fim de permitir aplicações cada vez mais eficientes.

RECOGNITION OF MUSICAL INSTRUMENTS USING MACHINE LEARNING: A CASE STUDY

ABSTRACT

The field of Music Information Retrieval (MIR) encompasses a variety of topics, including music transcription, source separation, and recognition of instruments and/or musical genres. A practical example of this field is Spotify, which uses recommendation systems capable of learning patterns of the reproduced content and suggesting similar music to users. However, instrument recognition can still present challenges depending on the dataset used, making pattern recognition more difficult. In this context, the objective of this research is to train a model capable of detecting and identifying instruments, and evaluate this model on different well-known datasets in the field of MIR. For this purpose, the OpenMIC-2018 dataset is used for training and three datasets to evaluate the model: MTG-Jamendo, NSynth and audio from live performances with separated instruments using Demucs. Accuracy will be one of the criteria used to assess the model's performance. By addressing this issue, we hope to contribute to advancements in the field of MIR, enabling more personalized music recommendations through improved accuracy in recommendation systems. Additionally, we seek to provide insights to the MIR community, aiding in music analysis and related fields, in order to facilitate increasingly efficient applications.

Reconhecimento de Instrumentos Musicais Utilizando Machine Learning: um estudo de caso

Gabryelle Soares Herculano de Almeida
(orientanda)

Universidade Federal de Campina Grande
Campina Grande, Brasil

gabryelle.almeida@ccc.ufcg.edu.br

Leandro Balby Marinho
(orientador)

Universidade Federal de Campina Grande
Campina Grande, Brasil

lbmarinho@computacao.ufcg.edu.br

RESUMO

A área de Recuperação de Informações Musicais (*Music Information Retrieval* ou MIR) engloba uma variedade de tópicos, incluindo a transcrição musical, a separação de fontes sonoras, o reconhecimento de instrumentos e/ou gêneros musicais. Um exemplo prático de um desses campos é o Spotify, que utiliza sistemas de recomendação capazes de aprender o padrão de conteúdo reproduzido e sugere aos usuários músicas similares. No entanto, o reconhecimento de instrumentos ainda pode ser desafiador de acordo com o conjunto de dados utilizado, dificultando o reconhecimento de padrões. Nesse contexto, essa pesquisa tem como objetivo treinar um modelo capaz de detectar e identificar instrumentos, além de avaliá-lo em diferentes conjuntos de dados amplamente conhecidos na área de MIR. Para isso, foram utilizados áudios do OpenMIC-2018¹ no treinamento e os modelos foram avaliados em três conjuntos de dados, sendo estes: MTG-Jamendo², NSynth³ e áudios de apresentações ao vivo com instrumentos separados utilizando o Demucs⁴. A acurácia será um dos critérios utilizados para avaliar o desempenho do modelo. Ao abordar essa problemática, espera-se contribuir para avanços na área de MIR, permitindo recomendações musicais mais personalizadas por meio do aprimoramento da precisão em sistemas de recomendação. Além disso, deseja-se fornecer insights para a comunidade de MIR, auxiliando na análise musical e em campos relacionados, a fim de permitir aplicações cada vez mais eficientes.

Palavras-chave

Reconhecimento de instrumentos, sistemas de recomendação, *Machine Learning*.

1. INTRODUÇÃO

Quando se trata de reconhecimento de instrumentos, existem diversas maneiras de realizar essa tarefa. Uma maneira comum de classificar os dados é através de metadados, onde são definidas tags como artista, álbum, ano de lançamento, entre outras; e os usuários preenchem tais informações. No entanto, esse método manual de classificação pode apresentar problemas de categorização, como divergências nas classificações entre diferentes grupos de pessoas e dificuldade em identificar alguns rótulos. Além disso, o processo manual é lento, o que se torna um desafio diante do grande volume de dados a serem rotulados [10]. Como resultado, a automação se torna essencial nesse contexto,

de forma a oferecer mais organização e precisão, especialmente diante da quantidade crescente de dados gerados diariamente, o que implica em uma demanda cada vez maior para rotulá-los. Ao automatizar o processo, além da otimização do tempo necessário para a classificação desse grande volume de dados, evitam-se erros causados por humanos e, dessa forma, consegue-se aprimorar a recomendação de conteúdo musical.

