UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS

LUCIENE CAVALCANTI RODRIGUES

Uma nova família de filtros digitais para classificação de dados com
aplicações ao pré-diagnóstico de patologias na laringe

São Carlos
2012

LUCIENE CAVALCANTI RODRIGUES

Uma nova família de filtros digitais para classificação de dados com
aplicações ao pré-diagnóstico de patologias na laringe

Tese apresentada ao Programa de Pós-
Graduação em Física do Instituto de
Física de São Carlos da Universidade de
São Paulo, para obtenção do título de
Doutor em Ciências.

Área de concentração: Física Aplicada
Opção: Física Computacional
Orientador: Prof. Dr. Rodrigo Capobianco
Guido.

Versão Corrigida

(Versão original disponível na Unidade que aloja o Programa)

São Carlos
2012

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Rodrigues, Luciene Cavalcanti

Uma nova família de filtros digitais para classificação de dados com aplicações ao pré-diagnóstico de patologias na laringe / Luciene Cavalcanti Rodrigues; orientador Rodrigo Capobianco Guido – versão corrigida -- São Carlos, 2012.

87 p.

Tese (Doutorado - Programa de Pós-Graduação em Física Aplicada Computacional) -- Instituto de Física de São Carlos, Universidade de São Paulo, 2012.

1. Processamento de Sinais. 2. Patologias na Laringe. 3. Classificadores. 4. Filtros Digitais. 5. *Wavelets*. I. Guido, Rodrigo Capobianco, orient. II. Título.

À minha família, com amor, admiração e gratidão por sua compreensão, carinho, presença e incansável apoio ao longo do período de elaboração deste trabalho.

AGRADECIMENTOS

Ao professor, orientador e amigo Rodrigo Capobianco Guido, que, nos anos de convivência, muito me ensinou, contribuindo para meu crescimento científico, intelectual e profissional. Agradeço por toda paciência nos momentos difíceis e pela grande quantidade de horas dispendida em suas reuniões para orientações.

Ao Prof. Dr. Sylvio Barbon Junior, que sempre incentiva e está prontamente disposto a ajudar a todos os integrantes do grupo de pesquisa, compartilhando seus conhecimentos e ainda nos divertindo.

Ao Prof. Dr. Everthon Silva Fonseca por toda ajuda e incentivo no início desta jornada.

Aos demais integrantes do UGA, Lucimar, Leonardo e João Paulo, pela companhia e apoio durante os anos de convivência profissional e pessoal.

À Vanessa Calisto que, na medida do possível, aguentava meu mau humor e me incentivava a sempre continuar trabalhando para alcançar meus objetivos.

A todos os amigos que cooperaram, incentivaram e mandaram energias positivas nos momentos mais difíceis.

Ao Instituto de Física de São Carlos, pela oportunidade de realização do curso de doutorado.

À direção, coordenação e professores da FATEC Rio Preto e do IFSP campus Votuporanga, pelo apoio e incentivo.

À Faculdade de Medicina de Ribeirão Preto (FMRP-USP) e à Dra. Lilian Neto Aguiar Ricz e toda a equipe do Departamento de Fonoaudiologia, pela parceria realizada.

A simplicidade é o último degrau da sabedoria.

Khalil Gibran

RESUMO

RODRIGUES, L. C. **Uma nova família de filtros digitais para classificação de dados com aplicações ao pré-diagnóstico de patologias**. 2012. 87 p. Tese (Doutorado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2012.

O presente trabalho de doutorado tem por objetivo a criação de uma nova família de filtros digitais específica para o processo de classificação de dados, particularmente aplicada ao pré-diagnóstico de patologias na laringe. Antes de explicar a criação dessa nova família de filtros, foi apresentada uma breve revisão bibliográfica sobre o funcionamento do aparelho fonador humano, sobre o processo de diagnóstico de patologias e sobre a transformada discreta *Wavelet*, que serviu de base para a construção dos filtros propostos. Em seguida, é descrita a tecnologia proposta para a criação da nova família de filtros, que é baseada na construção da Transformada *Wavelet* de Daubechies, além disso, apresenta-se uma breve comparação com outras técnicas já descritas na literatura para a mesma finalidade. Posteriormente, são apresentados os resultados obtidos com base na técnica proposta, verificando-se uma taxa de acerto na classificação de vozes normais de 100% e uma taxa de acerto de 95,52% para vozes patológicas.

Palavras-chave: Processamento de sinais. Patologias na laringe. Classificadores. Filtros digitais. *Wavelets*.

ABSTRACT

RODRIGUES, L. C. A new family of digital filters for data classification with applications to the pre-diagnosis of larynx pathologies. 2012. 87 p. Tese (Doutorado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2012.

The main purpose of this thesis is the development of a new family of digital filters used for data classification, particularly applied to the pre-diagnosis of pathologies in the larynx. A brief bibliographical review, that concentrates on the functioning of the human vocal tract, on the process of disease diagnosis, and on the discrete wavelet transform, which formed the basis for the construction of the proposed filters, is presented. The technology used to develop these new families of filters, that is based on the Daubechies' Wavelet Transform, is also described, moreover, a comparison with other techniques described in the specialized literature for the same purpose is also presented. The investigation shows the results obtained with the proposed technique, in which the accuracy of 100% in normal voice classifications and of 95,52% in pathological voice classifications, was obtained.

Keywords: Signal processing. Laryngeal pathology. Classifiers. Digital filters. Wavelets.

LISTA DE FIGURAS

Figura 1 - Cartilagens, ligamentos e músculos da laringe	26
Figura 2 - Nódulo vocal (A – Pregas vocais abertas, B – Pregas vocais fechadas). (1)	30
Figura 3 - Pólipo. (1)	31
Figura 4 - Edema de Reinke. (1).....	31
Figura 5 - Subamostragem do sinal (7)	37
Figura 6 - Alterações propostas para a Daubechies original	40
Figura 7 - Total de vozes disponíveis no banco de dados.	46
Figura 8 – Vozes separadas em classes.	47
Figura 9 - Resultados da Matriz de Confusão.	54
Figura 10 - Comparativo (Resultados do reconhecimento).	55
Figura 11 - Diferenciação entre vozes normais e patológicas.	56
Figura 12 – Porcentagem de acertos do diagnóstico realizado pelo sistema.	56
Figura 13 – Porcentagem de acertos (Normais e Patológicas).	57
Figura 14 - Interface para diagnóstico simples.....	62

LISTA DE TABELAS

Tabela 1 - Fonemas do português brasileiro. (20)	28
Tabela 2 - Pesquisas desenvolvidas na área de Detecção de Patologias na Voz. (25)	42
Tabela 3 - Resultados da classificação em porcentagem. (25)	43
Tabela 4 - Taxas de reconhecimento do sistema	58
Tabela 5 - Resultados Ten Fold Cross Validation	58
Tabela 6 - Taxas de reconhecimento (<i>Ten Fold Cross Validation</i>)	59

LISTA DE ABREVIATURAS E SIGLAS

ACC	<i>Accuracy</i>
AUC	<i>Operating Characteristic Curve</i>
BFS	<i>Backward Feature Selection</i>
BBFS	<i>Branch-and-Bound Feature Selection</i>
DPM	<i>Discriminative Paraconsistent Machine</i>
DWT	<i>Discrete Wavelet Transform</i>
FIR	<i>Finite Impulse Response</i>
FFS	<i>Forward Feature Selection</i>
FN	Taxa de Falsos Negativos
FP	Taxa de Falso Positivo
IFS	<i>Individual Feature Selection</i>
LDA	<i>Linear Discriminant Analysis</i>
LP	<i>Linear Prediction</i>
LS-SVM	<i>Least Squares Support Vector Machines</i>
MDVP	<i>Multi-Dimensional Voice Program</i>
MEEI VddB	<i>Massachusetts Eye & Ear Infirmary Voice Disorders Database</i>
MFCC	<i>Mel Frequency Cepstral Coefficients</i>
MLP	<i>Multi Layer Perceptron</i>
PCA	<i>Principal Component Analysis</i>
SVM	<i>Support Vector Machine</i>

TP	Taxa positiva ou Sensibilidade Especificidade
TN	Taxa negativa

SUMÁRIO

Capítulo 1

INTRODUÇÃO	23
1.1 Motivação	23
1.2 Objetivos específicos e contribuições.....	24
1.3 Organização do trabalho.....	24

Capítulo 2

REVISÃO BIBLIOGRÁFICA	25
2.1 Estrutura biológica do aparelho fonador	25
2.2 Perturbação da Voz (Disfonia)	29
2.3 Avaliação da Voz	31
2.4 Tecnologias tradicionais para detecção de patologias na laringe (Avaliação acústica).....	32
2.5 A Transformada <i>Wavelet</i>	35
2.5.1 - A Transformada <i>Wavelet</i> Discreta (DWT).....	36
2.5.2 - A transformada <i>Wavelet</i> de Daubechies.....	37
2.6 Trabalhos Anteriores Desenvolvidos para Detecção de Patologias na Laringe utilizando métodos não invasivos.....	40

Capítulo 3

A TÉCNICA PROPOSTA E OS RESULTADOS OBTIDOS.....	45
3.1 A base de dados necessária para o desenvolvimento do projeto	45
3.2 Metodologia proposta.....	47
3.3 Testes e Resultados.....	53

Capítulo 4

CONCLUSÕES E TRABALHOS FUTUROS.....	61
REFERÊNCIAS	63
APÊNDICE	67

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo, a motivação, os objetivos específicos e a organização do trabalho são apresentados e explicados.

1.1 Motivação

Visto que a fala representa um dos meios de comunicação mais importantes utilizados pelo homem, a grande motivação deste trabalho é o aperfeiçoamento de técnicas não invasivas para o pré-diagnóstico computacional de vozes normais e de vozes provenientes de indivíduos com patologias no trato vocal, de forma a contribuir para o bem-estar dos seres humanos, tendo em vista que a qualidade da voz é um fator muito relevante.

A inovação proposta neste projeto consiste na criação de famílias de filtros digitais otimizadas para reconhecer duas classes de vozes sob análise, a saber, normal e patológicas (oriundas de indivíduos com patologias no trato vocal - edemas e nódulos). A aplicação da técnica proposta visa realizar um pré-diagnóstico da situação do aparelho vocal de determinado locutor, auxiliando profissionais da saúde em seus laudos.

A técnica descrita neste trabalho produziu um sistema de classificação que proporcionou uma taxa de reconhecimento correto de 100% das vozes normais testadas e de 95,52% das vozes patológicas. Outros trabalhos voltados para a classificação de patologias na laringe demonstram uma taxa de reconhecimento correto acima de 85% mas utilizam técnicas e bases de dados diferentes, tornando a comparação inviável.

1.2 Objetivos específicos e contribuições

O objetivo principal deste trabalho foi o desenvolvimento de novas famílias de filtros digitais, particularmente otimizadas para a classificação da situação do aparelho vocal de um locutor qualquer. Particularmente, o sistema de apoio à decisão médica elaborado é utilizado para trazer informações quanto à presença ou à ausência de patologias na laringe. Objetivou-se, com base em um banco de vozes previamente laudado por profissionais da área médica, criar famílias de filtros digitais que reconheçam, sem a necessidade de um outro classificador associado, tal como uma rede neural, vozes normais e vozes de indivíduos com patologias no trato vocal.

1.3 Organização do trabalho

Tendo em vista os objetivos previamente detalhados, este trabalho está organizado da seguinte forma. O Capítulo 2 contém uma revisão da literatura, descrevendo os conceitos teóricos sobre patologias na laringe, características acústicas da voz, processos de classificação tradicionais, filtros *wavelets*, entre outros tópicos. No Capítulo 3 há uma descrição detalhada do projeto de tais filtros e da tecnologia utilizada. O Capítulo 4 apresenta a base de dados usada durante a fase de projeto e de testes dos filtros, além de resultados e discussões diversas. No Capítulo 5 estão presentes os comentários finais, que levam às conclusões. Ao final do trabalho, tem-se as referências utilizadas na pesquisa.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados os conceitos necessários ao desenvolvimento do presente projeto.

2.1 Estrutura biológica do aparelho fonador

De acordo com Alcaim (20), os sons de fala podem ser definidos fisicamente como ondas de pressão, resultantes de vibrações longitudinais das moléculas do ar, provocadas por órgãos do corpo humano destinados às funções vitais de comer, beber e respirar.

Segundo Ferreira (1), “A voz é um som audível e representa uma característica particular de cada indivíduo, podendo ser definida como a inter-relação de fatores orgânicos, psicológicos e sociais. É resultante da atividade laríngea, com inter-relação entre pressão e velocidade do fluxo de ar expiratório (que influenciam a intensidade), os diferentes padrões de adução e abdução das pregas vocais (que influenciam a sonoridade) e as propriedades de reflexão e configuração das estruturas do trato vocal (que influenciam na ressonância).”

A produção da voz é resultado da fonação acrescida de ressonância, sendo o som produzido pela vibração das pregas vocais. Observa-se que a produção da voz depende do sistema respiratório, do sistema fonador, do sistema de ressonância e do trato vocal. Sabe-se que o sistema respiratório é essencial para a eliminação do dióxido de carbono e para a absorção do oxigênio, além de produzir a pressão necessária para manter a vibração das pregas vocais. O sistema fonador possui três funções biológicas principais: impede os corpos estranhos de entrarem nas vias respiratórias, permite bloquear o ar dos pulmões e modifica a resistência das vias respiratórias.

O sistema de ressonância, formado pela laringe, faringe, boca e nariz, é o principal responsável pela qualidade, riqueza e timbre da voz, e a modificação do som sobre a ação de estruturas como a língua, mandíbula e palato mole (1).

O trato vocal supralaríngeo é uma estrutura importante para a produção de sons da fala, pois permite modificar o som laríngeo (vogais ou consoantes); as vogais, particularmente, são produzidas sem obstrução à passagem do ar pelo trato vocal, sendo criadas pela modificação da configuração do trato vocal, que age como um equalizador e que possui frequências de ressonância, chamadas de formantes, que atravessam o trato vocal com mais eficácia, sem atenuação (1-2).

A laringe possui músculos, que são os intrínsecos, relacionados diretamente à produção de voz, de contração rápida e altamente resistentes à fadiga (os músculos que têm origem e inserção na laringe) e os extrínsecos, que possuem participação indireta na fonação, proporcionando estabilidade à laringe nas estruturas do pescoço (os músculos com inserção na laringe e em estruturas externas) (3).

De acordo com Barbon Junior (7), a produção de voz pelo corpo humano, embora seja um mecanismo repleto de detalhes, consiste basicamente na propulsão de ar pelos pulmões, seguida de um processo de equalização realizado pelo trato vocal e elementos associados, como ilustrado na Figura 1.

Figura 1 - Cartilagens, ligamentos e músculos da laringe.
(<http://www.viaaereadifical.com.br/anatomia/anatomia.htm>)

Observa-se que as pregas vocais controlam o fluxo de ar fornecido pelos pulmões, fazendo com que esse sinal de excitação seja periódico (ou também chamado de período de *pitch*, classificando a voz como vozeada), vibrando em determinada frequência, ou aperiódico

(classificando a voz como não vozeada), similar a um sinal ruidoso. No primeiro caso encontram-se basicamente as vogais, enquanto no segundo caso estão os demais sons.

Qualquer alteração estrutural ou funcional nos elementos da figura 1, possivelmente causada por uma patologia, implica a obtenção de um sinal de voz diferente do normal. Quando essa alteração é substancial, o efeito na voz do locutor pode ser percebido mais facilmente por qualquer pessoa; além disso, o profissional da área médica é capaz de arriscar um pré-diagnóstico sem uma análise clínica detalhada. Já no caso da alteração não ser significativa, o seu impacto na voz do locutor pode não ser facilmente percebido pelo próprio locutor, por outra pessoa qualquer que convive com ele, ou até por um profissional da área médica. Para tais casos, a análise do sinal de voz por um programa de computador é aconselhável, podendo ser utilizada como uma técnica eficaz para alertar o locutor sobre a saúde de sua estrutura vocal, indicando que ele deverá procurar um profissional especializado para realizar uma análise mais detalhada.

