

Projeto Reconhecedor de Estilos Musicais

Introdução

A música é uma das formas mais universais de expressão humana, transcendendo barreiras culturais e linguísticas. Nos últimos anos, a tecnologia tem desempenhado um papel cada vez mais significativo na forma como interagimos com a música. Empresas de grande porte, como Spotify, Apple e Google, têm investido bilhões de dólares no desenvolvimento de softwares capazes de reconhecer, categorizar e recomendar músicas. Esses investimentos impulsionaram o surgimento de ferramentas avançadas de reconhecimento musical, como o Shazam, que identifica músicas com precisão em segundos.

Inspirado por esse cenário em constante evolução, este projeto tem como objetivo introduzir os alunos ao mundo dos modelos de reconhecimento de áudio por meio do desenvolvimento de um reconhecedor de estilos musicais. O processo de reconhecimento musical envolve o estudo e análise dos sinais de áudio, um campo amplamente explorado na disciplina de Sinais e Sistemas.

A Transformada de Fourier (FT) desempenha um papel fundamental nesse contexto, sendo uma ferramenta essencial para decompor os sinais musicais em suas frequências constituintes. Por meio dessa técnica, é possível representar um sinal de áudio no domínio da frequência, permitindo a extração de características como o espectro de potência, que revela informações cruciais sobre o ritmo, melodia e instrumentação de uma música. Esses dados servem como base para a categorização automática dos estilos musicais.

Além disso, o projeto busca explorar a aplicação de filtros e janelas temporais no pré-processamento dos sinais, aprimorando a qualidade dos dados analisados. Esses conceitos, combinados com técnicas modernas de aprendizado de máquina, como redes neurais convolucionais (CNNs), permitirão criar um sistema capaz de identificar padrões específicos de cada estilo musical com alta precisão.

Objetivos

O objetivo do projeto é utilizar conceitos de tratamento de sinais para desenvolver um classificador que seja capaz de receber como input um arquivo de áudio contendo o trecho de uma música e fornecer como output o nome do estilo musical correspondente. Recomenda-se que o código para esse reconhecedor seja elaborado com o auxílio de bibliotecas de Python como a “Librosa”, utilizando o ambiente do Google Colaboratory ou Jupyter Notebook.

Com o reconhecedor de estilos musicais finalizado, os alunos devem elaborar uma apresentação, juntamente com um breve relatório, explicando as técnicas que foram aplicadas e exibindo a eficácia do classificador na forma de uma matriz de confusão.

Dessa forma, a partir deste projeto, os participantes terão a oportunidade de aplicar os conceitos teóricos de Sinais e Sistemas em um cenário prático e inovador, unindo a análise de sinais no domínio do tempo e da frequência com algoritmos avançados de processamento de dados. Essa abordagem não apenas reforça a compreensão técnica, mas também destaca o impacto das ferramentas matemáticas e computacionais no avanço da interação entre humanos e música.

Metodologia

O processo de reconhecimento de um áudio é comumente dividido em cinco etapas bem definidas: escolha da base de dados, pré-processamento dos sinais, extração de características, filtragem de características e classificação.



A escolha da base de dados é uma etapa importante. Escolher dados já tratados e amplamente utilizados na literatura podem facilitar bastante nas etapas seguintes. Por outro lado, também é possível gerar seu próprio conjunto de dados para a utilização no

classificador. Porém, para isso, será necessário adicionar algumas camadas de pré-processamento dos sinais que podem dificultar um pouco o trabalho.

A etapa de pré-processamento consiste em tratar os dados dos sinais para entregá-los ao extrator de características. Nesta etapa, geralmente são retirados os ruídos, o som é suavizado e melhorado. Além disso, também pode-se realizar a decimação, visando diminuir a frequência amostral do sinal para a melhora do desempenho.

As etapas de extração e seleção de características são onde ocorrem as escolhas mais importantes. Deve-se escolher quais características são consideradas mais relevantes do sinal, para poder fazer o reconhecimento. Uma faixa de áudio possui uma gama de características. Dependendo de qual tipo de sinal está sendo classificado (voz, ambiente, música), algumas características podem ser mais relevantes do que outras.

