APRESENTAÇÃO DE ARTIGO

AUTOMATED DEEP LEARNING GENERATED FINGERPRINTS VS. MANUAL HEURISTICALLY DESIGNED FINGERPRINTS

PROJETO FINAL

Mariana Oliveira, *Deep learning* e música 20191

TÓPICOS A APRESENTAR

SUMÁRIO

TRABALHOS SELECIONADOS

Introdução

METADADOS

Now Playing: Continuous low-power music recognition
Escrito por Gfeller, Beat, et al. e publicado em *NIPS (Conference on Neural Information Processing Systems) Workshop: Machine Learning on the Phone* no ano de 2017.

An Industrial Strength Audio Search Algorithm
Escrito por Avery Wang e publicado em *ISMIR (International Society for Music Information Retrieval)* no ano de 2013.

PROBLEMA

Ambos os trabalhos possuem como problema principal o reconhecimento de uma música dado poucos segundos de áudio, possivelmente ruidoso.

A abordagem da Google foca em automatizar esse processo.

As duas abordagens trabalham com o ambiente *mobile*.

Fig 1. Aplicativos Shazam (topo) e Now Playing (abaixo). Fontes: https://variety.com/2018/digital/news/apple-shazam-acquisition-closed-1202954409/ e https://www.xda-developers.com/googles-now-playing-prepares-to-finally-add-support-for-showing-history/.

TRABALHOS SELECIONADOS

Introdução

NOW PLAYING

Um detector de música, executado continuamente, detecta se há música no ambiente. Caso positivo, um reconhecedor de música gera a impressão digital do segmento de áudio e um algoritmo de *matching* o busca em uma base de dados de impressões digitais de diversas músicas.

SHAZAM

Um usuário chama o serviço e faz amostragens de até 15 segundos de áudio. Dessa amostra é gerada a impressão digital e uma identificação é realizada no servidor por um algoritmo que usa uma análise *hashed time-frequency constellation* do áudio.

Fonte dos ícones: https://www.stickpng.com

AUDIO FINGERPRINTING

Conhecimentos prévios

DEFINIÇÃO, OBJETIVO E APLICAÇÕES

Uma impressão digital de áudio (*audio fingerprint*) é uma representação compacta de um segmento de áudio que encapsula informações relevantes deste. O objetivo é capturar a assinatura do segmento de áudio, que seja robusta ao ruído e à distorção e que permita diferenciá-lo outros sons.

Utilizado em tecnologias de identificação de contéudo baseado em impressão digital, cujo processo é descrito na imagem ao lado. A digital de áudio permite identificar, p.e., não só qual a música estamos ouvindo, mas também qual a parte da música estamos ouvindo.

Uma aplicação nesse sentido são sistemas de monitoramento baseados em impressão digital, utilizados por estações de rádio para monitorar direitos autorais de músicas e por anunciantes para verificar se os comerciais estão sendo transmitidos conforme acordado.

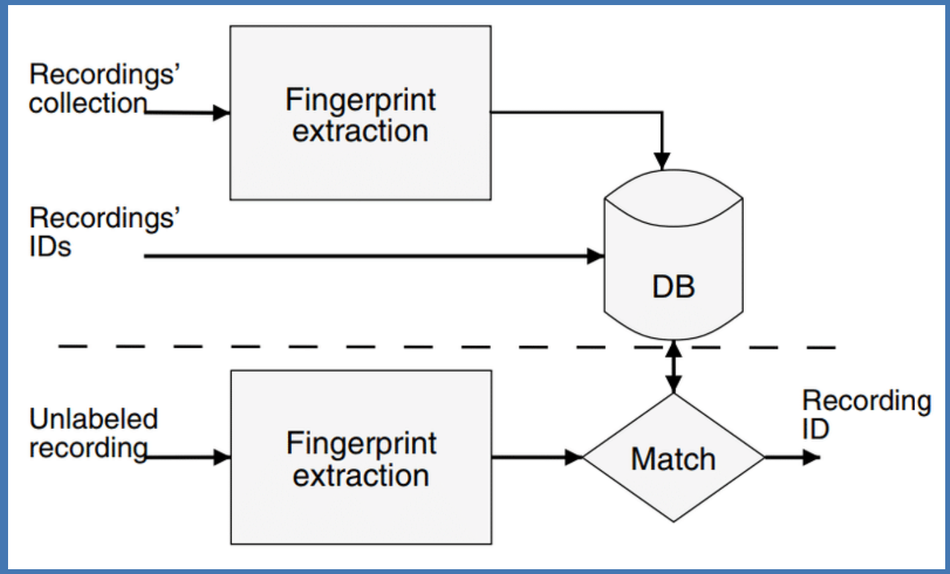

Fig 2. Processo de identificação de conteúdo baseado em digital de áudio. Extraído de http://mtg.upf.edu/files/publications/0e9cd9-Springer05-pcano.pdf

ANÁLISE DE FREQUÊNCIA

Conhecimentos prévios

ESPECTROGRAMA

Comumente, áudios são analisados pelo domínio da frequência e, geralmente, digitais de áudio baseiam-se nas características de um espectrograma.

Um espectrograma é uma decomposição aproximada do sinal ao longo do tempo e frequência. É construído ao aplicar a **Transformada de Fourier** (decompõe uma função temporal, um sinal, em frequências) em janelas de tamanho m, que irá representar (decompor em componentes seno e cosseno, ou senoides) o sinal no domínio da frequência.