Segundo Alcaim (20), dependendo dos sons da fala a serem gerados, três mecanismos básicos de excitação podem estar envolvidos:

- Sons sonoros: o fluxo de ar nos pulmões provoca a vibração das cordas vocais, gerando uma sequência de pulsos quase periódicos de pressão de ar, que atuam como fonte de excitação para o aparelho vocal.
- Sons fricativos surdos: é criada uma turbulência em constrições estreitas situadas no aparelho vocal, produzindo uma fonte de ruído contínuo com espectro relativamente amplo e uniforme.
- Sons oclusivos: consistem em um súbito desprendimento de excesso de pressão após uma obstrução completa em algum ponto no aparelho vocal.

Segundo Alcaim (20), a linguagem humana compreende um número de elementos básicos de som, denominados fonemas, caracterizados pelo fato de que duas palavras diferem apenas quando pelo menos um de seus elementos básicos diferirem. A Tabela 1 abaixo apresenta os fonemas do português brasileiro:

Tabela 1 - Fonemas do português brasileiro. (20)

Tipos de som	Exemplos
Vogais	/i/- vi, /ε/- vela, /e/- vê, /a/- vala, /u/- uva, /ô/- bola, /o/- bobo
Fricativas Sonoras	/v/- chuva, /z/- zelo, /ž/- gelo
Fricativas Surdas	/f/- faca, /s/- sala, /š/- chuva
Oclusivas Sonoras	/b/- bato, /d/- dedo, /g/- gola
Oclusivas Surdas	/p/- pato, /t/- tatu, /k/- capa
Nasais	/m/- mala, /n/- nada, /ñ/- manha
Laterais	/l/- ca/a, /l̃/- calha
Vibrantes	/r/- cara, /R̃/- carro

Segundo Alcaim (20), dependendo da caracterização acústica, os sons da fala podem ser classificados em:

- Vogais: sons sonoros, resultado de vibrações das cordas vocais e relativamente longos;
- Nasais: sons sonoros que se caracterizam pela saída de ar pela boca e pelas fossas nasais;
- Fricativos: podem ser sonoros (como /v/ de chuva) ou surdos (como /f/ de faca), dependendo da presença ou ausência de vibração das cordas vocais;
- Oclusivos: podem ser sonoros (como /b/ de bola) ou surdos (como /p/ de pato), caracterizados por um excesso de pressão criado em um ponto no aparelho vocal, seguido de um desprendimento repentino de ar.

De acordo com Alves (22), a voz é considerada adaptada quando sua qualidade vocal é aceitável socialmente, sem interferências na inteligibilidade da fala, permitindo o desenvolvimento profissional do indivíduo. A frequência, intensidade, modulação e projeção são apropriadas ao gênero e à idade do falante, transmitindo a mensagem emocional do discurso. Em relação aos diversos tipos vocais, a voz adaptada ou normal, é conceituada como a que não apresenta nenhuma alteração no seu som, tendo uma qualidade precisa e agradável, e a voz desviada ou alterada é conceituada como aquela que auditivamente apresenta sinais de funcionalidade inadequada das pregas vocais ou sinais de desequilíbrio do trato vocal.

O sinal de voz humano pode ser considerado um sinal acústico quase-periódico, visto que os ciclos vibratórios podem ser similares mas não idênticos. Essas variações próprias do som fundamental podem ocorrer como perturbações na frequência (*jitter*) ou perturbações na amplitude (*shimmer*). (2-3,17)

A laringe é responsável por várias atividades fisiológicas e, como os demais órgãos, sofre transformações fisiológicas ao longo da vida, que são facilmente perceptíveis na voz. Denomina-se voz saudável aquela que não apresenta alterações e cuja qualidade é aceitável socialmente, não interferindo na inteligibilidade da fala e que apresenta frequência, intensidade, modulação e projeção apropriadas para o gênero e idade do falante, permitindo seu desenvolvimento profissional, excetuando-se a voz profissional, que considera fatores culturais, modismos e estilos das diferentes profissões que usam a voz como ferramenta de trabalho. (3)

2.2 Perturbação da voz (Disfonia)

Quando se fala de uma alteração da voz, independentemente de sua etiologia, geralmente refere-se a uma disfonia, ou seja, um distúrbio da comunicação oral, no qual a voz não consegue cumprir seu papel básico de transmissão da mensagem verbal e emocional do falante (3). A perturbação da voz é caracterizada por dificuldade ou alteração persistente na emissão vocal que impede a produção natural da voz, sendo uma condição onde o funcionamento da voz é inadequado para o uso familiar, social e profissional. (1)

De acordo com Ferreira (1), existem vários aspectos que devem ser levados em consideração para a classificação da normalidade da voz, visto que a voz pode variar episodicamente (face ao “stress”, ruído ambiente ou alterações hormonais) ou progressivamente (face à influência do tabaco, álcool ou cafeína), variando quanto à altura tonal, à sensação de intensidade e/ou quando a qualidade vocal é desagradável ou inadequada para a idade e sexo do indivíduo, do ponto de vista social e/ou familiar, ou quando o indivíduo percebe desconforto ou dor ao falar. A aceitação das variações vocais dentro do limite de normalidade depende de fatores culturais, do grau de impacto da qualidade vocal e da duração dos sintomas.

Existem vários tipos de lesões que podem acometer o trato vocal, em especial, as lesões de massa localizadas e alterações nos tecidos da prega vocal, onde se destacam os nódulos e edemas, que são objeto deste estudo.

As lesões de massa localizadas afetam a zona membranosa das pregas vocais, a glote e o espaço ventricular, podendo ser lesões uni ou bilaterais, assim como benignas, pré-malignas ou malignas. Como exemplo dessas lesões, pode-se listar nódulos, pólipos, quistos, papilomas, granulomas, úlceras de contato, hiperplasia epitelial (leucoplasia, hiperqueratose, queratose e acantose) e neoplasias (1).

As lesões benignas mais comuns das pregas vocais são os nódulos (Figura 3), que são lesões sólidas, grosseiramente simétricas e que se desenvolvem ao nível do ponto nodular (junção do terço anterior e médio das pregas vocais) e resultam normalmente de hiperfuncionamento vocal. Sua gravidade depende da dimensão dos nódulos, do grau de lesão e da forma de compensação usada pelo indivíduo. Normalmente desencadeiam timbre rouco, soproso, perda de amplitude e fadiga vocal (1-2).

Figura 2 - Nódulo vocal (A – Pregas vocais abertas, B – Pregas vocais fechadas). (1)

Por sua vez, os pólipos (Figura 3) são lesões geralmente unilaterais e frequentemente com componente edematoso e angiomatoso. São lesões que ocorrem mais frequentemente no sexo masculino, desencadeando timbre rouco, diplofônico, perda de amplitude e fadiga vocal. (1)

Figura 3 - Pólipo. (1)

Como exemplos das alterações nos tecidos pode-se observar o edema de Reinke, (exemplificado na Figura 4), a laringite, o sulco glótico, a verruga nas pregas vocais, o trauma e a hemorragia (1). Este estudo concentrou-se também no edema de Reinke, que se caracteriza por uma acumulação de fluido gelatinoso, mucóide, na camada superficial da lâmina própria, geralmente estendendo-se ao longo de toda a prega vocal membranosa. Resulta numa voz grave, timbre rouco e áspero (1).

Figura 4 - Edema de Reinke. (1)

2.3 Avaliação da voz

A avaliação da voz é um processo clínico, geralmente multidisciplinar, com uma série de instâncias e procedimentos que devem ser considerados. Esse processo serve para compreender o mecanismo normal e fisiopatológico da produção de voz, a significância de

um transtorno vocal e planejar as medidas terapêuticas necessárias para restabelecer a organização funcional do sistema de produção de voz. (3)

De acordo com Dajer (3), pode-se dividir a avaliação dos sinais de voz em 5 dimensões:

- 1) análise perceptivo-auditiva (escalas analógicas e numéricas);
- 2) autoavaliação da voz e seu impacto (medir o grau do desvio e como interfere);
- 3) dados aerodinâmicos e acústicos (medidas fonatórias e medidas acústicas);
- 4) perfil da extensão vocal e
- 5) análise visual da laringe.

O sistema de produção da voz é a forma mais elaborada de comunicação humana, por isso dificilmente é utilizado um método único de avaliação que seja abrangente e preciso. A avaliação da voz humana implica uma complexa combinação de dados que permitem a construção de um quadro conceitual que possibilita e/ou suporta a explicação do “como” e “porquê” da perturbação vocal apresentada (3).

A literatura descreve uma grande variedade de técnicas para avaliação vocal, sendo essas observacionais e instrumentais, nas quais, em contexto clínico, a avaliação da voz envolve a recolha de informações através de entrevista (história clínica/anamnese); avaliação da fisiologia laríngea (laringoscopia indireta e/ou endoscopia e/ou estroboscopia e/ou electroglotografia); avaliação áudio-perceptual; exame funcional (avaliação musculoesquelética e aerodinâmica); análise acústica e auto-avaliação do impacto psicossocial da voz (3).

2.4 Tecnologias tradicionais para detecção de patologias na laringe (Avaliação acústica)

Segundo Dajer (3), a avaliação perceptivo-auditiva é uma das ferramentas mais usadas na clínica vocal, principalmente pelo fato de que a voz é um fenômeno perceptivo em resposta a um estímulo acústico, e tem como objetivo definir as características vocais, avaliando-a

qualitativamente em relação a possíveis transtornos ou desvios vocais. Os termos utilizados na maioria das escalas de avaliação perceptivo-auditiva para descrever desvios na qualidade vocal são: rouquidão, aspereza e soprosidade. Esse tipo de análise faz parte da prática profissional dos fonoaudiólogos e é objeto de estudo e aprimoramento, e por sua característica de subjetividade ou grau de confiabilidade tem sido questionado por diversos pesquisadores.

Por definição de Dajer (3), rouquidão é uma irregularidade vibratória da mucosa das pregas vocais durante a fonação e provoca ruídos de baixa frequência. Essa característica pode ser encontrada em patologias como nódulos, edemas e hiperemias. O termo aspereza é empregado para descrever uma característica vocal associada à rigidez de mucosa, provocada por certa irregularidade vibratória dependendo da presença ou não de fenda glótica e de associações com outras alterações laríngeas, cuja característica perceptiva apresenta ruídos de alta *frequência* na análise acústica e espectral devido ao grande vazamento de ar provocado pela vibração de uma mucosa enrijecida, característica predominante nos casos de sulco vocal.

Visando o diagnóstico de doenças da laringe, os médicos usualmente realizam exames invasivos como a videolaringoscopia e a videoestroboscopia (4). Dessa forma, muitos trabalhos têm sido desenvolvidos no intuito de realizar uma classificação de patologias utilizando-se soluções computacionais baseadas em *software*, que são menos invasivas e servem como ferramenta para segunda opinião médica (17).

Os métodos tradicionais invasivos, além de gerar desconforto, apresentam um problema relacionado ao reflexo na cavidade supraglotal, que pode levar a avaliações equivocadas sobre as condições da laringe do paciente. Assim, além dos exames invasivos e da avaliação perceptual acústica, a utilização de outras tecnologias pode auxiliar cada vez mais no diagnóstico preciso e simplificado das doenças vocais (2,17).

A avaliação acústica utiliza métodos não invasivos e tem a vantagem de propiciar medidas objetivas, ou seja, dados quantitativos extraídos automaticamente, por meio de um processamento computacional, sendo possível verificar características do aspecto vocal, além de garantir a análise, extração e quantificação de padrões do sinal vocal (3,5). Esse tipo de avaliação amplia as possibilidades de diagnóstico pois aumenta a utilidade de instrumentos para detecção precoce de alterações laríngeas; possibilita o acompanhamento terapêutico ao longo do tratamento; possui uma compreensão acústica da saída vocal mais clara e, ainda, permite uma melhor associação entre as análises perceptivo-auditiva e acústica (6).

Segundo Dajer (3), a implementação da análise acústica no processamento de sinais de voz permitiu definir os parâmetros e medidas acústicas usadas na atualidade na clínica de voz e os principais parâmetros utilizados são:

- *Medidas de frequência fundamental ou frequência glótica*: é o número de ciclos vibratórios completos da mucosa das pregas vocais por segundo, em que os valores normais médios são de 80 a 150Hz com média de 113Hz para homens; de 150 a 250Hz com média de 204Hz para mulheres, e média de 235,8Hz para crianças.

- *Medidas de perturbação*: a vibração das pregas vocais produz um pulso de ar glótico quase-periódico; esta quase-periodicidade, embora normal no funcionamento da laringe, está limitada a certos valores. As medidas de perturbação de um sinal referem-se a quanto um determinado período de vibração se diferencia do outro que o sucede, e por meio delas podem-se discriminar os casos patológicos dos normais. Várias medidas na perturbação de *frequência* e amplitude foram propostas na literatura. De forma genérica, o *jitter* representa a perturbação do período / frequência, e o *shimmer* representa a perturbação na amplitude do sinal.

- *Medidas de ruído*: avaliam o ruído em diferentes faixas de *frequência* do espectro. O ruído é um sinal acústico sem periodicidade, originado da superposição de vários movimentos de vibração com diferentes *frequências* as quais não apresentam relação entre si.

A avaliação acústica clínica tem sido cada vez mais presente na realidade fonoaudiológica e tornou-se uma ferramenta valiosa na clínica, pois apoiam as observações subjetivas nas imagens de videoestroboscopia e podem ser utilizadas para detectar níveis de perturbação fonatória acima dos valores de referência.

Segundo Alcaim (20), as sequências de amostras da forma de onda de um sinal de voz podem ser consideradas como funções amostra de um processo aleatório real, discreto no tempo. Algumas características do sinal de voz, que são de fundamental importância para inúmeras técnicas de codificação digital de interesse, além de outras aplicações em processamento de voz, são obtidas a partir de sua resposta no domínio da frequência.

Segundo Barbon Junior (2), a verificação da perturbação no domínio da frequência possibilita o uso de outro conjunto de informações, diferentes das obtidas no domínio do tempo, onde estão as informações utilizadas neste estudo. De acordo com Alcaim (20), as principais técnicas de codificação no domínio da frequência são a codificação em sub-bandas

e a codificação por transformada. Ambas utilizam parte dos recursos binários para representar uma informação paralela, que é empregada para o cálculo dos parâmetros de adaptação dos quantizadores e do número de bits alocados para a codificação da informação principal.

Para a interpretação adequada dos sinais de voz, este estudo propõe a análise do sinal indireto a partir de uma transformação de base utilizando-se filtros digitais especialmente criados. O projeto de tais filtros baseou-se na Transformada *Wavelet* Discreta, a qual, assim, é revisada a seguir.