A etapa de classificação consiste em fornecer ao classificador as características extraídas. O grande desafio dessa etapa é a escolha do classificador e dos hiperparâmetros a serem utilizados. Na literatura há vários reconhecedores, como será visto mais adiante.

Especificações

1. O projeto deverá ser realizado por equipes com **3 integrantes**;
2. A partir de alguma base de dados, serão selecionados sinais de áudio da categoria escolhida que servirão como entrada para o modelo;
3. A equipe irá construir um modelo reconhecedor, o qual:
 - a. Poderá conter alguma etapa de pré-processamento (não obrigatório);
 - b. Deverá realizar **pelo menos 3 procedimentos de extração de características**, onde um deles seja a Transformada de Fourier (ou contenha a Transformada de Fourier por trás dos panos, ex: geração do espectrograma);
 - c. Possuirá um algoritmo reconhecedor (um classificador) que passa por treinamento e em seguida tem sua eficácia verificada a partir de um conjunto de teste (a equipe escolhe qual métrica determina essa eficácia). A **acurácia deve ser especificada** no arquivo do projeto.
 - d. Independentemente da métrica utilizada, a **matriz de confusão deve ser mostrada na apresentação**.
 - e. No conjunto de validação deve ter ao menos **um áudio com ruído** (qualquer tipo de ruído, desde que seja perceptível) e **um áudio com eco**. Se não existirem arquivos com essas características na base de dados, o grupo deve aplicar esses efeitos manualmente. O código do projeto deve deixar evidente quais são os áudios com ruído e eco e o output gerado pelo classificador para esses dois arquivos.
4. Obs: Solicita-se que a execução completa do modelo não dure mais do que **20 minutos**.

Exemplos

Exemplos de base de dados:

- a. GTZAN
- b. Richard Trebichavský's music-genres-dataset
- c. MGD+ Dataset
- d. Ccmusic-database/music_genre
- e. MachineHack Hackathon - Music Genre Classification
- f. Vicsuperman - Prediction of music genre

Exemplos de pré-processamento:

- a. Repartição em janelas de $\approx 20\text{ms}$ (janelamento)
- b. Window functions como retangular, Hann, Hamming e Blackman
- c. Sobreposição de janelas
- d. Decimação do sinal
- e. Linear-frequency spectrogram
- f. Critical bands
- g. Gammatone filters
- h. Mel-scale spectrogram (como o Log-Mel spectrogram)
- i. Constant-Q transform (CQT) spectrogram
- j. Pyramids representation
- k. Wavelets representation
- l. Scattering transform

Exemplos de extração de parâmetros (Obs.: alguns dos métodos a seguir podem não ter sido colocados na categoria mais adequada, portanto essa classificação pode ser contestável. Alguns deles também podem ser similares, embora estejam com nomenclaturas distintas. Recomenda-se verificar o devido conceito das extrações selecionadas):

- a. **Temporais:**
 - i. Time domain Envelope
 - ii. Zero crossing rate (ZCR)
 - iii. Temporal waveform moments
 - iv. Autocorrelation coefficients
 - v. Short-time energy (STE)
 - vi. Pitch
- b. **Especrais:**
 - i. Energy
 - ii. Spectral centroid
 - iii. Spectral envelope
 - iv. Spectral moments
 - v. Spectral flatness
 - vi. Spectral slope

