UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Luiz Felipe Lehmen Lersch

OTIMIZAÇÃO DO SISTEMA DE PROCESSAMENTO DE ÁUDIO DA PLATAFORMA EFONO

Santa Maria, RS
2022

Luiz Felipe Lehmen Lersch

OTIMIZAÇÃO DO SISTEMA DE PROCESSAMENTO DE ÁUDIO DA PLATAFORMA EFONO

Trabalho de Conclusão de Curso apresentado ao Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharelado em Ciência da Computação**

Orientador: Prof. Dr. Celio Trois

Co-orientador: Prof. Dr. João Carlos Damasceno Lima

513
Santa Maria, RS
2022

Lersch, Luiz Felipe Lehmen

Otimização do Sistema de Processamento de Áudio da Plataforma eFono / por Luiz Felipe Lehmen Lersch. – 2022.
22 f.: il.; 30 cm.

Orientador: Celio Trois
Co-orientador: João Carlos Damasceno Lima
Trabalho de Conclusão de Curso - Universidade Federal de Santa Maria, Centro de Tecnologia, Bacharelado em Ciência da Computação, RS, 2022.

1. Processamento de Áudio. 2. Avaliação Fonoaudiológica. 3. Recorte de Áudio. 4. Frequência de Sinal. I. Trois, Celio. II. Lima, João Carlos Damasceno. III. Título.

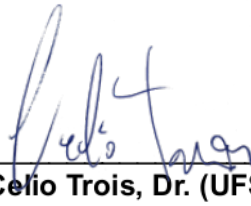
E-mail: lflersch@inf.ufsm.br

Luiz Felipe Lehmen Lersch

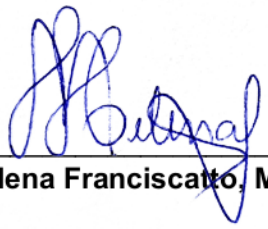
Otimização do Sistema de Processamento de Áudio da Plataforma eFono

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

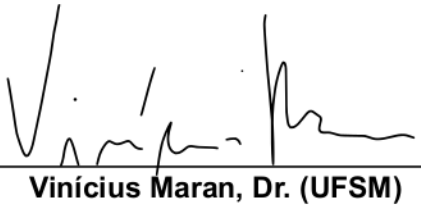
Aprovado em 18 de Fevereiro de 2022:



Celio Trois, Dr. (UFSM)
(Presidente/Orientador)



Maria Helena Franciscatto, Me. (UFSM)



Vinícius Maran, Dr. (UFSM)

Santa Maria, RS
2022

Otimização do Sistema de Processamento de Áudio da Plataforma eFono

Luiz F. L. Lersch[1], **Celio Trois**[1], **João C. D. Lima**[1]

[1]Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)

{lflersch, trois, caio}@inf.ufsm.br

Resumo. *O transtorno fonológico é definido como uma alteração de fala caracterizada pela produção inadequada dos sons e uso inadequado das regras fonológicas da língua. Nesse contexto, surge o aplicativo eFono, projetado para auxiliar fonoaudiólogos no processo de avaliação de fala. Para tanto, são gravados pelo profissional, com auxílio do aplicativo, áudios dos pacientes pronunciando palavras-chave. Com o uso do eFono, foram percebidos problemas no processamento de som, como lentidão e sons indesejados. Este trabalho propõe melhorar o processo de aquisição de áudio do eFono, otimizando o processamento e a classificação da pronúncia das palavras faladas pelas crianças. Foi desenvolvida uma abordagem que permite o recorte de áudio no aplicativo Android do eFono. Além disso, foi realizada uma pesquisa a respeito da taxa de amostragem da frequência dos áudios gravados, realizando testes com diferentes exemplares buscando encontrar a taxa de amostragem sem perda de qualidade, com o menor tamanho. Os resultados alcançados através do desenvolvimento da atividade de recorte proporcionam a diminuição do tamanhos dos arquivos de som, dessa forma reduzindo o tempo de envio dos áudios ao servidor e o seu armazenamento. Ainda, o desenvolvimento do fluxo dos testes de qualidade das frequências facilita a execução de futuros testes, os quais poderão usufruir de uma base maior de amostras.*

Palavras-chave: Processamento de Áudio; Avaliação Fonoaudiológica; Recorte de Áudio; Frequência de Sinal

Optimization of the eFono Platform Audio Processing System

Luiz F. L. Lersch[1], **Celio Trois**[1], **João C. D. Lima**[1]

[1]Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)

{lflersch, trois, caio}@inf.ufsm.br

Abstract. *The phonological disorder is defined as a speech sound disorder characterized by the inadequate production of sounds and inadequate use of the phonological rules of the language. In this context, the eFono application appears, designed to assist speech-language therapists in the speech evaluation process. To this end, audios of patients pronouncing keywords are recorded by the professional, with the help of the application. Using eFono, problems were noticed in sound processing, such as slowness and unwanted sounds. This work proposes to improve the eFono audio acquisition process, optimizing the processing and classification of the pronunciation of words spoken by children. An approach that allows audio clipping in the eFono Android application was developed. In addition, a research was carried out regarding the sampling rate of the frequency of recorded audios, carrying out tests with different samples seeking to find the sampling rate without loss of quality, with the smallest size. The results achieved through the development of the clipping activity provide a reduction in the sizes of sound files, thus reducing the time of sending the audios to the server and their storage. Furthermore, the development of the frequency quality test flow facilitates the execution of future tests, which will be able to benefit from a larger sample base.*

Keywords: Audio Processing, Phonological Assessment, Audio Clipping, Signal Frequency

1. Introdução

O reconhecimento de voz é uma área que vem crescendo mais rapidamente dentro do ramo da computação, além disso, é uma prática que tem aplicação em diferentes áreas, tais como saúde, educação, segurança, comunicação entre outras. Sua popularização se deve principalmente a sua facilidade de uso e o baixo custo [Vashisht et al. 2021]. Na área da saúde, especificamente, com a avaliação da frequência de áudio, o reconhecimento de voz tem auxiliado no campo de diagnóstico de problemas fonéticos e novos estudos sugerem que possa ser usado até mesmo no reconhecimento de emoções [Xue et al. 2018].

A Universidade Federal de Santa Maria (UFSM), atualmente, mantém o *eFono*, sistema que possui um aplicativo que utiliza o processamento de arquivos de som para realizar a triagem[1] fonológica infantil. O aplicativo funciona da seguinte maneira: primeiramente apresenta-se ao indivíduo objetos e ações, após isso é gravada a voz do paciente elicitando as palavras e, por fim, o sistema processa os áudios e realiza a triagem inicial dos pacientes [Gassen 2021]. O sistema eFono já é utilizado atualmente; periodicamente fonoaudiólogos coletam dados em consultas e, em seguida, os áudios dos pacientes são armazenados em nuvem para análise dos fonoaudiólogos.

Para realizar o processamento dos áudios primeiramente são gerados espectrogramas[2] a partir das gravações. Após isso, com a revisão por parte do profissional avaliando a fala como certa ou errada, são utilizadas técnicas de aprendizagem de máquina para que o sistema possa extrair características dos espectrogramas, aprender com esses dados e identificar padrões. Por fim, depois de realizado o treinamento, o sistema pode classificar a pronúncia de cada palavra da avaliação como correta ou incorreta.