Dito isso, sistemas de recomendação são uma subclasse de *Machine Learning* e podem ser definidos como sistemas que tentam prever novos conteúdos por meio de técnicas algorítmicas. O objetivo de um sistema de recomendação, independentemente de onde é utilizado, é facilitar as operações do usuário. Para isso, existem duas arquiteturas básicas: recomendações baseadas em conteúdo, onde o sistema recomenda conteúdos com base nas interações do usuário, como pesquisas, compras, tempo gasto visualizando uma página, entre outros; e recomendações com filtragem colaborativa, que são determinadas com a colaboração de múltiplos agentes presentes no sistema, como através de interesses similares de dois ou mais usuários. Essas recomendações podem ser realizadas de forma explícita, como o ranqueamento de itens, ou de forma implícita, a partir de históricos de pesquisa [9].

Com o avanço da tecnologia, os usuários esperam cada vez mais rapidez e sistemas que sejam capazes de responder a perguntas antes mesmo de serem feitas. Nesse sentido, este trabalho tem o intuito de analisar e avaliar a acurácia de um modelo de reconhecimento de instrumentos em conjuntos de dados distintos.

A seguir, serão abordados os conceitos fundamentais do som e do áudio, juntamente com a sua representação na computação. Em seguida, na seção 3, serão descritos os datasets utilizados neste trabalho: o dataset OpenMIC-2018, detalhado na subseção 3.1, o dataset DEMUCS na subseção 3.2, o dataset MTG-Jamendo na subseção 3.3, e, por fim, o dataset NSynth na subseção 3.4. Na seção 4, será explicada a metodologia utilizada para realizar os treinamentos, análises e experimentos. Os resultados obtidos serão apresentados e discutidos na seção 5. Por fim, serão expressados os agradecimentos e fornecidas as referências bibliográficas utilizadas neste trabalho.

2. FUNDAMENTOS DO SOM NA COMPUTAÇÃO

O som é a vibração do ar, ou seja, uma variação na pressão do ar que um ouvido humano é capaz de detectar. Já na forma digital, tem-se o áudio, que é a conversão do som em números. Essa conversão é realizada por meio de um microfone, que transforma as variações da pressão do ar em sinais elétricos (voltagem) e essas voltagens são, então, convertidas em números que representam valores de uma onda sonora. Já uma onda sonora

¹ <https://github.com/cosmir/openmic-2018>

² <https://github.com/MTG/mtg-jamendo-dataset>

³ <https://magenta.tensorflow.org/datasets/nsynth>

⁴ <https://github.com/facebookresearch/demucs>

caracteriza-se por dois valores fundamentais: a amplitude, que representa o volume, e a frequência, que representa o tom. Sendo assim, um som completo é um mix de ondas, portanto, um áudio é um mix de frequências em diferentes amplitudes. Um som é um sinal contínuo e, para ser processado digitalmente, precisa ser convertido num sinal discreto por meio de amostragem. Essa taxa de amostragem, também conhecida como *sampling rate*, determina o número de amostras tiradas por segundo para representar o sinal contínuo digitalmente.

Normalmente costuma-se guardar dados no formato de tabelas, que são formadas por linhas e colunas. No entanto, os dados de áudio não possuem uma estrutura clara e organizada como as tabelas convencionais, apresentando desafios adicionais na sua representação. Existem diversas maneiras de representar um áudio a depender da finalidade; por exemplo, para avaliar aspectos temporais, é mais comum utilizar uma representação na forma de ondas, já para avaliar aspectos frequenciais, pode-se utilizar uma representação do espectro [6].

