Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Sergio Pinto Gomes Junior

Reconhecimento de Emoções em Sinais de Fala Usando Transferência de Aprendizado

Rio de Janeiro

2019

Sergio Pinto Gomes Junior

Reconhecimento de Emoções em Sinais de Fala Usando Transferência de Aprendizado

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sinais e Sistemas de Comunicações.

Orientadores: Prof. Dr. Michel Pompeu Tcheou

Prof. Dr. Flávio Rainho Ávila

Rio de Janeiro

2019

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

G633 Gomes Junior, Sergio Pinto.
Reconhecimento de emoções em sinais de fala usando transferência de aprendizado / Sergio Pinto Gomes Junior. – 2019.
99f.

Orientadores: Michel Pompeu Tcheou, Flávio Rainho Ávila.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia eletrônica - Teses. 2. Interação homem-máquina - Teses. 3. Redes neurais (Computação) - Teses. 4. Sistemas de processamento da fala - Teses. 5. Emoções - Teses. 6. Aprendizado do computador - Teses. I. Tcheou, Michel Pompeu. II. Ávila, Flávio Rainho. III. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. IV. Título.

CDU 004.032.26

Bibliotecária: Júlia Vieira – CRB7/6022

_____ _____

Assinatura Data

Sergio Pinto Gomes Junior

Reconhecimento de Emoções em Sinais de Fala Usando Transferência de Aprendizado

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sinais e Sistemas de Comunicação.

Aprovado em: 28 de Fevereiro de 2019

Banca Examinadora:

Prof. Dr. Michel Pompeu Tcheou (Orientador)

Faculdade de Engenharia - UERJ

Prof. Dr. Flávio Rainho Ávila (Orientador)

Faculdade de engenharia - UERJ

Prof. Dr. Amaro Azevedo de Lima

CEFET-RJ

Profa. Dra. Karla Tereza Figueiredo Leite

Instituto de Matemática e Estatística - UERJ

Prof. Dr. João Baptista de Oliveira e Souza Filho

COPPE - UFRJ

Rio de Janeiro

2019

AGRADECIMENTO

Agradeço aos meus orientadores, Michel Tcheou e Flávio Ávila, pela paciência, dedicação e ensinamentos desde a graduação.

Aos professores e funcionários do PEL, agradeço pelo comprometimento com que realizam seu trabalho, mesmo em meio à tantas dificuldades apresentadas nesse período.

À Carolina, minha esposa, por ter sido tão companheira e por todos os seus conselhos durante a elaboração deste trabalho.

Aos meus pais, pelo cuidado e dedicação para que eu pudesse chegar até aqui.

À UERJ, por ser a instituição que me abriu as portas ainda na graduação e me ensinou, além de uma profissão, a ser uma pessoa melhor.

Digo: o real não está na saída nem na chegada:
ele se dispõe para a gente é no meio da travessia

Guimarães Rosa

RESUMO

GOMES JUNIOR, Sergio Pinto. *Reconhecimento de Emoções em Sinais de Fala Usando Transferência de Aprendizado.* 99 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2019.

A fala tem se tornado um meio de interação entre o ser humano e os computadores cada vez mais importante. Visando tornar essa interação ainda mais natural, pesquisadores têm proposto diferentes sistemas de reconhecimento de emoções na fala. Na área de reconhecimento de emoções em sinais de fala, as redes neurais profundas vêm sendo foco de intensa investigação. Visto isso, neste trabalho foi avaliado o efeito da técnica de transferência de aprendizado e do aumento da base de dados na acurácia de uma rede neural convolucional residual para a predição de emoções, comparando-a com outras técnicas de classificação tais como: a ResNet sem pré-treino, o Modelo de Mistura de Gaussianas e a Rede Neural Probabilística. Para isto, foram utilizadas as amostras das classes Felicidade, Neutra, Raiva e Tristeza contidas nas bases de dados IEMOCAP e EmoDb visando o treino e teste dos sistemas propostos. Nos experimentos com o GMM foi alcançada uma taxa de reconhecimento de 85,77% para a base de dados EmoDb e 66,83% para a IEMOCAP. Já a rede probabilística desenvolvida nesse trabalho conseguiu classificar corretamente 79,64% das amostras de teste da base de dados EmoDb. Nos experimentos com a ResNet, foram gerados os espectrogramas dos sinais de voz para serem utilizados no lugar de imagens. Nesses experimentos foi observado que as técnicas de aumento da base e de transferência de aprendizado contribuem significativamente para um melhor reconhecimento das emoções. Nesse caso, a rede convolucional classificou corretamente 81,26% das amostras.

Palavras-chave: Reconhecimento de emoções; Fala; Redes Neurais Convolucionais; Transferência de Aprendizado.

ABSTRACT

GOMES JUNIOR, Sergio Pinto. *Speech Emotion Recognition using Transfer Learning.* 99 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2019.

Speech has become an increasingly important mean of interaction between humans and computers. Aiming at making this interaction even more natural, researchers have proposed different systems of speech emotion recognition. In this area, in recent years, deep neural networks have been the focus of research. Given this, in this work we investigated the effect of techniques of transfer learning and data augmentation on the accuracy obtained by a residual convolutional neural network, comparing it to other classification strategies like ResNet without pre-training, the Gaussian Mixture Model and the Probabilistic Neural Network. In addition, samples of the Happiness, Neutral, Anger and Sadness classes contained in two emotion datasets (IEMOCAP and EmoDb) were used for training and testing of the proposed systems. In the experiments with the GMM, a accuracy of 85.77 % was achieved for the EmoDb dataset, and 66.83 % for the IEMOCAP. On the other hand, the probabilistic network developed in this work was able to correctly classify 79.64 % of the test samples from the EmoDb dataset. In the experiments with ResNet, the spectrograms of the speech signals were generated to be used instead of images. In these experiments it was observed that the techniques of data augmentation and transfer learning greatly contribute to the result of the emotion recognition. Using those techniques, the convolutional network correctly classified 81.26 % of the test samples.

Keywords: Emotion Recognition; Speech; Convolutional Neural Networks; Transfer Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

IEMOCAP	Interactive Emotional Dyadic Motion Capture
EmoDb	Berlin Database of Emotional Speech
MFCC	Mel-Frequency Cepstral Coefficients
STFT	Short-Time Fourier Transform
GMM	Gaussian Mixture Models
PNN	Probabilistic Neural Network
CNN	Convolutional Neural Network
ReLU	Rectified Linear Units

SUMÁRIO

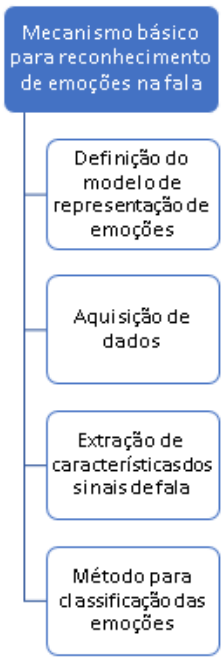
INTRODUÇÃO

A fala tem se tornado um meio de interação entre o ser humano e os computadores cada vez mais importante, dado que ela é o meio mais rápido e natural de comunicação. Este advento é consequente ao grande enfoque dado desde o final dos anos cinquenta à pesquisa relativa ao reconhecimento automático de fala por máquinas, o qual busca converter um discurso humano em uma sequencia de palavras [2]. Podemos perceber a evolução dessa área por meio do aumento da complexidade dos assistentes virtuais dos sistemas operacionais que utilizamos. Entretanto, nenhum desses assistentes consegue se comunicar com tanta naturalidade quanto um ser humano, pois ainda não possui a habilidade de compreender as emoções do falante ou usuário. A busca por essa naturalidade introduziu um campo de pesquisa relativamente recente, o reconhecimento de emoções na fala, entendido como a classificação automática do estado emocional do falante através do sinal proveniente de sua fala [2].

A tarefa de reconhecimento das emoções contidas na fala é muito desafiadora, especialmente pelas seguintes razões [2]: 1) não está claro quais características da fala são mais eficientes na distinção entre as emoções; 2) a variabilidade acústica introduzida pela existência de diferentes sentenças, falantes, estilos de fala e velocidades de fala afetam diretamente a maioria das características comuns da fala extraída, como o *pitch* e os contornos de energia [3]; 3) um mesmo enunciado pode conter mais de uma emoção, e cada emoção corresponde a uma porção diferente do enunciado falado, sendo bastante difícil determinar os limites entre essas partes; 4) a forma como uma determinada emoção é expressa sofre a influência das especificidades do falante, sua cultura e o meio onde vive; 5) um indivíduo pode se manter por dias num mesmo estado emocional, como a tristeza, e neste caso outras emoções são transitórias, durando não mais do que alguns minutos, gerando dúvidas sobre a identificação pelo detector automático dos estados emocionais de longo prazo ou transitórios. 6) a emoção não tem uma definição teórica comumente acordada [4], mas os indivíduos reconhecem as emoções quando as sentem, o que permitiu aos pesquisadores estudarem e definirem diferentes aspectos das emoções.

A maneira convencional de criar um mecanismo capaz de reconhecer a emoção da fala pode ser dividida em quatro principais etapas, que estão apresentadas na Figura 1 e serão detalhadas na sequência. Primeiramente, precisamos definir um modelo adequado

Figura 1: - Mecanismo convencional para reconhecimento de emoções contidas na fala



de representação de emoções [5]. Esta tarefa apresenta duas questões principais: como representar a emoção em si e como quantificar otimamente o eixo do tempo. Começando por representar a emoção de maneira adequada para garantir o encaixe adequado com a literatura psicológica enquanto escolhe uma representação que pode ser manuseada por uma máquina, dois modelos são geralmente encontrados na prática. O primeiro modelo é de classes discretas, como as "seis grandes" categorias de emoções de Ekman, incluindo raiva, desgosto, medo, felicidade e tristeza - frequentemente adicionados a uma classe "neutra" [6]. Já o segundo modelo possui uma abordagem de dimensão de valor contínuo e é formado por dois eixos: o eixo de ativação, conhecido por ser bem acessível em particular por características acústicas e indicado por comportamentos de alerta e escalas bipolares como calmo/excitado e acordado/sonolento; e o eixo de valência, que é conhecido por ser bem acessível por características linguísticas [7], sendo caracterizado por escalas bipolares como positivo/negativo [8]. A ativação se refere à quantidade de energia necessária para expressar uma certa emoção. De acordo com alguns estudos fisiológicos feitos por Williams e Stevens [9] sobre o mecanismo de produção de emoções, descobriu-se que o sistema nervoso simpático é estimulado pelas emoções da Alegria, Raiva e Medo. Isso induz um aumento da frequência cardíaca, aumento da pressão arterial, alterações na

profundidade dos movimentos respiratórios, maior pressão subglótica, secura da boca e, ocasionalmente, tremor muscular. A fala resultante é correspondentemente alta, rápida e enunciada com forte energia de alta frequência, um tom médio mais alto e maior amplitude de tons. Por outro lado, com o despertar do sistema nervoso parassimpático, assim como com a tristeza, a frequência cardíaca e a diminuição da pressão sanguínea aumentam e a salivação aumenta, produzindo uma fala lenta, baixa e com pouca energia de alta frequência. Assim, características acústicas, tais como o tom, o tempo, a qualidade da voz e a articulação do sinal de fala, correlacionam-se a emoção subjacente [10]. No entanto, as emoções não podem ser distinguidas usando apenas o eixo de ativação. Por exemplo, tanto a raiva quanto a felicidade são emoções que correspondem a alta-ativação, mas elas transmitem um efeito diferente. Essa diferença é caracterizada pela dimensão de valência. Infelizmente, não há acordo entre os pesquisadores sobre como, ou mesmo se, os recursos acústicos se correlacionam com essa dimensão [11]. Portanto, enquanto a classificação entre emoções de alta-ativação - também chamadas de alta-excitação - e emoções de baixa ativação pode ser alcançada com alta-precisão, a classificação entre emoções diferentes ainda são desafiadoras.

