UNIVERSIDADE DE ÉVORA

ESCOLA DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE INFORMÁTICA

Técnicas de Deep Learning para Detecção de Eventos em Áudio

Treino de modelos acústicos a partir de sinais puros

Sara Marina Albino Rijo

Orientador | Luís Miguel Mendonça Rato
Co-Orientador | José Miguel Gomes Saias

Mestrado em Engenharia Informática

Dissertação

Évora, 2017

UNIVERSIDADE DE ÉVORA

ESCOLA DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE INFORMÁTICA

Técnicas de Deep Learning para Detecção de Eventos em Áudio

Treino de modelos acústicos a partir de sinais puros

Sara Marina Albino Rijo

Orientador | Luís Miguel Mendonça Rato
Co-Orientador | José Miguel Gomes Saias

Mestrado em Engenharia Informática

Dissertação

Évora, 2017

Para ti, tia.

Prefácio

Este documento contém uma dissertação intitulada "Técnicas de Deep Learning para Detecção de Eventos em Áudio", um trabalho da aluna Sara Marina Albino Rijo, estudante de Mestrado em Engenharia Informática na Universidade de Évora. O orientador deste trabalho é o Professor Doutor Luís Rato e o co-orientador o Professor José Saias, ambos do Departamento de Informática da Universidade de Évora. A autora do trabalho é licenciada em Engenharia Informática, pelo Instituto Superior Manuel Teixeira Gomes. A presente dissertação foi entregue em Novembro de 2017.

Agradecimentos

A concretização deste projecto só foi possível graças aos contributos de diferentes pessoas. Assim, desejo manifestar os meus sinceros agradecimentos a todos os que contribuíram direta ou indiretamente para a realização deste projecto, nomeadamente:

Ao Prof. Luís Rato e Prof.José Saias pela paciência, orientação deste projecto e apoio recebido ao longo deste processo de formação académica, contribuindo com as suas opiniões e o seu saber que permitiram discutir ideias, partilhar pontos de vista e definir metodologias essenciais para a concretização deste trabalho.

Aos verdadeiros impulsionadores e mentores deste projecto, Dr.António Roldão e Mr.Sumon Sadhu, pela oportunidade de integração na muse.ai, pela transmissão de conhecimentos e saberes na busca de soluções inovadoras e pelo desafio proposto de ir para além do estado da arte já existente nesta área.

Ao corpo docente do curso de mestrado em Engenharia Informática da Universidade de Évora pelos conhecimentos transmitidos.

Aos meus amigos, que souberam estar sempre presentes.

Ao meu namorado e colega, Gonçalo Luís, pelo incentivo e apoio constantes, mesmo nos momentos mais difíceis.

Aos meus pais e restante família, pela sua paciência, dedicação e encorajamento, suporte essencial para a concretização deste projecto, bem como de todo o meu percurso académico.

A todos, **MUITO OBRIGADA**

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

DNN Deep Neural Network

GMM Gaussian mixture model

HMM Hidden Markov Models

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IA Inteligência Artificial

MFCC Mel-frequency cepstrum

MPEG Moving Picture Experts Group

PCA Principal Component Analysis

RNN Recurrent Neural Network

SVM Principal Component Analysis

t-SNE t-Distributed Stochastic Neighbor Embedding

WAV Waveform Audio File Format

XML EXtensible Markup Language

Sumário

O vídeo é atualmente um dos recursos mais utilizados e em constante expansão no mundo digital, sendo que em 2019 será responsável por 80% do tráfego mundial de Internet. Neste panorama, surgiu a problemática da incapacidade humana e (até agora) tecnológica para descrever, interpretar e interagir com este elevado volume de dados multimédia. Assim, têm sido desenvolvidos esforços para encontrar formas de automatizar e melhorar o processo de análise de conteúdo de vídeo e consequentemente melhorar a eficiência, usabilidade e acessibilidade dos vídeos armazenados por milhões de pessoas em todo o mundo. Este processo pode focar-se na análise do som e imagem em simultâneo ou independentemente. Esta dissertação descreve a concretização de um projecto de análise de som, que consistiu no desenvolvimento e implementação de um sistema de classificação de áudio utilizando uma abordagem baseada na utilização da waveform do som e redes neuronais, excluindo a convencional fase de extração de características habitualmente utilizada em aprendizagem máquina. Esta metodologia permite ultrapassar as limitações ao nível da ineficência temporal de abordagens típicamente utilizadas. Sendo o áudio um componente de relevo no vídeo, torna-se assim possível detectar e distinguir elementos de destaque, como por exemplo as partes mais cómicas, faladas ou musicais. Elaborou-se também um conjunto de dados de sons exclusivamente para o projecto, tendo em vista colmatar a utilização de dados de 'laboratório', isto é, obtidos em ambientes controlados, que induzem a falsos positivos e não representam a estocacidade do som no 'mundo real'. Os resultados obtidos foram bastante satisfatórios, tanto pela rapidez do processo como pela sua precisão, que atingiu taxas de reconhecimento dos sons na ordem dos 90%.

Palavras chave: Redes Neuronais Profundas, Classificação de Som, Aprendizagem Máquina, Dados Reais, Alta Dimensionalidade

Abstract

Deep Learning for Sound Event Detection

Training acoustic models on waveforms

Video is currently one of the most used media resources, it's use is expanding worldwide and predictions point that by 2019 video will be responsible for 80% of the world's internet traffic. With this in mind the problem of the impossibility for humans and technology (so far) to describe, interpret and interact with this amount of multimedia data rouse. Thus efforts have been made to find ways to automate and improve the video content analysis process and consequently better the efficiency, usability and accessibility of video stored by millions of people arround the world. This process can focus on the simultaneous analysis of image and sound or independently. This dissertation describes a project that consisted in the development and implementation of an audio classification system using an emerging approach based on the use of the sound waveform and neural networks, excluding the convetional feature extraction phase normally used in machine learning. As the audio is an important component of video, this system allows detection of important elements like the funnyest parts, where is speech and where is music. The results were very satisfactory, both in terms of processing speed and precision, that reached classification scores arround 90%.

Keywords: Deep Neural Networks, Sound Classification, Machine Learning, Real Data, High Dimensionality

1

Introdução

A estrutura curricular do mestrado em Engenharia Informática da Universidade de Évora, engloba no 3º e 4º semestre a elaboração de uma dissertação que pretende ser o culminar do processo de aprendizagem e trabalho académico realizados durante o curso, constituindo uma aplicação dos conhecimentos e capacidades adquiridas pelos estudantes ao longo do seu percurso académico.

Este relatório, visa apresentar o projecto – "Técnicas de Deep Learning para Detecção de Eventos em Áudio"– que consiste num sistema de aprendizagem máquina treinado para detecção e classificação de eventos de áudio, nomeadamente fala, gargalhadas, aplausos, gritos, música e silêncio. O sistema disponibiliza também uma interface gráfica que permite ao utilizador inserir um *link* de youtube, fazer a classificação do aúdio presente no vídeo introduzido e visualizar os resultados obtidos.

1.1 Enquadramento e Motivação

O crescente uso da Internet associado à rápida evolução e utilização do vídeo neste meio está a originar um novo paradigma tecnológico. Este paradigma, relaciona-se por um lado com a necessidade de lidar com

grandes quantidades deste tipo de dados multimédia, muitas vezes de conteúdo duplicado e com descrições pouco claras e precisas e por outro com a necessidade de melhorar a experiência dos utilizadores no acesso, filtragem e pesquisa dos mesmos.

A tendência é que no futuro todas as pessoas usem o vídeo como forma de interagirem, de se informarem e ocuparem o seu tempo. E o futuro está a chegar depressa, visto que os números que traduzem a quantidade de vídeo a entrar a cada minuto para a world wide web não pára de crescer. Basta pensarmos sobre a quantidade de dados capturados pelas tecnologias que nos rodeiam (desde smartphones, drones, sistemas de vigilância, etc.) e que são posteriormente carregados para sites públicos como o YouTube, Facebook, Vine, Snapchat, entre outros.

Desse modo, nas redes sociais os vídeos estão cada vez mais a ocupar o lugar de fotografias. Segundo Zuckerberg (2014), "In five years most of Facebook will be vídeo". Outras estatísticas indicam também que em 2020 o vídeo online será responsável por 80% do tráfego mundial de Internet, sendo que grande parte dos acessos serão feitos através dispositivos móveis [Cis16].

A questão que se coloca é: Como poderemos extrair valor de todo esse vídeo da mesma forma que o fazemos a partir de outras fontes de "big data"? Para avançar, é necessário concentrar atenções no desenvolvimento de soluções de software que permitam uma análise 'inteligente' do conteúdo de vídeo que nos poderá dar respostas e conhecimentos acerca desses dados. Nos dias de hoje, a extracção de conhecimento de conteúdo de vídeo é um processo maioritariamente manual, feito através da observação humana - basicamente, um ser humano visualiza um vídeo e anota o que está a acontecer, tornando a tarefa morosa, algo subjectiva e por vezes imprecisa.

A análise inteligente de conteúdo de vídeo é o processo que permite analisar vídeo automaticamente, de forma a detectar e determinar eventos temporais e espaciais. Embora o conteúdo visual seja, em grande parte dos casos, a maior fonte de informação em ficheiros de vídeo, também podemos encontrar informação bastante valiosa em componentes como o áudio. Segundo Dimitrova et al. [DZS+02], uma análise combinada e comparativa destes componentes é muito mais eficaz na caracterização do vídeo e na obtenção de uma análise mais precisa e completa. No entanto, a análise de som tem sido pouco explorada e pouco considerada neste domínio.

Uma das motivações inerentes ao presente trabalho está relacionada com o facto de considerar que a análise de som é um processo crítico e com enorme potencialidade na análise de conteúdo de vídeo. Basta pensarmos num exemplo simples: se estivermos a ouvir um programa de TV e não o podermos visualizar, conseguimos perceber em grande parte o seu conteúdo e reter as suas ideias chave. Por outro lado, se estivermos apenas a visualizar esse mesmo programa, torna-se bastante difícil decifrar qual o seu conteúdo.

Desta forma, torna-se desafiante explorar uma nova perspectiva desta problemática e ainda mais intersectá-la com conceitos de inteligência artificial, *machine learning* e *data mining*. A análise de som, mais especificamente a classificação de eventos recorrendo a algoritmos de machine/deep learning, é uma tarefa que tem, à partida, alguns desafios inerentes, dos quais se podem citar:

- Alta resolução temporal dos sinais de áudio (tipicamente 44100 amostras por segundo);
- Inexistência de conjuntos de dados representativos com eventos de áudio do 'mundo' real;
- Comportamento estocástico dos diferentes tipos de sons a classificar;
- Dificuldade de aplicação dos algoritmos tradicionais de machine learning a problemas de alta resolução temporal (também chamado "curse of dimensionality")

Do ponto de vista prático, pretende-se que este trabalho seja um contributo ao nível do estado da arte da análise de som, perspectivando-se que a informação obtida neste processo possa também ser fundida com dados adicionais, nomeadamente provinentes da análise de conteúdo de imagens, permitindo obter resultados promissores na combinação destes métodos para processamento de vídeo.

1.2 Objectivos

A finalidade do presente trabalho é demonstrar a aplicabilidade e importância da análise de som para o processo de análise automática de conteúdo de vídeo, apresentando uma metodologia inovadora baseada na classificação de diferente tipos de sons, utilizando redes neuronais como algoritmo classificador.

Foram analisados diversos tipos de métodos para classificação de eventos de som propostos por diversos autores, de forma a perceber que tipo de metodologia deveria ser desenhada para que a mesma tivesse viabilidade de aplicação num produto pronto a utlizar pelo utilizador comum, isto é, que permita obter resultados precisos, em dados 'do mundo real' (sem qualquer tratamento), e de forma relativamente rápida.

O objectivo geral do trabalho foi desenvolver um sistema de classificação baseado apenas na forma de onda do áudio, sem recorrer a qualquer tipo de extracção de características. Os valores que definem o sinal foram a entrada directa da rede neuronal artificial. As classes de eventos seleccionadas para distinção através desta rede foram: voz, música, gargalhadas, aplausos, gritos e silêncio.

A abordagem global definida levou à estruturação de um plano de objectivos específicos, tendo em vista colmatar os desafios identificados inicialmente, inerentes à problemática:

- O primeiro foi o desenvolvimento de uma aplicação para extracção automática de um conjunto de dados 'real', baseado em sons de vídeos já disponíveis na Internet;

- O segundo foi a criação de uma aplicação que possibilitasse a visualização e tratamento dos dados do conjunto de dados extraído;

- O terceiro foi o desenvolvimento de uma interface para que o utilizador final pudesse testar a classificação de som nos seus próprios vídeos;

- O quarto foi o desenvolvimento do sistema de classificação de eventos, baseado em redes neuronais artíficiais.

De sublinhar que cada um destes objectivos específicos constituiram diferentes contributos originais do presente trabalho, considerando-se o desenvolvimento do sistema de deep learning para classificação de eventos sonoros o de maior relevância.

Outro dos propósitos desta dissertação foi colmatar a principal falha das abordagens tradicionais - a ineficiência temporal - mantendo os níveis de exatidão padrão. Espera-se ainda que os conhecimentos resultantes deste estudo contribuam para reforçar a importância do desenvolvimento de sistemas de análise de conteúdo de vídeo.

1.3 Estrutura do Documento

A dissertação tem a seguinte estrutura:

No capítulo 2 são apresentados os fundamentos teóricos necessários para a compreensão deste projecto.

No capítulo 3 é apresentado o estado da arte, referenciando trabalhos e investigações anteriormente realizadas na área de detecção/classificação de eventos.

O quarto capítulo refere-se à metodologia utilizada para implementação do sistema, a nível de front-end e back-end, elaboração do conjunto de dados e treino dos algoritmos classificadores.

O quinto capítulo apresenta uma breve análise dos resultados obtidos, demonstrando a performance dos vários algoritmos classificadores utilizados, relacionando os mesmos com a iteração de vários parâmetros associados ao problema em causa.

O sexto capítulo apresenta as conclusões retiradas do projecto assim como as melhorias que poderão ser efectuadas no sistema.

2

Conceitos Fundamentais

De modo a compreender o sistema implementado, este capítulo apresenta as teorias que suportam temas que constituem o cerne deste projecto, tal como a análise de som em computador, detecção de eventos sonoros, machine learning e deep learning, redes neuronais e dados de alta dimensionalidade.

2.1 Som

O som é um fenómeno físico causado pela vibração da matéria, sendo que esta vibração provoca alterações de pressão no ar que rodeia a matéria em causa. Estas alterações de pressão propagam-se através do ar e quando atingem o ouvido humano, perturbam o tímpano onde o som é captado e posteriormente interpretado no cérebro. A vibração tem uma origem, denominada fonte sonora, que força o ar a vibrar (e.g. altifalantes ou laringe).

Sendo o áudio componente base do presente trabalho, torna-se necessário revelar que, embora totalmente relacionado com o som, segundo o dicionário online de Português (Priberam), áudio define-se como o "processo de gravação, reprodução, transmissão ou recepção de som". O som é a base do áudio, que é a componente audível deste (20Hz-20kHz) [Bar12].

A necessidade de gravar e reproduzir som, surgiu muito antes do aparecimento do computador digital e como tal era armazenado em suportes analógicos (ex: discos de vinil e cassetes), porém com a evolução tecnologica e o aparecimento do microprocessador, a digitalização do áudio tornou-se possível. Para que o computador seja capaz de 'ouvir' som é necessário discretizar a perturbação sonora através de conversores de analógico para digital (CAD). Por sua vez, para a reprodução do áudio a partir de suporte digital são necessários conversores de digital para analógico (CDA).

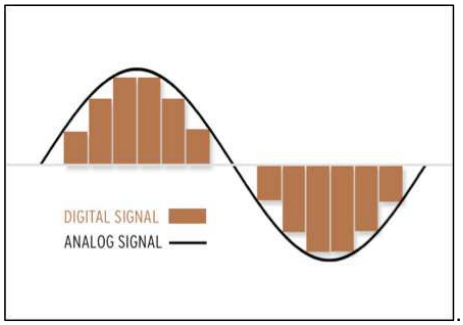


Figura 2.1: Som Digital vs Som Analógico

Quando temos que processar e analisar áudio digital, devemos considerar dominar alguns conceitos e metodologias, que serão apresentados seguidamente.

Taxa de amostragem: É o número de amostras (pontos) existentes por segundo num sinal contínuo, ou seja, os pontos que o tornam discreto. Por exemplo, uma taxa de amostragem de 44,1 kHz significa que, por segundo, existem 44100 amostras (igualmente espaçadas no tempo), o que é equivalente a existir uma amostra a cada 0,0227 ms. Quanto maior a taxa de amostragem, mais próximo o sinal é a sua representação original, sendo que o desejável é ter a melhor representação possível do sinal. No entanto há que considerar que altas taxas de amostragem resultam em ficheiros significativamente maiores, pelo que devemos encontrar um meio termo tendo em conta o objectivo de utilização do som. As taxas de amostragem mais comuns são apresentadas na seguinte tabela:

Taxa de amostragem	Qualidade do som
44 100 Hz	qualidade CD
22 000 Hz	qualidade rádio
8 000 Hz	qualidade telefone

Figura 2.2: Taxas de Amostragem

O teorema de **Nyquist–Shannon**, importante postulado directamente relacionado com o conceito anteriormente apresentado, demonstra que é necessário uma taxa de amostragem de no mínimo duas vezes o valor da frequência máxima do sinal de áudio para possibilitar a conversão para digital de todas as frequências nele contidas. Esta taxa de amostragem mínima necessária é também denominada taxa de Nyquist.

Resolução: A resolução é o número de bits disponíveis para representar o valor de cada amostra de áudio. Tal como a taxa de amostragem, quanto maior o número de bits, melhor é a exatidão do sinal digitalizado em relação ao sinal original, tornando o ficheiro maior em tamanho.Existem diversos standards como 16, 20 e 24 bits. Uma amostra de áudio com uma resolução de 16 bits permite 65536 valores de amplitude diferentes.

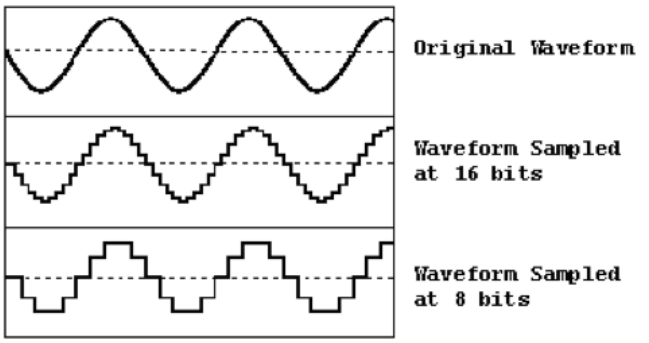


Figura 2.3: Resoluções de Áudio

Número de canais: O número de canais de um ficheiro de áudio representa o número de fontes independentes existentes e está diretamente relacionado com a espacialização do som. Na maioria dos casos são utilizados um canal (mono) ou dois canais (stereo), sendo que no último caso tenta-se simular a impressão de que o som ouvido é originado de duas direções diferentes (canal esquerdo e canal direito).

Forma de Onda: É representável através de um gráfico que demonstra o comportamento do som, em termos da sua amplitude (eixo Y), no domínio do tempo (eixo X).

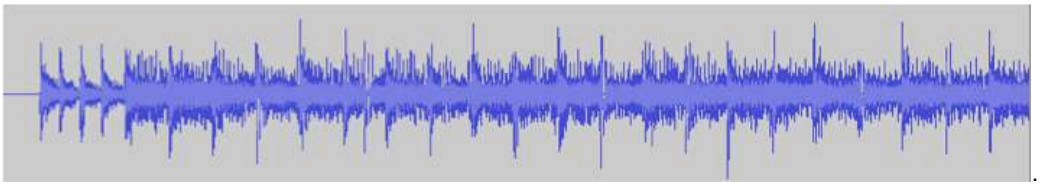


Figura 2.4: Forma de Onda de um Ficheiro de Áudio

Espectrograma: Gráfico que analisa a densidade espectral de uma onda sonora (eixo Y), num determinado período de tempo (eixo X). Para obter esta representação é necessário converter o sinal para o domínio da frequência, através da aplicação de uma Transformada de Fourier ao sinal original.

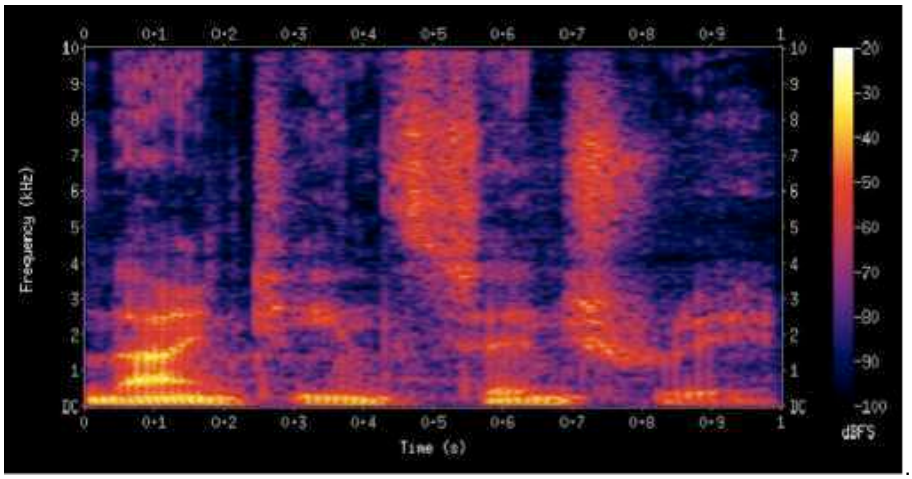


Figura 2.5: Exemplo de um Espectrograma

MFCC: São características de um som que são calculadas a partir da sua frequência e que têm em conta a percepção do ouvido humano, baseadas na representação perceptual dos 'pitches' do som. O pitch é

a percepção de quão alto (maior frequência) ou baixo (frequência mais baixa) um som é para o ouvido humano em relação à nota (frequência fundamental) de que deriva.