2.5 A transformada *Wavelet*

Segundo Araújo (19), é usual a medição de grandezas que variam continuamente no tempo, como, por exemplo, a pressão arterial, fluxo de fluído por um duto, variação de temperatura em uma superfície, sinal acústico da voz, brilho de uma estrela, entre outros. Todas essas grandezas são ditas funções do tempo, uma vez que variam com ele, e são matematicamente representadas por $x(t)$ para denotar a sua relação intrínseca com a variável t , que representa o tempo. Todas essas variáveis exemplificadas anteriormente podem ser ditas contínuas em relação ao tempo. Por razões técnicas, ao invés do sinal $x(t)$ usualmente armazenam-se as mostras $x[n]$ desse sinal, igualmente espaçadas no tempo. Às vezes também se representam tais amostras por $x[n]$, remetendo ao fato de que cada amostra possui o valor da função (ou sinal) no instante nT , ou seja, $x[n] = x(nT)$. O sinal amostrado será uma boa representação se satisfizer às condições do Teorema da Amostragem, não criando assim um falseamento do sinal (*aliasing*). E ainda mais, quando são tomadas, essas amostras não são armazenadas com precisão infinita, mas são arredondadas de forma a serem representadas por alguns dígitos. Esse processo é chamado quantização. Uma das ferramentas básicas utilizadas para processar essa sequência de amostras, que representam o sinal, são os filtros digitais. (19)

2.5.1 - A Transformada *Wavelet* Discreta (DWT)

Na opinião de Araújo (19), as *wavelets* surgem como uma nova forma de ver o mundo, já que as representações são feitas em diferentes escalas, tornando possível arbitrar dentre diversas formas de representação no domínio tempo-frequência. Isso propicia diferentes níveis de detalhes em cada representação, pois as informações estão contidas em diferentes bandas de frequências. Assim, a análise em resoluções múltiplas é uma forma de seccionar o espaço em diversas faixas de frequências não sobrepostas.

A Transformada *Wavelet* Discreta (DWT) (8-9,15) é, portanto, uma ferramenta para análise tempo-frequência de um sinal, sendo, notadamente, mais eficiente do que a Transformada de Fourier (10-11). A DWT trabalha com a convolução do sinal discreto $x[\cdot]$ com um par de filtros (passa-baixas $h[\cdot]$) e passa-altas ($g[\cdot]$), gerando-se um novo nível com a metade da faixa de frequência do sinal original a cada vez que o sinal é submetido ao processo, sendo que apenas o novo sinal obtido pela aplicação do filtro passa-baixas é usado para continuar, recursivamente, a decomposição. (4,7,13-14)

O processo de conversão de domínio de $x[\cdot]$ consiste em filtrá-lo, via convolução, utilizando os filtros $h[\cdot]$ e $g[\cdot]$. Esse processo origina dois outros sinais, $x_{low}[\cdot]$ e $x_{high}[\cdot]$, respectivamente. (9) A concatenação de $x_{low}[\cdot]$ com $x_{high}[\cdot]$, nessa ordem, forma um novo sinal de comprimento igual ao de $x[\cdot]$, que corresponde a sua DWT. Cada vez que tal processo é aplicado, diz-se que um nível de decomposição ocorreu. Os sinais $x_{low}[\cdot]$ e $x_{high}[\cdot]$ são sempre sub-amostrados por 2 devido ao fato de que eles contêm apenas metade da largura de banda do sinal original, de acordo com o Teorema da Amostragem (Figura 5) (7). O processo conhecido como *wrap-around* (9) é utilizado para manter o comprimento do sinal transformado, já que, sem ele, certamente a DWT de $x[\cdot]$ teria comprimento superior ao de $x[\cdot]$, tendo em vista as propriedades da convolução.

Figura 5 – Subamostragem do sinal (7)

Quanto maior o suporte (tamanho) dos filtros utilizados e, portanto, melhor a resposta em frequência dos mesmos, as informações sobre quais faixas de frequências estão em determinadas sub-bandas são mais exatas. Por outro lado, quanto menor for o suporte dos filtros, mais precisas serão as informações sobre as localizações temporais. Esse fato faz com que, em algumas aplicações, seja necessário balancear os suportes, para obter informações razoavelmente precisas tanto no tempo quanto na frequência.

Existem diversas famílias de filtros que se diferenciam pelo suporte dos filtros e pelas características de resposta em frequência e fase dos mesmos, fazendo com que as funções de escala e *wavelet* também sejam alteradas. Podem-se destacar as seguintes famílias de *wavelets*: Haar, Daubechies, Symmlets, Coiflets, Beylkin, Vaidnyanathan, entre outras.

2.5.2 A transformada *Wavelet* de Daubechies

De acordo com Araújo (19), Ingrid Daubechies foi precursora na construção de uma família de *wavelets* ortogonais com suporte compacto e um número específico de momentos nulos. A construção de tais *wavelets* começa com o cálculo dos coeficientes dos filtros passa-baixas e passa-altas associados de tal forma que condições de ortogonalidade e momento sejam satisfeitas. As *wavelets* de Daubecheis possuem o número máximo de momentos nulos (N) para um dado tamanho de suporte (2N).

Segundo RAFIEE, PIFER, CAMPOS, S. e CAMPOS, A. (15,18), as *wavelets* ortogonais criadas por Ingrid Daubechies no início dos anos 80 foram as primeiras a tornar prática a realização da análise discreta. Construídas com base no projeto de filtros ortogonais, com máximo de relação na resposta em frequência em 0 e metade da taxa de amostragem, impondo como restrição o número de momentos de decaimento (que definem a banda de transição) em uma determinada faixa, proporcionaram uma característica conhecida como *maximally flat*, obtendo-se assim uma melhor resolução no domínio tempo-frequência.

As funções Daubechies têm sido utilizadas em diversas áreas, na maioria das vezes com suportes entre 4 e 20. Sinais biológicos normalmente armazenam pouca energia nas faixas de altas frequências, sendo melhor analisados pelos filtros de Daubechies (15,18).

Uma das vantagens da ortogonalidade dos filtros de Daubechies é que os coeficientes das transformadas tornam-se fáceis de serem calculados, pois isso pode ser feito com base em uma simples operação de produto interno, sendo o cálculo de cada coeficiente independente dos demais. A propriedade mais importante é que as bases ortogonais permitem-nos reconstruir um sinal com o conhecimento apenas dos coeficientes dados pelo produto interno do sinal com as funções da base. (19)

Segundo Araújo (19), a criação de espaços ortonormais tem por objetivo definir formas que sejam eficientes para representar um sinal e que sejam úteis ao ser utilizadas no processamento desses sinais e na extração de informação dos mesmos. Quando se representa um sinal em uma base, o que se faz é a projeção deste em um subespaço.

Os três grupos de equações abaixo são reunidos de modo a formar um sistema nas incógnitas $h_0, \dots, h_{k-1}$, o que corresponde à definição da transformada *wavelet* de Daubechies,

$$\sum_{t=0}^{k-1} (-1)^t h_t t^b = 0 \quad , \quad \sum_{t=0}^{k-1} h_t = 2 \quad , \quad \sum_{t=0}^{k-1} h_{t+2l} = 2\delta_{0,l} \quad ,$$

$$\sum_{t=0}^{k-1} (-1)^t h_t t^b = 0, \quad \textbf{Momento}$$

$$\sum_{t=0}^{k-1} h_t = 2,$$

$$\sum_{t=0}^{k-1} h_t \cdot h_{t+2l} = 2 \delta_{0,l}, \quad \textbf{Ortogonalidade}$$

onde $b = 0, 1, \dots, \frac{k}{2} - 1$ e $l \in \mathbb{Z}$. O primeiro grupo é o responsável por fazer com que a função *wavelet* tenha $\frac{k}{2}$ momentos nulos. O segundo grupo, constituído por uma única equação, cuida da normalização dos coeficientes e vem do fato de que para a definição das funções *scaling* de um dado formato, a área correspondente embaixo de cada curso deve ser unitária. O terceiro grupo cuida da ortogonalidade entre os filtros.

Neste trabalho, o processo pelo qual os filtros de Daubechies foram construídos sofre uma alteração, fazendo com que eles se tornem filtros digitais otimizados para a detecção das patologias retromencionadas, conforme Figura 6.

$$\begin{aligned}
& \text{Eq1} = \text{Energia}(\text{vetor_características}) - \text{Energia}(\text{DWT}(\text{vetor_características})) = 1 \\
& \text{Eq2} = h_0 \cdot h_2 + h_1 \cdot h_3 + h_2 \cdot h_4 + h_3 \cdot h_5 + h_4 \cdot h_6 + h_5 \cdot h_7 = 0 \\
& \text{Eq3} = h_0 \cdot h_4 + h_1 \cdot h_5 + h_2 \cdot h_6 + h_3 \cdot h_7 = 0 \\
& \text{Eq4} = h_0 \cdot h_6 + h_1 \cdot h_7 = 0 \\
& \text{Eq5} = h_0 - h_1 + h_2 - h_3 + h_4 - h_5 + h_6 - h_7 = 0 \\
& \text{Eq6} = -h_1 + 2 \cdot h_2 - 3 \cdot h_3 + 4 \cdot h_4 - 5 \cdot h_5 + 6 \cdot h_6 - 7 \cdot h_7 = 0 \\
& \text{Eq7} = -h_1 + 4 \cdot h_2 - 9 \cdot h_3 + 16 \cdot h_4 - 25 \cdot h_5 + 36 \cdot h_6 - 49 \cdot h_7 = 0 \\
& \text{Eq8} = -h_1 + 8 \cdot h_2 - 27 \cdot h_3 + 64 \cdot h_4 - 125 \cdot h_5 + 216 \cdot h_6 - 343 \cdot h_7 = 0
\end{aligned}$$

~~$\sum_{t=0}^{k-1} h_t = 2$~~
 $\sum_{t=0}^{k-1} h_t \cdot h_{t+2l} = 2 \delta_{0,l}$
 $\sum_{t=0}^{k-1} (-1)^t h_t t^b = 0$

Figura 6 - Alterações propostas para a Daubechies original

2.6 Trabalhos Anteriores Desenvolvidos para Detecção de Patologias na Laringe utilizando métodos não invasivos

No trabalho de Barbon Junior (2), nota-se que a principal contribuição foi o desenvolvimento de um algoritmo que associa uma técnica baseada em aprendizado de máquina com lógica paraconsistente anotada, promovendo uma classificação adequada para distinção de vozes normais e patológicas com um reconhecimento de mais de 80% (nódulos nas pregas vocais e Edema de Reinke). A lógica utilizada no trabalho de (2) foi eficiente justamente por possibilitar a detecção e efetiva avaliação da qualidade dos parâmetros, e porque a análise da localização da saída da DPM (*Discriminative Paraconsistent Machine*) no plano paraconsistente possibilitou uma indicação adicional àquela obtida com outros classificadores.

No trabalho de Souza (21), o sistema foi projetado com base em uma estrutura composta por duas SVMs (*Support Vector Machine*) com kernels Gaussianos, associadas com uma rotina de decisão. Nessa abordagem proposta, cada SVM (modificada) é treinada levando em conta apenas uma das classes, em que, no momento da aplicação do sistema para realizar uma classificação, a rotina de decisão retrocitada cuida de verificar qual das SVMs tem sua função objetivo mais próxima dos parâmetros de entrada analisados. Os testes realizados com os parâmetros extraídos de janelas de 1024 amostras, com 50% de sobreposição, mostraram-

se eficientes no pré-diagnóstico de diversas patologias na laringe. Os resultados mais promissores para a identificação correta de ambas as classes utilizaram sexo (SX), desvio-padrões dos MFCCs (*Mel Frequency Cepstral Coefficients*) (DP M) e suas derivadas, desvios padrões das energias das sub-bandas de frequências (DP E) e desvio-padrão do *pitch* (DP P) como parâmetros de entrada. Já para identificação correta das vozes normais apenas, os parâmetros desvio-padrão do *pitch* (DP P) e da dimensão fractal (DP D) foram suficientes para produzir quase que a totalidade dos resultados corretos. Por sua vez, para identificação das vozes patológicas, o uso da combinação de sexo (SX) com o desvio-padrão do *pitch* (DP P) forneceu resultados consideráveis. Tais resultados permitiram observar que o único parâmetro não-acústico utilizado foi relevante nas experiências.

O trabalho de Fonseca (12) demonstra que os algoritmos de análise tempo-frequência da DWT associados ao filtro LP inverso da vogal sustentada /a/ podem levar a um parâmetro de classificação adequada para a distinção de vozes patológicas, principalmente as de pacientes com edema de Reinke e nódulo nas pregas vocais. Evidenciou-se uma importante relação existente entre a escala frequencial dos componentes de detalhe em altas frequências e a ordem utilizada para o filtro LP (*Linear Prediction*) inverso, bem como melhorias obtidas na classificação com o método de inteligência artificial LS-SVM (*Least Squares Support Vector Machines*), levando a mais de 90% de acerto entre os sinais de voz normal e de pacientes com nódulo nas pregas vocais, mais de 85% de acerto entre as vozes normais e de pacientes com edema de Reinke, e acima de 82% entre as duas diferentes patologias.

O trabalho de Sáenz-Lechón (25) aborda uma comparação entre diversos trabalhos desenvolvidos na área e cita a falta de padronização do banco de dados, das técnicas utilizadas e das patologias a serem estudadas, como fator predominante para a comparação dos resultados (Tabela 2).

Tabela 2 - Pesquisas desenvolvidas na área de Detecção de Patologias na Voz. (25)

Primeiro Autor	DB	Características	Classificador	Melhores Resultados (%)
Banci (1986)	30 + 53	<i>Pitch, noise, residual</i>	<i>Mahalanobis distance</i>	91.6
Kasuya (1986)	804 + 66	<i>Perturbation, noise</i>	<i>Thresholds</i>	-
Childers (1992)	52 + 29	<i>Linear prediction coefficients (LPC)</i>	<i>Vector quantization</i>	82.6
Plante (1993)	209 + 88	<i>Cepstral, perturbation, noise</i>	<i>Thresholds</i>	89
Wallen (1996)	9 + 20	<i>Perturbation, cepstral, LPC</i>	<i>Multi-layer perceptron</i>	85.5
Gavidia (1996)	10 + 24	<i>Spectral</i>	<i>Hidden Markov Models</i>	92.8
Boyanov (1997)	50 + 150	<i>Perturbation, noise, energies</i>	<i>K-nearest neighbours, linear discriminant analysis, self-organized maps</i>	96.5
Ritchings (1998)	20 + 20	<i>Spectral</i>	<i>Multi-layer perceptron</i>	90.5
Tadeusiewicz (1998)	-	<i>Formants, perturbation</i>	<i>Multi-layer perceptron</i>	84.75
Cheol-Woo (1999)	33 + 33	<i>Noise, perturbation</i>	<i>Multi-layer perceptron</i>	91.6
Alonso (2005)	100 + 68	<i>High order statistics, bispectrum, chaos</i>	<i>Multi-layer perceptron</i>	98.3

O trabalho de Sáenz-Lechón (25) tem como objetivo o desenvolvimento de uma metodologia que permite que os resultados de diferentes classificadores e as características possam ser comparados. Assim, adotam-se questões metodológicas que são comuns na verificação de locutor. As vozes são selecionados a partir de MEEI VddB (*Massachusetts Eye & Ear Infirmary Voice Disorders Database*) sendo 53 falantes normais, 173 pacientes patológicos. Cada arquivo é dividido em quadros, usando janelas de *Hamming* de 40 ms, com 50% de sobreposição entre quadros consecutivos. Após serem retirados os quadros com silêncio, são calculados 18 coeficientes, MFCCs seguindo uma abordagem não paramétrica, com base no sistema humano de percepção auditiva. O detector é um *perceptron* base multi-camada *feedforward* (MLP), com três camadas. A camada de entrada é feita com os parâmetros MFCC, a camada oculta tem 12 neurônios e a camada de saída tem dois nós. A aprendizagem supervisionada é realizada pelo algoritmo *backpropagation*. Os pesos iniciais são aleatórios, e o treinamento é realizado on-line, ou seja, os pesos são atualizados imediatamente depois de cada exemplo ser apresentado à rede. O banco de dados é dividido em dois subgrupos: um conjunto de treinamento com 70% dos processos normais e patológicos e de um conjunto de teste com a restante de 30%. Como resultado obteve-se a matriz de confusão demonstrada na Tabela 3:

Tabela 3 - Resultados da classificação em porcentagem. (25)

Decisão do Detector	Diagnóstico Atual	
	Patológico	Normal
Patológico	92.73 $\pm$ 5.24	13.66 $\pm$ 6.49
Normal	7.27 $\pm$ 5.24	86.34 $\pm$ 6.49

O trabalho de Gómez-Vilda (24) formaliza o uso da densidade espectral da fonte glotal, adicionando estimativas do modelo de inversão biomecânica da prega vocal, para definir assinaturas para o estudo de patologias a partir do que conclui que a detecção de patologias depende dos componentes glotais da voz (assinatura). Utiliza Análise Discriminante Linear (LDA) para detectar os parâmetros mais relevantes e a técnica de detecção é baseada em uma mistura gaussiana de modelos. Segundo o autor, esta técnica é mais relevante que a análise tempo-frequência e visa oferecer algumas semânticas para parâmetros usados na detecção de patologias para conectá-los com a dinâmica das pregas vocais e os parâmetros clássicos de perturbação como *jitter* e *shimmer*. O banco de dados utilizado, que contém 200 vozes, foi balanceado por sexo, idade e patologias.