Uma vez que o modelo de representação das emoções foi definido, a próxima questão é a aquisição de dados com rótulos que seguem o modelo escolhido. Existem diversas bases de dados que foram criadas para a pesquisa nessa área. A Tabela 1 apresenta algumas dessas bases e suas características. Dela podemos observar que as bases diferem bastante entre si. Em um projeto de reconhecimento de emoções na fala é importante observarmos como a base foi projetada para avaliarmos se ela possui as emoções que queremos classificar, se ela foi criada na língua que escolhemos, e se ela possui um tamanho suficiente para utilizarmos o método de classificação escolhido. O grande desafio nessa etapa se dá pela subjetividade e a incerteza dos rótulos. Não é surpreendente, que até mesmo os seres humanos geralmente discordam em algum grau sobre qual emoção está presente no discurso de outras pessoas [12]. Em algumas bases de dados, o desempenho do reconhecimento humano gira em torno de apenas 65% [13]. Outra questão das bases de dados é o grau de naturalidade, dado que muitas são formadas por falas gravadas em situações atuadas. Esse formato possui um maior controle por parte dos pesquisadores, mas pode deixar a desejar na aproximação de emoções reais.

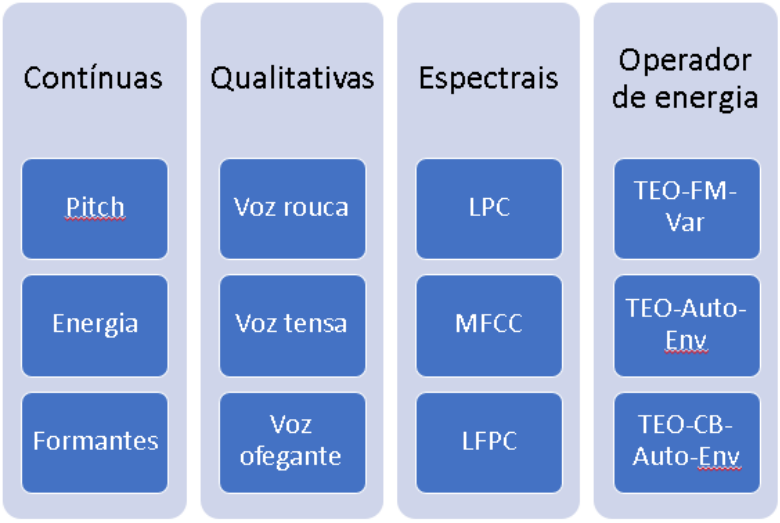
Outra questão importante no projeto de um sistema de reconhecimento de emoções

Tabela 1: - Características de algumas base de dados comuns

Nome	Acesso	Língua	Tamanho	Emoções
Berlin emotional database [14]	Público e gratuito	Alemão	800 sentenças	Raiva, alegria, tisteza, medo, desgosto, tédio, neutra
Baby ears [15]	Privada	Inglês	509 sentenças	Aprovação, atenção, proibição
SUSAS [16]	Público com taxa de licença	Inglês	16000 sentenças	*Estresse simulado, tarefa de rastreamento de carga de trabalho calibrada, tarefa de aquisição e rastreamento compensatório, gravações em uma montanha-russa e no cockpit de um helicóptero*
RECOLA [17]	Público e gratuito	Francês	9,5 horas de conteúdo	Em termos de ativação e valência
IEMOCAP [1]	Público e gratuito	Inglês	12 horas de conteúdo	Rótulos categóricos, como raiva, felicidade, tristeza, neutralidade, bem como rótulos dimensionais, como valência e ativação

de fala é a extração de características dos sinais que melhor reflitam o conteúdo emocional, que devem ser robustos contra ruídos, idiomas diferentes, ou até mesmo, influências culturais. A pesquisa dessas características é um subcampo muito importante dessa área, visto que as técnicas de reconhecimento de padrões raramente são independentes do domínio do problema, portanto uma seleção adequada de recursos afeta significativamente o desempenho da classificação. Nesse subcampo, os pesquisadores focam, principalmente, em quatro questões [2]. A primeira é a região de análise usada para a extração de recursos. Enquanto alguns pesquisadores seguem a estrutura normal de dividir o sinal de fala em pequenos intervalos, chamados de quadros, de onde cada vetor de característica local é extraído, outros preferem extrair estatísticas globais de todo o enunciado de fala [2]. Outro ponto importante é o estudo de quais grupos de características possuem um melhor desempenho para essa tarefa. As características da fala podem ser agrupadas em contínuas, qualitativas, espectrais ou baseadas no operador de energia Teager [2]. Baseados nos desempenhos dos sistemas encontrados na literatura, utilizamos neste trabalho os *Mel Frequency Cepstrum Coefficients* (MFCC), que fazem parte do grupo dos espectrais, como pode ser visto na Figura 2. Um terceiro ponto é o efeito do processamento de fala comum, tais como a pós-filtragem e a remoção de silêncio, no desempenho geral do classificador. Por fim, se é suficiente usar recursos acústicos para modelar emoções ou se é necessário combiná-los com outros tipos de recursos, tais como informações linguísticas,

Figura 2: - Categorias dos parâmetros da fala



de discurso, ou faciais [2]. Dadas todas essas questões, algumas pesquisas mais recentes utilizaram sistemas fim-a-fim, onde essas características são escolhidas automaticamente pelo sistema [5].

A última etapa consiste na escolha do método para classificação das emoções baseadas nos parâmetros da fala que foram extraídos na etapa anterior. Vários tipos de ferramentas de aprendizagem de máquina vem sendo utilizados para esta tarefa, tais como o *Hidden Markov Model* (HMM) - utilizado em [18] [19] [13], o *Gaussian Mixture Model* (GMM) - utilizado em [20] [15] [21], a *Suport Vector Machine* (SVM) - utlizado em [19] [18] [22] , as redes neurais artificiais (RNA) - utilizadas em [23] [24] [25], o algoritmo *K Nearest Neighbor* (kNN) [14] [26] - e, recentemente, as redes neurais profundas, como a *Convolutional Neural Network* (CNN) [27] e a *Recurrent Neural Network* [28] (RNN). De fato, não houve acordo sobre qual classificador é o mais adequado para a classificação de emoções, visto que cada classificador tem suas próprias vantagens e limitações. Os classificadores estatísticos são amplamente utilizados no contexto do reconhecimento de emoções na fala [2]. Na abordagem estatística para reconhecimento de padrões, cada classe é modelada por uma distribuição de probabilidade baseada nos dados de treinamento disponíveis. Os classificadores estatísticos vem sendo usados em muitas aplicações de reconhecimento de fala. Enquanto o HMM é uma das ferramentas mais utilizadas na tarefa de reconhecimento automático de fala, do Inglês *Automatic Speech Recognition*(ASR), o GMM é mais eficiente em identificação e verificação de falantes [29]. O

HMM e o GMM têm propriedades importantes, como a facilidade de implementação e sua sólida base matemática [2].

O reconhecimento da emoção contida na fala se mostrou particularmente útil em diversas aplicações como a utilização de aplicativos tutoriais, em que a resposta desses sistemas ao usuário depende da emoção detectada [30]. Também foi empregado como uma ferramenta de diagnóstico para terapeutas [31]. O reconhecimento de emoções de fala também tem sido usado em aplicativos de call center para determinar o estado emocional do cliente e ajustar o atendimento de acordo com o seu estado [32]. Outra possibilidade de aplicação é na monitoração do rádio de um agente de segurança no exercício de seu trabalho para mensurar seu nível de estresse e avaliar sua condição de continuar sendo exposto a atividades com alto nível de estresse envolvido.

Ao realizar uma revisão bibliográfica da área de reconhecimento de emoções na fala para analisar o que tem sido estudado nos últimos anos, podemos perceber, principalmente, um maior enfoque em comparar diferentes métodos de classificação, com um crescente interesse em redes neurais profundas e também uma tentativa de criar sistemas fim-a-fim, onde os parâmetros da fala são selecionados automaticamente pelo próprio sistema. Em [33], os autores pesquisaram a diferença entre o desempenho do GMM e do KNN. Para isso, eles implementaram dois sistemas de reconhecimento de emoções na fala utilizando como parâmetros os *Mel Frequency Cepstrum Coefficients* (MFCC), as *Wavelets*, e o *Pitch*, que são parâmetros espectrais. As categorias emocionais utilizadas no trabalho foram felicidade, raiva, neutra, surpresa, medo e tristeza contidas nas amostras de voz da base de dados *Berlin Emotion Database*. O sistema baseado em GMM foi o que apresentou o melhor desempenho entre os dois, reconhecendo corretamente 92% das amostras de teste rotuladas como raiva. Além dessa categoria, o GMM obteve um resultado superior no reconhecimento das emoções de tristeza, neutra e de medo. Já para a emoção felicidade, o KNN obteve um resultado melhor, 90% contra 67% do GMM. Para a emoção de surpresa, os dois métodos obtiveram o mesmo resultado, 25%. Já em [34], os pesquisadores desenvolveram um sistema de reconhecimento de emoções de fala fim-a-fim, baseado em redes neurais profundas: CNN e RNN, aplicadas diretamente a espectrogramas das amostras de áudio da base de dados *Interactive Emotional Dyadic Motion Capture* (IEMOCAP). Os pesquisadores desenvolveram dois sistemas: no primeiro, eles utilizaram somente uma rede neural convolucional e obtiveram uma acurácia de 66%, considerando

as quatro emoções. Já o segundo sistema, era formado por uma combinação de uma rede neural convolucional e um modelo de rede neural recorrente chamado de *Long Short-Term Memory* (LSTM), que atingiu uma acurácia de 68%.