Filtragem de Áudio: É o processo de selecção ou supressão de certos componentes de frequência na quantidade desejada. Existem diversos tipos de filtros que podem ser aplicados a um sinal de áudio, entre eles:

- Filtros de Passa Alto: Atenuam as baixas frequências do sinal de entrada e deixam passar as altas frequências;
- Filtros de Passa Baixo: Atenuam as altas frequências do sinal de entrada e deixam passar as baixas frequências;
- Filtros de Passa Banda: São úteis para selecionar uma banda específica de frequências de interesse, rejeitando as restantes;

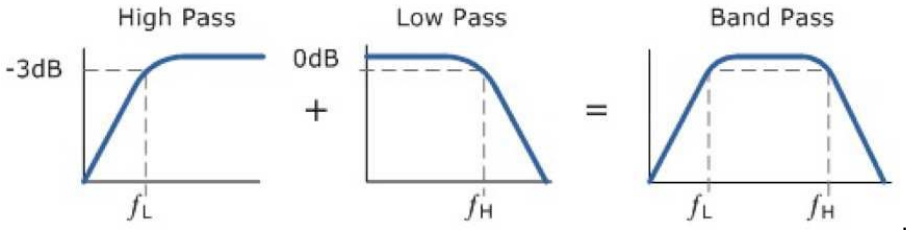


Figura 2.6: Filtros de Som

2.2 Detecção de Eventos de Som

Um evento de áudio é considerado um 'rótulo' que as pessoas usam geralmente para descrever um evento reconhecível numa determinada região do som. Tal rótulo permite que as pessoas entendam o conceito por detrás dele e o associem a outros eventos relacionados. Estes eventos possibilitam também fazer uma representação simbólica de uma determinada cena, isto é, podemos associar uma hora de ponta no centro de Lisboa a sons de buzinas, carros, passos, sirenes, vozes entre outros.

A detecção destes eventos é uma área que visa análise de sinais acústicos e a sua conversão em descrições dos eventos sonoros presentes na cena auditiva, sendo que a principal preocupação é reconhecer o instante de tempo exacto onde ocorrem os eventos, isto é, o tempo inicial e final em que os mesmos acontecem. É esta particularidade que distingue este tipo de sistemas de outros sistemas de reconhecimento de áudio já conhecidos, tais como os sistemas de reconhecimento de voz e de música [Cak14].

Os sistemas automáticos de detecção e classificação de áudio dividem-se em duas categorias:

- Monofónicos: para um dado intervalo de tempo apenas conseguem classificar um único evento sonoro;
- Polifónicos: detectam vários tipos de eventos, quando sobrepostos, no mesmo intervalo de tempo;

Este trabalho foca-se no desenvolvimento de um sistema monofónico.

A detecção de eventos de som pode ser utilizada em uma grande variedade de aplicações, incluindo indexação e recuperação de informação em bases de dados multimédia, monitorização em cuidados de saúde, e vídeo-vigilância. Além disso, os eventos detectados podem ser utilizados como representação de nível médio em outras áreas de investigação, como o reconhecimento de contextos de áudio, 'tagging' automático e segmentação de áudio.

2.3 Machine Learning vs Deep Learning

A área de Machine Learning (ML), pode ser vista como um campo da inteligência artificial (IA), onde o principal objectivo é a concepção e o desenvolvimento de algoritmos e técnicas que permitam que os computadores adquiram conhecimento de forma automática. Estes sistemas têm a função de analisar informações e generalizá-las, para extrair novos conhecimentos, representando uma promessa no que diz respeito à sua aplicação na sociedade, conduzindo a uma mudança importante nos 'ecosistemas digitais'.

Por sua vez, o Deep Learning (DL) pode ser visto como uma área que deriva do campo descrito anteriormente, que se concentra num subconjunto de ferramentas e técnicas mais específicas, que tentam modelar abstrações de alto nível dos dados. No processo original de ML a engenharia de características e de extracção de características são um processo chave bastante demorado, em que os dados originais são transformados num conjunto reduzido de características representativas. Em DL, estas duas etapas são 'ignoradas', sendo que esta fase é alocada automaticamente para o sistema de aprendizagem subjacente, podendo os algoritmos usar dados em bruto como entrada. Esta abordagem é considerada como o estado da arte actual da IA e tem-se mostrado a mais eficaz e com melhores resultados em diversos domínios, nomeadamente:

- Som (Reconhecimento de Voz)
- Texto (Classificação de Comentários)
- Imagens (Visão Computacional)
- Séries Temporarais (Dados de Sensores, Atividade Web)
- Vídeo (Detecção de Movimento)

No domínio de problemas de classificação de som, a utilização de algoritmos de DL, como redes neuronais profundas, permitem a utilização do sinal de som em cru como entrada da rede (valores de amplitude) em substituição das características como MFCC ou espectrogramas utilizadas na abordagem tradicional. Em sistemas em que o utilizador espera por um output na interacção com uma aplicação (como o desenvolvido no presente trabalho), a utilização de deep learning permite que os resultados sejam processados muito mais rapidamente, visto que grande parte do pré-processamento é dispensado, tornando a experiência mais apelativa aos utilizadores.

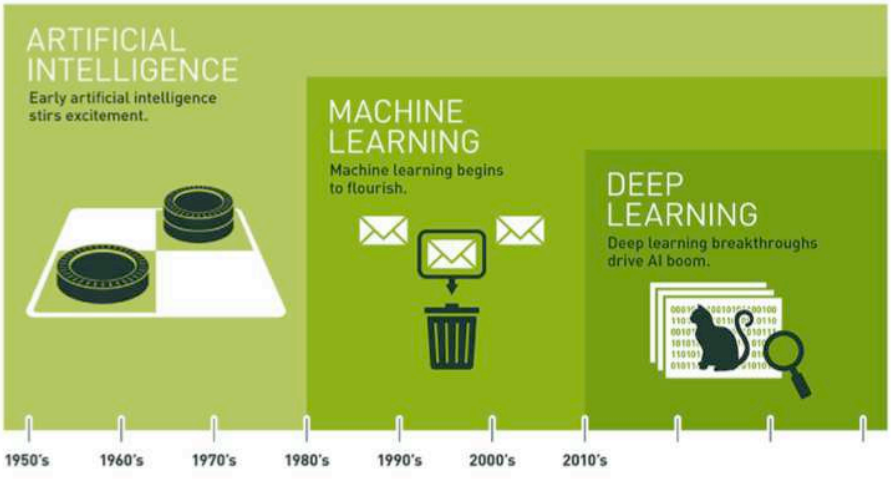


Figura 2.7: AI,ML e DL

2.4 Dados de Alta Dimensionalidade

Conjuntos de dados de alta dimensionalidade são aqueles que apresentam dados com centenas ou milhares de atributos/características. Este tipo de dados representam um problema na área de IA, visto que quanto maior for o número de dimensões, mais difícil é de organizar, classificar ou encontrar padrões num conjunto de dados.[Fer14]

Um fenómeno que se relaciona com este problema, chamado de 'curse of dimensionality', diz que à medida que a dimensionalidade aumenta, o desempenho do classificador só aumenta até que o número óptimo de atributos seja atingido [GTGVBA+15]. Aumentar ainda mais a dimensionalidade sem aumentar o número de amostras de treino resulta numa diminuição no desempenho do classificador.

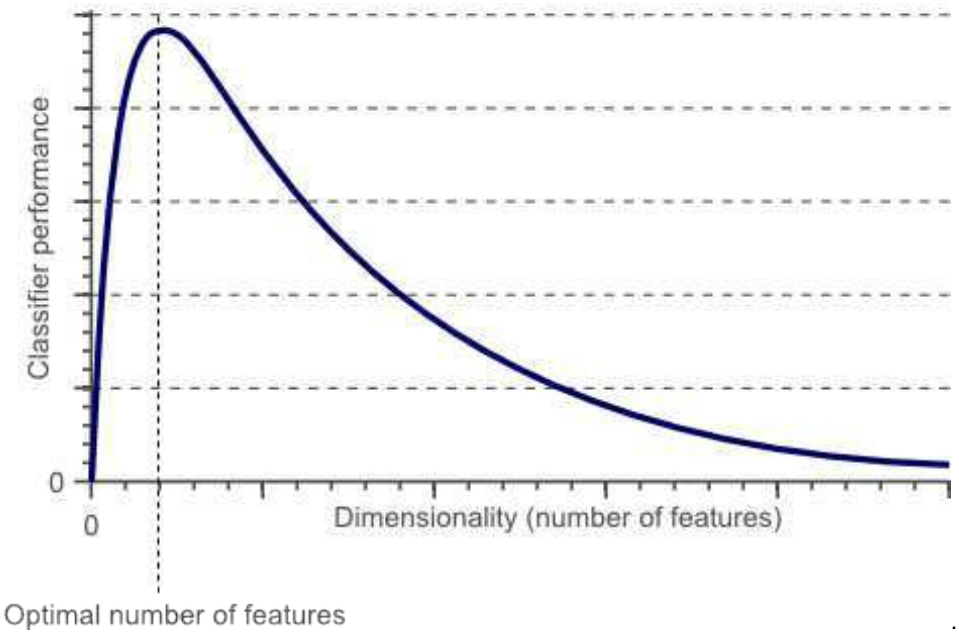


Figura 2.8: 'Curse of Dimensionality'

Este trabalho visa lidar com sinais sonoros puros, caracterizados pela sua alta resolução temporal, pelo que foi tido em conta este fenómeno com o intuito de dar respostas às seguintes questões:

- Qual a janela temporal de tamanho mínimo para que o evento possa ser interpretado correctamente e a performance do classificador maximizada?

- Como visualizar o conjunto de dados de forma a poder ter noção do seu comportamento e detectar possíveis *outliers* do conjunto de dados?

As respostas a estas perguntas serão apresentadas ao longo deste trabalho.

2.5 Algoritmos de Classificação em Deep Learning

Os métodos de aprendizagem automática constituem uma ferramenta poderosa que consegue realizar operações de otimização com a mínima intervenção humana [CWL06]. Neste trabalho, o método de aprendizagem escolhido foi as redes neuronais, de arquitecturas profundas. As redes neuronais são normalmente

utilizadas quando o volume de dados de entrada é demasiado grande, retornando em grande parte das vezes melhor exatidão face a outro tipo de abordagens convencionais.

Existem diversos tipos de redes neuronais que estão na vanguarda das aplicações mais recentes e com mais sucesso em Deep Learning tais como as redes profundas, autoencoders, redes convolucionais e redes recorrentes. O presente capítulo destina-se a apresentar os métodos de aprendizagem automática profunda utilizados e descrever as diferentes tipologias testadas.

A capacidade de aprender através de exemplos e de generalizar a informação aprendida é, sem dúvida, o atractivo principal da solução do problema através de redes neuronais. Para além disso, este tipo de algoritmo constitui uma boa alternativa à solução de problemas nos quais não se conhece, ou pouco se conhece, da 'física' do problema.[Hä07]

2.5.1 Redes Neuronais Profundas

Embora tenham existido múltiplas aplicações bem sucedidas utilizando redes neuronais 'superficiais' - com apenas uma camada intermédia - a tendência tem sido aumentar o número de camadas de forma a tentar obter melhores resultados e eliminar algumas fases de pré-processamento existentes, considerando que a complexidade da rede neuronal do cérebro humano é, de facto, bastante elevada e organizada por uma arquitectura profunda. Segundo alguns estudos, o processamento de informações no nosso cérebro é feito em várias fases, funcionamento este que é particularmente claro no sistema visual e no sistema auditivo.[Ben09]

Do ponto de vista computacional, uma Rede Neuronal Profunda (Deep Neural Network,DNN) é um rede neuronal artificial feed-forward que apresenta mais do que uma camada oculta entre os seus nós de input e output. Cada camada oculta, usa uma função de activação que mapeia o input proveniente da camada anterior para um estado escalar que é depois enviado para a camada seguinte, permitindo que a rede capture características ricas e não-lineares dentro do conjunto de dados. Quanto mais se avançar na rede neural, mais complexas são as características que a rede pode reconhecer, uma vez que os nós agregam e recombinam recursos da camada anterior.[HDY+12]

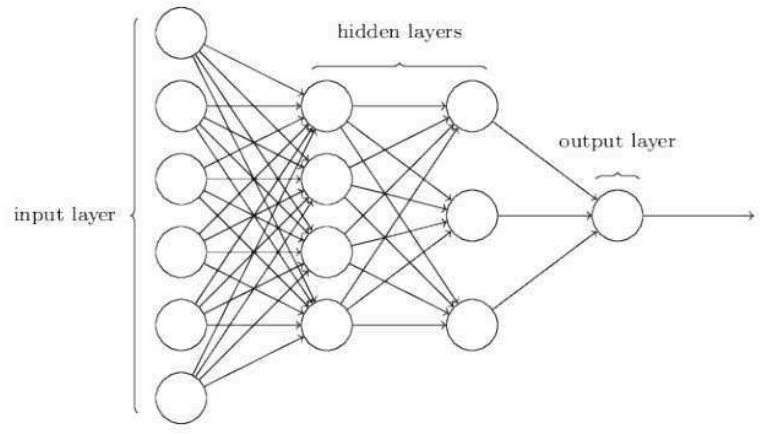


Figura 2.9: Rede Neuronal Profunda

As DNN's com muitas camadas ocultas e muitas unidades por camada são modelos muito flexíveis com um número muito grande de parâmetros. Esta característica torna estes algoritmos capazes de modelar

relações muito complexas e altamente não-lineares entre entradas e saídas, sendo esta propriedade bastante importante na modelação acústica.

2.5.2 Redes Neuronais Recorrentes

Redes Neuronais Recorrentes (Recurrent Neural Networks, RNN) constituem uma ampla classe de redes cuja evolução do estado depende tanto da entrada corrente quanto do estado atual. Essa propriedade (ter estado dinâmico) proporciona a possibilidade de realizar um processamento dependente do contexto e aprender dependências de longo prazo: um sinal que é fornecido a uma rede recorrente num instante de tempo t pode alterar o comportamento dessa rede num momento t+k, $k > 0$.[Bez16]

Segundo Campos [Cam16], uma rede recorrente pode ter conexões que voltem dos nós de saída aos nós de entrada, ou até mesmo conexões arbitrárias entre quaisquer nós. Deste modo, o estado interno de uma rede recorrente pode ser alterado conforme os conjuntos de dados de entradas que lhe sejam apresentados, podendo-se dizer que as redes têm memória. Esta propriedade é particularmente útil na solução de problemas que não dependam somente das entradas atuais, mas de todas as anteriores. Ao aprender, a rede recorrente alimenta as suas entradas através da rede, incluindo alimentação de dados de volta, das saídas até às entradas.

Este tipo de redes foram projectadas para reconhecer padrões em sequências de dados, como texto, genomas, caligrafia, vídeo, dados de sensores, entre outros, sendo um dos tipos de redes mais poderosos em problemas que lidam com áudio, devido as propriedades sensitivas que estes sistemas dinâmicos têm a lidar com sequências temporais.

Existem diversos tipos de arquitecturas de redes neuronais recorrentes,entre as mais conhecidas:

- Redes Hopfield
- Redes Hierárquicas
- Redes Recursivas Bi-Direccionais
- Redes Holman
- Redes Long Short Term Memory

As redes Long Short Term Memory (LSTM) são um tipo especial de RNN, capaz de aprender dependências a longo prazo. Foram introduzidos por Hochreiter & Schmidhuber em 1997 e têm sido optimizadas até aos dias de hoje, tendo resultados comprovados em muitos campos, nomeadamente em processamento de linguagem natural.

Estas redes contêm unidades LSTM em vez de, ou para além de, outras unidades de rede. Uma unidade LSTM é uma unidade da rede recorrente que se destaca por se lembrar de valores por períodos de tempo longos ou curtos. A chave para essa habilidade é que não é utilizada nenhuma função de ativação dentro de seus componentes recorrentes. Assim, o valor armazenado não é 'esquecido' iterativamente ao longo do tempo. As unidades LSTM são muitas vezes implementadas em "blocos"contendo várias unidades LSTM que, por possuírem uma estrutura interna, podem armazenar um valor por uma quantidade arbitrária de 'passos' de tempo.

Ao contrário das RNN tradicionais, uma rede LSTM é adequada para aprender através da experiência a classificar, processar e prever séries temporais quando há certos atrasos de tamanho desconhecido no tempo e intervalos entre eventos importantes. Esta propriedade, bastante adequada a eventos de áudio,

foi a chave para escolha desde algoritmo.

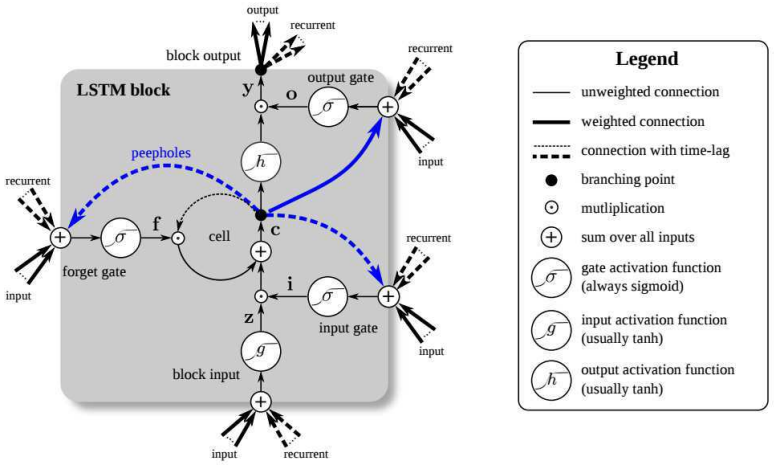


Figura 2.10: Funcionamento de um bloco LSTM

3

Estado da Arte

O "estado da arte" ou "estado do conhecimento" é considerado uma das partes mais importantes de um trabalho académico, uma vez que faz referência à produção académica e científica existente sobre o assunto pesquisado em determinado campo do conhecimento, em diferentes épocas e lugares, e de que forma e em que condições têm sido produzidas.

A partir desta compreensão, a comunidade científica refere-se ao seu contributo em termos de melhoria e desenvolvimento de novos postulados, conceitos e paradigmas.

Pesquisas recentes têm apresentado algumas das metodologias e resultados obtidos em trabalhos na área de detecção de eventos. A maioria dos métodos utilizados têm por base uma abordagem típica de machine learning, sendo a fase de extracção de características e a escolha do algoritmo classificador os pontos chave em comum entre todas elas. Todas estas investigações utilizam um esquema de aprendizagem supervisionada onde são utilizados eventos de áudio associados às respectivas legendas para treinar os classificadores. A abordagem utilizada nesta tese teve em conta os pontos fortes destes trabalhos e tentou modificar/inovar algumas das estratégias utilizadas de modo a superar algumas das suas limitações.

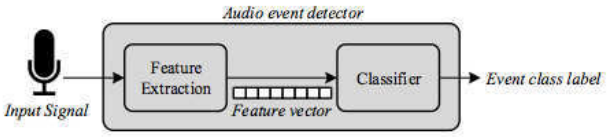


Figura 3.1: Abordagem Típica de Detecção de Eventos

3.1 Extracção de Características de Áudio

Geralmente, um dos primeiros passos num sistema automático de detecção de eventos é extrair características, isto é, identificar quais os componentes do sinal de áudio que são bons para identificar o seu conteúdo. Nesta fase é feito um procedimento baseado no processamento do sinal de entrada reduzindo também a dimensionalidade das amostras originais para que os algoritmos classificadores consigam lidar melhor com os conjuntos de dados. Este procedimento pode originar vectores de coeficientes, parâmetros ou até pequenas imagens, que representam características chave dos eventos de áudio [ASS16] .

De acordo com a revisão de literatura efectuada, é possível enumerar alguns dos métodos utilizados:

Motifs

Os motifs são padrões frequentes que podem ser encontrados em sons, e que têm uma importância especial visto que são 'a menor unidade estrutural que possui uma identidade temática" no contexto de um som. Grove e Larousse [Duc90] defendem que um motif pode ter aspectos harmónicos, melódicos e / ou rítmicos, acrescentando que "é mais frequentemente ser pensado em termos melódicos". Para a extracção destes "atributos" são utilizados algoritmos (e.g. MrMotif), que discretizam os sinais e procuram padrões nas sequências.

Em termos de áreas de aplicação, destaca-se o uso dos mesmos no campo da bioinformática, captura de movimentos, meterologia, vídeo-vigilância, entre outras. Apesar de ser pouco comum a sua utilização em problemas de detecção de eventos de som, Burred [Bur12] e Baptista [Bap15], utilizaram esta característica para fazer a classificação de eventos sonoros, tendo obtido uma exatidão de 70,8%.