No artigo de Arjmandi (23) são aplicadas algumas características acústicas de tempo-longo, geralmente calculadas pela média das medidas de perturbações de tempo a partir da movimentação da janela ao longo do segmento do sinal de voz. Os parâmetros utilizados na análise do sinal de voz foram: medidas de rouquidão, aspereza, tremor, intensidade da amplitude e ruído de frequência (turbulência), ruído anormal, irregularidade, variação da frequência fundamental, nível de energia relativa em alta frequência e densidade espectral (conteúdo de frequência). Os parâmetros acústicos foram quantificados em um vetor de características. Foi utilizado um banco de dados com 50 vozes normais e 50 patológicas, dividido aleatoriamente 150 vezes entre treinamento e teste. Foram extraídas 22 características acústicas de cada, utilizando-se MDVP, 2 métodos de redução de características LDA e PCA em comparação com estratégias de seleção IFS, FFS, BFS e BBFS.

Os resultados demonstrados com o uso de LDA obtiveram ACC de 94,26% e AUC de 97,94%, IFS com 91,55% e BFS com 91%; desta forma, métodos de extração de características são mais confiáveis que métodos de seleção de características para a identificação de patologias na voz, e a complexidade destes algoritmos é mais apropriada em

comparação aos métodos de redução de características. SVM tem estabelecido resultados adequados de generalização para distinção entre vozes normais e patológicas.

CAPÍTULO 3

A TÉCNICA PROPOSTA E OS RESULTADOS OBTIDOS

Este capítulo apresenta a técnica utilizada para a criação dos filtros de classificação das patologias, assim como os resultados obtidos.

3.1 A base de dados necessária para o desenvolvimento do projeto

Para a elaboração deste projeto foi utilizado um banco de vozes previamente laudado pelo Departamento de Otorrinolaringologia e Cirurgia de Cabeça e Pescoço do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto (FMRP-USP). Segundo (12), o banco de dados possui arquivos no formato *wave* cujos sinais de voz contém o fonema do português brasileiro /a/ sustentado em nível confortável, considerado o melhor som vocálico para extração e medição de parâmetros da voz, visando sua classificação como normal ou patológica.

Os sinais foram arquivados em formato *wave* pelos próprios profissionais do departamento. Para a gravação da voz utilizaram-se o software *SoundForge 7.0®* (*Sony Pictures Digital Inc®*) a uma taxa de amostragem de 22.500Hz e 8 bits, um microfone unidirecional *Shure®* e uma sala com tratamento acústico, monitorada para que o ruído fosse inferior a 50 dB, verificado pelo decibelímetro *Decibel®*. Solicitou-se a cada paciente, independentemente de ser portador de patologia ou não, sustentar um som vocálico /a/ conforme seus padrões naturais de conversação. Os arquivos de áudio foram analisados e nota-se que não há ruídos e podem ser processados adequadamente visando sua classificação.

Inicialmente o banco de dados de vozes estava separado por pastas indicativas de cada paciente e, em cada pasta, havia arquivos com a vogal /a/, /e/ e /i/. Como para este trabalho foram utilizados apenas os arquivos com a vogal /a/, foi necessária a separação dos arquivos em pastas considerando-se apenas o tipo de voz, ou seja, normais, edemas ou nódulos.

Toda a classificação das vozes foi realizada por profissionais e confirmadas por exames de videolaringoscopia e videoestroboscopia, sendo 83 vozes normais, 27 com edemas de Reinke e 40 com nódulos nas pregas vocais, de acordo com a Figura 7. Nota-se que, neste banco de dados, há proporcionalmente uma quantidade muito maior de vozes oriundas de indivíduos normais em comparação com a quantidade de vozes oriundas de indivíduos com edemas ou nódulos.

Figura 7 - Total de vozes disponíveis no banco de dados.

Em contrapartida, se pensarmos em duas classes de vozes, normais e patológicas, obtém-se um equilíbrio maior, como consta na Figura 8.

Figura 8 - Vozes separadas em classes.

Todas as vozes de cada um dos grupos, isto é, 83 vozes de indivíduos normais e 67 vozes de indivíduos com patologia, foram utilizadas para a construção dos filtros e para os testes. Visto que com esta metodologia de trabalho uma voz poderia ser identificada por seu próprio filtro, o sistema classificador já se encarrega de separar estes casos e de obter uma nova classificação para cada indivíduo.

3.2 Metodologia proposta

Apoiado por um *software*, que foi desenvolvido por esta autora utilizando-se da linguagem de programação Java, três etapas foram seguidas para possibilitar a construção do conjunto de filtros digitais proposto para detecção das patologias, conforme abaixo:

Etapas 1 (Algoritmo 1): Procedimentos iniciais

Passo 1.1: Realizar a leitura do arquivo *wave* de cada voz;

Passo 1.2: Separar o cabeçalho do arquivo dos dados, extraíndo-se as amostras brutas;

Passo 1.3: Dividem-se as amostras pelo módulo da maior delas, normalizando os sinais;

Passo 1.4: Separa-se os dados em janelas de tamanho 1024;

Passo 1.5: Realiza-se a DWT nível 10, com filtro Haar, em cada janela do sinal, visando decompor o mesmo em bandas de energia diversas. Tendo em vista a taxa de amostragem dos sinais em 22050 Hz, a máxima frequência presente nos mesmos é de $22050 / 2 = 11025$ Hz. Assim, a decomposição em nível 10 implica que, na primeira sub-banda, a resolução é de $11025/(2^{10}) = 10,76$ Hz e na última sub-banda a resolução é de $11025/2 = 5512.50$ Hz;

Passo 1.6: Calcula-se a energia de cada uma das 8 sub-bandas Bark obtidas na decomposição de nível 10;

Passo 1.7: Calcula-se a variância estatística de todos os vetores de energia, de cada janela de tamanho 1024, resultando em um vetor de 8 posições que representa o “vetor de características” do sinal sob análise.

Algoritmo 1.

INÍCIO**PARA CADA ARQUIVOPACIENTE**

ABRIR ARQUIVOPACIENTE

DIVIDIR OS DADOS EM JANELAS DE 1024

PARA CADA JANELA

REALIZAR A DWT(JANELA)

CALCULAR ENERGIA DA SUB-BANDA BARK

FIMPARA

CALCULAR VARIANCIA DAS ENERGIAS DO SINAL

FIMPARA**FIM**

Etapa 2 (Algoritmo 2): Geração dos filtros para cada sinal-exemplo. Cada grupo de filtros, isto é, filtros obtidos com vozes-normais, com vozes-nódulo e com vozes-edema, constitui o que se chama de “mundo”. Dessa forma, três “mundos” existem, isto é, mundo normal, mundo edema e mundo nódulo.

Passo 2.1: Especificar o filtro *wavelet* de Daubechies de suporte 8, original. Em seguida, substituir a equação de conservação de energia pela equação Eq1 especificada abaixo. As demais equações abaixo (Eq2 até Eq8) correspondem ao filtro original de Daubechies:

$$\text{Eq1} = \text{Energia}(\text{vetor_características}) - \text{Energia}(\text{DWT}(\text{vetor_características})) = 1$$

$$\text{Eq2} = h_0 \cdot h_2 + h_1 \cdot h_3 + h_2 \cdot h_4 + h_3 \cdot h_5 + h_4 \cdot h_6 + h_5 \cdot h_7 = 0$$

$$\text{Eq3} = h_0 \cdot h_4 + h_1 \cdot h_5 + h_2 \cdot h_6 + h_3 \cdot h_7 = 0$$

$$\text{Eq4} = h_0 \cdot h_6 + h_1 \cdot h_7 = 0$$

$$\text{Eq5} = h_0 - h_1 + h_2 - h_3 + h_4 - h_5 + h_6 - h_7 = 0$$

$$\text{Eq6} = -h_1 + 2 \cdot h_2 - 3 \cdot h_3 + 4 \cdot h_4 - 5 \cdot h_5 + 6 \cdot h_6 - 7 \cdot h_7 = 0$$

$$\text{Eq7} = -h_1 + 4 \cdot h_2 - 9 \cdot h_3 + 16 \cdot h_4 - 25 \cdot h_5 + 36 \cdot h_6 - 49 \cdot h_7 = 0$$

$$\text{Eq8} = -h_1 + 8 \cdot h_2 - 27 \cdot h_3 + 64 \cdot h_4 - 125 \cdot h_5 + 216 \cdot h_6 - 343 \cdot h_7 = 0$$

Observa-se que na equação Eq1, a energia do vetor de características, oriundo da etapa 1, é um valor escalar. A energia¹ da sua transformada (DWT), que também é um escalar, quando subtraída do valor anterior, deve ser unitária. Esse é o critério chave da presente abordagem, isto é, o filtro proposto mantém unitárias as diferenças de energia entre os sinais original e transformado. Deve ser notado que a DWT do vetor de características é um sinal escrito como função dos coeficientes, ainda não determinados, h_0 até h_7 .

No sistema, observa-se ainda que as equações de Eq2 até Eq4 correspondem às condições de ortogonalidade dos filtros, oriundas dos filtros de Daubechies. Por fim, as equações de Eq5 até Eq8 correspondem às equações de momentos, que gerenciam a frequência de corte dos filtros em meia-banda dos filtros. A reunião das 8 equações sob 8 incógnitas forma o sistema a ser resolvido para encontrar os coeficientes dos filtros desejados.

Para resolver o sistema, o método numérico de Newton foi utilizado, admitindo-se um erro total de 10^{-8} .

¹ A Energia é calculada pela soma dos quadrados de cada elemento do vetor de características.

Algoritmo 2.

INÍCIO

PARA CADA VETORDECARACTERISTICAS

ENQUANTO ERRO>0

GERARFILTRO(VETORDECARACTERISTICAS)

FIMENQUANTO

FIMPARA

FIM

De posse dos três “mundos”, isto é, dos três conjuntos de filtros, além de todos os sinais de vozes normais, de vozes com edemas e de vozes com nódulos, realizam-se os testes. O sistema realiza a abertura do arquivo de áudio de cada paciente e realiza-se a sua DWT usando cada um dos 83 filtros do mundo normal, cada um dos 27 filtros do mundo edema e cada um dos 40 filtros do mundo nódulo (Algoritmo 3). Para cada um dos sinais transformados, mede-se a diferença entre as energias desses sinais transformados e do sinal original. Comprovadamente a diferença mais próxima de zero é encontrada quando o sinal é transformado por seu próprio filtro; desta forma, o sistema automaticamente utiliza a segunda classificação encontrada.

Algoritmo 3.

INÍCIO

PARA CADA VETORDECARACTERISTICAS

CALCULAR ENERGIA DO VETORDECARACTERISTICAS

PARA CADA FILTRONORMAL

REALIZAR A DWT(VETORCARACTERISTICAS,FILTRONORMAL)

CALCULAR ENERGIA DO VETORDETRANSFORMADO

FIMPARA

PARA CADA FILTROEDEMA

REALIZAR A DWT(VETORCARACTERISTICAS,FILTROEDEMA)

CALCULAR ENERGIA DO VETORTRANSFORMADO

FIMPARA

PARA CADA FILTRONODULO

REALIZAR A DWT(VETORCARACTERISTICAS,FILTRONODULO)

CALCULAR ENERGIA DO VETORTRANSFORMADO

FIMPARA

CALCULARMENORDIFERENCA(VETOR DE ENERGIAS DOS SINAIS TRANSFORMADOS)

FIMPARA

FIM

Foi utilizada, como forma de reportar os resultados, uma matriz de confusão, que visa mostrar o número de classificações corretas em oposição às classificações preditas em cada classe. Os resultados foram expressos utilizando-se o seguinte modelo (Matriz 1):

NorNor	NorEde	NorNod
EdeNor	EdeEde	EdeNod
NodNor	NodEde	NodNod

Matriz 1 - Modelo da Matriz de Confusão.

no qual:

NorNor → Total de vozes normais reconhecidas como normais;

NorEde → Total de vozes normais reconhecidas como edemas;

NorNod → Total de vozes normais reconhecidas como nódulos;

EdeNor → Total de vozes edemas reconhecidas como normais;

EdeEde → Total de vozes edemas reconhecidas como edemas;

EdeNod → Total de vozes edemas reconhecidas como nódulos;

NodNor → Total de vozes nódulos reconhecidas como normais;

NodEde → Total de vozes nódulos reconhecidas como edemas;

NodNod → Total de vozes nódulos reconhecidas como nódulos.

De acordo com este modelo criado, nota-se que o maior número de classificações corretas deverá estar posicionado na diagonal principal da matriz, ou seja (Matriz 2):

NorNor	NorEde	NorNod
EdeNor	EdeEde	EdeNod
NodNor	NodEde	NodNod

Matriz 2 - Modelo da Matriz de Confusão evidenciando-se os melhores classificadores.

Pensando-se em uma classificação que diferencie apenas vozes normais de vozes patológicas, pode-se utilizar o modelo de matriz de confusão demonstrado na Matriz 3:

NorNor	NorPat
PatNor	PatPat

Matriz 3 - Modelo da Matriz de Confusão (Normais e Patológicos).

no qual:

NorNor → Total de vozes normais reconhecidas como normais;

NorPat → Total de vozes normais reconhecidas como patológicas;

PatNor → Total de vozes patológicas reconhecidas como normais;

PatPat → Total de vozes patológicas reconhecidas como patológicas.

De acordo com este segundo modelo criado, nota-se que o maior número de classificações corretas também deverá estar posicionado na diagonal principal da matriz (Matriz 4), ou seja:

NorNor	NorPat
PatNor	PatPat

Matriz 4 - Modelo da matriz de confusão evidenciando-se os melhores classificadores (normais e patológicos).

3.3 Testes e Resultados

Segundo os testes realizados foi obtida a matriz seguinte (Matriz 5), que corresponde à melhor matriz para diferenciação entre vozes normais e patológicas.

83	0	0
2	10	15
1	16	23

Matriz 5 - Resultados da Matriz de Confusão

De acordo com a matriz de confusão acima podem-se observar graficamente os resultados obtidos e suas respectivas classificações (Figura 9):

Figura 9 - Resultados da Matriz de Confusão.

Para uma melhor análise dos resultados, a Figura 10 demonstra uma comparação entre a classificação correta e os resultados obtidos, sendo que a coluna verde representa a classificação correta e a coluna azul, o reconhecimento dado pelo sistema:

Figura 10 - Comparativo (Resultados do reconhecimento).

Nota-se que, para vozes normais, o reconhecimento foi muito bom, já para vozes oriundas de pacientes com edemas e nódulos, nota-se que o sistema diagnostica como patológico, mas não realiza uma diferenciação adequada entre as patologias citadas.

De forma mais específica, os resultados podem ser analisados em uma matriz de confusão que diferencie apenas entre vozes normais e patológicas (Matriz 6):

$$\begin{pmatrix} 83 & 0 \\ 3 & 64 \end{pmatrix}$$

Matriz 6 - Matriz de confusão (Normais e Patológicos)

Na Figura 1 podem-se visualizar quantitativamente os resultados obtidos na matriz de confusão da Matriz 6:

Figura 11 – Diferenciação entre vozes normais e patológicas.