Objetivo

O presente trabalho possui como objetivo a comparação, com outros métodos de classificação, do resultado da aplicação de uma rede neural convolucional em conjunto com as técnicas de transferência de aprendizado e aumento da base de dados para a tarefa de reconhecimento de emoções em sinais de fala. Para essa comparação, realizamos também experimentos com o Modelo de Mistura de Gaussianas e a Rede Neural Probabilística.

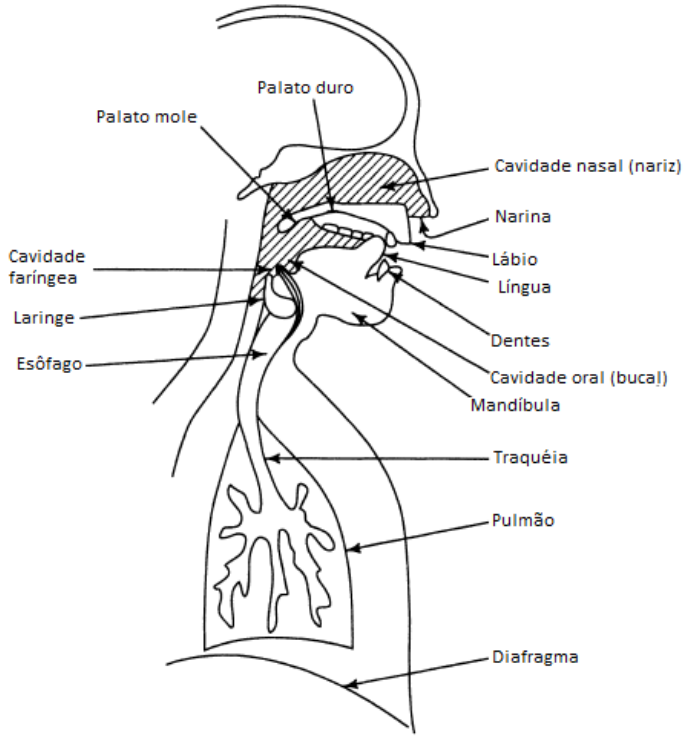
Organização do Texto

O primeiro capítulo deste trabalho introduz o processamento da fala e os coeficientes mel-cepstrais. Já no segundo capítulo, são apresentadas as duas bases de dados utilizadas nos experimentos realizados neste trabalho: a EmoDb e a IEMOCAP. Os capítulos 3, 4 e 5 apresentam a configuração dos experimentos utilizando o GMM, a PNN e a CNN, respectivamente, e as suas bases teóricas. Os resultados dos experimentos são apresentados e discutidos no Capítulo 6. Por fim, temos o capítulo de conclusão, que resume o que foi feito neste trabalho e as indicações para trabalhos futuros relacionados a esta pesquisa.

1 PROCESSAMENTO DA FALA E OS COEFICIENTES MEL-CEPSTRAIS

A fala nada mais é do que uma onda sonora de pressão acústica originada a partir de movimentos de estruturas anatômicas que compõem o sistema humano de produção da fala [35]. Como podemos ver na Figura 3, as estruturas que participam deste mecanismo são os pulmões, a traqueia, a laringe, a cavidade faríngea, a boca, e a cavidade nasal. A cavidade faríngea e a boca geralmente são tratadas como uma estrutura única conhecida como trato vocal. Já a cavidade nasal, pode ser também chamada de trato nasal. Outras estruturas com movimentos mais finos que também realizam um papel importante na produção da fala incluem as cordas vocais, o palato mole, a úvula, a língua, os dentes e os lábios. Estas estruturas, conhecidas como articuladores, podem se mover para diversas posições e, com isso, gerar sons de fala diferentes. A mandíbula, que também é considerada um articulador, é responsável por movimentos grossos e finos que afetam o tamanho e a forma do trato vocal, assim como as posições de outros articuladores.

Figura 3: - Estruturas participantes do mecanismo de produção da fala

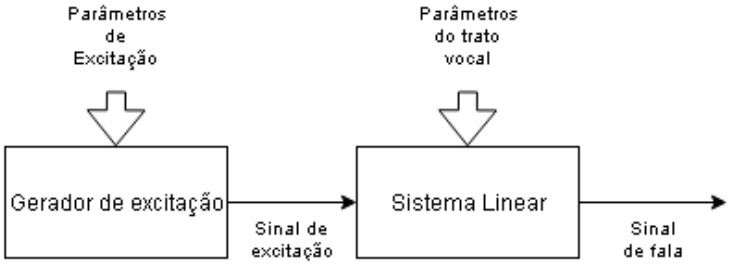


Fonte: Adaptado de [36]

A produção da fala envolve o movimento de um fluxo de ar, produzido pela compressão dos pulmões, que passa através da traqueia e da faringe e sai pela cavidade oral e/ou nasal. Quando esse fluxo é alterado devido a vibrações das cordas vocais, pulsos glotais são produzidos e excitam o trato vocal. Quando isso ocorre, o som produzido é chamado de vozeado. As cordas vocais atuam para prover uma excitação periódica. O tempo entre sucessivas aberturas das cordas vocais é chamado de período fundamental, enquanto a taxa de vibração das cordas é conhecida como frequência fundamental ou *pitch*. Já quando o ar passa direto pelas cordas vocais, sem a interferência da vibração delas, e é forçado através de alguma constrição em algum ponto do trato vocal, o som produzido é conhecido como surdo. Uma característica importante de sons surdos é que eles não possuem *pitch* e a excitação do trato vocal corresponde a ruídos de largo espectro. Fisicamente, os sons de fala podem ser descritos em termos do *pitch* e de frequências formantes. De fato, essa descrição constitui um método de análise utilizado pela maioria dos algoritmos de compressão de voz [37]. As frequências formantes, que podem ser chamadas apenas de formantes, são as frequências ressonantes do trato vocal. Os picos da resposta em frequência do trato vocal correspondem a essas frequências [38].

O sistema da Figura 3 pode ser descrito pela teoria acústica, e técnicas numéricas podem ser usadas para criar uma simulação física completa da geração e transmissão de sons no trato vocal, mas, na maioria das vezes, é suficiente modelar a produção de um sinal de fala amostrado por um modelo de tempo discreto tal como o mostrado na Figura 4. O sistema linear de tempo discreto variante no tempo, à direita da Figura 4, simula a modelagem de frequência do tubo do trato vocal. O gerador de excitação, à esquerda, simula os diferentes modos de geração de som no trato vocal. As amostras de um sinal de fala são assumidas como sendo a saída do sistema linear variante no tempo.

Figura 4: - Modelo de produção da fala fonte-filtro



Em geral, tal modelo é chamado de modelo de produção da fala fonte-filtro. A res-

posta de frequência de curta duração do sistema linear simula a modelagem de frequência do sistema do trato vocal e, como o trato vocal muda de forma de maneira relativamente lenta, é razoável supor que a resposta linear do sistema não varia ao longo dos intervalos de tempo na ordem de 10 ms ou mais. Assim, é comum caracterizar o sistema linear de tempo discreto por uma função da forma

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=0}^{N} a_k z^{-k}} = \frac{b_0 \prod_{k=1}^{M} (1 - d_k z^{-1})}{\prod_{k=1}^{N} (1 - c_k z^{-1})} \tag{1}$$

onde os coeficientes de filtro a_k e b_k , rotulados como parâmetros do trato vocal na Figura 4, mudam a uma taxa na ordem de 50 a 100 vezes por segundo. Alguns dos pólos (c_k) da função do sistema ficam próximos do círculo unitário e criam ressonâncias para modelar as frequências dos formantes. Na modelagem detalhada da produção de fala, às vezes é útil empregar zeros (d_k) da função do sistema para modelar sons nasais e fricativos [39].

A caixa rotulada como gerador de excitação na Figura 4 cria uma excitação apropriada para o tipo de som que está sendo produzido. Para sons vozeados, a excitação para o sistema linear é uma sequência quase-periódica de pulsos discretos (glotais). A frequência fundamental da excitação glótica determina o tom percebido da voz. Os pulsos glóticos individuais de duração finita têm um espectro de baixa passagem que depende de vários fatores [40]. Portanto, a sequência periódica de pulsos glóticos lisos possui um espectro de linhas harmônicas com componentes que diminuem em amplitude a medida que a frequência cresce. Frequentemente, é conveniente incluir a contribuição do espectro do pulso glotal no modelo do sistema de trato vocal expresso em (1). Isso pode ser alcançado por um pequeno aumento na ordem do denominador sobre o que seria necessário para representar as ressonâncias formantes. Para sons surdos, o sistema linear é excitado por um gerador de números aleatórios que produz um sinal de ruído em tempo discreto com espectro plano. Tanto para os sons vozeados quanto para os surdos, o sistema linear impõe sua resposta de frequência no espectro para criar os sons da fala.

Este modelo de som de fala, que assume a saída de um filtro digital de variação lenta no tempo submetido a uma excitação, que capta a natureza da distinção entre vozeado e surdo na produção de fala é a base para pensar sobre o sinal de fala, e uma grande variedade de representações digitais desse sinal são baseados neste modelo. Ou seja, o sinal de fala pode ser representado pelos parâmetros do modelo em vez da forma de onda amostrada no tempo. Assumindo que as propriedades do sinal de fala e do modelo

são constantes em curtos intervalos de tempo, é possível estimar os parâmetros do modelo, analisando blocos curtos de amostras do sinal. É através desses modelos e técnicas de análise que somos capazes de construir propriedades do processo de produção de fala em representações digitais do sinal de fala.

O primeiro passo em qualquer sistema de reconhecimento automático de voz é extrair características, isto é, identificar os componentes do sinal de áudio que são bons para identificar o conteúdo linguístico e descartar todas as outras coisas que transportam informação relacionada ao ruído de fundo. Como visto na seção anterior, os sons gerados por um ser humano são filtrados pela forma do trato vocal, incluindo a língua, os dentes, etc. Essa forma determina o som produzido. Se pudermos determinar a forma com precisão, isto deve nos dar uma representação precisa do fonema que está sendo produzido, bem como a emoção inerente. A forma do trato vocal se manifesta na envoltória do espectro avaliado em curto período de tempo, e o papel dos coeficientes mel-cepstrais é representar com precisão esta envoltória.