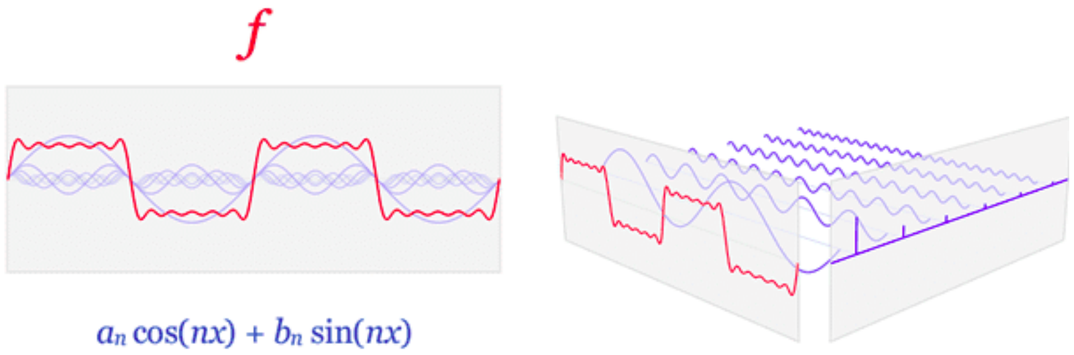

Fig 3. A função f é resolvida em termos de senos e cossenos. Os componentes de frequência de f estão organizados no spectro da frequência e representados por picos no domínio da frequência.
Extraído de https://en.wikipedia.org/wiki/Fourier_transform.

ANÁLISE DE FREQUÊNCIA

Conhecimentos prévios

ESPECTROGRAMA

Visualmente, são representados em 2D, em um gráfico tempo X frequência e colorido de acordo com a amplitude (volume)/energia; ou em 3D, como cascatas, que pode ser visualizado em tempo real pelo Chrome Music Lab.

Para criar uma impressão digital, devemos extrair as características que melhor definem o áudio do espectrograma. Existem diversas abordagens possíveis para esse fim.

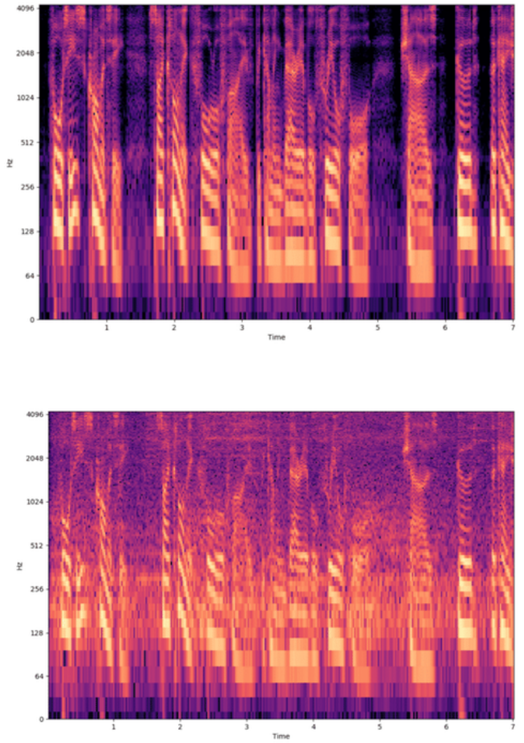

Fig 4. Espectograma de uma fala (topo) e o mesmo com ruído (abaixo). Picos se mantém. Fonte: https://blog.chirp.io/audio-fingerprinting-what-is-it-and-why-is-it-useful/

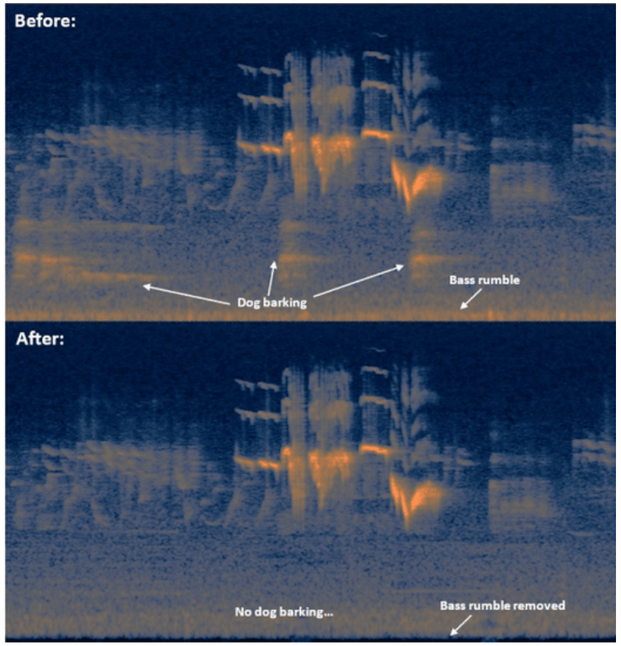

Fig 5. Exemplo de como o espectograma pode ser útil. Fonte: https://blogs.bl.uk/sound-and-vision/2018/09/seeing-sound-what-is-a-spectrogram.html