Para uma melhor análise, podemos visualizar o gráfico acima também em termos de porcentagem de acertos por tipo (Figura 12).

Figura 12 - Porcentagem de acertos do diagnóstico realizado pelo sistema.

Se as vozes oriundas de indivíduos com edemas e nódulos forem unificadas em um único grupo denominado “patológico”, observam-se os seguintes resultados (Figura 13):

Figura 13 - Porcentagem de acertos (Normais e Patológicas).

Partindo-se dos resultados demonstrados na Figura 13, pode-se dizer que, de acordo com o banco de dados de vozes utilizado, o sistema obteve um reconhecimento de 100% das vozes normais e de 95,52% das vozes patológicas.

De acordo com alguns trabalhos desenvolvidos na área (23 e 25), podem-se demonstrar os resultados obtidos através de algumas características do sistema de diagnóstico, conforme calculado na Tabela 4:

Tabela 4 - Taxas de reconhecimento do sistema

Taxa	Cálculo	Resultado
Taxa positiva (TP) ou Sensibilidade	É a razão entre arquivos patológicos corretamente classificados e o número total de vozes patológicas	$TP = 64 / 67 = 0,955$
Taxa negativa (TN) ou Especificidade	é a razão entre os arquivos normais corretamente classificados e o número total de ficheiros normais.	$TN = 83 / 83 = 0$
Taxa de falsos negativos (FN)	é a relação entre arquivos patológicos erroneamente classificados e o número total de arquivos patológico.	$FN = 3 / 67 = 0,04477$
Taxa de falso positivo (FP)	é a relação entre arquivos normais erroneamente classificados e o total número de ficheiros normais	$FP = 0 / 83 = 0$
Acurácia (ACC)	Probabilidade da classificação ser correta	$ACC = (TP + TN) / (TP + TN + FN + FP)$ $ACC = 0,955219$

Foram realizados testes utilizando-se *Ten Fold Cross Validation* (Matriz 4) com grupos de vozes separados igualmente em 10 grupos (sendo 8 normais, 2 edemas e 3 nódulos em cada grupo), obteve-se o seguinte resultado (baseado na Matriz 3):

Tabela 5 - Resultados Ten Fold Cross Validation

Teste 1		Teste 2		Teste 3		Teste 4		Teste 5		Teste 6		Teste 7		Teste 8		Teste 9		Teste 10	
7	1	6	2	7	1	5	3	6	2	5	3	8	0	6	2	3	5	2	6
3	2	4	1	3	2	0	5	4	1	2	3	4	1	2	3	1	4	3	2

Sendo assim, realizando-se a média dos resultados encontrados, obteve-se:

$$\begin{pmatrix} 5,5 & 2,5 \\ 2,6 & 2,4 \end{pmatrix}$$

Matriz 7 - Matriz de Confusão (*Ten Fold Cross Validation*)

Calculando-se as taxas de reconhecimento de cada teste, obteve-se:

Tabela 6 - Taxas de reconhecimento (*Ten Fold Cross Validation*)

Taxas		Teste 1		Teste 2		Teste 3		Teste 4		Teste 5		Teste 6		Teste 7		Teste 8		Teste 9		Teste 10	
TN	FP	0.875	0.125	0.75	0.25	0.875	0.125	0.625	0.375	0.75	0.25	0.625	0.375	1	0	0.75	0.25	0.375	0.625	0.25	0.75
FN	TP	0.6	0.4	0.8	0.2	0.6	0.4	0	1	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.2	0.8	0.6	0.4

CAPÍTULO 4

CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo contém as conclusões obtidas a partir dos resultados mostrados no capítulo anterior, destacando a aplicabilidade da criação e uso dos filtros para detecção de patologias, além de sugestões para trabalhos futuros.

Evidencia-se aqui que a principal contribuição deste trabalho é a criação de famílias de filtros para a detecção de patologias na laringe, as quais, sem a combinação com outras técnicas, tais como Redes Neurais ou Máquinas de Vetores de Suporte, realiza a classificação adequada para a distinção das vozes normais e vozes patológicas (edemas e nódulos). Esse fato implica que os filtros digitais projetados já incorporam em suas criações os critérios estatísticos necessários para que ocorra o “aprendizado” do sistema de classificação, isto é, os filtros criados podem ser classificados como “inteligentes” pois suas formulações basearam-se em um conceito de aprendizagem. Tal fato, associado ao uso dos filtros para o pré-diagnóstico de patologias na laringe, é novo na literatura.

Os resultados dos testes realizados com o banco de vozes fornecido pela Faculdade de Medicina de Ribeirão Preto, na qual existem os arquivos de áudio com o /a/ sustentado de 83 com vozes normais, 27 pacientes com edemas e 40 pacientes com nódulos, também já diagnosticados pelos fonoaudiólogos do departamento de Fonoaudiologia. Os resultados foram satisfatórios para as vozes normais, visto que se obteve uma taxa de acertos de 100% e 95,52% das vozes patológicas.

Para testes individuais com a família de filtros criada, foi desenvolvida uma interface gráfica onde o profissional de saúde poderá selecionar um arquivo de voz e, ao clicar no botão diagnóstico, o sistema fará a abertura do arquivo de áudio e a aplicação da técnica desenvolvida neste trabalho, exibindo ao final, o pré-diagnóstico computacional (Figura 14).

Figura 14 - Interface para diagnóstico simples

A realização do janelamento das vozes e a sua representação em um vetor de 8 elementos com suas respectivas variâncias facilitou a convergência do sistema de equações não lineares para a criação dos novos filtros baseando-se nos filtros *Wavelet* de Daubechies, visto que se observou que, com mais de 12 elementos por equação, não ocorria a convergência do sistema de equações. É importante, ainda, mencionar que foram testados outros critérios para caracterizar a equação Eq1 do sistema de equações do Capítulo anterior, tais como o uso de fractais, entropia e lacunaridade, sendo que em todos os casos não houve convergência do sinal, optando-se, assim, pelo uso do critério de energia.

Já está em fase final de desenvolvimento o *software* para implantação de um prontuário eletrônico de pacientes e aquisição da voz, integrado com um sistema de apoio à decisão médica (Apêndice 2), a ser utilizado pelo Departamento de Otorrinolaringologia e Cirurgia de Cabeça e Pescoço do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto (FMRP-USP).

Esta técnica, nova na literatura, caracteriza a construção de famílias de filtros *Wavelet* que incorporam uma aprendizagem estatística e, por isso, não necessita a associação com técnicas de Inteligência Artificial. Em comparação com sistemas tradicionais de classificação, este sistema incorpora, em um só operador, as fases de extração de características e de classificação.

No tocante à comparação dos resultados obtidos por esta técnica nota-se, como em outros trabalhos da área, problemas na verificação da melhor ou pior técnica visto que cada trabalho utiliza uma base de dados diferente.

REFERÊNCIAS

- 1 FERREIRA, M. J. **Protocolo de Anamnese estandardizado na área da voz**. 2010. 13p. Dissertação (Mestrado – Ciências da fala e audição) - Portugal: Universidade de Aveiro, Aveiro, 2010.

- 2 BARBON JUNIOR, S. **Identificação de patologias na laringe com base na Discriminative Paraconsistent Machine**. 2011. 94 p. Tese (Doutorado – Física Computacional), Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2011.

- 3 DAJER, M. E. **Análise de sinais de voz por padrões visuais de dinâmica vocal**. 2010. 154 p. Tese (Doutorado). Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

- 4 PARRAGA, A. **Aplicação da transformada wavelet packet na análise e classificação de sinais de vozes patológicas**. 2002. 163 p. Dissertação (Mestrado - Engenharia Elétrica) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

- 5 DELLACQUA, A. **Análise acústica da voz de obesos mórbidos pré e pós tratamento cirurgico para a obesidade**. Dissertação (Mestrado – Engenharia Eletrica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2007.

- 6 BEHLAU, M. S. **Voz o livro do especialista**. Rio de Janeiro: Revinter, 2001.

- 7 BARBON JUNIOR, S. **Dynamic time warping baseado na transformada wavelet**. 2007. 113 p. Dissertação (Mestrado) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2007.

- 8 ADDISON, P. S. **The illustrated wavelet transform handbook**: introductory theory and applications in science, engineering, medicine and finance. Edinburg-UK: Institute of Physics Publishing, 2002.

- 9 JENSEN, A. et al. **Ripples in mathematics**: the discrete wavelet transform. New York-USA: Springer-Verlag, 2000.

- 10 STRANG, G.; NGUYEN, T. **Wavelets and filter-banks**. Wellesley: Wellesley-Cambridge Academic Press, 1997.
- 11 WALKER, J. S. **A primer on wavelets and their scientific applications**. Whashington USA: Chapman and Hall/CRC, 1999.
- 12 FONSECA, E. S. **Wavelets, predição linear e LS-SVM aplicados na análise e classificação de sistemas de vozes patológicas**. 2008. 107 p. Tese (Doutorado – Engenharia Elétrica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.
- 13 HAYKIN, S.; VEEN, B. V. **Sinais e sistemas**. Porto Alegre: Bookman, 2002.
- 14 ESHAGHI, M.; MOLLAEI, M. R. K. Voice activity detection based on using wavelet packet. **Digital Signal Processing**, v. 20, n. 4, p. 1102–1115, 2010.
- 15 RAFIEE, J. et al. Wavelet basis functions in biomedical signal processing. **Expert Systems with Applications**, v. 38, n. 5, p. 6190–6201, 2011.
- 16 MALEKNEJAD. K.; YOUSEFI, M.; NOURI, K. Computational methods for integrals involving functions and Daubechies wavelets. **Applied Mathematics and Computation**, v. 189, n. 2, p. 1828–1840, 2007.
- 17 VAZIRI, G.; ALMASGANJ, F.; BEHROOZMAND, R. Pathological assessment of patients speech signals using nonlinear dynamical analysis. **Computers in Biology and Medicine**, v. 40, p. 54–63. 2010.
- 18 PIFER, A. C.; CAMPOS, C. F. P. S.; CAMPOS, A. L. P. S. *Aplicação de wavelets para remoção de ruídos em sinais unidimensionais*. **Holos**, v. 1, n. 24, 2008. Disponível em: <www.etfrn.br/ojs/index.php/HOLOS/article/download/115/141>. Acesso em: 15 mar. 2012.
- 19 ARAÚJO, L. C. **Banco de filtros wavelets com fator de escala maior que dois**. 2007. Dissertação (Mestrado – Engenharia Elétrica) - Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, 2007.
- 20 ALCAIM, A.; OLIVEIRA, C. A. S. **Fundamentos do processamento de sinais de voz e imagem**. Rio de Janeiro: Editora Interciência, 2011.

- 21 SOUZA, L. M. **Detecção inteligente de patologias na laringe baseada em máquinas de vetores de suporte e na transformada wavelet**. 2011. Dissertação (Mestrado) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2011.
- 22 ALVES, L. A. **Uso prolongado da voz em professoras universitárias: uma questão de saúde do trabalhador**. Tese (Doutorado) - Escola de Enfermagem de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2011.
- 23 ARJMANDI, M. K. et al. Identification of voice disorders using long-time features and support vector machine with different feature reduction methods. **Journal of Voice**, v. 25, n. 6, p. e275–e289, nov. 2011.
- 24 GÓMEZ-VILDA, P. et al. Glottal source biometrical signature for voice pathology detection. **Speech Communications**, v. 51, n. 9, p. 759-781, 2009. DOI: 10.1016/j.specom.2008.09.005
- 25 SÁENZ-LECHÓN, N. et al. Methodological issues in the development of automatic systems for voice pathology detection. **Biomedical Signal Processing and Control**, v. 1, n. 2, p. 120–128, abr. 2006. DOI: [10.1016/j.bspc.2006.06.003](https://doi.org/10.1016/j.bspc.2006.06.003)

APÊNDICE

1. Trechos do códigos-fonte da implementação do *software*.

O trecho de código abaixo indica o cálculo das equações:

```
double eq1(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
int comprimento_do_sinal_original=8;
double u[]=new double[comprimento_do_sinal_original]; //sinal original
u[0]=VetorOriginal[0];
u[1]=VetorOriginal[1];
u[2]=VetorOriginal[2];
u[3]=VetorOriginal[3];
u[4]=VetorOriginal[4];
u[5]=VetorOriginal[5];
u[6]=VetorOriginal[6];
u[7]=VetorOriginal[7];

int comprimento_do_filtro=8;
double s[]= new double[comprimento_do_filtro];
s[0]=S0;
s[1]=S1;
s[2]=S2;
s[3]=S3;
s[4]=S4;
s[5]=S5;
s[6]=S6;
s[7]=S7;

int comprimento_do_sinal_filtrado=comprimento_do_sinal_original+comprimento_do_filtro-1;
double y[]=new double[comprimento_do_sinal_filtrado]; //sinal filtrado
for(int n=0;n<comprimento_do_sinal_filtrado;n++) // faz a convolucao
{
    y[n]=0;
    for(int k=0;k<comprimento_do_filtro;k++)
        if(n-k>=0 && n-k<8)
            y[n]+=u[n-k]*s[k];
}
return energia(u)-energia(y);
}

double eq2(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return S0*S2+S1*S3+S2*S4+S3*S5+S4*S6+S5*S7; }
double eq3(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return S0*S4+S1*S5+S2*S6+S3*S7; }
double eq4(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return S0*S6+S1*S7; }
double eq5(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return S0-S1+S2-S3+S4-S5+S6-S7; }
double eq6(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return -S1+2*S2-3*S3+4*S4-5*S5+6*S6-7*S7; }
double eq7(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return -S1+4*S2-9*S3+16*S4-25*S5+36*S6-49*S7; }
double eq8(double S0, double S1, double S2, double S3, double S4, double S5, double S6, double S7){
    return (-S1+8*S2-27*S3+64*S4-125*S5+216*S6-343*S7); }
```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Ctr;

import Model.Wave;
import java.io.File;
import java.io.FileOutputStream;
import java.util.Vector;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