Os coeficientes mel-cepstrais (do Inglês, *Mel Frequecy Cepstral Coefficient* – MFCC), introduzidos por Davis e Mermelstein [41], são parâmetros do tipo espectrais vastamente utilizados no reconhecimento automático de voz e no reconhecimento automático de locutor. Por este motivo, existem diversas pesquisas na área de reconhecimento de emoções por meio da fala que também utilizam os MFCC como parâmetros dos sinais de voz [2].

A Figura 5 indica os procedimentos que devem ser realizados para a obtenção desses coeficientes. O primeiro processo refere-se à divisão do sinal em blocos com duração entre 20 e 40 milissegundos. Um sinal de áudio está constantemente se alterando e isso tende a tornar os cálculos mais complicados, com a divisão do sinal em pequenos blocos podemos assumir que dentro desses blocos os sinais são estatisticamente estacionários [41].

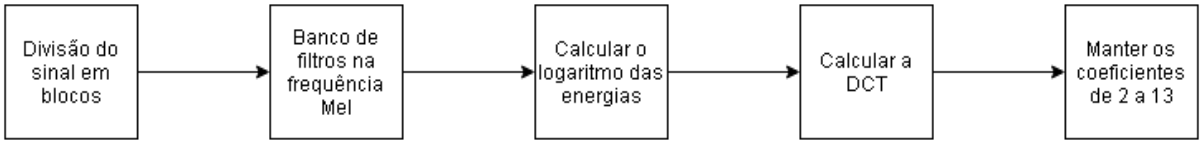
O próximo passo é calcular o espectro de potência de cada quadro. A cóclea é um órgão que compõe o ouvido interno e é responsável por transformar a vibração dos líquidos e de estruturas adjacentes do ouvido em sinais que se propagam pelo sistema nervoso. Dependendo da frequência do som que foi recebido pelo ouvido, a cóclea vibra em uma determinada parte, assim, diferentes terminações nervosas são ativadas informando ao cérebro as frequências de excitação. Essa segunda etapa do cálculo dos coeficientes mel-cepstrais efetua um trabalho similar ao da cóclea humana, identificando quais são as frequências que estão presentes em cada quadro do sinal de fala.

O terceiro procedimento se faz necessário pelo fato da cóclea não conseguir distinguir entre frequências muito próximas. Nessa etapa, bancos de filtros na frequência Mel são utilizados para o cálculo da quantidade de energia em diferentes faixas de frequência do espectro. Essa dificuldade de discernimento da cóclea se torna mais acentuado conforme as frequências aumentam. Devido a isso, não precisamos nos preocupar tanto com as variações de frequências maiores, e as larguras dos filtros do banco podem aumentar conforme as frequências crescem. A escala Mel, que relaciona a frequência percebida de um *pitch* puro com a sua frequência medida real, nos diz exatamente como espaçar os bancos de filtros e quão largos eles devem ser. Essa escala de frequência Mel é utilizada na obtenção dos coeficientes mel-cepstrais, pois ela aproxima esses parâmetros ao que os humanos ouvem. O próximo passo também é motivado pela audição humana. Nós não escutamos os sons em uma escala linear. Por isso, no quarto procedimento, é realizado o cálculo do logaritmo das energias dos bancos de filtros [41].

A penúltima etapa consiste em computar a transformada discreta do cosseno (do Inglês, *Discrete Cosine Transform* – DCT) dos logaritmos das energias dos bancos de filtros. Essas energias são correlacionadas por causa da sobreposição dos filtros. A DCT é utilizada para descorrelacionar esses valores para tornar possível a utilização de matrizes co-variância diagonais em classificadores como o HMM e o GMM.

Por último, em geral, são mantidos somente os coeficientes de número dois até o coeficiente de número treze. Nos coeficientes maiores, após o décimo terceiro, as energias variam muito rapidamente, e essa variação implica em uma degradação do desempenho dos classificadores [42].

Figura 5: - Etapas do cálculo dos coeficientes mel-cepstrais



Os coeficientes mel-cepstrais, como mencionado inicialmente, descrevem apenas a envoltória do espectro de potência do sinal contido em um quadro. Entretanto, as variações desses coeficientes no tempo também são informações importantes para a representação dos sinais de voz. Para extrair essas informações devem ser calculados os coeficientes deltas e os coeficientes delta-deltas, também conhecidos como coeficientes di-

ferenciais e de aceleração, respectivamente [43]. A Equação (2) é utilizada para o cálculo dos coeficientes deltas

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{2}$$

onde d_t é um coeficiente delta, calculado no quadro t em termos dos coeficientes estáticos c_{t+n} e c_{t-n}, e N é tipicamente igual a dois [42]. Já os coeficientes delta-deltas são obtidos replicando a derivada sobre os resultados obtidos na primeira derivação.

2 BASES DE DADOS DE EMOÇÕES EM SINAIS DE FALA

Neste capítulo, descrevem-se as bases de dados utilizadas neste trabalho. São fornecidos detalhes a respeito das metodologias de sua criação, cujo propósito principal é disponibilizar gravações de conteúdo fidedigno às emoções desejadas.

2.1 Berlin Database of Emotional Speech

A base de dados *Berlin Database of Emotional Speech* (EmoDb) foi desenvolvida por um grupo de estudos alemão e é composta por falas produzidas de maneiras atuadas [14]. Ela possui oitocentas sentenças que foram geradas por dez atores amadores, divididos igualmente entre ambos os sexos, as quais simularam sete emoções: raiva, medo, tédio, felicidade, tristeza, desgosto e um estado neutro. Essas emoções foram escolhidas como os rótulos das sentenças, pois o grupo de estudos já havia publicado outros trabalhos utilizando essas emoções, e eles tinham como objetivo comparar esses trabalhos [44] [45]

Os autores decidiram utilizar emoções atuadas, mesmo existindo muitos argumentos contra, devido à dificuldade em se conseguir montar uma base de dados com emoções básicas utilizando situações reais. São raras as situações no nosso dia a dia onde expressamos de forma clara as nossas emoções, bem como existe um problema ético em gravar pessoas expressando emoções plenas. Além disso, eles tiveram como premissa que todos os atores interpretassem todas as emoções e repetissem as mesmas frases para permitir que os pesquisadores realizassem comparações com o conteúdo da base. O material utilizado nas interpretações foi composto por dez frases, apresentadas na Tabela 2, utilizadas no dia a dia das pessoas, para que os atores pudessem se familiarizar mais rapidamente com os textos e os interpretassem da maneira mais natural possível.

Para garantir a qualidade e a naturalidade da base foi realizado um teste de percepção com vinte pessoas. As gravações foram apresentadas a essas pessoas de forma aleatória e elas foram perguntadas qual era a emoção que o ator transmitira ao pronunciar aquela frase e o quão convincente ele havia sido. As falas com uma taxa de reconhecimento maior que 80% e naturalidade maior que 60% foram escolhidas para outros testes. No total, quinhentas das oitocentas sentenças ficaram de fora. A Figura 6 apresenta o resultado da taxa de reconhecimento de cada emoção nesse teste. Por meio desse resultado, podemos perceber uma significativa diferença entre as emoções. Mais dois

Tabela 2: - Frases utilizadas pelos atores para compor a base de dados EmoDB

Frase original	**Tradução**
Der Lappen liegt auf dem Eisschrank.	A toalha de mesa está em cima da geladeira
Das will sie am Mittwoch abgeben.	Ela vai entregá-lo na quarta-feira.
Heute abend könnte ich es ihm sagen.	Esta noite eu poderia contar a ele.
Das schwarze Stück Papier befindet sich da oben neben dem Holzstück.	A folha de papel preta está localizada lá em cima, ao lado do pedaço de madeira.
In sieben Stunden wird es soweit sein.	Em sete horas será.
Was sind denn das für Tüten, die da unter dem Tisch stehen?	E as malas que estão lá embaixo da mesa?
Sie haben es gerade hochgetragen und jetzt gehen sie wieder runter.	Eles acabaram de levar para cima e agora estão descendo novamente.
An den Wochenenden bin ich jetzt immer nach Hause gefahren und habe Agnes besucht.	Nos fins de semana, eu sempre dirigia para casa e visitava Agnes.
Ich will das eben wegbringen und dann mit Karl was trinken gehen.	Eu só quero tirar isso e depois tomar uma bebida com Karl.
Die wird auf dem Platz sein, wo wir sie immer hinlegen.	Ela estará na praça onde sempre a deixamos.

testes foram realizados. Em um deles foi solicitado que fosse avaliada a força das emoções apresentadas em cada sentença, e em outro teste foram julgadas as sílabas tônicas de cada sentença.

Figura 6: - Resultado do teste de reconhecimento das emoções contidas nas sentenças da base EMODb

Essa base de dados foi escolhida para o nosso trabalho por dois motivos: já serviu de base para muitos estudos [46] [47] [48] e está disponível gratuitamente na Internet [1].

[1] http://emodb.bilderbar.info/docu/#download

2.2 Interactive Emotional Dyadic Motion Capture Database

A *Interactive emotional dyadic motion capture database* (IEMOCAP) é uma base de dados audiovisuais desenvolvida por pesquisadores da *University of Southern California* cujo objetivo principal foi criar uma base de dados com um grande corpus emocional, por meio do auxílio de muitos indivíduos capazes de expressar emoções genuínas [1]. Ela foi desenvolvida após os autores perceberem que uma das maiores limitações da área de estudo das expressões das emoções é a falta de bases de dados com interações genuínas, sem seguir um roteiro, e capturadas dentro de um contexto. Outras limitações observadas nas bases de dados existentes foram o pequeno número de pessoas que atuaram para gerar os dados, o tamanho reduzido dessas bases, e que a maioria delas era composta somente de sinais de áudio [1].

