Universidade Federal do Rio de Janeiro

LEONARDO AREIAS FANZERES

AMPLIAÇÃO DA CONSCIÊNCIA SITUACIONAL DO INDIVÍDUO SURDO ATRAVÉS DE SISTEMA DE RECONHECIMENTO DE SONS DO AMBIENTE

Rio de Janeiro

2014

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LEONARDO AREIAS FANZERES

AMPLIAÇÃO DA CONSCIÊNCIA SITUACIONAL DO INDIVÍDUO SURDO ATRAVÉS DE SISTEMA DE RECONHECIMENTO DE SONS DO AMBIENTE

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Orientadora: Profª. Adriana Santarosa Vivacqua, D.Sc.

Co-Orientador: Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Rio de Janeiro

2014

F 218 Fanzeres, Leonardo Areias

Ampliação da consciência situacional do indivíduo surdo através de sistema de reconhecimento de sons do ambiente / Leonardo Areias Fanzeres. – 2014.

97 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti, Programa de Pós-Graduação em Informática, Rio de Janeiro, 2014.

Orientadora: Adriana Santarosa Vivacqua

Co-Orientador: Luiz Wagner Pereira Biscainho

1. Computação Móvel. 2. Tecnologias Assistivas para Surdos. 3. Consciência Situacional. 4. Reconhecimento de Sons do Ambiente - Teses. I. Vivacqua, Adriana Santarosa (Orient.). II. Biscainho, Luiz Wagner Pereira (Co-Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti, Programa de Pós-Graduação em Informática. IV. Título.

CDD

LEONARDO AREIAS FANZERES

AMPLIAÇÃO DA CONSCIÊNCIA SITUACIONAL DO INDIVÍDUO SURDO ATRAVÉS DE SISTEMA DE RECONHECIMENTO DE SONS DO AMBIENTE

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Aprovada em ____/____/_____ .

_____

Profª. Adriana Santarosa Vivacqua, D.Sc., IM, UFRJ

_____

Prof. Luiz Wagner Pereira Biscainho, D.Sc., DEL/Poli e PEE/COPPE, UFRJ

_____

Eng. Denise Del Re Filippo, D.Sc., ESDI, UERJ

_____

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D., DEL/Poli e PEE/COPPE, UFRJ

_____

Profª. Jonice de Oliveira Sampaio, D.Sc., IM, UFRJ

Resumo

FANZERES, Leonardo. **Ampliação da consciência situacional do indivíduo surdo através de sistema de reconhecimento de sons do ambiente**. 2014. 97 f. Dissertação (Mestrado em Informática) – PPGI, Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

A condição de isolamento acústico do indivíduo pode diminuir muito a sua consciência situacional, importante para o convívio social, o desempenho de funções e a sua própria segurança. Esta pesquisa aborda o problema a partir das dificuldades enfrentadas por indivíduos surdos, realizando um estudo exploratório no domínio da computação móvel assistiva. São especificados requisitos e apresentadas soluções a problemas encontrados no desenvolvimento de um sistema de reconhecimento de sons do ambiente que visa à ampliação da consciência situacional de indivíduos surdos. O aplicativo proposto executa todo o processamento no próprio dispositivo móvel, desde a captura do sinal sonoro à visualização do espectrograma, extração de *features* de áudio e classificação. Durante a pesquisa foi conduzido um experimento com *smartphones*, sendo produzida uma base de conhecimento com instâncias correspondentes a 300 registros de áudio distribuídos em 30 classes. Com estes dados foram realizados testes de classificação utilizando os algoritmos *nearest neighbor*, *naive Bayes*, *Bayes network* e um *ensemble* de florestas de decisão aleatórias. Os resultados foram bastante satisfatórios comparados aos obtidos em outros experimentos similares realizados em plataformas *desktop*. Para minimizar os problemas observados na execução do aplicativo em ambiente não monitorado foi elaborada uma equação com o objetivo de indicar o nível de confiança da classificação, acrescentando uma informação de apoio ao processo de reconhecimento do som. A personalização da base de conhecimento foi outra solução proposta pelo estudo, visando a atender à demanda por reconhecimento de som de todos os potenciais usuários do sistema. Ademais, foi realizado um teste alfa do aplicativo com um grupo de usuários surdos, proporcionando um retorno inspirador e relatando uma avaliação positiva do sistema proposto. Publicado sob o nome VSom, o aplicativo está disponível gratuitamente *online*.

Palavras-chave: reconhecimento de sons do ambiente, consciência situacional, consciência de contexto sonoro, tecnologias assistivas para surdos, computação móvel, computação ubíqua, classificação do som, extração de *features* de áudio.

Abstract

FANZERES, Leonardo. **Ampliação da consciência situacional do indivíduo surdo através de sistema de reconhecimento de sons do ambiente**. 2014. 97 f. Dissertação (Mestrado em Informática) – PPGI, Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

Acoustic insulation can greatly reduce the situational awareness of an individual, important to the social life, the performance of tasks and her/his own security. This research addresses the problem from the difficulties faced by deaf individuals, conducting an exploratory study in the field of assistive mobile computing. The study specifies requirements and presents solutions to problems that emerged during the development of an environmental sound recognition system which aims to expand the situational awareness of deaf individuals. The proposed application performs all processing in the mobile device itself, from signal capturing to spectrogram display, audio features extraction and classification. During the research, an experiment with smartphones was conducted, and it was produced a knowledge base with instances corresponding to 300 audio records distributed in 30 classes. With these data, classification tests were performed using the algorithms nearest neighbor, naive Bayes, Bayes network and an *ensemble* of random decision forests. The results were quite satisfactory compared to those obtained in other similar experiments on desktop platforms. To minimize problems observed in the execution of the application in unmonitored environments, an equation was elaborated with the purpose to indicate the confidence level of classification, adding support information to the sound recognition process. The personalization of the knowledge base was another solution proposed by the study, in order to meet the demand for sound recognition of all potential users of the system. In addition, an alpha test was conducted with a group of deaf users, providing an inspiring feedback and reporting a positive evaluation of the proposed system. Published under the name VSom, the application is available online for free.

Keywords: environmental sound recognition, situational awareness, sound context awareness, assistive technologies for the deaf, mobile computing, ubiquitous computing, sound classification, audio features extraction.

Agradecimentos

Agradeço sinceramente aos meus orientadores Profª. Adriana Santarosa Vivacqua e Prof. Luiz Wagner Pereira Biscainho pelo suporte e aconselhamento durante o estudo realizado. Gostaria de agradecer também aos colegas, funcionários e professores do PPGI, especialmente à Profª. Jonice de Oliveira Sampaio, à Profª. Carla Amor Divino Moreira Delgado e ao Prof. Marcello Goulart Teixeira, que tanto contribuíram para o meu crescimento durante o curso.

Quero expressar minha especial gratidão, por toda a colaboração e informação proporcionadas, aos integrantes do Projeto Surdos-UFRJ, Profª. Vivian Rumjanek, Prof. Flavio Eduardo P. da Silva, professor e intérprete Tiago Batista e aos participantes do teste do aplicativo VSom realizado no Laboratório Didático de Ciências para Surdos (LaDiCS).

Agradeço também à CAPES e ao CNPq pelo financiamento da pesquisa, permitindo, além da realização deste estudo, a publicação e apresentação de um artigo em maio de 2014.

Finalmente, gostaria de agradecer à minha família pelo apoio e amor dedicados, sem os quais esta pesquisa não teria sido possível.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos Utilizados

API – Application Programming Interface

ARFF – Attribute-Relation File Format

ASR – Automatic Speech Recognition

BC – Base de Conhecimento

BPMN – Business Process Modeling Notation

ESR – Environmental Sound Recognition

FFT – Fast Fourier Transform

GPI – Group Pertaining Index

GZIP – GNU Zip

ICA – Independent Component Analysis

INES – Instituto Nacional de Educação de Surdos

LaDiCS – Laboratório Didático de Ciências para Surdos

LIBRAS – Língua Brasileira de Sinais

LPC – Linear Predictive Coefficients

MATLAB – Matrix Laboratory

MFCC – Mel-Frequency Cepstral Coefficients

MP – Matching Pursuit

NN – Nearest Neighbor

RMS – Root Mean Square

RNA – Redes Neurais Artificiais

Sumário

1 Introdução

A consciência de um indivíduo sobre o que ocorre em um ambiente depende muito da sua capacidade de perceber manifestações sonoras e identificar os eventos relacionados a elas. A condição de isolamento acústico pode, portanto, comprometer significativamente a consciência situacional do indivíduo, importante para o convívio social, o desempenho de funções e a sua própria segurança. Em muitas situações cotidianas, um indivíduo ouvinte não se dá conta de quanto depende da sua audição para perceber o que está ocorrendo ao seu redor. A consciência do contexto sonoro é necessária em um número muito maior de situações do que frequentemente se imagina. Nesta pesquisa este problema é abordado a partir das dificuldades de adquirir consciência situacional enfrentadas por indivíduos surdos, realizando um estudo exploratório no domínio da computação móvel assistiva, especificando requisitos e apresentando soluções aos problemas encontrados no desenvolvimento de um sistema de reconhecimento de sons do ambiente (ESR, acrônimo de *environmental sound recognition*). No estudo realizado por Matthews et al. [1], no qual foi testado um sistema de ESR com a participação de usuários surdos, são fornecidos exemplos dessa problemática. Entre outros casos mencionados, uma participante relatou que uma vez havia esquecido o aspirador de pó ligado durante toda a noite, pois o aparelho não apresentava nenhum sinal visual de que estava em funcionamento. Para diminuir dificuldades decorrentes de problemas como esse, propõe-se a utilização de um sistema ubíquo de reconhecimento de sons do ambiente, apontando uma alternativa para a ampliação da consciência situacional desses indivíduos.

Devido à mobilidade e à capacidade de processamento, *smartphones* são os primeiros dispositivos a fornecer ubiquidade computacional real [2]. Inúmeros benefícios são oferecidos através de dispositivos móveis, o que levou esta tecnologia a ser considerada um serviço universal [3]. No entanto, uma desvantagem de alguns aplicativos móveis é a dependência de processamento remoto ou de recursos fornecidos por infraestrutura local. Como solução a este problema, propõe-se um aplicativo que executa todo o processamento no próprio dispositivo móvel, desde a captura do sinal de áudio ao reconhecimento do som.

São tratados na presente pesquisa temas relacionados diretamente ao desenvolvimento do sistema como especificação de requisitos, interação humano-computador e questões de

implementação, além de detalhes sobre o processo de reconhecimento do som. Sobre este último tema, os estudos existentes abordam predominantemente o reconhecimento automático da fala (ASR, acrônimo de *automatic speech recognition*), sons do ambiente e música. No caso de sons do ambiente, costuma-se utilizar um modelo pré-definido de categorias, geralmente aplicado à indexação/recuperação de documentos de áudio/vídeo e à sistemas de vigilância. No entanto, o reconhecimento de sons não estruturados, no qual não é possível definir previamente as classes de som tratadas, ainda é um tema pouco explorado [4] [5], especialmente quando se trata de processamento baseado em tecnologia móvel. Neste documento é apresentado um experimento [6] sobre o desempenho da classificação do som utilizando *smartphones*, constituído por três fases: registro das amostras, preparação da base de conhecimento (BC) e testes de classificação. A BC completa possui as instâncias correspondentes a 300 registros de áudio distribuídos em 30 classes. A partir destes dados são realizados testes de classificação com os algoritmos *nearest neighbor* (NN) [7], *naive Bayes* [7], *Bayes network* [7] e um *ensemble* de florestas de decisão aleatórias [7]. Os resultados são organizados de forma a permitir visualizar tanto o desempenho detalhado dos classificadores utilizando a BC mais completa, quanto o comportamento desses classificadores de acordo com o aumento da BC. Além de buscar referências para a configuração do sistema, procura-se, através do experimento, demonstrar que aplicativos móveis podem atingir acurácia similar à obtida em sistemas baseados em plataforma *desktop*.

Neste estudo é abordada também a questão da representação da incerteza na classificação. O reconhecimento do som realizado na prática com o aplicativo sendo executado em um ambiente não monitorado, no qual não há controle sobre a ocorrência de eventos sonoros, pode apresentar resultados bastante incoerentes. Uma possível alternativa para minimizar este problema é informar ao usuário sobre o nível de confiança da classificação. Para tanto, propõe-se um indicador de pertencimento da nova instância ao grupo (GPI, acrônimo de *group pertaining index*) formado pelas instâncias da classe prevista que pode ser calculado com baixo custo computacional.

A característica do sistema de executar todo o processamento no próprio dispositivo impõe certas limitações que podem comprometer a abrangência dos sons a serem tratados. Por mais eficiente que seja o algoritmo utilizado, a sua execução vai implicar um custo

computacional que se eleva quando há aumento da BC. Diante das limitações de recursos e da ampla variedade dos sons do ambiente [8], este estudo propõe a personalização da BC como alternativa para atender à demanda por reconhecimento de som de todos os potenciais usuários.

Esta pesquisa inclui também a publicação do aplicativo de registro e reconhecimento de sons do ambiente denominado VSom[1], disponível gratuitamente *online*. Ademais, procurou-se realizar uma avaliação do sistema através de um teste inicial com usuários surdos, proporcionando um retorno inspirador e demostrando uma percepção positiva do aplicativo pelos participantes.

Este documento está organizado da seguinte forma. O próximo capítulo reúne a argumentação necessária para as definições da pesquisa. No Capítulo 3 são apresentadas e comparadas abordagens anteriores com propostas semelhantes à do presente estudo. No Capítulo 4 são definidos os tópicos necessários para a elaboração do experimento, descrito no capítulo seguinte. No Capítulo 6 é apresentado o cálculo do GPI, proposto nesta pesquisa para oferecer apoio ao resultado da classificação do som. No Capítulo 7 é justificada a opção pela personalização da BC. No Capítulo 8 é apresentado o aplicativo móvel de ESR. No Capítulo 9 é descrito o teste da versão alfa do aplicativo, realizado com um grupo de potenciais usuários. No capítulo seguinte são feitas as considerações finais da pesquisa.

[1] Vsom está disponível para download em https://play.google.com/store/apps/details?id=app.vsom

2 Definições da Pesquisa

2.1 Problemática

A condição de isolamento acústico do indivíduo surdo pode, em diversas situações, comprometer seriamente a sua consciência situacional, importante para o convívio social, o desempenho de funções e a sua própria segurança. No estudo realizado por Matthews et al. [1], foi testado um sistema de reconhecimento do som com a participação de usuários surdos que proporcionaram relatos demonstrando os problemas que enfrentam devido à diminuição da consciência situacional. Em um estudo similar conduzido por Ho-ching et al. [9], foram entrevistados dez participantes surdos e um consultor sobre tecnologia assistiva para identificar as necessidades de obtenção de consciência situacional dos surdos, dividindo-as em três categorias: consciência da presença de outros, interação com aparelhos baseados no som e saindo do ambiente doméstico. Além da problemática apresentada nos estudos mencionados, membros do Instituto Nacional de Educação de Surdos (INES) confirmaram a carência de consciência situacional enfrentada por esses indivíduos.

2.2 Proposição

A proposição da pesquisa foi formulada a partir da problemática identificada e é constituída por questões norteadoras como mostrado abaixo.

Q1 A computação móvel pode ser utilizada para o reconhecimento de sons do ambiente em tempo real?

Q2 Como desenvolver um sistema ubíquo de reconhecimento de sons do ambiente para ampliar a consciência situacional de indivíduos surdos?

2.3 Objetivo

Esta pesquisa tem como objetivo descobrir um caminho viável para o desenvolvimento de um sistema ubíquo de reconhecimento de sons do ambiente que permita a ampliação da consciência situacional de indivíduos surdos. Para tal, o estudo especifica requisitos e apresenta soluções às dificuldades encontradas. Ademais, a pesquisa visa a fornecer um estudo qualitativo com potenciais usuários do sistema proposto.

2.4 Delimitação do estudo

Nesta pesquisa são considerados sons do ambiente quaisquer eventos sonoros detectados pelo sistema de acordo com a amplitude e entropia do sinal de áudio. Mais detalhes sobre a detecção automática de eventos sonoros são apresentados na Seção 8.7. Sons da fala e de música podem ser incluídos, porém considerando-se somente o reconhecimento das classes correspondentes. No caso da fala, por exemplo, os sons podem ser classificados como voz grave ou voz aguda, podendo indicar também se se fala baixo ou se grita. No entanto, não estão no escopo deste estudo funcionalidades de ASR ou de identificação de música. A pesquisa apresentada pode, porém, auxiliar estudos que procurem abordar tais funcionalidades em tecnologia móvel através das informações proporcionadas pelo experimento realizado e pelas soluções propostas. O estudo não aborda tampouco problemas relacionados ao reconhecimento em ambientes com sons sobrepostos.

2.5 Abordagem de solução

2.5.1 Metodologia

A pesquisa consistiu em uma abordagem exploratória no domínio da computação móvel assistiva. Como levantamento de informações, além da pesquisa bibliográfica, foram consultados profissionais da educação de surdos do INES, que confirmaram a carência de consciência situacional enfrentada pelos surdos. Ademais, a participação no simpósio Caminhos da Inclusão[2] realizado em novembro de 2012 pelos integrantes do Projeto Surdos-

[2] http://projetosurdos.bioqmed.ufrj.br/?p=467

UFRJ[3] foi fundamental para a compreensão, ao menos parcial, da experiência do indivíduo surdo na sociedade. O objetivo desse levantamento foi identificar as principais necessidades dos usuários quanto à consciência situacional, proporcionando assim uma melhor orientação ao desenvolvimento do sistema.

> *Q1. A computação móvel pode ser utilizada para o reconhecimento de sons do ambiente em tempo real?*

Para responder à primeira questão norteadora da pesquisa (Q1) foi conduzido um experimento com um protótipo do sistema utilizando uma BC com 30 classes de sons. O experimento foi realizado de forma a proporcionar também parâmetros para a configuração do aplicativo a fim de se obter o melhor *trade-off* entre acurácia da classificação e responsividade da interface, considerando os requisitos do sistema especificados no presente estudo (Seção 4.1).

> *Q2. Como desenvolver um sistema ubíquo de reconhecimento de sons do ambiente para ampliar a consciência situacional de indivíduos surdos?*

Para responder à segunda questão norteadora da pesquisa (Q2) foram apresentadas soluções quanto à representação da incerteza e à construção da BC, além das especificações e da publicação do aplicativo proposto. Foi conduzido também um estudo qualitativo através de um teste alfa do aplicativo com usuários surdos. O teste proporcionou informações importantes para as próximas versões do sistema e, potencialmente, para outros estudos sobre sistemas de ESR projetados para usuários surdos. O detalhamento das funcionalidades do aplicativo, apresentado nesta pesquisa, e a sua disponibilização para *download* gratuito são também parte da resposta à Q2, contribuindo para futuras abordagens sobre o tema.

2.5.2 Opção pela computação móvel

Como descrito em 2.5.1, foi identificada a necessidade de se ampliar a consciência situacional do indivíduo surdo. A partir dessa problemática, a pesquisa explora a possibilidade de se utilizar computação móvel para o reconhecimento de sons do ambiente em tempo real considerando características como: ubiquidade do sistema, capacidade de processamento compatível com a tarefa e adotabilidade da tecnologia utilizada.

[3] http://projetosurdos.bioqmed.ufrj.br/

Nos estudos realizados por Ho-ching et al. [9] e Matthews et al. [1] foram desenvolvidos sistemas de ESR baseados em plataforma *desktop*. Em ambos os estudos foram conduzidos testes com usuários surdos, que expressaram as suas prioridades quanto aos sons que gostariam de monitorar, relatando a necessidade do uso do sistema em vários ambientes em casa, no trabalho e em público. Portanto, uma solução projetada para atender a tais situações deve proporcionar robustez e ubiquidade. Estas observações foram consideradas no projeto do sistema para que o seu uso não interfira, ou interfira minimamente, nas dinâmicas em que o usuário poderá estar inserido nos ambientes mencionados. A necessidade de reconhecer o som em vários ambientes também foi confirmada por profissionais da educação de surdos do INES, expressando a importância de adotar uma solução utilizando tecnologia que favoreça a ubiquidade do serviço. Essas informações, e também o levantamento de sons e ambientes com prioridade de detecção documentado por Hersh et al [10], orientaram a escolha de tecnologias móveis para a condução da presente pesquisa. Ademais, no estudo realizado por Azar et al. [11], para o qual foi desenvolvido um sistema de ESR para surdos, os participantes do teste demonstraram grande interesse na possibilidade de o sistema, baseado em plataforma *desktop*, ser desenvolvido para dispositivos móveis. Propôs-se portanto o uso da computação móvel para o desenvolvimento de um sistema, com funções de reconhecimento do som, que apresente visualmente ao usuário a ocorrência de eventos sonoros ao seu redor. Este recurso deverá auxiliá-lo no reconhecimento dos sons emitidos, e das suas respectivas fontes emissoras, através da classificação baseada em sons previamente rotulados.