REVISÃO DA LITERATURA

Trabalhos relacionados

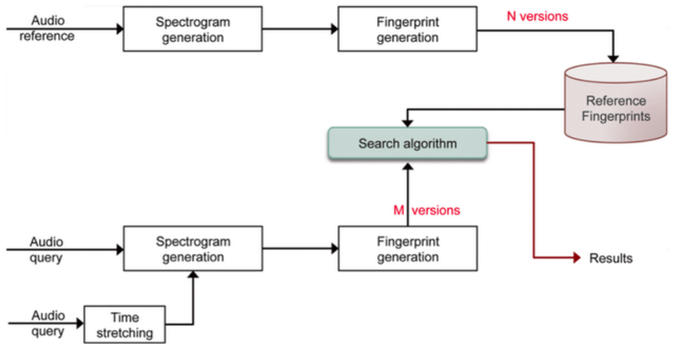

Fig 6. Arquitetura sistema de digital de áudio baseado em espectograma.
Fonte: https://link.springer.com/article/10.1007/s11042-015-3081-8

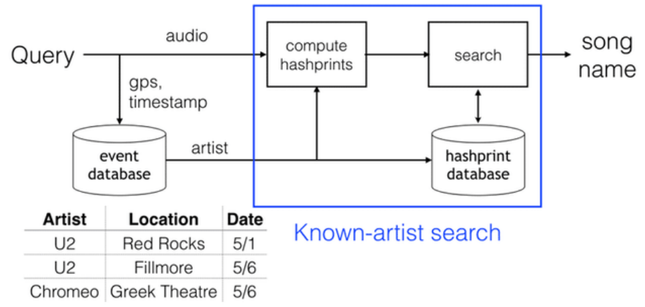

Fig 7. Arquitetura sistema de identificação de música ao vivo.
Fonte: http://pages.hmc.edu/ttsai/assets/LiveSongID_TMM17.pdf

A SPECTROGRAM-BASED AUDIO FINGERPRINTING SYSTEM FOR CONTENT-BASED COPY DETECTION

Ouali et al. (2016) introduziu um método de fingerprinting de áudio baseado em imagens de espectogramas. A ideia é que, embora o espectrograma do áudio original e sua cópia pareçam muito semelhantes, as distorções podem alterar a informação visual. Para reduzir a incompatibilidade de áudio devido a essas distorções, o espectrograma é convertido em imagens binárias (matriz binária; 1 acima da média, 0 abaixo da média), a partir das quais são geradas impressões digitais diferentes. O componente de variação é a média das intensidades. As impressões digitais são então comparadas com referências conhecidas.

KNOWN-ARTIST LIVE SONG IDENTIFICATION USING AUDIO HASHPRINTS

Tsai et. al (2017) para o problema de identificação de música ao vivo propõe o seguinte: Para a extração de feature, aplica-se a transformada constante-Q, que é uma transformação onde o espaçamento e a largura de seus filtros combinam com os tons da escala musical. As digitais utilizam então uma representação *Hamming* (binarização). O áudio referência passa pelo mesmo processo, porém adicionando variações de *pitch*, dado que se trata de uma versão ao vivo. A busca consiste em identificar os candidatos mais próximos e refinar em seguida, resultando em um ranking de pontuação.

APLICATIVOS

Trabalhos relacionados

Fig 8. Logo Audible Magic.
Fonte: https://www.audiblemagic.com/

Fig 9. Aplicativo SoundHound.
Fonte: https://www.soundhound.com/soundhound

AUDIBLE MAGIC

Fundada em 1999 para permitir uma nova experiência do usuário com uma tecnologia de identificação de áudio, a Audible Magic foi pioneira no uso de Reconhecimento Automático de Conteúdo em diversas aplicações. É utilizada por grandes nomes nas indústrias de mídia e tecnologia, e oferece diversas soluções. Também utiliza a ténica de *fingerprinting*.

SOUNDHOUND

Aplicação muito similar ao Shazam, porém com outras *features*, como assistente pessoal, acesso a letras de música em tempo real e a possibilidade do usuário poder cantar ou murmurar uma música a ser reconhecida.

ARQUITETURAS

Comparação de soluções

DETECTOR DE MÚSICA

Comparação de soluções

ESTRUTURA E FINALIDADE

Em um chip DSP (*Digital Signal Processor;* microprocessador especializado em processamento digital de sinal) separado, encontra-se um detector de música, rodando continuamente, que identifica se há música tocando no ambiente.

Recursos são extremamente limitadas no chip DSP para evitar a perda de bateria pelo dispositivo.

Esse detector evitar o cálculo das digitais de áudio, que é algo custoso computacionalmente.

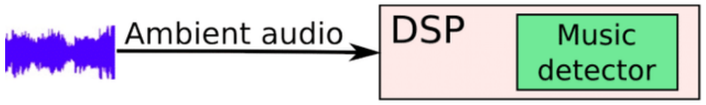

Fig 10. Figura extraída do artigo "Now Playing: Continuous low-power music recognition".

ARQUITETURA

O detector de música funciona da seguinte maneira: A partir do fluxo de áudio detectado, extrai-se *features log Mel*. Então, uma rede neural calcula a probabilidade de uma música estar tocando, usando uma janela dos vetores de *features*.

A rede estrutura-se em 6 camadas convolucionais seguida por uma *multilayer perceptron*. A rede foi treinada com subconjuntos da base AudioSet e um conjunto adicional de áudio ruidoso; todos rotulados em "música presente" e "música não presente".
Ao final, uma janela deslizante de poucos segundos passa sobre o fluxo de predições da rede, extraindo a média do intervalo. Após c predições de confiança acima do limite t, uma detecção é registrada.