Com o uso do aplicativo notou-se que, em determinadas condições de instabilidade ou lentidão na conexão com a Internet, o processo de envio dos áudios ao servidor acontecia de maneira lenta. Pra corrigir esse problema e deixar a aplicação mais fluida, uma alternativa desejável era alterar as propriedades dos arquivos de som, focando principalmente em duração do som e taxa de amostragem.

O recorte de áudio é uma técnica utilizada para extrair apenas o conteúdo relevante do áudio, retirando porções excedentes ou ruídos. No contexto do eFono, como o áudio é capturado durante a consulta, o arquivo pode conter, além da fala da palavra chave, as orientações do profissional e falas incompreensıveis da criança, objetos dispensáveis para a avaliação do fonoaudiólogo e que podem impactar negativamente na acurácia do processamento computacional [Franciscatto et al. 2021].

Já o controle da taxa de amostragem é um mecanismo que pode ser utilizado para reduzir o tamanho de um áudio, já que está associado ao número de amostras de sinal analógico utilizado para formar o áudio [Filomeno et al. 2003]. Quando muito alta, a amostragem pode deixar o arquivo com tamanho excessivo, necessitando maior largura de banda e tempo para sua transmissão. Uma amostragem muito baixa também oferece riscos, já que pode-se perder faixas essenciais para o processamento, podendo resultar em um arquivo de som distorcido e dificultar a identificação de padrões por parte do aprendizado de máquina.

[1] Processo no qual se define a prioridade do tratamento com base na gravidade do seu estado.

[2] gráfico que mostra a intensidade por meio do escurecimento ou coloração do traçado, as faixas de frequência no eixo vertical e o tempo no eixo horizontal

Considerando os benefícios que a manipulação de propriedades de áudio podem acarretar no sistema eFono, este trabalho apresenta uma estratégia de otimização do envio e processamento dos áudios na plataforma, propondo uma interface de recorte dos áudios capturados em meio a consultas, bem como o estudo de amostragens de frequência para futura manipulação. A atividade de recorte foi desenvolvida utilizando a linguagem de programação Java e incorporada ao aplicativo Android da plataforma eFono. Já o estudo de taxas de amostragem avaliou áudios gravados em avaliações fonoaudiológicas utilizando diferentes frequências.

Como resultado da pesquisa, os testes com as frequências tiveram desempenho insatisfatório, devido à baixa quantidade de áudios disponíveis na base de avaliações. Entretanto, o presente trabalho desenvolveu o fluxo e implementou o software que possibilitará a execução de novos testes à medida que novas avaliações forem feitas.

Já a implementação da atividade de recorte propiciou a diminuição do tamanhos das gravações e consequentemente a redução do tempo de envio dos áudios ao servidor. Essas gravações tem em média quatro segundos de duração, no entanto, a fala da criança dura em torno de um segundo. Sendo assim, espera-se que os arquivos fiquem com aproximadamente um segundo de duração após o processo de segmentação. Dessa maneira, com o uso da interface, calcula-se que o tamanho dos arquivos seja por volta de quatro vezes menor.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta o referencial teórico, com conceitos fundamentais para o entendimento do trabalho. Ainda na mesma seção, são apresentados trabalhos relacionados existentes na literatura. Na Seção 3 é apresentada a arquitetura do sistema eFono, sendo aprofundado o processo de captura de fala e classificação. A Seção 4 descreve a implementação da atividade de recorte, apresentando a arquitetura, seu fluxo, o desenvolvimento e tecnologias utilizadas. A seção ainda discorre sobre a avaliação da amostragem de frequência, de como foram realizados os testes e seus resultados. Por ultimo, a Seção 5 apresenta a conclusão desse trabalho, assim como atividades futuras relacionadas.

2. Referencial Teórico

Esta seção aborda conceitos essenciais para o entendimento deste trabalho, além disso, faz correlações com outros estudos que tratam do assunto principal ou de temas semelhantes. Para tanto, inicialmente, serão conceituados transtorno dos sons da fala, fonemas e o fluxo de avaliação fonológica. Após isso são introduzidos conceitos de processamento de fala e, por último, serão apresentadas propriedades de arquivos de som.

2.1. Avaliações Fonológicas

Este trabalho tem como objetivo a otimização do processamento de áudio do aplicativo voltado a avaliações fonológicas eFono. Para que isso seja feito, faz-se necessário que sejam entendidos alguns conceitos fonológicos. As regras do sistema de escrita alfabético usam do fato de que palavras faladas podem ser quebradas em unidades fonológicas definidas como fonemas [Mann 1993], definidos como uma unidade mínima de característica sonora distinta [Bloomfield 1933]. Seguindo esse raciocínio é possível exemplificar um fonema nas palavras 'avô' e 'avó', palavras da língua portuguesa que possuem as mesmas letras, mas são distinguidas pelos fonemas /ó/ e /ô/.

No primeiro ano de vida a criança descobre a própria voz e sua capacidade de se comunicar e, no final deste, inicia a produção das primeiras palavras, enquanto seu vocabulário aumenta progressivamente. Por volta dos dois anos é capaz de manter conversação com turnos[3] e aos três já está pronta para manter uma conversa coesa [Prates and Martins 2011]. No processo de aquisição da linguagem oral, a criança se familiariza com a estrutura da língua materna e organiza informações linguísticas necessárias. O desenvolvimento acontece de maneira majoritária por meio da experimentação dos diversos processos fonológicos na tentativa de aproximar a produção de fala à do adulto. A Tabela 1 apresenta alguns dos processos de aquisição da língua portuguesa e a idade esperada para a sua superação [Prates and Martins 2011].

Tabela 1. Exemplos de processos fonológicos e idade esperada para superação

Processo fonológico	**Idade máxima**	**Exemplos**
Plosivação de fricativa	18 meses	**F**ada - **p**ada
Frontalização de velar	36 meses	**C**asa - **t**asa
Simplificação de líquida	42 meses	Ca**r**eta - ca**l**eta/ca**i**eta/ca**e**ta
Simplificação da fricativa velar	42 meses	Ca**rr**o - cao/ca**l**o
Posteriorização para velar	42 meses	**T**a**t**u - **c**a**c**u
Posteriorização para palatal	54 meses	**S**apo - **ch**apo
Frontalização de palatal	54 meses	**Ch**apéu - **s**apéu

Fonte: adaptado de [Prates and Martins 2011]

O transtorno de fala é conceituado como um comprometimento da articulação dos sons, fluência e/ou voz [Association et al. 1993]. Quando uma criança supera a idade em que deveria pronunciar um processo fonológico de maneira correta, mas continua praticando-o, é possível que ela tenha um transtorno fonológico. Nesse contexto se faz fundamental a aplicação de avaliações por parte dos fonoaudiólogos, a fim de determinar se o paciente possui algum tipo de alteração fonológica.

Essas avaliações fonoaudiológicas são dependentes dos áudios dos pacientes e em sistemas de reconhecimento de fala, esse áudio é analisado com base em suas propriedades. Exemplos de propriedades serão discutidos a seguir.

2.2. Propriedades de Áudio

Um arquivo de som contém as variações de amplitude e frequência de uma onda sonora que são captadas por um microfone e depois transformadas em sinais elétricos que variam continuamente com o tempo. A amostragem é definida como os valores do sinal analógico que são amostrados periodicamente, ou seja, quantas vezes por segundo o som é registrado. Dessa forma, quanto maior a taxa de amostragem, menor é o intervalo entre cada amostra[Filomeno et al. 2003]. Ainda, o tamanho de um áudio é diretamente associado a sua duração e ao comprimento da sua taxa de amostragem.