No entanto, a opção pela computação móvel pode acarretar desvantagens, como a dependência de alguns aplicativos em processamento remoto, geralmente fornecido através de computação em nuvem [12]. Além disso, os aplicativos imersivos que visam a oferecer informação em tempo real são mais sensíveis a falhas ao utilizarem serviços disponibilizados remotamente. Outra dependência observada está relacionada com a necessidade do uso de recursos fornecidos pelos chamados "espaços inteligentes" ou, mais especificamente, "casas inteligentes", "escritórios inteligentes" e "cidades inteligentes". A indisponibilidade desta infraestrutura pode desabilitar o serviço oferecido se o aplicativo não tiver o recurso embarcado no dispositivo. Cáceres e Friday [13] apontaram a importância de se reduzir a

necessidade de infraestrutura local em aplicativos ubíquos. Dadas estas circunstâncias, este estudo propõe um sistema de ESR que não é dependente de infraestrutura e executa todo o processamento no dispositivo móvel, desde a captura do sinal de áudio à classificação e apresentação do resultado do reconhecimento do som ao usuário.

A escolha do sistema operacional Android[4] para o desenvolvimento do aplicativo se deve aos fatos de haver grande variedade de dispositivos compatíveis com a plataforma e de ser notável a sua adoção, alcançando aproximadamente 87% do mercado brasileiro[5] e 79% do mercado global[6] de *smartphones*.

2.6 Relevância

Segundo os dados do Censo divulgados pelo IBGE, em 2010 existiam no Brasil 2,1 milhões de pessoas com deficiência auditiva grave, das quais 344,2 mil eram surdas[7]. Apesar do notável desenvolvimento da educação de surdos, o suporte tecnológico para a assistência a esses indivíduos pode estar sendo subutilizado pela sociedade. Existem atualmente mais de três milhões de aplicativos ativos nas lojas dos cinco maiores distribuidores [8] [9] [10] [11] [12]. A oferta é extremamente variada, porém não há sequer uma categoria para aplicativos assistivos. O entusiasmo com os serviços oferecidos pela computação móvel pode estar eclipsando questões mais prementes como a assistência a indivíduos com necessidades especiais, que, no caso dos surdos, são centenas de milhares de pessoas no país. A pesquisa sobre o desenvolvimento de um sistema que proporcione a ampliação da consciência situacional do indivíduo surdo através do uso de tecnologia móvel pode ser uma importante contribuição para a equiparação funcional desse expressivo grupo da sociedade.

Nos testes realizados por Azar et al. [11] e Ho-ching et al. [9], é possível perceber a satisfação dos participantes com a proposta dos sistemas utilizados nos respectivos

[4] Android é marca registrada da Google Inc.

[5] http://www.statista.com/statistics/245189/market-share-of-mobile-operating-systems-for-smartphone-sales-in-brazil/

[6] IDC Worldwide Mobile Phone Tracker, August 7, 2013, http://www.idc.com/getdoc.jsp?containerId=prUS24257413

[7] ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/Caracteristicas_Gerais_Religiao_Deficiencia/caracteristicas_religiao_deficiencia.pdf

[8] http://www.appbrain.com/stats/number-of-android-apps

[9] http://techcrunch.com/2014/06/02/itunes-app-store-now-has-1-2-million-apps-has-seen-75-billion-downloads-to-date/

[10] http://www.cnet.com/news/windows-phone-store-hits-more-than-300000-apps/

[11] http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=RssLanding&cat=news&id=1940045

[12] http://www.ibtimes.com/blackberry-announces-amazon-appstore-partnership-triples-available-apps-instant-1604936

experimentos, que apresentam algumas características em comum com o sistema proposto nesta pesquisa, como descrito no Capítulo 3. Além dos sistemas correlatos apresentados no capítulo mencionado, existe uma gama mais ampla de tecnologias assistivas para surdos e indivíduos com dificuldades de audição extensamente documentada por Hersh et al [10], que vão desde dispositivos vestíveis, adaptados ao ouvido e implantados, a diversas categorias de sistemas de alerta e alarme com sinais vibro-táteis e visuais, além de sistemas de comunicação com *design* específico. Apesar de proporcionarem grande assistência a indivíduos surdos ou com dificuldade auditiva, muitas vezes essas tecnologias são caras e de difícil acesso, tornando complexa a sua adoção. Portanto, além de apontar uma alternativa mais acessível utilizando computação móvel e disponível gratuitamente, a presente pesquisa oferece informações relevantes a futuros projetos e estudos sobre sistemas de ESR para usuários surdos.

No estudo realizado por Matthews et al. [1] foram descritas situações nas quais a consciência do contexto sonoro é necessária. Um participante comentou que quando há algo errado com seu carro, o problema tende a passar despercebido até um momento em que se torna caro demais consertá-lo. Em outro caso mencionado, um participante contou que uma vez ele e sua esposa queimaram acidentalmente comida, o que acionou o alarme de incêndio. Como ambos eram surdos, eles não notaram que o alarme estava tocando até que um amigo ouvinte os visitou. Estes relatos são um exemplo da variedade de ambientes com os quais é necessário lidar, bem como a urgência do serviço em certos casos. Neste mesmo estudo, os participantes relataram também a necessidade de receber alguma informação sobre o nível de confiança da classificação. Essa questão foi abordada na presente pesquisa através da elaboração de uma equação que propõe calcular a incerteza do resultado da classificação, apresentada no Capítulo 6.

Quanto ao experimento descrito no Capítulo 5, este pode contribuir como referência a outros estudos sobre a classificação do som em dispositivos móveis, sobretudo em abordagens genéricas de ESR. Além dos resultados do experimento e de sua respectiva análise, a pesquisa fornece também as BCs utilizadas, disponíveis *online*[13].

[13] http://purl.org/vsom/experiment_1/kbases

O reconhecimento de sons não estruturados ainda é um tema pouco explorado [4] [5], especialmente quando se trata de processamento baseado em tecnologia móvel. Este estudo aborda os problemas de abrangência e dinamicidade de conhecimento do sistema com a proposta de personalização da BC, apresentada no Capítulo 7.

3 Estudos Correlatos

A seguir são apresentados experimentos sobre classificação do som similares ao conduzido neste estudo (Capítulo 5), além de projetos e pesquisas prévias sobre sistemas que proporcionam a ampliação da consciência situacional do indivíduo surdo, preferencialmente através de dispositivos móveis.

3.1 Experimentos

Em 1993, realizando experimentos em ESR, Goldhor [14] verificou o comportamento da classificação de sons do ambiente com relação ao número de atributos da *feature* extraída. O estudo apresenta, entre outros experimentos, a variação da acurácia de acordo com o aumento do número de instâncias para cada classe de som. Neste experimento somente foram testados dois estados da BC, com 5 instâncias por classe e com classes completas, que continham entre 8 e 19 instâncias, fornecendo, portanto, pouca informação sobre a tendência do comportamento da classificação. No entanto, verifica-se que a acurácia é maior quando são utilizadas as classes completas.

Temko e Nadeu [15] realizaram um experimento de classificação de sons do ambiente, obtendo no teste com melhor desempenho uma média de acurácia de 88% utilizando *support vector machines* baseadas em árvores de decisão binárias. No entanto, o estudo não é aplicado a dispositivos móveis.

Em 2013, Mogi e Kasai [4] conduziram um experimento sobre ESR utilizando registros realizados com *smartphones*. O estudo dá ênfase à avaliação das *features* extraídas do áudio, proporcionando uma comparação do desempenho da classificação quando se aplicam diferentes combinações entre *Mel-Frequency Cepstral Coefficients* (MFCC), *Matching Pursuit* (MP) [16] e *Independent Component Analysis* (ICA). O classificador utiliza o algoritmo NN. O experimento demonstrou uma acurácia maior com a combinação de MP com ICA, sendo este obtido a partir do vetor de MFCC. No entanto, o cálculo de ICA implica alto custo computacional, sendo pouco indicado para situações com limitação

de recursos. Diante desse problema, os autores propõem o uso de processamento baseado em nuvem para viabilizar a execução da solução proposta.

3.2 Sistemas

Matthews et al. [1] desenvolveram o protótipo de um sistema de ESR para usuários surdos treinado para um ambiente específico, neste caso, um escritório. Além de reconhecer os sons ocorrentes no ambiente, o sistema indica a direção da fonte emissora. Sua pesquisa oferece importante contribuição com detalhes sobre as impressões e preferências dos participantes sobre o sistema, sendo especialmente interessantes as suas demandas quanto à:

- confiabilidade: necessidade de indicação da acurácia do reconhecimento do som apresentado;
- usabilidade: possibilidade de interpretar o reconhecimento do som rapidamente (*glanceability*) e de dispor de um histórico dos sons reconhecidos.

Essas informações tiveram significativa influência na especificação dos requisitos do sistema proposto na presente pesquisa. Ademais, os participantes do estudo indicaram os eventos sonoros que gostariam de reconhecer ou de ser alertados. O sistema, no entanto, é baseado em plataforma *desktop* e também apresenta limitações quanto à responsividade devido ao atraso da apresentação do reconhecimento sonoro em relação ao momento da ocorrência do som capturado.

Ho-ching et al. [9] desenvolveram um protótipo que proporciona duas formas de visualização do som: um espectrograma e um mapa do ambiente que indica a localização das fontes emissoras do som, sendo esta implementada em modo de simulação. No experimento, o usuário deve, a partir da visualização que lhe é fornecida, indicar se o som que ocorre é de alguém batendo à porta ou de telefone tocando. Esta tarefa é realizada simultaneamente a uma tarefa principal. Os resultados indicam a capacidade de inferência dos usuários ao utilizarem as formas de visualização mencionadas. Apesar do menor percentual de acerto na visualização com uso do espectrograma, os usuários demonstraram grande interesse nesta funcionalidade, conseguindo reconhecer outros eventos sonoros, além dos empregados no experimento. No

entanto, esta forma de ampliação da consciência situacional demanda muita atenção do usuário, podendo não ser viável em diversos locais e situações. O processo de localização das fontes emissoras demonstrou ser bastante eficiente no auxílio do reconhecimento do som, porém implica um mapeamento prévio do ambiente.

Azar et al. [11] desenvolveram o protótipo de um sistema de ESR para usuários surdos que indica a direção da fonte emissora do som, além de oferecer funcionalidades básicas de ASR. No entanto, como nos estudos mencionados acima, o projeto é totalmente baseado em plataforma *desktop*. O teste do sistema contou com a participação de dez usuários surdos, que se mostraram entusiasmados com os resultados experimentados e expressaram grande interesse na possibilidade de o sistema ser desenvolvido para dispositivos móveis.

O aplicativo AudioVision desenvolvido por Hipke et al. [17] proporciona funcionalidades semelhantes às do aplicativo aqui proposto. A principal diferença entre as duas abordagens está na classificação do áudio, que no caso do aplicativo citado é realizada em um servidor remoto. No presente estudo, propõe-se que este processo seja realizado no dispositivo detector com independência da disponibilidade de outros equipamentos ou informações adicionais no momento da classificação.

Lu et al. [18] apresentaram em 2009 um *framework* para o desenvolvimento de aplicativos de reconhecimento do som, no qual a classificação é feita em estágios consecutivos, hierarquicamente. O reconhecimento é portanto refinado em cada estágio. No experimento apresentado pelos autores, o primeiro estágio decide a qual das três classes genéricas pertence o registro: fala, música ou som ambiente, utilizando árvore de decisão e modelos de Markov. No segundo estágio, somente foi implementado um classificador para os sons da fala, que procura distinguir o registro de áudio entre fala feminina e masculina. Para eventos classificados como sons ambientes, adotou-se aprendizado não supervisionado. Neste caso o sistema agrupa os registros por similaridade e solicita ao usuário que forneça um nome sempre que uma nova categoria é identificada. No entanto, na classe genérica de sons do ambiente, a única inferência efetuada sobre o som capturado é quanto à sua relevância, que é baseada na duração e na frequência com que o som ocorre. Eles também apresentaram uma avaliação da acurácia dos classificadores utilizando registros de sons do ambiente já nomeados fornecidos por

estudo anterior [19]. O teste incluiu as seguintes classes: "Andando", "Dirigindo carros", "Andando de elevadores" e "Andando de ônibus", que alcançaram 93%, 100%, 78% e 25% de acurácia, respectivamente. O estudo, no entanto, não é direcionado a usuários surdos, mas ao desenvolvimento de aplicativos de rastreamento de atividades do usuário. Ou seja, mais que informar quais sons estão ocorrendo no ambiente, a proposta é revelar padrões de atividade do usuário reconhecidos no sinal de áudio capturado pelo dispositivo. O estudo oferece também informações detalhadas sobre processamento de áudio com baixo custo computacional. Um ponto importante neste estudo é que todo o processo é realizado em dispositivo móvel, uma característica comum ao sistema proposto na presente pesquisa. Quanto ao reconhecimento de sons do ambiente, o estudo propõe um sistema escalável ao grande público através da implementação de um algoritmo de aprendizado não supervisionado adaptativo, porém não é abordado o problema do aumento da base de cada usuário. O estudo não apresenta informações sobre a capacidade do sistema de realizar a classificação do som em tempo real com bases de dados maiores, especialmente com um maior número de classes, mantendo a acurácia verificada.

Rossi et al. [20] apresentaram em seu estudo um aplicativo de ESR em tempo real. O sistema utilizou uma base pré-definida com seis registros de áudio de 30 segundos de duração para cada classe. Apesar do baixo percentual de acerto da classificação, de aproximadamente 58%, o sistema oferece uma importante contribuição com a proposta de dois modos de execução: um autônomo, no qual todo o processo de reconhecimento é realizado no próprio dispositivo; e um modo baseado em servidor, no qual as *features* extraídas do áudio são enviadas ao servidor que executará o processo de classificação. Os autores também apresentaram uma comparação entre o desempenho dos dois modos de operação propostos utilizando dois modelos distintos de *smartphones*. Após o teste, a constatação foi de que o modo servidor não demonstrou benefícios que justificassem a sobrecarga de comunicação. No entanto, o uso dos recursos de processamento de um servidor poderiam oferecer benefícios significativos quanto ao tempo de execução e consumo de energia se fossem utilizados algoritmos mais complexos nas tarefas de classificação.

Em 2012, o engenheiro Will Powell[14] apresentou o protótipo de um sistema de realidade aumentada capaz de reconhecer a fala e traduzir o texto extraído em poucos instantes[15]. Ele utilizou uma placa Raspberry Pi[16], óculos Vuzix Star 1200, um microfone e, para a tradução, a API da Microsoft[17]. O objetivo do protótipo é permitir a conversa entre duas pessoas se expressando em idiomas distintos. No entanto, à época, o aplicativo ainda apresentava um pequeno atraso no ritmo da conversa devido ao tempo requerido para o processamento da informação. Apesar de o usuário surdo não ter sido o foco deste projeto, esta funcionalidade pode lhe ser útil com ou sem tradução, simplesmente pelo fato de apresentar a transcrição da fala em um dispositivo móvel dentro do seu campo visual e sem ocupar as suas mãos.

No projeto Lumisonic [21] foi desenvolvido um aplicativo para dispositivos móveis que propõe a visualização do som a partir de formas abstratas que procuram permitir ao usuário surdo a percepção do ritmo e da intensidade dos sons que ocorrem no ambiente. O aplicativo não tem, no entanto, a preocupação de proporcionar a equiparação funcional do indivíduo.

Na Tabela 1 é apresentada uma comparação entre aplicativos, protótipos e sistemas correlatos, incluindo o sistema proposto nesta pesquisa. Dos sistemas mencionados na comparação, além do VSom, cinco apresentam aplicativo para dispositivos móveis (Tópico 1), dos quais somente Lumisonic e VSom estão disponíveis *online* (Tópico 12). Ademais, dentre os sistemas baseados em computação móvel, apenas Lumisonic, AudioVision e VSom são direcionados a usuários surdos (Tópico 4). Todos os sistemas tiveram seus protótipos ou versões alfa testados (Tópico 11). Quanto à preocupação com a disponibilidade do serviço, somente Lumisonic, SoundSense, AmbientSense e VSom apresentam propostas eficazes, com todo o processamento sendo executado no próprio dispositivo (Tópico 2). Similarmente ao VSom, outros dois sistemas propõem uma abordagem genérica de ESR (Tópico 3), não sendo, portanto, baseados em um modelo pré-definido de classes de sons do ambiente. Dentre estes, apenas o AudioVision foi

[14] http://www.willpowell.co.uk/

[15] http://www.youtube.com/watch?v=vw6dJDMmnlw

[16] http://www.raspberrypi.org/archives/344

[17] Microsoft é marca registrada da Microsoft Corporation.

desenvolvido especialmente para usuários surdos. Além do VSom, quatro sistemas apresentam um espectrograma ou outro recurso de visualização do som em tempo real (Tópico 5), porém dentre estes apenas o Lumisonic é desenvolvido para dispositivos móveis. Quatro sistemas, além do VSom, disponibilizam alguma função que alerta o usuário sobre sons ocorrentes com sinais visuais e/ou vibro-táteis (Tópico 6). Essa característica é fundamental para que a ampliação da consciência situacional seja efetiva sem exigir que o usuário esteja consultando constantemente a tela do aplicativo. Através de diferentes recursos, mais da metade dos sistemas auxilia o usuário a identificar as fontes emissoras de som (Tópico 7). No entanto, somente dois sistemas, ambos baseados em plataforma *desktop*, oferecem algum recurso que indica a direção da fonte emissora (Tópico 8), dos quais o protótipo em Python implementa uma simulação da funcionalidade de localização explícita da fonte (Tópico 9). Entre os sistemas apresentados, somente os óculos tradutores de Will Powell e o protótipo em MATLAB oferecem funcionalidades de ASR (Tópico 10).

Tabela 1 - Comparação entre aplicativos, protótipos e sistemas correlatos

	TÓPICO	Lumisonic [21]	Óculos tradutores de Will Powell	Protótipo em MATLAB [11]	Protótipo em Python [9]	IC2Hear [1]	Audio Vision [17]	Sound Sense [18][18]	Ambient Sense [20]	VSom (aplicativo proposto)
1	Apresenta aplicativo para dispositivos móveis	●	●[19]				●	●	●	●
2	Executa todo o processamento no próprio dispositivo móvel	●						●	●[20]	●
3	Propõe abordagem genérica de ESR						●	●		●
4	Direcionado especificamente a usuários surdos	●		●	●	●	●			●
5	Apresenta espectrograma ou outro recurso de visualização do som em tempo real	●		●	●	●				●
6	Disponibiliza função de alertar o usuário					●	●	●	●	●
7	Auxilia o usuário a identificar as fontes emissoras de som			●	●	●	●		●	●
8	Indica a direção da fonte emissora			●	●					
9	Indica explicitamente a localização das fontes emissoras de som				●[21]					
10	Oferece funcionalidade de ASR		●	●[22]						
11	Protótipo ou versão alfa testado/a	●	●	●	●	●	●	●	●	●
12	Sistema disponibilizado *online*	●								●

[18] SoundSense é um *framework* para o desenvolvimento de aplicativos móveis e implementa funcionalidades genéricas, não se encaixando portanto na maioria dos itens enumerados nesta tabela, que se referem a características mais específicas dos sistemas.

[19] O protótipo de Will Powell não foi desenvolvido com as tecnologias móveis padronizadas por grandes fabricantes como Google, Apple ou Microsoft, mas está montado de forma a permitir bastante mobilidade.

[20] AmbientSense disponibiliza função de reconhecimento do som com processamento totalmente executado no dispositivo, além de uma opção com execução em servidor.

[21] O protótipo em Python implementa uma simulação da funcionalidade de localização das fontes emissoras de som.

[22] O protótipo em MATLAB oferece funcionalidade de ASR utilizando um vocabulário expansível, treinado inicialmente com 12 palavras.

4 Definições Iniciais do Sistema

Além de procurar responder à primeira questão norteadora da pesquisa (Q1), o motivo da realização do experimento consistiu em fornecer resultados esclarecedores ao desenvolvimento do sistema. Portanto, para a elaboração do experimento foi desenvolvido um protótipo, sendo necessário definir previamente requisitos e questões de implementação do sistema.

4.1 Requisitos

Com base no levantamento de dados realizado na pesquisa bibliográfica, na entrevista com profissionais da educação de surdos do INES e no conhecimento obtido durante a participação do simpósio Caminhos da Inclusão, foram especificados os requisitos do sistema, apresentados a seguir.

R1 Disponibilidade do serviço

O sistema deve realizar todo o processo de registro e classificação de áudio com os recursos do próprio dispositivo detector, sem depender de processamento ou informação remota, seja por rede ou por qualquer outro dispositivo acoplado.