No total, o modelo tem 8 k parâmetros e ocupa menos de 10KB de memória.

DETECTOR DE MÚSICA

Comparação de soluções

ARQUITETURA (OBS.)

- Todas as camadas, menos a última, usam a função de ativação ReLU e a *batch normalization;*
- Cada camada convolucional reduz a dimensionalidade da entrada por um fator de 2;
- O *kernel stride* usado é de 2;
- A janela móvel no final ajuda a filtrar alguns dos erros da rede neural e garante que o buffer de áudio contenha uma quantidade suficiente de música a ser reconhecida.

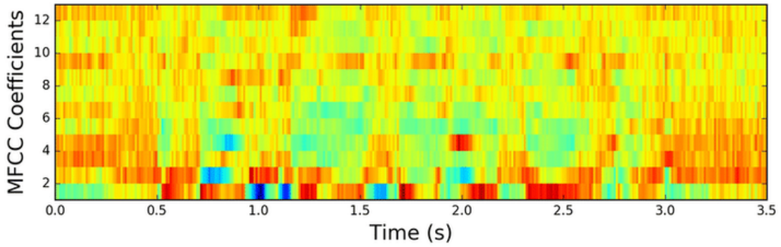

Fig 11. Exemplo de coeficientes *Mel-frequency cepstral*. Fonte: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

layer	size-in	kernel
separable conv2d	446x1x32	4x1, 2
separable conv2d	222x1x32	4x1, 2
separable conv2d	110x1x32	4x1, 2
separable conv2d	54x1x32	4x1, 2
separable conv2d	26x1x32	4x1, 2
separable conv2d	12x1x32	4x1, 2
flatten	5x1x32	
fully connected	160	
fully connected	8	

Fig 12. Arquitetura do detector de música extraída do artigo "Now Playing: Continuous low-power music recognition".

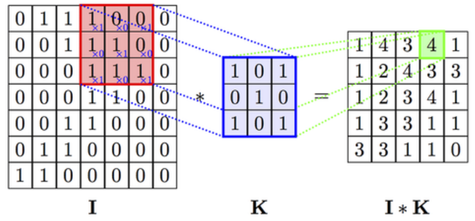

Fig 13. Operação de Convolução. Fonte: https://github.com/PetarV-/TikZ/tree/master/2D%20Convolution

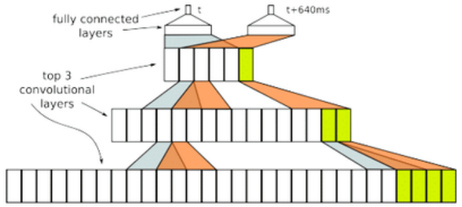

Fig 14. Esquema visual da rede convolucional. Fonte: Artigo do Medium "Understanding of Convolutional Neural Network (CNN)" por Prabhu.

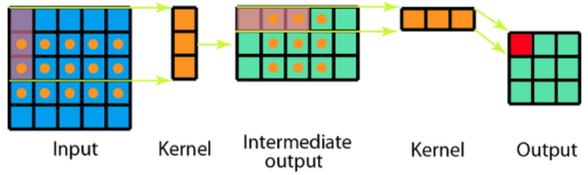

Fig 15. Convolução separada.
Fonte: Artigo do Medium "A Comprehensive Introduction to Different Types of Convolutions in Deep Learning" por Kunlun.

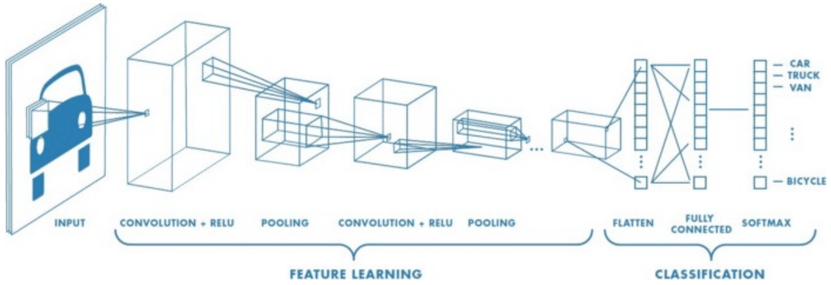

Fig 16. Esquema visual da rede convolucional.
Fonte: Artigo do Medium "Understanding of Convolutional Neural Network (CNN)" por Prabhu.

CRIAÇÃO DE DIGITAIS

Comparação de soluções

REDE NEURAL CONVOLUCIONAL COM DEVIDE-AND-ENCODE

A partir do áudio (poucos segundos), uma rede neural (Neural Network Fingerprinter, NNFP) analisa o espectograma e emite uma única digital a cada segundo. A estrutura da rede se dá por camadas convolucionais seguidas por um bloco *two-level devide-and-encode*, que quebra a representação em múltiplos ramos. Todas as camadas, exceto o último bloco, utilizam a função de ativação ELU e *batch normalization*. A rede foi treinada com a função *triplet loss*, que, para cada segmento de áudio e seus exemplos, a distância é minimizada entre eles enquanto a distância deles para outros segmentos de áudio é maior. Segmentos de áudio são considerados iguais apenas se suas posições iniciais diferirem em menos de algumas centenas de milissegundos e forem da mesma música. O modelo NNFP é treinado em segmentos de áudio ruidosos correpondendo a segmentos de uma música referência.