[3]conversa na qual os participantes falam alternadamente

A Figura 1 apresenta 2 senoides[4], sendo a Figura 1.a utilizando uma representação de uma taxa de 1.200 Hz e a Figura 1.b de 15.360 Hz. A imagem ainda apresenta quanto tempo de sinal é representado em uma janela de 21 amostras, enquanto a primeira imagem representa aproximadamente 17 segundos, a segunda representa menos de 5. Assim sendo, já que a figura da esquerda consegue representar a mesma onda que a da direita com uma taxa de amostragem menor, é possível dizer que a segunda apresenta uma quantidade de amostras excessiva.

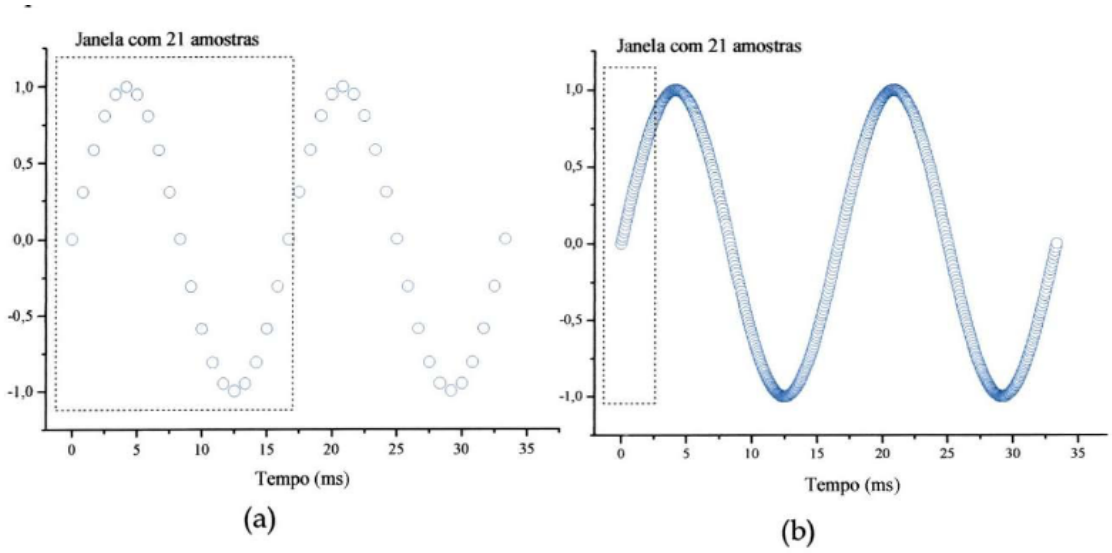


Figura 1. Frequência de onda. [Fontes et al. 2005]

Um fator importante a ser considerado quando escolhida a taxa de amostragem é o teorema de Nyquist [Nyquist 1928], que diz que a taxa de amostragem deve ser duas vezes maior que frequência máxima encontrada [Landau 1967]. Quando uma frequência é capturada acima da metade da taxa de amostragem, ocorre um efeito chamado de *aliasing*, caracterizado pela ambiguidade no sinal detectado, com a apresentação das maiores frequências de maneira distorcida [Carvalho et al. 2008].

Um exemplo de *aliasing* é apresentado na Figura 2. O sinal original é representado em vermelho, os pontos representam as amostras e a linha azul é o resultado final da conversão analógica-digital. É possível perceber uma distorção comparando-se o sinal original ao convertido. Essa distorção acontece pois a frequência de amostragem é menor que a frequência do sinal original, quando, seguindo o teorema de Nyquist, deveria ser no mínimo 2 vezes maior.

[4]curva matemática que descreve uma oscilação repetitiva suave

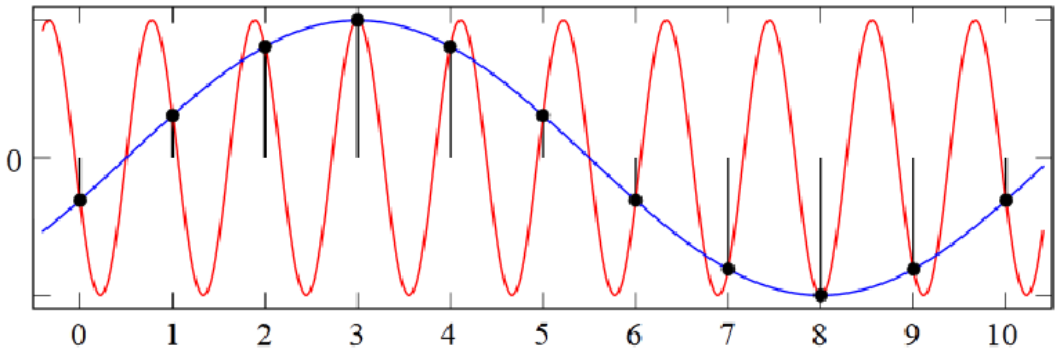


Figura 2. Aliasing. [Pavan 2022]

2.3. Frequência de Voz

Estudos afirmam que a frequência de voz fundamental humana difere principalmente pela idade e pelo sexo. Entre homens, a frequência fundamental é encontrada aproximadamente em 120hz, enquanto para as mulheres, ela é obtida em torno de 210hz [Traunmüller and Eriksson 1995]. Quanto à idade, nos homens a frequência começa a diminuir na puberdade em ritmo cada vez menor até perto dos 35 anos, aumentando novamente após os 55 anos [Hollien and Shipp 1972]. Nas mulheres, a frequência é estável até a menopausa, onde diminui aproximadamente 15hz até os 70 anos [Chevrie-Muller et al. 1971].

No entanto, alguns sons da fala podem chegar a frequências mais altas, como é o caso dos fricativos, que são vocalizados através de uma estreita constrição na cavidade oral, onde um fluxo rápido de ar cria turbulência e as flutuações de velocidade aleatória no fluxo agem como uma fonte de som [Jongman et al. 2000]. Esses sons têm intervalos de frequência altos, geralmente com o limite inferior por volta dos 1000hz e o superior podendo ultrapassar os 8000hz [Strevens 1960]. Os fonemas /f/, /s/, /ch/ são fricativos presentes na língua portuguesa exemplificados, respectivamente, nas palavras ”faca”[’fa.ka], ”sapo”[’sa.pu], ”chapéu”[cha.’pEw].

2.4. Trabalhos Relacionados

Recentemente foi desenvolvido um software que busca fornecer uma solução confiável e válida para fonoaudiólogos trabalharem com crianças com transtornos dos sons da fala, o *Table to Tablet (T2T)* [Jesus et al. 2015]. A proposta original contava com 18 exercícios diferentes inseridos em 8 áreas de atividade como rima, manipulação de fonemas e segmentação. Segundo os autores, foi utilizada uma abordagem web devido a popularização de dispositivos *mobiles* e sua compatibilidade com diversos sistemas operacionais.

