UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS ALEXANDRE PERON DOS SANTOS

CLASSIFICAÇÃO AUTOMÁTICA DE CENAS ACÚSTICAS USANDO ALGORITMOS DE CLUSTERIZAÇÃO

MONOGRAFIA

CAMPO MOURÃO
2019

CARLOS ALEXANDRE PERON DOS SANTOS

CLASSIFICAÇÃO AUTOMÁTICA DE CENAS ACÚSTICAS USANDO ALGORITMOS DE CLUSTERIZAÇÃO

Trabalho de Conclusão de Curso de graduação apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Ms. Juliano Henrique Foleiss

CAMPO MOURÃO
2019

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Às **19:00** do dia **27 de novembro de 2019** foi realizada na sala **E103** da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação do(a) acadêmico(a) **Carlos Alexandre Peron Dos Santos** com o título **Classificação Automática de Cenas Acústicas Usando Algoritmos de Clusterização**. Estavam presentes, além do(a) acadêmico(a), os membros da banca examinadora composta por: **Prof. Ms. Juliano Henrique Foleiss** (orientador(a)), **Prof. Dr. Rodrigo Campiolo e Prof. Dr. Rogério Aparecido Gonçalves**. Inicialmente, o(a) acadêmico(a) fez a apresentação do seu trabalho, sendo, em seguida, arguido(a) pela banca examinadora. Após as arguições, sem a presença do(a) acadêmico(a), a banca examinadora o(a) considerou _____ na disciplina de Trabalho de Conclusão de Curso 2 e atribuiu, em consenso, a nota _____ (_____). Esse resultado foi comunicado ao (à) acadêmico(a) e aos presentes na sessão pública. A banca examinadora também comunicou ao (à) acadêmico(a) que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor responsável do TCC no prazo de **onze dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações:

Campo Mourão, **27 de novembro de 2019**

Prof. Dr. Rodrigo Campiolo
Membro 1

Prof. Dr. Rogério Aparecido Gonçalves
Membro 2

Prof. Ms. Juliano Henrique Foleiss
Orientador

A ata de defesa assinada encontra-se na coordenação do curso.

Resumo

Peron, Carlos. Classificação Automática de Cenas Acústicas Usando Algoritmos de Clusterização. 2019. 41. f. Monografia (Curso de Bacharelado em Ciência da Computação), Universidade Tecnológica Federal do Paraná. Campo Mourão, 2019.

O problema de Classificação de Cenas Acústicas consiste em atribuir um rótulo de ambiente a um sinal de áudio. Entre os rótulos estão parques, aeroportos, ruas e praças públicas, por exemplo. Neste trabalho são propostas quatro abordagens baseadas em aprendizagem de máquina e processamento de sinais para este problema. O objetivo principal é minimizar o custo computacional necessário para treinar os modelos e realizar predições, mantendo o desempenho da classificação em níveis aceitáveis. O principal método utilizado consiste em descrever os áudios com *Mel-Frequency Cepstral Coefficients* e depois agrupá-los com uma abordagem baseada em *K-means* em 2 níveis. Este agrupamento descreve as classes utilizando os sons comuns entre os áudios de cada classe, promovendo generalização e diminuindo a quantidade de dados necessária para geração do modelo, o que diminui o custo computacional do sistema. Esta abordagem reduziu a quantidade de dados necessários para o treinamento para pouco menos de 10% do total e obteve acurácia de 62% na base de dados *DCASE 2018 Task 1a*. Este resultado é comparável com os resultados obtidos no sistema *baseline*, que utiliza Redes Neurais Convolucionais.

Palavras chave: Aprendizagem de Máquina. Processamento de Sinais. Processamento de Áudio. Seleção de Instâncias por K-means.

Abstract

Peron, Carlos. Automatic Acoustic Scenes Classification Using Clustering Algorithms. 2019. 41. f. Monograph (Undergradute Program in Computer Science), Federal University of Technology – Paraná. Campo Mourão, PR, Brazil, 2019.

The Acoustic Scene Classification problem deals with assigning an environment-related label to an audio signal. Among the labels are parks, airports, streets and public squares. In this work we present four approaches to this problem based on machine learning and digital signal processing. Our main objective was to minimize the computing power required for model training and making predictions, while keeping classification performance at acceptable levels. Our highest performing method consists in describing audios with Mel-Frequency Cepstral Coefficients and then grouping them with a 2-level K-means clustering approach. This clustering approach describes classes using sounds that are common among audios of the same class. This promotes generalization and lowers the number of data points needed for model training. In turn, this lowers the system computing power requirements. This approach reduced the number of data points to around 10% of the total, and achieved 62% accuracy in the DCASE 2018 Task 1a dataset. This result is comparable with the results obtained by the *baseline* system, which is based on convolutional neural networks.
s

Keywords: Machine Learning. Signal Processing. Audio Processing. K-means Instance Selection.

Lista de figuras

Lista de tabelas

Siglas

CCA: Classificação de Cenas Acústicas
DCASE: *Detection and Classification of Acoustic Scenes and Events*
KNN: *K-Nearest Neighbors*
MARSYAS: *Music Analysis, Retrieval and Synthesis for Audio Signals*
MFCC: *Mel-Frequency Cepstral Coefficients*
RBF: *Radial Basis Function*
RNC: Rede Neural Convolucional
RPET: *Repeating Pattern Extraction Technique*
STFT: *Short Time Fourier Transform*
SVM: *Support Vector Machine*
UPG: Unidade de Processamento Gráfico

Sumário

Capítulo 1

Introdução

Sons carregam grande quantidade de informação sobre os ambientes do dia a dia. No entanto, isto dificulta o processamento computacional quando o objetivo é reconhecer padrões. Dependendo do que se deseja reconhecer, muitos dados são inúteis e portanto são considerados ruídos em diferentes contextos (DANG et al., 2018).

Sons são difíceis de modelar matematicamente (NGUYEN; PERNKOPF, 2018). Portanto, é difícil escrever programas que capturem apenas as regiões de interesse em um áudio. Além disso, dependendo das condições de captura, as representações digitais de áudio possuem perdas e nem sempre são fiéis ao som original. No entanto, existem formas de tratar sons usando a teoria de Processamento de Sinais, amplamente utilizada em várias áreas da Engenharia e Computação como processamento de sinais médicos, telecomunicações e processamento sísmico (MIRANDA, 2001).

1.1. Problema

A Classificação de Cenas Acústicas (CCA) é um problema cujo objetivo é reconhecer ambientes a partir de sinais sonoros. Por exemplo, dado um sinal de áudio, o objetivo é classificar o ambiente em que foi gravado (WALDEKAR; SAHA, 2018), este podendo ser um aeroporto, interior de um ônibus, o ambiente ao ar livre em uma praça pública, etc. Como não é possível modelar matematicamente um "ambiente" em função de sinais de áudio, técnicas baseadas em dados são interessantes. Uma dessas técnicas é a aprendizagem de máquina, que possibilita a inferência de padrões nos dados que estão correlacionados a conceitos de interesse (JAIN et al., 2000). No caso do problema de CCA, os dados são os sinais de áudio rotulados com os conceitos de interesse, que neste caso, são os ambientes. No entanto, algoritmos de processamento de sinais e aprendizagem de máquina podem ser computacionalmente custosos. Portanto, o desafio é desenvolver técnicas que são computacionalmente eficientes e

que possuam níveis de acerto aceitáveis para o problema.

1.2. Objetivos

O objetivo deste trabalho é desenvolver e avaliar um sistema de classificação automática de cenas acústicas, de forma a diminuir o custo computacional sem comprometer a acurácia do sistema. Especificamente, um sistema de aprendizagem de máquina é utilizado para aprender padrões nos áudios que correlacionam com as cenas acústicas. São implementadas e avaliadas abordagens de processamento de sinais que visam reduzir a quantidade de dados a serem usados. Desta forma, o custo de treinar e realizar predições com o sistema são reduzidos em relação a sistemas de aprendizagem de máquina modernos baseados em aprendizagem profunda.

1.3. Justificativa

Um sistema classificador de cenas acústicas pode ser usado em várias aplicações importantes, tais como: tornar dispositivos móveis sensíveis ao contexto (ERONEN et al., 2006), maior precisão no sistema de navegação de robôs (CHU et al., 2006), auxiliar aparelhos auditivos (WALDEKAR; SAHA, 2018), melhorias em sistema de vigilância (RADHAKRISHNAN et al., 2005). Além disso, essa implementação pode ajudar a distinguir e reconhecer ambientes acústicos quando outras modalidades de sinais não estão disponíveis, como informação visual em ambientes sem iluminação (HAO et al., 2018). O reconhecimento de ambientes com características previamente conhecidas pode também facilitar a detecção de eventos sonoros isolados nesses ambientes, onde uma significante parte do "barulho" ou "ruído" típico do ambiente, pode ser removido para análises posteriores desses eventos sonoros (YANG et al., 2018).

1.4. Organização do Texto

No Capítulo 1 o problema foi brevemente contextualizado. No Capítulo 2 são apresentadas técnicas de processamento de áudio e especificação de classificadores. No Capítulo 3 são apresentadas propostas de soluções para esse problema. No Capítulo 4 são apresentados os métodos utilizados. No Capítulo 5 são apresentados os resultados obtidos. Por fim, as conclusões são apresentadas no Capítulo 6.

Capítulo 2

Revisão Bibliográfica

Neste capítulo são apresentados alguns conceitos importantes no contexto de classificação automática de cenas acústicas. Primeiro são apresentadas algumas técnicas sobre processamento de áudio e extração de características. Também são apresentados conceitos relevantes sobre aprendizagem de máquina, aprendizagem supervisionada e aprendizagem não-supervisionada.

2.1. Transformada de Fourier

A transformada de Fourier é uma operação matemática usada para transformar a representação de sinais, em um dado período de tempo, em uma soma ponderada de senóides em várias frequências (MIRANDA, 2001). Assim, o resultado de uma transformada de Fourier é a representação de um determinado sinal convertido para o domínio da frequência (BAILEY; SWARZTRAUBER, 1994).

A transformada de Fourier de tempo curto, ou *Short Time Fourier Transform* (STFT), é a uma variação da transformada de Fourier tradicional que estabelece uma relação entre as amostras no domínio do tempo e um espaço tempo-frequência, ou seja, em 2 dimensões, de um sinal de áudio. Na STFT, a transformada de Fourier é calculada em pequenas amostras subsequentes, também chamadas de quadros, ou *frames*. Desta forma, cada janela representa uma pequena parte do sinal todo (MEHALA; DAHIYA, 2008).

Como a distribuição espectral varia bastante em diferentes instantes do tempo, é necessário uma representação que permita identificar essa mudança. Porém, ao segmentar o áudio em várias janelas, ocorre algo indesejado, conhecido como efeito de borda (NUTTALL, 1981), onde o ruído existente nos dados finais de uma janela e nos dados iniciais da janela subsequente, são evidenciados. A fim de suavizar esse efeito, pode-se aplicar a técnica de sobreposição ou *overlap*, que consiste em reusar a parte final dos dados da janela atual como o início da próxima janela, suavizando a transição entre elas (MEHALA; DAHIYA, 2008),

conforme ilustra a Figura 2.1.