/**
 *
 * @author Luciene
 */
public class testes0208v1 {

    JFileChooser arquivo = new JFileChooser();
    Vector dados = new Vector();
    Vector linha = new Vector();
    String sCabecalho = "Nome arquivo*Tipo*FTodos1*FTodos2";/**FNor1*FNor2*FEde1*FEde2*FNod1*FNod2";
    String sLinha="\n";

    private Wave wave;
    private AbrirWave abrirWave;
    private File diretorioPrincipal;
    private File diretorioPaciente;
    //private int cont = 1;
    private int bark = 0;
    private double[][] energiaJanelas;
    private double[][] matrizDados;

    int qtdeArquivos = 0;
    private String resultado;

    private double matrizFiltrosNormais[][]={
    {0.2455764201855001,0.7620068063710879,0.6725015674333131,-0.029829929444961656,-
    0.19937397118383735,0.03287606578067356,0.03505238709442987,-0.01129653917738696},
    {0.22729199248221382,0.7052714799501726,0.6224303664572867,-0.027608937755619012,-
    0.18452955346948852,0.030428273572117404,0.03244255656928587,-0.01045545372736594},
    {0.18599137456600884,0.577118492233732,0.5093302151331796,-0.02259219177673751,-
    0.15099918357452738,0.024899233648943996,0.026547506688908692,-0.008555621292361473},
    {0.18268072281483433,0.5668457666763932,0.5002641228340533,-0.022190050121281425,-
    0.14831139381710756,0.024456026582621002,0.02607496031564792,-0.008403330990297548},
    {0.2436049835313772,0.7558895735046898,0.6671028640928552,-0.029590460947728603,-
    0.19777343822398327,0.03261214353164647,0.03477099378850796,-0.011205852899843646},
    {0.26988363548126176,0.8374304300451588,0.7390659402423568,-0.03278250329846257,-
    0.2191080565584658,0.03613014696812952,0.038521881108110174,-0.01241467344155744},
    {0.2267734464845206,0.703662467247079,0.6210103482162376,-0.027545950476492307,-
    0.18410856608509923,0.030358854234879773,0.032368541828686266,-0.010431600561114092},
    {0.21464605456582633,0.6660320010215584,0.5877999525762996,-0.02607284794892554,-
    0.17426280693156773,0.02873532322092559,0.030637536727875444,-0.00987373935511731},
    {0.19196531407992848,0.5956552172459191,0.5256896183846321,-0.023317840412197238,-
    0.15584919337433847,0.025698983186302,0.02740019783964068,-0.008830423090153612},
    {0.29853291044997526,0.9263271673917908,0.8175208762159927,-0.03626250292676765,-
    0.24236729178038005,0.039965512952081034,0.04261113965915326,-0.013732542872357595},
    {0.18256779039963458,0.5664953451294993,0.49995486176502324,-0.02217633233040022,-
    0.14821970836914508,0.024440907975109456,0.026058840890460015,-0.008398136088231069},
    {0.20768347036109824,0.6444275792699626,0.5687331839202159,-0.025227109602284713,-
    0.16861015485059863,0.02780322079779709,0.02964373122921503,-0.009553459805536959},
    {0.18598827748356345,0.5771088821990983,0.5093217338921981,-0.022591815577138318,-
    0.1509966691734064,0.02489881903305324,0.026547064626382264,-0.008555478826268222},
    {0.19221433901535695,0.596427924558817,0.5263715635782268,-0.02334808922944213,-
    0.1560513670613019,0.0257323209153559,0.027435742451074555,-0.008841878261367076},
    {0.2730289639015117,0.8471901686336295,0.7476793009677591,-0.0331645632893462,-
    0.2216616267153601,0.03655122169496143,0.0389708300309849,-0.012559358854343668},
    {0.27639381331254254,0.857631065816184,0.7568938114707869,-0.0335732882819499,-
    0.2243934174508308,0.03700168436762171,0.03945111231533786,-0.012714142254024318},
    {0.19855884946561922,0.6161145057971152,0.5437457610650029,-0.024118750756925968,-
    0.16120222903221365,0.0265816799163196,0.02834132709987461,-0.009133726358218517},
    {0.21843877124857977,0.677800541033308,0.5981861611221184,-0.02653354556380966,-
    0.17734196651066977,0.02924306579268988,0.03117889071130331,-0.010048204690849013},

```

{0.2009284494864088,0.6234672123104475,0.5502348194484068,-0.02440658377191791,-0.16312601538721172,0.026898905511914328,0.028679552313536895,-0.009242728189295647},
{0.18593619908905876,0.5769472865090579,0.5091791192148305,-0.02258548966509638,-0.1509543887447013,0.0248918471392444,0.026539631208947333,-0.008553083215063472},
{0.21106249240472177,0.6549124531604366,0.5779865056316899,-0.025637556130816034,-0.17135345179680403,0.028255580804090057,0.03012603644640258,-0.009708895147693236},
{0.28537536503912486,0.88550020546021,0.7814894449177916,-0.03466426865420167,-0.23168519095974172,0.038204072142281054,0.04073309544544051,-0.013127294505679529},
{0.1895405908823405,0.5881314672917003,0.5190496073048304,-0.02302331163841268,-0.15388065464997921,0.025374378082588796,0.027054104611086913,-0.008718885587590538},
{0.26467830380268187,0.8212786424855033,0.7248113399423319,-0.03215021671088069,-0.2148820496507033,0.03543329331033148,0.0377788973118049,-0.01217522767883334},
{0.2057153558770283,0.6383206548696231,0.5633435782144827,-0.024988044645824117,-0.1670123190317644,0.0275397432978266,0.029362812113732928,-0.009462926348138707},
{0.266165882079033,0.8258944959568306,0.7288850157527792,-0.032330911400517026,-0.21608975675946773,0.03563243995979513,0.03799122701974893,-0.01224365642400984},
{0.18458847929673125,0.5727654043313682,0.5054884404757375,-0.0224217834497914,-0.14986022731488471,0.024711423772404022,0.02634726422265513,-0.008491087973734242},
{0.19223441882940834,0.596490230846067,0.5264265513755617,-0.023350528304968882,-0.15606766908356062,0.025735009061421076,0.027438608546338825,-0.00884280193476347},
{0.1772988216581192,0.5501460961235952,0.48552599381935574,-0.021536315810538956,-0.14394203699806654,0.023735535029239783,0.025306773848451095,-0.008155763014431897},
{0.22710911216329316,0.7047040148504552,0.6219295557482053,-0.02758672346075236,-0.1843810800313343,0.030403790824994174,0.032416453119590204,-0.010447041214935397},
{0.1835526151889311,0.5695511889762334,0.5026517665164977,-0.02229595804185157,-0.1490192494204154,0.02457274947899201,0.026199410004174348,-0.008443438124178762},
{0.19845462384068793,0.6157911008238857,0.5434603432057475,-0.024106090571409278,-0.16111761228958846,0.026567726913426713,0.02832645043969895,-0.009128931969349935},
{0.21459400108694607,0.6658704826429934,0.5876574061294096,-0.026066525063360328,-0.1742205467308918,0.028728354662647518,0.030630106866775867,-0.009871344890033705},
{0.20535332146223412,0.6371972868850043,0.5623521609146056,-0.024944068676788427,-0.16671839733143104,0.02749127664444572,0.029311137077346016,-0.009446272729899651},
{0.18838958642905831,0.5845599793332801,0.5158976259445257,-0.022883500244442957,-0.152946198774880545,0.025293073839141917,0.02718065775868442,-0.008759670615611748},
{0.21893123643101445,0.6793286267535597,0.5995347580551769,-0.026593364831616624,-0.17774177989266143,0.029308993611493472,0.031249182802833043,-0.010070858137066164},
{0.1904272200544557,0.5908826168661122,0.5214776071410082,-0.02313100962353711,-0.15460047448805045,0.025493073839141917,0.02718065775868442,-0.008759670615611748},
{0.24160421139107338,0.7496813146345691,0.6616238266534508,-0.029347429097457283,-0.19614908892063787,0.03234429405144361,0.03448541327757493,-0.011113817187087129},
{0.1900308531370323,0.5896527174780977,0.5203921716051184,-0.02308286331872451,-0.15427867956037414,0.025440010988723918,0.027124082267395568,-0.008741437698917581},
{0.24961206913346207,0.774529148557477,0.6835530366300097,-0.030320135806351765,-0.20265035804724751,0.033416330437112,0.035628416051095735,-0.011482179420910328},
{0.21042132626020366,0.6529229585429632,0.5762306968417097,-0.025559674301448846,-0.17083291387087196,0.0281697459331053,0.03003451950079142,-0.009679401442779338},
{0.2199867627531518,0.6826038526126423,0.6024252762310192,-0.0267215785896512,-0.17859872077637748,0.02945030014562081,0.03139984351042057,-0.010119412450390634},
{0.24418426629734943,0.757687048238424,0.6686892076343448,-0.02966082586315738,-0.19824373543494106,0.032689693885647596,0.034853677800819285,-0.011232499963334702},
{0.18973469755011077,0.5887337607768712,0.519581155460891,-0.023046889344041623,-0.1540382407016426,0.025400363467590878,0.027081810169895696,-0.008727814421158204},
{0.2180954301926437,0.6767351772604657,0.597245934865433,-0.02649184025893401,-0.17706322122339532,0.029197101676380878,0.03112988387429114,-0.01003241096893281},
{0.21960388228957953,0.6814158007669143,0.6013767728295218,-0.026675070471300147,-0.1782878749775281,0.029399042767990543,0.031345193010140875,-0.010101799911883336},
{0.1842006405401793,0.571561967242238,0.5044263589799021,-0.02237467305240373,-0.14954535607026634,0.024659502612941444,0.026291905999684926,-0.008473247353268374},
{0.20082999594185497,0.6231617177071599,0.5499652082089328,-0.024394624715403313,-0.16304608477278223,0.026885725235059388,0.02866549953211793,-0.009238199316684976},
{0.20463282785181916,0.6349616445759081,0.5603791169148444,-0.024856551017120723,-0.16613345651250507,0.027394821963199617,0.02920829731401585,-0.009413129953850703},
{0.255406580780506,0.7925091211920963,0.6994210835793191,-0.031023989512944248,-0.20735469732081968,0.03419205941765157,0.0364554965383913,-0.011748727519401208},
{0.2266902627064033,0.7034043536828488,0.6207825526439811,-0.027535846223688407,-0.18404103240182146,0.030347718168317665,0.0323566685796653,-0.010427774099242408},
{0.20722434054623642,0.6430029309113445,0.5674758746072261,-0.025171339549227303,-0.16823740516067012,0.02741755686513762,0.0295781972663656,-0.009532339789465487},
{0.1899354483430602,0.5893566829389824,0.520130909251679,-0.023071274590984372,-0.15420122410830012,0.025427238857425413,0.027110464649847153,-0.00873704906912974},
{0.1866619705329046,0.5791993055736836,0.5111666163581472,-0.022673648418069244,-0.1515436144318502,0.024989008380179464,0.026643224304622078,-0.008586468771962789},
{0.18542718121521556,0.5753678389214596,0.5077851933739351,-0.02252365975792738,-0.15054113687452986,0.024823703361061766,0.026466976466531622,-0.008529668343434037},


```
{0.23005863157245282,0.7138561714932042,0.6300067010390134,-0.027944998721078012,-
0.18677568044632428,0.03079865200117578,0.03283745321395716,-0.010582719394195712},
{0.20887561046845018,0.648126708344586,0.571997813684705,-0.025371917704235143,-
0.1695780071681409,0.02796281623680333,0.02981389152588415,-0.009608298366248378},
{0.1756078550249085,0.5448967547405785,0.480893239520371,-0.021330822261427993,-
0.1425685820251171,0.02350905714062991,0.025065303635048983,-0.008077942986982108},
{0.22268859274355182,0.6909874459592151,0.6098241335891885,-0.027049767256645516,-
0.18079223175859133,0.02981200238244589,0.031785489618525335,-0.01024369689233395},
{0.21683446352737162,0.6728224840922102,0.5937928262231774,-0.026338671861784606,-
0.17603949129282684,0.02902829221575184,0.030949899608574985,-0.009974406379873051},
{0.173175441014519,0.5373515284965048,0.47423427475345037,-0.021035452765217264,-
0.14059442417352838,0.02318352546262357,0.02471822249507613,-0.007966087104389218},
{0.19607507225086613,0.6083935520819839,0.5369317097572023,-0.023816502138351216,-
0.1591820932661787,0.026248566642125853,0.027986162479167534,-0.009019265364693878},
{0.22202794807632736,0.6889375108203385,0.608014983615061,-0.026969519390027834,-
0.1802558799755232,0.02972355986210571,0.03169119240306006,-0.010213307173483921},
{0.23047085317770946,0.7151352668912944,0.6311355540268349,-0.0279950708794359,-
0.1871103471801487,0.030853837367101355,0.032896291726487094,-0.01060168162806991},
{0.20824928476362284,0.6461832625946925,0.5702826448671859,-0.025295838528675542,-
0.1690695176196264,0.02787896810082806,0.029724492832653635,-0.009579487323001467},
{0.3714916283881988,1.1527130704481026,1.0173155150266637,-0.04512472759332224,-
0.3015996452646328,0.04973271946319007,0.053024913184934824,-0.017088650982812686},
{0.23034514852554558,0.7147452139681798,0.6307913166788438,-0.027979801657325193,-
0.1870082924483497,0.030837008901199146,0.03287834925414432,-0.010595899201862722},
{0.22969820029788593,0.7127377779428726,0.6290196738768978,-0.027901217483930017,-
0.18648306027336567,0.030750399965075885,0.0327860070020305,-0.010566139520562795},
{0.19296079027742588,0.5987441116825284,0.528415691607191,-0.02343875993986628,-
0.15665738189082598,0.02583225062674803,0.027542287283698955,-0.008876215091912944},
{0.42012505894811164,1.3036192733963987,1.1504962913263423,-0.05103218320804604,-
0.3410832413513382,0.05624342542192917,0.059966613175627737,-0.01932579351154485},
{0.21680285975001087,0.6727244197363665,0.5937062804962064,-0.02633483298161785,-
0.17601383341169943,0.029024061321504556,0.030945388638682736,-0.009972952603045293},
{0.4079359260074847,1.2657972290099562,1.1171168202760162,-0.04955158105846049,-
0.33118735705664565,0.05461162894870613,0.058226795460809805,-0.018765092212543265},
```

```
};
```

```
double matrizFiltrosNodulos[][] = {
```

```
{0.22327237127126323,0.6927988707336026,0.6114227886010307,-0.02712067826788979,-
0.18126617890414282,0.02989015459780878,0.03186881533406147,-0.01027055076130161},
{0.22896313711728672,0.710456927255717,0.627006731670386,-0.027811930072710683,-
0.18588629098551473,0.0306511994810948796,0.032681087648919305,-0.010532326542870467},
{0.2282541460112171,0.7082569764295197,0.625065186835715,-0.027725809611083337,-
0.18531068860380245,0.030557080005107696,0.032579889697268695,-0.01049971288313846},
{0.230304023576277,0.7146176060682412,0.6306786975893446,-0.027974806249646416,-
0.1869749046970167,0.030831503378574626,0.032872479278354655,-0.010594007450202565},
{0.229034999944362,0.7106799126845096,0.6272035252542806,-0.027820659180577585,-
0.18594463362771005,0.03066161530720136,0.03269134500029872,-0.010535632239894798},
{0.1926323592740066,0.5977250127916273,0.5275162959643038,-0.023398865745012855,-
0.15639074149967833,0.025788282565184603,0.027495408635748984,-0.008861107234016613},
{0.23001878203376905,0.7137325211049264,0.6298975746122737,-0.02794015823628529,-
0.18674332815137987,0.030793317221663525,0.03283176528408823,-0.010580886311546316},
{0.2212942881382946,0.6866610142987245,0.6060058841342667,-0.02688040242932123,-
0.17966024992588908,0.029625342564348893,0.03158647334198665,-0.010179558745085932},
{0.19422493885346107,0.6026666781018757,0.5318775137895086,-0.023592314840628282,-
0.15768369509133673,0.026001486059493914,0.027722725719924166,-0.008934366049176813},
{0.2303344202966046,0.7147119250088115,0.6307619378371244,-0.027978498509766785,-
0.1869958261535302,0.030835572680408457,0.03287681795878055,-0.01059540570228957},
{0.229530014193792,0.7122159079850005,0.6285591027091192,-0.02788078808107877,-
0.1863465164983329,0.03072788437739277,0.0327620009332876,-0.010558402943441281},
{0.23447179424319212,0.7275499128963323,0.6420919770242591,-0.02848106130802273,-
0.1903585560598745,0.03138945557325489,0.03346736664835203,-0.010785725305628619},
{0.23021921098000925,0.7143544383928871,0.6304464415613229,-0.027964504145882568,-
0.18690604863945418,0.030820149257149532,0.03286037353973117,-0.010590106062537782},
{0.22328862875138727,0.6928493165805598,0.6114673090841362,-0.027122653048223696,-
0.18127937772173014,0.02989233103635118,0.031871135847920486,-0.010271298606966147},
{0.22349356308659896,0.6934852137818156,0.6120285138673421,-0.02714754622304733,-
0.1814457559603745,0.029919766222025915,0.03190038718093262,-0.010280725606287825},
{0.2208709084228682,0.6853472960492656,0.6048464750915385,-0.026828974906149328,-
0.17931652435517045,0.02956866342811893,0.03152604217493137,-0.010160083237060307},
{0.22939288043101072,0.7117903913146961,0.6281835671819348,-0.0278641305759925,-
0.18623518291488692,0.03070952586152819,0.03274242712513007,-0.010552094777035762},
{0.17289234334768694,0.5364730958325268,0.4734590226973394,-0.02100106516635384,-
0.1403645881574055,0.023145626312900243,0.02467781450722523,-0.007953064584222258},
```