Uma outra ferramenta projetada com o objetivo de auxiliar fonoaudiólogos é o INFONO, que disponibiliza um aplicativo *desktop* para avaliação de transtorno fonológico [Ceron 2015]. Ao final do estudo, obteve-se um programa com suporte a dezenove consoantes. Segundo a autora, por ser executado em computadores, o INFONO tem uma série

de vantagens como atratividade para crianças, utilização de desenhos animados e maior rapidez no processamento de dados.

No contexto de ferramentas de recorte de áudio, [Qian 2016] propôs um gravador e editor de áudio online, desenvolvido utilizando HTML, CSS e JavaScript. O programa disponibiliza ao usuário carregar áudios da Internet ou da própria máquina, fazer edições desse, e por fim salvá-lo novamente na Internet ou no computador.

No âmbito do controle de amostragem de frequência de sinais, [Fontes et al. 2005] propôs uma avaliação da influência da frequência de amostragem no desempenho de um identificador e classificador de faltas em sistemas elétricos. A proposta surgiu em meio a dificuldade de manipulação de dados quando trabalhado com uma frequência de amostragem elevada, pois a alta demanda de amostras a ser computada causava lentidão no processamento. Por fim, o autor concluiu a taxa de amostragem de frequência pode ser reduzida preservando as características do sinal.

A proposta de recorte apresentada nesse trabalho assemelha-se ao trabalho apresentado por [Qian 2016], porém utilizando desenvolvimento mobile ao invés de web. Já o estudo de frequências assemelha-se ao trabalho de [Fontes et al. 2005], diferindo quanto ao contexto, já que este trabalho lida com sinais do som da voz humana e o outro com sinais de sistemas elétricos. Visto que o sistema eFono é o contexto da presente proposta, esse será apresentado no próximo capítulo.

3. Sistema eFono

Esta seção tem o objetivo de apresentar o sistema eFono com o intuito de esclarecer o contexto onde este trabalho atuará, assim como os seus impactos no fluxo do sistema. A subseção a seguir apresentará a arquitetura do eFono e seus módulos componentes.

3.1. Arquitetura

A arquitetura é dividida em 4 módulos principais (Figura 3), sendo eles Aplicativo Android, API REST, Página WEB e Serviço de classificação. O módulo Aplicativo Android tem como função principal a disponibilização de uma interface para que o fonoaudiólogo possa aplicar os testes fonológicos. Além disso, ele também é encarregado da identificação do paciente e do envio das gravações ao servidor para a realização da triagem. Este módulo foi desenvolvido em [Moro 2018] e revisado em [Gassen 2021].

Proposto por [Almeida 2018] e posteriormente reestruturado em [Schmaedeck 2021], o módulo API REST é responsável por centralizar as informações do sistema. É ele que gerencia os pacientes e os testes fonológicos, além de armazenar as gravações capturadas pelo Aplicativo Android. A API REST também recebe dados criados no módulo Página Web, desenvolvido em [Schmaedeck 2021], o qual disponibiliza uma interface de criação de testes fonológicos, assim como outra para a revisão dos resultados da classificação.

O módulo Serviço de Classificação, desenvolvido em [Almeida 2018] com base no trabalho de [Franciscatto et al. 2018], utiliza de técnicas de inteligência artificial para classificar a pronúncia do paciente no áudio como correta ou incorreta. O resultado do processo de classificação, é enviado para o módulo API REST, deixando-o a disposição dos fonoaudiólogos para futuras consultas.

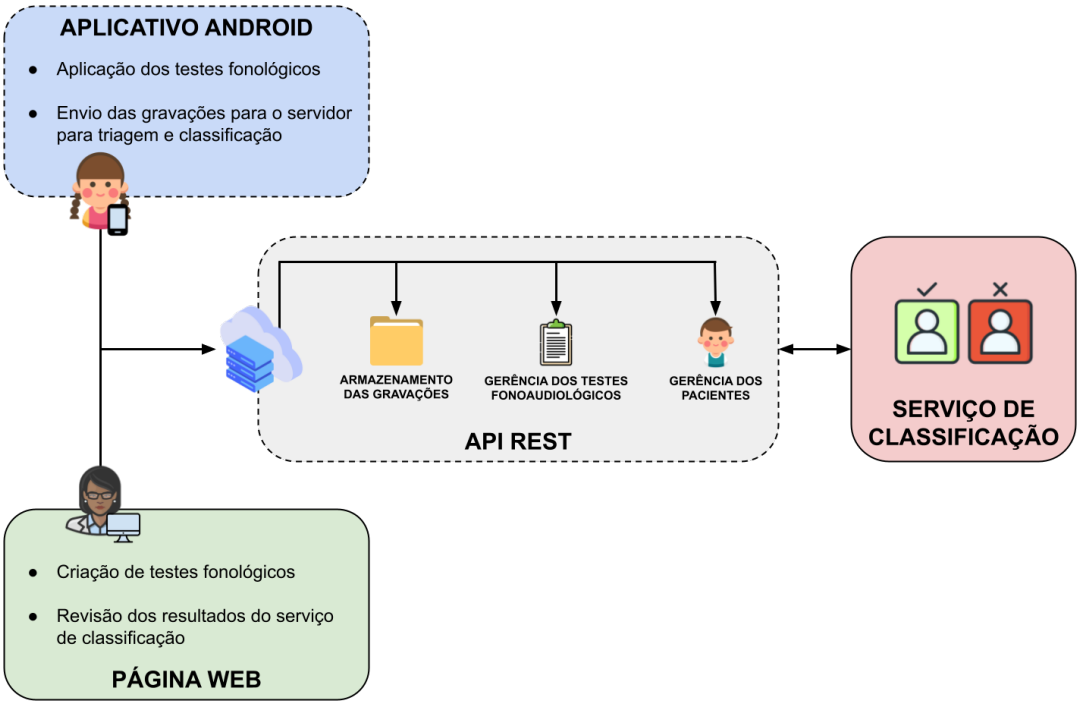


Figura 3. Arquitetura do projeto. Fonte: Adaptado de [Schmaedeck 2021]

Este trabalho propõe primeiramente a atividade de recorte de áudio. Esta atividade está localizada dentro do módulo Android (Figura 4). Além da diminuição do tamanho do áudio, a segmentação permite que somente a palavra falada da criança seja mantida. Dessa forma, é facilitada a comunicação Aplicativo Android-API REST e o processamento desse áudio pelo módulo Serviço de Classificação. Outra vantagem que a atividade traz é a redução do espaço necessário para armazenamento de áudios.

Além da atividade de recorte, esse trabalho ainda propõe um estudo de amostragem de frequência. Esse estudo consome áudios fornecidos pela API REST para realizar a avaliação de diferentes frequências. Essa avaliação é feita utilizando o módulo Serviço de Classificação para o processamento dos áudios. O estudo visa encontrar a melhor frequência de captura para, assim como a atividade de recorte, melhorar a comunicação Aplicativo Android-API REST e reduzir o espaço de armazenamento de áudios.