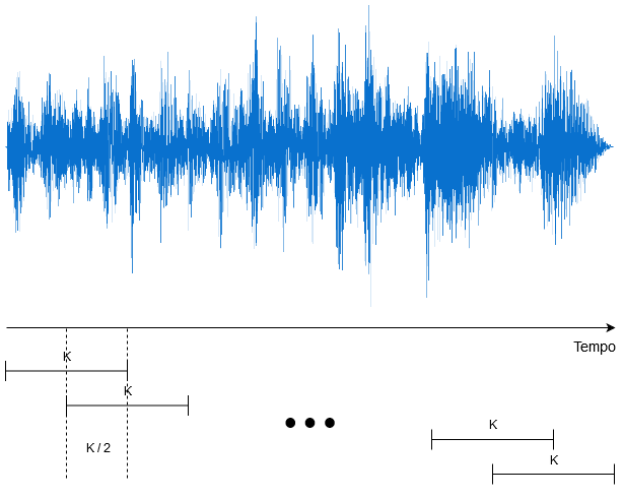


Figura 2.1. Exemplo do uso de *overlap*. K = quantidade de amostras de cada janela. *Overlap* = 50%

2.2. Descritores de áudio

Os descritores de áudio são ferramentas analíticas que tem como propósito identificar características que representam, descrevem e resumem um dado sinal de áudio (HERRERA et al., 1999). Segundo Simurra (2015), essas características podem ser obtidas através de várias formas de representações do áudio, como a partir do espectro de frequência ou tempo, por exemplo.

Os descritores de áudio são desenvolvidos com base no conhecimento especialista da área. Por exemplo, uma pessoa que conhece de processamento de música cria equações que representam determinado padrão que serve para descrever músicas. Ou seja, um descritor de áudio é uma forma de codificar conhecimento especialista (FOLEISS, 2018).

O objetivo dessas descrições do áudio é reduzir a complexidade da informação total e focar em aspectos específicos (SIMURRA, 2015). Por exemplo, sistemas de classificação de áudio usam descritores de áudio para codificar a representação espectral da STFT em vetores mais enxutos, contendo as informações necessárias para separação das classes de interesse.

2.2.1. *Mel-Frequency Cepstral Coefficients*

Mel-Frequency Cepstral Coefficients (MFCC) é um descritor útil para representar a amplitude do espectro de forma compacta, ou seja, com menos coeficientes (LOGAN, 2000). A distinção entre frequências não é percebida linearmente pelo ouvido humano. A sensibilidade nas frequências entre 0Hz e 1000Hz é aproximadamente linear. Acima de 1000Hz ela se torna logarítmica. Em outras palavras, a sensibilidade para distinguir entre frequências mais baixas

é maior do que para frequências mais altas. A escala Mel é utilizada para codificar a percepção humana em uma representação linear (STEVENS et al., 1937).

Para o processo de extração das características MFCC, as seguintes tarefas são necessárias:

- Janelar o sinal em *frames* de tempo curto.
- Mapear as potências do espectrograma obtido em uma escala Mel, onde esse mapeamento é linear para as frequências abaixo de 1 kHz, e logarítmico para as demais.
- Calcular o logaritmo das potências em cada frequências da escala Mel.
- Calcular a transformada discreta do cosseno (AHMED et al., 1974) da lista dos logaritmos de potências obtida, como se fossem um sinal de áudio.
- Extrair os valores de amplitudes do espectrograma gerado.

O número n de valores obtidos representam então o número de coeficientes MFCC gerados para cada *frame*.

2.2.2. *Marsyas*

Music Analysis, Retrieval and Synthesis for Audio Signals (MARSYAS) é um conjunto de descritores de áudio para o processamento de sinais de áudio em geral, mas com um enfoque em análise de sinais musicais (TZANETAKIS; COOK, 1999). Este conjunto teve grande contribuição na área de recuperação de informação musical, e até os dias atuais continua sendo referência nessa área.

Para descrever os áudios, características são extraídas levando em conta 3 aspectos: conteúdo de tom, conteúdo rítmico e textura de timbre. Embora as características de ritmo e tom sejam focadas em descrição musical, as características de timbre servem para descrever qualquer tipo de áudio. Portanto, para o problema de classificação automática de cenas acústicas elas podem ser usadas. Algumas das características usadas para descrever timbre são mostradas a seguir.

Spectral Centroid

O *Spectral Centroid*, ou Centroide Espectral, pode ser definido como o centro de gravidade da magnitude espectral de uma STFT. O *centroid* é uma medida do formato espectral sendo que valores mais altos de *centroids* correspondem a texturas mais "claras", de frequências altas. Podemos calcular o *centroid* a partir da Equação 2.1.

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{2.1}$$

Onde $M_t[n]$ é a magnitude da transformada de Fourier em um *frame* t de frequência n.

Spectral Rolloff

O *Spectral Rolloff* também é uma medida do formato espectral. Conforme mostrado na Equação 2.2, o *rolloff* é definido como a frequência R_t, no qual 85% da distribuição da magnitude está concentrada.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n] \tag{2.2}$$

Spectral Flux

O *Espectral Flux* é definido como o quadrado da diferença de duas sucessivas distribuições espectrais com magnitudes normalizadas. Assim, o *Spectral Flux* é uma medida da quantidade de mudanças ocorrida em um espectro local. A Equação 2.3 é usada para o cálculo do *Espectral Flux*:

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{2.3}$$

Onde $N_t[n]$ é a magnitude normalizada de uma transformada de Fourier em um *frame* t e $N_{t-1}[n]$ é a magnitude da transformada de Fourier do *frame* anterior.

2.3. Aprendizado de Máquina

Aprendizado de Máquina ou *Machine Learning* é a área de estudo que tem como objetivo prover a capacidade de "aprender" para os computadores, sem que eles tenham sido explicitamente programados, melhorando seu desempenho em tarefas específicas, conforme as realiza diversas vezes (MITCHELL, 1997). É uma forma de Inteligência Artificial com foco em estatística e reconhecimento de padrões. A aprendizagem de máquina funciona por meio de análise de conjuntos de dados, ao invés de projeto explícito de algoritmos para realização de tarefas. Caracteriza-se por prover soluções baseadas em dados para problemas difíceis e de larga escala como reconhecimento de fala, análise de mercados, análise de crédito, carros autônomos, sistemas *web* inteligentes, *business intelligence* (DAS et al., 2015).

Aprendizagem de máquina pode ser dividida em três tipos de aprendizagem diferentes: aprendizagem supervisionada, aprendizagem não-supervisionada e aprendizagem por reforço (HAYKIN; HAYKIN, 2009).

2.3.1. Aprendizagem Supervisionada

A aprendizagem supervisionada é usada quando o conjunto de dados disponível é constituído por exemplos e por rótulos para cada exemplo. O objetivo dessa forma de aprendizagem

é obter um mapeamento entre os exemplos e seus rótulos, de forma que exemplos que não foram usados para criar o mapeamento sejam rotulados corretamente (DUDA et al., 2000). Aprendizagem supervisionada é dividida entre problemas de classificação e de regressão. No caso de problemas de classificação, o mapeamento é realizado entre exemplos e um número finito de rótulos distintos, denominados classes. Já no caso de regressão, o mapeamento é realizado entre exemplos e um conjunto de valores em um espaço contínuo, como um espaço vetorial de números reais (DUDA et al., 2000).

O presente trabalho está relacionado a um problema de classificação: dado um sinal de áudio o objetivo é atribuí-lo a uma das classes conhecidas pelo modelo. A seguir são apresentados alguns classificadores utilizados no contexto de reconhecimento de padrões.

Support Vector Machine (SVM)

Support Vector Machine (SVM) é uma técnica de aprendizado de máquina embasada pela teoria de aprendizado estatístico. Essa teoria estabelece alguns princípios a serem seguidos na obtenção de classificadores com boa generalização. Assim, dada uma nova entrada existente no mesmo domínio em que as entradas anteriores utilizadas para o aprendizado ocorreram, o classificador pode aprender a qual classe essa entrada desconhecida pertence (CORTES; VAPNIK, 1995). A SVM tem como objetivo enfatizar a diferença existente entre dados pertencentes a classes distintas, calculando a chamada "fronteira de decisão". Essa fronteira é uma divisão linear que separa os dados pertencentes a classes diferentes, dando ênfase à aspectos comuns entre dados de mesma classe (LORENA; CARVALHO, 2007). Porém, em situações reais, os dados tendem a ser não linearmente separáveis. As SVMs lidam com esses casos mapeando o conjunto de dados de entrada, utilizados para o treinamento, em seu espaço original, para um novo espaço de maior dimensão. Ou seja, dado um conjunto de dados de entrada não linearmente separáveis, a SVM os mapeará em uma dimensão suficientemente alta onde seja possível separar todos os dados de forma linear (LORENA; CARVALHO, 2007). Isto torna as SVMs robustas mesmo diante de dados de grandes dimensões.

K-Nearest Neighbors (KNN)

O classificador *K-Nearest Neighbors* (KNN) usa o conceito de distância entre pontos para embasar um algoritmo simples de aprendizagem supervisionada. Um vetor de características pode ser visto como um ponto no espaço. Espera-se que pontos próximos no espaço pertençam à mesma classe. Em outras palavras, pontos cujo valor das características são semelhantes, tem maior probabilidade de pertencerem à mesma classe. A partir disso, o algoritmo KNN classifica um exemplo x de rótulo desconhecido em três etapas:

- O conjunto de distâncias D entre x e todos os pontos do conjunto de treinamento T é computado.

- Os k pontos de T equivalentes às menores distâncias em D são encontrados.
- Os rótulos desses k pontos do conjunto de treino são contabilizados em um histograma. A classe com a maior contagem é atribuída a x.

As principais vantagens do algoritmo KNN estão relacionadas à eficiência computacional. O gargalo do algoritmo está em ter que encontrar os k pontos mais próximos de x. No entanto, é possível usar estruturas de dados para diminuir o esforço necessário. Além disto, o KNN é um algoritmo simples de entender, cuja interpretação de resultados é direta por suas propriedades geométricas. No entanto, o KNN sofre com espaços de alta dimensionalidade. Dessa forma, é usual usar algoritmos de redução de dimensionalidade quando KNN é empregado como classificador (BIJALWAN et al., 2014). Um parâmetro importante para otimizar é o próprio parâmetro k, que indica a quantidade de vizinhos mais próximos a serem usados para montar o histograma de rótulos. Assim, é comum usar um procedimento de validação para escolher o valor de k apropriado para o problema. (HALL et al., 2008).

Redes Neurais Convolucionais (RNC)

Redes Neurais Convolucionais são um tipo especial de redes neurais especializadas no processamento de dados de duas dimensões, como imagens e áudios (PENG, 2018).