Um espectrograma é uma representação visual de como a frequência de um sinal de áudio varia ao longo do tempo. Mesmo em áudios com as mesmas notas, o espectrograma difere baseado no instrumento que origina a nota. O mel espectrograma é uma variação do espectrograma que utiliza uma escala de frequência mais adequada à percepção humana. Para isso, ao invés de utilizar uma escala logarítmica, é utilizada a escala mel, que é obtida através de uma conversão de frequências em Hertz para unidades mel [5]. Na Fig. 1, tem-se uma representação de quatro mel espectrogramas de diferentes instrumentos. No eixo vertical, tem-se a frequência em Hertz, indo de baixas até altas frequências. O eixo horizontal representa a evolução do sinal ao longo do tempo. Cores com intensidades mais altas representam altas frequências. Com essas informações fornecidas pelo espectrograma, é possível identificar padrões nos áudios, como harmônicos, mudanças na intensidade ao longo do tempo, entre outros.

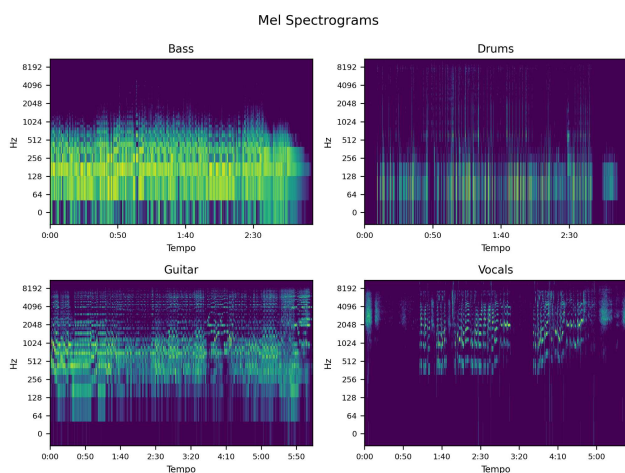


Figura 1: Mel Espectrogramas de diferentes instrumentos musicais

3. DATASETS

3.1 OpenMIC-2018

O dataset usado para treinamento do modelo foi o OpenMIC-2018, uma coleção de áudios rotulados produzido em uma colaboração entre o Spotify, o MARL (*Music and Audio Research Laboratory*), da Universidade de *New York*, e o *NYU Center for Data Science*. Composto por 20.000 amostras de áudio, cada uma com duração de 10 segundos, os áudios são polifônicos

e foram rotulados com a presença de instrumentos entre 20 instrumentos distintos, sendo alguns deles: guitarra, baixo, bateria, flauta, piano, voz, entre outros. Não há um mecanismo aprovado para acessar diretamente os dados de áudio, por isso, são disponibilizadas *features* pré-treinadas com base na arquitetura VGG⁵ para detecção de objetos em imagens. O modelo, conhecido por “VGGish”, foi treinado utilizando mel espectrogramas e produz um vetor de 128 dimensões e foi aplicada uma técnica de *ZCA-whitening* nas *features*, que é uma técnica de pré-processamento de dados utilizada para reduzir a redundância dos dados e melhorar a eficácia dos algoritmos de aprendizado [11]. O dataset foi rotulado manualmente por 2.500 contribuidores, reunindo por volta de 230 mil respostas nas 20 classes de instrumentos [3].

3.2 DEMUCS

O Demucs [8] é um modelo de separação de fontes musicais desenvolvido por pesquisadores do Facebook. Ele é capaz de separar áudios de instrumentos, como bateria, baixo e vocais do restante da música. O modelo é baseado numa arquitetura convolucional U-Net, inspirada pelo Wave-U-Net⁶. Normalmente, modelos de separação de fontes musicais funcionam sob o domínio do espectrograma ou de ondas sonoras. No entanto, as versões mais recentes do Demucs são modelos de separação híbridos que utilizam Transformers. O modelo foi treinado com o conjunto de dados MUSDB HQ⁷ e um conjunto de dados extra de 800 novas músicas. Recentemente, foi lançada uma nova versão que permite separar os instrumentos em áudios em seis fontes distintas, sendo elas: baixo, bateria, guitarra/violão, piano e vocal. Para fins de avaliações e testes neste estudo, foi criado um conjunto de dados que consiste em áudios monofônicos com instrumentos separados pelo Demucs. Os áudios foram extraídos tanto de performances gravadas em estúdio quanto de performances realizadas ao vivo com o intuito de verificar se o modelo atende também a classificar corretamente áudios que contém ruídos.