R2 Ubiquidade

O sistema deve poder ser utilizado em casa, no trabalho e em locais públicos como o interior de um ônibus, um restaurante ou uma sala de aula.

R3 Adotabilidade

O sistema deve ser compatível com tecnologia de custo acessível e de ampla adoção.

R4 Abrangência de conhecimento

O sistema deve ser capaz de reconhecer sons ambientes de qualquer tipo, não havendo, portanto, possibilidade de assumir como premissa qualquer estruturação prévia dos eventos a serem reconhecidos.

R5 Dinamicidade de conhecimento

O sistema deve utilizar uma BC que suporte acréscimo e exclusão de classes de som dinamicamente, possibilitando assim a sua atualização.

R6 Responsividade

O tempo do reconhecimento do evento sonoro, incluindo a apresentação do resultado, deve ser o mínimo possível, não ultrapassando 5 segundos, verificado em estudos anteriores [22] como o tempo limite de espera tolerado por usuários de dispositivos móveis.

R7 Confiabilidade

Com consciência das dificuldades técnicas que implica o reconhecimento de eventos sonoros não estruturados, é importante que, além de obter alta acurácia na classificação, o sistema apresente o resultado do reconhecimento do som acompanhado da informação referente ao seu nível de confiança.

R8 Informatividade

A principal função informativa do sistema é o reconhecimento do som, porém o sistema deve oferecer também recursos de visualização em tempo real do som que estiver sendo capturado ou reproduzido a níveis mais baixos de descrição. Ou seja, o sistema deve oferecer um retorno do som sem a interpretação da classificação, podendo indicar, por exemplo, somente a intensidade e a frequência do sinal sonoro. Esta descrição de baixo nível não deve exigir nenhum aprendizado prévio ou capacidade de cognição específica por parte do usuário. Esta informação é fundamental para o usuário saber com precisão o momento em que os sons ocorrem e em que estão sendo registrados, reconhecidos ou reproduzidos.

R9 Usabilidade

A interação com o usuário deve ocorrer de forma que demande o mínimo de atenção para interpretar a informação apresentada e procurar deixar as suas mãos livres. Ademais, o sistema deve disponibilizar uma função de alerta visual e/ou vibro-tátil ao usuário quanto à

ocorrência de sons relevantes no ambiente. O sistema deve disponibilizar também o histórico dos sons reconhecidos, de forma que o usuário tenha a alternativa de verificar os sons que foram reconhecidos nos momentos em que não esteve olhando para os resultados apresentados pelo sistema. A preocupação principal deste requisito é permitir que o usuário utilize o sistema sem comprometer as demais tarefas que estiver realizando. O sistema deve também priorizar o uso de imagens na interface gráfica, procurando contornar a dificuldade de indivíduos surdos com a interpretação da linguagem escrita [23].

4.2 Implementação

Devido à adotabilidade requerida (R3), o aplicativo foi desenvolvido para ser executado em sistema operacional Android a partir da versão 2.3.3. A implementação do aplicativo utiliza a linguagem de programação Java.

Como especificado no requisito de disponibilidade do serviço (R1), todo o ciclo do processamento é executado no próprio dispositivo móvel, o que inclui desde a captura do sinal à visualização do espectrograma, extração de *features* de áudio e classificação.

Foram utilizadas as seguintes bibliotecas de código aberto:

- jAudio [24] – implementada em Java, a jAudio possui as classes responsáveis pela extração de *features*.
- Weka [25] – biblioteca de mineração de dados com diversos métodos estatísticos implementados. Desde a versão 3.0, lançada em 1999, o Weka é disponibilizado inteiramente em Java, executável em dispositivos *desktop* com a máquina virtual Java instalada. No experimento aqui apresentado foi utilizada uma versão adaptada por RJ Marsan[23] à máquina virtual Dalvik, utilizada nos dispositivos com sistema operacional Android.

[23] https://github.com/rjmarsan/Weka-for-Android

5 Experimento

Utilizando um protótipo do aplicativo proposto neste estudo, foi realizado um experimento [6], cujas etapas são descritas a seguir. Além de verificar o desempenho dos algoritmos de classificação utilizando as BCs geradas em dispositivos móveis, o experimento procurou analisar o comportamento da classificação com o aumento da base e fornecer informações que permitissem estabelecer parâmetros para o desenvolvimento de um sistema compatível com uma BC dinâmica. Ademais, foram conduzidos testes em dispositivo móvel para verificar a duração do treinamento dos classificadores e do reconhecimento do som.

5.1 Considerações sobre os requisitos do sistema

A seguir são feitas considerações sobre alguns dos requisitos mencionados na Seção 4.1 que mais influenciaram na elaboração do experimento.

> *R1. O sistema deve realizar todo o processo de registro e classificação de áudio com os recursos do próprio dispositivo detector, sem depender de processamento ou informação remota, seja por rede ou por qualquer outro dispositivo acoplado.*
>
> *R6. O tempo do reconhecimento do evento sonoro, incluindo a apresentação do resultado, deve ser o mínimo possível, não ultrapassando 5 segundos, verificado em estudos anteriores [22] como o tempo limite de espera tolerado por usuários de dispositivos móveis.*

Estes requisitos implicaram a seleção de algoritmos de classificação com menor custo computacional, compatíveis com a capacidade de processamento de dispositivos móveis.

> *R4. O sistema deve ser capaz de reconhecer sons ambientes de qualquer tipo, não havendo, portanto, possibilidade de assumir como premissa qualquer estruturação prévia dos eventos a serem reconhecidos.*

Esta exigência quanto à abrangência dos sons a serem reconhecidos limitou as alternativas de *features* de áudio, pois dificulta ou inviabiliza o uso de dicionários pré-definidos como no caso de MP.

> *R5. O sistema deve utilizar uma BC que suporte acréscimo e exclusão de classes de som dinamicamente, possibilitando assim a sua atualização.*

Este requisito implica que o experimento forneça referências que auxiliem na definição dos parâmetros para o desenvolvimento de um sistema compatível com uma BC dinâmica, e apresente informações que orientem a busca de uma alternativa adequada para a estruturação dos dados de acordo com esta abordagem.

5.2 Registro das amostras

Diante do desafio enfrentado de classificar sons não estruturados utilizando tecnologia móvel, optou-se pela máxima qualidade encontrada na maioria dos dispositivos para definir a configuração da captura do sinal de áudio, conforme indicado na Tabela 2.

Tabela 2 - Dados de configuração da captura do sinal

Tamanho da sequência	0,4 a 2,7 s
Taxa de amostragem	48000 Hz
Profundidade de bits	16
Configuração de canal	mono

Como mencionado anteriormente, o experimento não aborda os problemas relacionados à classificação de registros com sons sobrepostos. A amostragem foi realizada em ambientes diversos, sendo a maior parte dos registros realizada em ambiente residencial com relativo controle sobre a qualidade da captura, mas nenhum em estúdio de gravação. Dos 300 registros, 80 foram realizados em ambientes como parque, restaurante, estacionamento, via pública ou praia.

Ao efetuar um registro, o usuário o classifica definindo o nome do som no atributo "classe". O valor deste atributo será previsto no processo de reconhecimento do som. Os valores resultantes das *features* são a média e o desvio padrão dos valores extraídos de cada uma das janelas do sinal original, que têm a duração de aproximadamente 23ms. Usar janelas maiores reduz o custo total do processamento do sinal, porém procura-se manter o tamanho das janelas em torno de 20ms para não comprometer demasiadamente as características temporais do sinal, efetuando um *trade-off* entre perda de informação e custo computacional. Uma função de Hann (*hanning*) [26] é aplicada para suavizar as extremidades das janelas, minimizando assim, na transformada rápida de Fourier (FFT, acrônimo de *fast* Fourier *transform*), o efeito conhecido como vazamento espectral [27]. Não é aplicada sobreposição das janelas, pois a informação contida nas *features* não é prejudicada e o sinal não precisa ser ressintetizado. As classes de som utilizadas no experimento são apresentadas na Tabela 3. Cabe frisar que não se trata de um modelo pré-definido pelo sistema, pois o aplicativo proposto permite que as classes sejam adicionadas de acordo com a necessidade do usuário.

Tabela 3 - Lista de classes de som definidas

CLASSE DE SOM	DESCRIÇÂO
Alerta garagem	Alerta de abertura/fechamento de porta de garagem.
Ar escapando	Ar sob pressão escapando continuamente.
Aspirador de pó	Aparelho em funcionamento.
Assovio	Pessoa assoviando.
Batendo à porta	Pessoa batendo à porta.
Batendo palma	Pessoa(s) batendo palmas.
Canto de pássaro	Pássaro(s) cantando.
Carros passando	Carros passando continuamente sobre rua pavimentada.
Chafariz	Água jorrando de chafariz continuamente.
Chaves	Molho de chaves sendo sacudido.
Chuveiro	Água saindo de chuveiro continuamente.
Espirro	Pessoa espirrando.
Fechadura	Porta sendo trancada/destrancada com chave.
Guarda de trânsito apitando	Apitos intermitentes.
Máq. lavar roupa	Máquina enxaguando em movimento oscilatório.
Máq. lavar roupa centrifugando	Máquina centrifugando continuamente.
Mexendo embalagem plástica	Embalagem plástica sendo mexida/amassada.
Ondas do mar	Ondas oceânicas quebrando na praia.
Panela de pressão	Ar sob pressão escapando intermitentemente da panela.
Pedido de silêncio (sshhh)	Chiado feito com a boca.
Pessoas falando	Grupo de pessoas falando.
Ronco	Pessoa roncando.
Secador de cabelo	Aparelho em funcionamento.
Sinal de micro-ondas	Sinal emitido quando aparelho termina programa.
Toque de telefone	Toque de telefone eletrônico.
Torneira	Água saindo de torneira continuamente.
Tosse	Pessoa tossindo.
Veículo pesado passando	Caminhões/ônibus passando sobre rua pavimentada.
Voz aguda	Pessoa com voz aguda falando.
Voz grave	Pessoa com voz grave falando.

5.3 Base de conhecimento

A fim de comparar e classificar os dados de forma eficiente, a descrição do áudio é feita através de *feature vectors*. O processo de obtenção destes vetores é denominado extração de *features* de áudio [26]. Trata-se de uma descrição de baixo nível, baseada na análise automática do sinal de áudio. Esta descrição permite a classificação de registros de som, que será aplicada no processo de reconhecimento. Além da classe do som e do ambiente onde este foi registrado, a BC é constituída pelos valores resultantes da extração de *features* do áudio. É uma tarefa complexa descobrir quais são as *features* mais indicadas para a classificação de sons do ambiente. O uso de MFCC é amplamente adotado como descritor de áudio em sistemas de ASR, pois trata-se de uma descrição inspirada na percepção humana, com ênfase nas frequências mais baixas do sinal [26]. Na pesquisa em ESR ainda não há uma abordagem amadurecida para a extração de *features*, porém MFCC e *Linear Predictive Coefficients* (LPC) demostraram bom desempenho durante a classificação de sons do ambiente [28] [29]. Anteriormente, Goldhor [14] pôde verificar em seus experimentos que a acurácia em ESR utilizando MFCC se estabilizou em 98% com vetores de dimensão entre 12 e 16.

Chu et al. [30] verificaram que o uso de mais *features* não necessariamente melhora o desempenho da classificação. Com base nesta constatação, buscou-se na presente pesquisa um conjunto reduzido de descritores que complementasse a composição da BC, fornecendo as informações necessárias para proporcionar uma acurácia satisfatória da classificação sem aumentar significativamente o custo computacional do processo. Durante a seleção de *features*, a estratégia adotada foi pressupor que existe algum conjunto ideal de descritores para sistemas de ESR. Porém, uma vez que o conjunto de *features* esteja definido, algumas classes são mais nitidamente destacadas do que outras, provocando heterogeneidade na acurácia da classificação. Ou seja, algumas classes produzem *clusters* bem definidos, enquanto outras apresentam padrões mais dispersos. Além de MFCC e LPC, as demais *features* utilizadas no experimento foram selecionadas após testes preliminares aplicados na última base, composta por 30 classes com 10 instâncias cada. Uma abordagem genérica de ESR implica uma variedade de sons muito superior à atingida na base testada. Porém, a utilização de uma BC completa, pré-definida e contendo todos os sons da categoria não é uma solução praticável devido à natureza diversificada e em constante transformação dos sons do ambiente, constituindo um elevado número de classes.

Ainda que se estabelecesse um número fixo de registros por classe de som e se reduzisse a precisão da medição na extração de *features*, que são variáveis contínuas, a consulta a esta informação seria inviável devido ao grande volume de informação, especialmente se se considera a limitação de recursos no processamento executado exclusivamente em dispositivo móvel (R1) e o tempo requerido para a apresentação do resultado do reconhecimento sonoro (R6). A questão da completude da BC é tratada com mais detalhes no Capítulo 7. Como requerido em R5, o sistema proposto nesta pequisa utiliza uma BC que deverá poder ser atualizada constantemente com o acréscimo ou exclusão de classes de som e de seus respectivos registros, portanto tornou-se necessário avaliar também o comportamento dos classificadores de acordo com a variação do tamanho da base. A composição final utilizada é apresentada na Tabela 4. No caso dos atributos numéricos, a quantidade representa a dimensão do vetor resultante da extração da *feature* correspondente. Os valores finais das *features* serão a média e o desvio padrão dos valores extraídos das janelas do sinal.

Tabela 4 - Composição da base de conhecimento

ATRIBUTO			QUANTIDADE	TIPO
Classes rotuladas		Som (class) [24]	1	nominal
		Ambiente (environment) [25]	1	nominal
Features de Áudio	Desvio padrão	*Spectral Rolloff Point* [24]	1	numérico
		Spectral Flux [24]	1	numérico
		Standard Deviation of Spectral Flux	1	numérico
		Compactness [24]	1	numérico
		Spectral Variability [24]	1	numérico
		MFCC [26]	13	numérico
		LPC [26]	9	numérico
	Média	*Spectral Rolloff Point*	1	numérico
		Spectral Flux	1	numérico
		Standard Deviation of Spectral Flux	1	numérico
		Compactness	1	numérico
		Spectral Variability	1	numérico
		MFCC	13	numérico
		LPC	9	numérico

[24] Ao se efetuar um registro, lhe é atribuído um som (“class” na implementação). Este atributo será previsto no processo de reconhecimento do som.

[25] O atributo “ambiente” (“environment” na implementação) não foi considerado no experimento.

5.4 Testes de classificação

Os testes de desempenho dos classificadores foram realizados através de validação cruzada com 10 iterações usando uma BC com instâncias correspondentes a 300 registros de áudio distribuídos em 30 classes, além dos testes incluindo bases menores, realizados para avaliar o comportamento da classificação com a variação do tamanho da base. O experimento utiliza as *features* apresentadas na Tabela 4, com exceção do atributo "ambiente" ("environment" na implementação) que foi desconsiderado nos testes de classificação. As BCs utilizadas estão disponíveis *online*[26]. Para facilitar a condução do experimento, os testes foram realizados em dispositivo *desktop*, porém utilizando somente as BCs construídas com o protótipo do aplicativo e executando as mesmas classes neste implementadas. Os testes foram realizados em janeiro de 2014 utilizando um *notebook* Dell Vostro[27] 3500 com processador Intel Core[28] i3 CPU M 370 @ 2.40GHz × 4 com 4GB de RAM, executado no sistema operacional Ubuntu[29] 13.10 32-bit.

Com a preocupação de empregar classificadores de diferentes tipos, foram utilizados métodos baseados em busca exaustiva, probabilidade e árvores de decisão. Para o teste com busca exaustiva foi utilizado o algoritmo NN. Neste caso a classe prevista será a classe à qual pertence a instância mais próxima da instância testada. Quanto aos métodos probabilísticos, foram utilizados os classificadores *naive Bayes* e *Bayes network*. Nestes testes, os atributos das instâncias, originalmente numéricos, são convertidos em atributos nominais. Quanto ao teste baseado em árvores de decisão, foi utilizado um *ensemble* de florestas de decisão aleatórias. Árvores de decisão apresentam excelente desempenho quanto ao tempo da classificação, porém a acurácia é frequentemente prejudicada. O classificador utilizado procura compensar a menor acurácia deste método aplicando uma série de iterações com florestas de decisão.

A seguir são apresentadas as matrizes de confusão resultantes dos testes de classificação com a base completa. As linhas das matrizes apresentadas nas Tabelas 5, 6, 7 e 8 representam as classes testadas e as colunas representam as classes previstas. Os detalhes de configuração dos classificadores utilizados estão disponíveis no Apêndice A.

[26] http://purl.org/vsom/experiment_1/kbases

[27] Dell e Vostro são marcas registradas da Dell Computer Corporation.

[28] Intel e Intel Core são marcas registradas da Intel Corporation.

[29] Ubuntu é marca registrada da Canonical.

5.4.1 Nearest neighbor

No teste com NN não é executado treinamento. A classe prevista corresponde à classe da instância mais próxima da instância avaliada, utilizando a distância euclidiana como referência.

Instâncias classificadas corretamente: 278

Percentual de acerto: 93 %

Tabela 5 - Matriz de confusão resultante do teste com *nearest neighbor*

	Alerta de garagem	Ar escapando	Aspirador de pó	Assovio	Batendo à porta	Batendo palma	Canto de pássaro	Carros passando	Chafariz	Chaves	Chuveiro	Espirro	Fechadura	Guarda de trânsito apitando	Máq. lavar roupa	Máq. lav. roupa centrifugando	Mexendo embalagem plástica	Ondas do mar	Panela de pressão	Pedido de silêncio (sshhh)	Pessoas falando	Ronco	Secador de cabelo	Sinal de micro-ondas	Toque de telefone	Torneira	Tosse	Veículo pesado passando	Voz aguda	Voz grave
Alerta de garagem	8													1				1												
Ar escapando		10																												
Aspirador de pó			10																											
Assovio				10																										
Batendo à porta					8											1												1		
Batendo palma						10																								
Canto de pássaro							9								1															
Carros passando								10																						
Chafariz									10																					
Chaves										10																				
Chuveiro											10																			
Espirro												10																		
Fechadura													10																	
Guarda de trânsito apitando														10																
Máq. lavar roupa															10															
Máq. lav. roupa centrifugando																10														
Mexendo embalagem plástica																	10													
Ondas do mar																		10												
Panela de pressão																			10											
Pedido de silêncio (sshhh)											1									9										
Pessoas falando																					10									
Ronco													1									9								
Secador de cabelo																							10							
Sinal de micro-ondas																								10						
Toque de telefone																									10					
Torneira									2		5						1									2				
Tosse																	1										9			
Veículo pesado passando								4								1												5		
Voz aguda																													9	1
Voz grave																														10

5.4.2 Naive Bayes

Neste teste, os atributos numéricos são convertidos em nominais após discretização supervisionada [7].

Instâncias classificadas corretamente: 266

Percentual de acerto: 89 %

Tabela 6 - Matriz de confusão resultante do teste com *naive Bayes*

	Alerta de garagem	Ar escapando	Aspirador de pó	Assovio	Batendo à porta	Batendo palma	Canto de pássaro	Carros passando	Chafariz	Chaves	Chuveiro	Espirro	Fechadura	Guarda de trânsito apitando	Máq. lavar roupa	Máq. lav. roupa centrifugando	Mexendo embalagem plástica	Ondas do mar	Panela de pressão	Pedido de silêncio (sshhh)	Pessoas falando	Ronco	Secador de cabelo	Sinal de micro-ondas	Toque de telefone	Torneira	Tosse	Veículo pesado passando	Voz aguda	Voz grave
Alerta de garagem	9																									1				
Ar escapando		10																												
Aspirador de pó			10																											
Assovio				7																							2			1
Batendo à porta					9																								1	
Batendo palma						9							1																	
Canto de pássaro	1						6																						1	2
Carros passando								10																						
Chafariz									10																					
Chaves										10																				
Chuveiro											9															1				
Espirro												6															3			1
Fechadura													10																	
Guarda de trânsito apitando	1													9																
Máq. lavar roupa															10															
Máq. lav. roupa centrifugando																10														
Mexendo embalagem plástica																	8									1		1		
Ondas do mar																		8										2		
Panela de pressão																			10											
Pedido de silêncio (sshhh)																				9								1		
Pessoas falando								1													8								1	
Ronco																						10								
Secador de cabelo																							10							
Sinal de micro-ondas																								10						
Toque de telefone																									10					
Torneira									1		1						1						1			5				1
Tosse												1															8	1		
Veículo pesado passando								1																				9		
Voz aguda																				1									8	1
Voz grave																				1										9

5.4.3 Bayes network

Neste teste, as redes neurais artificiais (RNA) foram configuradas com no máximo dois pais para cada nó. Os atributos numéricos são convertidos em nominais após discretização supervisionada [7].