O termo Rede Neural Convolucional é atribuído a redes neurais que possuam camadas convolucionais. Uma arquitetura bastante utilizada é a *LeNet* (Lecun et al., 1998). Ela é composta por uma camada de entrada, seguida por uma ou mais camadas convolucionais. Uma camada de *pooling* sucede cada camada convolucional. Uma camada totalmente conectada normalmente sucede a última camada de *pooling*. Por fim, uma camada *Softmax* é usada como saída (Lecun et al., 1998). Uma ilustração da arquitetura *LeNet* pode ser vista na Figura 2.2

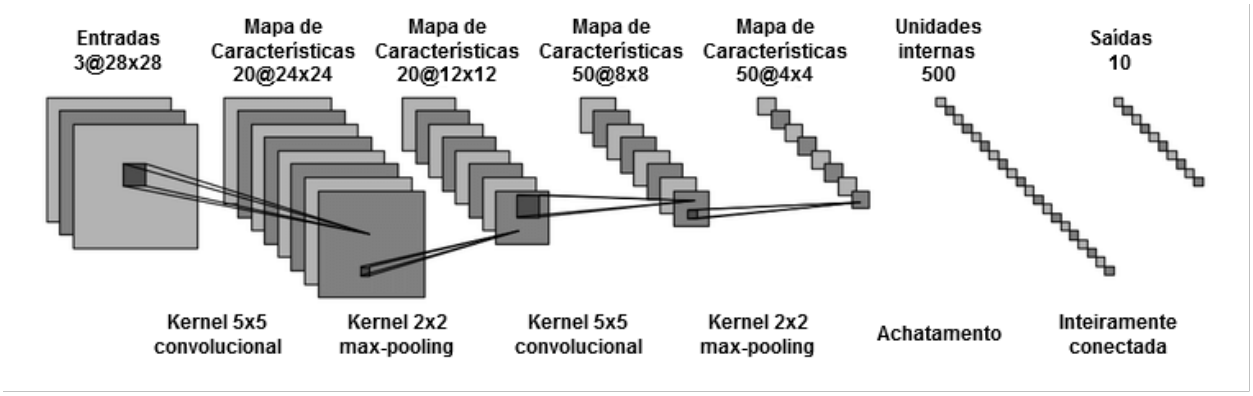


Figura 2.2. Ilustração da arquitetura *LeNet*. Fonte: (ZHOU et al., 2017)

As camadas convolucionais são compostas por um conjunto de filtros de convolução. Cada filtro representa uma característica aprendida pela rede neural (KHAN et al., 2019). Portanto, a saída de cada camada convolucional é chamada de Mapa de Características.

Para aplicar o mesmo filtro por toda a entrada, vários neurônios deslocados na imagem compartilham o mesmo conjunto de pesos, diminuindo a quantidade de parâmetros a serem treinados. Isto promove generalização e invariabilidade à translação (GOODFELLOW et al., 2016). As camadas de *pooling* são usadas para resumir os mapas de características e diminuir sua dimensionalidade. Como consequência, as camadas de *pooling* também corroboram para a generalização da rede, uma vez que detalhes específicos dos exemplos de treino são abstraídos, e as características tendem a ser mais genéricas. Isto também contribui para a invariabilidade à rotação (GOODFELLOW et al., 2016). As redes convolucionais representam o conhecimento aprendido de forma hierárquica. As camadas mais próximas da entrada ficam responsáveis por extrair características de mais baixo nível, enquanto as camadas mais próximas da saída representam conceitos mais abstratos (GOODFELLOW et al., 2016). Uma ilustração desse processo pode ser visto na Figura 2.3.

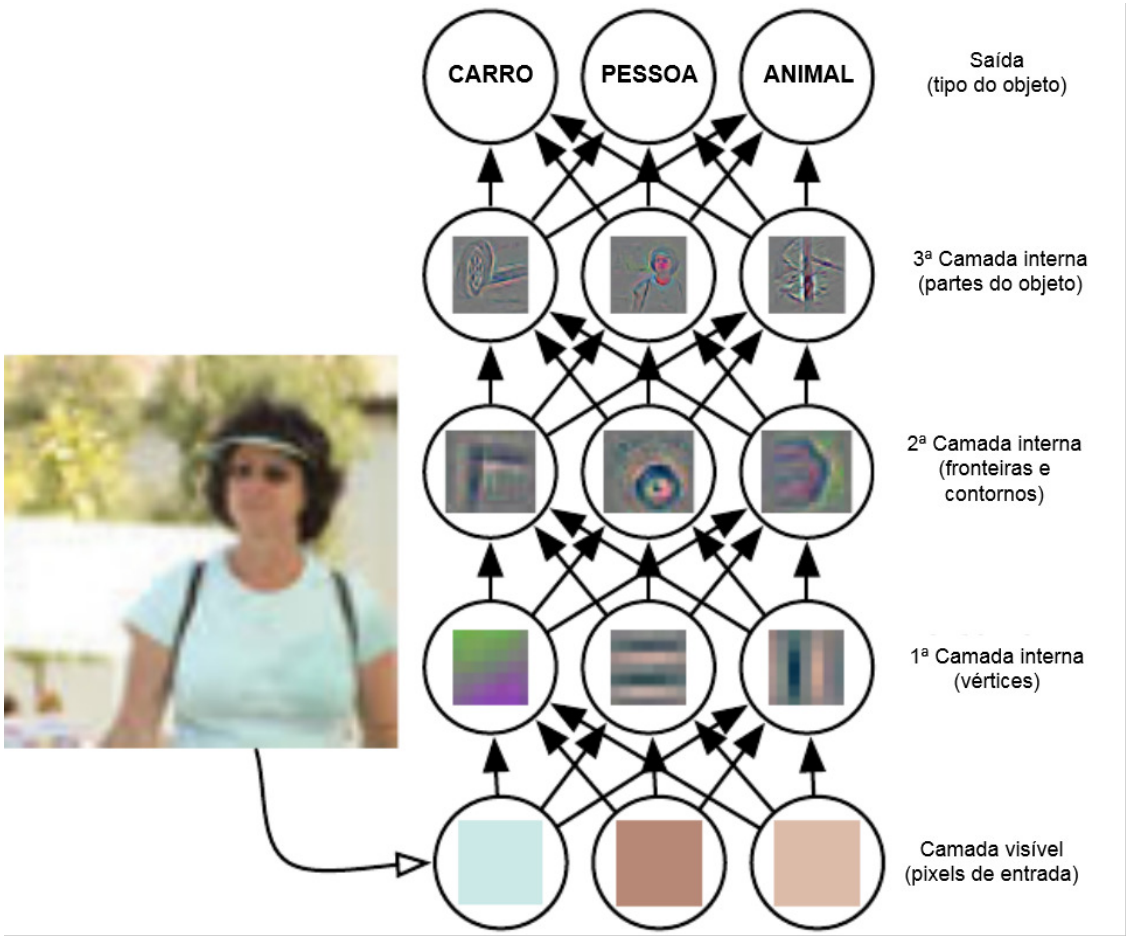


Figura 2.3. Ilustração de um modelo de aprendizagem profunda. Adaptado de (GOODFELLOW et al., 2016)

2.3.2. Aprendizagem Não-Supervisionada

Ao contrário da aprendizagem supervisionada, os métodos de aprendizagem não supervisionada não podem ser mapeados para pra um problema de regressão ou classificação, uma vez que os dados não estão rotulados (GHAHRAMANI, 2004). A aprendizagem não supervisionada é usada para inferir e reconhecer tendências nos dados, não conhecidas a priori (SATHYA; ABRAHAM, 2013). Um problema que surge é diferenciar padrões relevantes de padrões sem importância para a tarefa em questão. A aprendizagem não supervisionada é um processo que modela a estrutura central dos dados, descobrindo grupos de exemplos similares, ou determinando como esses dados estão espacialmente distribuídos (GHAHRAMANI, 2004). Uma das técnicas de aprendizagem não supervisionada é a de clusterização, ou agrupamento, que consiste em dividir dados não rotulados em grupos de acordo com alguma similaridade entre eles (CARON et al., 2018).

K-means

O algoritmo de clusterização *K-means* é um método não supervisionado, não determinístico e iterativo de clusterização que tem como objetivo agrupar dados de acordo com sua proximidade espacial (YADAV; SHARMA, 2013). Segundo Shinde e Tidke (2014), o processo para realizar esse agrupamento é exemplificado pelos seguintes passos:

1. Definir um número K de grupos;
2. Escolher K exemplos aleatórios do conjunto de dados. Estes exemplos são os usados como uma aproximação inicial dos centroides;
3. Calcular a distância de todos os exemplos do conjunto de dados para cada um dos centroides. Atribuir cada exemplo ao centroide mais próximo.
4. Atualizar a posição dos centroides. A nova posição de cada centroide x é dada pela posição média de todos os pontos atribuídos a ele no passo 3.
5. Voltar ao passo 3 caso o valor de algum centroide tenha sido alterado na última iteração.

Uma ilustração desse processo pode ser visto na Figura 2.4.

Quando os valores de todos os centroides permanecerem os mesmos em duas iterações seguidas, dizemos que o algoritmo convergiu (YADAV; SHARMA, 2013). O *K-means* é o algoritmo de clusterização mais utilizado pois é simples de implementar e executa em tempo linear em relação ao número de exemplos. O custo computacional é dado por $O(tkn)$ tal que t é o número de iterações, k é número de centroides e n é o número de exemplos no conjunto de dados.

Entretanto, o *K-means* também apresenta alguns pontos negativos (JAIN, 2008). Primeiramente, o algoritmo não é determinístico pois leva a conjuntos de centroides diferentes de acordo com a escolha aleatória dos exemplos iniciais. Outro problema é que é necessário predefinir o valor de K antes da execução do algoritmo. Por fim, o algoritmo é sensível