Mel Frequency Cepstral Coefficients (MFCCs)

MFCCs são um recurso amplamente utilizado na área de processamento e classificação de áudio. Foram introduzidos por Davis e Mermelstein na década de 80, e desde então são considerados como estado da arte no que respeita à extracção de características.

Para obtenção destes coeficientes, o sinal começa por ser fragmentado em segmentos de tamanho fixo. Em cada segmento o sinal é considerado estacionário, sendo-lhe aplicada uma janela de Hamming. Por sua vez, cada janela é transformada para o domínio da frequência usando a Transformada de Fourier Discreta (DFT). É então calculada a magnitude dos coeficientes de Fourier e elevados ao quadrado passando por um banco de filtros Mel. De cada um dos filtros obtem-se um coeficiente que representa a energia ao qual se aplica a função logarítmica, resultando nos coeficientes log Mel-spectral. Para obter os coeficientes cepstrais calcula-se a Discrete Cosine Transform (DCT), através da qual se tenta descorrelacionar os parâmetros espectrais. Selecionam-se os primeiros treze coeficientes cepstrais, sendo os restantes descartados por conterem pouca informação relevante[Mat14]. A figura seguinte esquematiza este processo.

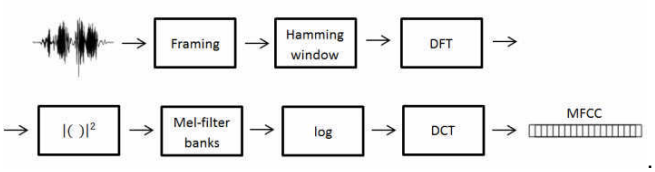


Figura 3.2: Processo de Extracção de MFCC

Esta caracteristica é uma das mais 'famosas' em trabalhos de investigação relacionados com processsamento de áudio. Na área de processamento de linguagem natural, Costa [Cos13] utilizou esta característica no seu processo de reconhecimento de vogais isoladas. Pye [Pye00] usou estes coeficientes no seu trabalho para classificar diferentes géneros musicais. Relativamente a classificação de eventos sonoros, Duarte [Dua16], Zhang [ZE16] e Gutierrez [GAFC+16] também utilizaram esta opção, apresentando em geral bons resultados com o seu uso, em conjunto com diferentes algoritmos classificadores.

Espectrogramas

Tal como indicado no capítulo 2, um espectrograma é uma representação gráfica do áudio, processada no domínio da frequência, onde as características espectrais podem ser visualizadas com detalhe. Normalmente o eixo horizontal representa a frequência ,o eixo vertical o tempo e a cor a amplitude dos espectros. Através de um espectrograma, é possível identificar um determinado som, visualizando o comportamento das suas linhas espectrais, que podem apresentar diferentes características como a variação, largura e estabilidade ao longo do tempo.

Estas imagens são normalmente gerada na fase de pré-processamento do áudio, com o intuito de servirem de input a diversos algoritmos classificadores, havendo uma tendência na combinação deste método com redes neuronais artificiais. Vários autores como Khunarsa [KLR10], Boddapati[Bod17] e McLoughlin et al. [MZX+15] utilizaram esta metodologia nos seus projectos de classificação de sons (até um máximo de 10 classes), sendo que os resultados obtidos apresentam taxas de exatidão acima dos 90%.

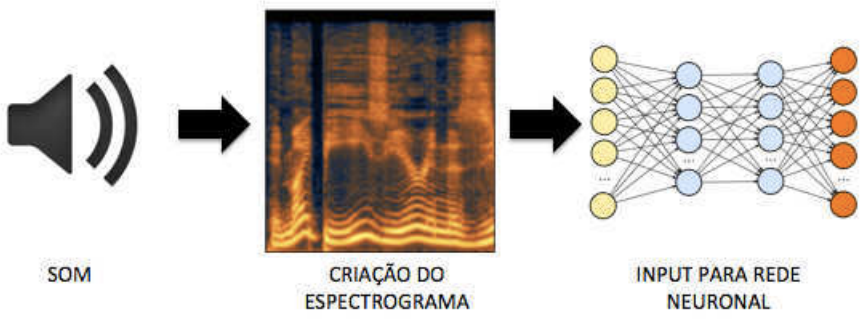


Figura 3.3: Processo Associado à Utilização de Espectrogramas

Tal como nos MFCC, uma das desvantagens do ponto de vista de implementação prática deste método é o tempo "gasto"para gerar estas características, visto que é necessário converter o sinal do domínio do tempo para o domínio da frequência e aplicar uma série de outros algoritmos intermédios computacionalmente complexos que geram as imagens/coeficientes.

3.2 Classificação de Eventos de Áudio

No que diz respeito à classificação de eventos são diversos os algoritmos utilizados, que dependem do tamanho e qualidade do conjunto de dados e do tipo de características extraídas, não existindo um consenso na comunidade científica quanto a qual o método mais eficaz e preciso.

Em [WRF14], foram testadas *Random Forests* e *Hidden Markov Models* (HMM) para detectar recursos semânticos em áudio, extraído de vídeos do youtube, sendo que o primeiro algoritmo obteve melhores resultados. Num outro trabalho, Zhang et al. [ZLL03] comparou o desempenho de diferentes classificadores num problema de classificação de diferentes amostras de áudio e concluiu que a implementação mais precisa era a que usava *Support Vector Machines* (SVM).

Vuegen et al. [VBK+13] propuseram uma metodologia baseada em MFCC's e *Gaussian Mixture Models* (GMM), cujo desempenho não foi satisfatório, segundo os autores devido ao tamanho relativamente pequeno do conjunto de dados utilizados e também pela grande variação de características existente em algumas das classes utilizadas.

Para além destes classificadores, destaca-se algum crescimento na aplicação de métodos de deep learning, ainda que no geral esta área seja pouco utilizada na área de classificação de áudio. As redes neuronais profundas (DNN) e convolucionais (CNN) são as mais utilizadas neste contexto, sendo que o mais comum neste tipo de implementações é a conversão dos sinais de áudio para o formato de imagem (espectrogramas) e depois a utilização destas redes para processar e classificar a imagem. McLoughlin [MZX+15] e Piczak [Pic15], transmitem a mesma ideia nos seus estudos de que, apesar do uso de DNN's ser uma solução viável, o uso de CNN's produz em geral melhores taxas de exatidão, sendo também uma mais valia em casos em que os dados de treino são limitados.

A utilização de sinais em bruto como input de algoritmos classificadores para detecção de eventos é uma estratégia que praticamente não foi observada em trabalhos publicados até hoje. No entanto, o centro de desenvolvimento da Google Deep Mind introduziu recentemente esta metodologia numa das suas pesquisas para reconhecimento da fala, tendo este estudo obtido grande impacto na comunidade de inteligência artificial, pelos resultados bastante promissores que foram obtidos [vdODZ+16].

4

Sistema e Metodologia

O sistema desenvolvido no âmbito desta dissertação permite a detecção de vários eventos no som de um vídeo fornecido pelo utilizador - aplausos, gargalhadas, música, voz, silêncio e gritos. Para fazer a classificação do áudio, foram utilizados dois tipos de redes neuronais artificiais – Redes Neuronais Recorrentes e Redes Neuronais Profundas – que foram treinadas e as suas performances comparadas, utilizando um conjunto de dados criado para o efeito. A abordagem utilizada é generalista, pelo que podem ser adicionados mais eventos de áudio ao conjunto de dados para utilização em projectos futuros.

Neste projecto pretendeu-se explorar novas técnicas de classificação 'directa' de áudio em bruto (raw audio), utilizando apenas o seu sinal como entrada da rede neuronal. O benefício da utilização deste tipo de dados, em vez da típica utilização de características extraídas previamente, são os seguintes:

- Os dados utilizados estão completos, isto é, contêm toda a informação que é relevante para distinguir entre classes;

- A extensão temporal exigida na fase de extracção de características nem sempre é aceitável para implementação em aplicações de utilização em tempo real.

O principal motivo pela qual esta abordagem é evitada por grande parte dos investigadores, relaciona-se com o problema da alta dimensionalidade já referido anteriormente neste trabalho.

4.1 Conjunto de Dados

Um dos passos mais importantes para obter experiências válidas no campo da inteligência artificial é o uso de conjuntos de dados obtidos no mundo real, isto é, não tratados ou conseguidos em ambientes controlados, já que os problemas com que esta área se depara nos dias de hoje envolvem cada vez mais dados reais e obtidos no ambiente que nos rodeia, que se caracterizam pela sua estocasticidade e inclusão de ruído. O uso de conjuntos de dados de 'laboratório' induzem a falsos bons resultados que posteriormente dificultam a sua aplicação efectiva em problemas do mundo real.

Para além disto, é importante que estes conjuntos de dados sejam tão equilibrados quanto possível (i.e, número similar de instâncias para cada classe) e tão completos quanto possível (que descrevam os eventos escolhidos de todas as formas possíveis). Por outro lado, a comunidade cientifica evoca também a importância de utilização de conjuntos de dados partilhados entre a mesma, quando existentes. Os conjuntos de dados cientificos existentes com sons relevantes para o presente trabalho (ex: aplausos, gargalhadas, e voz) como o ICSI Meeting Corpus, AMI Meeting Corpus, TWSES corpus ou a base de dados SEMAINE, foram obtidos em salas isoladas, isentas de ruídos, com microfones específicos instalados a distâncias exactas da plateia. De acordo com a problemática exposta no inicio deste capítulo, este tipo de dados não se enquadram nos objectivos delineados neste projecto por terem sido obtidos em condições controladas, pelo que não foram considerados para aplicação no presente trabalho.

Deste modo, foi necessário construir um conjunto de dados de raíz tendo em conta os eventos considerados para classificação:

- Aplausos
- Gargalhadas
- Música
- Voz
- Silêncio
- Gritos ('Cheering')

O conjunto de sons utilizados foi construído através da obtenção manual de sons de vídeos do youtube e de sons extraídos por um 'script' desenvolvido exclusivamente para o efeito.

4.1.1 Script para Extracção do Conjunto de Dados

O script para extracção do conjunto de dados foi desenvolvido em linguagem python tendo por objectivo criar um conjunto de dados de audio automaticamente a partir de vídeos legendados, que estão disponíveis na web, em plataformas como o youtube. Esta ferramenta consulta um ficheiro que contém links de vídeos, e constrói um conjunto de dados equilibrado de eventos de audio relevantes ao utilizador. Os eventos considerados relevantes são introduzidos pelo utilizador sempre que o script é iniciado.

Neste projecto foi utilizado um ficheiro HTML com todos os links de TED Talks publicados até ao inicio do ano 2016. A metodologia utilizada segue os seguintes passos:

1. Pedido input ao utilizador , referente a quais os eventos que quer para formar o conjunto de dados;

2. Stemming das palavras introduzidas pelos utilizadores de forma a normalizar e facilitar a consulta e detecção dos eventos nas legendas;

3. Consulta de link de vídeo a analisar;

4. Download do audio e legendas do vídeo;

5. Conversão das legendas para formato DFXP, de modo a facilitar a consulta e análise de campos chave, visto que este formato utiliza uma estrutura XML;

6. Verificação de legendas, nomeadamente de conteúdo textual e tempos do vídeo;

7. São cortadas e guardadas as parte do audio que contem o(s) evento(s) pretendido;

8. Se a parte analisada do audio não contiver nenhum evento relevante,apenas palavras no seu conteúdo, é guardada na pasta relativa ao evento 'voz';

9. Repetem-se os passos de 4 até 8 para cada link existente na lista de TED Talks.

O conjunto de dados gerado está sujeito a uma fase de detecção de outliers, visto que as legendas introduzidas em vídeos disponiveis na web nem sempre são precisas e considerando também que os diálogos presentes em cada vídeo podem conter palavras consideradas eventos.

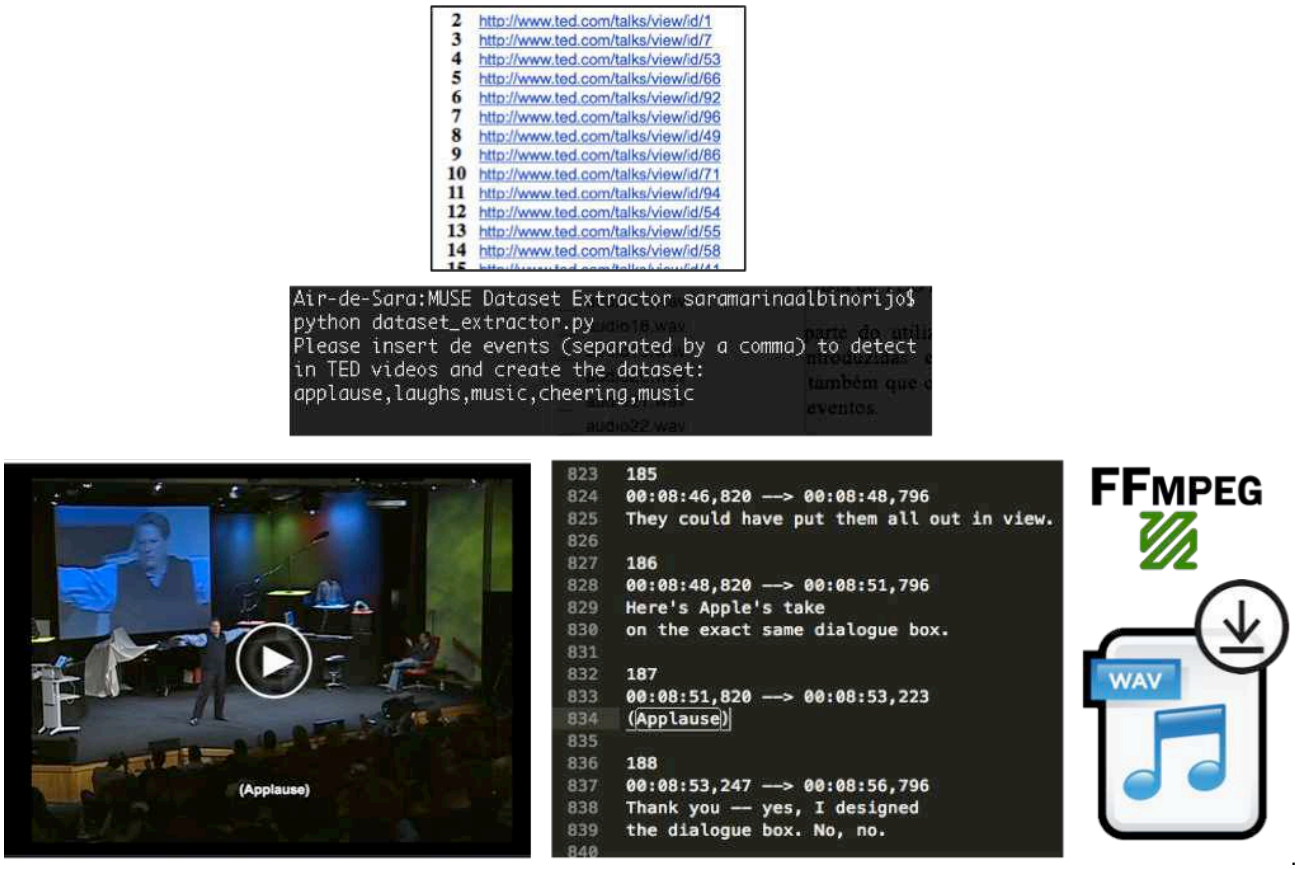


Figura 4.1: Método para Elaboração do Conjunto de Dados

4.1.2 Caracterização do Conjunto de Dados

Os sons que constituem o conjunto de dados foram extraídos através do script para extracção previamente apresentado, sendo que alguns sons foram obtidos manualmente através de diferentes vídeos disponíveis na

web, de modo a completar algumas categorias. Existem um total de 882 arquivos no conjunto de dados ,tendo cada uma dos ficheiros uma duração variável de 2 segundos até 50 segundos. A duração total dos ficheiros de áudio é de 48 minutos. O conjunto de dados equilibrado foi construído com cerca de oito minutos de exemplos de cada evento. O conjunto de dados consiste em arquivos de áudio em formato .wav com uma taxa de amostragem de 44100Hz. As categorias de eventos, consideradas para classificação foram:

- Voz - Vozes de Mulher + Vozes de Homem;
- Música - Instrumentos a Solo + Instrumentais de Música (Pop, Rock, Clássica, Jazz, Blues, Electro, Dance, etc.);
- Aplausos - Individuais + Audiência;
- Gargalhadas - Individuais + Audiência;
- Silêncio - Silêncio Puro + Ruídos de Fundo com Baixa Amplitude;
- Cheering - Cheering Talk Show's + Cheering de Eventos de Desporto;

Para o treino da rede neuronal cada ficheiro do conjunto de dados foi dividido em partes de 20ms. As partes de áudio de 20ms foram submetidas a uma análise de conteúdo que verifica se o som é silêncio ou não. Se a "chunk"for considerada maioritariamente silêncio, é rejeitada para treinar a rede neuronal. Esta fase foi adicionada à metodologia porque grande parte dos sons de voz, risos e aplausos extraídos contêm pequenas "pausas"ao longo do tempo, dependendo do ritmo de cada um destes eventos, que iriam induzir a rede em erro.

4.1.3 Visualização e Detecção de Outliers

Em desafios de machine learning os dados são uma peça essencial que faz parte da formulação de qualquer problema, independentemente da área de que advêm. A quantidade de dados colectados é cada vez maior, no entanto a dificuldade está em encontrar informação útil 'escondida' no seu conteúdo - existem diversos algoritmos que o fazem, mas até que ponto é que entendemos os resultados obtidos?

A visualização de dados é uma área de importância crescente em ML, e cujo objectivo é a construção de representações visuais de dados abstratos de forma a facilitar o seu entendimento e possibilitar a descoberta de novas informações e interacção com os dados.

Desta forma, foi delineada uma estratégia para possibilitar a visualização dos dados que fazem parte deste projecto. Analisando o conjunto de dados percebemos que o mesmo contém um elevado numero de 'chunks', sendo que cada 'chunk' de som é composta por 882 atributos, ou pontos de amplitude. Tal como referimos anteriormente, lidar com dados de alta dimensionalidade representa um grande desafio em IA, sendo também um problema no que diz respeito à visualização de dados.

O anexo A apresenta um estudo que teve como objectivo apurar se a visualização também é possível através de dados em bruto, explorando duas técnicas de redução de dimensionalidade : Principal Component Analysis e t-Distributed Stochastic Neighbor Embedding. Analisando os resultados pode-se concluir que neste caso, a fase de extracção de características -geração de espectrogramas- é absolutamente essencial para se conseguir uma visualização clara dos clusters. Para obtenção das visualizações em 2D e 3D, foi utilizada o Tensorflow, uma biblioteca open-source para machine learning desenvolvida pela Google.

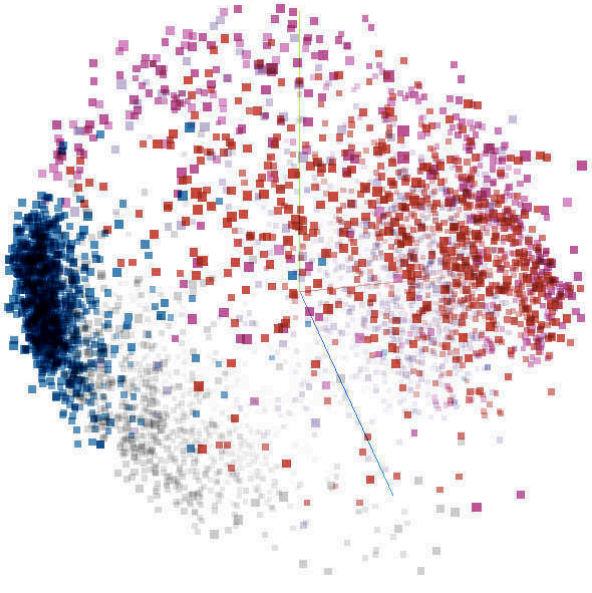


Figura 4.2: Visualização 3D do Conjunto de Dados

Na figura 3D apresentada podemos ver as amostras de cada classe pertencente ao conjunto de dados agrupadas em diferentes clusters, sendo que cada cor representa a classe dessas mesmas amostras.