3.3 MTG-JAMENDO

O presente conjunto de dados foi criado com músicas polifônicas disponíveis no Jamendo sob licenças *Creative Commons* e tags fornecidas pelos usuários que fazem upload do conteúdo. Esse dataset contém cerca de 55 mil áudios com ao menos 30 segundos de duração e 195 categorias de gênero musical, instrumento e humor/tema [1]. Além disso, é possível realizar download dos áudios em alta ou baixa qualidade. Na Tabela 1 é possível verificar mais informações sobre o dataset.

Informação	Tags	Tracks	Artistas
Gênero	95	55.215	3.547
Instrumento	41	25.135	2.006
Humor/Tema	59	18.486	1.533
Total	195	55.609	3.565

Tabela 1: Informações sobre o dataset MTG-Jamendo

3.4 NSYNTH

Mais um conjunto de dados considerado uma referência no estudo de MIR, o NSynth apresenta cerca de 300 mil áudios monofônicos

⁵ <https://arxiv.org/abs/1409.1556>

⁶ <https://github.com/f90/Wave-U-Net>

⁷ <https://sigsep.github.io/datasets/musdb.html>

categorizados de acordo com sua fonte, família e qualidade. A fonte se refere ao método de produção do som, sendo ele acústico, eletrônico ou sintetizado; a família se refere à classificação dos áudios de acordo com o instrumento presente; e a qualidade se refere à resolução do som. Apesar do NSynth ser um dataset robusto, os áudios foram produzidos através de um piano MIDI em diferentes velocidades, no entanto, alguns instrumentos não foram capazes de produzir todos os tons. Além disso, o dataset contém alguns áudios duplicados com várias velocidades diferentes [4]. Tais características podem acabar interferindo na classificação de instrumentos.

4. METODOLOGIA

Para o presente estudo, foram utilizadas diversas bibliotecas e pacotes de Python. Um dos pacotes foi o Librosa [7], desenvolvido para aplicações que envolvem processamento e análise de áudios. Esse pacote possui desde funções que possibilitam a visualização de áudios até funções de extração de features contendo os algoritmos mais conhecidos quando se trata de informação musical, como o *Mel Spectrogram*, *Spectral Centroid*, *Spectral Rolloff*, e muito mais. Para o treinamento e avaliação do modelo, foi utilizada a biblioteca Scikit-Learn [2], uma das bibliotecas mais populares quando se trata de ciência de dados e que fornece uma grande gama de recursos para modelagem estatística, análise de dados, e suporte ao aprendizado de máquina, com algoritmos de classificação, regressão, clusterização, pré-processamento, entre outros.

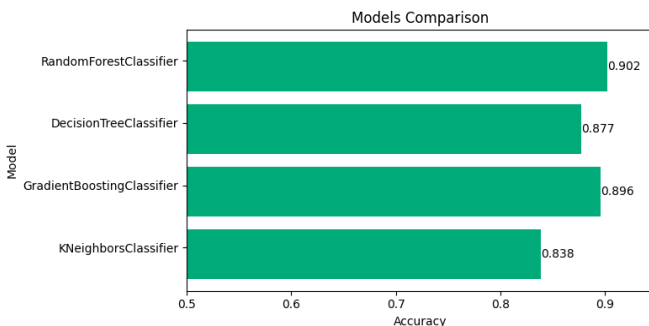


Figura 2: Acurácia de cada modelo nos dados do OpenMIC-2018

Foram verificados quatro modelos de classificação, sendo estes: *Random Forest Classifier*, *Decision Tree Classifier*, *Gradient Boosting Classifier* e *K-Neighbors Classifier*; com o objetivo de analisar qual seria o melhor algoritmo para treinar o modelo com os dados do OpenMIC-2018. Na Figura 2 podemos verificar que os números resultantes são bem próximos, com o pior algoritmo sendo o *K-Neighbors*, com 84% de acurácia, e o melhor algoritmo sendo o *Random Forest*, com 90% de acurácia.