Instâncias classificadas corretamente: 269

Percentual de acerto: 90 %

Tabela 7 - Matriz de confusão resultante do teste com *Bayes network*

	Alerta de garagem	Ar escapando	Aspirador de pó	Assovio	Batendo à porta	Batendo palma	Canto de pássaro	Carros passando	Chafariz	Chaves	Chuveiro	Espirro	Fechadura	Guarda de trânsito apitando	Máq. lavar roupa	Máq. lav. roupa centrifugando	Mexendo embalagem plástica	Ondas do mar	Panela de pressão	Pedido de silêncio (sshhh)	Pessoas falando	Ronco	Secador de cabelo	Sinal de micro-ondas	Toque de telefone	Torneira	Tosse	Veículo pesado passando	Voz aguda	Voz grave
Alerta de garagem	9																									1				
Ar escapando		10																												
Aspirador de pó			10																											
Assovio				7																					1		1			1
Batendo à porta					8			1																					1	
Batendo palma						9							1																	
Canto de pássaro	1						7																							2
Carros passando								10																						
Chafariz									10																					
Chaves										10																				
Chuveiro											10																			
Espirro												6														3			1
Fechadura													10																	
Guarda de trânsito apitando	1													9																
Máq. lavar roupa															10															
Máq. lav. roupa centrifugando																10														
Mexendo embalagem plástica																	10													
Ondas do mar																		8										2		
Panela de pressão																			10											
Pedido de silêncio (sshhh)					1															9										
Pessoas falando								1													7								2	
Ronco																						10								
Secador de cabelo																							10							
Sinal de micro-ondas																								10						
Toque de telefone																									10					
Torneira																	1						1			7				1
Tosse												1															8	1		
Veículo pesado passando								1										1										8		
Voz aguda																				1									8	1
Voz grave																				1										9

5.4.4 Ensemble de florestas de decisão aleatórias

Neste teste são efetuadas dez iterações com florestas de decisão compostas por 5 árvores cada. Para cada árvore são selecionadas aleatoriamente 6 *features*. Não é efetuada poda e a profundidade das árvores é ilimitada.

Instâncias classificadas corretamente: 277

Percentual de acerto: 92 %

Tabela 8 - Matriz de confusão resultante do teste com *ensemble* de florestas de decisão aleatórias

	Alerta de garagem	Ar escapando	Aspirador de pó	Assovio	Batendo à porta	Batendo palma	Canto de pássaro	Carros passando	Chafariz	Chaves	Chuveiro	Espirro	Fechadura	Guarda de trânsito apitando	Máq. lavar roupa	Máq. lav. roupa centrifugando	Mexendo embalagem plástica	Ondas do mar	Panela de pressão	Pedido de silêncio (sshhh)	Pessoas falando	Ronco	Secador de cabelo	Sinal de micro-ondas	Toque de telefone	Torneira	Tosse	Veículo pesado passando	Voz aguda	Voz grave
Alerta de garagem	8						1		1																					
Ar escapando		10																												
Aspirador de pó			10																											
Assovio				10																										
Batendo à porta					9																								1	
Batendo palma						9							1																	
Canto de pássaro							7								1														1	1
Carros passando								9													1									
Chafariz									10																					
Chaves										10																				
Chuveiro											10																			
Espirro												8														1			1
Fechadura													10																	
Guarda de trânsito apitando								1						9																
Máq. lavar roupa															10															
Máq. lav. roupa centrifugando																10														
Mexendo embalagem plástica																	10													
Ondas do mar																		10												
Panela de pressão																			10											
Pedido de silêncio (sshhh)																	1			9										
Pessoas falando								1													9									
Ronco																						10								
Secador de cabelo																							10							
Sinal de micro-ondas																								10						
Toque de telefone																									10					
Torneira											1											1	1			7				
Tosse												1	1														8			
Veículo pesado passando								2																				8		
Voz aguda					1																								8	1
Voz grave																													1	9

5.4.5 Comportamento da classificação com aumento da base

Nos testes de classificação com aumento do número de instâncias por classe (Figura 1), a inclusão das instâncias foi feita aleatoriamente. Já nos testes de classificação com aumento do número de classes (Figura 2), estas foram adicionadas à base, de 3 em 3, na seguinte ordem:

1º Canto de pássaro, Ondas do mar, Torneira;

2º Secador de cabelo, Carros passando, Chaves;

3º Voz grave, Veículo pesado passando, Alerta de garagem;

4º Assovio, Aspirador de pó. Batendo à porta;

5º Espirro, Tosse, Voz aguda;

6º Toque de telefone, Pessoas falando, Mexendo embalagem plástica;

7º Chuveiro, Pedido de silêncio (sshhhh), Batendo palma;

8º Guarda de trânsito apitando, Chafariz, Ronco;

9º Sinal de micro-ondas, Fechadura, Máq. lavar roupa;

10º Máq. lav. roupa centrifugando, Panela de pressão, Ar escapando.

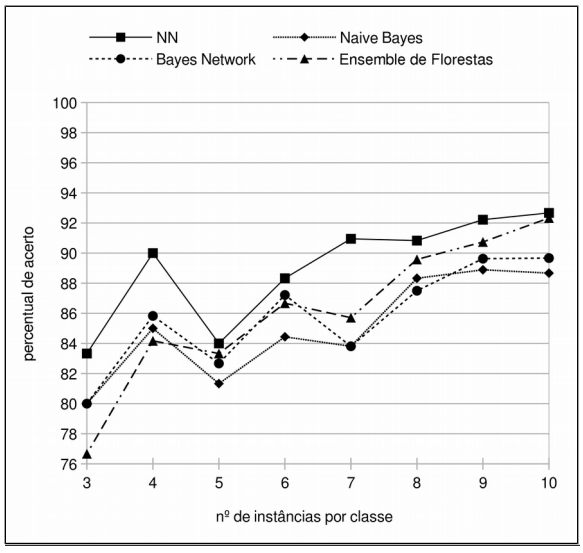


Figura 1 - Comportamento da classificação com o aumento do número de instâncias por classe. Em todos os pontos do gráfico a base possui 30 classes.

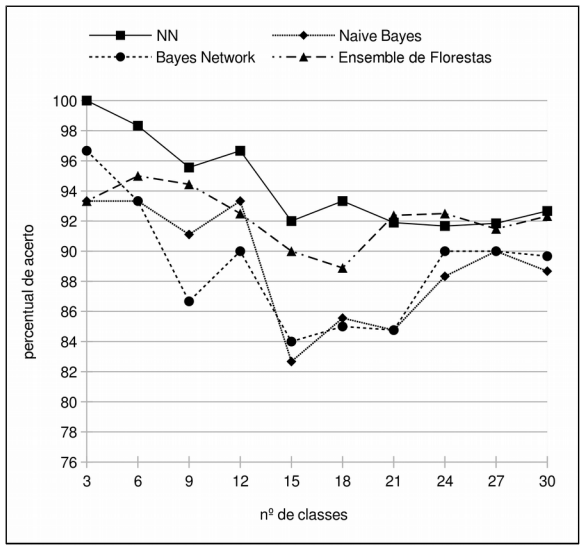


Figura 2 - Comportamento da classificação com o aumento do número de classes. Em todos os pontos do gráfico as classes possuem 10 instâncias cada.

5.5 Duração do reconhecimento do som

Para avaliar a responsividade do aplicativo (R6), foram realizados testes em dispositivo móvel verificando a duração do treinamento dos classificadores e do reconhecimento do som (Figura 3) utilizando a base completa, que contém as instâncias correspondentes a 300 registros de áudio distribuídos em 30 classes. O reconhecimento do som inclui a extração de *features* de áudio e a classificação, que são tarefas iniciadas assincronamente e posteriormente sincronizadas, portanto, a medição do tempo de execução foi efetuada avaliando o processo como um todo. Ou seja, o tempo é medido desde o início da extração de *features* até o final da classificação. O dispositivo utilizado possui referência de modelo Xperia[30] C1604 e processador de 1 GHz com sistema operacional Android 4.1.1. O tempo referente ao treino do classificador só interfere no uso do aplicativo quando a BC é modificada, pois o classificador precisa ser novamente treinado.

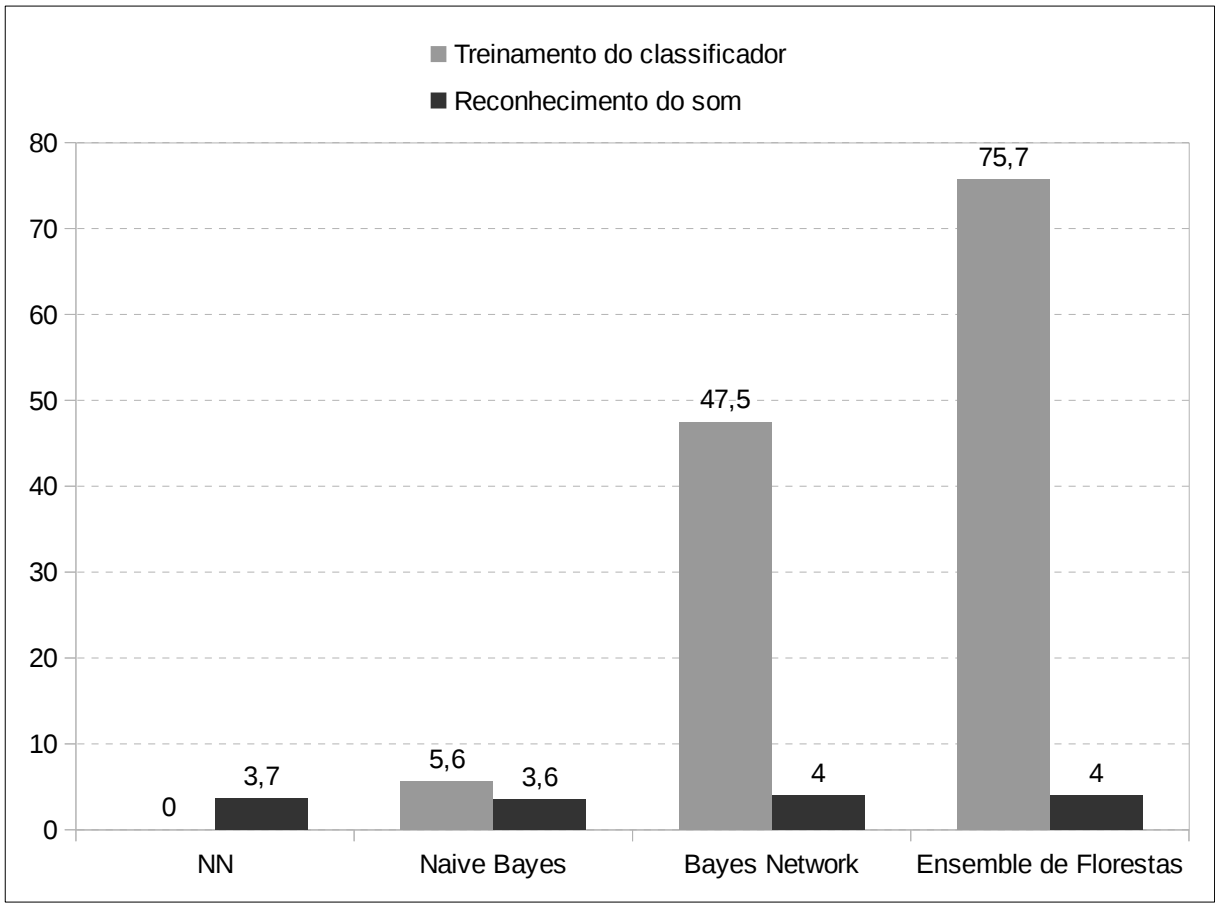


Figura 3 - Duração em segundos do treinamento do classificador e do reconhecimento do som.

[30] Xperia é marca registrada da Sony Mobile Communications AB.

5.6 Análise dos resultados

Neste experimento foi feita a análise das matrizes de confusão e dos dados referentes ao percentual de acerto da classificação e ao tempo do reconhecimento do som. Os *outputs* completos dos testes com algumas métricas adicionais estão disponíveis no Apêndice A.

A partir de uma visão geral do experimento, os níveis de acurácia alcançados foram bastante satisfatórios comparando-se a estudos anteriores similares realizados em plataforma *desktop* [15] [31] [32]. Esses resultados demonstram que a limitação de recursos imposta pelo uso exclusivo da computação móvel não inviabilizou a solução adotada.

Utilizando a base completa com 300 instâncias, o uso de NN demonstrou ser a opção com melhor desempenho, alcançando uma acurácia de 92,7%. Porém, neste classificador, a busca na BC é feita de forma exaustiva, o que provoca a elevação do tempo da classificação proporcionalmente ao aumento do número de registros. Apesar da alta acurácia alcançada, outra desvantagem deste algoritmo é não permitir que a parte mais custosa do processamento, que é o cálculo das distâncias, seja executada no treinamento, pois este cálculo só pode ser realizado após o conhecimento da nova instância. No entanto, é uma alternativa interessante para bases não muito maiores do que a utilizada, pois, além da alta acurácia alcançada, o tempo do reconhecimento do som foi similar ao dos outros métodos.

A independência entre *features* assumida pelo classificador *naive Bayes* não comprometeu o desempenho da classificação, inferior em apenas 1 ponto percentual ao resultado obtido com *Bayes network*. Portanto, o uso de RNA não resultou em uma vantagem notável, especialmente pelo fato de o treinamento ter requerido um tempo equivalente a aproximadamente 8 vezes o da opção sem RNA, como demonstrado na Figura 3.

A classificação com *naive Bayes*, apesar de ter obtido a menor acurácia, atingindo 88,7%, resultou ser uma boa opção para o aplicativo, pois além de parte do processamento poder ser adiantada no treinamento, este teve uma duração bem inferior aos testes com RNA e árvores de decisão. Esta diferença proporciona, portanto, margem para a utilização de bases significativamente maiores.

As matrizes de confusão proporcionam uma visão mais detalhada do desempenho dos classificadores. Neste experimento destacam-se sobretudo as classes com baixo percentual de

acerto. É possível, por exemplo, verificar que a classe “Torneira”, que obteve o pior desempenho geral, apresentou acurácia de 20% com NN, 50% com *naive Bayes* e 70% com *Bayes network* e árvores de decisão. Propositalmente, foram incluídas na base classes com sons similares à mencionada, como “Chafariz” e “Chuveiro”, que receberam a maior parte das instâncias mal classificadas com o uso de NN. Já nos demais classificadores, as instâncias mal classificadas foram reconhecidas, na sua maioria, como sons ainda mais distintos. Portanto, é possível perceber que, para estes sons, a classificação com NN gerou erros mais previsíveis segundo o padrão da audição humana do que com os demais algoritmos. Também é possível verificar que a classe “Voz grave” absorveu o maior número de classificações erradas, o que pode significar que a classe esteja pouco consistente, ou seja, com amostras demasiadamente distintas entre si. No entanto, a segurança dessas afirmações depende da realização de um experimento utilizando uma base com um número maior de instâncias por classe.

Quanto aos testes com aumento da BC, o *ensemble* de florestas de decisão teve comportamento mais estável que os demais métodos, como demonstrado nos gráficos das Figuras 1 e 2. Porém o tempo requerido para seu treinamento foi o mais elevado, tornando seu uso pouco indicado para o aplicativo, mesmo com a alta acurácia registrada, classificando corretamente 92,3% das instâncias. Analisando-se a partir de uma visão geral dos testes com aumento da BC, da Figura 1 é possível perceber que o desempenho dos classificadores demonstrou tendência a se manter em um valor fixo quando a base possuía pelo menos 8 instâncias por classe. Pela Figura 2, foi verificado que a acurácia começou a se estabilizar quando a base alcançou o número de 24 classes. Estes dados proporcionaram uma referência importante para a construção da BC e a configuração do aplicativo móvel.

6 Representação da Incerteza

Nos testes com validação cruzada apresentados no Capítulo 5, as instâncias utilizadas possuem *features* geradas a partir de um universo conhecido de sons. O aprendizado dos classificadores é supervisionado, com registros efetuados e rotulados pelo usuário. Os testes avaliam a consistência das classes, a seleção de *features* e o classificador utilizado, mas não a capacidade do aplicativo de efetivamente reconhecer eventos sonoros em situações reais. Na prática, durante o processo de reconhecimento, sons não conhecidos pelo aplicativo, ou seja, sons pertencentes a classes não registradas na BC, são igualmente classificados e o resultado apresentado pode ser bastante incoerente. Nos classificadores *naive Bayes* e *Bayes network*, por exemplo, onde o resultado é baseado na probabilidade, a classificação de eventos sonoros não previstos no aplicativo pode chegar a apresentar resultados de valor 1 (probabilidade máxima), ignorando portanto a incerteza contida na informação. Este problema também foi abordado no estudo realizado por Matthews et al. [1], no qual os autores propõem três níveis de confiança para serem apresentados aos usuários. Os níveis foram baseados em limiares que eles estabeleceram a partir dos resultados retornados pelo sistema de ESR usado por Malkin et al. [33]. Para abordar esta questão na presente pesquisa, optou-se por apresentar ao usuário o nível de confiança da classificação baseado na distância euclidiana, que foi a função de distância utilizada no algoritmo de busca do teste com NN realizado no experimento, sendo este o classificador que obteve maior percentual de acerto. Para tanto, foi elaborada uma equação que tem o objetivo de indicar o grau de pertencimento de uma nova instância ao grupo formado pelas instâncias da classe prevista, acrescentando à classificação uma informação de apoio. Documentada em artigo publicado em maio de 2014 [6], a ideia é medir quão próximo o som detectado está da classe apresentada no reconhecimento. No cálculo do indicador, demonstrado nas Equações (1), (2), (3) e (4), somente são utilizadas as instâncias pertencentes à classe prevista, contornando assim o problema do custo computacional, mencionado na Seção 5.6, decorrente da busca exaustiva implementada no algoritmo NN. Como referência ao indicador de pertencimento ao grupo é utilizado o acrônimo GPI, do inglês *group pertaining index*.

As equações são formuladas a partir de uma matriz P_{mxn} formada pelas instâncias da classe prevista expressas nas linhas e com os seus respectivos atributos expressos nas colunas. Considerando as instâncias como pontos do espaço $\mathbb{R}^n$ definem-se:

$d(x,y)$ como a distância euclidiana entre os pontos x e y

c como o centroide de P

m como o número de pontos em P

n como a dimensão do espaço P

a como o ponto correspondente à instância gerada a partir da nova amostra de som

p^* como o ponto mais próximo de a

$g(a,P)$ como o indicador de pertencimento de a ao grupo formado pelas instâncias de P

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

$$c_j = \frac{\sum_{i=1}^{m} p_{ij}}{m}, j = 1,2,3,...,n \quad (2)$$

$$d(a,p^*) \leq min(d(a,p_i)), i = 1,2,3,...,m \quad (3)$$

$$g(a,P) = d(a,c) - d(p^*,c) \quad (4)$$

Através dessa abordagem, GPIs com valores negativos indicam alto grau de pertencimento ao grupo. Para os valores positivos é necessário ajustar o resultado de acordo com a base, porém com as instâncias do experimento observamos que valores entre 0 e 1 indicaram um grau médio de pertencimento enquanto valores acima de um, na maioria das vezes, ocorriam em classificações erradas. Com base nessas observações diferenciaram-se seis níveis de confiança, conforme apresentado na Tabela 9.

Tabela 9 - Níveis de confiança da classificação

NÍVEL	INTERVALO
0	$g(a,P) \geq 2$
1	$2 > g(a,P) \geq 1.5$
2	$1.5 > g(a,P) \geq 1$
3	$1 > g(a,P) \geq 0.5$
4	$0.5 > g(a,P) \geq 0$
5	$g(a,P) < 0$

Devido a oscilações de acordo com a classe prevista, é possível perceber que, para obter resultados mais estáveis, os intervalos ainda precisam de um ajuste mais preciso, considerando as particularidades do conjunto de instâncias de cada classe. Apesar disso, verificou-se um benefício significativo na interpretação dos resultados da classificação com a aplicação dos limiares estabelecidos.