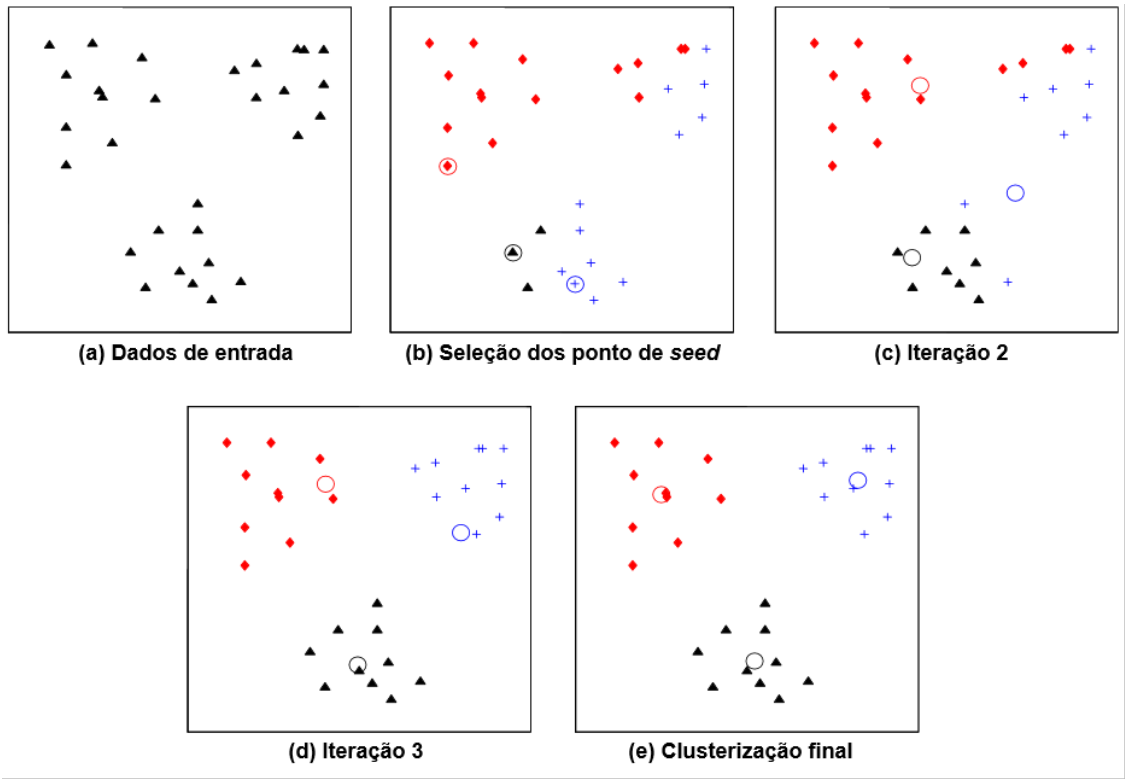


Figura 2.4. Ilustração do processo de clusterização do algoritmo *K-means*. (a) Dados de entrada de duas dimensões com 3 grupos; (b) Três pontos de *seed* selecionados como centroides e primeiro agrupamento dos dados; (c) e (d) Iterações intermediárias atualizando o rótulo dos dados e seus centroides; (e) Agrupamento final obtido pelo algoritmo *K-means* após a convergência. Fonte: (JAIN, 2008)

a *outliers*, também conhecidos como anomalias. Caso hajam anomalias no conjunto, os centroides podem ser deslocados em direção a elas, diminuindo sua expressividade.

2.4. Considerações

Foram apresentadas técnicas de processamento de áudio, além de descritores de áudios para aspectos rítmicos, de timbre e de tom. Também foram exibidas algumas técnicas de Aprendizado de Máquina, tanto para aprendizagem supervisionada quanto para aprendizagem não-supervisionada.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados alguns trabalhos que propõem soluções para classificação automática de cenas acústicas, como Dang et al. (2018), Nguyen e Pernkopf (2018) e Golubkov e Lavrentyev (2018).

Todos os trabalhos foram desenvolvidos usando a base de dados *TUT Urban Acoustic Scenes 2018* (HEITTOLA et al., 2018), a mesma usada neste trabalho.

3.1. *Acoustic Scene Classification Using Ensemble Of ConvNets*

No trabalho de Dang et al. (2018), é apresentada uma proposta para solucionar o problema de classificação de cenas acústicas. São apresentadas técnicas de processamento de áudio e modelos de classificadores focados em Redes Neurais Convolucionais.

Para a extração de informação dos áudios, esses foram processados de duas maneiras: por meio do sinal mono e por meio do sinal estéreo. As características usadas foram geradas a partir de *Log mel-spectrogram* e *gammatone based spectrogram* com *filter-banks* de 128 *bins*, usando uma taxa de amostragem de 48 KHz, janelamento de 2048 Hz e *overlap* de 1024 Hz. Os áudios originalmente com duração de 10 segundos, foram divididos em segmentos menores de 2 segundos cada.

Para a classificação dos áudios, foram utilizadas Redes Neurais Convolucionais, tanto para os sinais mono quanto para os estéreo. Para os sinais estéreo, uma Rede Neural Convolucional (RNC) com duas camadas convolucionais foi treinada usando os canais *left* e *right* e uma outra RNC, também com duas camadas convolucionais, para a soma dos canais $left + right$ e para a subtração dos canais $left - right$. Para os sinais mono, foram obtidos os componentes *harmonic* e *percussive* a partir dos canais *left* e *right*, e esses componentes foram usados para treinar mais duas redes neurais convolucionais, uma com apenas uma

camada convolucional e outra com duas. Todas as características de *Log mel-spectrogram* e *gammatone based spectrogram* foram extraídas e concatenadas em um único vetor, que serviu de entrada para o treinamento de cada RNC.

Depois de todos os classificadores treinados, cada um classifica os áudios individualmente e é contabilizado o voto majoritário entre eles, definindo assim, o rótulo equivalente a classe a qual o áudio provavelmente pertence. Considerando a quantidade de acertos em todas as classes, a acurácia média foi de 76.7%.

3.2. *Acoustic Scene Classification Using a Convolutional Neural Network Ensemble And Nearest Neighbor Filters*

A proposta apresentada por Nguyen e Pernkopf (2018) também faz uso de Redes Neurais Convolucionais e voto majoritário entre os classificadores.

Os sinais de áudio foram convertidos em algumas representações de tempo-frequência com segmentos de 1 segundo, como a gerada pela STFT comum, a obtida através da filtragem em escala *Mel*, e a obtida através da filtragem de *nearest neighbor*. A partir dessas representações, foram extraídas 128 características de *log mel-energies*. Para realização da STFT foram utilizados janelamentos de 40 ms com 20 ms de *overlap*. A taxa de amostragem foi mantida em 48 KHz. Todas as características obtidas dos espectrogramas foram convertidas em uma escala logarítmica e normalizada, subtraindo seu respectivo valor médio e dividido pelo seu respectivo desvio padrão. Adicionalmente, para detectar repetições nos áudios que acontecem intermitentemente ou em um período de tempo não fixo, foi utilizado o filtro de *nearest neighbor*, com base na técnica *Repeating Pattern Extraction Technique* (RPET). Mais informações sobre o funcionamento dessa técnica podem ser vistas em Rafii e Pardo (2012).

Uma vez em posse das 128 características de *log mel-energies*, a RNC foi treinada com essas características. Outra RNC de mesma estrutura foi treinada, só que dessa vez fazendo uso da versão filtrada pelo *nearest neighbor*. As RNCs, individualmente, fazem a classificação por voto, considerando 3 aspectos, as médias, os pesos das médias e a seleção do modelo, ou seja, para cada um desses aspectos, um par de RNC no formato descrito acima, foi treinado.

Após contabilizado os votos de todos os classificadores, a classe mais votada é enfim escolhida. Calculando a média de todas as classes, o desempenho obtido foi de 69.3% de acurácia.

3.3. *Acoustic Scene Classification Using Convolutional Neural Networks And Different Channels Representations And Its Fusion*

Neste trabalho, Golubkov e Lavrentyev (2018) apresentam uma abordagem diferente das anteriores, pois antes da conversão dos sinais de áudio para espectrogramas, é realizada uma escala de amplitude (de -1 até 1) nesses sinais.

A partir dos 2 canais de áudio disponíveis, *left* e *right*, foram geradas outras representações de canais, como a *middle*, obtida a partir da soma dos canais $left + right$, a *side*, obtida a partir da subtração dos canais $left - right$, e a *harmonic* e *percussive*, obtidas através do algoritmo de separação *harmonic-percussive sound separation* (TACHIBANA et al., 2014). Foram utilizadas 11 representações diferentes para os áudios. MFCC para o sinal mono, *Mel-spectrograms* para os sinais *left, right, middle, side, harmonic* e *percusive* e *CQT-spectrograms* para os sinais *left* e *right*.

Para cada áudio, 40 coeficientes de MFCC foram extraídos sem o uso de janelamento, ou seja, do trecho de áudio inteiro de 10 segundos. Para o cálculo do *Mel-spectrograms* entretanto, aplicou-se o janelamento com 2048 amostras, e *overlap* de 1024 amostras. O espectrograma resultante foi normalizado pela subtração do valor médio e dividido pelo valor do desvio padrão. O *CQT-spectrogram* foi gerado a partir 12 intervalos de frequência por cada oitava, com taxa de amostragem de 48 KHz.

As abordagens *Log-mel*, *mel-spectrogram* e MFCC apresentaram resultados muito parecidos, próximo de 65% de acurácia.

3.4. Considerações

Um resumo das técnicas utilizadas e os resultados apresentados pelos trabalhos relacionados são mostrados na Tabela 3.1.

Tabela 3.1. Técnicas e resultados dos trabalhos relacionados

Trabalho	**Classificador**	**Características**	***Hardware***	**Pós-processamento**	**Acurácia**
1	RNC	*Log mel-spectrogram, Gammatone based spectrogram*	UPG	*Ensemble*	76,7 %
2	RNC	*Log mel-energies*	UPG	*Ensemble*	69,3 %
3	RNC	MFCC, *Mel-spectrograms, CQT-spectrograms*	UPG	*Ensemble*	65 %

As abordagens mais eficazes propostas para solucionar o problema de classificação de cenas acústicas utilizam aprendizado de máquina supervisionado. Elas fazem o uso de classificadores baseados em arquiteturas modernas de redes neurais, como redes neurais convolucionais, que demandam grande capacidade computacional. Com estas abordagens, aplicações em dispositivos com poder computacional inferior são inviáveis, tanto para treinar

os modelos quanto para fazer predições em tempo real. Desta forma, neste trabalho propomos e avaliamos sistemas que podem ser usados para fazer predições em computadores com menor poder computacional, como computadores pessoais e dispositivos móveis.

Capítulo 4

Metodologia

Os trabalhos com os melhores resultados propostos para solucionar o problema de classificação de cenas acústicas, fazem uso, em sua grande maioria, de classificadores que demandam grandes quantidades de processamento. Normalmente, esses sistemas utilizam redes neurais para extração de características e classificação. Em específico, vários sistemas usam redes neurais convolucionais (NGUYEN; PERNKOPF, 2018), (DANG et al., 2018) e (YANG et al., 2018), e redes neurais recorrentes (ROMA et al., 2013), (ADAVANNE et al., 2017) e (REN et al., 2017). Entretanto, estes sistemas são computacionalmente custosos e requerem *hardware* dedicado, apesar de serem consideravelmente eficazes.

A proposta deste trabalho é apresentar uma abordagem computacionalmente mais eficiente para este problema usando classificadores como *KNN* e *SVM*. A ideia é encontrar uma arquitetura de extração de características e de classificadores que dê resultados competitivos, mas com menor custo computacional para classificação do que os métodos baseados em redes neurais.

4.1. Base de dados

Detection and Classification of Acoustic Scenes and Events (DCASE) é um simpósio de desafios relacionados à classificação de áudios. Cada desafio acompanha conjuntos de dados bem documentados e rotulados. Em 2018 houveram 5 desafios: *Acoustic scene classification*, *General-purpose audio tagging of Freesound content with AudioSet labels*, *Bird audio detection*, *Large-scale weakly labeled semi-supervised sound event detection in domestic environments* e *Monitoring of domestic activities based on multi-channel acoustics*.