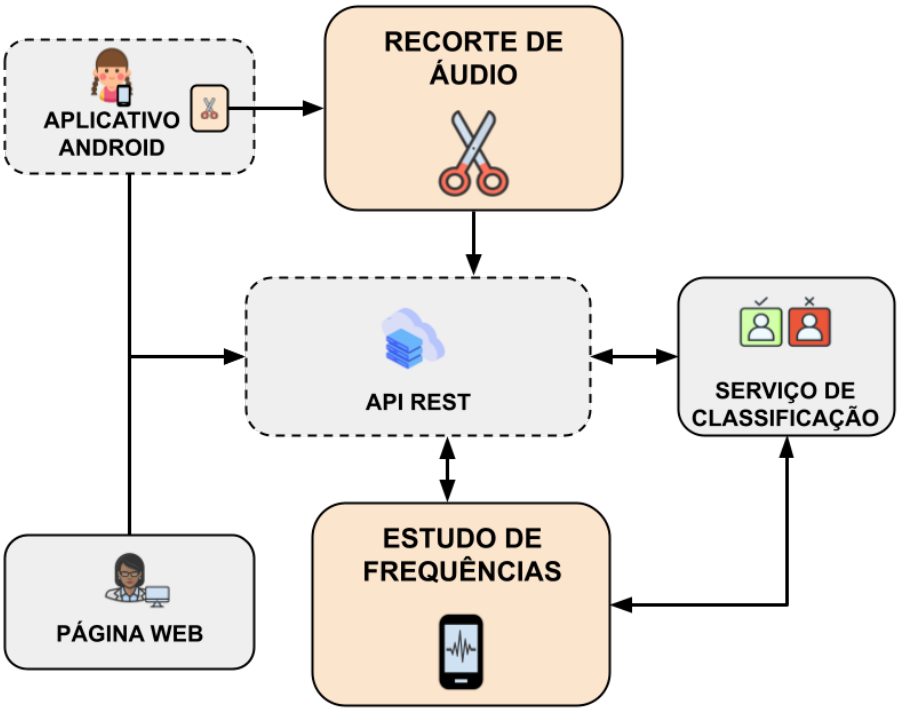


Figura 4. O presente trabalho na arquitetura do projeto. Fonte: do autor

A Figura 5 apresenta o fluxo de processamento de áudio desde a sua captura até a sua classificação. A primeira etapa do processo de captura e classificação ocorre no módulo Aplicativo Android, quando acontece a coleta os áudios dos testes fonológicos por parte dos fonoaudiólogos. O manuseamento da taxa de frequência terá influência na primeira etapa, onde o áudio é capturado através de um sinal analógico e tem que ser convertido para um sinal digital. A taxa de frequência definirá a quantidade de amostras por tempo que será utilizada na construção do sinal digital. O recorte de áudio ocorre na logo após essa construção, facilitando a próxima etapa, chamada de pré-processamento, onde são removidos ruídos e realizados cortes.

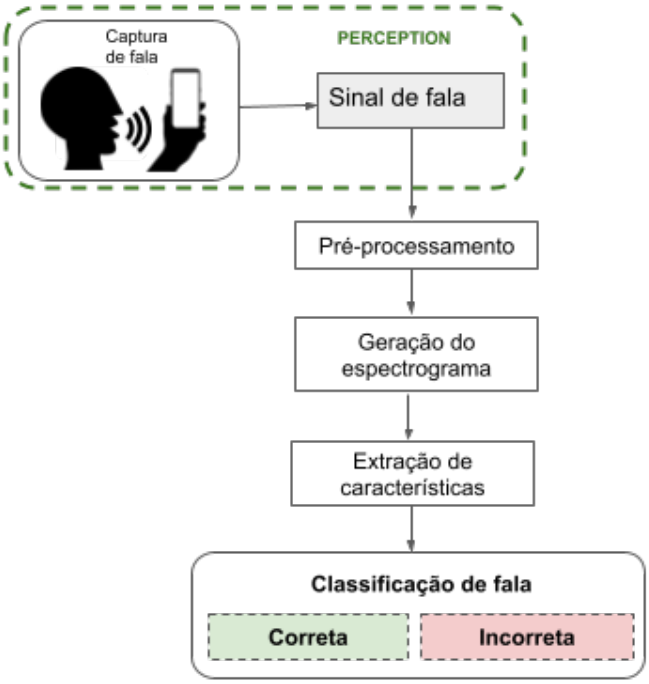


Figura 5. Captura de fala e Classificação. Fonte: [Franciscatto et al. 2018]

Após isso, com a geração de espectrogramas, o Módulo de Classificação é capaz de distinguir padrões na fala do indivíduo e, consequentemente detectar características que definem erros de pronúncia [Franciscatto et al. 2018]. Por fim, após analisadas as características da fala do indivíduo essa fala é classificada como correta ou incorreta.

4. Otimização no Envio e Processamento dos Áudios

Nesta seção serão abordados detalhes a respeito do desenvolvimento da atividade de recorte e sua implementação, assim como da aplicação dos testes de taxa de amostragem. O recorte dos arquivos de som será implementado na Aplicativo Android, sendo incorporado no processo de captura de fala. Seu principal impacto será no envio de áudios para o servidor, mais precisamente na transição do módulo Aplicativo Android para o módulo API REST.

4.1. Fluxo de Segmentação dos Áudios

O fluxo da segmentação dos áudios acontece como apresentado na Figura 6. Primeiramente o fonoaudiólogo realiza a gravação da criança pronunciando a palavra-alvo. No processo de gravação é comum a captação de instruções do profissional e falas indesejadas da criança, objetos dispensáveis para o processamento de áudio que ocupam em média três vezes mais espaço que a pronúncia da palavra alvo. Após a gravação, é então disponibilizada uma interface para que o fonoaudiólogo possa retirar as partes indesejadas do áudio, deixando apenas a pronúncia da palavra-alvo, diminuindo o tamanho do arquivo em torno de quatro vezes. Por fim, esse áudio recortado é enviado ao servidor para que possa ser processado.

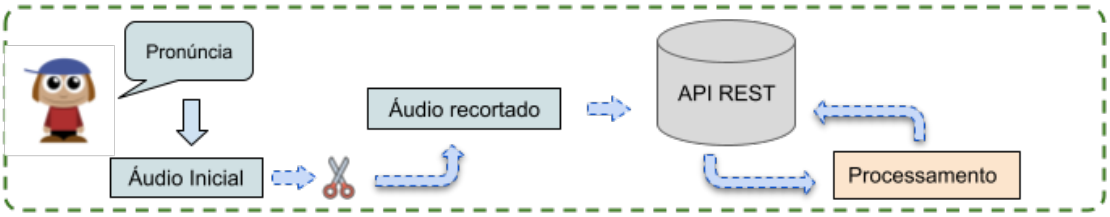


Figura 6. Fluxo da atividade de recorte. Fonte: do autor

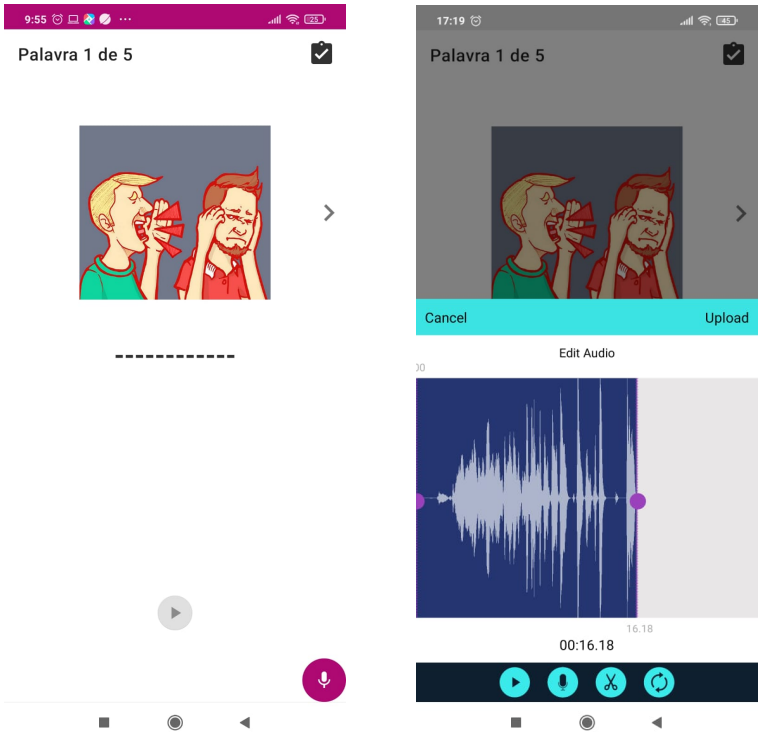
Na próxima subseção, a atividade de segmentação será abordada em detalhes, no contexto das interfaces necessárias para sua disponibilização e funcionamento.