7 Personalização da Base de Conhecimento

A tarefa de classificar sons não estruturados foi um dos maiores desafios enfrentados nesta pesquisa. Para o reconhecimento da fala ou música, os processos de classificação normalmente empregam modelos pré-definidos que podem ser atualizados periodicamente, acompanhando a evolução do universo de sons a serem tratados. De forma similar, os estudos existentes sobre ESR empregam, na maioria dos casos, um modelo pré-definido de categorias e são usualmente aplicados na indexação e recuperação de documentos de áudio e vídeo. No entanto, o tema é abordado em situações específicas e/ou monitoradas, contando com o apoio da pré-estruturação de informações. Em uma abordagem genérica de ESR, utilizando somente os recursos de processamento do dispositivo móvel, esta estratégia não é viável devido à ampla variedade de sons incluídos na categoria [8]. Ademais, a responsividade requerida em R6 torna ainda mais complexa a utilização de bases de grande porte. Diante dessa problemática, adotou-se a estratégia de personalização da BC. Para tanto, o sistema define a classe genérica "Som" ("Sound" na implementação), porém não estabelece previamente os sons que serão incluídos, permitindo aos usuários definir seu próprio universo de sons. Esta abordagem satisfaz os requisitos de abrangência (R4) e dinamicidade de conhecimento (R5) do sistema. Usando dados personalizados, apenas as informações referentes aos sons definidos pelo usuário serão processadas, reduzindo o custo computacional da classificação e evitando a necessidade de escalar o sistema com recursos adicionais com o uso de *cloud computing*, por exemplo, o que comprometeria a satisfação do requisito R1. No entanto, no sistema proposto, personalizar implica encarregar o usuário da construção desta base, o que restringe o universo de sons conhecidos pelo aplicativo à capacidade de amostragem do usuário. Assim como descrito no Capítulo 9, usuários surdos podem realizar tarefas de amostragem de som, mas sua capacidade é limitada, na maioria das vezes, a eventos sonoros vinculados a alguma manifestação visual, ou a sons que podem ser produzidos por eles mesmos. Para outros sons, há também a possibilidade de que os usuários usem o espectrograma como uma referência de eventos ocorrentes, mas isso requer muita prática para se conseguir amostras confiáveis.

Ainda que não abordado nesta pesquisa, o compartilhamento da BC entre usuários poderia solucionar as limitações observadas na opção pela personalização. Hipke et al. já haviam sugerido em seu estudo [17] o uso de *crowdsourcing* para fomentar a construção da base do seu aplicativo. Afinal, segundo eles, para a maioria dos usuários, alguns sons podem ser especialmente difíceis de serem capturados como vidro quebrando, sirene, buzina, bebê chorando, etc. Os autores também expressaram a preocupação com o uso desses sons em ambientes distintos levantando a seguinte questão: será que o som capturado por um usuário poderá atender às necessidades de outros usuários em diferentes ambientes acústicos? Além deste problema, também considera-se no presente estudo a possibilidade de incompatibilidade entre as diferentes BCs e os dispositivos que as recebem. De fato, foram verificados alguns problemas neste sentido, porém devido ao pequeno número de dispositivos testados ainda não é possível afirmar se esta incompatibilidade é uma limitação que impediria o intercâmbio de BCs. Percebeu-se que alguns dispositivos, mesmo de diferentes modelos e fabricantes, são mutuamente compatíveis. Diferenças na captura de áudio podem ocorrer devido ao modelo do microfone, a ruídos de alta frequência emitidos pelo dispositivo e/ou devido à impossibilidade de desativar a função de ganho automático de áudio do sistema, o que resultaria na captura de sinais alterados, confundindo os classificadores. Esta última hipótese é menos provável, uma vez que todos os dispositivos testados eram compatíveis com os serviços de reconhecimento de voz, que só fornece um desempenho satisfatório com o ganho automático de áudio desligado. Portanto, estudos adicionais precisam ser realizados para esclarecer se o compartilhamento em larga escala de BCs é uma abordagem viável.

8 Aplicativo Proposto

É apresentada aqui a versão 1.0.1 do aplicativo VSom publicada no momento da defesa desta dissertação, sendo mais atual do que a versão utilizada no teste alfa (Capítulo 9). Além das funcionalidades e características do aplicativo aqui descritas, esta versão disponibiliza um tutorial (Apêndice B) com o objetivo de habilitar o usuário a utilizar o aplicativo sem a necessidade de uma explicação presencial.

8.1 Funcionalidades

O aplicativo consiste em quatro funcionalidades básicas, com os respectivos ícones apresentados na Figura 4. As funcionalidades de registro e de reconhecimento pontual de sons disponibilizam um espectrograma (Figuras 5 e 10) acionado automaticamente durante a captura do áudio, proporcionando a visualização das frequências do sinal sendo processado. A parte vermelha do espectrograma corresponde ao fragmento do som sendo registrado ou reconhecido. A ideia de oferecer esse retorno visual é permitir que o usuário, surdo ou ouvinte, possa interagir durante as tarefas de registro e de reconhecimento com maior precisão. O espectrograma pode ser também acionado pelo usuário com um toque sobre a área da sua imagem, o que permite ver as frequências dos sons ocorrentes no ambiente mesmo quando nenhum processo de registro ou reconhecimento estiver sendo executado.

Figura 4 - Funcionalidades básicas do aplicativo

8.1.1 Registro de sons

Apresentada na inicialização do aplicativo, esta tela (Figura 5) permite ao usuário construir sua própria base de registros sonoros. O processo é iniciado pelo usuário, porém o registro só é acionado quando um evento sonoro é detectado. A captura do sinal é interrompida automaticamente quando o final do evento é detectado, porém o usuário tem a opção de encerrar a captura em qualquer momento. Mais detalhes sobre a detecção de eventos sonoros são apresentados na Seção 8.7. Após a captura, será solicitado ao usuário rotular o som, ou seja, definir a classe correspondente, e indicar o ambiente em que o registro foi efetuado. Na Figura 6 pode ser visualizado o modelo do processo correspondente a esta funcionalidade.

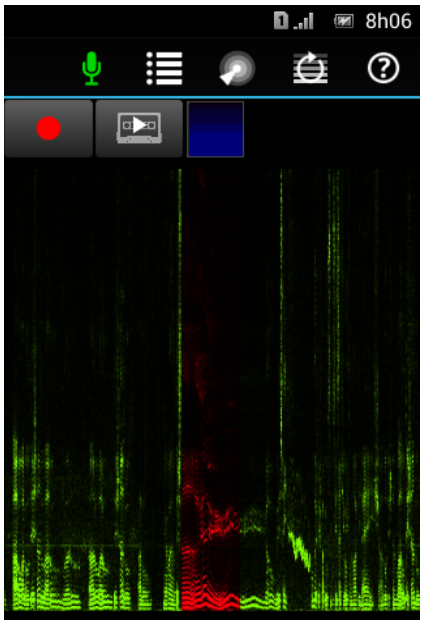


Figura 5 - Registro de sons

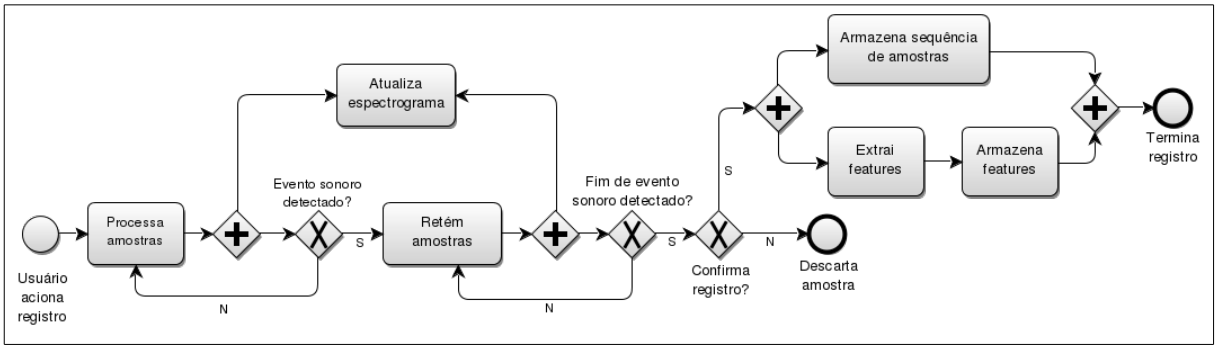


Figura 6 - Modelo do processo executado no registro de sons, utilizando padrão BPMN

8.1.2 Exploração de sons

Esta funcionalidade permite que o usuário navegue em seus dados. A tela correspondente é mostrada na Figura 7, onde é apresentada uma lista das classes de som armazenadas e seus respectivos registros. Através de um registro de som, o usuário pode reproduzir o áudio gravado, cujas frequências serão exibidas em um espectrograma. A lista também permite ao usuário editar ou excluir seus itens. O formulário de alteração dos dados do som (Figura 8) permite, entre outras funções, especificar uma meta-categoria que diferencia o som entre quatro níveis de importância: ignorável, habitual, importante ou urgente. Isso aplicará um filtro aos resultados e poderá acionar alertas personalizados na tela de reconhecimento automático. Quanto aos processos de interação inerentes à exploração de dados, não são verificados padrões significativos. Além de navegar pelos dados registrados, nesta tela o usuário pode realizar as seguintes ações:

- editar dados de um som ou ambiente
- excluir um som ou ambiente (e seus respectivos registros de som)
- reproduzir um registro de som
- excluir um registro de som

Figura 7 - Exploração de sons

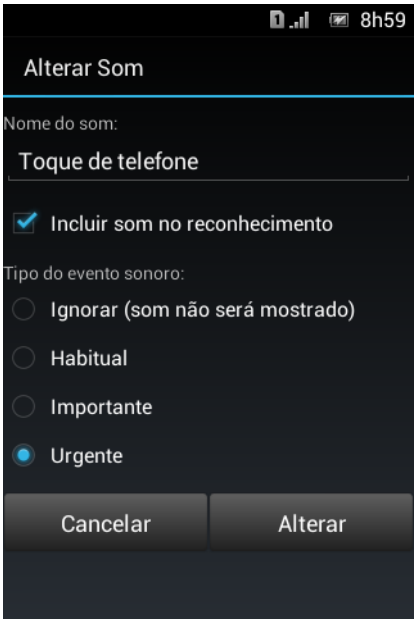


Figura 8 - Formulário de alteração dos dados do som

8.1.3 Reconhecimento pontual de sons

Esta funcionalidade permite ao usuário definir o momento em que a tarefa de reconhecimento será executada. É apresentado um espectrograma para auxiliar o usuário a identificar o momento em que o som ocorre (Figura 10). Opcionalmente, o usuário pode definir o ambiente para melhorar a acurácia da classificação do som. Uma vez que o reconhecimento estiver concluído, o resultado é informado e o processo é encerrado. Como mostrado na Figura 9, além do rótulo da classe prevista, o resultado informa também o índice de acerto (nível de confiança) da classificação. Na Figura 11 pode ser visualizado o modelo do processo correspondente a esta funcionalidade.

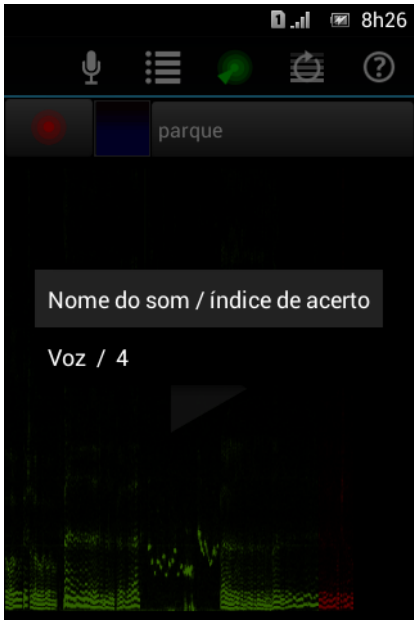


Figura 9 - Reconhecimento pontual

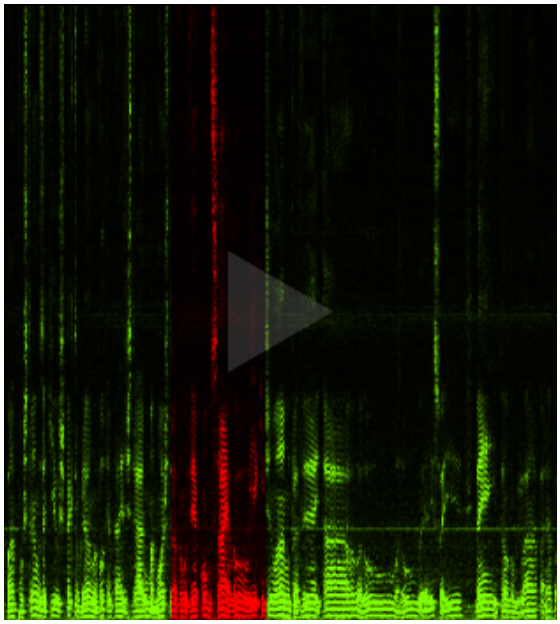


Figura 10 - Espectrograma

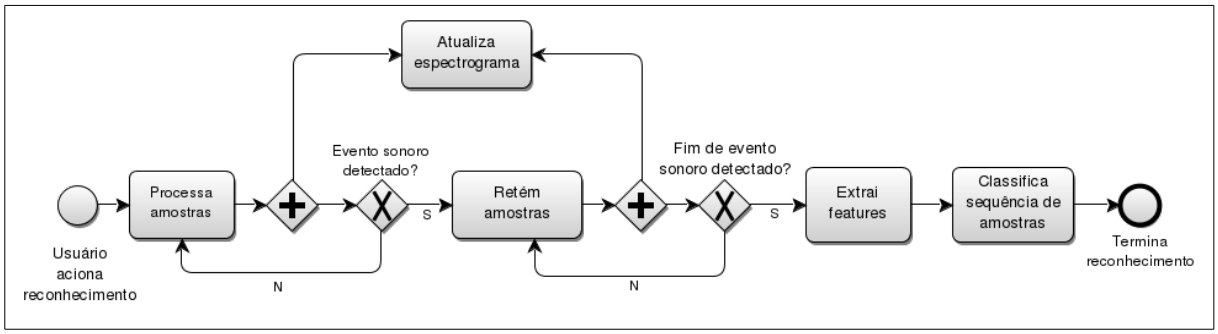


Figura 11 - Modelo do processo executado no reconhecimento pontual de sons, utilizando padrão BPMN

8.1.4 Reconhecimento automático de sons

Nesta funcionalidade o aplicativo fornece uma lista atualizável em tempo real dos sons ocorrentes. Cada item da lista apresenta o momento em que o evento sonoro foi detectado, o rótulo da classe de som prevista e o nível de confiança da classificação correspondente. A lista também avisa o usuário sobre a importância do som, definido no formulário de alteração do som (Figura 8), que é acessado através da funcionalidade de exploração de sons. Como mostrado na Figura 12, o aplicativo apresenta fundo preto para sons habituais, azul para sons importantes e vermelho para sons urgentes. Na Figura 13 pode ser visualizado o modelo do processo correspondente a esta funcionalidade, que é executado continuamente em *threads* separadas para cada evento sonoro detectado a partir da tarefa "Processa amostras" até ser interrompido pelo usuário.

Figura 12 - Reconhecimento automático

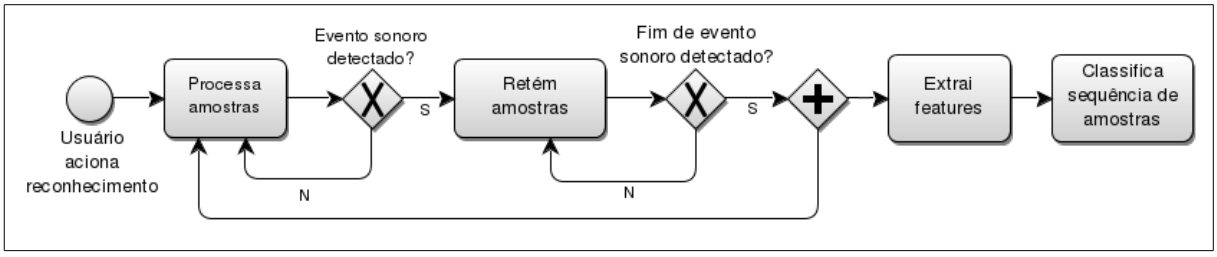


Figura 13 - Modelo do processo executado no reconhecimento automático de sons, utilizando padrão BPMN

8.2 Modelo conceitual

Como é de se esperar em uma abordagem não estruturada (referindo-se aos sons), o modelo é bastante simples, restringindo-se aos conceitos essenciais do sistema, que são: Som, Registro de som e Ambiente, como mostrado na Figura 14.

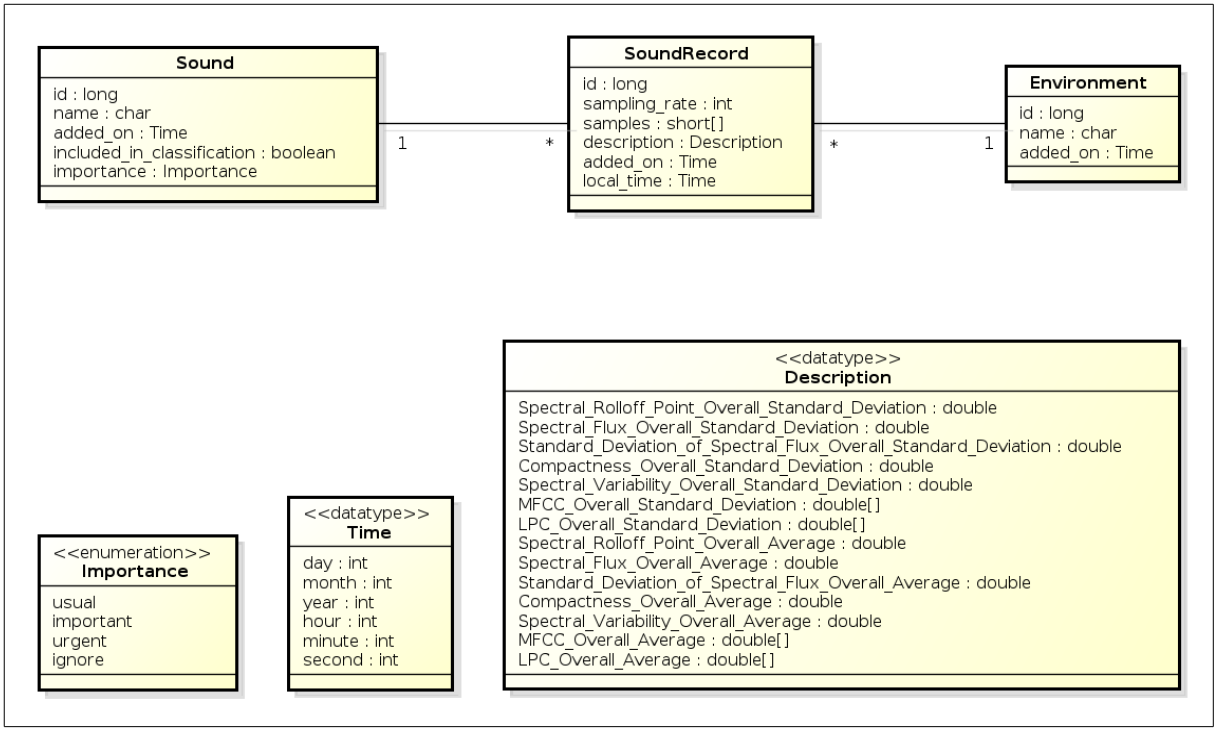


Figura 14 - Modelagem do sistema

Tabela 10 - Descrição das classes do modelo

CLASSE	DESCRIÇÃO
Sound	Evento sonoro.
SoundRecord	Registro de sinal de áudio.
Environment	Ambiente onde o evento sonoro é registrado.
Description	Descrição de baixo nível do sinal de áudio. São as *features*.
Importance	Grau de importância, que pode ser: habitual, importante, urgente ou ignorável.
Time	Ponto no tempo definido por: dia, mês, ano, hora, minuto e segundo.

Tabela 11 - Detalhes da classe “Sound”

CLASSE: Sound	
ATRIBUTO	DESCRIÇÃO
id	Identificador.
name	Nome do som. Ex: Voz aguda.
added_on	Data e hora da inserção do som.
included_in_classification	Indica se o som é ou não incluído no processo de classificação.
importance	Importância do som.

Tabela 12 - Detalhes da classe “SoundRecord”

CLASSE: SoundRecord	
ATRIBUTO	DESCRIÇÃO
id	Identificador.
sampling_rate	Taxa de amostragem.
samples	Sequência de amostras do sinal de áudio digital em 16 bits.
description	*Features* de áudio.
added_on	Data e hora da inserção do registro de som.
local_time	Data e hora locais da inserção do registro de som.

Tabela 13 - Detalhes da classe “Environment”

CLASSE: Environment	
ATRIBUTO	DESCRIÇÃO
id	Identificador.
name	Nome do ambiente. Ex: Restaurante.
added_on	Data e hora da inserção do ambiente.

8.3 Persistência dos dados

Considerando o modelo apresentado na Figura 14, com exceção dos dados correspondentes aos atributos "samples" e "description" da classe "SoundRecord", todos os demais dados são armazenados em um banco de dados SQLite[31]. As sequências de amostras do sinal de áudio digital ("samples") são armazenadas em arquivos compactados no formato GZIP[32]. Os valores das *features* ("description"), são armazenados em arquivo de formato Atributo-Relação ou *Attribute-Relation File Format* (ARFF)[33].