ROBUST CONSTELLATIONS + COMBINATORIAL HASHING

A partir do áudio, é obtido seu espectograma. A feature extraída do espectograma são os picos, dado sua robustez quanto a ruído. Assim, o espetograma é reduzido a um conjunto esparso de coordenadas, chamado de mapa de constelação (informação de amplitude é eliminada). Esse conjunto de pontos identifica unicamente uma música. Porém cada ponto é dependente do tempo, o que torna ineficiente em identificar segmentos de áudio, pois comparação ponto a ponto sem o contexto do tempo perde sentido. Para resolver esse problema, são criadas *hashes* pelo mapa de constelação que irá associar pares de ponto e são invariante no tempo.

CRIAÇÃO DE DIGITAIS

Comparação de soluções

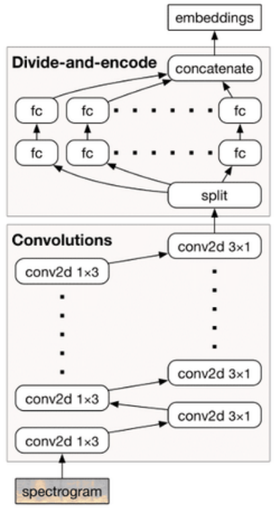

Fig 17. Estrutura da rede de digital de áudio extraída do artigo "Now Playing: Continuous low-power music recognition".

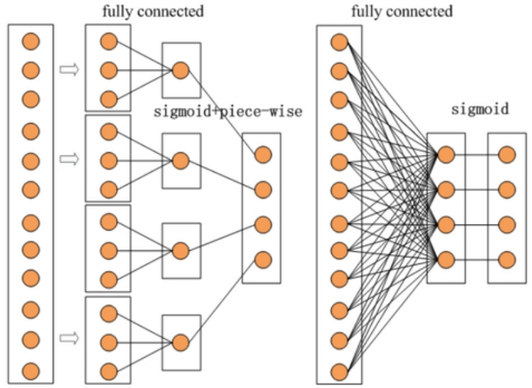

Fig 18. Bloco *divide-and-encode* extraído do artigo "Simultaneous Feature Learning and Hash Coding with Deep Neural Networks".

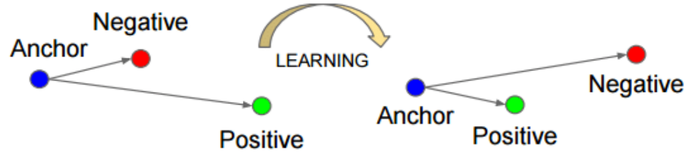

Fig 19. Na função de custo Triplet Loss, um baseline (âncora) é comparado com uma entrada positiva (verdade) e uma entrada negativa (falsa). A função minimiza a distância entre o âncora e o positivo, pois ambos tem a mesma identidade; enquanto maximiza a distância entre o âncora e o negativo, pois ambos tem identidades diferentes. extraído do artigo "FaceNet: A Unified Embedding for Face Recognition and Clustering".