- vii. Spectral roll-off
- viii. Spectral flux
- ix. Spectral irregularity features
- x. Narrow-Band Auto Correlation Function features (NB-ACF)
- c. **Cepstrais:**
 - i. Mel-frequency cepstral coefficients (MFCC) and their first and second derivatives (Δ MFCC and $\Delta\Delta$ MFCC)
 - ii. Linear prediction cepstral coefficients (LPCC)
 - iii. Gammatone feature cepstral coefficients (GFCC ou GTCC)
 - iv. Constant-Q cepstral coefficients (CQCC).
 - v. Homomorphic Cepstral Coefficients (HCC)
 - vi. Bark- Frequency Cepstral Coefficients (BFCC)
 - vii. Spectral Dynamic Features (SDF)
- d. **Motivados por percepção:**
 - i. Loudness
 - ii. Sharpness
 - iii. Perceptual spread
- e. **Baseados na imagem de espectrogramas:**
 - i. Histogram of oriented gradients (HOG)
 - ii. Subband power distribution (SPD)
 - iii. Local binary pattern (LBP)
 - iv. Log-Gabor filtering
- f. **Baseados em tempo e frequência:**
 - i. Matching pursuit (MP, MP-Gabor)
 - ii. Orthogonal matching pursuit (OMP)
 - iii. Binary wavelet packet tree (WPT)

Exemplos de reconhecedores (Obs.: alguns dos algoritmos a seguir podem/devem ser utilizados em conjunto com outro reconhecedor):

- a. PCA
- b. Linear discriminant analysis (LDA)
- c. Filter approaches
- d. Support vector machines (SVM)
- e. Multiple kernel learning (MKL)
- f. Random-forest
- g. Hidden Markov models (HMM)
- h. Gaussian mixture model (GMM)
- i. Learning vector quantization (LVQ)
- j. K Nearest Neighbours (KNN)
- k. Discriminant cluster selection (DSS)
- l. BayesNet
- m. Naive Bayes

Principais critérios avaliados

Os critérios abaixo não necessariamente possuem o mesmo peso no projeto, e o não cumprimento de algum desses critérios não significa a anulação da nota. São critérios que, se bem atingidos, irão favorecer a nota do grupo.

- Cumprimento da apresentação em formato de Pitch dentro do tempo de 4min
- Explicação do processamento de sinais efetuado pelo modelo
- Explicação do funcionamento dos extractores, qual a sua relevância para o modelo construído e por que foi escolhido para ser usado
- Explicação do classificador, qual a sua relevância para o modelo construído e por que foi escolhido para ser usado
- Desempenho do classificador, segundo as métricas adotadas.
- Justificativa do desempenho observado do classificador: tanto para resultados bons quanto ruins (explicar o porquê), e como ele poderia ser melhorado
- Resultados apresentados serem compatíveis com os resultados observados quando o modelo for executado (seja na apresentação ou durante a avaliação)
- Execução completa do modelo não durar mais do que 20min

Entregas

- **Primeira entrega, 13/12** - As equipes deverão informar seus integrantes na planilha de grupos que será disponibilizada no Classroom.
- **Entrega Final**, data: **02/04** - Apresentações dos projetos, que terão duração de aproximadamente **4 minutos**, e entregas dos trabalhos finais. Durante a apresentação, a equipe irá justificar as escolhas realizadas na construção do seu modelo e, se possível, mostrar a execução do reconhecedor criado. A apresentação valerá **30% da nota do projeto**.
- **Entrega do Relatório**, será feita na mesma data da apresentação do projeto: **02/04** - O relatório deverá contar com detalhes do projeto, assim como a **justificativa da escolha de cada método extrator**. O relatório deve ter **até 3 páginas** (capa + 2 páginas de textos).

Links e Vídeos Interessantes

https://www.researchgate.net/publication/267450621_AUTOMATIC_SPEECH_RECOGNITION_USING_FOURIER_TRANSFORM_AND_NEURAL_NETWORK

<https://www.ijert.org/research/speech-recognition-using-fast-fourier-transform-algorithmIJERTCONV10IS08007.pdf> <https://developer.nvidia.com/blog/essential-guide-to-automatic-speech-recognition-technology/> <https://www.scaler.com/topics/nlp/architecture-of-automatic-speech-recognition/> <https://realpython.com/python-scipy-fft/>

<https://www.youtube.com/watch?v=BXghmsHmKY&list=PL4K9r9dYCOoqmykdiyCq2jyAb0zwO0p-b>