Para alcançar esse objetivo, o conteúdo da base deveria ser cuidadosamente selecionado. Portanto, os atores foram solicitados a trabalhar com duas abordagens [1]. Na primeira, foi proposto aos participantes a memorização e o ensaio de roteiros. O uso de roteiros fornece uma maneira de restringir o conteúdo semântico e emocional da base. Um profissional de teatro selecionou três roteiros de um total de mais de cem peças. Além disso, essas peças foram selecionados de modo que cada um deles consistia de um papel feminino e masculino. Este requisito foi imposto para equilibrar os dados em termos de gênero. Uma vez que essas emoções são expressas dentro de um contexto adequado, elas são mais propensas a serem transmitidas de uma maneira genuína, em comparação com gravações de frases isoladas. Na segunda abordagem, os sujeitos foram solicitados a improvisar com base em cenários hipotéticos que foram projetados para provocar emoções específicas. Os tópicos para os cenários espontâneos foram selecionados seguindo as orientações fornecidas por [49]. Como relatado em seu livro, os autores entrevistaram indivíduos que foram convidados a lembrar de situações no passado que provocaram certas emoções neles. Os cenários hipotéticos foram baseados em algumas situações comuns, por exemplo, perda de um amigo, separação, etc. Nesse cenário, os sujeitos estavam livres para usar suas próprias palavras para se expressarem. Ao conceder aos atores uma considerável quantidade de liberdade na expressão de suas emoções, foi esperado pelos autores que os resultados proporcionassem uma genuína percepção das emoções. No início do trabalho os autores definiram as emoções de felicidade, de raiva, de tristeza, de frustração e um estado neutro como alvo, mas no momento da produção

da base foram incluídas as emoções de desgosto, de medo, de excitação e de surpresa. Os rótulos emocionais recebidos pelas sentenças gravadas foram dadas por meio de avaliações subjetivas.

A base de dados contou com sete atores profissionais e três alunos do departamento de teatro da University of Southern California. Ao todo, cinco homens e cinco mulheres foram escolhidos após uma audição. Os indivíduos gravaram, em duplas formadas por um homem e uma mulher, em cinco sessões, onde cada uma durou aproximadamente seis horas, incluindo períodos de descanso. Após as sessões, os diálogos foram segmentados por turnos, onde cada turno foi definido como segmento contínuo por ator. No total, a base de dados contém 10039 turnos, onde 5255 foram de sessões em que os atores seguiram um roteiro, e 4784 foram extraídos das sessões espontâneas.

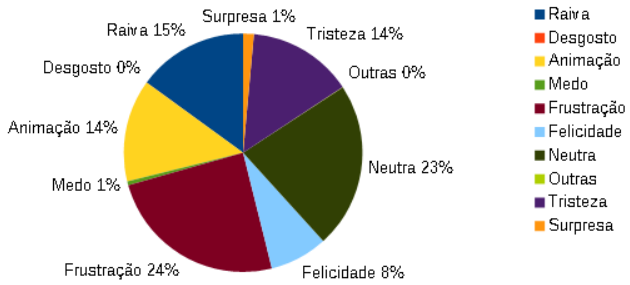
Na maioria das bases de dados, os atores são solicitados a enunciar uma frase expressando uma determinada emoção a qual é utilizada posteriormente como rótulo para essa enunciação, como foi desenvolvida a base de dados EmoDb. Uma desvantagem desse método é não garantir que a expressão oral gravada reflita a emoção alvo. Adicionalmente, uma determinada exibição pode extrair diferentes percepções emocionais. Para evitar esse problema, o IEMOCAP foi rotulado com uma combinação de avaliações subjetivas. Para tal, alguns alunos da USC avaliaram o conteúdo emocional dos turnos gravados utilizando diferentes metodologias. Foram utilizados dois esquemas: de anotações discretas baseadas em categorias e anotações baseadas em atributos contínuos. No esquema discreto, os avaliadores utilizaram categorias, tais como felicidade, tristeza, etc. Já para o esquema contínuo, as gravações foram avaliadas com os atributos de ativação - que é indicado por comportamentos de alerta e escalas bipolares como calmo/excitado e acordado/sonolento; a valência - que é caracterizada por escalas bipolares como prazer/desprazer, feliz/infeliz e positivo/negativo, e a dominância - que representa o grau de controle que o indivíduo tem sobre determinada situação, e indicada por escalas bipolares como controle/controlado e autônomo/guiado [8]. Essas duas abordagens fornecem informações complementares das emoções manifestadas no corpus.

A etapa de avaliação do conteúdo emocional em anotações discretas baseadas em categorias contou com seis avaliadores. Cada turno foi classificado por três pessoas diferentes. Por questões de simplicidade, foi adotado o método de voto majoritário para a atribuição do rótulo ao turno, se a categoria com o maior número de votos fosse única.

No total, 74,6% dos turnos seguiram esse critério, em 66,9% dos turnos onde os atores seguiram um roteiro e em 83,1% correspondem a improviso. Essa avaliação possui como pressuposto que o conteúdo emocional não varia dentro de um turno, dado que a sua duração média é pequena. Os avaliadores tiveram a liberdade para classificar um turno com mais de uma emoção, visto que uma mistura de emoções é normalmente observada nas interações humanas [12].

Em um primeiro momento, os autores da base de dados definiram que ela seria dividida em quatro classes, onde cada classe representaria as emoções presentes nas encenações dos turnos, que são: raiva, tristeza, felicidade, frustração e um estado emocional neutro. Entretanto, devido a naturalidade com a qual os atores desenvolveram as representações, os autores observaram que utilizar apenas essas classes tornaria a descrição das gravações empobrecida e menos assertiva. Em vista desse problema, os autores adicionaram às classes inicialmente definidas as emoções de desgosto, de medo, de surpresa e de animação.

Figura 7: - Distribuição dos dados do IEMOCAP por categoria nas sessões roterizadas



Na Figura 7 e na Figura 8 temos a distribuição dos dados em relação as classes, levando em consideração apenas os turnos onde a categoria com maior número de votos foi única. Elas representam, respectivamente, as sessões onde as gravações seguiram um roteiro e as sessões onde os atores puderam improvisar. As emoções de medo e de desgosto representaram menos de 1% cada uma nos dois formatos de sessão.

A Tabela 3 representa uma matriz de confusão da base de dados, onde os rótulos atribuídos aos turnos após a votação majoritária são considerados as classes reais, enquanto as classificações de todos os avaliadores representam as classes previstas. A taxa

Figura 8: - Distribuição dos dados do IEMOCAP por categoria nas seções espontâneas

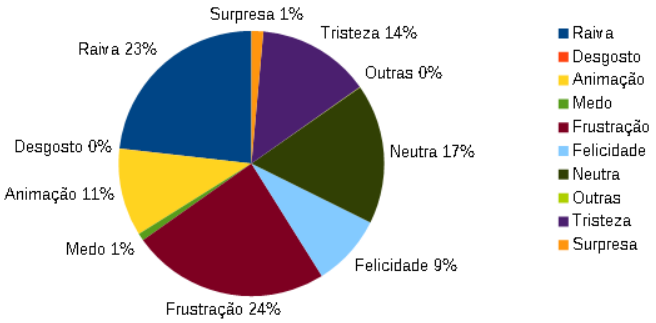


Tabela 3: - Matriz Confusão entre as categorias emocionais baseadas em avaliação humana [1]

Rótulos	Neu	Fel	Tri	Rai	Sur	Med	Des	Fru	Ani	Out
Neutra	0.74	0.02	0.03	0.01	0.00	0.00	0.00	0.13	0.05	0.01
Felicidade	0.09	0.70	0.01	0.00	0.00	0.00	0.00	0.01	0.18	0.01
Tristeza	0.08	0.01	0.77	0.02	0.00	0.01	0.00	0.08	0.01	0.02
Raiva	0.01	0.00	0.01	0.76	0.01	0.00	0.01	0.17	0.00	0.03
Surpresa	0.01	0.04	0.01	0.03	0.65	0.03	0.01	0.12	0.09	0.01
Medo	0.03	0.00	0.05	0.02	0.02	0.67	0.02	0.05	0.15	0.00
Desgosto	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.17	0.17	0.00
Frustração	0.07	0.00	0.04	0.11	0.01	0.01	0.01	0.74	0.01	0.02
Animação	0.04	0.16	0.00	0.00	0.02	0.00	0.00	0.02	0.75	0.00

Neu - Neutra, Fel - Felicidade, Tri - tristeza, Rai - Raiva, Sur - Surpresa,
Med - Medo, Des - Desgosto, Fru - Frustração, Ani - Animação, Out - Outras

de classificação, em média, foi de 72%. A matriz nos mostra que as emoções neutra, de raiva e de desgosto foram confundidas por alguns avaliadores com a emoção de frustração. Podemos observar também que alguns avaliadores tiveram dificuldade em diferenciar entre felicidade e animação.

Vale ressaltar que os autores utilizaram também um procedimento alternativo para descrever o conteúdo emocional dos enunciados. Esse procedimento envolveu a avaliação, por duas pessoas, das dimensões emocionais de valência, ativação e dominância. Os avaliadores foram orientados a classificar as três dimensões com valores inteiros de um a cinco, levando em consideração o conteúdo emocional da base de dados. Este procedimento permite uma descrição mais generalizada do conteúdo efetivo dos atores em um espaço contínuo e também a análise da variação da expressão emocional de um discurso [1].

Além das avaliações descritas anteriormente, os autores solicitaram para seis atores avaliarem o conteúdo emocional das seções que participaram. Eles classificaram as

sentenças da mesma forma que os outros avaliadores, em categorias e atributos. Foi realizada uma comparação, apresentada na Tabela 4, entre o resultado dessas avaliações e as que foram realizadas pelos observadores externos. Para essa comparação, foram tomadas como classes reais os rótulos definidos pela votação majoritária. Na tabela, os cabeçalhos das colunas apresentam o gênero do ator e a sessão avaliada, por exemplo, H03 representa a parte encenada pelo homem na sessão 3, já o M01 representa a parte atuada por uma mulher na sessão 1. Analisando o resultado, podemos perceber que ocorreu uma significativa diferença na percepção dos dois grupos das emoções presentes nas sentenças avaliadas. Fato que mostra o desafio, até mesmo para os humanos, de se classificar as emoções presentes na fala.

Tabela 4: - Comparação da taxa de reconhecimento em porcentagem entre a avaliação por si e pelos outros para os cenários espontâneos (avaliação categórica)

Avaliação	M01	M02	M03	H01	H03	H05	Média
Autoavaliação	0,79	0,58	0,44	0,74	0,57	0,54	0,60
Externa	0,76	0,80	0,79	0,81	0,80	0,77	0,79

3 MÉTODOS DE CLASSIFICAÇÃO

Neste capítulo, são apresentados os métodos de classificação empregados neste trabalho no reconhecimento de emoções em sinais de fala.

3.1 Reconhecimento de emoções através do Modelo de misturas de Gaussianas

Nesta seção, apresenta-se, inicialmente, uma visão geral do sistema de reconhecimento de emoções usando o modelo de mistura de Gaussianas (GMM). Em seguida, apresenta-se a fundamentação teórica por trás do projeto e aplicação de uma classificador baseado em GMM.