Adicionalmente, foi também desenvolvida uma ferramenta para visualização das predições e reprodução dos sons de eventos contidos no conjunto de dados, sendo que a aplicação permite a seleccão de cada som, a sua reprodução, selecção do modelo de rede neuronal utilizado para predição e visualização de um gráfico de área respectivo às predições feitas para cada som seleccionado pelo utilizador. Visto que a extracção das várias 'chunks' que servem de entrada à rede neuronal é feita a partir de cada um dos sons que compõem o conjunto de dados, esta ferramenta permite visualizar quais os sons que representam outliers e devem ser descartados ,e quais os que devem ser considerados para input dos modelos optimizados. A figura seguinte apresenta um screenshot da aplicação descrita:

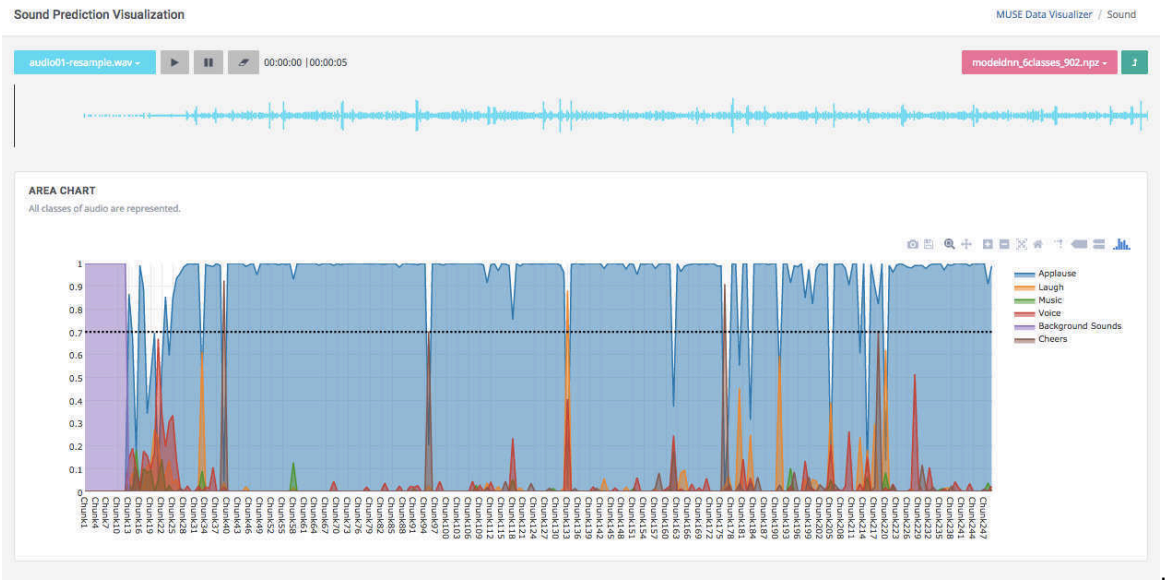


Figura 4.3: Ferramenta de Visualização de Predições

Um dos métodos mais utilizados para detecção de outliers que também foi considerado neste trabalho, é fazer uma visualização isolada do cluster relativo a cada evento, perceber quais os pontos que mais distam da sua centróide, e descartá-los. No entanto, visto que algumas das classes consideradas são bastante 'amplas', isto é, existem centenas de géneros de músicas ou timbres de voz, facilmente uma amostra mal representada num conjunto de dados se confunde com um outlier - acaba por ser um ponto externo ao cluster. Ao eliminarmos esses mesmos pontos, estamos a retirar a capacidade do modelo generalizar esse tipo de sons menos comuns.A figura seguinte demonstra um desses casos:

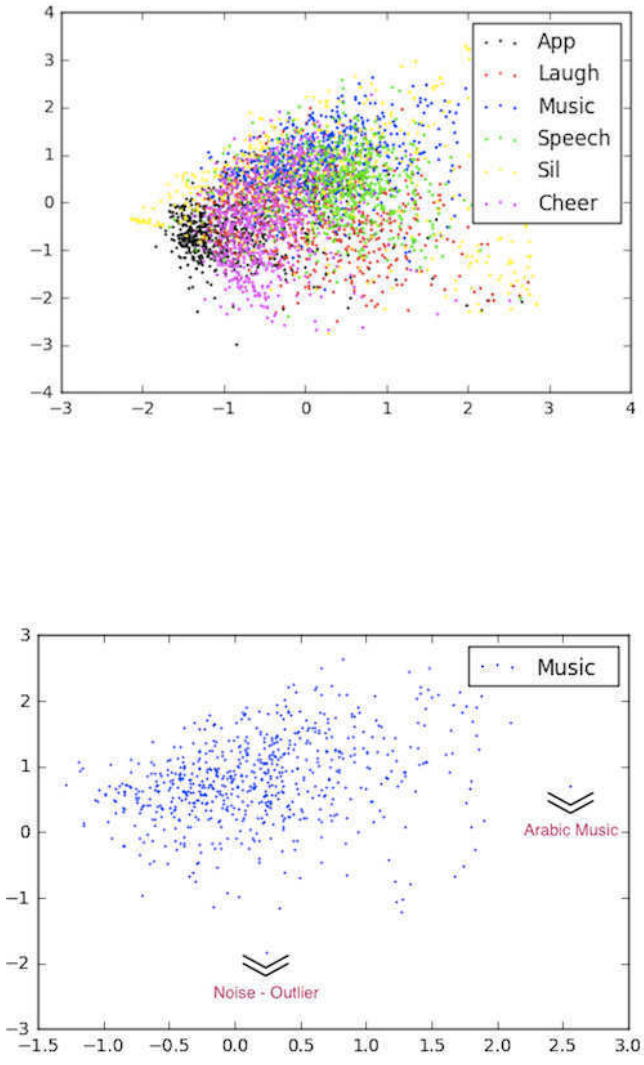


Figura 4.4: Detecção de Outliers pelo Método Tradicional

Desta forma, este método não foi considerado.

4.2 Metodologia do Sistema de Detecção de Eventos

A metodologia utilizada para a criação deste sistema contou com diferentes fases de desenvolvimento:

4.2.1 Pré-Processamento dos Dados

a) O conjunto de dados elaborado para este projecto contém ficheiros de áudio mono e stereo, pelo que foi necessário converter os ficheiros que estão em stereo para mono. Para isto, aplicou-se uma conversão de stereo para mono fazendo a média dos valores de amplitude de ambos os canais.

b) Para além disto, os ficheiros de áudio presentes no conjunto de dados são obtidos em diferentes ambientes com diferentes características. O gravador pode estar perto ou longe das fontes de som, os ruídos de fundo variam, a qualidade das gravações diferem, entre outros. Todos estes factores implicam diferenças ao nível da amplitude entre os ficheiros de audio, provocando uma diminuição no desempenho do algoritmo classificador. De forma a colocar mais ênfase nas características espectrais do audio, aplicou-se uma normalização do pico de amplitude, na segunda fase do pré-processamento:

$$\hat{s}_k = \frac{1}{\max\limits_{i \in [1,...,n]} |s_i|} s_k$$

onde $\hat{s}_k$ é o sinal normalizado, s_k é o sinal em bruto, $k \in [1, ..., n]$ e n é o numero de samples do sinal.

c) A terceira fase do pré-processamento baseia-se na aplicação de um filtro butterworth passa-banda na faixa de 100 a 4000Hz, que passa as frequências de som dentro de uma determinada faixa e rejeita (atenua) as frequências fora dessa faixa. O objectivo foi eliminar frequências desnecessárias, muito baixas e muito elevadas, tipicamente associadas ao ruído ou a gamas menos significativas para os seres humanos, de modo a que no som predominem apenas as frequências "mestre"que caracterizam os diferentes tipos de eventos.

d) Na quarta fase de pré-processamento, os sons são sempre divididos em blocos de 20 milissegundos, sem sobreposição.

4.2.2 Sistema de Treino de Redes Neuronais

O sistema de classificação desenvolvido para o sistema em causa, foi baseado em redes neuronais. No entanto, a utilização de inputs de som em bruto foi uma abordagem idealizada sem qualquer base teórica ou prática e com poucas ou nenhumas referências na comunidade cientifica. Deste modo, o primeiro passo foi o desenvolvimento de uma pequena prova de conceito, que se entende por um modelo prático que possa provar o conceito (teórico) estabelecido por uma pesquisa ou artigo técnico.

Foi criado um conjunto de dados sintético de sons com diferentes frequências no seu conteúdo, e através de entradas à rede neuronal no domínio do tempo, provado que era possível fazer uma classificação precisa das frequências contidas nos sinais gerados. O anexo B apresenta esta prova de conceito.

Após este pequeno estudo, foram feitos vários testes com o conjunto de dados de eventos de áudio até que os melhores tipos de redes neuronais e parâmetros de rede para a detecção de eventos fossem encontrados.

No final, foram escolhidos dois tipos de redes com níveis aceitáveis de exatidão ($> 90\%$), que diferem no que respeita a velocidade e exatidão.

Cada uma destas redes foi implementada em Python, recorrendo às bibliotecas theano [the], lasagne [las] e nolearn [nol].

A DNN foi a primeira escolha, uma vez que é um tipo de rede bastante simples que pode ser implementado em qualquer dispositivo. Relativamente à sua estrutura, utilizaram-se 6 camadas ocultas compostas por 1000 nós cada, sendo que a função de activação escolhida para cada camada foi a tangente hiperbólica. De modo a reduzir a possibilidade de *overfitting*, foram adicionadas 6 camadas de *dropout* com uma probabilidade de 50%. Esta técnica consiste em excluir aleatóriamente alguns nós e respectivas ligações durante a fase de treinos. O método de optimização utilizado foi o *Nesterov Momentum* e a *learning rate* definida em 0.04. Relativamente à camada de saída, a função de activação utilizada foi a sigmóide.

Estes parâmetros foram definidos empiricamente depois de várias experiências, apresentando neste caso os melhores resultados. A figura seguinte demonstra parte da sua codificação:

```
# Load all chunks to input the neural network
X_train_sound, y_train_sound = csv_to_soundlist("./")

net1 = NeuralNet(
    layers=[('input', layers.InputLayer),
            ('dropout1', layers.DropoutLayer),
            ('hidden1', layers.DenseLayer),
            ('dropout2', layers.DropoutLayer),
            ('hidden2', layers.DenseLayer),
            ('dropout3', layers.DropoutLayer),
            ('hidden3', layers.DenseLayer),
            ('dropout4', layers.DropoutLayer),
            ('hidden4', layers.DenseLayer),
            ('dropout5', layers.DropoutLayer),
            ('hidden5', layers.DenseLayer),
            ('dropout6', layers.DropoutLayer),
            ('hidden6', layers.DenseLayer),
            ('output', layers.DenseLayer),
            ],
    # layer parameters:
    input_shape=(None,1,INPUT_NODE),
    # # dropout1
    dropout1_p=0.2,
    # hidden1
    hidden1_num_units=1000, #1000
    hidden1_nonlinearity=lasagne.nonlinearities.tanh,
    # dropout2
    dropout2_p=0.5,

    # hidden2
    hidden2_num_units=1000,
    hidden2_nonlinearity=lasagne.nonlinearities.tanh,
    # dropout3
    dropout3_p=0.5,

    # hidden3
    hidden3_num_units=1000,
    hidden3_nonlinearity=lasagne.nonlinearities.tanh,
    # dropout4
    dropout4_p=0.5,
    # hidden5
    hidden5_num_units=1000,
    hidden5_nonlinearity=lasagne.nonlinearities.tanh,
    # dropout6
    dropout6_p=0.5,

    # hidden6
    hidden6_num_units=1000,
    hidden6_nonlinearity=lasagne.nonlinearities.tanh,

    #output
    output_nonlinearity=lasagne.nonlinearities.sigmoid,
    output_num_units= NR_OUTPUTS,

    # optimization method:
    update=nesterov_momentum,
    update_learning_rate=0.04,
    update_momentum=0.9,
    max_epochs=10000,
    verbose=1,

    batch_iterator_train=BatchIterator(batch_size=1000),
    batch_iterator_test=BatchIterator(batch_size=1000),

    regression=True,

    objective_loss_function=multilabel_objective,
    custom_scores=[("validation score", lambda x, y: 1- np.mean(np.abs(x - y)))]
    )


## Train the network
nn = net1.fit(X_train_sound, y_train_sound)

## Dump the network weights to a file
nn.save_params_to("./dnn.npz")
```

Figura 4.5: Código Python para Implementação da DNN

A segunda opção foi uma RNN, porque, contrariamente ao DNN, este tipo de rede pode modelar diretamente informação sequencial que está naturalmente presente no áudio, isto é, este tipo de RNA considera uma componente temporal dos dados de entrada. A implementação é muito semelhante à da rede anterior, diferindo apenas no modo como se inicializa cada camada e considerando a modificação dos valores óptimos referentes a cada variável da arquitectura da rede.

```
net1 = NeuralNet(
    layers=[('input', layers.InputLayer),
            ('dropout1', layers.DropoutLayer),
            ('hidden1', layers.LSTMLayer),
            ('dropout2', layers.DropoutLayer),
            ('hidden2', layers.LSTMLayer),
            ('dropout3', layers.DropoutLayer),
            ('hidden3', layers.LSTMLayer),
            ('dropout4', layers.DropoutLayer),
            ('hidden4', layers.LSTMLayer),
            ('dropout5', layers.DropoutLayer),
            ('hidden5', layers.LSTMLayer),
            ('dropout6', layers.DropoutLayer),
            ('hidden6', layers.LSTMLayer),
            ('output', layers.DenseLayer),
            ],
```

Figura 4.6: Inicialização de Camadas LSTM

Nas tabelas seguintes são apresentadas as configurações de parâmetros de redes para os modelos com maior exatidão:

Parameter	Value
Size of Sound Chunks	20ms
Silence Detector	YES
Amplitude Normalization	YES
Bandpass Filter	100 – 4000 Hz
NN Type	Deep Neural Network
Number of hidden layers	6
Number of hidden neurons in each layer	1000
Activation function for hidden layer neurons	Hyperbolic Tangent
Number of dropout layers	6
Probability of dropout layers	0.5
Activation function for output layer neurons	Sigmoid
Learning rate	0.04
Optimization Method	Nesterov Momentum
Momentum	0.9
Mini Batch Size	1000
Precision	90.2%

Table 1

Tabela 4.1: Parâmetros DNN

Parameter	Value
Size of Sound Chunks	20ms
Silence Detector	YES
Amplitude Normalization	YES
Bandpass Filter	100 – 4000 Hz
NN Type	Recurrent Neural Network - LSTM
Number of hidden layers	6
Number of hidden neurons in each layer	800
Activation function for hidden layer neurons	Hyperbolic Tangent
Number of dropout layers	6
Probability of dropout layers	0.5
Activation function for output layer neurons	Sigmoid
Learning rate	0.08
Optimization Method	Nesterov Momentum
Momentum	0.9
Mini Batch Size	1000
Precision	93.4%

Table 2

Tabela 4.2: Parâmetros RNN

Relativamente à fase de treino destas redes, optou-se por dividir os dados em conjunto de treino (80%) e de teste (20%). Os testes de validação foram feitos com um conjunto de dados que não pertence ao conjunto de dados atrás referido, compreendendo o áudio de três vídeos com diferentes eventos em seu conteúdo. Tanto para a DNN como para a RNN, o treino foi feito recorrendo a *mini-batches* de tamanho fixo ,isto é, a cada iteração da rede é processado um pequeno subconjunto do conjunto de dados de treino (neste caso de 1000 amostras), em vez do típico processo que utiliza apenas uma única amostra por iteração, de forma a tornar o processo de treino computacionalmente mais eficiente, quer a nível diminuição de dados carregados para a memória quer a nível de redução temporal do processo de treino.

De destacar que a fase de ajuste de parâmetros das redes neuronais, apesar de bastante morosa, é uma tarefa de extrema importância em deep learning - também chamada engenharia de arquitecturas - sendo um factor chave para alcançar resultados óptimos no contexto de um problema.

4.2.3 Pós-Processamento de Dados

Embora a rede neuronal apresente uma boa exatidão, o tamanho de cada bloco de áudio usado para input (20ms) é pequeno demais para retornar resultados significativos e estáveis, já que o áudio muda as suas características muito rapidamente. Para lidar com isso, a predição foi feita analisando o evento mais frequente a cada meio segundo de áudio, ou seja, a cada 25 predições de blocos de 20ms. Em algumas ocasiões a probabilidade de um evento pertencer a uma certa classe, muda significativamente comparando com as frames de aúdio anteriores e posteriores. Este tipo de comportamento não é realista, visto que os eventos sonoros com que lidamos neste projecto raramente ocorrem por apenas 20 ms e desaparecem. Com esta tarefa de pós-processamento, os resultados tornaram-se mais estáveis e a janela de estabilização de meio segundo demonstra-se mais significativa para os utilizadores, oferecendo boa exatidão e pouca instabilidade.

No front-end da aplicação são apresentados os resultados das predições do sistema e pós-processamento de modo a que o utilizador consiga observar e analisar os resultados facilmente. A figura seguinte demonstra um conjunto de predições da rede no intervalo de 0.5 segundos de áudio, sendo possível verificar qual o evento que predomina (gargalhada, apresentada a cor verde):

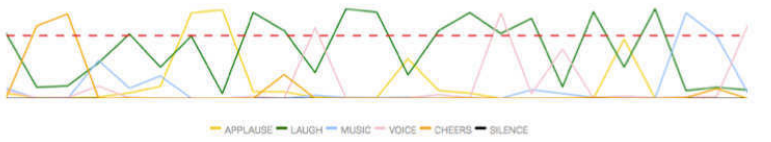


Figura 4.7: Gráfico de output da rede neural para 0.5s de áudio

Se a predição do evento estiver abaixo do valor de threshold (70% - representado a linha tracejada vermelha na imagem) é considerado como um "evento desconhecido".

A metodologia descrita anteriormente pode ser resumida pela seguinte figura:

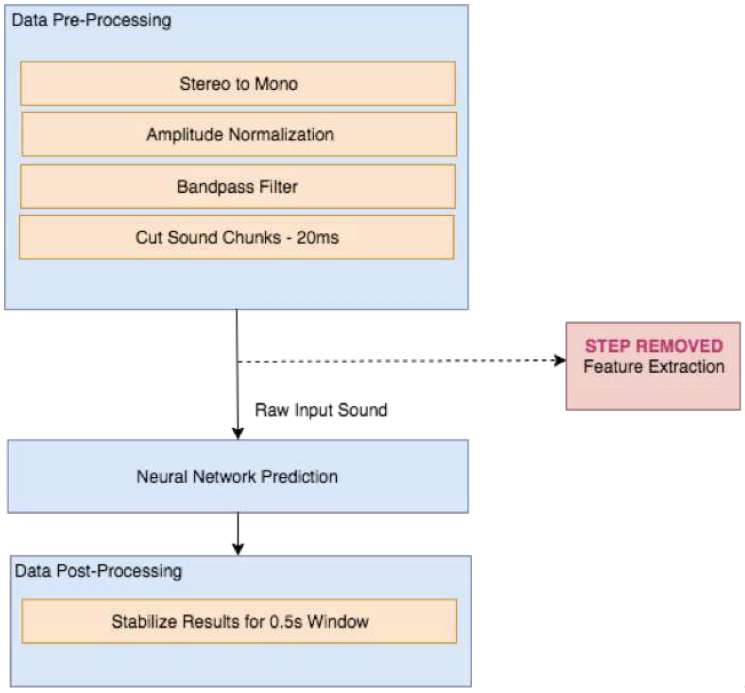


Figura 4.8: Metodologia utilizada no desenvolvimento do sistema

4.3 Desenvolvimento da Aplicação

4.3.1 Arquitectura

A aplicação criada foi desenvolvida como um Software as a Service (SaaS) que é composto por duas partes: back-end e front-end.

Para o front-end foi criada uma interface web simples e intuitiva para possibilitar a interacção com o utilizador. O back-end é responsável pelo processamento dos pedidos e pela organização dos resultados num sistema de ficheiros. Os ficheiros criados no processamento de cada pedido são organizados em pastas, sendo que cada pasta tem o nome igual ao título do vídeo. O conteúdo de cada pasta tem a seguinte estrutura:

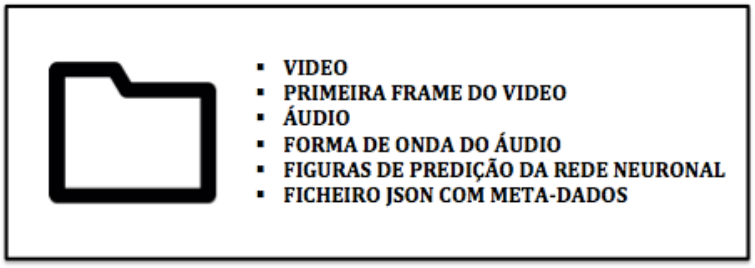


Figura 4.9: Estrutura base das pastas de processamento

Para servir os conteúdos estáticos como as páginas HTML e recursos inerentes às mesmas é utilizado um servidor de NGNIX [ngn] . Relativamente aos pedidos a conteúdos dinâmicos, são passados para um servidor de gevent [gev] que por sua vez os executa. Os conteúdos dinâmicos são todos scripts de python que estão responsáveis pelo processamento da aplicação, desde o download do vídeo até á sua predição.