4.1.1. Interfaces para segmentação de áudio

O fluxo do processo de recorte é composto pelas Figura 7.a e Figura 7.b, responsáveis pela coleta e recorde de áudio, respectivamente. A Figura 7.a mostra a tela de coleta da palavra 'gritar', que contém no seu canto inferior direito o botão de iniciar gravação. Esse componente é o responsável por iniciar a atividade de recorte. Ainda, essa é a tela responsável por manter o áudio ao fim do processo de recorte. Após a ativação do componente, o áudio começa a ser gravado. Finalizada a gravação é chamada a interface de recorte conforme mostra a Figura 7.b. Essa tela pode ser separada em três faixas de componentes: superior, central e inferior.

A faixa superior conta com um botão em cada lateral. No canto superior esquerdo está o botão de cancelar, que cancela o processo, e no canto superior direito está o botão de *upload*, que finaliza o processo e envia o áudio ao servidor. A faixa central conta com um espectrograma do áudio gravado e, ainda, duas linhas de cortes dinâmicas que se locomovem pelo espectrograma utilizando a técnica de *drag and drop* (Arrastar e soltar), que captura de movimento do usuário na tela, desde que ele toca na tela até o momento em que retira o objeto de toque. Essas linhas de corte representam a faixa de áudio que será retirada do arquivo de som assim que o botão de recortar for tocado.

Na faixa inferior estão disponíveis quatro botões. O primeiro botão, novamente simbolizado pela imagem de *'play'*, toca o áudio que está entre as linhas de corte do espectrograma, oferecendo uma demonstração de como ficará o arquivo após o recorte. O segundo, com o símbolo de um microfone, é encarregado de regravar o áudio quando acionado. O botão com ícone de tesoura, tem a função de recortar a faixa de áudio entre as linhas de corte. Já o último botão, representado por um símbolo de retorno, volta o áudio para o estado anterior, caso já tenha sido realizado algum corte.



a: Interface de avaliação

b: Interface de recorte de áudio

Figura 7. Telas do fluxo de recorte. Fonte: do autor

4.1.2. Implementação da segmentação de áudio

A aplicação mobile do eFono é baseada em Android, sistema operacional *open-source* baseado no *kernel* Linux, e foi desenvolvida utilizando a linguagem Kotlin[5]. Entretanto, aproveitando a compatibilidade Kotlin-Java fornecida pelo Android, e a maior amplitude de material disponível para linguagem Java[6], a interface de recorte foi desenvolvida utilizando essa última linguagem.

Primeiramente, para que o áudio pudesse ser capturado, foram solicitadas no arquivo de manifesto[7] as permissões 'RECORD_AUDIO', 'WRITE_EXTERNAL_STORAGE' e 'READ_EXTERNAL_STORAGE'. Essas permissões garantem que o usuário possa executar, na ordem, a gravação do áudio, a escrita e a leitura no armazenamento externo.

Após a solicitação de permissões foi criado o botão de iniciar gravação, representado no canto inferior direito da Figura 7.a. O componente foi inserido na interface de avaliação através do arquivo marcação (.xml), responsável pelo design da aplicação. Ao novo botão foi atribuída a função de acionar a nova interface de gravação e recorte de áudio. Essa função foi implementada utilizando a linguagem Kotlin e respeitando o padrão de arquitetura já existente (*Model-View-ViewModel*).

[5]Disponível em: https://kotlinlang.org/

[6]Disponível em: https://www.java.com

[7]arquivo com dados básicos sobre uma aplicação que ajudam a plataforma executar de forma apropriada.

Como mostra a Figura 8, a chamada da atividade de recorte é anexada ao evento *'trimEvent'* através do método *'observe'*. Esse evento é então associado ao botão da tela de avaliação no arquivo de layout da mesma. Para tratar o retorno da atividade, assim como a Figura 9 mostra, é verificado o código de retorno. Se o resultado for válido, o áudio é salvo pelo *viewModel*[8] da atividade que controla o fluxo de avaliação.

```
viewModel.evaluationStepLiveDatas.forEach { it: EvaluationStepLiveData
    it.trimEvent.observe( owner: this, { it: Void!
        startActivityForResult(
            Intent( packageContext: this, AudioTrimmerActivity::class.java),
            requestCode: 1001)

    })
}
```

Figura 8. Chamada da Atividade. Fonte: do autor

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode === 1001) {
        if (resultCode === Activity.RESULT_OK) {
            val path = data?.getStringExtra( name: "INTENT_AUDIO_FILE")
            viewModel.saveAudioPath(path)
        }
    }
}
```

Figura 9. Tratamento do retorno da Atividade. Fonte: do autor

A classe *AudioTrimmerActivity* é responsável pelo fluxo de gravação e de recorte. Para realizar a transição entre os estados é utilizada a característica visibilidade que faz parte dos componentes Android[9]. Dessa maneira, quando o estado deve ser trocado, a propriedade visibilidade é definida como *'VISIBLE'* aos componentes que devem aparecer, enquanto os que não serão utilizados tem a visibilidade definida como *'GONE'*. A interface pode ser dividida em dois estados principais, o de gravação e o de recorte.

A gravação dos áudios é realizada através da classe *SoundFile*, a qual contém todas as propriedades de arquivos de som. Após o fim da gravação ele é gravado no disco para que possa ser realizada a criação do espectrograma e posteriormente o recorte.

O recorte de áudio utiliza principalmente do componente nativo *MarkerView* para definir os tempos de aparação do áudio dentro do espectrograma. A Figura 10 apresenta um exemplo do uso do marcador na função que trata o seu movimento. As posições do marcador são guardadas para serem usadas posteriormente como tempos iniciais e finais do novo áudio. Quando o botão de *upload* é acionado, o áudio é então recortado com o conteúdo dentro dos marcadores e retornado para a aplicação principal.

[8]classe projetada para armazenar e gerenciar dados relacionados à IU considerando o ciclo de vida.
[9]Disponível em: https://www.android.com