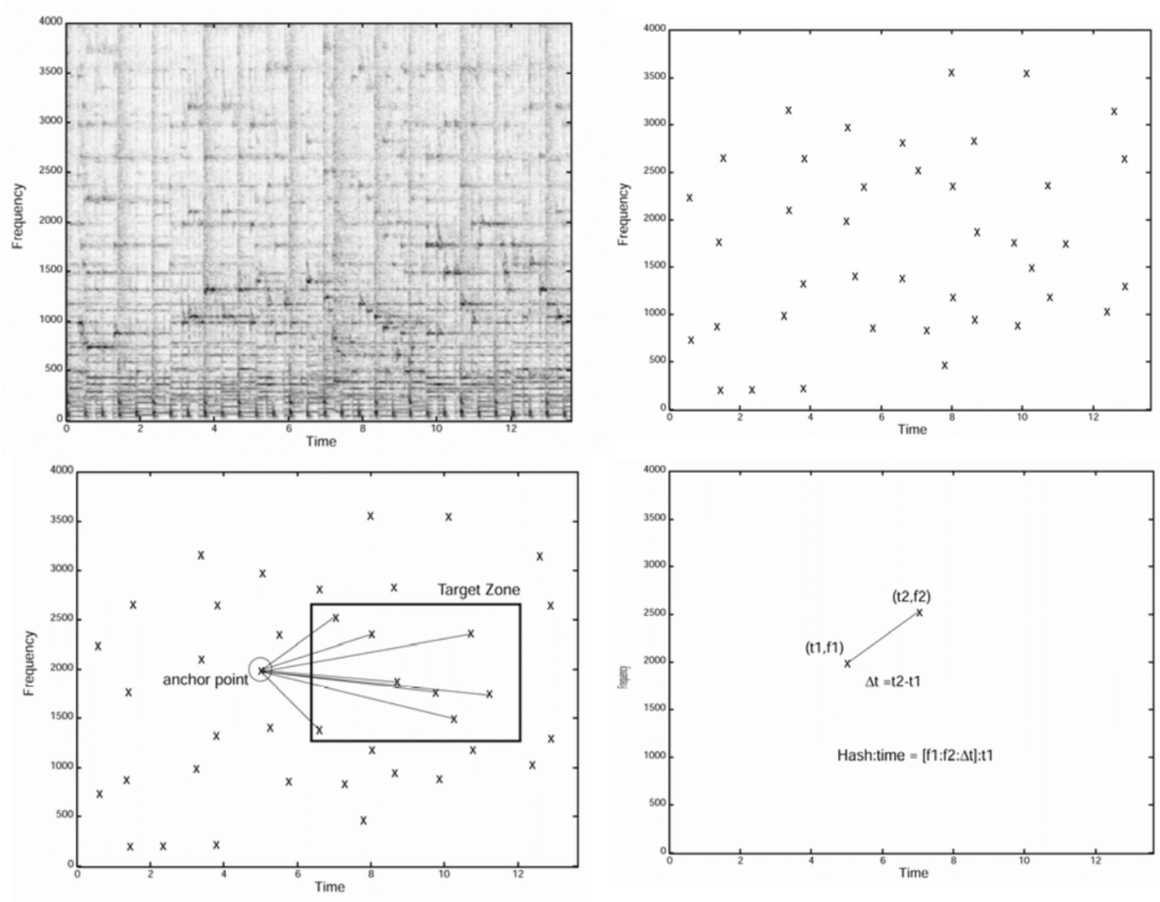

Fig 20. Processo de criação de digital de áudio extraída do artigo "An Industrial-Strength Audio Search Algorithm".

CORREPONDÊNCIA DE DIGITAIS

Comparação de soluções

NEAREST NEIGHBOR SEARCH + PONTUAÇÃO REFINADA

A busca ocorre em dois estágios: Primeiro, cada digital da query é pesquisada no banco de dados de modo a encontrar os primeiros K vizinhos mais próximos. Após uma pontuação mais refinada é feita com os candidatos promissores.

O banco de dados foi comprimido e algumas estratégias foram adotadas a tornar a busca menos custosa, como a minimização do erro decorrente da compressão das digitais de referência (q: | ||q - x||2 - ||q - x̂||2|). Como essaa é aproximada, pode não encontrar digitais próximas a algumas digitais da consulta. Assim, para ser mais preciso, recupera-se as digitais das músicas promissoras. Em seguida, dada a sequência de digitais do buffer de áudio e as de uma música no banco, estima-se a similaridade entre eles *pairwise* que são somadas para obter a pontuação final.

SCATTERPLOT DA LOCALIZAÇÃO DE HASHES

Shazam considera a seguinte proposição: Dado um áudio A e A' subconjunto de A, então *shazam*(A') está contido em *shazam*(A), sendo *shazam*(A) um conjunto de hashes.

Porém apenas verificar por esse propriedade, no momento de correspondência de digitais, não é o suficiente. Um conjunto estar subcontido em outro não significa obrigatoriamente que os elementos em comum estarão na mesma ordem. Shazam certifica-se de que ambas essas características irão ocorrer. Para isso, a amostra de áudio e a referência são ordenadas e comparadas por meio de um *scatterplot*. Se há correspondência, uma linha diagonal surge.

CORREPONDÊNCIA DE DIGITAIS

Comparação de soluções

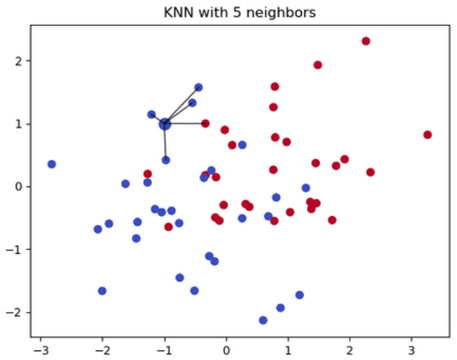

Fig 21. Exemplo de top-5 vizinhos mais próximos. Fonte: https://importq.wordpress.com/.

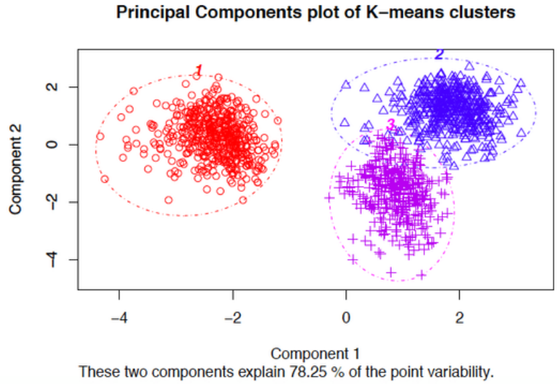

Fig 22. Exemplo de análise de cluster por *K-means*. Fonte: https://www.mailman.columbia.edu/research/population-health-methods/cluster-analysis-using-k-means.

Fig 23.Esquema completo extraído do artigo "Now Playing: Continuous low-power music recognition".