Para o desafio de *Acoustic Scene Classification*, foi disponibilizada a base de dados *TUT Urban Acoustic Scenes 2018* (HEITTOLA et al., 2018), coletado pela *Tampere University of Technology* entre janeiro e março de 2018, que contém gravações de sons ambiente de

diversos locais em 6 cidades europeias: Barcelona, Helsinque, Londres, Paris, Estocolmo e Viena. Em cada uma dessas cidades, entre 5 e 6 minutos de áudio foram capturados em 10 ambientes em comum: aeroportos, shoppings, estações de metrô, calçadas de pedestres, praças públicas, parques, ruas com tráfego de automóveis, trens em movimento, ônibus em movimento e bondinhos em movimento. Cada gravação foi dividida em vários segmentos de apenas 10 segundos, e foram agrupados em 3 conjuntos: teste (6122), treino (2518) e avaliação (2518), totalizando 11.158 áudios. Todos os sinais de áudio foram capturados a uma taxa de amostragem de 48 KHz e resolução de 24 bits.

4.2. *Baseline*

Além da base de dados, a DCASE (MESAROS et al., 2018) também fornece um sistema *baseline*, que opcionalmente pode servir como ponto de partida para futuras implementações de sistemas para classificação automática de cenas acústicas. Para a fragmentação dos áudios, o *baseline* utilizou janelas de 40ms com 50% de *overlap* e como características foram usadas *Log mel-bands energies* extraídas com 40 bandas.

O classificador utilizado foi uma rede neural convolucional, que consiste em 2 camadas convolucionais 2D e uma camada totalmente conectada. Essa rede neural tem uma entrada de tamanho 40x500, que é o equivalente ao número de características *Log mel-bands energies* extraídas dos 500 *frames* no intervalo. O classificador foi treinado usando o *Adam Optimizer* (KINGMA; BA, 2014), com uma taxa de aprendizagem de 0.001. A primeira camada convolucional possui 32 filtros, e tamanho do *kernel* igual a 7. Já a segunda camada convolucional possui 64 filtros, e tamanho do *kernel* também igual a 7.

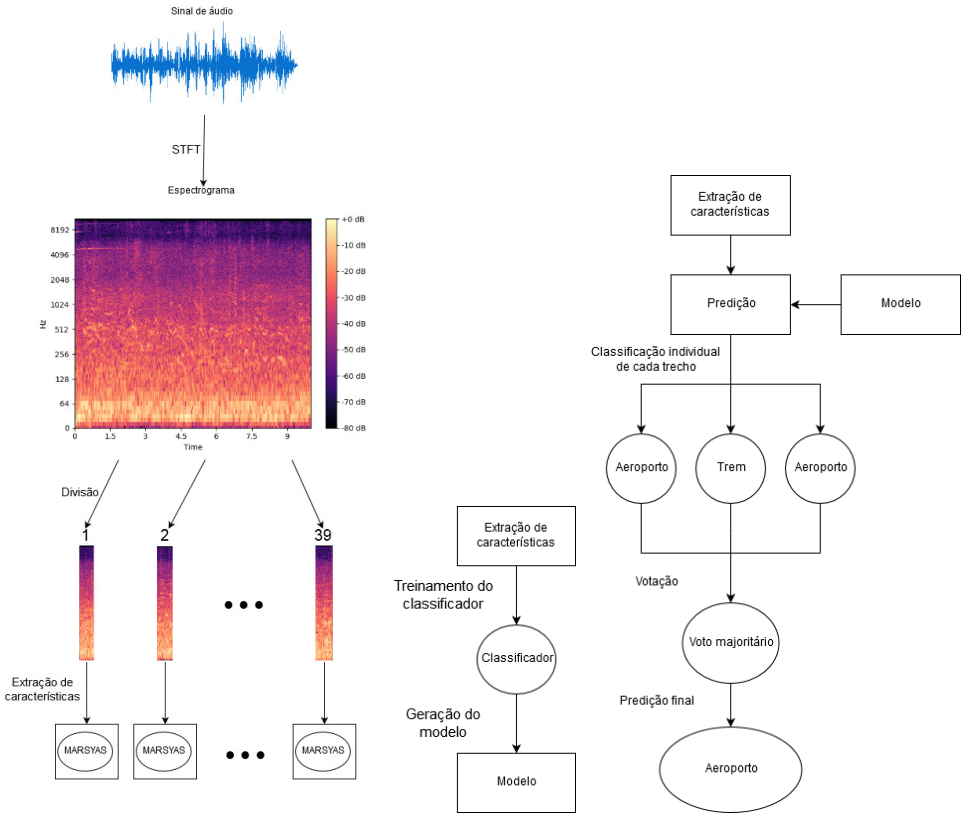
O resultado obtido pelo *baseline* na classificação dos áudios da base de dados *TUT Urban Acoustic Scenes 2018* apresenta acurácia média de 59,7%.

4.3. Experimento 1

Para cada entrada do conjunto de treino, foram obtidas as informações espectrais da relação tempo/frequência a partir do processamento da STFT dos áudios, com o tamanho da janela $TJ = 2048$ e sobreposição de 25% entre janelas adjacentes.

A matriz obtida após a realização desse processo, na qual cada coluna representa um instante no tempo e cada linha representa o valor de determinada frequência do sinal de áudio, foi então subdivida em 39 matrizes, onde cada matriz representa aproximadamente 5% do tempo de duração da matriz original, com 50% de sobreposição. Para cada uma dessas matrizes, foi extraído o conjunto de características *MARSYAS* de todas as colunas respectivamente. Cada matriz foi resumida em um único vetor-coluna. Este vetor é obtido pela média das características no tempo. Por fim, o áudio é representado pelo conjunto

de 39 vetores de características, cada um com 28 características. Todos os vetores de características do conjunto de treino são normalizados usando *z-score*. A normalização é realizada independentemente em cada característica, uma vez que possuem escalas distintas. No treinamento, os vetores de características de um mesmo áudio são apresentados como exemplos independentes para o classificador e todos recebem o rótulo do áudio completo. No procedimento de teste, a extração de características é realizada da mesma forma. A normalização usa os mesmos parâmetros encontrados no conjunto de treino. A predição é realizada em duas etapas. Na primeira, é realizada a predição individual dos vetores de características de um mesmo áudio. Na segunda etapa, o rótulo para o áudio completo é decidido por meio de voto majoritário. O processo utilizado no experimento 1 é ilustrado pela Figura 4.1.



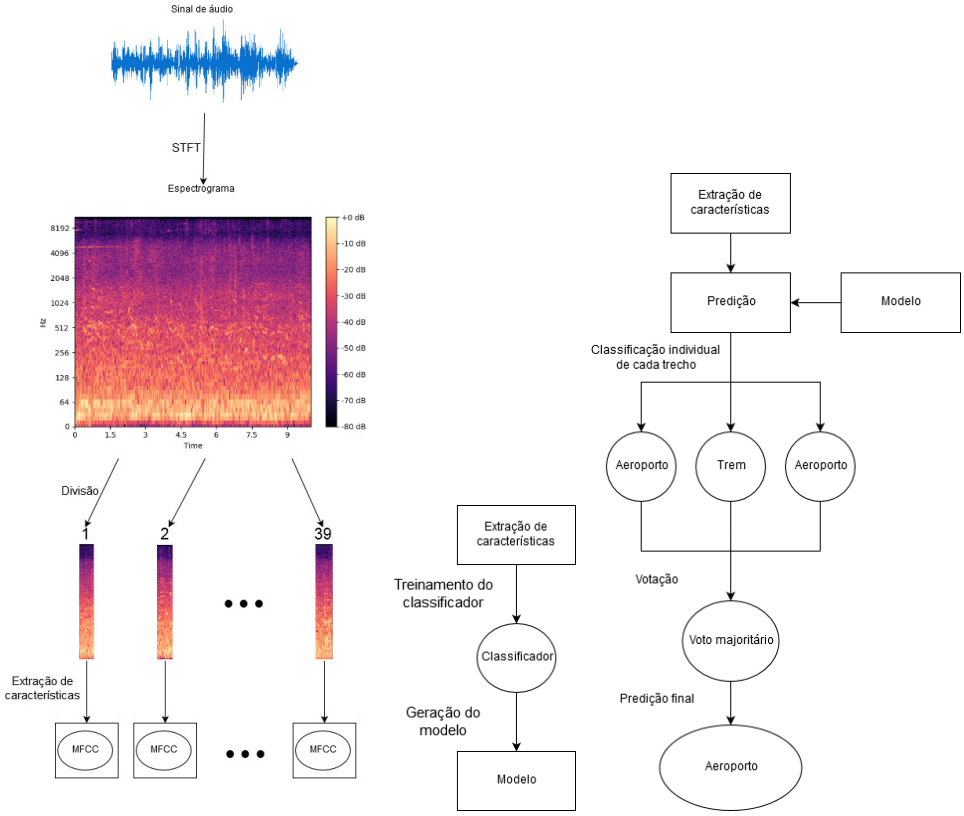
(a) Extração de características (b) Treino do modelo (c) Classificação

Figura 4.1. Fluxo de processamento do experimento 1

4.4. Experimento 2

Para cada áudio do conjunto de treino foram obtidas as informações espectrais da relação tempo/frequência a partir do processamento da STFT dos áudios, com o tamanho das janelas TJ 2048 e sobreposição de 25%. Uma matriz é obtida após a realização desse processo, onde

cada coluna representa um instante no tempo e cada linha representa o valor de determinada frequência do sinal de áudio. Essa matriz foi então subdivida em 10 matrizes iguais e para cada uma dessas matrizes, foram extraídas as características de MFCC, com coeficiente $c = 8$. Cada matriz foi resumida em um único vetor-coluna. Este vetor é obtido pela média das características no tempo. Por fim, o áudio é representado pelo conjunto de 10 vetores de características, cada um com 8 características. Todos os vetores de características do conjunto de treino são normalizados usando *z-score*. A normalização é realizada independentemente em cada característica, uma vez que possuem escalas distintas. No treinamento, os vetores de características de um mesmo áudio são apresentados como exemplos independentes para o classificador e todos recebem o rótulo do áudio completo. No procedimento de teste, a extração de características é realizada da mesma forma. A normalização usa os mesmos parâmetros encontrados no conjunto de treino. A predição é realizada em duas etapas. Na primeira, é realizada a predição individual dos vetores de características de um mesmo áudio. Na segunda etapa, o rótulo para o áudio completo é decidido por meio de voto majoritário. O processo utilizado no experimento 2 é ilustrado pela Figura 4.2.