3.1.1 Visão Geral do Sistema

No primeiro sistema desenvolvido em nosso trabalho, apresentado no diagrama em blocos da Figura 9, escolhemos utilizar um classificador Bayesiano baseado em Modelo de Misturas Gaussianas (do Inglês, *Gaussian Mixture Model* - GMM), visto que estudos anteriores relataram um bom desempenho e baixo custo computacional [2]. Dessa maneira, o GMM foi escolhido como o método de referência a ser comparado aos demais métodos desenvolvidos neste trabalho.

Figura 9: - Diagrama do sistema com GMM

Inicialmente, os áudios da bases de dados (EmoDb ou IEMOCAP) foram pré-processados utilizando um detector de atividade vocal (do Inglês, *Voice Activity Detector* – VAD) com o intuito de extrair os períodos de silêncio dos áudios. Em seguida, setenta por cento dos dados foram separados para a fase de treinamento e os outros trinta para a fase de teste. Após esta etapa, foi realizada a extração de parâmetros. Nesse caso, decidiu-se trabalhar com os coeficientes mel-cepstrais (MFCC) das amostras de áudio das bases, pois tais proporcionam uma boa caracterização da emoção em sinais de fala, permi-

tindo um bom desempenho em sistemas de reconhecimento encontrados na literatura [2]. Os coeficientes MFCC de uma parcela da base de áudios são usados para o projeto do classificador,que consiste na obtenção de modelos de misturas de Gaussianas definidos para cada emoção. Os coeficientes da outra parcela são usados para avaliar o desempenho do classificador que se baseia nos modelos obtidos na etapa de treinamento. Na etapa do treinamento, cada classe de emoção foi aprendida individualmente através de um algoritmo que é uma variação do algoritmo *Expectation Maximization* (EM) conhecida como algoritmo de agrupamento Figueiredo-Jain [50]. Nessa variação, o próprio algoritmo estima a quantidade de Gaussianas a ser utilizada em cada classes. Vale ressaltar que os sinais de voz são divididos em quadros, e a classificação da emoção é realizada por quadro a partir dos coeficientes MFCC calculados. Dessa maneira, realiza-se votação majoritária que trabalha da seguinte maneira: após cada quadro ser classificado com um determinado rótulo de emoção, verifica-se qual foi a classe definida para a maior parte dos quadros e então define-se que a amostra será classificada com esse rótulo.

3.1.2 Classificação com base em GMM

As distribuições Gaussianas simples possuem limitações significativas quando lidamos com base de dados reais. Para essas situações, uma das formas de se obter uma melhor caracterização utilizando Gaussianas é dada por meio de uma combinação linear de duas ou mais destas funções para modelar um conjunto de dados [51]. Tais superposições podem ser formuladas como modelos probabilísticos conhecidos como mistura de distribuições. Portanto, um modelo de misturas de Gaussianas é um modelo paramétrico representado como uma soma ponderada de densidades Gaussianas que possui a seguinte forma:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k g(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3}$$

onde π_k são os coeficientes da mistura e $g(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ é a k-ésima densidade Gaussiana dada por:

$$g(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right). \tag{4}$$

Cada densidade possui seu vetor de média $\boldsymbol{\mu}_k$, de dimensão D, e uma matriz

de covariância $\mathbf{\Sigma_k}$, de dimensão $D \times D$. Com todos esses parâmetros adequadamente ajustados, quase toda densidade contínua pode ser aproximada com alguma precisão arbitrária. Além disso, os coeficientes da mistura satisfazem dois requisitos básicos para serem considerados probabilidades [51]: $\sum_{k=1}^{K} \pi_k = 1$ e $0 \leq \pi_k \leq 1$.

Seja $\mathbf{z}$ uma variável aleatória binária com dimensão K, em que um elemento particular z_k é igual a um e os demais elementos do vetor são iguais a zero. Os valores de z_k, portanto, satisfazem as condições $z_k \in \{0, 1\}$ e $\sum_k z_k = 1$, e o vetor $\mathbf{z}$ possui K possíveis estados. A distribuição conjunta $p(\mathbf{x}, \mathbf{z})$ pode ser escrita como

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}). \tag{5}$$

A distribuição marginal sobre $\mathbf{z}$ é especificada em termos dos coeficientes de mistura, tal que

$$p(z_k = 1) = \pi_k, \tag{6}$$

lembrando que os parâmetros π_k satisfazem as condições para serem probabilidades. Pode-se escrever a distribuição de $\mathbf{z}$ na forma

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{7}$$

De forma similar, a distribuição condicional de $\mathbf{x}$ dado um valor particular de $\mathbf{z}$ é uma Gaussiana:

$$p(\mathbf{x}|z_k = 1) = g(\mathbf{x}|\boldsymbol{\mu_k}, \mathbf{\Sigma_k}), \tag{8}$$

a qual pode também ser escrita na forma:

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} g(\mathbf{x}|\boldsymbol{\mu_k}, \mathbf{\Sigma_k})^{z_k}. \tag{9}$$

A distribuição marginal de $\mathbf{x}$ é então obtida pela soma da distribuição conjunta sobre todos os estados possíveis de $\mathbf{z}$

$$p(\mathbf{x}) = \sum_{z} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k g(\mathbf{x}|\boldsymbol{\mu_k}, \mathbf{\Sigma_k}) \tag{10}$$

onde utilizamos (7) e (9). Portanto, a distribuição marginal de $\mathbf{x}$ é uma mistura de Gaussianas da forma (3). Sejam N observações $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, dado que $p(\mathbf{x}) = \sum_z p(\mathbf{x}, \mathbf{z})$, para cada observação $\mathbf{x}_n$ existe uma variável latente $\mathbf{z}_n$. Visto que $\mathbf{z}_n = [z_{n1}\, z_{n2} \cdots z_{nk} \cdots z_{nK}]^T$, podemos escrever a probabilidade condicional de $\mathbf{z}$ dado $\mathbf{x}$, $\gamma(z_{nk})$, da seguinte forma

$$\begin{aligned} \gamma(z_{nk}) \equiv p(z_{nk} = 1|x) &= \frac{p(z_{nk} = 1)p(\mathbf{x}|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k g(\mathbf{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})}{\sum_{j=1}^{K} \pi_j g(\mathbf{x}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})} \end{aligned} \tag{11}$$

Observe que π_k corresponde à probabilidade a priori de $z_{nk} = 1$, e a função $\gamma(z_{nk})$ como a probabilidade a posteriori, uma vez que $\mathbf{x}$ seja observado.

3.1.2.1 Máxima Verossimilhança

Para um dado conjunto de observações $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, deseja-se modelar esses dados através de uma mistura de Gaussianas. Pode-se representar esse conjunto de dados através uma matriz $\mathbf{X}$, de dimensão $N \times D$, onde cada linha é composta pelo vetor $\mathbf{x_n^T}$. A forma da distribuição de mistura de Gaussianas é definida pelo vetor $\boldsymbol{\pi} = [\pi_1\, \pi_2 \cdots \pi_K]^T$; pela matriz $\boldsymbol{\mu}$, onde cada linha corresponde ao vetor $\boldsymbol{\mu}_k^T$; e pelo tensor $\boldsymbol{\Sigma}$, onde cada elemento corresponde a matriz $\boldsymbol{\Sigma}_k$. Por meio da equação (3), o logaritmo da função de verossimilhança é definido por

$$\ln\ p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k g(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{12}$$

Note que a complexidade da mistura de Gaussianas é significativamente maior se comparada a uma simples Gaussiana, devido ao somatório em função de K dentro do logaritmo. Como resultado dessa complexidade, a solução de máxima verossimilhança para encontrar os parâmetros ótimos não possui uma forma fechada, havendo a necessidade do uso de métodos númericos de otimização [52]. O método de Maximização da Esperança (*Expectation maximization* – EM) é uma forma de se obterem esses parâmetros.

3.1.2.2 Maximização da Esperança

O algoritmo de Maximização da Esperança (*Expectation maximization* – EM) é capaz de encontrar soluções de máxima verossimilhança em modelos com variáveis latentes [53]. Para compreender este método é necessário verificar, inicialmente, as condições que devem ser satisfeitas quando a função de verossimilhança atinge seu máximo. Primeiro calcula-se as derivadas de (12) em relação às médias de μ_k, igualando-as a zero. Assim, obtém-se

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{13}$$

onde

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{14}$$

Pode-se interpretar N_k como sendo o número esperado de pontos associados ao k-ésimo *cluster*. Na equação (13), a média μ_k do k-ésima componente Gaussiana é obtida por meio da média ponderada de todos os pontos no conjunto de dados, no qual o fator de ponderação para a amostra x_n é dado pela probabilidade a posteriori $\gamma(z_{nk})$.

Em segundo, calcula-se a derivada da (12) em relação a $\boldsymbol{\Sigma}_k$, igualando-a a zero. Dessa forma, a solução de máxima verossimilhança para a matriz de covariância da k-ésima Gaussiana é descrita por:

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \tag{15}$$

que possui forma similar do resultado correspondente para uma Gaussiana simples ajustado ao conjunto de dados. Observe que, novamente, cada dado de observação é ponderado pela probabilidade a posteriori correspondente,$p(z_{nk} = 1|x)$, e com o denominador dado pelo número efetivo de pontos associados ao componente correspondente.

Por fim, maximiza-se (12) em função dos coeficientes da mistura π_k. Neste caso, adiciona-se à função de verossimilhança a restrição $\sum_{k=1}^{K} \pi_k = 1$. Através da técnica de multiplicadores de Lagrange, obtém-se a seguinte função objetivo

$$\ln\ p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \tag{16}$$

Derivando-se essa expressão em relação a π_k e igualando-a a zero, resulta que

$\lambda = -N$, e que:

$$\pi_k = \frac{N_k}{N}. \tag{17}$$

Note que π_k corresponde ao valor esperado da probabilidade a posteriori associada a k-ésima componente da Gaussiana.