8.4 Algoritmo de classificação

Após a análise dos testes de classificação descritos no Capítulo 5, o aplicativo foi configurado com o classificador *naive Bayes* com os mesmos parâmetros utilizados no experimento. Não foi utilizada nenhuma função de reforço no treinamento do classificador dependente da avaliação do usuário, que só interfere na classificação incluindo ou removendo amostras de som do aplicativo. Além disso, o aplicativo também disponibiliza a opção de excluir temporariamente uma classe de som do processo de reconhecimento, conforme mostrado na Figura 8.

8.5 GPI e nível de confiança da classificação

Quanto ao nível de confiança da classificação, este é processado de acordo com os limiares apresentados no Capítulo 6, definidos a partir do indicador de pertencimento (GPI) da nova instância ao grupo de instâncias da classe prevista. Na atual implementação do sistema, o nível zero foi configurado para indicar um som não reconhecido. Para níveis de 1 a 5, o valor é apresentado juntamente com o resultado da classificação, sendo 1 o nível mais baixo de confiança e 5 o mais elevado.

[31] http://www.sqlite.org/about.html

[32] http://www.gzip.org/format.txt

[33] http://www.cs.waikato.ac.nz/ml/weka/arff.html

8.6 Responsividade

A principal preocupação no projeto da interface foi assegurar a responsividade do aplicativo. O processamento necessário para realizar a extração de *features* e a classificação do som pode facilmente bloquear a interface por alguns segundos. A implementação de tarefas assíncronas resolveu a maioria dos problemas, mas nem todo o processamento pode ser executado em *background*, já que algumas funções devem fornecer informações para o usuário em tempo real. O espectrograma é um exemplo disso, apresentando informação espectral sobre os segmentos de áudio em intervalos equivalentes a frações de segundo. Cada coluna do gráfico exibido é gerada a partir da metade dos 2048 valores resultantes da FFT, calculada sobre o mesmo número de amostras do sinal original capturado a 48000 Hz. Utiliza-se somente a metade porque a informação espectral é redundante. A opção por 1024 valores para a exibição do espectrograma se deve à adequação da resolução da imagem às dimensões da tela do dispositivo móvel. Ademais, a FFT utiliza necessariamente vetores com dimensão igual a uma potência de 2. A qualidade da exibição do espectrograma é configurável pelo usuário como alta, média ou baixa, exibindo respectivamente 23, 12 e 8 colunas por segundo aproximadamente. A qualidade padrão do espectrograma é a mais baixa. O objetivo desta configuração é permitir ao usuário o ajuste desta função à capacidade de processamento do dispositivo, evitando assim atrasos perceptíveis entre a ocorrência real dos eventos sonoros e a sua exibição espectral. Levando-se em conta a condição do usuário, que não pode verificar auditivamente esses atrasos, o aplicativo avisa sobre suas ocorrências nas telas de registro e reconhecimento pontual do som transformando o quadrado azul em laranja/ocre (Figura 15) de acordo com o atraso verificado.

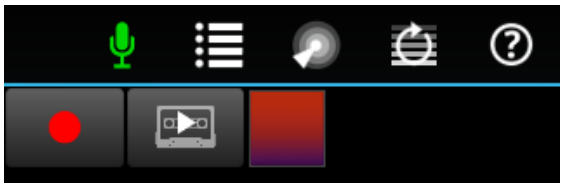


Figura 15 - Aviso de atraso do espectrograma

Outra importante questão relacionada com a responsividade é o tempo que o aplicativo dedica ao reconhecimento do som. Com o classificador *naive Bayes*, o processo de

reconhecimento, que inclui extração de *features* e classificação, teve duração de 3,6 segundos. Esse resultado respeita o limite de 5 segundos estabelecido em R6, que se refere ao tempo máximo de espera tolerado por usuários de dispositivos móveis. No entanto, este limite é somente um ponto de referência para o desenvolvimento de um sistema com responsividade satisfatória. Como mencionado em R6, o tempo do reconhecimento do som deve ser o mínimo possível. Uma excelente *performance* seria alcançada se o resultado do reconhecimento fosse apresentado quase que imediatamente após a detecção do evento sonoro.

8.7 Detecção de eventos sonoros

A detecção automática de eventos sonoros é a capacidade do aplicativo de detectar o início e o fim de um evento sonoro ocorrido no ambiente. No registro e no reconhecimento pontual de sons, esta função é necessária especialmente devido aos seguintes problemas:

- eventual demora de resposta a toques na tela dos dispositivos que poderiam prejudicar a marcação manual do início e do final do evento sonoro;
- dificuldade de capturar manualmente sons breves como estalos ou batidas;
- dificuldade do usuário perceber o início e o final de sons que não estejam associados a nenhuma manifestação visual ou tátil.

Ademais, no reconhecimento automático, a detecção evita que o aplicativo classifique sequências de sinal de áudio sem eventos sonoros relevantes, como trechos de silêncio, ruído branco de baixa intensidade ou sons distantes. No aplicativo foi implementado o método de detecção utilizado por Lu et al. [18] no desenvolvimento de seu *framework* para sistemas com funções de sensoriamento de áudio. O método, denominado pelos autores como "controle de admissão de quadro" (do inglês *frame admission control*), é baseado na amplitude e na entropia espectral do sinal. Baixa amplitude indica segmentos de silêncio ou com sons de baixa intensidade. Quanto à entropia espectral, a informação é interpretada da seguinte forma: uma alta entropia indica um espectro plano (silêncio ou ruído branco) enquanto uma baixa entropia indica um padrão bem destacado no espectro. Portanto, entropia espectral e amplitude são informações que se complementam para definir se um quadro será admitido ou

não. A admissão do quadro ocorrerá se a amplitude ou entropia atingirem os limiares definidos. No caso da amplitude, o limiar define o valor mínimo, enquanto que na entropia é definido o valor máximo. Com este método, sons de alta intensidade serão admitidos de qualquer forma independentemente da entropia, pois a amplitude será alta. Portanto, um ruído branco de alta intensidade como um chiado forte, por exemplo, será admitido como evento sonoro. Já os sons de baixa intensidade só serão admitidos se apresentarem baixa entropia, o que indica que são sons com padrão bem destacado no espectro como, por exemplo, uma pessoa falando baixo ou longe do dispositivo. Entretanto, um ruído branco de baixa intensidade, como som de ventilação de ar por exemplo, não será admitido, pois apresenta baixa amplitude e alta entropia. A amplitude do sinal é obtida a partir do RMS (acrônimo de *root mean square*) da sequência de n amostras, cujo cálculo é apresentado na Equação (5). Quanto à entropia, após efetuada a FFT do quadro q, obtém-se a magnitude espectral expressa em um vetor de dimensão n, que é tratado como uma função densidade de probabilidade p. Finalmente, a entropia espectral H é calculada como apresentado na Equação (6).

$$x_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{5}$$

$$H_q = -\sum_{i=1}^{n} p_i \log p_i \tag{6}$$

8.8 Detalhes de publicação

Devido ao armazenamento de informações pessoais do usuário (referindo-se aos registros de som) realizado pelo aplicativo e também ao fato de este ainda estar em fase experimental, a sua publicação foi efetuada sob restrição de conteúdo a usuários com maturidade média. A classificação quanto à maturidade é requerida pelo serviço de distribuição de aplicativos da Google Play[34].

Para que os usuários possam começar a experimentar todas as funcionalidades logo após a instalação do aplicativo, foram incorporadas algumas classes de som com seus

[34] Google Play é marca registrada da Google Inc.

respectivos registros, conforme mostrado na Tabela 14. A importância dos sons (atributo "importance" da classe "Sound") foi definida para que, ao utilizar o reconhecimento automático, o usuário perceba o quanto antes que o aplicativo oferece a possibilidade de definir esta meta-classe.

Tabela 14 - Detalhes da base de exemplo fornecida junto com o aplicativo

CLASSE DE SOM	DESCRIÇÃO	IMPORTÂNCIA	Nº DE REGISTROS DE SOM
Assovio	Pessoa assoviando.	importante	16
Batendo à porta	Pessoa batendo à porta.	urgente	13
Batendo palma	Pessoa(s) batendo palmas.	importante	11
Chaves	Molho de chaves sendo sacudido.	habitual	13
Pessoas falando	Grupo de pessoas falando.	habitual	20
Toque de telefone	Toque de telefone eletrônico.	urgente	15
Voz	Pessoa falando.	importante	25

9 Teste Inicial com Usuários (Teste Alfa)

Com o objetivo de realizar uma avaliação do sistema, foi conduzido um teste inicial com um grupo de potenciais usuários utilizando uma versão alfa do aplicativo. Esta versão não disponibilizava o tutorial incorporado (Apêndice B). Há também algumas diferenças de implementação com relação à versão 1.0.1 apresentada no Capítulo 8, especialmente quanto à otimização do processo de captura do sinal de áudio visando a um menor consumo de energia. No entanto, estas diferenças não interferiram no teste, pois a bateria dos dispositivos suportou o processamento durante toda a dinâmica. Quanto ao tutorial, este não se fez necessário devido à explicação presencial e ao auxílio do intérprete de português-LIBRAS durante o encontro.

Participaram do teste cinco usuários, dos quais somente quatro puderam responder ao questionário de avaliação (Apêndice C). Dentre estes, o grau de surdez variou de moderada a profunda, e a idade de 21 a 28 anos. Todos estavam familiarizados com dispositivos móveis. O grupo era composto por três homens e duas mulheres. Quanto ao nível de estudos, dois haviam concluído até o ensino médio, dos quais um era formado em escola técnica; dois eram estudantes do ensino superior e um completou a graduação. Quatro eram membros de equipes de projetos na universidade, dentre os quais um trabalhou como professor de crianças. O teste foi realizado no Laboratório Didático de Ciências para Surdos (LaDiCS) durante cerca de uma hora e consistiu em três partes: apresentação, teste e avaliação do aplicativo. Na primeira parte, o aplicativo foi apresentado aos participantes, quando foram explicadas sua finalidade e operação. Além de esclarecer dúvidas sobre o funcionamento do sistema, os participantes levantaram várias questões sobre os sons a serem abordados. Depois de estarem familiarizados com a interface, eles realizaram alguns testes de reconhecimento de sons como bater à porta e de registro e visualização do espectrograma da própria voz e de assovios, dentre outros sons. A terceira parte consistiu na avaliação do aplicativo. O encontro foi organizado pelos integrantes do Projeto Surdos-UFRJ: Profa. Vivian Rumjanek, Prof. Flavio Eduardo P. da Silva e o professor e intérprete Tiago Batista.

Sobre o grau de surdez, as categorias utilizadas neste estudo seguem o padrão utilizado pela área de saúde e, tradicionalmente, pela área educacional [34]. Abaixo são apresentados detalhes das categorias, sendo considerados surdos os indivíduos com surdez severa, profunda ou total e parcialmente surdos (ou com deficiência auditiva) os indivíduos com surdez leve e moderada.

- Surdez leve - perda auditiva de até 40 decibéis. O indivíduo não consegue perceber igualmente todos os fonemas das palavras. Além disso, a voz baixa ou distante não é ouvida. No entanto, a maioria dos sons do ambiente são ouvidos.
- Surdez moderada - perda auditiva entre 40 e 70 decibéis. Para que o indivíduo compreenda a palavra, é necessária uma voz de certa intensidade. No entanto, vários sons do ambiente ainda são ouvidos.
- Surdez severa - perda auditiva entre 70 e 90 decibéis. Somente compreende a palavra com voz forte e próxima ao ouvido. A percepção de sons do ambiente se limita a barulhos intensos.
- Surdez profunda - perda auditiva entre 90 e 120 decibéis. O indivíduo não consegue compreender a palavra. Somente sons muito intensos podem ser percebidos.
- Surdez total - perda auditiva superior a 120 decibéis, nenhum som é percebido.

9.1 Comentários dos participantes

Durante a apresentação do aplicativo, os participantes expressaram a necessidade de reconhecer sons relacionados com chamados ao seu nome, assim como a sinais sonoros emitidos por outros indivíduos surdos. Eles também perguntaram sobre a possibilidade de indicar quando uma determinada pessoa está falando, através da identificação da sua voz. Além disso, eles queriam saber se era possível distinguir se a voz detectada era de uma pessoa que estava calma ou nervosa, alegre ou triste. E à medida que se acostumaram com o espectrograma, ficaram interessados na possibilidade de reconhecer sons através da interpretação do gráfico gerado. Outra questão levantada foi sobre a quantidade de sons que o

aplicativo é capaz de armazenar, pois consideraram importante reconhecer uma ampla gama de sons. Eles também expressaram a necessidade de reconhecer sons com o usuário em movimento. Uma questão levantada por quase todos os participantes foi a preferência pelo uso de imagens em vez de texto para representar os sons no aplicativo. Eles também expressaram sua preocupação com o consumo da bateria. Ademais, os participantes indicaram alguns dos sons que eles gostariam de poder reconhecer, que foram os seguintes: sons de animais, chuva, raios, vento, música, batida de palmas, tiros, choro, buzinas e gritos.

9.2 Avaliação

Ao final do encontro, os participantes formalizaram suas considerações e avaliaram o aplicativo respondendo ao questionário disponível no Apêndice C. Um dos participantes, por motivos pessoais, não pôde estar presente durante a avaliação. Cada campo de pontuação contém o número total de itens selecionados pelos usuários. Por razões de legibilidade, campos com pontuação zero foram deixados vazios.

Tabela 15 - Pontuação total de acordo com a questão:
Marque os ambientes onde você sente mais necessidade de ser alertado sobre sons ocorrentes

AMBIENTE	PONTUAÇÃO	AMBIENTE	PONTUAÇÃO	AMBIENTE	PONTUAÇÃO
aeroporto	3	loja		rodoviária	3
carro	3	metrô	3	rua com veículos	3
casa	4	ônibus	3	rua de pedestres	2
centro cultural		parque	3	rua tranquila	1
estação de metrô	2	piscina		sala de aula	2
estacionamento	1	praia		shopping center	
evento esportivo		restaurante	2	supermercado	2
laboratório		reunião social		trabalho	2

Além da avaliação do aplicativo, os participantes puderam expressar suas prioridades quanto à consciência situacional como na Questão 1 (Tabela 15), na qual foram consultados sobre os ambientes onde sentem maior necessidade de serem alertados sobre sons

ocorrentes, além dos comentários acrescentados especificando os sons que gostariam de poder reconhecer. Nas Questões 2 e 3 (Tabela 16), procura-se conhecer a percepção dos participantes quanto ao benefício efetivo que o aplicativo ofereceu. Nas questões seguintes foram feitas consultas sobre tópicos específicos do aplicativo. Na Questão 4 (Tabela 17) foi solicitado aos participantes a avaliação das quatro funcionalidades básicas do aplicativo, além do espectrograma e da reprodução de amostras. O guia visual, mencionado no enunciado dessa questão, foi fornecido aos participantes em uma página impressa com as telas das funções a serem avaliadas e seus respectivos nomes. Na Questão 5 (Tabela 19) os participantes indicaram os problemas encontrados e na Questão 6 (Tabela 18) expressaram sua opinião sobre possíveis melhorias no aplicativo.

Tabela 16 - Pontuação sobre questões relacionadas à utilidade efetiva do aplicativo

QUESTÃO	NÃO	SÓ UMA VEZ	SIM, ALGUMAS VEZES	SIM, MUITAS VEZES	NÃO SABE / NÃO RESPONDEU
O aplicativo ajudou você a perceber o que ocorria no ambiente?		1	2		1
O aplicativo permitiu você perceber algo no ambiente que seria muito difícil sem o seu uso?	1		1	1	1

Tabela 17 - Pontuação total de acordo com a questão: Avalie as seguintes funções do aplicativo

FUNÇÃO	MUITO RUIM	RUIM	REGULAR	BOA	MUITO BOA
Espectrograma				4	
Registro de amostras		1		2	1
Exploração de sons				4	
Reprodução de amostras		1		2	1
Reconhecimento pontual			2	1	1
Reconhecimento automático			1	3	

Tabela 18 - Pontuação total de acordo com a questão: Indique eventuais problemas que você encontrou no aplicativo

PROBLEMA ENCONTRADO	PONTUAÇÃO
difícil de usar	1
interface confusa	2
processamento lento	1
ícones e letras pequenas	1
espectrograma de baixa qualidade	2
espectrograma não mostra nada interessante	1
o aplicativo não fazia o que eu esperava	
o aplicativo reconheceu sons errados	1
não consegui registrar amostras de som	1
não consegui usar o explorador de sons	
não entendi o que significa o “índice de acerto”	1

Tabela 19 - Pontuação total de acordo com a questão: Indique a sua avaliação sobre possíveis melhorias no aplicativo

POSSÍVEL MELHORIA	DESNECESSÁRIA	ÚTIL	IMPORTANTE
reconhecer sons mais rapidamente	1	1	2
emitir sinal visual mais chamativo ao reconhecer som		2	2
emitir sinal vibratório ao reconhecer som			4
indicar a direção da fonte emissora do som		2	2
salvar histórico dos sons reconhecidos		1	3
espectrograma com mais qualidade		1	3
salvar imagens do espectrograma			4
mudar as cores do aplicativo		1	3

9.3 Visão geral do teste

Através do retorno dos usuários e dos temas suscitados durante o encontro, é possível reafirmar a importância de proporcionar a ampliação da consciência situacional de indivíduos surdos. Apesar de alguns problemas apontados pelos participantes no questionário de

avaliação, três deles expressaram claramente que o aplicativo os ajudou a perceber o que estava acontecendo ao redor. No entanto, para atender a todos os cenários de reconhecimento de som solicitados durante o teste, será necessário implementar novas funcionalidades no sistema, especialmente quanto à identificação de voz. De um modo geral, a receptividade dos participantes foi muito inspiradora e a avaliação do aplicativo foi boa.

O teste também esclareceu uma preocupação, considerada frequentemente ao longo da pesquisa, quanto à capacidade de usuários surdos efetuarem registros de som. Durante o encontro de cerca de 1 hora de duração, os participante tiveram aproximadamente 10 minutos dedicados ao uso do aplicativo. Mesmo nesse curto espaço de tempo eles foram capazes de, individualmente ou em duplas, experimentar as 4 funcionalidades básicas: Registro, Exploração, Reconhecimento Pontual e Reconhecimento Automático.

10 Considerações Finais

10.1 Contribuições

O estudo sobre sistemas de ESR utilizando computação móvel é um tema pouco explorado, especialmente quando direcionado a um público específico, como no caso dos surdos. A formalização dos requisitos e a apresentação de soluções às dificuldades encontradas no desenvolvimento de um sistema com estas características constituem, portanto, uma contribuição relevante para o domínio das tecnologias assistivas.

O experimento conduzido nesta pesquisa (Capítulo 5) e a respectiva análise dos resultados demonstraram que é possível utilizar computação móvel para o reconhecimento de sons do ambiente em tempo real. Utilizando somente *smartphones*, o experimento forneceu informações importantes para o desenvolvimento de aplicativos móveis de ESR e pode contribuir para futuros estudos sobre o tema ao destacar êxitos e limitações da solução proposta. Como indicado no Capítulo 5.4, as BCs empregadas nos testes foram disponibilizadas *online* e podem ser utilizadas em futuros experimentos. A pesquisa forneceu também um novo indicador, denominado GPI, descrito no Capítulo 6. A sua função é extrair os valores necessários para definir o nível de confiança do reconhecimento do som. A equação proposta pode ser utilizada em outros processos de classificação onde as instâncias possuam atributos numéricos. O cálculo do indicador implica baixo custo computacional e pode auxiliar sobretudo abordagens com limitações de recursos, como no caso de sistemas móveis. O experimento mencionado e a proposta de representação da incerteza foram apresentados em artigo publicado em maio de 2014 [6].

A limitação de recursos inspirou também outra solução apresentada nesta pesquisa. Trata-se da personalização da BC, descrita no Capítulo 7. Utilizando dados personalizados, algumas dificuldades relacionadas à abrangência e dinamicidade de conhecimento do sistema são minimizadas, pois apenas as informações referentes aos sons definidos pelo usuário do sistema e, portanto, uma quantidade menor de classes, serão processadas, reduzindo o custo computacional da classificação. No aplicativo proposto, devido ao perfil dos usuários potenciais e à natureza extremamente diversificada e em constante

transformação dos sons do ambiente, a construção da BC é de difícil realização. Apesar de o tema não ser abordado neste estudo, o compartilhamento de BCs apresenta-se como uma potencial solução às limitações observadas na opção pela personalização.

Apesar de o escopo do estudo não incluir a transcrição da fala ou o reconhecimento de músicas, esta pesquisa também pode auxiliar estudos que procurem abordar tais funcionalidades. Questões de usabilidade tratadas na pesquisa e detalhes dos métodos utilizados para o processamento do áudio podem também contribuir em projetos e estudos semelhantes baseados em dispositivos móveis.