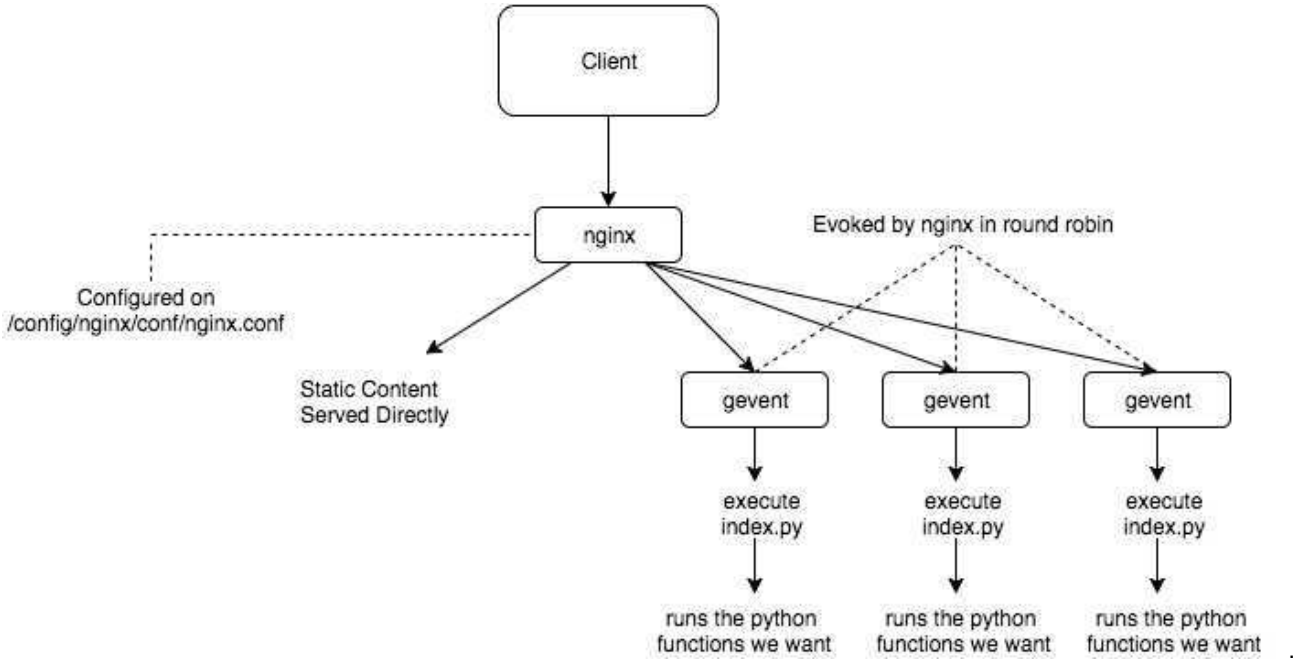


Figura 4.10: Funcionamento do Servidor

Este servidor, utilizado para suportar a aplicação desenvolvida, não foi desenvolvido no âmbito desta tese, pelo que não será feita uma análise detalhada ao mesmo.

O ponto de entrada da aplicação é um programa python chamado events.py que recebe pedidos do cliente, processa-os e retorna resultados para o front-end. Este script contém um sistema de gestão de processamento que verifica se o pedido a um determinado vídeo já foi tratado anteriormente, evitando assim repetição de processamento. O processamento de cada pedido para um novo vídeo ocorre no processo que é criado pelo script events.py. A criação de um novo processo para fazer o processamento nuclear do pedido teve dois objectivos: - Permitir a libertação do servidor para receber novos pedidos; - Retornar informações acerca do processamento ao front-end para que o mesmo possa informar o utilizador acerca do estado de processamento;

O fluxograma seguinte mostra o funcionamento do ficheiro events.py:

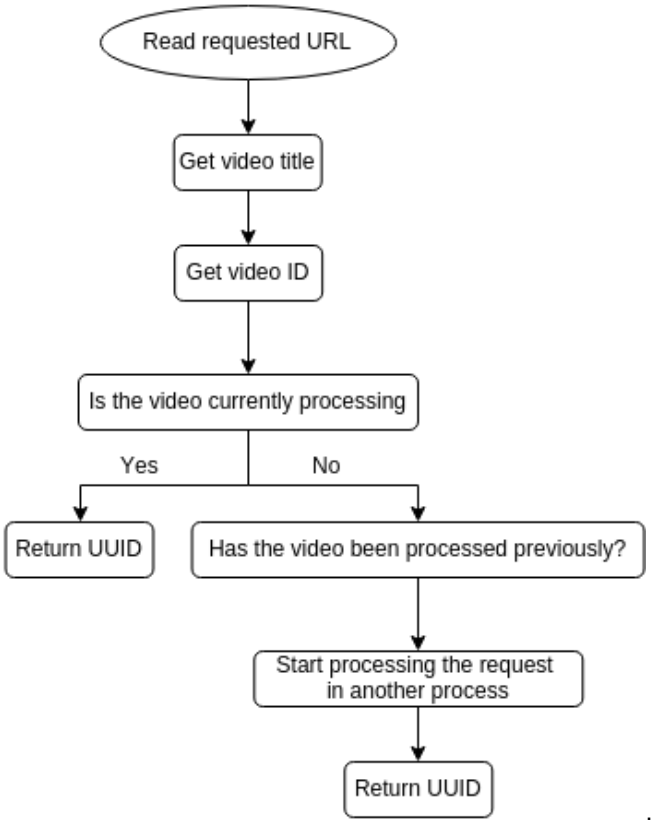


Figura 4.11: Fluxograma do ficheiro events.py

O código que é executado no novo processo é o app-dnn.py ou app-rnn.py, dependendo da escolha do tipo de rede neural feita pelo utilizador. É nestes scripts que é feito o processamento central e predições relativas ao vídeo inserido pelo utilizador. As tarefas executadas são as seguintes:

- Criação da pasta onde irão estar contidos os resultados do processamento;
- Criação de um ficheiro de status, onde é actualizada a percentagem relativa ao estado do processamento. Este ficheiro é acedido através de javascript pelo front-end, e visualizado pelo utilizador;
- Extracção do título do vídeo, relativo ao url introduzido pelo utilizador;
- Download do vídeo respectivo;
- Extracção do áudio a partir do vídeo já extraído;
- Pré-Processamento do Som: normalização de amplitude, aplicação do filtro de passa banda e divisão em "chunks"de 20ms;
- Predição da Rede Neural, para o conjunto de "chunks"que compõem todo o áudio;
- Pós-Processamento do Aúdio - estabilização dos resultados em intervalos de 0.5s;
- Criação de um ficheiro json com resultados relativos aos eventos detectados, a serem passados para o front-end;
- Criação de gráficos a serem utilizados no front-end;

- Remoção de ficheiros temporários gerados durante o processamento anterior;

- Renomear a pasta e actualizar o nome no ficheiro de status,de forma a passar a localização de todos os elementos necessários, ao front-end.

O fluxograma seguinte mostra todo este processo, sendo que a única diferença entre os ficheiros app-dnn.py e app-rnn.py é o tipo de cada camada na inicialização da rede neuronal.

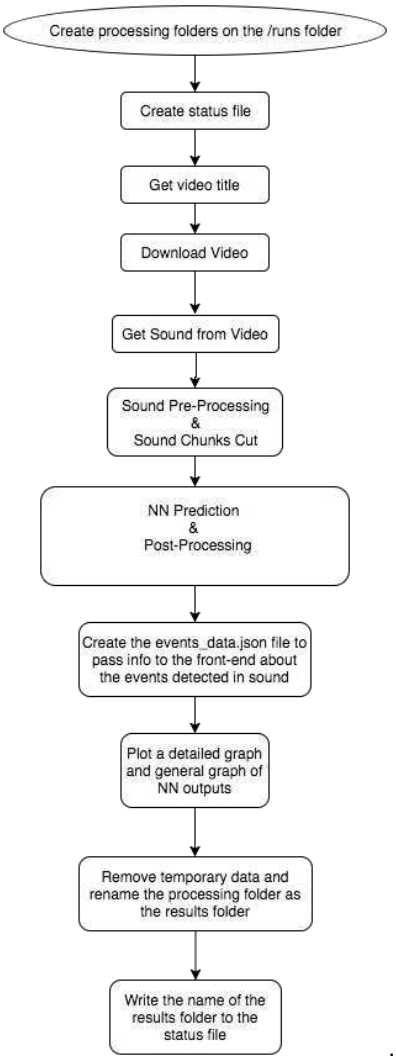


Figura 4.12: Fluxograma dos ficheiros app-dnn.py e app-rnn.py

4.3.2 Interface

O utilizador deverá inserir o link do vídeo na aplicação e escolher o tipo de rede neuronal a ser utilizado para fazer a classificação do áudio:

- O uso da DNN, ou deep neural network , permite que a classificação seja feita mais rapidamente, mas com menor exatidão (89%);

- O uso da RNN ,ou recurrent neural network, permite que a classificação seja feita com mais exatidão (93%), mas com uma rapidez consideravelmente inferior face à primeira opção;

A interface disponibiliza ainda uma lista de vídeos que a aplicação correu anteriormente que pode se acedida pelo user bastando clicar em 'Previous Runs'.

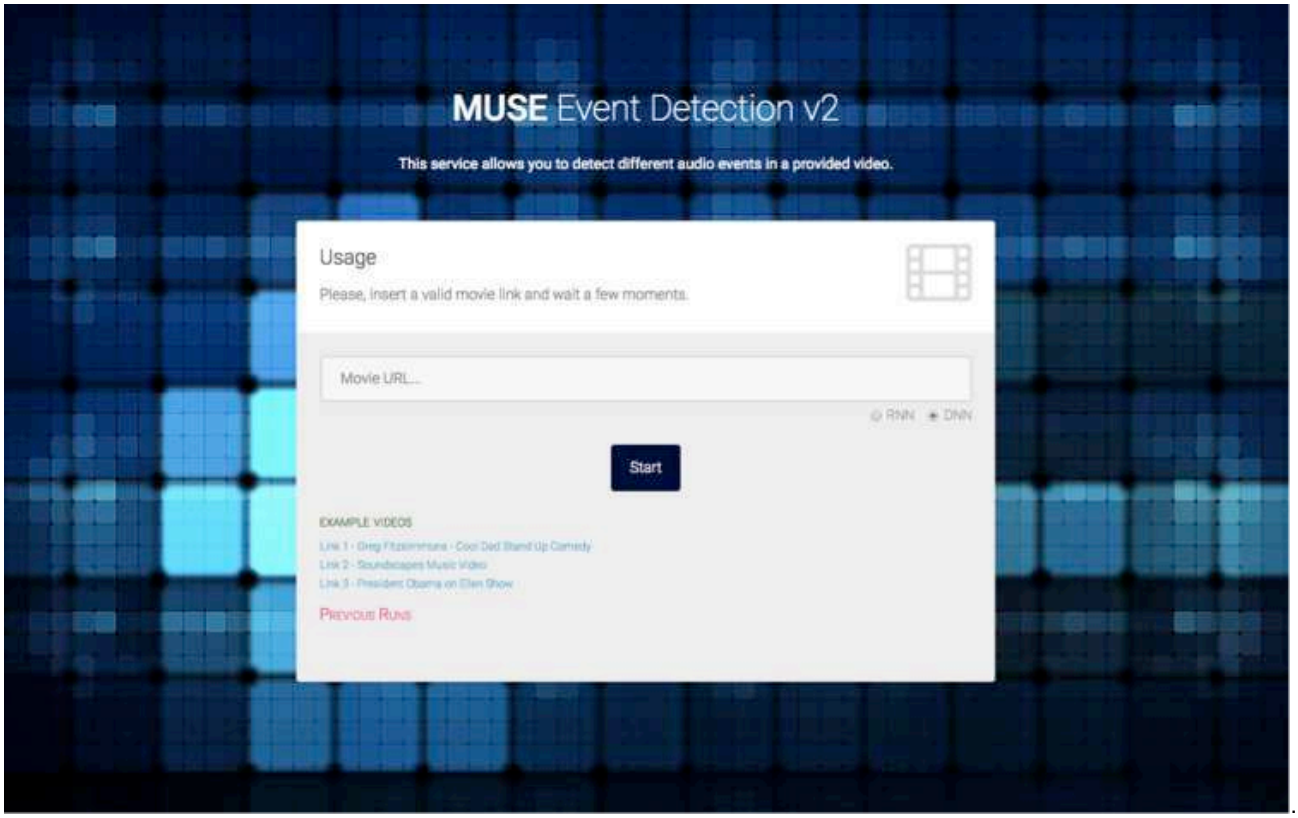


Figura 4.13: Interface Inicial do Sistema

Figura 4.14: Interface para Apresentação de Resultados

4.3.3 Linguagens de Programação/Plataformas

As linguagens de programação utilizadas e em que se basearam os algoritmos apresentados no projecto foram as linguagens Python para back-end e HTML, CSS e Javascript para front-end. Apesar de actualmente existirem diversas linguagens que suportam a implementação de algoritmos de machine learning, optou-se pela utilização de Python como linguagem base deste projecto visto que se trata-se de uma linguagem de alto nível, livre, com bibliotecas de inteligência artificial altamente documentadas, amplamente utilizadas e ao mesmo tempo bastante robustas. Relativamente às bibliotecas de python utilizadas podemos enumerar:

Sklearn

Biblioteca de machine-learning open-source que inclui diversos tipos de algoritmos de classificação, regressão, clustering e visualização.

Theano

Biblioteca que permite definir, optimizar e avaliar expressões matemáticas envolvendo matrizes multidimensionais de forma eficiente.

Lasagne

Biblioteca leve utilizada para construir e treinar redes neuronais de diferentes tipos, em conjunto com a biblioteca Theano.

Matplotlib

Biblioteca para geração de gráficos.

Nolearn

Biblioteca que permite uma abstracção de alto-nível em relação à biblioteca Lasagne.

Numpy

Biblioteca matemática fundamental para aplicações cientificas em Python.

Por outro lado, para o download do vídeo e seu processamento foram utilizadas duas ferramentas:

FFMpeg

Software open-source que contém um elevado número de bibliotecas e programas capazes de lidar com dados múltimédia, nomeadamente vídeo e áudio.

Youtube-dl

Programa de linha de comandos utilizado para fazer download de sites como o youtube, vimeo, entre outros.

5

Análise de Resultados

No presente capítulo, torna-se necessário proceder à respetiva apresentação e análise dos resultados obtidos ao longo de todo o projecto, a fim de se poderem extrair algumas ilações.

Uma das partes cruciais deste trabalho foi conseguir obter o conjunto optimo de parâmetros que maximiza-se a exatidão da rede neuronal, tanto a nível dos parâmetros associados à rede, como a nível de pré e pós processamento.

Os valores nas tabelas 5.1 a 5.9 foram obtidos utilizando como configuração base os parâmetros da tabela 4.1. Em cada "subsecção", o valor do parâmetro mencionado é iterado, mantendo todos os outros fixos. Desta forma,podemos observar o efeito da modificação de cada parâmetro individual no sistema.

5.1 Efeitos dos Parâmetros de Pré-Processamento

Efeito do tamanho da "chunk":

Chunk Size	0.50s	0.25s	0.10s	0.020s	0.010s
Accuracy	73.6%	80.2%	85.2 %	**90.2%**	88.1%

Tabela 5.1: Efeito da Variação do Tamanho da "Chunk"de Audio

Efeito da normalização de amplitude e detecção de silêncio:

AN & SD	No	Yes
Accuracy	60.3%	**90.2%**

Tabela 5.2: Efeito da Normalização e Detecção de Silêncio

Efeito do filtro de passa-banda:

Bandpass Range(Hz)	None	100-4000	200-4000	300-4000	100-3000	100-2000	100-5000
Accuracy	37.4%	**90.2%**	87.2%	86.4%	80.2%	71.2%	67.4%

Tabela 5.3: Efeito do Filtro de Passa Banda

5.2 Efeitos dos Parâmetros da Rede Neuronal

Efeito de Número de Neurónios por Camada:

Number of Hidden Neurons	100	200	500	800	1000	2000	3000	4000
Accuracy	52.3%	65.1%	88.1%	**90.2%**	87.4%	85.1%	84.7%	82.2%

Tabela 5.4: Efeito do Número de Neurónios por Camada

Efeito da Função de Activação:

Activation Function	rectify	sigmoid	softmax	tanh	elu	softplus	linear
Accuracy	90 %	71.9%	62%	**90.2%**	87.2%	71.9%	45%

Tabela 5.5: Efeito da Função de Activação

Efeito do Tamanho dos Mini-Batches:

Mini Batch Size	200	500	1000	2000	2500
Accuracy	90.1 %	89.1%	**90.2%**	87.8%	87.6%

Tabela 5.6: Efeito do Tamanho dos Mini-Batches

Efeito da Learning Rate:

Learning Rate	0.005	0.01	0.02	0.04	0.08	0.1	0.5
Accuracy	81.1	85.2	88.1	**90.2%**	89.8 %	83.3%	40.2%

Tabela 5.7: Efeito da Learning Rate

Efeito do Método de Optimização:

Optimization Method	nesterov_ momentum	sgd	rmsprop	adam	adagrad
Accuracy	**90.2%**	83.2%	40.2%	21.2%	87.0%

Tabela 5.8: Efeito do Método de Optimização

Efeito do Momentum:

Momentum	0.1	0.3	0.5	0.7	0.8	0.9
Accuracy	84%	86%	86.5%	87.2%	88.1%	**90.2%**

Tabela 5.9: Efeito do Momentum

Com base nas tabelas apresentadas, foram escolhidos os valores de parâmetros que maximizaram a exatidão da rede neuronal. Esta abordagem apenas considerou que fossem feitos testes a um pequeno conjunto de valores para cada parâmetro, não garantindo que os parâmetros escolhidos representem o melhor conjunto de valores de parâmetros.

5.3 Rapidez do Sistema

A tabela abaixo representa o tempo da fase de predição, que inclui a inicialização da rede neuronal, a predição do som e a criação dos gráficos da respectivo processo. De notar que estes tempos foram obtidos numa máquina local sem qualquer hardware dedicado (Macbook Air 1,7 GHz i5,4 GB 1333 MHz DDR3, Mid 2011).

VIDEO DURATION	**DNN**	**RNN**
30 s	15 s	3.12 m
1 m	24 s	3.37 m
4 m	1.40 m	5.35 m
10 m	4.14 m	9.23 m
15 m	7.20 m	12.44 m

Tabela 5.10: Rapidez DNN vs RNN

5.4 Evolução do Sistema

Ao longo do desenvolvimento do sistema de classificação, ajustaram-se diversos valores de parâmetros, foram aplicados diversos métodos de pré-processamento e feitas melhorias no que respeita ao tamanho do dataset, podendo-se considerar que a aplicação do filtro de passa-banda foi um dos fatores chave para que a exatidão do sistema chegassem a valores satisfatórios, conforme demonstra a figura 5.1, com a evolução do sistema.

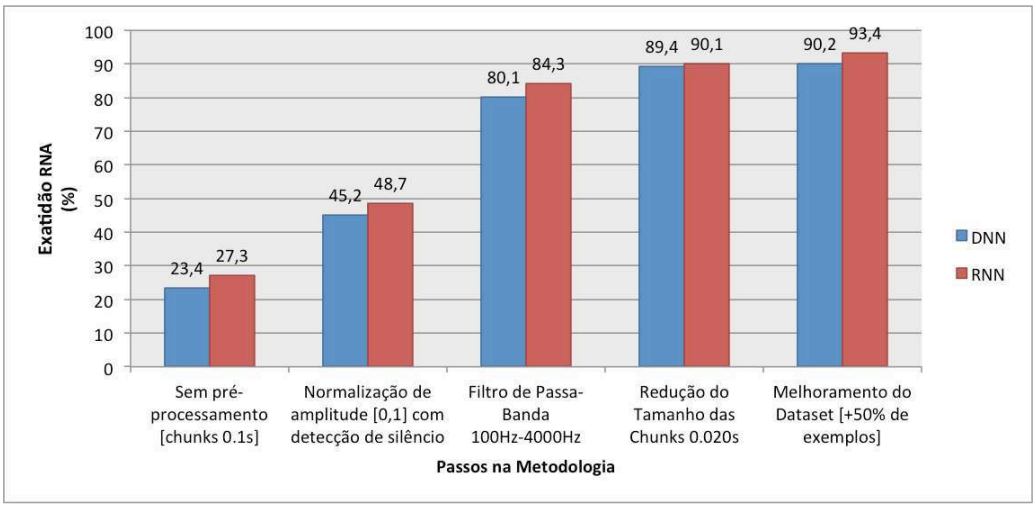


Figura 5.1: Evolução do Sistema

5.4.1 Limitações e Desafios

A implementação do projecto por si só impôs-se como um desafio, considerando que em cada uma das etapas desenvolvidas surgiram limitações e dificuldades.

O primeiro desafio surgiu quando me foi proposto desenvolver um sistema que utilizasse redes neuronais com entradas 'puras' de som, sem qualquer tipo de extracção de características. Toda a investigação desde a prova de conceito até a obtenção de resultados satisfatórios em eventos de aúdio revelou-se uma longa e trabalhosa tarefa.

Na fase inicial, e apesar de haver uma prova de conceito algo conclusiva num conjunto de dados sintético, os resultados não eram satisfatórios, visto que existem grandes diferenças para a classificação de dados reais, sujeitos a ruídos e comportamentos estocáticos. A aplicação do filtro passa-banda aos sons de entrada revelou-se um factor chave no sucesso do projecto.

Outro problema foi a construção automática de um conjunto de dados relevante, visto que a elaboração manual do mesmo não poderia ser considerada por ser bastante morosa. O script de extracção elaborado, baseado na análise de legendas, extraiu o conjunto de dados quase completo, no entanto, devido à imprecisão de grande parte das legendas, foi necessário definir uma metodologia de visualização e detecção de outliers. Elaborar este processo num contexto de dados de alta-dimensão revelou-se uma tarefa também desafiante.

Uma das maiores limitações foi a utilização de hardware não-dedicado para o efeito, isto é, não foram utilizadas placas-gráficas ou componentes para maximizar a capacidade de processamento, o que resultou em tempos bastante longos no treino das redes neuronais.

A última etapa do trabalho, relativa à integração de todas as partes num sistema pronto a usar pelo utilizador foi sem dúvida uma das partes mais interessantes do trabalho, visto que permite visualizar os resultados de forma real e proporcionar uma interacção directa com o utilizador.