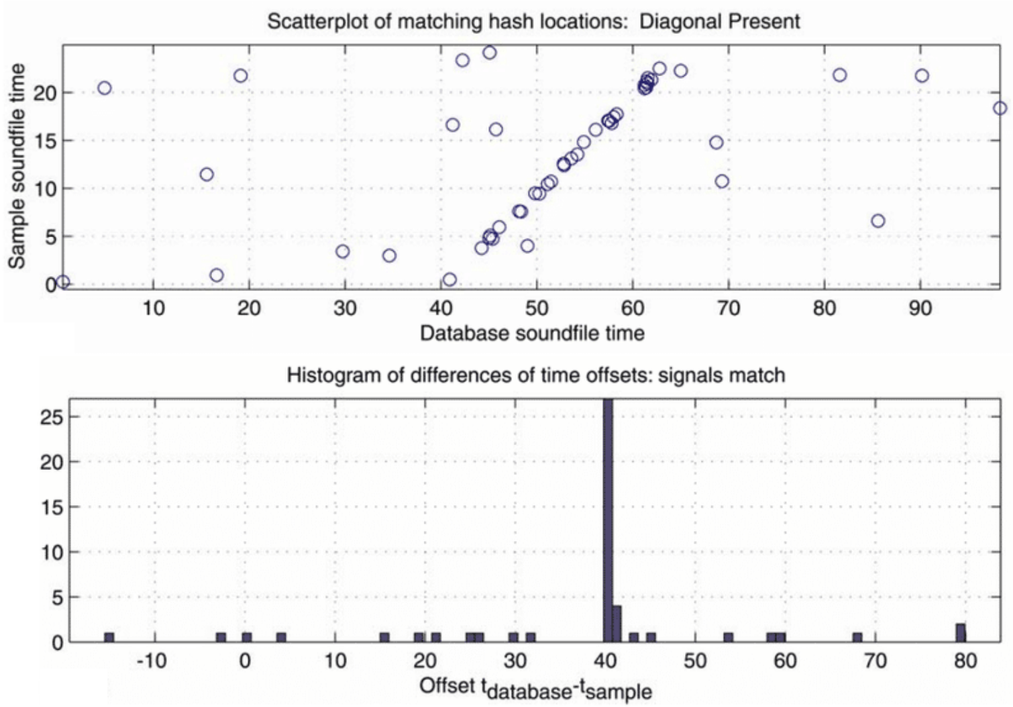

Fig 24. Correspondência de digitais extraída do artigo "An Industrial-Strength Audio Search Algorithm". A pontuação é o número de pontos correspondentes no pico do histograma.

AVALIAÇÃO DE RESULTADOS

Comparação de soluções

Para avaliar o desempenho do detector de música, ele foi testado em pedaços curtos de áudio (16s-40s) com regiões de músicas de um conjunto de teste com 450h de áudio, 12k instâncias. O dataset de teste contém vários ruídos de fundo e apresenta-se em volumes variados, desde imperceptíveis pelos humanos até muito altos. Foi necessário um *trade-off* entre alto *recall* (sempre acionar quando uma música está tocando) e evitar falsos positivos. Aceitando uma taxa de falsos positivos de cerca de uma vez a cada 20 minutos em um áudio não silencioso, foi mantido um *recall* de 75,5%.

No total, o Now Playing consome cerca de 0,9% da bateria do Pixel 2 por dia.

O algoritmo funciona bem com níveis significativos de ruído e distorções. Uma propriedade da técnica de realizar o histograma do gráfico de dispersão é que as descontinuidades são irrelevantes, garantindo "imunidade" à interferências.

Foi realizado um teste em 250 amostras de áudio de comprimentos e níveis de ruído variados com um banco de 10.000 músicas populares. Segmentos de áudio de 15, 10 e 5 segundos foram tirados do meio de cada música do banco de teste. Uma amostra de ruído foi gravada em um bar barulhento para simular condições reais e adiciona a cada segmento.

AVALIAÇÃO DE RESULTADOS

Comparação de soluções

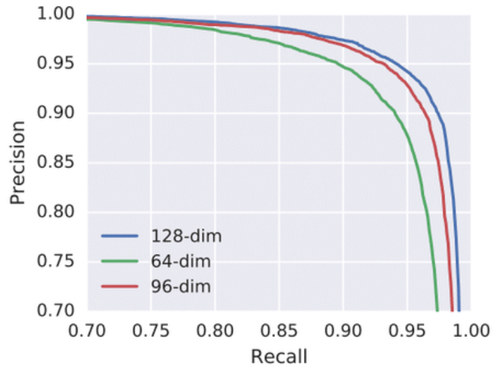

Fig 25. Dado o espaço limitado disponível, foi necessário encontrar o tamanho ideal de digital. Foi selecionado tamanho de 96 dimensões, dada eficiência e espaço (não tão longe de 128). O desempenho da NNFP e o algoritmo de correspondência foram avaliados usando digitais de 64, 96 e 128 dimensões em um conjunto de 20k segmentos de 8s de 10k músicas diferentes.
Extraído do artigo "Now Playing: Continuous low-power music recognition".

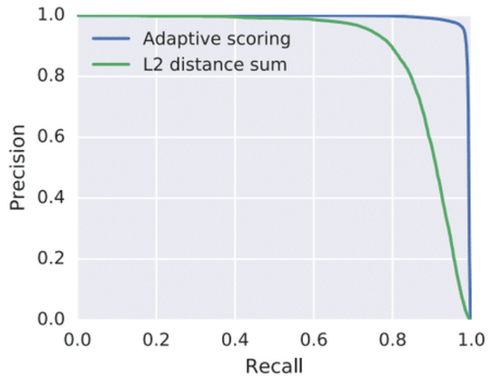

Fig 26. Comparação de desempenho do algoritmo de correspondência.
Extraído do artigo "Now Playing: Continuous low-power music recognition".