Como resultado da pesquisa, além do material documentado, foi publicado o aplicativo móvel de ESR denominado VSom, disponível gratuitamente *online*[35]. O aplicativo pode ser utilizado para futuros testes sobre ampliação da consciência situacional com participação de usuários surdos, pessoas com deficiência auditiva ou também usuários de protetores auriculares, fones de ouvido ou outros equipamentos que provoquem isolamento acústico do indivíduo. O aplicativo é útil também para auxiliar na compreensão da pesquisa realizada. Dados, interações e limitações tratados no estudo poderão ser ilustrados pelo aplicativo, pois o leitor poderá experimentar as suas funcionalidades.

Foram fornecidos também os resultados do teste alfa realizado com um grupo de potenciais usuários. Testes com estas características são escassos, especialmente utilizando computação móvel. Assim como ocorreu com a presente pesquisa com relação a abordagens anteriores, futuros estudos poderão se orientar pelas impressões dos participantes registradas no teste aqui documentado.

10.2 Limitações

Durante o uso do aplicativo, foi observado que sons sobrepostos podem ser classificados como se tratassem de um único som. Um exemplo deste problema foi verificado com o aplicativo sendo utilizado numa rua com trânsito de veículos, onde o som predominante era o de “Carros passando”, e o resultado apresentado foi “Alerta de garagem”, pois no momento da detecção passava uma pessoa com um molho de chaves preso à cintura, o

[35] Vsom está disponível para download em https://play.google.com/store/apps/details?id=app.vsom

que produzia um som que, combinado ao som dos veículos e outros sons de fundo, se assemelhou à classe “Alerta de garagem”, cujos registros também foram realizados em ruas com trânsito de veículos.

Na atual versão do aplicativo, o seu uso somente é indicado em ambientes monitorados como casa ou escritório, devido à necessidade de se manter o dispositivo protegido de movimentos e golpes que possam gerar interferências no sinal sonoro capturado. Nestes ambientes o usuário terá facilidade em utilizar o aplicativo sendo executado em um *smartphone* que o alertará com sinais visuais. Já quanto ao uso em locais públicos ou em atividades que exijam maior mobilidade, alternativas utilizando computação vestível seriam mais adequadas. No entanto, o uso desse tipo de dispositivo não é abordado no presente estudo. Esta limitação impediu que a ubiquidade requerida (R2) fosse satisfeita completamente.

Devido às características dos potenciais usuários, testes do aplicativo são de complexa realização. Porém, para que seja fornecido um estudo mais consistente sobre a utilidade efetiva do sistema, será necessária a condução de testes que reúnam um número maior de usuários e que incluam o uso do aplicativo durante um período mais longo, permitindo assim que os participantes o experimentem em diversos ambientes.

10.3 Conclusão

Os resultados do experimento, com desempenho comparável ao obtido em sistemas de ESR baseados em plataforma *desktop* [15] [31] [32], apontaram algumas potencialidades do uso de dispositivos móveis para este fim. Quanto à avaliação da duração dos processos, foi possível verificar que, apesar da limitação dos recursos de *hardware*, a execução do reconhecimento do som teve duração entre 3,6 e 4s, sendo, portanto, compatível com a responsividade requerida por usuários de dispositivos móveis (R6).

A abrangência e dinamicidade de conhecimento do sistema foram uns dos temas mais complexos tratados neste estudo. A utilização de uma BC com 300 instâncias, sendo 30 classes de som com 10 instâncias cada, foi fundamental para estabelecer algumas referências de configuração. No entanto, é preciso utilizar bases maiores, especialmente com mais

classes, para confirmar as tendências observadas no comportamento dos classificadores. O aumento do número de classes é o fator que pode tornar o reconhecimento inviável devido ao aumento do custo computacional, daí a importância de se verificar a sua influência na classificação.

A preocupação com a incerteza da classificação em sistemas de ESR foi relatada pelos participantes do estudo realizado por Matthews et al. [1]. Portanto, na presente pesquisa considerou-se a importância de fornecer essa informação para incentivar a adoção do sistema. O indicador de pertencimento ao grupo (GPI), apresentado no Capítulo 6, mostrou-se capaz de oferecer referências para informar ao usuário sobre o nível de confiança da classificação. Porém, para se obter resultados mais estáveis, o cálculo do indicador precisa ser aperfeiçoado, provavelmente avaliando outras características do grupo para permitir a representação necessária.

Durante o teste alfa, o aplicativo foi apresentado pelo autor com a ajuda de um intérprete. Para facilitar futuros testes será importante disponibilizar mais material de apoio como, por exemplo, vídeos em LIBRAS com explicações sobre as funcionalidades básicas, substituindo a apresentação presencial. Conforme verificado no teste, o aplicativo VSom foi capaz de proporcionar benefícios aos participantes. Com estudos adicionais sobre o compartilhamento da BC e o uso do aplicativo em dispositivos vestíveis, será possível desenvolver um sistema efetivamente funcional para usuários surdos, com maior potencial de adoção e que poderá ser utilizado em um maior número de ambientes.

Referências

[1] MATTHEWS, T.; FONG, J.; MANKOFF, J. Visualizing non-speech sounds for the deaf. In: ASSETS'05 - INTERNATIONAL ACM SIGACCESS CONFERENCE ON COMPUTERS AND ACCESSIBILITY, 7., 2005, Baltimore. **Proceedings ...** New York: ACM, p.52-59, 2005.

[2] CAMPBELL, A; CHOUDHURY, T. From smart to cognitive phones. **IEEE Pervasive Computing**, [S. l.], v.11, n.3, p.7-11, Jul./Sep. 2012.

[3] ARAÚJO, R. Computação ubíqua: princípios, tecnologias e desafios. In: SBRC 2003 - SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Natal. **Anais ...** [S. l. : s. n.], v.8, p.11-13, 2003.

[4] MOGI, R.; KASAI, H. Noise-robust environmental sound classification method based on combination of ICA and MP features. **Artificial Intelligence Research**, [S. l.], v.2, n.1, p.107-121, Dec. 2013.

[5] CHU, S.; NARAYANAN, S.; KUO, C.-C. J. Unstructured environmental audio: representation, classification and modeling. IN: WANG, W. (ED.). **Machine audition:** principles, algorithms and systems. Hershey: IGI Global, 2011. p.1-21.

[6] FANZERES, L.; BISCAINHO, L. W.; VIVACQUA, A. Reconhecimento de sons não estruturados do ambiente utilizando dispositivos móveis. In: CONGRESSO DE ENGENHARIA DE ÁUDIO AES-BRASIL, 12., 2014, São Paulo. **Anais ...** Rio de Janeiro: Sociedade de Engenharia de Áudio, p.31-38, 2014.

[7] WITTEN, I ; FRANK, E; HALL, M. **Data mining:** practical machine learning tools and techniques. 3rd ed., Burlington: Morgan Kaufmann, 2011.

[8] CHACHADA, S.; KUO, C.-C. Environmental sound recognition: a survey. In: APSIPA 2013 - SIGNAL AND INFORMATION PROCESSING ASSOCIATION ANNUAL SUMMIT AND CONFERENCE, 2013, Kaohsiung. **Proceedings ...** [S. l. : s. n.], p.1-9, 2013.

[9] HO-CHING, F. W.-L.; MANKOFF, J.; LANDAY, J. Can you see what I hear? The design and evaluation of a peripheral sound display for the deaf. In: CHI'03 - SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, Fort Lauderdale. **Proceedings ...** New York: ACM, p.161-168, 2003.

[10] HERSH, M.; JOHNSON, M. (EDS) COM ANDERSSON, C. ET AL. **Assistive technology for the hearing-impaired, deaf and deafblind**. London: Springer, 2003.

[11] AZAR, J.; SALEH, H.; AL-ALAOUI, M. Sound visualization for the hearing impaired. **International Journal of Emerging Technologies in Learning**, [S. l.], v.2, n.1, 2007.

[12] SAHU, D.; SHARMA, S.; DUBEY, V.; TRIPATHI, A. Cloud computing in mobile applications. **International Journal of Scientific and Research Publications**, [S. l.], v.2, n.8, p.1-9, Aug. 2012.

[13] CÁCERES, R.; FRIDAY, A. Ubicomp systems at 20: progress, opportunities, and challenges. **IEEE Pervasive Computing**, [S. l.], v.11, n.1, p.14-21, Jan. 2012.

[14] GOLDHOR, R. Recognition of environmental sounds. In: ICASSP'93 - IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 1993, Minneapolis, EUA. **Proceedings ...** Washington: IEEE Computer Society, p.149-152, 1993.

[15] TEMKO, A.; NADEU, C. Classification of acoustic events using SVM-based clustering schemes. **Pattern Recognition**, New York, v.39, n.4, p.682-694, Apr. 2006.

[16] MALLAT, S.; ZHANG, Z. Matching pursuits with time-frequency dictionaries. **IEEE Transactions on Signal Processing**, [S. l.], v.41, n.12, p.3397-3415, Dec. 1993.

[17] HIPKE, K.; HOLT, M.; PRICE, D.; BRUMET, A.; ANDERSON, K. **AudioVision:** sound detection for the deaf and hard-of-hearing. University of Washington. Disponível em: <http://www.cs.washington.edu/education/courses/cse481h/12wi/projects/accessible-sound/docs/paper.pdf> Acesso em: 30/05/2013.

[18] LU, H.; PAN, W.; LANE, N.; CHOUDHURY, T.; CAMPBELL, A. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In: MobiSys'09 - INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 7., 2009, Cracóvia. **Proceedings ...** New York: ACM, p.165-178, 2009.

[19] CHOUDHURY, T.; BORRIELLO, G.; CONSOLVO, S.; HAEHNEL, D.; HARRISON, B. ; HEMINGWAY, B. ; HIGHTOWER, J. ; KLASNJA, P.; KOSCHER, K. ; LAMARCA, A. ET AL. The Mobile Sensing Platform: an embedded activity recognition system. **IEEE Pervasive Computing**, Piscataway, v.7, n.2, p.32-41, Apr. 2008.

[20] ROSSI, M.; FEESE, S.; AMFT, O.; BRAUNE, N.; MARTIS, S.; TRÖSTER, G. AmbientSense: a real-time ambient sound recognition system for smartphones. In: PerMoby 2013 - INTERNATIONAL WORKSHOP ON THE IMPACT OF HUMAN MOBILITY IN PERVASIVE SYSTEMS AND APPLICATIONS, 2013, San Diego. **Proceedings ...** [S. l. : s. n.], 2013.

[21] LUMISONIC. Disponível em: <http://www.soundandmusic.org/projects/lumisonic> Acesso em: 30/05/2013.

[22] NIIDA, S.; UEMURA, S.; NAKAMURA, H. Mobile services - user tolerance for waiting time. **IEEE Vehicular Technology Magazine**, [S. l.], v.5, n.3, p.61-67, Sep. 2010.

[23] CARVALHO, C.; RAFAELI, Y. A língua de sinais e a escrita - possibilidades de se dizer, para o surdo. **Estilos da Clínica**, São Paulo, v.8, n.14, p.60-67, jun. 2003.

[24] MCKAY, C. **jAudio:** towards a standardized extensible audio music feature extraction system. Faculty of Music, McGill University, 2005.

[25] HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. The Weka data mining software: an update. **SIGKDD Explorations Newsletter**. New York: ACM, v.11, p.10-18, Nov. 2009.

[26] KIM, H.-G.; MOREAU, N.; SIKORA, T. **MPEG-7 audio and beyond: audio content indexing and retrieval**. Chichester: John Wiley & Sons, 2005.

[27] HARRIS, F. On the use of windows for harmonic analysis with the discrete Fourier transform. **Proceedings of the IEEE**, [S. l.], v.66, n.1, p.51-83, Jan. 1978.

[28] CHU, S.; NARAYANAN, S.; KUO, C.-C. J. Environmental sound recognition with time-frequency audio features. **Transactions on Audio, Speech and Language Processing**, Piscataway, v.17, n.6, p.1142-1158, Aug. 2009.

[29] ZENG, Z.; LI, X.; MA, X.; JI, Q. Adaptive context recognition based on audio signal. In: ICPR 2008 - INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 2008, Tampa. **Proceedings ...** [S. l. : s. n.], p.1-4, 2008.

[30] CHU, S.; NARAYANAN, S.; KUO, C.-C.; MATARIC, M. Where am I? Scene recognition for mobile robots using audio features. In: ICME 2006 - IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO, 2006, Toronto. **Proceedings ...** [S. l. : s. n.], p.885-888, 2006.

[31] WANG, J.-C.; WANG, J.-F.; HE, K.; HSU, C.-S. Environmental sound classification using hybrid SVM/KNN classifier and MPEG-7 audio low-level descriptor. In: IJCNN'06 - INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 2006, Vancouver. **Proceedings ...** IEEE, p.1731-1735, Jul. 2006.

[32] TOYODA, Y.; HUANG, J.; DING, S.; LIU, Y. Environmental sound recognition by multilayered neural networks. In: CIT'04 - INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, 4., 2004, Wuhan. **Proceedings ...** Washington: IEEE Computer Society, p.123-127, Sep. 2004.

[33] MALKIN, R. ET AL. First evaluation of acoustic event classification systems in CHIL project. In: HSCMA'05, Piscataway. **Proceedings ...** [S. l. : s. n.], 2005.

[34] LIMA, D. ET AL. **Educação infantil:** saberes e práticas da inclusão: dificuldades de comunicação e sinalização: surdez. 4a ed., Brasília: MEC, Secretaria de Educação Especial, 2006.

Apêndice A – Outputs dos Testes de Classificação

São apresentados aqui os *outputs* dos testes de classificação gerados pelo Weka, incluindo a configuração completa do esquema de cada classificador testado. Conforme mencionado na Seção 5.4, os testes foram realizados em dispositivo *desktop* para facilitar a condução do experimento. Porém, foram utilizadas somente as BCs construídas com o protótipo do aplicativo e executadas as mesmas classes neste implementadas. A mesma BC com 300 instâncias é testada com os 4 classificadores. Cada instância possui 54 atributos correspondentes às *features* de áudio, além do atributo "class". O Weka utiliza números para representar as classes da BC. Abaixo é apresentado o mapeamento aplicado (nº – nome da classe):

1 - Ondas do mar
2 - Canto de pássaro
3 - Torneira
4 - Secador de cabelo
5 - Carros passando
6 - Chaves
7 - Voz grave
8 - Veículo pesado passando
9 - Alerta garagem
10 - Assovio
11 - Aspirador de pó
12 - Batendo à porta
13 - Espirro
14 - Tosse
15 - Voz aguda
16 - Toque de telefone
17 - Pessoas falando
18 - Mexendo embalagem plástica
19 - Chuveiro
20 - Pedido de silêncio (sshhh)
21 - Batendo palma
22 - Guarda de trânsito apitando
23 - Chafariz
24 - Ronco
25 - Sinal de micro-ondas
26 - Fechadura
27 - Máq. lavar roupa
28 - Máq. lavar roupa centrifugando
29 - Panela de pressão
30 - Ar escapando

Nearest neighbor

```
=== Run information ===

Scheme:weka.classifiers.lazy.IBk -K 1 -W 0 -A
"weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-
last\""
Relation:     jAudio
Instances:    300
Attributes:   55

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         278               92.6667 %
Incorrectly Classified Instances        22                7.3333 %
Kappa statistic                          0.9241
Mean absolute error                      0.0108
Root mean squared error                  0.0687
Relative absolute error                 16.8276 %
Root relative squared error             38.2821 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.003      0.909     1         0.952      0.998    1
                 0.9       0          1         0.9       0.947      0.95     2
                 0.2       0          1         0.2       0.333      0.6      3
                 1         0          1         1         1          1        4
                 1         0.014      0.714     1         0.833      0.993    5
                 1         0          1         1         1          1        6
                 1         0.003      0.909     1         0.952      0.998    7
                 0.5       0          1         0.5       0.667      0.75     8
                 0.8       0          1         0.8       0.889      0.9      9
                 1         0          1         1         1          1        10
                 1         0          1         1         1          1        11
                 0.8       0          1         0.8       0.889      0.9      12
                 1         0          1         1         1          1        13
```

	0.9	0	1	0.9	0.947	0.95	14
	0.9	0.003	0.9	0.9	0.9	0.948	15
	1	0	1	1	1	1	16
	1	0	1	1	1	1	17
	1	0.007	0.833	1	0.909	0.997	18
	1	0.021	0.625	1	0.769	0.99	19
	0.9	0	1	0.9	0.947	0.95	20
	1	0	1	1	1	1	21
	1	0.003	0.909	1	0.952	0.998	22
	1	0.007	0.833	1	0.909	0.997	23
	0.9	0	1	0.9	0.947	0.95	24
	1	0	1	1	1	1	25
	1	0.003	0.909	1	0.952	0.998	26
	1	0.003	0.909	1	0.952	0.998	27
	1	0.007	0.833	1	0.909	0.997	28
	1	0	1	1	1	1	29
	1	0	1	1	1	1	30
Weighted Avg.	0.927	0.003	0.943	0.927	0.919	0.962	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	<-- classified as
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = 1
0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	b = 2
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	0	0	0	2	0	0	0	0	0	0	0	c = 3
0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = 4
0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	0	4	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	h = 8
1	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	0	0	8	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	n = 14
0	0	0	0	0	0	1	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	p = 16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	q = 17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	s = 19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0	0	t = 20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	u = 21
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	v = 22
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	w = 23
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	1	0	0	0	0	x = 24
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	y = 25
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	z = 26
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	aa = 27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	ab = 28
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	ac = 29
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	ad = 30

Naive Bayes

```
=== Run information ===

Scheme:weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 1 -R -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.1
Relation:     jAudio
Instances:    300
Attributes:   55

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=55 #classindex=0
Network structure (nodes followed by parents)
class(30):
Spectral Rolloff Point Overall Standard Deviation0(4): class
Spectral Flux Overall Standard Deviation0(4): class
Standard Deviation of Spectral Flux Overall Standard Deviation0(3): class
Compactness Overall Standard Deviation0(2): class
Spectral Variability Overall Standard Deviation0(5): class
MFCC Overall Standard Deviation0(6): class
MFCC Overall Standard Deviation1(6): class
MFCC Overall Standard Deviation2(4): class
MFCC Overall Standard Deviation3(3): class
MFCC Overall Standard Deviation4(3): class
MFCC Overall Standard Deviation5(2): class
MFCC Overall Standard Deviation6(3): class
MFCC Overall Standard Deviation7(2): class
MFCC Overall Standard Deviation8(3): class
MFCC Overall Standard Deviation9(2): class
MFCC Overall Standard Deviation10(2): class
MFCC Overall Standard Deviation11(2): class
MFCC Overall Standard Deviation12(3): class
LPC Overall Standard Deviation0(5): class
LPC Overall Standard Deviation1(6): class
LPC Overall Standard Deviation2(4): class
LPC Overall Standard Deviation3(3): class
LPC Overall Standard Deviation4(4): class
LPC Overall Standard Deviation5(2): class
LPC Overall Standard Deviation6(5): class
LPC Overall Standard Deviation7(3): class
LPC Overall Standard Deviation8(2): class
Spectral Rolloff Point Overall Average0(7): class
Spectral Flux Overall Average0(4): class
```

```
Standard Deviation of Spectral Flux Overall Average0(6): class
Compactness Overall Average0(4): class
Spectral Variability Overall Average0(2): class
MFCC Overall Average0(6): class
MFCC Overall Average1(7): class
MFCC Overall Average2(2): class
MFCC Overall Average3(7): class
MFCC Overall Average4(4): class
MFCC Overall Average5(3): class
MFCC Overall Average6(2): class
MFCC Overall Average7(1): class
MFCC Overall Average8(5): class
MFCC Overall Average9(3): class
MFCC Overall Average10(2): class
MFCC Overall Average11(4): class
MFCC Overall Average12(3): class
LPC Overall Average0(9): class
LPC Overall Average1(3): class
LPC Overall Average2(5): class
LPC Overall Average3(4): class
LPC Overall Average4(2): class
LPC Overall Average5(2): class
LPC Overall Average6(6): class
LPC Overall Average7(4): class
LPC Overall Average8(2): class
LogScore Bayes: -12275.69566057637
LogScore BDeu: -23954.600645748855
LogScore MDL: -22005.750814692656
LogScore ENTROPY: -9260.64887507328
LogScore AIC: -13729.648875073359


Time taken to build model: 0.16 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         266               88.6667 %
Incorrectly Classified Instances        34               11.3333 %
Kappa statistic                          0.8828
Mean absolute error                      0.0077
Root mean squared error                  0.0832
Relative absolute error                 12.0206 %
Root relative squared error             46.3331 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
```

```
                 0.8       0          1          0.8       0.889     1         1
                 0.6       0          1          0.6       0.75      0.992     2
                 0.5       0.01       0.625      0.5       0.556     0.966     3
                 1         0.003      0.909      1         0.952     1         4
                 1         0.007      0.833      1         0.909     0.998     5
                 1         0          1          1         1         1         6
                 0.9       0.021      0.6        0.9       0.72      0.994     7
                 0.9       0.017      0.643      0.9       0.75      0.998     8
                 0.9       0.007      0.818      0.9       0.857     0.987     9
                 0.7       0          1          0.7       0.824     0.997     10
                 1         0          1          1         1         1         11
                 0.9       0          1          0.9       0.947     0.999     12
                 0.6       0.003      0.857      0.6       0.706     0.993     13
                 0.8       0.017      0.615      0.8       0.696     0.993     14
                 0.8       0.01       0.727      0.8       0.762     0.994     15
                 1         0          1          1         1         1         16
                 0.8       0.007      0.8        0.8       0.8       0.982     17
                 0.8       0.003      0.889      0.8       0.842     0.999     18
                 0.9       0.003      0.9        0.9       0.9       0.999     19
                 0.9       0          1          0.9       0.947     1         20
                 0.9       0          1          0.9       0.947     1         21
                 0.9       0          1          0.9       0.947     0.999     22
                 1         0.003      0.909      1         0.952     1         23
                 1         0          1          1         1         1         24
                 1         0          1          1         1         1         25
                 1         0.003      0.909      1         0.952     1         26
                 1         0          1          1         1         1         27
                 1         0          1          1         1         1         28
                 1         0          1          1         1         1         29
                 1         0          1          1         1         1         30
Weighted Avg.    0.887     0.004      0.901      0.887     0.887     0.996

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z aa ab ac ad   <-- classified as
  8  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = 1
  0  6  0  0  0  0  2  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = 2
  0  0  5  1  0  0  1  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  1  0  0  0  0  0  0  0 |  c = 3
  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  d = 4
  0  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  e = 5
  0  0  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = 6
  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 |  g = 7
  0  0  0  0  1  0  0  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  h = 8
  0  0  1  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  i = 9
  0  0  0  0  0  0  1  0  0  7  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  j = 10
  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  k = 11
  0  0  0  0  0  0  0  0  0  0  0  9  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  l = 12
  0  0  0  0  0  0  1  0  0  0  0  0  6  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  m = 13
  0  0  0  0  0  0  0  1  0  0  0  0  1  8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  n = 14
  0  0  0  0  0  0  1  0  0  0  0  0  0  0  8  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 |  o = 15
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  p = 16
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0 |  q = 17
  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0 |  r = 18
  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  0  0 |  s = 19
```

```
 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  0 |  t = 20
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9  0  0  0  0  1  0  0  0  0 |  u = 21
 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0 |  v = 22
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0 |  w = 23
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0 |  x = 24
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0 |  y = 25
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0 |  z = 26
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0 | aa = 27
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0 | ab = 28
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0 | ac = 29
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10 | ad = 30
```