```

{0.22851247151815682,0.7090585427785961,0.6257726012861488,-0.027757188159724226,-
0.18552041306406522,0.030591662830089673,0.03261676182716936,-0.010511595881544655},
{0.19279564486308334,0.5982316767766247,0.5279634472511742,-0.02341869988664831,-
0.1565233067338733,0.02581014210549343,0.02751871522826291,-0.008868618386815299},
{0.2303767154368889,0.7148431639312974,0.6308777614496461,-0.02798363605941263,-
0.18703392039938743,0.03084123486008002,0.03288285496196001,-0.010597351282850099},
{0.2396116228484077,0.7434984489072269,0.6561671996809585,-0.02910539130084929,-
0.1945313835628247,0.03207754013446325,0.034201000853675764,-0.011022157920616704},
{0.23084286833088288,0.7162896044232823,0.6321543031943206,-0.02804025919907678,-
0.1874123718548823,0.030903640172427277,0.032949391365060636,-0.010618794361243689},
{0.23037444451550118,0.7148361174178216,0.6308715426188983,-0.02798336021279542,-
0.18703207672638744,0.03084093084489458,0.03288253082165579,-0.010597246820245716},
{0.2237162906105478,0.6941763220286377,0.6126384446125377,-0.02717460071923964,-
0.18162657979886884,0.029949583436247776,0.03193217822740257,-0.010290971094019387},
{0.22994270552500298,0.7134964609105443,0.6296892420233738,-0.027930917296592375,-
0.18668156458444204,0.030783132626967212,0.032820906492221955,-0.010577386784755002},
{0.2303505866390777,0.7147620880618821,0.6308062087432832,-0.027980462219698632,-
0.18701270744181625,0.030837736918074996,0.032879125464093335,-0.010596149355613312},
{0.22955007474736042,0.712278154508291,0.6286140377623632,-0.027883224817053298,-
0.18636280288375243,0.030730569944998218,0.03276486427940397,-0.010559325730853584},
{0.22369967248596626,0.6941247571266381,0.6125929365184771,-0.02717258213177352,-
0.18161308818803365,0.029947358717129894,0.03192980623690293,-0.01029020665867479},
{0.17477467051633444,0.5423138280707568,0.47861370314418994,-0.021229709620860862,-
0.14189277658199853,0.023397619202813948,0.024946488757399386,-0.008039651816779948},
{0.2304098303100254,0.714945917113802,0.630968445254194,-0.02798765849091672,-
0.18706080508056933,0.030845668049336264,0.03288758161833398,-0.010598874570230288},
{0.22954261053755376,0.7122549935766651,0.6285935973113181,-0.027882318146731564,-
0.1863567429813001,0.03072957068842295,0.03276379887429993,-0.010558982376477602},
{0.21274990853004683,0.6601483897854056,0.5826074296940904,-0.025842524930038354,-
0.17272339950469712,0.028481480357042472,0.030366890039632697,-0.009786516453329543},
{0.21258897507219016,0.6596490242918985,0.5821667196186714,-0.0258229764990873,-
0.17259274387185988,0.02845993570327313,0.030343919178436138,-0.00977911349863928},
{0.1959375983528994,0.6079809431871892,0.5365675658595804,-0.023800349928007187,-
0.1590741369156406,0.026230765020075313,0.027967182429937013,-0.00901314855247381},
};

String resultado2="";
double vetorVarianciaEnergias[] = new double[8];
double vetorEnergiaAux[];
int posicaoResultado=0;

double filtro[] = new double[8];
int comprimento_do_sinal_original=8;
int comprimento_do_filtro=8;
int comprimento_do_sinal_filtrado=comprimento_do_sinal_original+comprimento_do_filtro-1;
double y[]=new double[comprimento_do_sinal_filtrado];
double u[]=new double[comprimento_do_sinal_original]; //sinal original
double vetorResultados[];
public static void main(String[] args) {
    new testes0208v1().testar();
}

private void testar() {
try{
    System.out.println("inicio");
    System.out.println("Normais: " + matrizFiltrosNormais.length);
    Vector cabecalho = new Vector();
    cabecalho.add("Nome arquivo");
    cabecalho.add("Tipo");
    cabecalho.add("FTodos1");
    cabecalho.add("FTodos2");

//Normais
    diretoriopaciente = new File("D:\\DOC\\BD - Testes Finais - Todos\\Normais geral\\");
    //diretoriopaciente = new File("D:\\DOC\\BD - Testes Finais\\Normais geral\\");
    System.out.println("Dir: " + diretoriopaciente.getName());

    File[] arquivospaciente = diretoriopaciente.listFiles();
    for (int arqpaciente = 0; arqpaciente < arquivospaciente.length; arqpaciente++) {
        System.out.println("Arquivo: " + arqpaciente + " Nome: " + arquivospaciente[arqpaciente]);
        abrirArquivo(arquivospaciente[arqpaciente].toString());
        String nomeArquivo[] = arquivospaciente[arqpaciente].toString().split("/");
        linha = new Vector();
        sLinha += "\n";
        linha.add(nomeArquivo[nomeArquivo.length-1]);
    }
}
}

```

```

        sLinha += nomeArquivo[nomeArquivo.length-1]+"*";
        linha.add("Normal");
        sLinha += "Normal*";

    resultado="";
    eo = energia(vetorVarianciaEnergias);

    posicaoResultado=0;
    vetorResultados =new double[matrizFiltrosNormais.length+matrizFiltrosEdemas.length+matrizFiltrosNodulos.length];
    passarTodosFiltros();
    //gravarDisco(resultado, System.getProperty("user.dir") +"\\testes 19-07-2012\\teste1.txt");

    System.out.println("Todos os Filtros");
    encontraMelhoresResultados();
    dados.add(linha);
    }

//Edemas
    diretoriopaciente = new File("D:\\DOC\\BD - Testes Finais - Todos\\Edemas\\");
    System.out.println(" + Dir: " + diretoriopaciente.getName());

    arquivospaciente = diretoriopaciente.listFiles();
    for (int arqpaciente = 0; arqpaciente < arquivospaciente.length; arqpaciente++) {
        System.out.println(" + Arquivo: " + arqpaciente + " Nome: " + arquivospaciente[arqpaciente]);
        abrirArquivo(arquivospaciente[arqpaciente].toString());
        String nomeArquivo[] = arquivospaciente[arqpaciente].toString().split("/");
        linha = new Vector();
        sLinha += "\n";
        linha.add(nomeArquivo[nomeArquivo.length-1]);
        sLinha += nomeArquivo[nomeArquivo.length-1]+"*";
        linha.add("Edema");
        sLinha += "Edema*";

    resultado="";
    eo = energia(vetorVarianciaEnergias);

    posicaoResultado=0;
    vetorResultados =new double[matrizFiltrosNormais.length+matrizFiltrosEdemas.length+matrizFiltrosNodulos.length];
    passarTodosFiltros();
    //gravarDisco(resultado, System.getProperty("user.dir") +"\\testes 19-07-2012\\teste1.txt");

    System.out.println("Todos os Filtros");
    encontraMelhoresResultados();
    resultado+="\n";

        dados.add(linha);
    }

//Nodulos
    diretoriopaciente = new File("D:\\DOC\\BD - Testes Finais - Todos\\Nódulos\\");
    System.out.println(" + Dir: " + diretoriopaciente.getName());

    arquivospaciente = diretoriopaciente.listFiles();
    for (int arqpaciente = 0; arqpaciente < arquivospaciente.length; arqpaciente++) {
        System.out.println(" + Arquivo: " + arqpaciente + " Nome: " + arquivospaciente[arqpaciente]);
        abrirArquivo(arquivospaciente[arqpaciente].toString());
        String nomeArquivo[] = arquivospaciente[arqpaciente].toString().split("/");
        linha = new Vector();
        sLinha += "\n";
        linha.add(nomeArquivo[nomeArquivo.length-1]);
        sLinha += nomeArquivo[nomeArquivo.length-1]+"*";
        linha.add("Nodulo");
        sLinha += "Nodulo*";

    resultado="";
    eo = energia(vetorVarianciaEnergias);

    posicaoResultado=0;
    vetorResultados =new double[matrizFiltrosNormais.length+matrizFiltrosEdemas.length+matrizFiltrosNodulos.length];
    passarTodosFiltros();

    System.out.println("Todos os Filtros");
    encontraMelhoresResultados();
    resultado+="\n";

```

```

        dados.add(linha);
    }

    gravarDisco(sCabecalho+"\n"+sLinha, "D:\\DOC\\BD - Testes Finais\\resultados15-08Todos.txt");
    System.out.println("fim");
} catch (Exception erro) {
    JOptionPane.showMessageDialog(null, "Ocorreu um erro: " +
        erro.getMessage(), "Erro", JOptionPane.INFORMATION_MESSAGE);
}
}

public void gravarDisco(String dado, String nome) {
    try {
        FileOutputStream fos = new FileOutputStream(new File(nome), true);
        fos.write(dado.getBytes());
        fos.close();
    } catch (Exception ex) {
        System.err.println("ERRO" + ex.getMessage());
    }
}

public void abrirArquivo(String url) {
    try {
        wave = new Wave();
        abrirWave = new AbrirWave(wave);
        abrirWave.abrir(url);
        abrirWave.cabecalho();

        for (int i = 0; i < wave.getTamanhoChunk(); i++) {
            if (wave.getDoubleData()[i] >= 65535.0) {
                wave.getDoubleData()[i] = 0.0;
            }
        }

        if (wave.getNumCanais().equals("Mono")) {
            double[] doubleData = new double[wave.getTamanhoChunk() / 2];
            for (int c = 0; c < (wave.getTamanhoChunk() / 2); c++) {
                doubleData[c] = wave.getDoubleData()[c];
            }
            wave.setDoubleData(doubleData);
            wave.setTamanhoChunk(wave.getTamanhoChunk() / 2); // alterando o tamanho do chunk
        }

        // Achando a maior amplitude em módulo
        double maior_amplitude = 0;
        for (int i = 0; i < wave.getTamanhoChunk(); i++) {
            if (wave.getDoubleData()[i] > maior_amplitude) {
                maior_amplitude = wave.getDoubleData()[i];
            }
        }

        // Dividindo as amostras pelo módulo da maior delas
        double[] aux = new double[wave.getTamanhoChunk()];

        for (int i = 0; i < wave.getTamanhoChunk(); i++) {
            aux[i] = (wave.getDoubleData()[i] / maior_amplitude);
            if (i > aux.length - (0.01 * aux.length)) {
            }
        }
        wave.setDoubleData(aux);

        // Achando o valor médio do vetor e subtraindo de cada amostra
        double total = 0;
        for (int i = 0; i < wave.getTamanhoChunk(); i++) {
            total = total + wave.getDoubleData()[i];
        }
        aux = new double[wave.getTamanhoChunk()];
        for (int i = 0; i < wave.getTamanhoChunk(); i++) {
            if (wave.getDoubleData()[i] != 0) {
                aux[i] = wave.getDoubleData()[i] - (total / wave.getTamanhoChunk());
            }
        }
    }
}

```

```

wave.setDoubleData(aux);

// Calculando o nível máximo que se pode decompor o vetor
int n = 0;
n = (int) (Math.log(wave.getTamanho()) / Math.log(2));

// Decompondo o sinal usando transformada Wavelet-packet nível N
Wavelet transf = new Wavelet();

n = 10;

wave.separarJanelas(1024);

Wave waveJanelado[] = new Wave[wave.getQtdeJanelas()];

// calculando a escala Bark
bark = 0;

int maxfrequencia = Integer.parseInt(wave.getTaxaAmostragem()) / 2;
maxfrequencia = 1024;
if (maxfrequencia >= 0 && maxfrequencia < 100) bark = 0;
if (maxfrequencia >= 100 && maxfrequencia < 200) bark = 1;
if (maxfrequencia >= 200 && maxfrequencia < 300) bark = 2;
if (maxfrequencia >= 300 && maxfrequencia < 400) bark = 3;
if (maxfrequencia >= 400 && maxfrequencia < 510) bark = 4;
if (maxfrequencia >= 510 && maxfrequencia < 630) bark = 5;
if (maxfrequencia >= 630 && maxfrequencia < 770) bark = 6;
if (maxfrequencia >= 770 && maxfrequencia < 920) bark = 7;
if (maxfrequencia >= 920 && maxfrequencia < 1080) bark = 8;
if (maxfrequencia >= 1080 && maxfrequencia < 1270) bark = 9;
if (maxfrequencia >= 1270 && maxfrequencia < 1480) bark = 10;
if (maxfrequencia >= 1480 && maxfrequencia < 1720) bark = 11;
if (maxfrequencia >= 1720 && maxfrequencia < 2000) bark = 12;
if (maxfrequencia >= 2000 && maxfrequencia < 2320) bark = 13;
if (maxfrequencia >= 2320 && maxfrequencia < 2700) bark = 14;
if (maxfrequencia >= 2700 && maxfrequencia < 3150) bark = 15;
if (maxfrequencia >= 3150 && maxfrequencia < 3700) bark = 16;
if (maxfrequencia >= 3700 && maxfrequencia < 4400) bark = 17;
if (maxfrequencia >= 4400 && maxfrequencia < 5300) bark = 18;
if (maxfrequencia >= 5300 && maxfrequencia < 6400) bark = 19;
if (maxfrequencia >= 6400 && maxfrequencia < 7700) bark = 20;
if (maxfrequencia >= 7700 && maxfrequencia < 9500) bark = 21;
if (maxfrequencia >= 9500 && maxfrequencia < 12000) bark = 22;
if (maxfrequencia >= 12000 && maxfrequencia < 15500) bark = 23;
if (maxfrequencia >= 15500 && maxfrequencia < 220500) bark = 24;

energiaJanelas = new double[wave.getQtdeJanelas()][bark];
matrizDados = new double[wave.getQtdeJanelas()][8];

for (int l = 0; l < wave.getQtdeJanelas(); l++) {
    waveJanelado[l] = new Wave();
    waveJanelado[l].setDoubleData(wave.getDoubleDataJanela(l));
    waveJanelado[l].setTamanho(1024);
    waveJanelado[l].setTamanhoChunk(1024);
}
for (int janelas = 0; janelas < wave.getQtdeJanelas(); janelas++) {
    transf.transformada_wavelet_packet_nivel_k(waveJanelado[janelas], n, wave.HAAR);

    // Definindo a resolução
    double r = (Integer.parseInt(wave.getTaxaAmostragem())) / Math.pow(2, n + 1);

    // Definindo o vetor de características a partir da escala bark
    int inicio = 0;
    int fim = 0;
    for (int i = 0; i < bark; i++) {
        switch(i){
            case 0 : inicio=0; fim=(int)(100/r);break;
            case 1 : inicio=(int)(100/r)+1; fim=(int)(200/r);break;
            case 2 : inicio=(int)(200/r)+1; fim=(int)(300/r);break;
            case 3 : inicio=(int)(300/r)+1; fim=(int)(400/r);break;
            case 4 : inicio=(int)(400/r)+1; fim=(int)(510/r);break;
            case 5 : inicio=(int)(510/r)+1; fim=(int)(630/r);break;
            case 6 : inicio=(int)(630/r)+1; fim=(int)(770/r);break;
            case 7 : inicio=(int)(770/r)+1; fim=(int)(920/r);break;
            case 8 : inicio=(int)(920/r)+1; fim=(int)(1080/r);break;
        }
    }
}

```