(a) Extração de características (b) Treino do modelo (c) Classificação

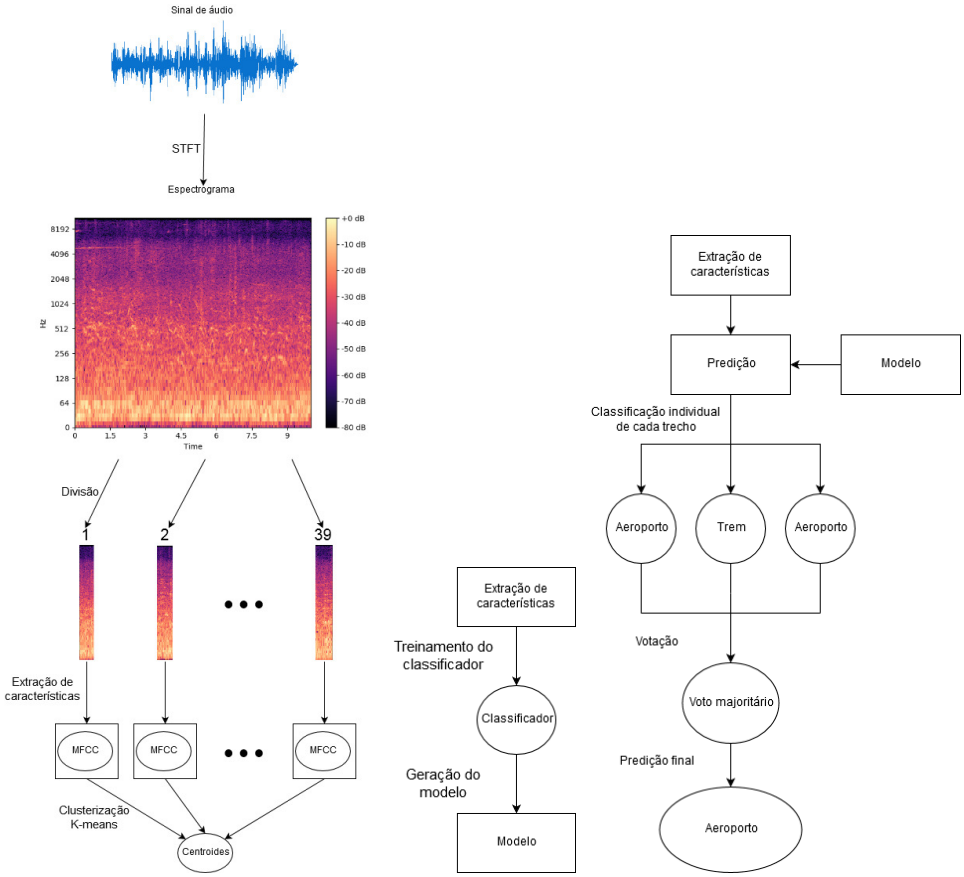
Figura 4.2. Fluxo de processamento do experimento 2

4.5. Experimento 3

Para cada entrada do conjunto de treino, foram obtidas as informações espectrais da relação tempo/frequência a partir do processamento da STFT dos áudios, com o tamanho da janela TJ 2048 e sobreposição de 25%. A matriz obtida após a realização desse processo, na qual cada coluna representa um instante no tempo e cada linha representa o valor de determinada frequência do sinal de áudio, foi então divida em 39 partes, onde cada parte possui aproximadamente 5% da matriz original, e 50% de sobreposição. Para cada uma dessas sub matrizes, foram extraídas as características de MFCC, com coeficiente $c = 8$, de todas as colunas respectivamente. Cada matriz foi resumida em um único vetor-coluna. Este vetor é obtido pela média das características no tempo. Por fim, o áudio é representado pelo conjunto de 39 vetores de características, cada um com 8 características. Todos os vetores de características do conjunto de treino são normalizados usando *z-score*. A normalização é realizada independentemente em cada característica, uma vez que possuem escalas distintas. Após a normalização dos dados, eles foram usados para o processamento do algoritmo *K-means* com k variando entre 5, 10, 20 e 39 para cada áudio independentemente. No treinamento, os vetores de características de um mesmo áudio são apresentados como exemplos independentes para o classificador e todos recebem o rótulo do áudio completo. No procedimento de teste, a extração de características é realizada da mesma forma. A normalização usa os mesmos parâmetros encontrados no conjunto de treino. A predição é realizada em duas etapas. Na primeira, é realizada a predição individual dos vetores de características de um mesmo áudio. Na segunda etapa, o rótulo para o áudio completo é decidido por meio de voto majoritário. Este experimento foi executado três vezes a fim de obter diferentes resultados ao alterar os valores iniciais dos centroides do algoritmo *K-means*. O processo utilizado no experimento 3 é ilustrado pela Figura 4.3.

4.6. Experimento 4

Para cada entrada do conjunto de treino, foram obtidas as informações espectrais da relação tempo/frequência a partir do processamento da STFT dos áudios, com o tamanho da janela TJ 2048 e sobreposição de 25%. A matriz obtida após a realização desse processo, na qual cada coluna representa um instante no tempo e cada linha representa o valor de determinada frequência do sinal de áudio, foi então divida em 39 partes, onde cada parte possui aproximadamente 5% da matriz original, e 50% de sobreposição. Para cada uma dessas sub matrizes, foram extraídas as características de MFCC, com coeficiente $c = 8$, de todas as colunas respectivamente. Cada matriz foi resumida em um único vetor-coluna. Este vetor é obtido pela média das características no tempo. Por fim, o áudio é representado pelo conjunto de 39 vetores de características, cada um com 8 características. Todos os vetores



(a) Extração de características (b) Treino do modelo (c) Classificação

Figura 4.3. Fluxo de processamento do experimento 3

de características do conjunto de treino são normalizados usando *z-score*. A normalização é realizada independentemente em cada característica, uma vez que possuem escalas distintas. Após a normalização dos dados, eles foram usados para o processamento do algoritmo *K-means* com k igual a 31 para cada áudio independentemente. Os valores dos centroides, de todos os áudios, obtidos nesse processo, foram divididos em 10 grupos, de acordo com a classe a qual cada áudio pertence. Para cada um desses grupos, um novo processamento com o algoritmo *K-means* foi realizado, com $k2$ variando entre 50, 100, 200, 500, 1.000, 2.000, 5.000 e 10.000. Um exemplo desse processamento em 2 níveis do algoritmo *K-means*, com menos exemplos, pode ser visto na Figura 4.4.

No treinamento, os vetores de características de um mesmo áudio são apresentados como exemplos independentes para o classificador e todos recebem o rótulo do áudio completo. No procedimento de teste, a extração de características é realizada quase da mesma forma, com exceção do segundo procedimento com *K-means*. A normalização usa os mesmos parâmetros encontrados no conjunto de treino. A predição é realizada em duas etapas. Na primeira, é realizada a predição individual dos vetores de características de um mesmo áudio. Na segunda etapa, o rótulo para o áudio completo é decidido por meio de voto majoritário. Este experimento foi executado três vezes a fim de obter diferentes resultados ao alterar os

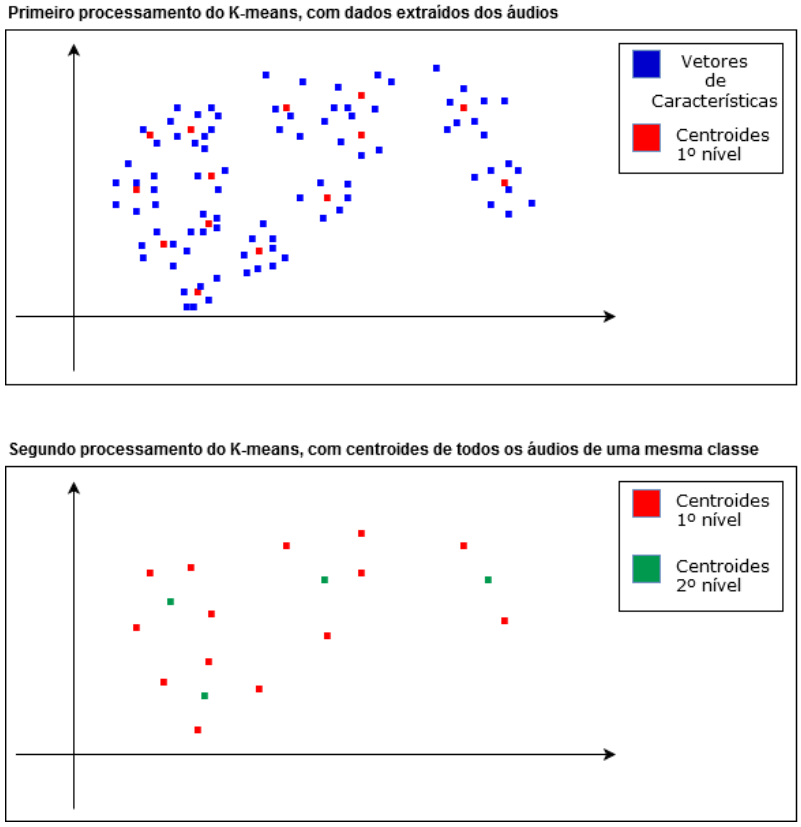


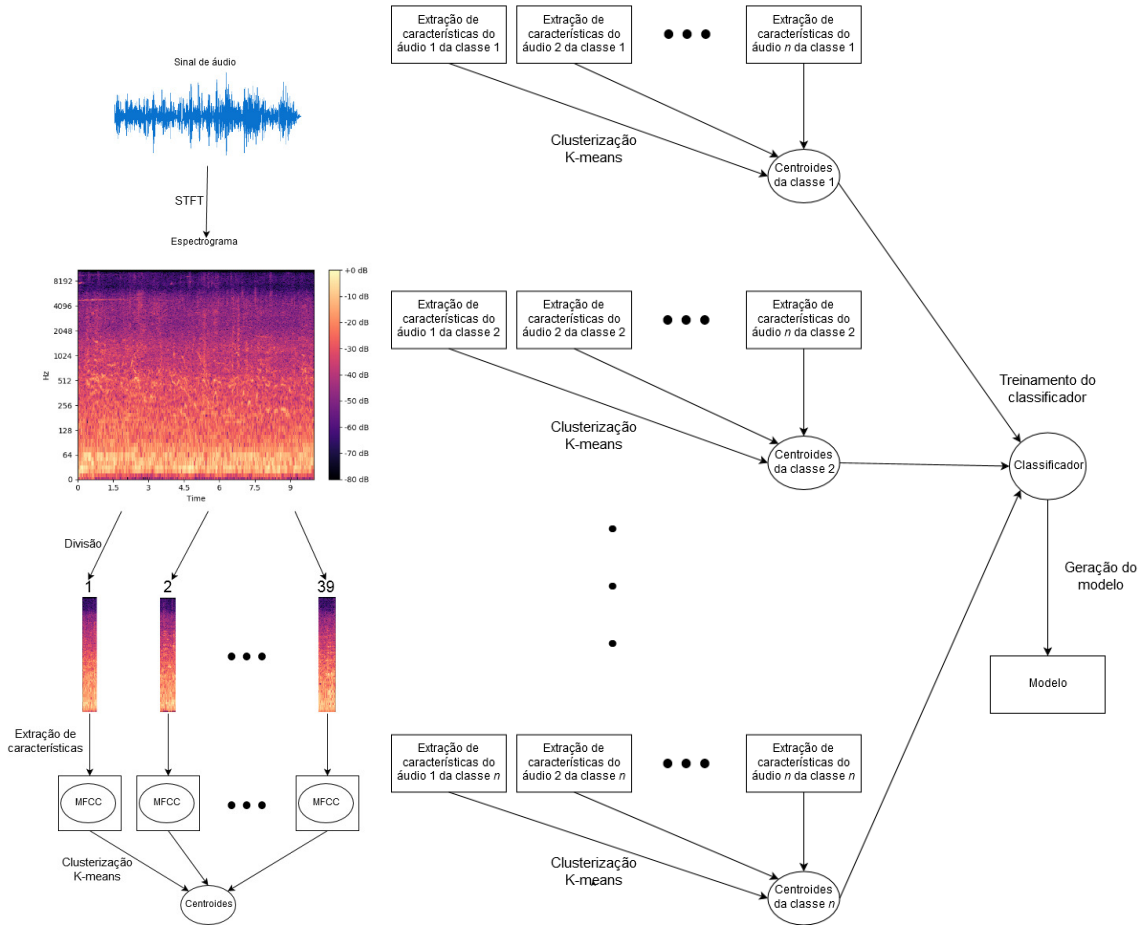
Figura 4.4. Ilustração do processamento do algoritmo *K-means* em 2 níveis

valores iniciais dos centroides do algoritmo *K-means*. O processo utilizado no experimento 4 é ilustrado pela Figura 4.5.