Bayes network

```
=== Run information ===

Scheme:weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 2 -R -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.1
Relation:     jAudio
Instances:    300
Attributes:   55

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=55 #classindex=0
Network structure (nodes followed by parents)
class(30):
Spectral Rolloff Point Overall Standard Deviation0(4): class
Spectral Flux Overall Standard Deviation0(4): class
Standard Deviation of Spectral Flux Overall Standard Deviation0(3): class Standard Deviation of Spectral Flux Overall Average0
Compactness Overall Standard Deviation0(2): class
Spectral Variability Overall Standard Deviation0(5): class Spectral Flux Overall Standard Deviation0
MFCC Overall Standard Deviation0(6): class
MFCC Overall Standard Deviation1(6): class
MFCC Overall Standard Deviation2(4): class
MFCC Overall Standard Deviation3(3): class MFCC Overall Standard Deviation7
MFCC Overall Standard Deviation4(3): class MFCC Overall Standard Deviation3
MFCC Overall Standard Deviation5(2): class MFCC Overall Standard Deviation10
MFCC Overall Standard Deviation6(3): class
MFCC Overall Standard Deviation7(2): class MFCC Overall Standard Deviation10
MFCC Overall Standard Deviation8(3): class
MFCC Overall Standard Deviation9(2): class
MFCC Overall Standard Deviation10(2): class
MFCC Overall Standard Deviation11(2): class MFCC Overall Standard Deviation7
MFCC Overall Standard Deviation12(3): class
LPC Overall Standard Deviation0(5): class
LPC Overall Standard Deviation1(6): class
LPC Overall Standard Deviation2(4): class
LPC Overall Standard Deviation3(3): class LPC Overall Standard Deviation4
LPC Overall Standard Deviation4(4): class
LPC Overall Standard Deviation5(2): class
```

```
LPC Overall Standard Deviation6(5): class LPC Overall Standard Deviation5
LPC Overall Standard Deviation7(3): class LPC Overall Standard Deviation5
LPC Overall Standard Deviation8(2): class LPC Overall Standard Deviation5
Spectral Rolloff Point Overall Average0(7): class
Spectral Flux Overall Average0(4): class Standard Deviation of Spectral Flux Overall Average0
Standard Deviation of Spectral Flux Overall Average0(6): class Spectral Variability Overall Average0
Compactness Overall Average0(4): class
Spectral Variability Overall Average0(2): class MFCC Overall Average0
MFCC Overall Average0(6): class
MFCC Overall Average1(7): class
MFCC Overall Average2(2): class
MFCC Overall Average3(7): class
MFCC Overall Average4(4): class
MFCC Overall Average5(3): class
MFCC Overall Average6(2): class MFCC Overall Average5
MFCC Overall Average7(1): class
MFCC Overall Average8(5): class
MFCC Overall Average9(3): class
MFCC Overall Average10(2): class
MFCC Overall Average11(4): class
MFCC Overall Average12(3): class
LPC Overall Average0(9): class
LPC Overall Average1(3): class LPC Overall Average0
LPC Overall Average2(5): class
LPC Overall Average3(4): class
LPC Overall Average4(2): class MFCC Overall Average1
LPC Overall Average5(2): class Standard Deviation of Spectral Flux Overall Average0
LPC Overall Average6(6): class
LPC Overall Average7(4): class
LPC Overall Average8(2): class LPC Overall Average2
LogScore Bayes: -11944.795141320468
LogScore BDeu: -36674.80515700046
LogScore MDL: -30749.366214917885
LogScore ENTROPY: -9277.477089075057
LogScore AIC: -16806.47708907499


Time taken to build model: 0.46 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         269               89.6667 %
Incorrectly Classified Instances        31               10.3333 %
Kappa statistic                          0.8931
Mean absolute error                      0.007
Root mean squared error                  0.0774
Relative absolute error                 10.8629 %
Root relative squared error             43.1041 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.8       0.003      0.889     0.8       0.842      0.999    1
                 0.7       0          1         0.7       0.824      0.994    2
```

	0.7	0.003	0.875	0.7	0.778	0.978	3
	1	0.003	0.909	1	0.952	1	4
	1	0.01	0.769	1	0.87	0.999	5
	1	0	1	1	1	1	6
	0.9	0.021	0.6	0.9	0.72	0.993	7
	0.8	0.01	0.727	0.8	0.762	0.997	8
	0.9	0.007	0.818	0.9	0.857	0.991	9
	0.7	0	1	0.7	0.824	1	10
	1	0	1	1	1	1	11
	0.8	0.003	0.889	0.8	0.842	0.997	12
	0.6	0.003	0.857	0.6	0.706	0.994	13
	0.8	0.014	0.667	0.8	0.727	0.993	14
	0.8	0.01	0.727	0.8	0.762	0.993	15
	1	0.003	0.909	1	0.952	1	16
	0.7	0.007	0.778	0.7	0.737	0.983	17
	1	0.003	0.909	1	0.952	0.999	18
	1	0	1	1	1	1	19
	0.9	0	1	0.9	0.947	1	20
	0.9	0	1	0.9	0.947	1	21
	0.9	0	1	0.9	0.947	0.997	22
	1	0	1	1	1	1	23
	1	0	1	1	1	1	24
	1	0	1	1	1	1	25
	1	0.003	0.909	1	0.952	1	26
	1	0	1	1	1	1	27
	1	0	1	1	1	1	28
	1	0	1	1	1	1	29
	1	0	1	1	1	1	30
Weighted Avg.	0.897	0.004	0.908	0.897	0.897	0.997	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	<-- classified as
8	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = 1
0	7	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	7	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	c = 3
0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = 4
0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	g = 7
1	0	0	0	1	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	1	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	1	0	0	7	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	k = 11
0	0	0	0	1	0	0	0	0	0	0	8	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	l = 12
0	0	0	0	0	0	1	0	0	0	0	0	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	1	0	0	0	0	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	n = 14
0	0	0	0	0	0	1	0	0	0	0	0	0	0	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	p = 16
0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	q = 17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	s = 19
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	t = 20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	1	0	0	0	0	u = 21
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	v = 22

```
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0 |  w = 23
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0  0 |  x = 24
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0  0 |  y = 25
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0 |  z = 26
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0 | aa = 27
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0 | ab = 28
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0 | ac = 29
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10 | ad = 30
```

Ensemble de florestas de decisão aleatórias

```
=== Run information ===

Scheme:weka.classifiers.meta.RandomCommittee -S 1 -I 10 -W
weka.classifiers.trees.RandomForest -- -I 5 -K 0 -S 1
Relation:     jAudio
Instances:    300
Attributes:   55

Test mode:10-fold cross-validation


=== Classifier model (full training set) ===

All the base classifiers:

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.5033

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.5033

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.4833

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.5333

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.4833

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.5033

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.4933

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.4967
```

```
Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.57

Random forest of 5 trees, each constructed while considering 6 random features.
Out of bag error: 0.4967


Time taken to build model: 0.89 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         277               92.3333 %
Incorrectly Classified Instances        23                7.6667 %
Kappa statistic                          0.9207
Mean absolute error                      0.0304
Root mean squared error                  0.0981
Relative absolute error                 47.1793 %
Root relative squared error             54.6568 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1          0          1         1          1          1        1
                 0.7        0.003      0.875     0.7        0.778      0.993    2
                 0.7        0          1         0.7        0.824      0.98     3
                 1          0.003      0.909     1          0.952      1        4
                 0.9        0.014      0.692     0.9        0.783      0.993    5
                 1          0          1         1          1          1        6
                 0.9        0.01       0.75      0.9        0.818      0.995    7
                 0.8        0          1         0.8        0.889      0.999    8
                 0.8        0          1         0.8        0.889      0.99     9
                 1          0          1         1          1          1        10
                 1          0          1         1          1          1        11
                 0.9        0.003      0.9       0.9        0.9        0.999    12
                 0.8        0.003      0.889     0.8        0.842      0.997    13
                 0.8        0.003      0.889     0.8        0.842      0.991    14
                 0.8        0.01       0.727     0.8        0.762      0.994    15
                 1          0          1         1          1          1        16
                 0.9        0.003      0.9       0.9        0.9        0.999    17
                 1          0.003      0.909     1          0.952      1        18
                 1          0.003      0.909     1          0.952      0.999    19
                 0.9        0          1         0.9        0.947      1        20
                 0.9        0          1         0.9        0.947      1        21
                 0.9        0          1         0.9        0.947      1        22
                 1          0.003      0.909     1          0.952      1        23
                 1          0.003      0.909     1          0.952      1        24
                 1          0          1         1          1          1        25
```

	1	0.007	0.833	1	0.909	1	26
	1	0.003	0.909	1	0.952	1	27
	1	0	1	1	1	1	28
	1	0	1	1	1	1	29
	1	0	1	1	1	1	30
Weighted Avg.	0.923	0.003	0.93	0.923	0.923	0.998	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	<-- classified as
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = 1
0	7	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	b = 2
0	0	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	c = 3
0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = 4
0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	0	2	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	1	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	0	0	9	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	l = 12
0	0	0	0	0	0	1	0	0	0	0	0	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	0	0	0	0	0	1	8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	n = 14
0	0	0	0	0	0	1	0	0	0	0	1	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	p = 16
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	q = 17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	s = 19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	9	0	0	0	0	0	0	0	0	0	0	t = 20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	1	0	0	0	0	u = 21
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	v = 22
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	w = 23
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	x = 24
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	y = 25
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	z = 26
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	aa = 27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	ab = 28
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	ac = 29
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	ad = 30

Apêndice B – Tutorial

É apresentado aqui o conteúdo do tutorial incorporado ao aplicativo, acessível pela opção "?" do menu principal. A navegação é feita de forma linear, além do acesso direto aos itens correspondentes às funcionalidades básicas do aplicativo: Registro, Exploração, Reconhecimento Pontual e Reconhecimento Automático.

Apresentação

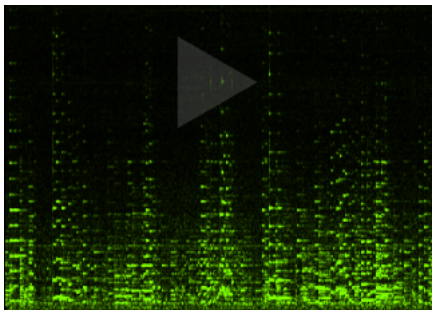
> VSom é um aplicativo desenvolvido para ajudar o indivíduo surdo a reconhecer sons que ocorrem no ambiente. A seguir é apresentado um tutorial com as instruções de uso.

Registro

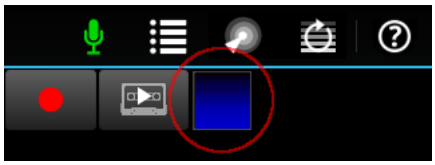
> Nessa tela você pode registrar os seus sons.
>
> Acionando o botão  , o som será detectado e você só precisa informar o nome do som e o ambiente.
>
> O botão  é um atalho para você reproduzir o som que acabou de ser registrado.
>
> São necessários no mínimo 6 registros para que um som seja incluído nas funções de reconhecimento.

O espectrograma é uma representação visual do som.

Nas telas  e  , clicando sobre o espectrograma você visualiza as frequências dos sons do ambiente.



Quando o espectrograma é acionado, um indicador alertará sobre atraso na imagem. Se o indicador estiver azul ou preto significa que o espectrograma está mostrando os sons no tempo certo.



Mas se estiver laranja, a imagem está sendo mostrada com atraso. Se isto acontecer você pode clicar nesse indicador para diminuir a qualidade do espectrograma e evitar o atraso da imagem.

Exploração

Nessa tela você pode explorar os registros, que são agrupados por sons ou por ambientes, conforme a seleção no topo da lista.

Ao clicar sobre um item da lista você verá todos os registros de som correspondentes.

Então você pode reproduzir os sons clicando sobre os seus registros. Porém os sons que não foram registrados por você (texto em azul) não podem ser reproduzidos.

A barra verde à esquerda do nome do som indica que ele está incluído no processo de reconhecimento.

Resumindo, na tela de exploração é possível:

 editar um som ou ambiente.

 excluir um som ou ambiente.

 reproduzir um registro de som.

 excluir um registro de som.

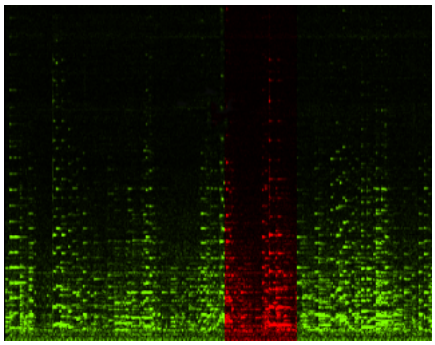
Reconhecimento Pontual

Nessa tela você indica o momento de iniciar o reconhecimento.

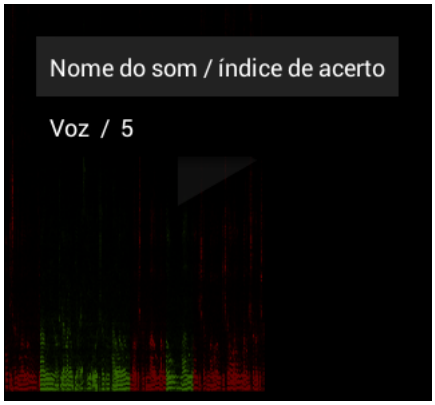
Acionando o botão  , o som será detectado e o resultado do reconhecimento será apresentado.

Se você quiser, pode também informar o ambiente onde se encontra para que o reconhecimento do som seja mais preciso.

Nas telas  e  , o trecho do som capturado é destacado em vermelho no espectrograma.



Ao final do processo o som reconhecido é apresentado e, ao lado, é informado o índice de acerto, sendo 1 o menor grau de acerto e 5 o maior.

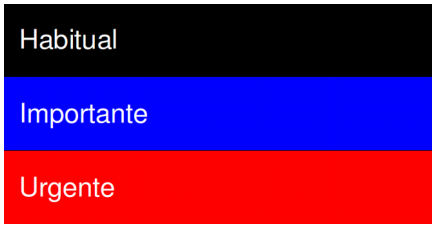


Reconhecimento Automático

Nessa tela os sons reconhecidos vão sendo informados em sequência. Para cada evento sonoro detectado, a lista de resultados informa:

- hora
- nome do som reconhecido
- índice de acerto

Além disso, a importância do som é indicada pela cor do item, como mostrado abaixo.



Apêndice C – Questionário para Validação com Usuários

Dados do participante

Idade: _____ Sexo: [ ] Feminino [ ] Masculino

Grau de surdez: [ ] Leve [ ] Moderada [ ] Severa [ ] Profunda [ ] Total

Nível de estudos: _____ Profissão: _____

Usuário de *smartphone* ou *tablet*: [ ] Sim [ ] Não

Identificação opcional

Nome: _____

E-mail: _____

Questionário

1. Marque abaixo os ambientes onde você sente mais necessidade de ser alertado sobre sons ocorrentes:

[ ] casa	[ ] metrô	[ ] evento esportivo	[ ] rua tranquila
[ ] trabalho	[ ] estação de metrô	[ ] shopping center	[ ] reunião social
[ ] sala de aula	[ ] rodoviária	[ ] loja	[ ] parque
[ ] laboratório	[ ] aeroporto	[ ] restaurante	[ ] centro cultural
[ ] ônibus	[ ] praia	[ ] rua com veículos	[ ] estacionamento
[ ] carro	[ ] piscina	[ ] rua de pedestres	[ ] supermercado

Outros ambientes: _____

_____

2. O aplicativo ajudou você a perceber o que ocorria no ambiente?

[ ] Não [ ] Só uma vez [ ] Sim, algumas vezes [ ] Sim, muitas vezes

3. O aplicativo permitiu você perceber algo no ambiente que seria muito difícil sem o seu uso?

[ ] Não [ ] Só uma vez [ ] Sim, algumas vezes [ ] Sim, muitas vezes

4. Com auxílio do guia visual, avalie as seguintes funções do aplicativo:

	1 - Muito ruim	2 - Ruim	3 - Regular	4 - Boa	5 - Muito boa
registro de amostras	[ ]	[ ]	[ ]	[ ]	[ ]
exploração de sons	[ ]	[ ]	[ ]	[ ]	[ ]
reprodução de amostras	[ ]	[ ]	[ ]	[ ]	[ ]
reconhecimento pontual	[ ]	[ ]	[ ]	[ ]	[ ]
reconhecimento automático	[ ]	[ ]	[ ]	[ ]	[ ]
espectrograma	[ ]	[ ]	[ ]	[ ]	[ ]

5. Indique eventuais problemas que você encontrou no aplicativo:

[ ] difícil de usar
[ ] interface confusa
[ ] processamento lento
[ ] ícones e letras pequenas
[ ] espectrograma de baixa qualidade
[ ] espectrograma não mostra nada interessante
[ ] o aplicativo não fazia o que eu esperava
[ ] o aplicativo reconheceu sons errados
[ ] não consegui registrar amostras de som
[ ] não consegui usar o explorador de sons
[ ] não entendi o que significa o “índice de acerto”

Outros problemas: _____

6. Indique a sua avaliação sobre possíveis melhorias no aplicativo:

	Desnecessária	Útil	Importante
reconhecer sons mais rapidamente	[ ]	[ ]	[ ]
emitir sinal visual mais chamativo ao reconhecer som	[ ]	[ ]	[ ]
emitir sinal vibratório ao reconhecer som	[ ]	[ ]	[ ]
indicar a direção da fonte emissora do som	[ ]	[ ]	[ ]
salvar histórico dos sons reconhecidos	[ ]	[ ]	[ ]
espectrograma com mais qualidade	[ ]	[ ]	[ ]
salvar imagens do espectrograma	[ ]	[ ]	[ ]
mudar as cores do aplicativo	[ ]	[ ]	[ ]

Outras melhorias: _____