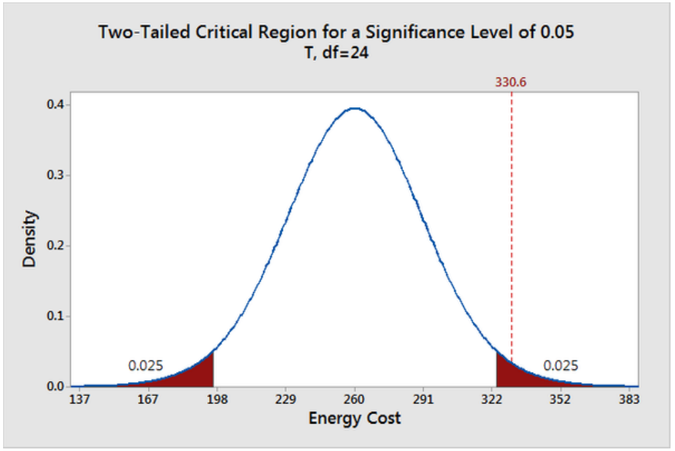

Fig 27. Exemplo de nível de significância. Um histograma das pontuações de trilhas que combinam incorretamente é gerado. O número de trilhas no banco de dados é levado em consideração e uma função de densidade de probabilidade da pontuação da trilha de correspondência incorreta com maior pontuação é gerada. Em seguida, uma taxa de falso positivo aceitável é escolhida

Fonte: https://blog.minitab.com/blog/adventures-in-statistics-2/understanding-hypothesis-tests-significance-levels-alpha-and-p-values-in-statistics

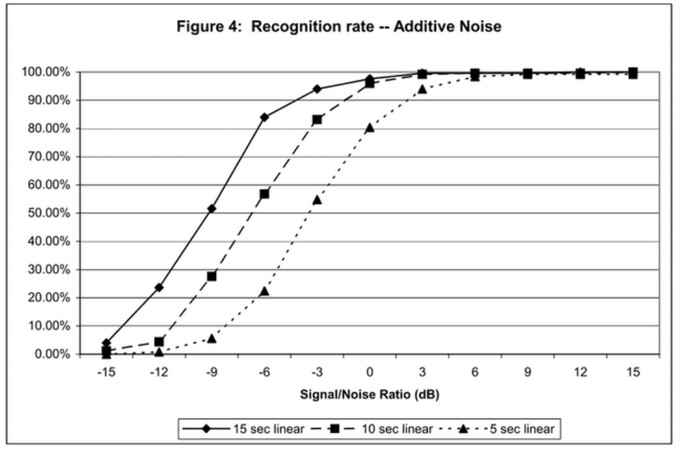

Fig 28. A taxa de reconhecimento cai para 50% para amostras de 15, 10 e 5 segundos a aproximadamente -9, -6 e -3 dB SNR, respectivamente.
Extraído do artigo "An Industrial-Strength Audio Search Algorithm".

O serviço pode encontrar uma faixa correspondente para uma amostra de áudio altamente corrompida em algumas centenas de milissegundos.

CONCLUSÃO

Comparação de soluções

Ambas soluções apresentam resultados muito bons, robustos ao ruído e com boa capacidade de reconhecimento de música. A solução da Google vai além e explora a capacidade desse processo funcionar automaticamente sem consumir muito recurso do dispositivo. O Shazam, por outro lado, com uma abordagem muito simples, se comparada com a da Google, atinge também ótimos resultados em um processo de identificação bem rápido; além disso também mostra que *hash* pode ser usado para simplificação de representação.

REFERÊNCIAS

Material sobre fingerprinting
https://medium.com/intrasonics/a-fingerprint-for-audio-3b337551a671
https://blog.chirp.io/audio-fingerprinting-what-is-it-and-why-is-it-useful/
http://mtg.upf.edu/files/publications/0e9cd9-Springer05-pcano.pdf

Material sobre espectograma
https://blogs.bl.uk/sound-and-vision/2018/09/seeing-sound-what-is-a-spectrogram.html

Material adicional sobre o Shazam
https://medium.com/@treycoopermusic/how-shazam-works-d97135fb4582
http://coding-geek.com/how-shazam-works/
https://www.youtube.com/watch?v=WhXgpkQ8E-Q
https://www.youtube.com/watch?v=Q4LYys9v9Ko

Material adicional sobre o Now Playing
https://www.xda-developers.com/how-google-pixel-2-now-playing-works/

Material sobre escala Mel
https://pdfs.semanticscholar.org/15ce/b6976fbf7b8fd2d10fd0b86c825ba0ceeea3.pdf
http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/logan00mel.pdf
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

Material sobre redes convolucionais
https://www.deeplearningbook.org/contents/convnets.html
https://www.esantus.com/blog/2019/1/31/convolutional-neural-networks-a-quick-guide-for-newbies
https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215
https://arxiv.org/pdf/1504.03410.pdf (**Divide-and-encode**)
https://arxiv.org/pdf/1503.03832v3.pdf (**Triplet loss function**)

A Spectrogram-based Audio Fingerprinting System For Content-based Copy Detection
https://link.springer.com/epdf/10.1007/s11042-015-3081-8?author_access_token=0iUys5eSYthhMUEAklgBYve4RwlQNchNByi7wbcMAY6XUyXfjYhf8fw0stKjGiJu0nCPVysWrcTAHjNo0NO3RtBj2FkTn6m8nIPVQNJ4xJL7w8tlZ0-W0k9psRCopxK_0AIn4iijrvlt792MT6SjDw%3D%3D

Known-artist Live Song Identification Using Audio Hashprints
http://pages.hmc.edu/ttsai/assets/LiveSongID_TMM17.pdf