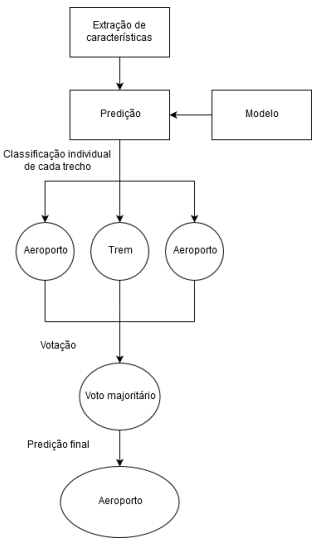
4.7. Considerações

Este trabalho apresenta soluções de baixo custo computacional para o problema de CCA. Desta forma, diferentemente dos trabalhos relacionados, optamos por não utilizar redes neurais como classificadores. Apesar de apresentarem bons resultados, o treino de redes neurais requer *hardware* especial, como Unidade de Processamento Gráfico (UPG). Portanto, escolhemos SVM e KNN como classificadores. Dessa forma, todos os experimentos foram repetidos a fim de testar o desempenho do sistema com 2 tipos de classificadores, o SVM, com $gamma = 0,01$, $C = 1,0$ e *kernel Radial Basis Function* (RBF), e o KNN, com $K = 15$.

Além dos quatro métodos apresentados nesta Seção, também foram avaliadas outras estratégias. No entanto, elas não foram apresentadas por serem pequenas variações dos métodos apresentados ou por não obterem resultados competitivos.



(a) Extração de características (b) Treino do modelo



(c) Classificação

Figura 4.5. Fluxo de processamento do experimento 4

Capítulo 5

Resultados e Discussão

Neste Capítulo são apresentados os resultados de todos os experimentos, além de uma breve discussão sobre o que esses resultados representam.

5.1. Resultados do Experimento 1

Os resultados do experimento 1 são apresentados na Tabela 5.1.

Tabela 5.1. Acurácia (em %) dos classificadores SVM e KNN para o experimento 1

Classificador	Acurácia
SVM	43,17
KNN	39,24

Os resultados são consideravelmente piores que os resultados obtidos nos trabalhos relacionados e pelo sistema *baseline*. Embora as características *MARSYAS* sejam amplamente utilizadas em classificação de músicas, elas não parecem ser eficientes para o problema de CCA. Isto mostrou a necessidade de testar outros conjuntos de características.

5.2. Resultados do Experimento 2

Os resultados do experimento 2 são apresentados na Tabela 5.2.

Tabela 5.2. Acurácia (em %) dos classificadores SVM e KNN para o experimento 2

Classificador	Acurácia
SVM	50,75
KNN	45,11

Neste experimento, foram usadas as características MFCC, que são uma representação mais compacta do timbre dos áudios. Os resultados mostram que este descritor atingiu

resultados significativamente superiores aos do experimento 1. Ao dividir os áudios em segmentos linearmente espaçados e, posteriormente aplicar o voto majoritário simples, assumimos que todos os segmentos são igualmente representativos para a classe do áudio. Entretanto, nem todos os segmentos do áudio possuem a mesma representatividade. Desta forma, notamos a necessidade de escolher quais segmentos serão provavelmente mais informativos para a discriminação entre as classes.

5.3. Resultados do Experimento 3

Os resultados do experimento 3 podem vistos na Tabela 5.3.

Tabela 5.3. Média de acurácia (em %) e desvio padrão dos classificadores SVM e KNN para o experimento 3

Classificador \ K	5	10	20	39
SVM	59.28 ±0.67	60.34 ±0.68	60.71 ±0.72	61.23 ±0.51
KNN	56.04 ±0.64	56.65 ±0.63	57.27 ±0.68	58.18 ±0.55

Neste experimento usamos o algoritmo de clusterização *K-means* para identificar os diferentes sons que compõem cada um dos áudios. Os resultados obtidos são significativamente melhores que os resultados obtidos nos dois primeiros experimentos. Nota-se também que, conforme o número de centroides aumenta, a acurácia dos classificadores também aumenta. O uso de centroides para representar cada áudio aumenta o alcance no espaço de características em relação à escolha dos segmentos linearmente espaçados do experimento anterior. Isto promove generalização, uma vez que agora uma região maior do espaço de características está representado (FOLEISS; TAVARES, 2019).

A quantidade de vetores de características de cada classe pode ser vista na Tabela 5.4. A Tabela 5.5 mostra o total de vetores usados no treinamento para cada valor de k. As Tabelas 5.3 e 5.5 mostram que conforme o k aumenta, o desempenho também melhora, embora o número de vetores usados para o treino também aumenta. Além disto, nota-se que o ganho de desempenho conforme k aumenta, é pequeno, considerando que muito mais vetores são usados para o treinamento.

5.4. Resultados do Experimento 4

Os resultados do experimento 4 são apresentados na Tabela 5.6. São apresentadas as acurácias médias do experimento para cada valor de $k2$, juntamente com seus respectivos desvios-padrão.

Nota-se que alguns resultados dessa abordagem possuem acurácia superior aos resultados do experimento anterior, principalmente quando o valor de $k2$ aumenta, onde é percebido uma tendência de aumento na acurácia dos classificadores. Podemos perceber

Tabela 5.4. Quantidade de vetores de características por classe

Classe	Número de Vetores de Características
Aeroporto	23361
Ônibus	24258
Metrô	23517
Estação de Metrô	23595
Parque	24258
Praça Pública	25272
Shopping	22815
Calçada	24063
Rua	24102
Trem	23517
Total	238758

Tabela 5.5. Percentual da quantidade total de vetores de características usados no treino para cada valor de k

K	% do total de vetores
5	12,82
10	25,64
20	51,28
39	100,00

Tabela 5.6. Média de acurácia (em %) e desvio padrão dos classificadores SVM e KNN para o experimento 4

K2 / Classificador	50	100	200	500	1.000	2.000	5.000	10.000
SVM	54,68 ±0,50	57,43 ±0,42	59,53 ±0,27	59,05 ±0,33	59,32 ±0,28	60,05 ±0,28	60,76 ±0,31	62,09 ±0,32
KNN	57,24 ±0,28	58,44 ±0,28	59,33 ±0,25	61,34 ±0,48	62,14 ±0,22	62,62 ±0,24	62,34 ±0,23	62,58 ±0,28

também, que diferentemente de todos os outros experimentos anteriores, de forma geral, o classificador KNN obteve resultados superiores aos do classificador *SVM*.

A Tabela 5.7 mostra o total de vetores usados no treinamento para cada valor de $k2$. As Tabelas 5.6 e 5.7 mostram que conforme $k2$ aumenta, o desempenho dos classificadores também aumenta, assim como o percentual total dos vetores usados para o treino. A melhora com KNN atinge um platô com aproximadamente 2.000 centroides por classe, enquanto com *SVM* a melhora é observada até 10.000 centroides.

5.5. Discussão

Uma comparação entre os melhores resultados obtidos de todos os experimentos, juntamente com o resultado obtido pelo sistema *baseline*, pode ser vista na Tabela 5.8.

A utilização do conjunto de características *MARSYAS* não se mostrou muito útil para o problema de CCA, visto que ele descreve vários aspectos de sinais de áudio que não são interessantes para o problema, ao contrário das características de MFCC, que, usadas

Tabela 5.7. Percentual da quantidade total de vetores de características usados no treino para cada valor de $k2$

K2	% do total de vetores
50	0,21
100	0,42
200	0,84
500	2,09
1000	4,19
2000	8,38
5000	20,94
10000	41,88

Tabela 5.8. Comparativo da acurácia (em %) de todos os experimentos e do sistema *baseline*. Também são mostrados a porcentagem do total de vetores usados para o treino.

Experimento	**1**	**2**	**3**	**4**	***baseline***
Acurácia	43,17	51,88	61,20	62,62	59,70
Clusterização	N/A	N/A	*K-means*	*k-means*	N/A
% do total de vetores para treino	100	100	100	8,38	N/A
Características	*MARSYAS*	*MFCC*	*MFCC*	*MFCC*	*Log-mel energies*

isoladamente, obtiveram resultados superiores. Entretanto, somente a capacidade descritiva dessas características, não nos garante bons resultados, pois, ao considerar que cada trecho dos áudios tem capacidades descritivas iguais, atribuímos o mesmo poder de decisão à trechos muito dispersos dos demais, que não representam a classe a qual estão rotulados de forma uniforme.

O algoritmo *K-means* agrupa trechos de áudios similares, reduzindo a influência de trechos anômalos no treinamento dos classificadores. Analogamente, ao realizar o processamento do algoritmo *K-means* novamente, dessa vez usando as tendências de todos os áudios de uma mesma classe, é possível identificar tendências mais gerais entre seus áudios. Esta característica é desejável para o problema em questão, pois as diferenças entre os áudios da classe "aeroporto" gravados na cidade X, e os áudios da classe "aeroporto" gravados na cidade Y provavelmente possuem diferenças, mas elas tendem a serem minimizadas quando comparadas com áudios da classe "parque" por exemplo.

Como o algoritmo *K-means* é não-determinístico, ao alterar a posição dos centroides no início do processamento, resultados distintos podem ser obtidos, como os exibidos nas Seções 5.3 e 5.4, nos quais os testes foram realizados mais de uma vez com os valores iniciais dos centroides de forma aleatória. Mesmo assim, os resultados ainda são robustos, como evidenciado pelo baixo valor do desvio padrão. Dessa forma podemos concluir que o fator não-determinístico do algoritmo *K-means* não impactou negativamente os resultados.

A Tabela 5.6 mostra que os resultados obtidos com $k2 = 2.000$ não são significativamente diferentes dos resultados com $k2 = 10.000$ para o classificador KNN. Portanto, ao utilizar essa técnica, foi possível generalizar a distribuição espacial de todas as classes usando pouco menos de 10% dos vetores de características de cada classe, conforme mostrado na

Tabela 5.7.

A Tabela 5.8 mostra que o método em 2 níveis (experimento 4) leva a melhores resultados que todos os outros métodos. Além disto, este resultado é obtido usando menos de 10% de todos os vetores disponíveis para o treino, enquanto o método avaliado no experimento 3 precisa de todos os vetores para atingir um resultado competitivo. O *K-means* em 2 níveis encontra tendências entre áudios diferentes da mesma classe, enquanto o *K-means* em 1 nível apenas encontra tendências em cada áudio de forma independente. Com *K-means* em 2 níveis, padrões similares em áudios diferentes da mesma classe podem ser descritos com um único centroide. Isto faz com que o alcance no espaço de características seja preservado, mesmo usando menos pontos.