Vale ressaltar que os resultados (13), (15) e (17) não constituem uma solução fechada para os parâmetros do modelo da mistura posto que as funções a posteriori $\gamma(z_{nk})$ dependem destes parâmetros por meio de (11). No entanto, esses resultados sugerem um esquema iterativo simples para se encontrar uma solução para o problema da máxima verossimilhança, que, como veremos, é uma instância do algoritmo EM para o caso particular do modelo de mistura de Gaussianas. O algoritmo de EM transcorre da seguinte maneira [51]:

1. Escolhem-se valores iniciais para as médias $\boldsymbol{\mu}_k$, covariâncias $\boldsymbol{\Sigma}_k$ e coeficientes de mistura π_k e avalia-se o logaritmo da função de verossimilhança;

2. Etapa E: avaliam-se as probabilidades a posteriori através da equação (11);

3. Etapa M: reestimam-se as médias, covariâncias e coeficientes de mistura usando os resultados de (13), (15) e (17);

4. Avalia-se o logaritmo da função de verossimilhança e verifica-se a convergência a partir dos parâmetros ou da própria função de verossimilhança. Caso o critério de convergência não seja satisfeito, retorna-se ao passo 2.

Note que a cada atualização dos parâmetros, resultantes de uma etapa E seguida de uma etapa M, garante-se o aumento do logaritmo da função de verossimilhança. Na prática, atinge-se a convergência quando a variação na função de verossimilhança, ou alternativamente nos parâmetros, é inferior de algum limiar.

3.1.3 Algoritmo de Agrupamento Figueiredo-Jain

Uma das questões mais desafiadoras na estimativa de uma mistura de densidades de probabilidade se trata da definição do número apropriado de componentes [54]. Enquanto uma mistura com muitos componentes pode resultar em *overfitting*, uma com

poucos componentes pode não se aproximar da densidade real. O algoritmo Figueiredo-Jain (FJ) foi proposto para superar três questões principais do algoritmo EM básico. O algoritmo EM apresentado na seção anterior requer que o usuário defina o número de componentes. O algoritmo FJ ajusta o número de componentes durante a etapa de estimação pela exclusão de componentes que não são suportados pelos dados. Isso leva ao outro ponto de falha do algoritmo EM, a fronteira do espaço de parâmetros. O FJ evita essa fronteira quando exclui os componentes que estão se tornando singulares. O FJ também permite iniciar com um número arbitrariamente grande de componentes, essa possibilidade aborda o problema de inicialização do algoritmo EM. As atribuições iniciais para as médias dos componentes podem ser distribuídos em todo o espaço ocupado pelas amostras de treinamento, até mesmo definindo um componente para cada amostra de treinamento. A maneira clássica de selecionar o número de componentes de mistura é adotar a hierarquia "modelo-classe/modelo", onde alguns modelos candidatos (Funções densidade de probabilidade de mistura) são calculados para cada classe de modelo (número de componentes) e, em seguida, é selecionado o melhor modelo. A ideia por trás do algoritmo FJ é abandonar essa hierarquia e encontrar o melhor modelo diretamente. Utilizando o critério de comprimento mínimo da mensagem [55] [56] aplicado a modelos de mistura nos leva à função objetivo:

$$\Lambda(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, X) = \frac{V}{2} \sum_{c:\alpha_c>0} ln\left(\frac{N\alpha_c}{12}\right) + \frac{C_{nz}}{2} ln\frac{N}{12} + \frac{C_{nz}(V+1)}{2} - ln\ p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (18)$$

onde N é o número de pontos de treinamento, V é o número de parâmetros livres que especificam um componente, e C_{nz} é o número de componentes com pesos diferentes de zerona mistura ($\alpha_c > 0$). O último termo $ln\ p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ é o logaritmo da função verossimilhança dos dados de treinamento dados os parâmetros da distribuição.

O algoritmo EM pode ser utilizado para minimizar a 18 com um C_{nz} fixo. Isto leva à etapa M com a fórmula de atualização do peso da componente:

$$\alpha_c^{i+1} = \frac{max\{0, (\sum_{n=1}^{N} \omega_{n,c}) - V/2\}}{\sum_{j=1}^{C} max\{0, (\sum_{n=1}^{N} \omega_{n,c}) - V/2\}} \quad (19)$$

Esta fórmula contém uma regra explícita de aniquilar componentes definindo seus pesos para zero. Os passos de maximização acima não são adequados para o algoritmo EM

básico. Quando o C inicial é alto, pode acontecer que todos os pesos se tornem zero porque nenhum dos componentes tem suporte suficiente dos dados. Portanto, um algoritmo *Component-wise EM* (CEM) é adotado [50]. O CEM atualiza os componentes um por um, computando o passo E (atualizando W) após cada atualização do componente, onde o EM básico atualiza todos os componentes simultaneamente. Quando um componente é aniquilado, sua massa de probabilidade é imediatamente redistribuída fortalecendo os componentes restantes.

Quando o CEM converge, não é garantido que o mínimo de X seja encontrado, porque a regra de aniquilação na equação 19 não leva em consideração a diminuição causada pela diminuição do C_{nz}. Após a convergência, o componente com o menor peso é removido e o CEM é executado novamente, repetindo até $C_{nz} = 1$. Então a estimativa com o menor $\Lambda(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, X)$ é escolhida. A implementação do algoritmo FJ usa uma função de custo modificada em vez de $\Lambda(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, X)$.

$$\Lambda'(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, X) = \frac{V}{2} \sum_{c:\alpha_c>0} ln\alpha_c + \frac{C_{nz}(V+1)}{2} + lnN - ln\ p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad (20)$$

3.2 Reconhecimento de emoções com Redes Neurais Probabilísticas

Nesta seção, apresenta-se inicialmente uma visão geral do sistema de reconhecimento de emoções usando redes neurais probabilísticas (PNN – *Probabilistic Neural Networks*). Em seguida, apresenta-se a fundamentação teórica relacionada às redes neurais probabilísticas.

3.2.1 Visão Geral do Sistema

O sistema desenvolvido neste trabalho para reconhecimento de emoções baseado em PNN está ilustrado na Figura 10. Observe que o arcabouço é o mesmo daquele apresentado no Capítulo 3.1, a menos da etapa de classificação. Portanto, da mesma forma, removem-se os trechos de silêncio dos sinais de fala da base de dados e extraem-se os coeficientes MFCC. Contudo, neste caso, usa-se o PNN como classificador no lugar do GMM. Tal escolha foi motivada pelos resultados obtidos em [57]. Neste artigo, são comparadas diferentes redes neurais no reconhecimento de emoções em sinais de fala,

onde a rede PNN apresentou um desempenho superior às demais redes. Além disso, a PNN é considerada uma rede neural rápida e robusta quando o número de parâmetros é reduzido [25].

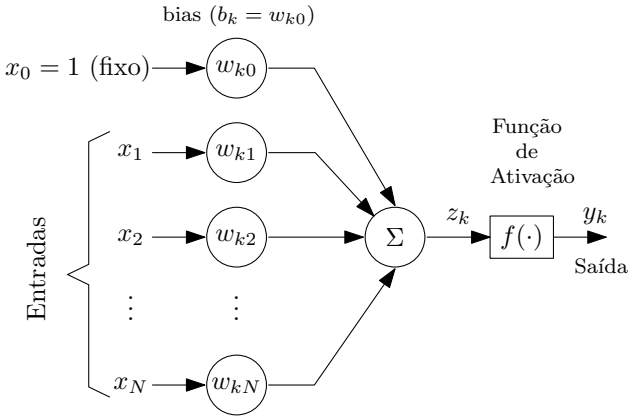
Figura 10: - Diagrama do sistema de reconhecimento de emoções com PNN

3.2.2 Redes Neurais Probabilísticas

As redes neurais têm suas origens nas tentativas de se encontrar representações matemáticas do processamento de informações realizado em sistemas biológicos [58]. O trabalho nessa área foi motivado desde o início pelo reconhecimento de que o cérebro humano computa de uma forma totalmente diferente do computador digital convencional. O cérebro é um computador altamente complexo, não linear e paralelo. Ele tem a capacidade de organizar as células que o constituem, conhecidas como neurônios, de modo a realizar certos cálculos - por exemplo, o reconhecimento de padrões, a percepção e o controle motor - muitas vezes mais rápido do que o computador digital mais rápido atualmente existente. Ele realiza rotineiramente tarefas de reconhecimento perceptivo em aproximadamente 100-200 ms [59], enquanto tarefas de complexidade muito menor demoram muito mais em um computador poderoso. Visto isso, foi desenvolvida a ideia de Rede Neural Artificial (RNA) que é um modelo computacional inspirado na maneira como as redes neurais biológicas no cérebro humano processam informações. Graças a muitos resultados inovadores em reconhecimento de voz, visão computacional, e processamento de texto, as RNAs têm sido vastamente utilizadas em pesquisa científica e na indústria [5, 12, 23, 25, 59, 60].

A unidade básica de computação em uma rede neural é o neurônio, ilustrado na Figura 11, frequentemente chamado de nó ou unidade. Ele recebe informações de entrada de alguns outros nós ou de uma fonte externa e calcula uma saída. Cada entrada tem um peso associado, que é atribuído com base em sua importância relativa para outras entradas. O nó aplica uma função, denominada de função de ativação, à soma ponderada de suas entradas.

Figura 11: - Modelo não-linear de um neurônio de uma rede neural artificial



Matematicamente, o k-ésimo neurônio de uma rede neural é representado pela seguinte equação:

$$y_k = f\left(\sum_{j=0}^{N} w_{kj}x_j\right), \tag{21}$$

onde N é a dimensão do vetor de entrada, $f(\dot{})$ é a função de ativação, w_{kj} são os pesos e x_j corresponde às entradas, sendo que x_0 é fixo e igual a 1. Alternativamente, tem-se:

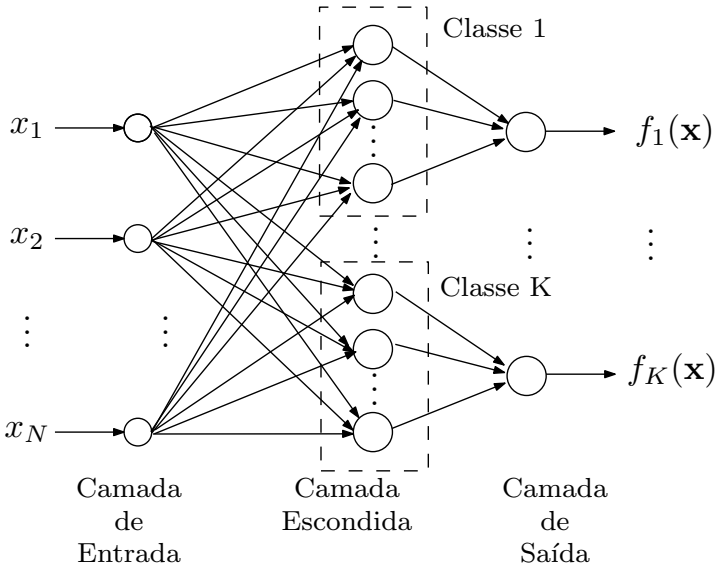
$$y_k = f\left(\sum_{j=1}^{N} w_{kj}x_j + b_k\right), \tag{22}$$

onde $b_k = w_{k0}$ é denominado *bias*. Vale ressaltar que o papel da função de ativação é introduzir a não-linearidade na saída de um neurônio. Isso é importante porque a maioria dos problemas do mundo real não é linear. Em geral, as redes neurais empregam diferentes arquiteturas envolvendo múltiplas camadas e apresentam um grande nível de interconexão entre os neurônios [59]. A mais conhecida é a rede neural *feedforward*, formada por uma camada de entrada e outra de saída, bem como por camadas intermediárias ou escondidas. Neste caso, todos os neurônios de uma determinada camada estão interligados a todos os neurônios da camada subsequente. A rede neural requer treinamento para a obtenção de seus pesos e isso é realizado através do procedimento de *backpropagation* associado a um algoritmo de otimização como o gradiente descendente [51, 52, 59].