```
public void markerTouchMove(MarkerView marker, float x) {
    float delta = x - mTouchStart;

    if (marker == markerStart) {
        mStartPos = trap((int) (mTouchInitialStartPos + delta));
        mEndPos = trap((int) (mTouchInitialEndPos + delta));
    } else {
        mEndPos = trap((int) (mTouchInitialEndPos + delta));
        if (mEndPos < mStartPos)
            mEndPos = mStartPos;
    }

    updateDisplay();
}
```

Figura 10. Exemplo de uso do marcador. Fonte: do autor

4.2. Pesquisa de Taxa de Amostragem de Frequência

Foi realizada uma pesquisa a respeito do desempenho de áudios com diferentes taxas de amostragem. Para isso foram realizados testes com o aprendizado de máquina utilizado na execução da triagem fonológica. O objetivo foi de diminuir o tamanho dos arquivos de som mantendo a qualidade do processamento, dessa maneira, facilitando o manuseamento desses na execução do processo de treinamento e a comunicação com o servidor.

Para realização dos testes, tomou-se como base o teorema de Nyquist, que mostra que a taxa de amostragem deve ser 2 vezes maior do que sua frequência máxima para que o sinal não apresente distorções. Conforme já mencionado, a frequência da fala humana tende a alcançar seu limite por volta de 8khz e somando-se ainda 2khz para eventuais anormalidades é encontrada uma suposta frequência ideal de 20khz para os testes.

Assim, os testes foram realizados considerando a suposta frequência ideal de 20khz e a frequência com que atualmente o aplicativo do eFono captura seus áudios, 44hz. Além disso, com o objetivo de testar uma frequência menor que a de 20khz e outra entre a de 20khz e a atualmente utilizada foram acrescentadas as frequências de 8khz e 32khz, ambas com intervalo de 12khz para a de 20khz.

Para a realização dos testes, o primeiro passo foi a aquisição das avaliações e seus áudios. Ao todo foram colhidos 4966 áudios de 88 avaliações diferentes do servidor do e-fono. Entretanto, nem todos áudios puderam ser utilizados pela falta de revisão fonoaudiológica nas avaliações. As revisões são necessárias para que o algoritmo de treinamento possa identificar se a palavra foi falada corretamente ou incorretamente e, dessa maneira, classificá-la. Pós treinamento, a revisão fonológica é utilizada novamente no teste de acurácia.

A aquisição das informações das avaliações foi feita através de uma consulta mongo, que por padrão retorna um arquivo BSON. Para facilitar a utilização dessas informações foi desenvolvido um programa em Java que possibilitasse a tradução BSON-JSON. No total foram colhidas 1145 revisões cobrindo um total de 84 palavras, divididas em 19 avaliações diferentes. Dessa maneira, como os testes foram feitos por palavra-

chave, cada teste teve em média 14 amostras.

Após a aquisição de todas as informações, o primeiro passo foi a conversão dos áudios gravados em 44khz para as outras frequências. Para tanto, foi utilizado o programa SoX[10], utilizado para manipulação de áudios através de linha de comando, passando como parâmetro o endereço do arquivo, a frequência desejada e o endereço de criação do novo arquivo.

Com todos os áudios disponíveis nas frequências desejadas, pode-se dar início ao processo de extração de características. Para isso, foi desenvolvido em Python um programa que, a partir dos áudios, gerasse os espectrogramas. A Figura 11 mostra um trecho de código onde se realiza a geração do espectrograma. Os espectrogramas são gerados inicialmente coloridos e são depois convertidos para escala cinza.