6

Conclusão

A detecção e classificação de eventos de áudio é uma área de pesquisa praticamente inseparável de muitas aplicações de análise de vídeo, tendo um papel chave na indexação, navegação e recuperação dos mesmos. Uma análise de áudio gerada de forma concisa e inteligente não só permitirá uma interação mais informativa entre utilizadores e computadores durante a utilização e consulta de vídeos, mas também ajudará a construir sistemas de indexação e recuperação de vídeo mais rápidos e significativos. Recentemente, a área de abstração de vídeo tem vindo a atrair grande interesse, tanto a nível de projectos de investigação como por parte das grandes indústrias, sendo uma das áreas em que existe maior espectativa de evolução nos anos que se avisinham.

Este documento teve por objectivo apresentar a concretização do projecto de dissertação que desenvolve algumas soluções às tematicas indicadas anteriormente. O mesmo foi realizado no âmbito do Mestrado de Engenharia Informática da Universidade de Évora, e teve com intuito de dar resposta a alguns dos objectivos que são inerentes à finalização deste grau académico:

- Teste à capacidade de pôr em prática os princípios e as técnicas associadas à conceção e implementação de casos reais;

- Aplicação dos conhecimentos obtidos nas unidades curriculares do curso face a problemas ligados às mesmas, bem como a aplicação dos conhecimentos a novos paradigmas não abordados no curso;

A metodologia delíneada utilizou redes neuronais profundas e recorrentes para fazer a classificação de eventos de áudio, utilizando apenas os sinais no domínio do tempo como entrada destes algoritmos. Foram obtidos resultados na ordem dos 90% de exatidão.

A partir da investigação desenvolvida, concluímos que o uso de técnicas de *deep learning* origina níveis de exatidão bastante similares a outras técnicas mais utilizadas, nomeadamente ao uso de MFCCs, que é uma característica tradicionalmente usada na detecção de eventos no reconhecimento de fala. Concluímos também que a classificação de eventos com redes neuronais recorrentes proporciona melhores resultados em comparação com outros tipos de rede testados, devido ao seu elevado potencial em lidar com sequências temporais. De notar também que que os valores dos hiperparâmetros de aprendizagem , como a learning rate, o tamanho do mini-batches e o momentum, podem ter uma importância significativa na aprendizagem das DNN e RNN. Encontrar um intervalo útil para a optimização destes parâmetros pode acelerar significativamente o processo de aprendizagem e resultar num aumento dos níveis de exatidão.

Em comparação a outros trabalhos desenvolvidos nesta área, não se pode concluir que os resultados obtidos sejam melhores ou piores. Na verdade grande parte das investigações desenvolvidas são meramente teóricas, o que não aconteceu no presente projecto. O objectivo não foi apenas maximizar a exatidão dos algoritmos de classificação mas sim conseguir um meio termo entre a performance e a possibilidade de aplicação prática de toda a metodologia desenhada.

Deste modo, vale também a pena referir que o sistema desenvolvido no âmbito desta dissertação está a ter aplicação prática na indústria de análise de vídeo, pela empresa muse.ai, e tem tido um impacto muito positivo no comunidade de investigadores e investidores envolvidos nesta área.

Relativamente aos objectivos e metas delineados ao longo da presente dissertação (nomeadamente nos capítulos 1.2 e 2.4), podemos concluir que todos foram alcançados com sucesso, no entanto existem alguns aspectos que poderiam melhorar o sistema desenvolvido, mas que por limitação de tempo, não houve oportunidade de desenvolver. Estes aspetos incidem em grande parte na optimização do sistema de classificação e podem ser desenvolvidos em trabalho futuro:

- **Permitir a detecção simultânea de eventos**

A detecção simultânea de eventos não foi considerada neste projecto devido à inexistência de conjuntos de dados 'reais' com sobreposição e legendados. A criação de um conjunto de dados deste tipo é uma tarefa bastante morosa e exclusivamente manual, pelo que não foi considerada, no entanto seria uma componente de bastante importância a introduzir no sistema;

- **Adicionar mais classes para classificação**

A introdução de novos sons para classificação enriquecerá o sistema, bastando apenas construir conjuntos de dados para as novas classes a introduzir, sendo que o sistema está preparado para classificar um número variável de sons, a definir pelo utilizador;

- **Aumentar a exatidão da predição em algumas classes**

Apesar de os resultados gerais serem satisfatórios, algumas das classes de sons têm uma gama muito ampla de diferentes sons - como é o caso da classe de música - pelo que o conjunto de dados deve ser melhorado com novos sons que completem a classe e aumentem a sua exatidão;

- **Melhorar o sistema de detecção de outliers**

O sistema de detecção de outliers desenvolvido exige que o utilizador tenha de 'visualizar' as predições de cada som isoladamente e decida se o deseja manter ou não no conjunto de dados. Automatizar este processo é sem dúvida uma alteração de peso a considerar numa reformulação futura;

Uma vez que são poucos os estudos com aplicações práticas realizados nesta área, este projecto estimulou, motivou e possibilitou um leque diversificado de conhecimentos, com ênfase no desenvolvimento de sistemas end-to-end e de *deep learning*. Orgulho e prazer são talvez os sentimentos que melhor traduzem o sinto ao finalizar algo com muito significado pessoal, sendo que este trabalho espera inspirar futuras pesquisas e desenvolvimento em áreas de maior necessidade, nomeadamente em campos como a Medicina.

A

Estudo de Técnicas para Visualização do Dataset

ESTUDO DE TÉCNICAS PARA VISUALIZAÇÃO DO DATASET

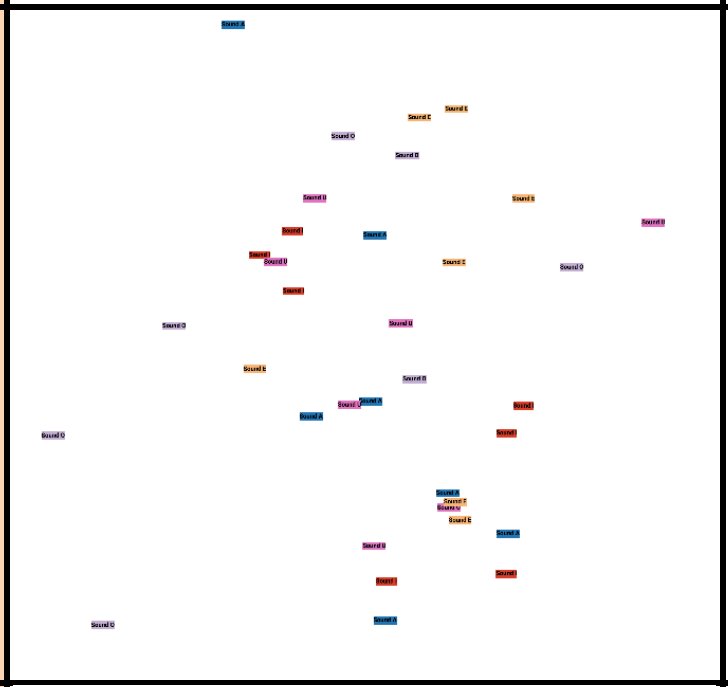
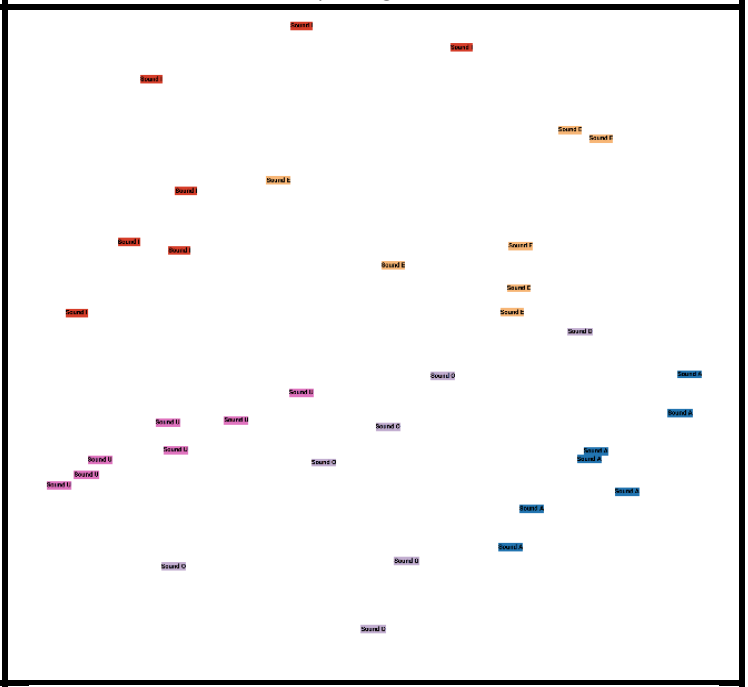
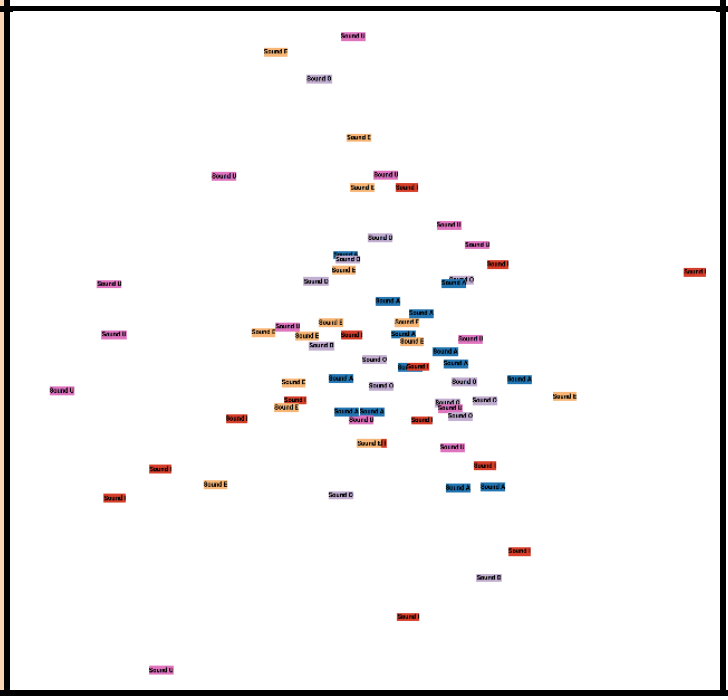
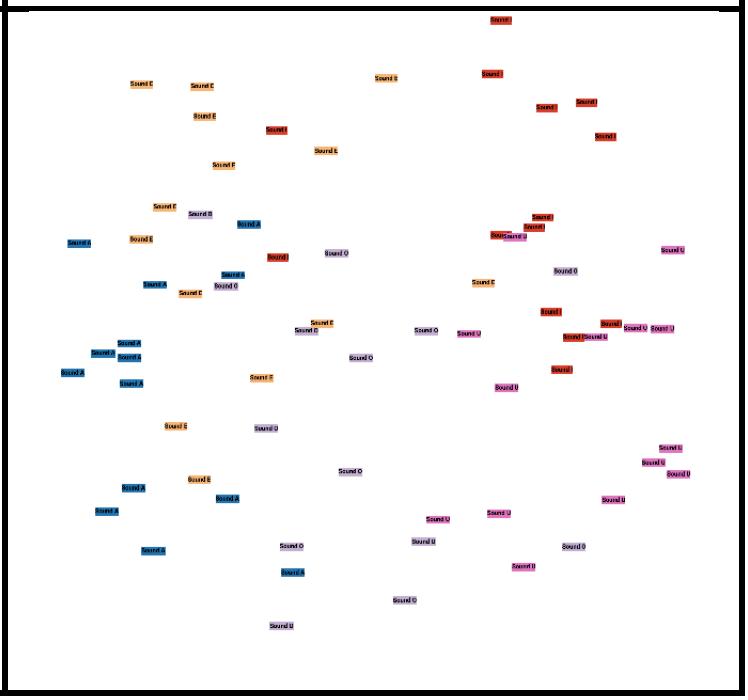
Objectivo :
- Identificar o melhor método de redução de dimensionalidade que permita uma visualização clara do dataset de áudio;
- Detecção de Outlires baseado na visualização de cada ‘cluster’ de áudio

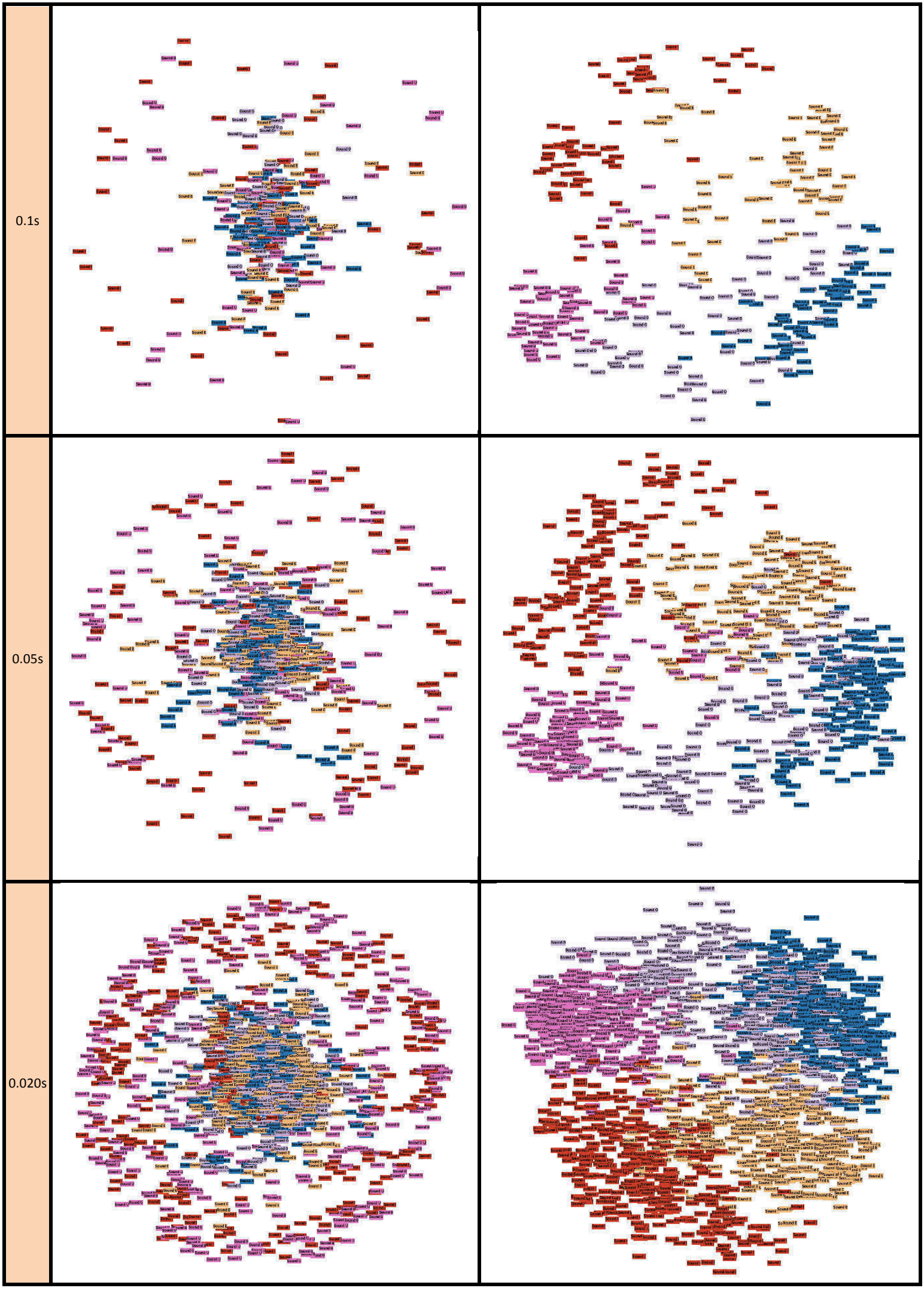
1.TESTES INICIAIS

Dataset : Sons A , E , I , O , U
Fonte : 7 pessoas diferentes (3 homens , 4 mulheres)
Tamanho dos Ficheiros : 1 Segundo
Número Total de Ficheiros : 35 (7 por classe)

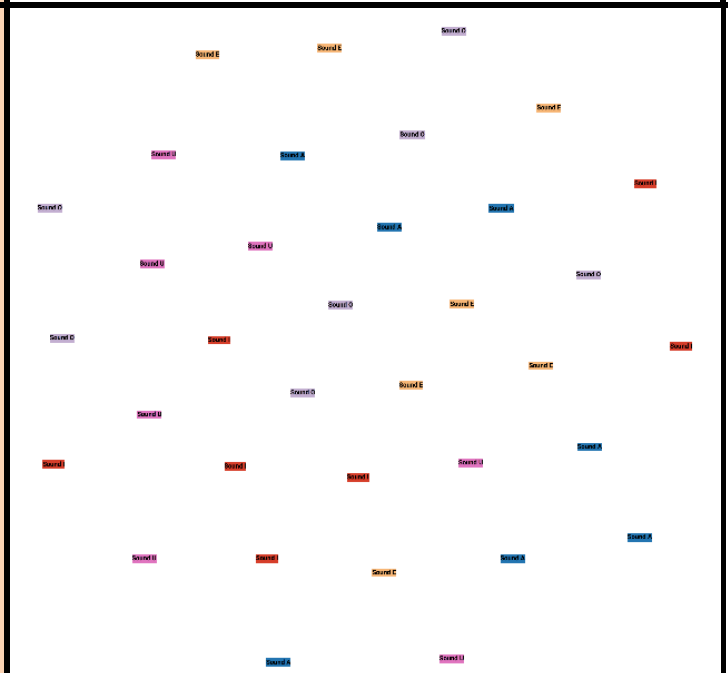
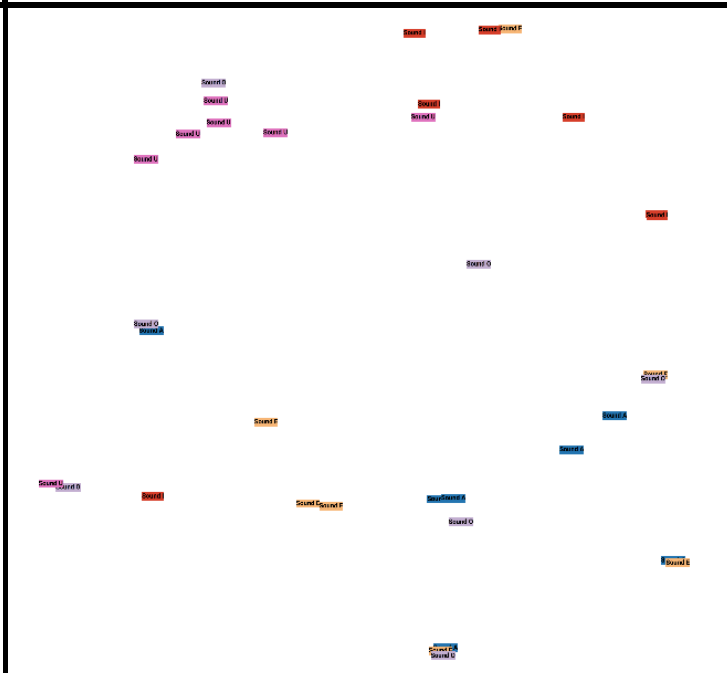
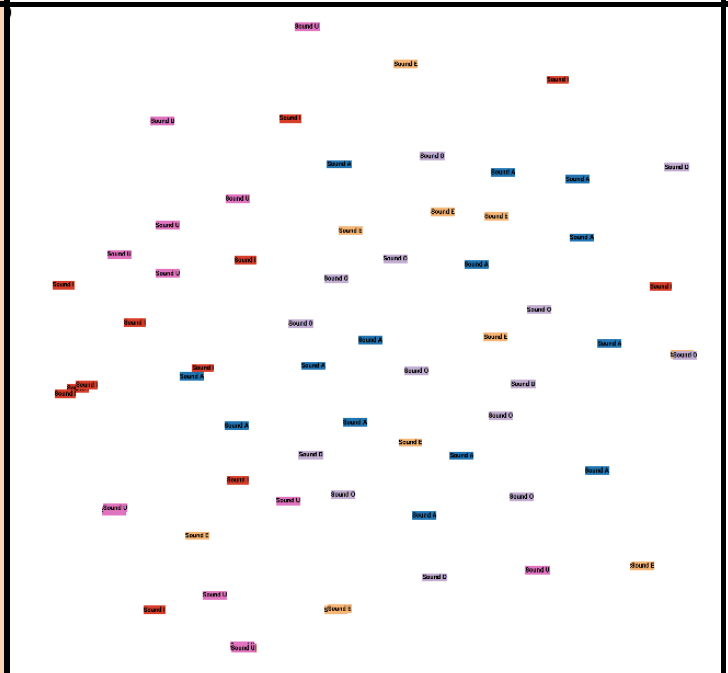
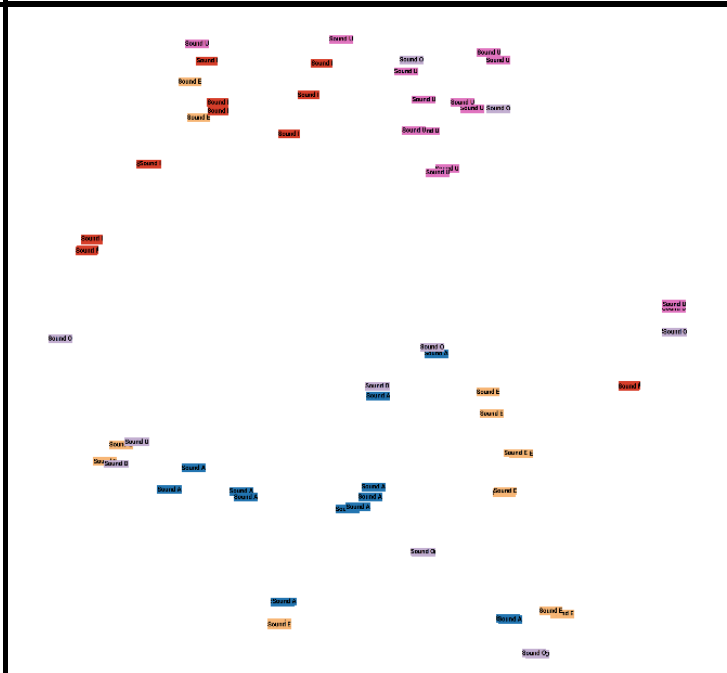
Visualização obtidas através da ferramenta **Tensorflow Embedding Projector** :