Ainda sobre o treinamento do modelo, os dados do OpenMIC-2018 possuem uma informação de “relevância” com valores em ponto flutuante para cada um dos áudios. Essa relevância diz respeito à confiabilidade da anotação, por ser um dataset anotado por pessoas, foi realizado um cálculo baseado na quantidade de pessoas que votaram que um instrumento fazia parte de determinado áudio. Assim, no treinamento, foram consideradas anotações com relevância igual ou acima de 0.8.

Por razões de simplificação dos modelos, apenas alguns instrumentos foram considerados para o treinamento e avaliações deste trabalho. No Demucs e MTG-Jamendo foram considerados: baixo, bateria, guitarra/violão, piano e vocal. Já no NSynth, foram considerados baixo, guitarra, teclado e vocal (bateria não está presente no dataset). Além disso, antes da avaliação todos os áudios tiveram seus momentos de silêncio filtrados. Após isso, os áudios do Demucs e MTG-Jamendo foram divididos em vários áudios de 5 segundos, enquanto os áudios do NSynth já possuem apenas 4 segundos de duração. Assim, cada parte de um áudio era classificado individualmente e, após essa etapa, o instrumento com maior número de predições era o resultado da classificação final do áudio.

5. RESULTADOS

Como já foi mencionado anteriormente, os modelos treinados com dados do OpenMIC-2018 foram testados em diferentes conjuntos de dados. A métrica F1-score, utilizada para verificar o desempenho de modelos de classificação binária ou multi-classe, é calculada a partir dos valores de precisão e revocação, onde a precisão indica a proporção de amostras classificadas corretamente como positivas em relação ao total de resultados classificados como positivos (verdadeiros positivos e falsos positivos). Já a revocação indica a proporção de amostras classificadas corretamente como positivas em relação ao total de resultados que deveriam ter sido classificados como positivos (verdadeiros positivos e falsos negativos). As fórmulas para precisão, revocação e F1-score se encontram abaixo.

$$\text{Precisão} = \frac{vp}{vp + fp}$$

$$\text{Revocação} = \frac{vp}{vp + fn}$$

$$\text{F1 - Score} = 2 * \frac{\text{precisão} * \text{revocação}}{\text{precisão} + \text{revocação}}$$

Com os resultados dessa métrica, que varia de 0 a 1, tem-se que o valor 1 indica um bom equilíbrio entre a precisão e a revocação, significando que quanto mais perto de 1, maior é a capacidade de o modelo classificar corretamente as instâncias.

Dito isso, a Figura 3 mostra os valores de F1-score para cada dataset em cada um dos instrumentos. O dataset NSynth resultou nos piores resultados de F1, tendo uma média de 0.33, já o MTG-Jamendo resultou numa média de 0.44, enquanto o Demucs resultou nos melhores valores, obtendo uma média de 0.81. Os resultados que estão zerados na imagem significam a inexistência de dados de determinado instrumento no dataset.

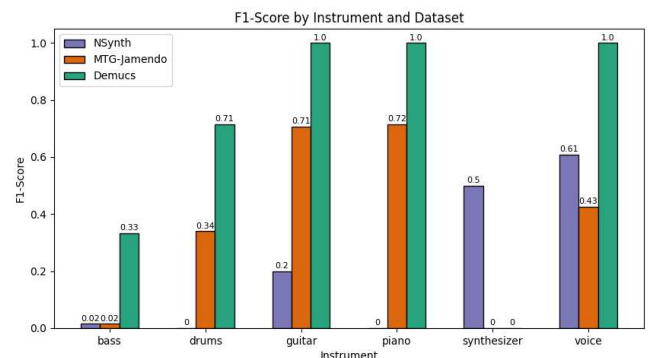


Figura 3: F1-Score para cada instrumento e dataset

Acredita-se que os resultados baixos de F1-Score para o NSynth advêm de problemas do conjunto de dados já mencionados

anteriormente, por exemplo, que todos os áudios foram produzidos através de gravações de um piano e, assim, o piano não consegue reproduzir bem o tom de alguns instrumentos. A Figura 3 mostra que para baixo e guitarra os resultados foram baixíssimos em relação ao Demucs, que foi o vencedor na comparação.