```

        case 9 : inicio=(int)(1080/r)+1; fim=(int)(1270/r);break;
        case 10 : inicio=(int)(1270/r)+1; fim=(int)(1480/r);break;
        case 11 : inicio=(int)(1480/r)+1; fim=(int)(1720/r);break;
        case 12 : inicio=(int)(1720/r)+1; fim=(int)(2000/r);break;
        case 13 : inicio=(int)(2000/r)+1; fim=(int)(2320/r);break;
        case 14 : inicio=(int)(2320/r)+1; fim=(int)(2700/r);break;
        case 15 : inicio=(int)(2700/r)+1; fim=(int)(3150/r);break;
        case 16 : inicio=(int)(3150/r)+1; fim=(int)(3700/r);break;
        case 17 : inicio=(int)(3700/r)+1; fim=(int)(4400/r);break;
        case 18 : inicio=(int)(4400/r)+1; fim=(int)(5300/r);break;
        case 19 : inicio=(int)(5300/r)+1; fim=(int)(6400/r);break;
        case 20 : inicio=(int)(6400/r)+1; fim=(int)(7700/r);break;
        case 21 : inicio=(int)(7700/r)+1; fim=(int)(9500/r);break;
        case 22 : inicio=(int)(9500/r)+1; fim=(int)(12000/r);break;
        case 23 : inicio=(int)(12000/r)+1; fim=(int)(15500/r);break;
        case 24 : inicio=(int)(15500/r)+1; fim=(int)(22050/r);break;
    }

    for (int j = inicio; (j < fim && j < waveJanelado[janelas].getTamanho()); j++) {
        energiaJanelas[janelas][i] += waveJanelado[janelas].getDoubleData()[j] * waveJanelado[janelas].getDoubleData()[j];
    }
}

// Achando a maior amplitude em módulo
maior_amplitude = 0;
for (int i = 0; i < bark; i++) {
    if (energiaJanelas[janelas][i] > maior_amplitude) {
        maior_amplitude = energiaJanelas[janelas][i];
    }
}

for (int l = 0; l < wave.getQtdeJanelas(); l++) {
    for (int i = 0; i < bark; i++) {
        energiaJanelas[l][i] /= maior_amplitude;
        matrizDados[l][i] = energiaJanelas[l][i];
    }
}
} // transformadas para cada janela de 1 arquivo

for (int colunas=0;colunas<8;colunas++){
    vetorEnergiaAux = new double[wave.getQtdeJanelas()];
    for(int linhas=0;linhas<wave.getQtdeJanelas();linhas++){
        vetorEnergiaAux[linhas]=energiaJanelas[linhas][colunas];
    }
    vetorVarianciaEnergias[colunas]=calcularVarianciaEnergia(vetorEnergiaAux); // tem que separar por colunas, gerando outro
vetor
}
}
catch(Exception erro){
    JOptionPane.showMessageDialog(null, "Ocorreu um erro ao abrir WAVE: "+
        erro.getMessage(), "", JOptionPane.INFORMATION_MESSAGE);
}
}
double eo=0;

public void passarTodosFiltros(){

for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
    u[i] = vetorVarianciaEnergias[i]; }

    for(int posicaofiltro=0;posicaofiltro<matrizFiltrosNormais.length;posicaofiltro++) {

        resultado2+="*Nor"+posicaofiltro;
        for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
            filtro[i]=matrizFiltrosNormais[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);

```

```

        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros normais

    for(int posicaoFiltro=0;posicaoFiltro<matrizFiltrosEdemas.length;posicaoFiltro++) {
        resultado2+="*E"+posicaoFiltro;
        for (int i = 0; i< 8; i++) { // pega os dados da matriz e coloca no vetor
            filtro[i]=matrizFiltrosEdemas[posicaoFiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        //resultado+="*"+vetorResultados[posicaoResultado];
        posicaoResultado++;

    } // fim do for dos filtros edemas

    ##### Filtros nódulos
    for(int posicaoFiltro=0;posicaoFiltro<matrizFiltrosNodulos.length;posicaoFiltro++) {
        resultado2+="*Nod"+posicaoFiltro;
        for (int i = 0; i< 8; i++) { // pega os dados da matriz e coloca no vetor
            filtro[i]=matrizFiltrosNodulos[posicaoFiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;

    } // fim do for dos filtros edemas
}

public void passarFiltrosparaVozNormal(){
    for (int i = 0; i< 8; i++) { // pega os dados da matriz e coloca no vetor
        u[i] = vetorVarianciaEnergias[i]; }

    for(int posicaoFiltro=0;posicaoFiltro<matrizFiltrosNormais.length;posicaoFiltro++) {

        resultado2+="*Nor"+posicaoFiltro;
        for (int i = 0; i< 8; i++) { // pega os dados da matriz e coloca no vetor
            filtro[i]=matrizFiltrosNormais[posicaoFiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros normais

    for(int posicaoFiltro=0;posicaoFiltro<matrizFiltrosEdemas.length;posicaoFiltro++) {
        resultado2+="*E"+posicaoFiltro;
        if(ePraUsarPatologias(usarFiltrosPatologiasparaNormais, posicaoResultado)){
            for (int i = 0; i< 8; i++) { // pega os dados da matriz e coloca no vetor
                filtro[i]=matrizFiltrosEdemas[posicaoFiltro][i]; }
            for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
            {

```

```

        y[m]=0;
        for(int k=0;k<comprimento_do_filtro;k++)
            if(m-k>=0 && m-k<8)
                y[m]+=u[m-k]*filtro[k];
    }

    double et = energia(y);
    vetorResultados[posicaoResultado]=Math.abs(et-eo);
    //resultado+="*"+vetorResultados[posicaoResultado];
    posicaoResultado++;
    } else {
        vetorResultados[posicaoResultado]=1000;
        posicaoResultado++;
    }
}

} // fim do for dos filtros edemas

##### Filtros nódulos
for(int posicaofiltro=0;posicaofiltro<matrizFiltrosNodulos.length;posicaofiltro++) {
    resultado2+="*Nod"+posicaofiltro;
    if(ePraUsarPatologias(usarFiltrosPatologiasparaNormais, posicaoResultado)){
        for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
            filtro[i]=matrizFiltrosNodulos[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } else {
        vetorResultados[posicaoResultado]=1000;
        posicaoResultado++;
    }
}

} // fim do for dos filtros edemas
}

public void passarFiltrosparaVozEdema(){

for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
    u[i] = vetorVarianciaEnergias[i]; }

for(int posicaofiltro=0;posicaofiltro<matrizFiltrosNormais.length;posicaofiltro++) {

    resultado2+="*Nor"+posicaofiltro;
    for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
        filtro[i]=matrizFiltrosNormais[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros normais

for(int posicaofiltro=0;posicaofiltro<matrizFiltrosEdemas.length;posicaofiltro++) {
    resultado2+="*E"+posicaofiltro;
    for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
        filtro[i]=matrizFiltrosEdemas[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)

```

```

        y[m]+=u[m-k]*filtro[k];
    }

    double et = energia(y);
    vetorResultados[posicaoResultado]=Math.abs(et-eo);
    posicaoResultado++;

} // fim do for dos filtros edemas

##### Filtros nódulos
for(int posicaofiltro=0;posicaofiltro<matrizFiltrosNodulos.length;posicaofiltro++) {
    resultado2+="*Nod"+posicaofiltro;
    for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
        filtro[i]=matrizFiltrosNodulos[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros edemas
}

public void passarFiltrosparaVozNodulo(){

for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
    u[i] = vetorVarianciaEnergias[i]; }

for(int posicaofiltro=0;posicaofiltro<matrizFiltrosNormais.length;posicaofiltro++) {

    resultado2+="*Nor"+posicaofiltro;
    for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
        filtro[i]=matrizFiltrosNormais[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros normais

for(int posicaofiltro=0;posicaofiltro<matrizFiltrosEdemas.length;posicaofiltro++) {
    resultado2+="*E"+posicaofiltro;
    for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
        filtro[i]=matrizFiltrosEdemas[posicaofiltro][i]; }
        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros edemas

##### Filtros nódulos
for(int posicaofiltro=0;posicaofiltro<matrizFiltrosNodulos.length;posicaofiltro++) {
    resultado2+="*Nod"+posicaofiltro;
    for (int i = 0; i < 8; i++) { // pega os dados da matriz e coloca no vetor
        filtro[i]=matrizFiltrosNodulos[posicaofiltro][i]; }

```



```

        for(int m=0;m<comprimento_do_sinal_filtrado;m++) // faz a convolucao
        {
            y[m]=0;
            for(int k=0;k<comprimento_do_filtro;k++)
                if(m-k>=0 && m-k<8)
                    y[m]+=u[m-k]*filtro[k];
        }

        double et = energia(y);
        vetorResultados[posicaoResultado]=Math.abs(et-eo);
        posicaoResultado++;
    } // fim do for dos filtros edemas
}

```

```

public void encontraMelhoresResultados(){
    double valorMenor=0;
    int posicaoMenor=0;
    for(int i=0;i<vetorResultados.length;i++){
        if(i==0){
            valorMenor = vetorResultados[i];
            posicaoMenor=i;
        } else {
            if (valorMenor>vetorResultados[i]){
                valorMenor = vetorResultados[i];
                posicaoMenor=i;
            }
        }
    }
    System.out.println("Menor 1:" + posicaoMenor);
    resultado+="*"+posicaoMenor;
    linha.add(posicaoMenor);
    sLinha+=posicaoMenor+"*";

    double valorMenor2=0;
    int posicaoMenor2=0;
    for(int i=0;i<vetorResultados.length;i++){
        if(i==0){
            valorMenor2 = vetorResultados[i];
            posicaoMenor2=i;
        } else {
            if (valorMenor2>vetorResultados[i]){
                if(posicaoMenor!=i){
                    valorMenor2 = vetorResultados[i];
                    posicaoMenor2=i;
                }
            }
        }
    }
    System.out.println("Menor 2:" + posicaoMenor2);
    resultado+="*"+posicaoMenor2;
    linha.add(posicaoMenor2);
    sLinha+=posicaoMenor2+"*";

    double valorMenor3=0;
    int posicaoMenor3=0;
    for(int i=0;i<vetorResultados.length;i++){
        if(i==0){
            valorMenor3 = vetorResultados[i];
            posicaoMenor3=i;
        } else {
            if (valorMenor3>vetorResultados[i]){
                if(posicaoMenor!=i && posicaoMenor2!=i){
                    valorMenor3 = vetorResultados[i];
                    posicaoMenor3=i;
                }
            }
        }
    }
    resultado+="*"+posicaoMenor3;
    linha.add(posicaoMenor3);
    sLinha+=posicaoMenor3+"*";

    double valorMenor4=0;
    int posicaoMenor4=0;
    for(int i=0;i<vetorResultados.length;i++){
        if(i==0){
            valorMenor4 = vetorResultados[i];
            posicaoMenor4=i;
        }
    }
}

```

```

    } else {
        if (valorMenor4 > vetorResultados[i]) {
            if (posicaoMenor != i && posicaoMenor2 != i && posicaoMenor3 != i) {
                valorMenor4 = vetorResultados[i];
                posicaoMenor4 = i;
            }
        }
    }
}
System.out.println("Menor 4:" + posicaoMenor4);
resultado += "*" + posicaoMenor4;
linha.add(posicaoMenor4);
sLinha += posicaoMenor4 + "*";
}

public double calcularVarianciaEnergia(double sinal[]) {
    double variancia = 0, media = 0;
    for (int k = 0; k < sinal.length; k++) {
        media += sinal[k];
    }
    media = media / sinal.length;
    for (int i = 0; i < sinal.length; i++) {
        variancia += Math.pow(sinal[i] - media, 2);
    }
    return (variancia / (sinal.length));
}

private double energia(double Vetor[]) {
    double resp = 0;
    for (int j = 0; j < 8; j++) {
        resp += Math.pow(Vetor[j], 2);
    }

    return resp;
}
}

```


2. Trabalhos futuros

Já está em fase final de desenvolvimento o software para implantação de um prontuário eletrônico de pacientes e aquisição da voz, integrado com um sistema de apoio à decisão médica (Figura AP-1), a ser utilizado pelo Departamento de Otorrinolaringologia e Cirurgia de Cabeça e Pescoço do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto (FMRP-USP).

Prontuário: Status: << Selecione >>

Dados Pessoais | Dados de Contato | Dados Complementares | Hábitos Vocais | Análise da voz pré e pós prova de indução à fadiga vocal | Anamnese para Diagnóstico | Captura de Áudio

Cadastro de Paciente

Nome Completo:

Sexo: ☐ Masculino ☐ Feminino Data Nascimento: Estado Civil: << Selecione >>

RG: Órgão Expedidor: CPF:

Profissão:

Observações: << Selecione >>

Tipo de Necessidades Especiais: << Selecione >>

Novo Salvar Cancelar Alterar Excluir

Figura AP-1 - Interface do prontuário eletrônico do paciente

Nas figuras AP-2 (a e b) mostra-se a pesquisa dos prontuários, fazendo com que seja possível acessar as informações cadastradas anteriormente, como um histórico de visitas do paciente, com o qual se pode analisar o progresso do tratamento.

Interface de cadastro de pacientes com campos para busca por paciente (CPF ou Nome), nome completo, profissão, data de nascimento, estado civil, endereço principal, e contato. Inclui também uma seção para observações e tipo de necessidade especial.

Figura AP-2a. Tela de Cadastro de Pacientes

Interface para complementar dados do paciente, incluindo idade, peso, altura, e uma seção para a taxa de utilização vocal no dia-a-dia com uma escala deslizante. Também possui campos para sintomas e medicações utilizadas.

Figura AP-2b. Complementos de dados

Na Figura AP-3 (a e b), observa-se a interface para inserção dos dados complementares da entrevista (hábitos vocais) realizada com o paciente (chamados de parâmetros não acústicos para o diagnóstico) que podem influenciar fortemente o diagnóstico. E também os dados das Análises de pré/pós prova (coleta do áudio).

Interface para registrar hábitos vocais com sliders para: Abuso vocal (30%), Tosse persistente (60%), Resaca (40%) e Profissional (90%). Inclui uma legenda explicando a escala de 0 a 100.

Figura AP-3a. Tela de Cadastro de Hábitos Vocais

Interface para registrar dados de pré e pós prova, incluindo uma tabela com campos para data, tipo de prova, e resultados. Possui também sliders para avaliar a presença ou ausência de determinados fatores.

Figura AP-3b. Tela de Cadastro Pré/Pós Prova

O sistema integra várias telas de anamneses, que são perguntas que identificaram as possíveis causas do problema, podendo analisar se houve alguma influência (alergias, cigarros entre outras) nos distúrbios vocais.

Para a criação da tela de captura de voz (Figura AP-4) foi trabalhado um *layout* simples que fosse capaz de atender as necessidades do usuário que fará a coleta. Através do botão de gravação é possível capturar sons e ruídos em um determinado espaço de tempo com o auxílio de um dispositivo de entrada (Microfone), gerando um arquivo .wav (extensão padrão do

sistema operacional Windows para arquivos de áudio). Nesta mesma tela é possível gerar o resultado do diagnóstico através botão Resultado, o qual processa a voz e compara os dados com matrizes de valores de referência, classificando-a como Nódulo, Edema de Reinke ou Normal (dentro dos padrões).

Figura AP-4: Coleta e diagnóstico vocal

As demais telas foram elaboradas com o intuito de auxiliar o diagnóstico fonoaudiológico, no qual poderá considerar as alterações de acordo com os questionários respondidos e, junto aos demais exames e testes realizados é possível identificar as causas, realizando a adequação do melhor tratamento para a situação daquele paciente.

O sistema poderá emitir o relatório de evolução do paciente, contendo um histórico de suas passagens pela clínica, o que fará com que o médico avalie o progresso, caso contrário, torna-se mais evidente a necessidade de traçar novos tratamentos, visto que este não está resultando o desejado.