Os resultados dos experimentos 3 e 4, mostrados nas Seções 5.3 e 5.4, são diretamente comparáveis com o resultado obtido pelo sistema *baseline*. Entretanto, os resultados obtidos possuem custo computacional mais baixo e não necessitam de *hardware* especial para executar.

Capítulo 6

Conclusões

Neste trabalho foram desenvolvidos e avaliados sistemas de classificação automática de cenas acústicas. Os sitemas dos experimentos 3 e 4 atingiram resultados competitivos com o sistema *baseline* disponibilizado juntamente com a base de dados *TUT Urban Acoustic Scenes 2018*. No entanto, os sistemas propostos possuem a vantagem de não necessitarem de *hardware* especializado para o treinamento dos modelos. Além disto, por usarem apenas algoritmos de aprendizagem de máquina clássicos, o custo computacional é bem menor que o custo de sistemas baseados em aprendizagem profunda.

O método proposto baseado em clusterização com *K-means* em 2 níveis alcançou o melhor resultado de todos os métodos avaliados. Além disto, para obter o melhor resultado o método necessitou de apenas 8,38% do total de vetores de características. Isto diminui o custo computacional do treinamento, bem como contribui para o poder de generalização do modelo.

Embora os resultados obtidos não sejam melhores que os apresentados pelos trabalhos relacionados, eles mostram que é possível obter resultados satisfatórios com técnicas de aprendizagem de máquina tradicionais. Além disto, também servem de motivação para exploração de outras técnicas de processamento de sinais e aprendizagem de máquina que permitam com que sistemas de classificação possam ser usados em dispositivos com menor poder computacional.

Referências

ADAVANNE, Sharath; PERTILÄ, Pasi; VIRTANEN, Tuomas. Sound event detection using spatial features and convolutional recurrent neural network. *CoRR*, abs/1706.02291, 2017.

AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. *IEEE Transactions on Computers*, C-23, n. 1, p. 90–93, Jan 1974. ISSN 0018-9340.

BAILEY, David H.; SWARZTRAUBER, Paul N. *A Fast Method for the Numerical Evaluation of Continuous Fourier and Laplace Transforms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1994. 1105–1110 p. Disponível em: <http://dx.doi.org/10.1137/0915067>.

BIJALWAN, Vishwanath; KUMAR, Vinay; KUMARI, Pinki; PASCUAL, Jordan. Knn based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, v. 7, n. 1, p. 61–70, 2014.

CARON, Mathilde; BOJANOWSKI, Piotr; JOULIN, Armand; DOUZE, Matthijs. Deep clustering for unsupervised learning of visual features. *CoRR*, abs/1807.05520, 2018. Disponível em: <http://arxiv.org/abs/1807.05520>.

CHU, S.; MATARIC, M.; KUO, C.; NARAYANAN, S. Where am i? scene recognition for mobile robots using audio features. Toronto, Canada, v. 00, p. 885–888, 07 2006. Disponível em: <doi.ieeecomputersociety.org/10.1109/ICME.2006.262661>.

CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. In: . Hingham, MA, USA: Kluwer Academic Publishers, 1995. v. 20, n. 3, p. 273–297. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1022627411411>.

DANG, An; VU, Toan; WANG, Jia-Ching. Acoustic scene classification using ensemble of convnets. September 2018.

DAS, Sumit; DEY, Aritra; PAL, Akash; ROY, Nabamita. Article: Applications of artificial intelligence in machine learning: Review and prospect. *International Journal of Computer Applications*, v. 115, n. 9, p. 31–41, April 2015.

DUDA, Richard O.; HART, Peter E.; STORK, David G. *Pattern Classification (2nd Edition)*. New York, NY, USA: Wiley-Interscience, 2000. ISBN 0471056693.

ERONEN, A. J.; PELTONEN, V. T.; TUOMI, J. T.; KLAPURI, A. P.; FAGERLUND, S.; SORSA, T.; LORHO, G.; HUOPANIEMI, J. Audio-based context recognition. *Trans. Audio, Speech and Lang. Proc.*, IEEE Press, Piscataway, NJ, USA, v. 14, n. 1, p. 321–329, dez. 2006. ISSN 1558-7916. Disponível em: <http://dx.doi.org/10.1109/TSA.2005.854103>.

FOLEISS, Juliano Henrique. Automatic genre classification by representing tracks with multiple vectors. 2018.

FOLEISS, Juliano Henrique; TAVARES, Tiago Fernandees. Texture selection for automatic music genre classification. *arXiv preprint arXiv:1905.11959*, 2019.

GHAHRAMANI, Zoubin. Unsupervised learning. Springer-Verlag, p. 72–112, 2004.

GOLUBKOV, Alexander; LAVRENTYEV, Alexander. Acoustic scene classification using convolutional neural networks and different channels representations and its fusion. September 2018.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>.

HALL, Peter; PARK, Byeong U; SAMWORTH, Richard J. Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, JSTOR, p. 2135–2152, 2008.

HAO, WenJie; ZHAO, Lasheng; ZHANG, Qiang; ZHAO, HanYu; WANG, JiaHua. DCASE 2018 task 1a: Acoustic scene classification by bi-LSTM-CNN-net multichannel fusion. September 2018.

HAYKIN, S.; HAYKIN, S.S. *Neural Networks and Learning Machines*. Prentice Hall, 2009. (Neural networks and learning machines, v. 10). ISBN 9780131471399. Disponível em: <https://books.google.com.br/books?id=K7P36lKzI_QC>.

HEITTOLA, Toni; MESAROS, Annamaria; VIRTANEN, Tuomas. TUT Urban Acoustic Scenes 2018, Development dataset. Zenodo, abr. 2018. Disponível em: <https://doi.org/10.5281/zenodo.1228142>.

HERRERA, Perfecto; SERRA, Xavier; PEETERS, Geoffroy. Audio descriptors and descriptor schemes in the context of mpeg-7. 1999.

JAIN, Anil K. Data clustering: 50 years beyond k-means. In: DAELEMANS, Walter; GOETHALS, Bart; MORIK, Katharina (Ed.). *Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 3–4. ISBN 978-3-540-87479-9.

JAIN, A. K.; DUIN, R. P. W.; MAO, Jianchang. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 1, p. 4–37, Jan 2000. ISSN 0162-8828.

KHAN, Asifullah; SOHAIL, Anabia; ZAHOORA, Umme; QURESHI, Aqsa Saeed. A survey of the recent architectures of deep convolutional neural networks. *CoRR*, abs/1901.06032, 2019. Disponível em: <http://arxiv.org/abs/1901.06032>.

KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1412.html\#KingmaB14>.

Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 1558-2256.

LOGAN, Beth. Mel frequency cepstral coefficients for music modeling. *Proc. 1st Int. Symposium Music Information Retrieval*, 11 2000.

LORENA, Ana Carolina; CARVALHO, André Carlos Ponce Leon Ferreira de. Uma introdução às support vector machines. *RITA*, v. 14, n. 2, p. 43–67, 2007. Disponível em: <http://dblp.uni-trier.de/db/journals/rita/rita14.html\#LorenaC07>.

MEHALA, Neelam; DAHIYA, Ratna. A comparative study of fft, stft and wavelet techniques for induction machine fault diagnostic analysis. In: *Proceedings of the 7th WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2008. (CIMMACS'08), p. 203–208. ISBN 978-960-474-049-9. Disponível em: <http://dl.acm.org/citation.cfm?id=1569508.1569542>.

MESAROS, Annamaria; HEITTOLA, Toni; VIRTANEN, Tuomas. A multi-device dataset for urban acoustic scene classification. Submitted to DCASE2018 Workshop. 2018. Disponível em: <https://arxiv.org/abs/1807.09840>.

MIRANDA, Eduardo. *Composing Music with Computers with Cdrom*. Newton, MA, USA: Butterworth-Heinemann, 2001. ISBN 0240515676.

MITCHELL, Thomas M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

NGUYEN, Truc; PERNKOPF, Franz. Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters. September 2018.

NUTTALL, A. Some windows with very good sidelobe behavior. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 29, n. 1, p. 84–91, February 1981. ISSN 0096-3518.

PENG, Jieluo. Understanding of the convolutional neural networks with relative learning algorithms. p. 657–661, 01 2018.

RADHAKRISHNAN, Rathnakumar; DIVAKARAN, Ajay; SMARAGDIS, A. Audio analysis for surveillance applications. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, p. 158–161, 2005.

RAFII, Zafar; PARDO, Bryan. Music/voice separation using the similarity matrix. p. 583–588, 12 2012.

REN, Zhao; PANDIT, Vedhas; QIAN, Kun; YANG, Zijiang; ZHANG, Zixing; SCHULLER, Björn. Deep sequential image features for acoustic scene classification. *Workshop on Detection and Classification of Acoustic Scenes and Events*, 11 2017.

ROMA, G.; NOGUEIRA, W.; HERRERA, P. Recurrence quantification analysis features for environmental sound recognition. p. 1–4, Oct 2013. ISSN 1931-1168.

SATHYA, R.; ABRAHAM, Annamma. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, The Science and Information Organization, v. 2, n. 2, 2013. Disponível em: <http://dx.doi.org/10.14569/IJARAI.2013.020206>.

SHINDE, Sachin V.; TIDKE, Bharat A. Improved k-means algorithm for searching research papers. 2014.

SIMURRA, Ivan. A utilização descritores de áudio à análise e composição musical assistidas por computador: um estudo de caso na obra labori ruinae. 2015. Disponível em: <https://www.anppom.com.br/congressos/index.php/25anppom/Vitoria2015/paper/view/3467>.

STEVENS, S. S.; VOLKMANN, J.; NEWMAN, E. B. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, v. 8, n. 3, p. 185–190, 1937. Disponível em: <https://doi.org/10.1121/1.1915893>.

TACHIBANA, H.; ONO, N.; KAMEOKA, H.; SAGAYAMA, S. Harmonic/percussive sound separation based on anisotropic smoothness of spectrograms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 22, n. 12, p. 2059–2073, Dec 2014. ISSN 2329-9290.

TZANETAKIS, George; COOK, Perry. Marsyas: A framework for audio analysis. *Org. Sound*, Cambridge University Press, New York, NY, USA, v. 4, n. 3, p. 169–175, dez. 1999. ISSN 1355-7718. Disponível em: <http://dx.doi.org/10.1017/S1355771800003071>.

WALDEKAR, Shefali; SAHA, Goutam. Wavelet-based audio features for acoustic scene classification. September 2018.

YADAV, Jyoti; SHARMA, Monika. A review of k - mean algorithm. *International Journal of Engineering Trends and Technology*, v. 4, 07 2013.

YANG, Jeong Hyeon; KIM, Nam Kyun; KIM, Hong Kook. Se-resnet with gan-based data augmentation applied to acoustic scene classification. September 2018.

ZHOU, Yiren; SONG, Sibo; CHEUNG, Ngai-Man. On classification of distorted images with deep convolutional neural networks. 01 2017.