A respeito do MTG-Jamendo, os resultados não foram tão baixos quanto os do NSynth, no entanto, os dois conjuntos de dados obtiveram médias semelhantes. Observa-se, ainda, que o MTG-Jamendo ficou com um resultado pior do que o do NSynth para a categoria de voz.

Já sobre o Demucs, vencedor entre os três datasets, performou bem em praticamente todos os modelos de instrumentos. Referente aos dois instrumentos que não obtiveram F1 máximo, o baixo obteve precisão igual a 1 e revocação igual a 0.2; enquanto a bateria obteve precisão igual a 0.5 e revocação igual a 1. Os três datasets resultaram em valores pequenos no baixo. Foi observado que o principal erro dos modelos era confundir o baixo com bateria ou guitarra. Provavelmente, a confusão entre baixo e bateria se deve a ambos os instrumentos serem capazes de produzir sons mais graves; enquanto a confusão entre baixo e guitarra pode ser devido ao uso de efeitos de áudio, como distorção, delay, entre outros, ou devido ao uso de amplificadores, que podem moldar o timbre do instrumento.

No mais, tem-se que o dataset Demucs performou muito bem sob os dados do OpenMIC-2018, mesmo este possuindo áudios polifônicos e o Demucs tendo gerado áudios monofônicos.

6. AGRADECIMENTOS

Agradeço à minha família, à minha namorada e aos meus amigos por todo o apoio durante todo o tempo.

7. REFERÊNCIAS

- [1] Bogdanov, D., Won M., Tovstogan P., Porter A., & Serra X. (2019). The MTG-Jamendo Dataset for Automatic Music Tagging. Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019).
- [2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [3] Humphrey, Eric J., Durand, Simon, and McFee, B. (2018). "OpenMIC-2018: An Open Dataset for Multiple Instrument Recognition." in Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR).
- [4] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, p. 1068–1077. JMLR. org, 2017. DOI: <https://doi.org/10.48550/arXiv.1704.01279>
- [5] K. Racharla, V. Kumar, C. B. Jayant, A. Khairkar and P. Harish, "Predominant Musical Instrument Classification based on Spectral Features," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2020, pp. 617-622, doi: 10.1109/SPIN48934.2020.9071125.
- [6] Korstanje, J. (2021). "What is sound?". *Medium*. Disponível em: <https://towardsdatascience.com/what-is-sound-691988d780bb>
- [7] McFee, B et al. (2015). Librosa: v0.10.0.post2. Zenodo. DOI: <https://doi.org/10.5281/zenodo.7746972>
- [8] Rouard, Simon and Massa, Francisco and Défossez, Alexandre. Hybrid Transformers for Music Source Separation. 2023. ICASSP 23.
- [9] Vatsal P. (2021). "Recommendation Systems Explained". *Medium*. Disponível em: <https://towardsdatascience.com/recommendation-systems-explained-a42fe60591ed>
- [10] VIEIRA, Rômulo; ARAÚJO, João Teixeira; BATISTA, Edimilson; SCHIAVONI, Flávio Luiz. Automatic Classification of Instruments from Supervised Methods of Machine Learning. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO MUSICAL (SBCM), 18. , 2021, Recife. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 1-7. DOI: <https://doi.org/10.5753/sbcm.2021.19418>.
- [11] Yong, H., Zhang, L. (2022). An Embedded Feature Whitening Approach to Deep Neural Network Optimization. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13683. Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-20050-2_20