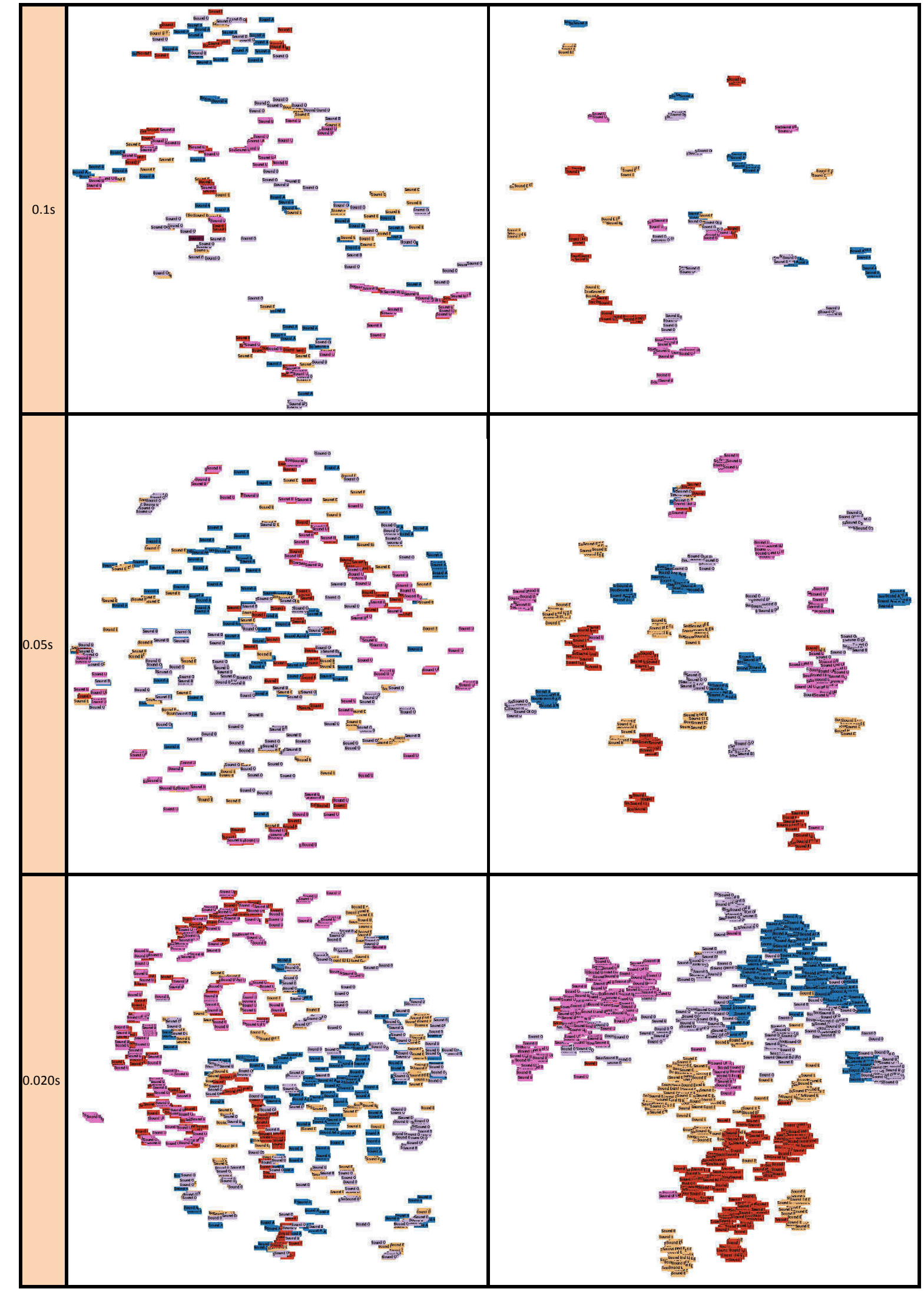
Principal Component Analysis (PCA)

chunks	Raw Sounds	Spectrograms
1s		
0.5s		


0.1s
0.05s
0.020s

t-Distributed Stochastic Neighbor Embedding (t-SNE)
(Perplexity = 5 , Learning Rate = 10 , Iterations = 600)

chunks	Raw Sounds	Spectrograms
1s		
0.5s		

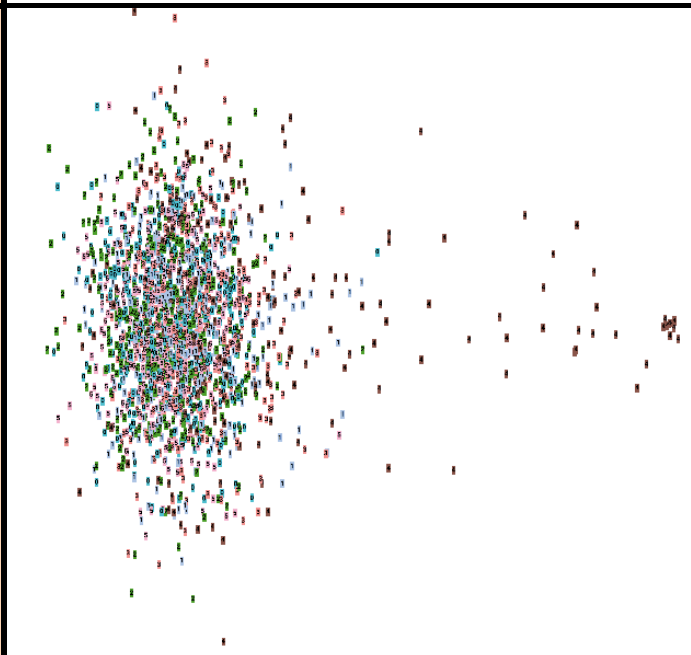
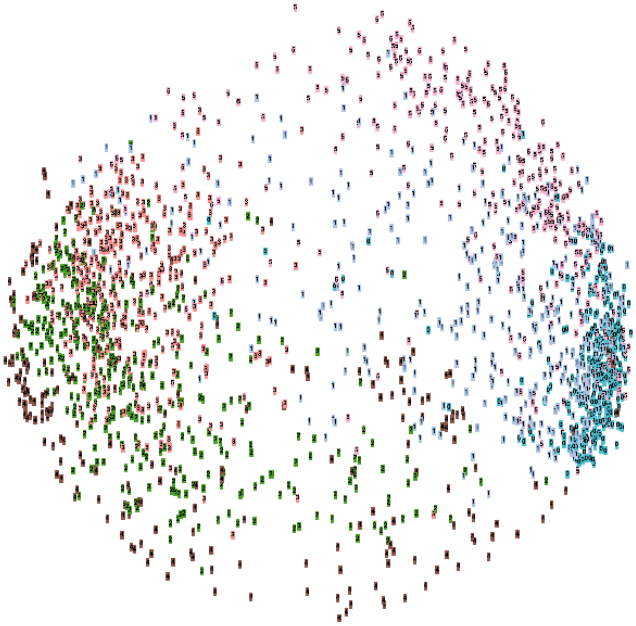
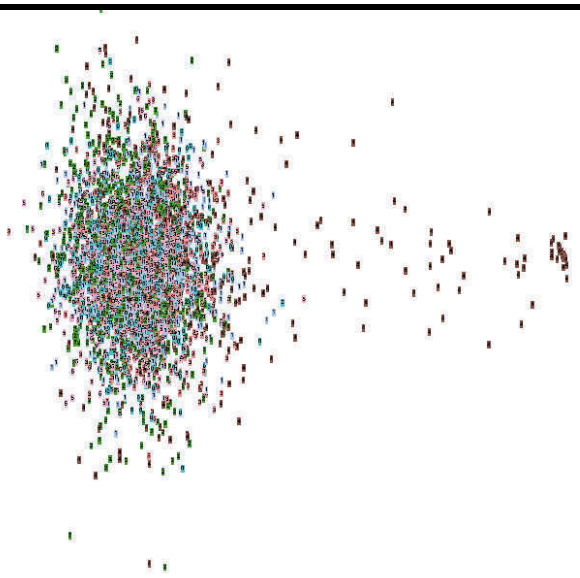
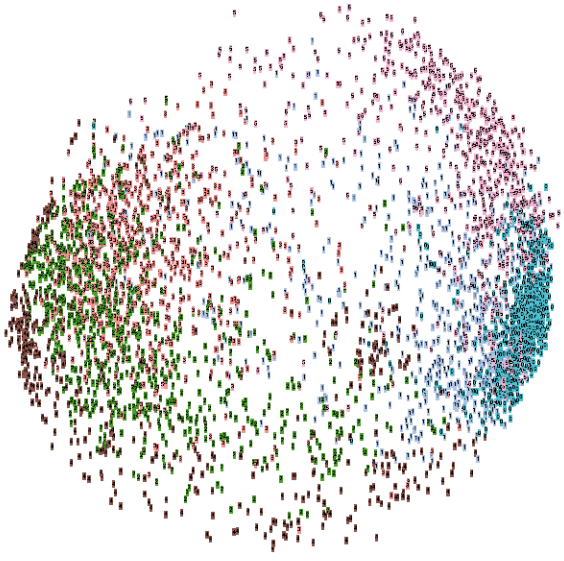

0.1s
0.05s
0.020s

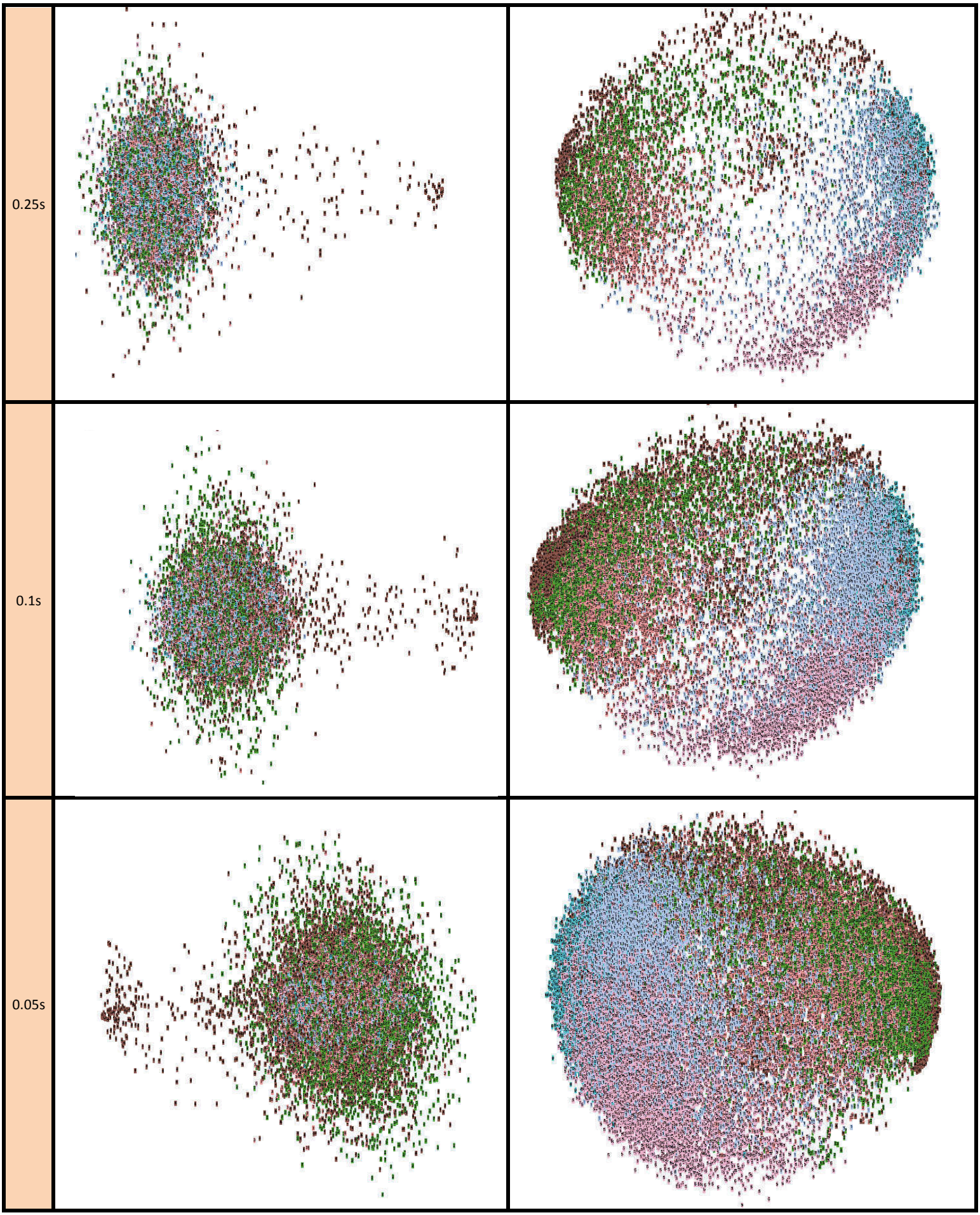
2.DATASET DE EVENTOS DE ÁUDIO

Dataset : Sons de Aplauso , Gargalhadas , Musica , Gritos , Voz e Silêncio
Tamanho dos Ficheiros : 1 Segundo
Número Total de Ficheiros : 1494 (249 por classe +– 4 minutos)

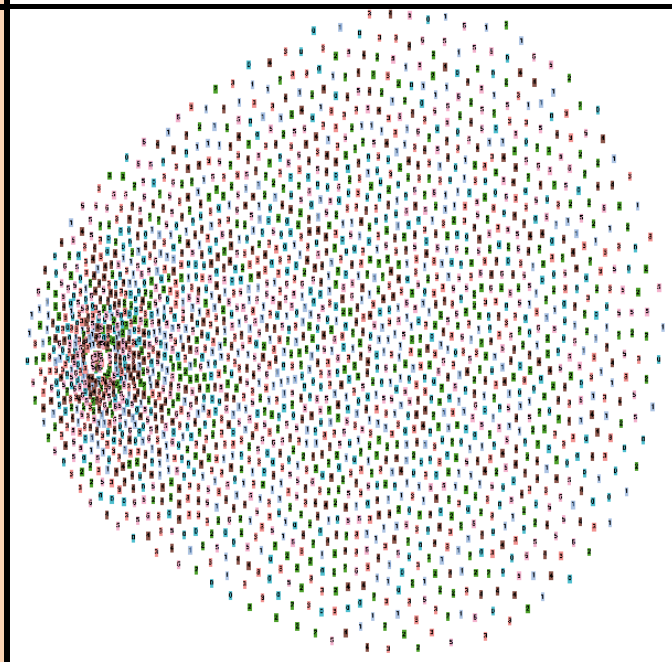
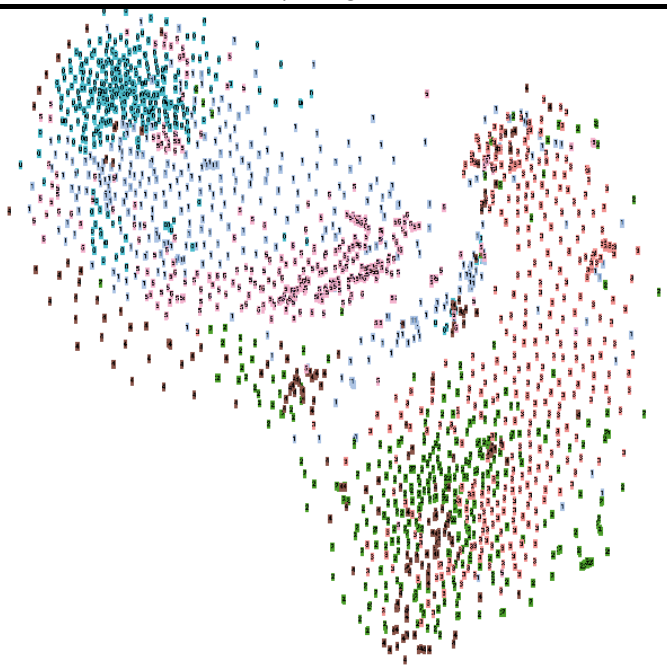
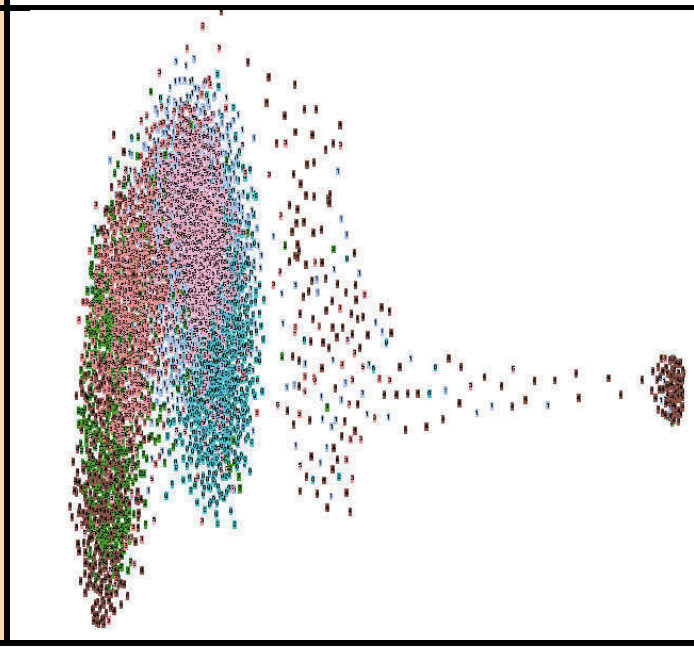
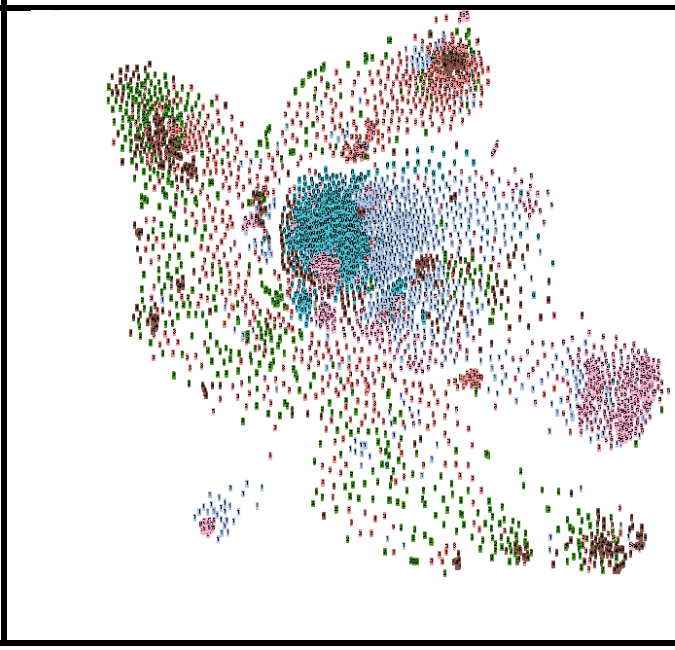
Visualização obtidas através da ferramenta **Tensorflow Embedding Projector** ::