```
if os.path.exists(audioPath):
    print ("Thread DOING: "+ audioPath)
    rate, data = get_wav_info(audioPath)
    nfft = 256  # Length of the windowing segments
    fs = 256    # Sampling frequency
    pxx, freqs, bins, im = plt.specgram(data, nfft,fs)
    plt.axis('off')
    try:
        plt.savefig('espectrogramas/images_'+audioFolder+"/"+audio+'.png',dpi=100,frameon='false',aspect='normal',
                    bbox_inches='tight',pad_inches=0)
    except Exception as e:
        os.makedirs('espectrogramas/images_'+audioFolder+"/"+audio)
        #print '../images_'+path+"/"+fn
        plt.savefig('espectrogramas/images_'+audioFolder+"/"+audio+'.png',dpi=100,frameon='false',aspect='normal',
                    bbox_inches='tight',pad_inches=0)
    img = Image.open('espectrogramas/images_'+audioFolder+"/"+audio+'.png').convert('LA') # convert to grayscale
    img.save('espectrogramas/images_'+audioFolder+"/"+audio+'_gray.png')
```

Figura 11. Trecho da geração de espectrograma. Fonte: do autor

O último passo do processamento de dados foi a geração dos arquivos de treinamento através das características extraídas dos espectrogramas. Para isso foram usados os descritores baseados em LBP (padrão binários local), LBP uniforme [Ojala et al. 2002] e LBP girado [Zhao et al. 2013]. O programa de treinamento foi desenvolvido em Python e utilizou as informações dos espectrogramas junto com as revisões fonoaudiológicas contidas nas avaliações. Para o aprendizado de máquina foi utilizada a biblioteca Scikit-learn[11].

A Figura 12 mostra um exemplo prático de como funciona o fluxo de processamento de áudio. Primeiramente é gerado um espectrograma com base no arquivo de som. Após isso, com base no espectrograma, os descritores LBP constroem um histograma para o treinamento.

[10]Disponível em: http://sox.sourceforge.net/
[11]Disponível em: https://scikit-learn.org/stable/

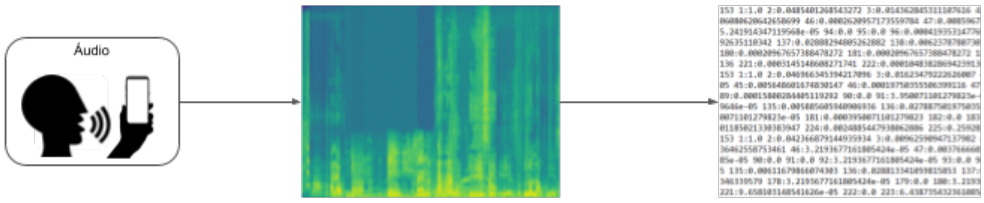


Figura 12. Fluxo de processamento de áudio. Fonte: do autor

Com todos os arquivos de aprendizado gerados foi preciso avaliá-los para saber como a frequência afeta o treinamento e qual teria o melhor desempenho. Para isso, foi desenvolvido, novamente em Python utilizando a biblioteca Scikit-learn, um programa que avaliasse os arquivos de treinamento. Já como divisão de amostras para testes e treinamento, foi utilizado a proporção 50-50, assim, tendo em vista que grande parte das palavras continham 14 amostras, foram utilizadas 7 amostras para treinamento e 7 para testes. Essa baixa quantidade de amostras, reflexo da falta de áudios munidos com revisão fonológica, refletiu diretamente no treinamento que, por esse motivo, apresentou uma baixa acurácia.

Para a aprendizagem foram utilizados três algoritmos como classificadores. Primeiramente, foram feitos testes com o classificador *Random Forest*. Esse algoritmo apresentou os piores resultados, tendo a frequência de 44khz o melhor desempenho com uma leve vantagem para a frequência de 20khz. As frequências de 32khz e 8khz com um resultado similar, tiveram os piores desempenhos.

Em seguida, foi usado o classificador *K-Nearest Neighbors*. Esse algoritmo apresentou resultados um pouco melhores que o anterior, mas também teve a frequência de 44khz o melhor desempenho com uma leve vantagem para a frequência de 20khz e as frequências de 32khz e 8khz com os piores desempenhos.

Por último, o classificador *Support Vector Machine* foi utilizado. Com o uso desse algoritmo todas as frequências tiveram um resultado similar e ficaram entre 23.2% e 23.8%. Foi o classificador que obteve a melhor média geral.

Tabela 2. Acurácia das frequências utilizando diferentes classificadores

Frequência	RF	KNN	SVM
44khz	16.8%	23.4%	23.6%
32khz	14.2%	20.6%	23.2%
20khz	15.4%	22.1%	23.8%
8khz	14.1%	21.6%	23.3%

Todos os testes apresentaram uma baixa acurácia, esse problema foi causado pela pouca quantidade de áudios revisados pelos fonoaudiólogos (a palavra mais revisada tinha 14 revisões). Essa quantidade de amostras é insuficiente para que o treinamento seja feito de forma eficiente. Sendo assim, por mais que uma amostra tenha apresentado resultados melhores, não é possível afirmar que uma taxa de frequência tenha um desempenho melhor que outra.

5. Conclusão

Com a utilização do aplicativo Android da plataforma eFono, desenvolvido com o objetivo de realizar uma triagem fonológica utilizando processamento de arquivos de som, parte dos profissionais fonoaudiólogos relataram lentidão no processo de envio de áudios ao servidor. Isso motivou a pesquisa por abordagens que pudessem solucionar o problema. Dentre as abordagens, optou-se por segmentar os áudios, deixando apenas as palavras pronunciadas pelas crianças. Além disso, com a realização de pesquisas a respeito das propriedades dos sons da fala e no seu processo de captura, compreendeu-se que através de testes com diferentes amostras, poderia ser avaliada a mudança na taxa de frequência com que os áudios são coletados. Ambas as abordagens poderiam, ainda, diminuir o tamanho do arquivo de áudio, reduzindo o tempo de envio dos mesmos para a plataforma eFono.

Este trabalhou implementou uma atividade de segmentação de áudio dentro do aplicativo Android. Essa atividade foi construída com o objetivo de facilitar o processamento de áudios, o envio desses ao servidor e o seu armazenamento. Ainda, a atividade foi desenvolvida em Java, para seu design foi utilizado um arquivo XML e para a chamada da atividade pelo aplicativo Android usou-se a linguagem Kotlin. Espera-se que, com o uso dessa interface de recorte, o tamanho dos áudios seja diminuído em torno de quatro vezes.

O trabalho também propôs uma pesquisa a respeito de diferentes taxas de frequência de áudio e seus desempenhos no processamento de áudio do sistema. Foi construído todo o fluxo de pré-processamento, geração de espectrograma, extração de características, treinamento e avaliação dos modelos de aprendizagem de máquina. Entretanto, devida a baixa quantidade de avaliações revisadas pelas fonoaudiólogas, os resultados não se mostraram válidos para se concluir a vantagem de uma frequência para outra.

Como trabalhos futuros, após a aplicação de novas avaliações, a gravação de novos áudios e, por fim, a execução da revisão desses áudios, pretende-se utilizar novamente o fluxo construído. Dessa maneira, poderão ser feitos testes com maior índice de confiança, já que a quantidade de amostras será maior. Ainda, pretende-se realizar testes de usabilidade para avaliar a atividade de recorte dos áudios, garantindo que a atividade possa se adequar tanto ao sistema quanto ao profissional fonoaudiólogo.

Referências

Almeida, A. T. R. (2018). Desenvolvimento de uma api rest para um sistema de auxílio na triagem.

Association, A. S.-L.-H. et al. (1993). Definitions of communication disorders and variations.

Bloomfield, L. (1933). Language history: From language (1933 ed.). In Hoijer, H., editor, *Language History: From Language*. Holt.

Carvalho, C. F., Chammas, M. C., and Cerri, G. G. (2008). Princípios físicos do doppler em ultra-sonografia. *Ciência Rural*, 38:872–879.

Ceron, M. I. (2015). Instrumento de avaliação fonológica (infono): desenvolvimento e estudos psicométricos. *Santa Maria: Universidade Federal de Santa Maria*.

Chevrie-Muller, C., Salomon, D., and Ferrey, G. (1971). Contribution a l'établissement de quelques constantes physiologiques de la voix parlée de la femme adolescente, adulte et age. *Journal Français d'Oto-Rhino-Laryngologie*, 16:433–455.

Filomeno, M. J. B. et al. (2003). Formato de arquivos de som na internet: uma visão contemporânea-usos, expectativas e tendências.

Fontes, A. V. et al. (2005). Avaliação da influência da frequência de amostragem no desempenho do neuranálise.

Franciscatto, M. H., Del Fabro, M. D., Lima, J. C. D., Trois, C., Moro, A., Maran, V., and Keske-Soares, M. (2021). Towards a speech therapy support system based on phonological processes early detection. *Computer speech & language*, 65:101130.

Franciscatto, M. H. et al. (2018). Sistema de recomendação para triagem de desordens dos sons da fala infantil baseado em um modelo de consciência de situação.

Gassen, F. H. d. S. (2021). Sistema de identificação digital e refatoração aplicadas em um aplicativo para avaliações fonológicas.

Hollien, H. and Shipp, T. (1972). Speaking fundamental frequency and chronologic age in males. *Journal of speech and hearing research*, 15(1):155–159.

Jesus, L. M., Santos, J., Martinez, J., Lousada, M., and Pape, D. (2015). The table to tablet (t2t) therapy software development approach. In *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–4. IEEE.

Jongman, A., Wayland, R., and Wong, S. (2000). Acoustic characteristics of english fricatives. *The Journal of the Acoustical Society of America*, 108(3):1252–1263.

Landau, H. (1967). Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706.

Mann, V. A. (1993). Phoneme awareness and future reading ability. *Journal of learning Disabilities*, 26(4):259–269.

Moro, A. (2018). Aplicação mobile para triagem fonológica infantil.

Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987.

Pavan, P. T. Z. (2022). Introdução a aquisição e processamento de sinais. https://edisciplinas.usp.br/pluginfile.php/4235029/mod_resource/content/0/AulaIntrodutoria.pdf/. acessado em 30/01/2022.

Prates, L. and Martins, V. d. O. (2011). Distúrbios da fala e da linguagem na infância. *Revista Médica de Minas Gerais*, 21(4 Supl 1):S54–S60.

Qian, L. (2016). *AuO: audio recorder and editor on the web*. PhD thesis, Massachusetts Institute of Technology.

Schmaedeck, M. V. (2021). Sistema para construção e revisão de avaliações fonológicas voltadas a triagem de distúrbios dos sons da fala em crianças.

Strevens, P. (1960). Spectra of fricative noise in human speech. *Language and speech*, 3(1):32–49.

Traunmüller, H. and Eriksson, A. (1995). The frequency range of the voice fundamental in the speech of male and female adults. *Unpublished manuscript*.

Vashisht, V., Pandey, A. K., and Yadav, S. P. (2021). Speech recognition using machine learning. *IEIE Transactions on Smart Processing & Computing*, 10(3):233–239.

Xue, Y., Hamada, Y., and Akagi, M. (2018). Voice conversion for emotional speech: Rule-based synthesis with degree of emotion controllable in dimensional space. *Speech Communication*, 102:54–67.

Zhao, Y., Jia, W., Hu, R.-X., and Min, H. (2013). Completed robust local binary pattern for texture classification. *Neurocomputing*, 106:68 – 76.