<https://www.youtube.com/watch?v=DpchUWUsYs0&list=PLASpGWv0ToUHyTZ6FL0K2rhTkVaV9GPND> <https://www.youtube.com/watch?v=E8HeD-MUrjY>

<https://www.youtube.com/watch?v=iCwMQJnKk2c&list=PLwATfeyAMNqlee7cH3q1bh4QJFAaeNv0>

https://www.youtube.com/watch?v=gMQyGASOZO0&list=PLuWx2S0SyaDd_eMm68ep0XEUDUpnw0Hcl <https://www.youtube.com/watch?v=q67z7PTGRi8&list=PLpCZr5mhfo86H0eRtTGuDSsFYsHcTzk9>

Exemplos de Datasets

1. GTZAN - <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>
2. Richard Trebichavský's music-genres-dataset - <https://github.com/trebi/music-genres-dataset>
3. MGD+ Dataset - <https://zenodo.org/records/8086643>
4. Ccmusic-database/music_genre - https://huggingface.co/datasets/ccmusic-database/music_genre
5. MachineHack Hackathon - Music Genre Classification - <https://www.kaggle.com/datasets/purumalgi/music-genre-classification>
6. Vicsuperman - Prediction of music genre - <https://www.kaggle.com/datasets/vicsuperman/prediction-of-music-genre>

Contato da Monitoria

- Carlos Eduardo - cebms@cin.ufpe.br

- Hugo Almeida - ham4@cin.ufpe.br
- Luisa Leiria - lfla@cin.ufpe.br
- Thiago Ramalho - trm4@cin.ufpe.br

Referências

- Abdusalomov, A. B., Safarov, F., Rakhimov, M., Turaev, B., & Whangbo, T. K. (2022). Improved feature parameter extraction from speech signals using machine learning algorithm. *Sensors*, 22(21).
- Becoulet, A., & Verguet, A. (2021). A depth-first iterative algorithm for the conjugate pair fast Fourier transform. *IEEE Transactions on Signal Processing*, 69: 1537-1547.
- Borandağ, E. (2019). Markov model based real time speaker recognition using k-means, fast fourier transform and mel frequency cepstral coefficients. *Celal Bayar University Journal of Science*, 15(3): 287-292.
- Duhamel, P., & Vetterli, M. (1990). Fast Fourier transforms: a tutorial review and a state of the art. *Signal Processing*, 19(4): 259-299.
- Fendji, J. L. K. E., Tala, D. C., Yenke, B. O., & Atemkeng, M. (2022). Automatic speech recognition using limited vocabulary: A survey. *Applied Artificial Intelligence*, 36(1).
- Li, S., Xue, K., Zhu, B., Ding, C., Gao, X., Wei, D., & Wan, T. (2020). Falcon: A Fourier transform based approach for fast and secure convolutional neural network predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8705- 8714.
- Liu, W., Liao, Q., Qiao, F., Xia, W., Wang, C., & Lombardi, F. (2019). Approximate designs for fast Fourier transform (FFT) with application to speech recognition. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(12): 4727-4739.
- Malik, M., Malik, M. K., Mehmood, K., & Makhdoom, I. (2021). Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80: 9411-9457.
- Polur, P. D., & Miller, G. E. (2005). Experiments with fast Fourier transform, linear predictive and cepstral coefficients in dysarthric speech recognition algorithms using hidden Markov model. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(4): 558-561.
- Rajaby, E., & Sayedi, S. M. (2022). A structured review of sparse fast Fourier transform algorithms. *Digital Signal Processing*, 123.

Shchekotov, I., Andreev, P., Ivanov, O., Alanov, A., & Vetrov, D. (2022). Ffc-se: Fast Fourier convolution for speech enhancement. arXiv preprint arXiv:2204.03042.

Sorensen, H. V., Jones, D., Heideman, M., & Burrus, C. (1987). Real-valued fast Fourier transform algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(6): 849-863.