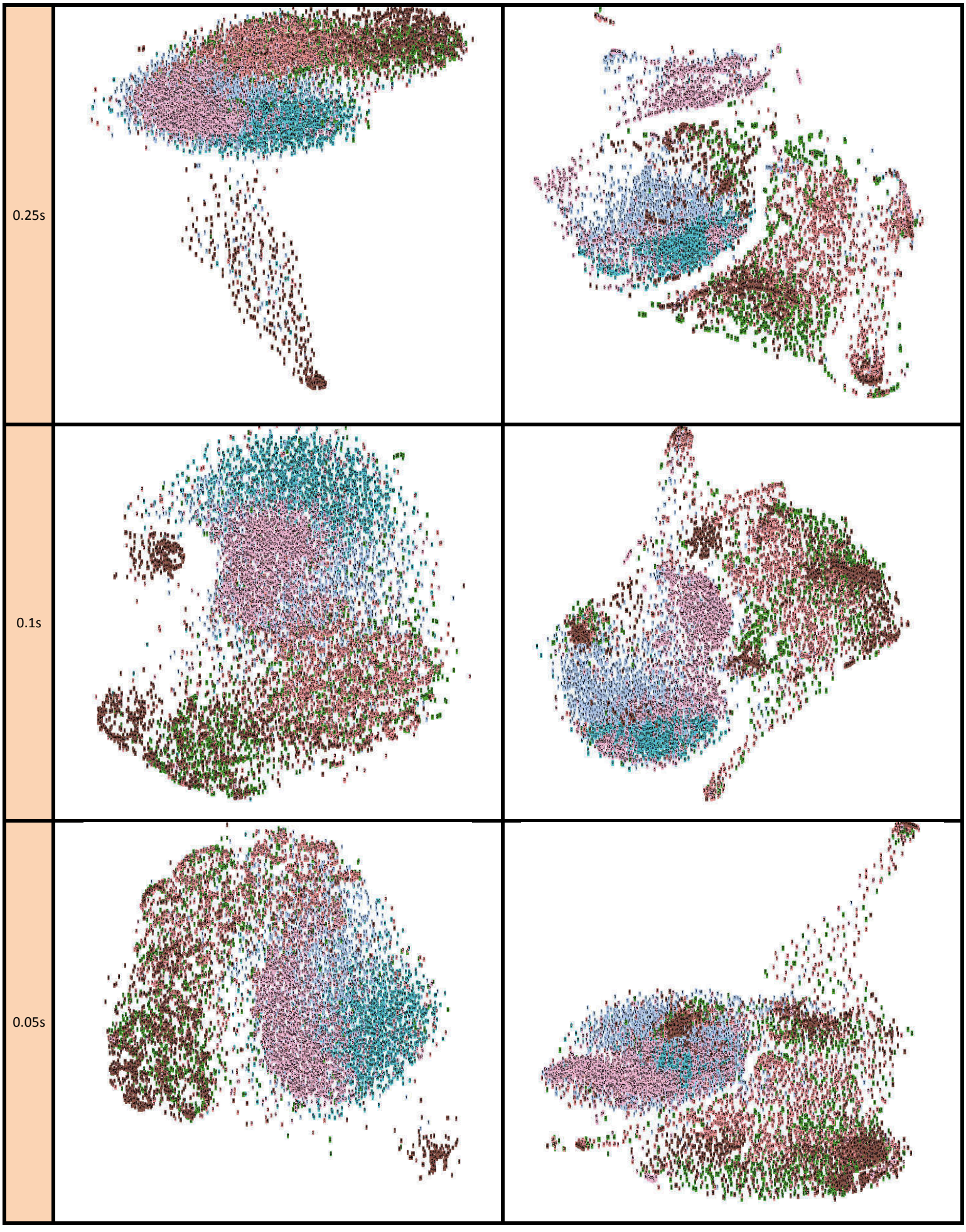
Principal Component Analysis (PCA)

chunks	Raw Sounds	Spectrograms
1s		
0.5s		


0.25s
0.1s
0.05s

t-Distributed Stochastic Neighbor Embedding (t-SNE)
(Perplexity = 30 , Learning Rate = 10 , Iterations = 600)

chunks	Raw Sounds	Spectrograms
1s		
0.5s		


0.25s
0.1s
0.05s

Conclusões:

- Qualquer um dos métodos de redução de dimensionalidade testados (PCA e t-SNE) apresentam resultados e representações inconclusivos, quando aplicados aos dados em bruto. À partida este resultado deve-se ao facto de demasiada informação importante ser perdida no processo de redução de dimensionalidade.
- Os mesmos métodos, quando aplicados a um dataset de espectrogramas dos respectivos dados, retornam visualizações bastante satisfatórias, sendo possível destinguir as diferentes classes de áudio.
- O método de Principal Component Analysis em conjunto com o dataset de espectrogramas de cada evento, retornam os resultados mais claros a nível visual.
- Pode-se concluir que no contexto de visualização de dataset's, à partida é sempre necessário um pré-processamento que transforme o áudio em caracteristicas/atributos relevantes (neste caso espectrogramas), não sendo possível a utilização de sinais puros.

B

Estudos Iniciais com Dataset Sintético

PROVA DE CONCEITO

CLASSIFICAÇÕES DE SINAIS ACÚSTICOS NO DOMÍNIO DO TEMPO

Objectivo : Entender se é viável utilizar redes neuronais profundas na classificação de sinais acústicos 'puros'.

A abordagem delíneada para fazer a prova de conceito do problema abordado na dissertação de mestrado baseou-se na composição de um dataset sintético de sons com diferentes frequências que foi treinado em redes neuronais profundas (DNN). A tarefa foi decomposta em tarefas menores, como a classificação de sons apenas com uma frequência, a fim de compreender a dinâmica dos factores envolvidos e compreender o comportamento das DNNs em sinais puros. As tarefas delíneadas foram as seguintes:

1.DNNs para classificar sons puros com apenas uma frequência no seu conteúdo;
2.DNNs para classificar sons puros com diferentes frequências no seu conteúdo;

O desafio desta abordagem foi tentar entender se é possível 'emular' implicitamente uma Fast Fourier Transform (FFT) através de uma rede neural profunda em vez de usar características que recorram a uma transformação para o domínio da frequência, como entrada da rede. Assim é evitada 'perda' de tempo ao ao nível de pré-processamento no backend do sistema e tornar possível uma experiência rápida a utilizadores que utilizam aplicativos em tempo real onde a rede neural será usada para classificar eventos de áudio diferentes.

Requerimentos

- Python
- Lasagne
- Theano
- Nolearn
- Sklearn
- Matplotlib
- Numpy

1) DNN'S PARA CLASSIFICAR SONS PUROS COM APENAS UMA FREQUÊNCIA NO SEU CONTEÚDO

a) Criação do Dataset

O primeiro passo nesta tarefa foi criar um conjunto de dados de sons puros com apenas uma frequência no seu conteúdo, suficientemente representativo para permitir o treino de uma rede neuronal. Assim, foi criado um script em python (puredataset_generator.py) que gera diferentes sons puros (0,1 segundos de duração) com ondas sinusoidais com parâmetros variáveis

$$y(t) = A\sin(2\pi f t + \varphi) = A\sin(\omega t + \varphi)$$

Uma onda sonora com uma certa frequência pode ter diferentes parâmetros variáveis: a fase, amplitude e variação de amplitude;

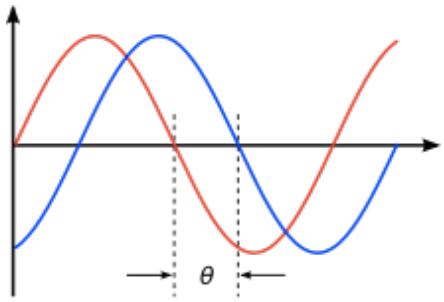


Figure 1 - Fase

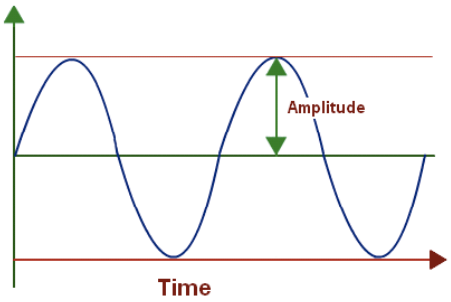


Figure 2 - Amplitude

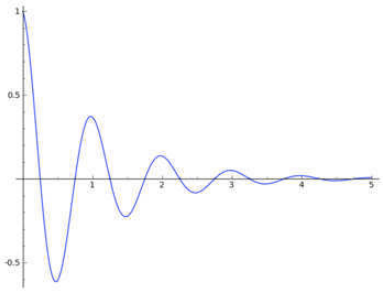


Figure 3 - Amplitude Variation

Tendo em conta estes parâmetros, o conjunto de dados sintéticos gerado tem as seguintes caracterísitcas :

- O conjunto de dados tem 50 classes que representam sons de 50 frequências diferentes, de 100Hz a 29500Hz com um 'salto' de 600Hz.
- O conjunto de dados tem 500 sons puros de parâmetros variáveis para cada frequência.
- Cada som tem 0,1 segundos de duração e taxa de amostragem de 44100Hz.
- A fase varia de 10% a 90% do período com um salto de 2,5%.
- A aplitude varia de 0,1 a 0,8 com saltos de 0,2.
- A variação de amplitude só representa amplitudes crescentes e decrescentes, embora existam infinitas possibilidades de comportamento deste parâmetro.

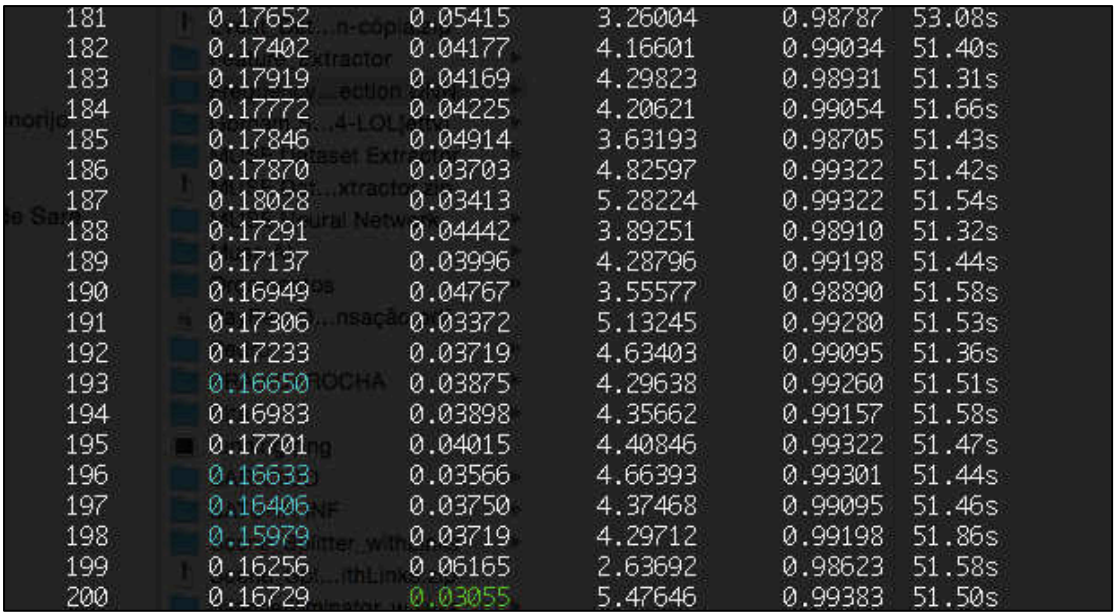
b) Classificação com Rede Neuronal Profunda

Para este problema de classificação foram testadas diferentes configurações de rede neural A rede que convergiu mais rápido e com melhores resultados (99,4%) é mostrada a negrito.

Input nodes	Hidden layers / nodes each	Activation function	Dropout Layers	Ouput Nodes/ function	Epochs	Learning rate	Momentum	Update Function	Accuracy
4410	2/800	rectify	7 (0.5)	50 softmax	200	0.1	0.9	Nesterov Momentum	8%
4410	6 / 800	tanh	7 (0.5)	50 softmax	200	0.01	-	Adam	11.1%
4410	9 / 1500	tanh	10 (0.5)	50 softmax	100	0.01	0.9	Nesterov Momentum	99.3%
4410	**6 / 800**	**tanh**	**7 (0.5)**	**50 softmax**	**200**	**0.01**	**0.9**	**Nesterov Momentum**	**99.4%**

Este é um problema de classificação multi-classe, significando que se trata de uma tarefa de classificação com mais de duas classes , sendo que cada amostra é atribuída a uma e somente uma 'label': um som com apenas uma frequência pode ter 100Hz ou 2500Hz, mas não ambas as frequências ao mesmo tempo.

Best Result :



No final do processo de treino, utilizamos a rede neuronal para fazer a classificação a um conjunto de dados de validação, que corresponderam a 30 sons puros (na gama de frequências treinadas) que foram retirados do YouTube, tendo retornado uma exatidão de 100% na classificação das mesmas.

2.DNNS PARA CLASSIFICAR SONS PUROS COM DIFERENTES FREQUÊNCIAS NO SEU CONTEÚDO;

a) Criação do Dataset

O conjunto de dados foi gerado com o mesmo script python referenciado na primeira tarefa, no entanto a criação dos diferentes sons baseia-se na adição de duas ou mais funções sinusoidais, considerando 5 frequências diferentes e parâmetros variáveis (variação de amplitude, fase e amplitude).

$$A\sin(2\pi f_1 t + \varphi) + A\sin(2\pi f_2 t + \varphi) + \cdots$$

As frequências consideradas foram 100Hz, 1600Hz, 3100Hz, 4600Hz, 6100Hz, 7600Hz e 9100Hz e foram consideradas algumas das combinações possíveis entre estas cincos frequências.

b) Classificação com Rede Neuronal Profunda

Ao contrário da rede anterior, esta rede é caracterizada por ter sido "desenhada" para fazer uma classificação multilabel, isto é, cada amostra é atribuída a um conjunto de 'labels' alvo. Isto pode ser pensado como uma propriedades de predição de uma amostra do conjunto de dados que não é mutuamente exclusiva,isto é, o som pode conter frequências diferentes. Por exemplo, o som composto sintéticamente pode ter frequência de 100Hz, 21200Hz ou 3120Hz ao

Input nodes	Hidden layers / nodes each	Activation function	Dropout Layers	Ouput Nodes/ function	Epochs	Learning rate	Momentum	Update Function	Accuracy
4410	6/400	tanh	7 (0.5)	7 sigmoid	45	0.1	0.9	Nesterov Momentum	68.4%
4410	6/800	tanh	7 (0.5)	7 sigmoid	45	0.01	-	Adam	80.2%
4410	6/800	tanh	7 (0.5)	7 sigmoid	45	0.01	0.5	Nesterov Momentum	63.1%
4410	**6 / 800**	**tanh**	**7 (0.5)**	**7 sigmoid**	**45**	**0.01**	**0.9**	**Nesterov Momentum**	**99.9%**

mesmo tempo ou apenas uma dela.

Exemplo de um vector de output com 3 frequências activas (1600Hz, 3100Hz e 6100Hz) :

[0 1 1 0 1 0 0]

100Hz 1600Hz 3100Hz 4600Hz 6100Hz 7600Hz 9100Hz

Melhor Resultado :

19	0.30711	0.04964	6.18619	0.99538	30.74s
20	0.29909	0.03194	9.36423	0.99678	32.75s
21	0.27634	0.03446	8.01869	0.99658	33.63s
22	0.27901	0.03913	7.13004	0.99652	30.96s
23	0.22613	0.02692	8.40054	0.99724	30.81s
24	0.22219	0.02573	8.63517	0.99754	31.05s
25	0.23262	0.02515	9.24769	0.99766	30.87s
26	0.21339	0.02857	7.46984	0.99752	30.87s
27	0.20651	0.01697	12.16545	0.99831	30.91s
28	0.18039	0.02235	8.07189	0.99797	31.43s
29	0.17229	0.01770	9.73601	0.99835	31.13s
30	0.18503	0.01796	10.30483	0.99828	30.98s
31	0.18179	0.01799	10.10615	0.99821	30.88s
32	0.16758	0.01480	11.32211	0.99847	30.94s
33	0.15242	0.01335	11.41543	0.99868	30.92s
34	0.15949	0.01436	11.10826	0.99858	30.79s
35	0.14811	0.01712	8.64984	0.99844	30.87s
36	0.14634	0.01700	8.60747	0.99856	30.87s
37	0.14053	0.01566	8.97491	0.99852	30.82s
38	0.12297	0.01055	11.66123	0.99894	30.94s
39	0.14720	0.01216	12.10835	0.99889	30.87s
40	0.12261	0.01175	10.43899	0.99886	30.95s
41	0.13861	0.01154	12.01096	0.99887	30.99s
42	0.11771	0.01057	11.13320	0.99896	31.22s

Bibliografia

[ASS16] Francesc Alias, Joan Claudi Socoró, and Xavier Sevillano. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. Disponível em: http://www.mdpi.com/2076-3417/6/5/143, 2016.

[Bap15] Fábio Baptista. Classificação de sons urbanos usando motifs e mfcc, 2015.

[Bar12] Lisarte Cristóvão Mendes Barbosa. Áudio digital : Uma abordagem ao áudio pela perspectiva de ensino, 2012.

[Ben09] Y. Bengio. Learning deep architectures for ai. 2009.

[Bez16] Eduardo Bezerra. Introdução à aprendizagem profunda. 2016.

[Bod17] Venkatesh Boddapati. Classifying environmental sounds with image networks, 2017.

[Bur12] Juan Jose Burred. Genetic motif discovery applied to audio analysis. 2012.

[Cak14] Emre Cakir. Multilabel sound event classification with neural networks. Disponível em: http://www.cs.tut.fi/~cakir/publications/Emre_CAKIR_master-science-thesis.pdf, 2014.

[Cam16] Lídio Campos. Uma metodologia biologicamente inspirada para projeto automático de redes neurais artificiais usando sistemas-l paramétricos com memória, 2016.

[Cis16] Cisco. Cisco visual networking index. Disponível em: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf, 2016.

[CKBK16] Inkyu Choi, Kisoo Kwon, Soo Hyun Bae, and Nam Soo Kim. DNN-based sound event detection with exemplar-based approach for noise reduction. Technical report, DCASE2016 Challenge, 2016.

[Cos13] Carlos Costa. Reconhecimento robusto de vogais isoladas, 2013.

[CWL06] Geng Cui, Man Wong, and Hon-Kwong Lui. *Machine Learning for Direct Marketing Response Models: Bayesian Networks with Evolutionary Programming*. Management Science, 2006.

[Dua16] Dami Duarte. Um estudo da relevancia da dinamica espectral na classificaçao de sons domesticos, 2016.

[Duc90] Albert Ducrocq. Music and discourse: Toward a semiology of music. In *Encyclopedie Larousse*. Editions Larousse, 1990.

[DZS+02] Nevenka Dimitrova, Hong-Jiang Zhang, Behzad Shahraray, Ibrahim Sezan, Thomas Huang, and Avideh Zakhor. Applications of video-content analysis and retrieval. Disponível em: `http://marco.uminho.pt/disciplinas/UCAN/BD/Artigos%20Recomendados/IEEEMMMagazinFinal.pdf`, 2002.

[Fer14] Artur Ferreira. Feature selection and discretization for high-dimensional data, 2014.

[GAFC+16] J.M. Gutierrez-Arriola, R. Fraile, A. Camacho, T. Durand, J.L. Jarrin, and S.R. Mendoza. Synthetic sound event detection based on MFCC. Technical report, DCASE2016 Challenge, 2016.

[gev] Gevent - coroutine-based python networking library. `http://www.gevent.org/`.

[GTGVBA+15] M. García-Torres, F. Gómez-Vela, D. Becerra-Alonso, B. Melián-Batista, and J. M. Moreno-Vega. Feature grouping and selection on high-dimensional microarray data. In *2015 International Workshop on Data Mining with Industrial Applications (DMIA)*, 2015.

[HDY+12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. Disponível em: `https://static.googleusercontent.com/media/research.google.com/pt-PT//pubs/archive/38131.pdf`, 2012.

[Hä07] Fabrício Pereira Härter. *Redes Neuronais Recorrentes aplicadas à assimilação de dados de dinâmica não linear*. PhD thesis, Institudo Nacional de Pesquisas Espaciais, 2007.

[KLR10] Peerapol Khunarsa, Chidchanok Lursinsap, and Thanapant Raicharoen. *Impulsive Environment Sound Detection by Neural Classification of Spectrogram and Mel-Frequency Coefficient Images*. 2010.

[las] Lasagne - lightweight library to build and train neural networks in theano. `http://lasagne.readthedocs.io/en/latest/`.

[Mat14] Jose Matos. Reconhecimento robusto de fala com redes de microfones em ambientes domésticos multi-sala, 2014.

[MZX+15] Ian McLoughlin, Haomin Zhang, Zhipeng Xie, Yan Song, and Wei Xiao. Robust sound event classification using deep neural networks, 2015.

[ngn] Ngnix http server. `https://nginx.org/en/`.

[nol] Nolearn python library. `https://pythonhosted.org/nolearn/`.

[Pic15] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015.

[Pye00] D. Pye. Content–based methods for the management of digital music, 2000.

[the] Theano - python library to define, optimize, and evaluate mathematical expressions. `http://deeplearning.net/software/theano/index.html`.

[VBK+13] Lode Vuegen, B Van Den Broeck, P Karsmakers, Jort F Gemmeke, B Vanrumste, and H Van Hamme. An mfcc-gmm approach for event detection and classification. Technical report, 2013.

[vdODZ+16] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. 2016.

[VW16] Toan H. Vu and Jia-Ching Wang. Acoustic scene and event recognition using recurrent neural networks. Technical report, DCASE2016 Challenge, 2016.

[WRF14] Yipei Wang, Shourabh Rawat, and Metze Florian. Exploring audio semantic concepts for event-based video retrieval, 2014.

[ZE16] Dongqing Zhang and Dan Ellis. Detecting sound events in basketball video archive, 2016.

[ZLL03] J. Zhang, L. Lu, and S. Z. Li. Content-based audio classification and segmentation by using support vector machines, 2003.

UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA

Contactos:
Universidade de Évora
Escola de Ciências e Tecnologia — ECT
Colégio Luis António Verney, Rua Romão Ramalho, nº59
7000-671 Évora | Portugal
Tel: (+351) 266 740 800
email: geral@ect.uevora.pt