As redes neurais probabilísticas são baseadas em princípios estatísticos bem estabelecidos, cuja motivação para o desenvolvimento recai sobre a necessidade de se encontrar

uma alternativa para o *backpropagation*, visto que este apresenta uma alta complexidade computacional e é susceptível a mínimos locais. Essas redes são baseadas na estratégia de decisão de Bayes e nas janelas de Parzen, que é um estimador não-paramétrico de funções de densidade de probabilidade [61, 62]. O funcionamento da PNN basicamente consiste em aprender a aproximar a função de densidade de probabilidade das amostras de treino. Sua arquitetura é formada por três camadas, como pode ser visto na Figura 12, que são as camadas de entrada, escondida e de saída. Na literatura [61, 62], encontra-se a denominação de camada de padrão para a camada escondida, e a camada de saída é subdividida em camadas de soma e de decisão.

Figura 12: - Arquitetura de uma rede neural probabilística



A arquitetura da Figura 12 se refere a uma PNN para reconhecimento de K classes distintas. A camada de entrada possui N nós, um para cada elemento do vetor de entrada, sendo que todos os nós de entrada estão conectados a todos os nós da camada escondida. Os nós da camada escondida são agrupados em K classes, sendo que cada classe apresenta sua cardinalidade, ou seja, possui um número distinto de nós. Cada nó corresponde a uma Gaussiana centrada em um vetor do conjunto de treinamento pertencente à k-ésima classe. Os nós de cada classe da camada escondida se conectam a um mesmo nó da camada de saída, resultando em uma soma de Gaussianas que forma uma função densidade de probabilidade. Portanto, na camada de saída existe um nó por classe.

Sejam os vetores do conjunto de treinamento da k-ésima classe $\{\mathbf{x}_{kc} : c = 1, \ldots, C_k\}$, sendo C_k sua cardinalidade. Cada vetor desse conjunto define uma Gaussiana para uma

dado vetor de entrada $\mathbf{x}$:

$$g_{kc}(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{N}}} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_{kc}\|^2}{2\sigma^2}\right\}, \tag{23}$$

onde σ é o fator de espalhamento. No k-ésimo nó de saída, somam-se os valores recebidos dos nós da camada escondida da classe k, resultando nas janelas de *Parzen* ou mistura de Gaussianas. Estas são definidas como

$$\begin{aligned} f_k(\mathbf{x}) &= \frac{1}{C_k}\sum_{c=1}^{C_k} g_{kc}(\mathbf{x}) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{1}{N}}}\frac{1}{C_k}\sum_{c=1}^{C_k}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_{kc}\|^2}{2\sigma^2}\right\} \end{aligned} \tag{24}$$

A escolha da classe $\hat{k}$ a qual pertence o vetor de entrada $\mathbf{x}$ é baseada no critério de máxima probabilidade a posteriori, ou seja:

$$\hat{k} = \arg\max_{k=1,\ldots,K} f_k(\mathbf{x}). \tag{25}$$

Vale ressaltar que o único parâmetro que precisa ser definido nessa rede é o fator de espalhamento das funções gaussianas, o que facilita na busca pela melhor rede para o problema proposto.

3.3 Reconhecimento de emoções com Redes Profundas Convolucionais

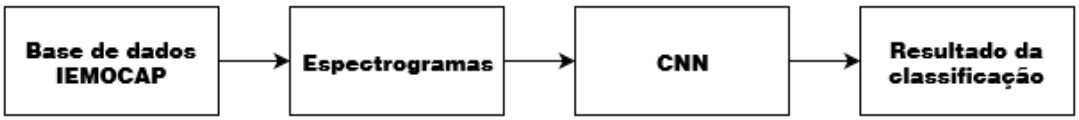
Nesta seção, apresenta-se, inicialmente, uma visão geral do sistema de reconhecimento de emoções usando redes profundas convolucionais. Em seguida, apresenta-se a fundamentação teórica relacionada às redes neurais convolucionais. Por fim, descreve-se o procedimento de transferência de aprendizado que foi adotado para permitir a adaptação de uma rede pré-treinada para o problema de reconhecimento de emoções.

3.3.1 Visão Geral do Sistema

Com base nos avanços científicos mais recentes e relevantes da área [5], desenvolvemos um sistema, ilustrado na Figura 13 que utiliza uma rede neural convolucional (do Inglês, *Convolutional Neural Network* - CNN) para realizar a tarefa de reconhecimento de emoções na fala de ponta a ponta, o que inclui a automatização da extração e esco-

lha de parâmetros [5]. Utilizamos neste trabalho a rede convolucional ResNet (*Residual Network*) com 34 camadas.

Figura 13: - Diagrama do sistema com CNN



As redes convolucionais são amplamente utilizadas na área de visão computacional, sendo atualmente o estado da arte em tarefas diversas como a classificação de imagens, a segmentação semântica, a transferência de estilo, a detecção de objetos, o reconhecimento facial, entre outras. Nossa ideia é explorar o sucesso dessa arquitetura para a aplicação de reconhecimento de emoções em fala, fazendo as devidas adaptações. Em particular, é necessário que se represente um sinal de fala, unidimensional, como uma matriz bi-dimensional, a qual poderia ser interpretada como uma imagem. O procedimento adotado, popular em outras aplicações relacionadas como reconhecimento de fala, é a partir do sinal de fala produzir um espectrograma, na escala perceptiva *mel*, quando então a matriz resultante é designada como *mel-espectrograma*. Os aspectos teóricos e práticos a respeito de espectrogramas podem ser vistos no Apêndice A. Além disso, dado que as CNN são redes neurais profundas que necessitam de uma grande quantidade de dados para a etapa de treinamento, decidimos realizar testes também com o aumento da base. Uma técnica muito utilizada para aumentar uma base de dados na área de visão computacional consiste em rotacionar as imagens originais e incluir essas novas imagens na base. Visto que espelhar um espectrograma faz com que a referência dos eixos da frequência e do tempo sejam invertidos, levando o conteúdo que temos em altas-frequências para as baixas-frequências e vice-versa, decidimos alterar algumas características das amostras de áudio originais, e adicionar esse resultado à base, em vez de realizarmos processamento diretamente no espectrograma. A primeira modificação que realizamos foi com o auxílio de um algoritmo de deslocamento de *pitch*, onde realizamos o deslocamento de 4 semitons do *pitch* de cada amostra de áudio. Essa alteração tornou os sons mais graves. Na segunda modificação, com o auxílio de um algoritmo de alargamento do tempo, modificamos a escala do tempo da amostra de áudio com um fator de alargamento igual a 1,5. Os aspectos teóricos e práticos a respeito da modificação de *pitch* e da escala de tempo

podem ser vistos no Apêndice B. Por último, criamos novas amostras de áudio, extraindo o silêncio dos arquivos originais, utilizando novamente o detector de atividade vocal com um limiar de 23 dB. A Tabela 5 exibe a quantidade de amostras separadas por classe para esta nova base aumentada.

Tabela 5: - Quantidade de amostras por classe da base de dados IEMOCAP aumentada

Classe	Quantidade de amostras
Felicidade	2380
Tristeza	4336
Raiva	4412
Neutra	6832

Um dos parâmetros mais importantes do algoritmo de treinamento da CNN que tivemos que definir foi a taxa de aprendizado. Essa taxa define o tamanho do passo de atualização dos pesos da rede na direção do gradiente, o qual é diretamente relacionado ao tempo necessário para a rede convergir ao mínimo da função custo. Uma pequena taxa de aprendizado faz o modelo convergir lentamente, enquanto uma taxa alta demais pode fazer o modelo divergir. Há portanto valores intermediários que permitem a convergência mais rápida e robusta.

Neste trabalho foram utilizadas taxas de aprendizagem diferenciais, um procedimento onde as camadas mais altas da rede mudam mais do que camadas mais profundas durante o treinamento. A criação de modelos de aprendizagem profunda em cima de arquiteturas pré-existentes é um método comprovado para gerar resultados muito melhores em tarefas de visão computacional. A maioria dessas arquiteturas são treinadas no *ImageNet* e, dependendo da similaridade de seus dados com as imagens no *ImageNet*, esses pesos precisarão ser alterados mais ou menos. Quando se trata de modificar esses pesos, as últimas camadas do modelo geralmente precisam de mais mudanças, enquanto os níveis mais profundos que já estão bem treinados para detectar recursos básicos, como bordas e contornos, precisarão de menos.

O procedimento que adotamos para definir a taxa de aprendizado consiste em iniciar o treinamento da rede com uma taxa muito baixa, digamos 1^{-8}, e, em seguida, aplicar o método do gradiente descendente uma única iteração e anotar o valor da função custo resultante. Após isso, repete-se o procedimento um certo número de vezes sempre usando uma taxa de aprendizado duas vezes maior que a anterior. Então plota-se o valor

da função custo em função da taxa de aprendizado. A curva resultante é usada para guiar a escolha da taxa de aprendizado a ser usada para o treinamento completo. Diversos critérios podem ser adotados, mas a recomendação é escolher aquela taxa a partir do qual a função custo começa a decair com maior intensidade. Isso não é o mesmo que escolher o ponto mínimo da curva, e o motivo é que esse ponto mínimo corresponde a um valor a partir do qual a função custo começa a aumentar, não sendo portanto a taxa ideal para o treinamento. Atribuímos esse valor encontrado a última camada e, seguindo as boas práticas, definimos cada taxa de aprendizado como 10 vezes menor do que a posterior.

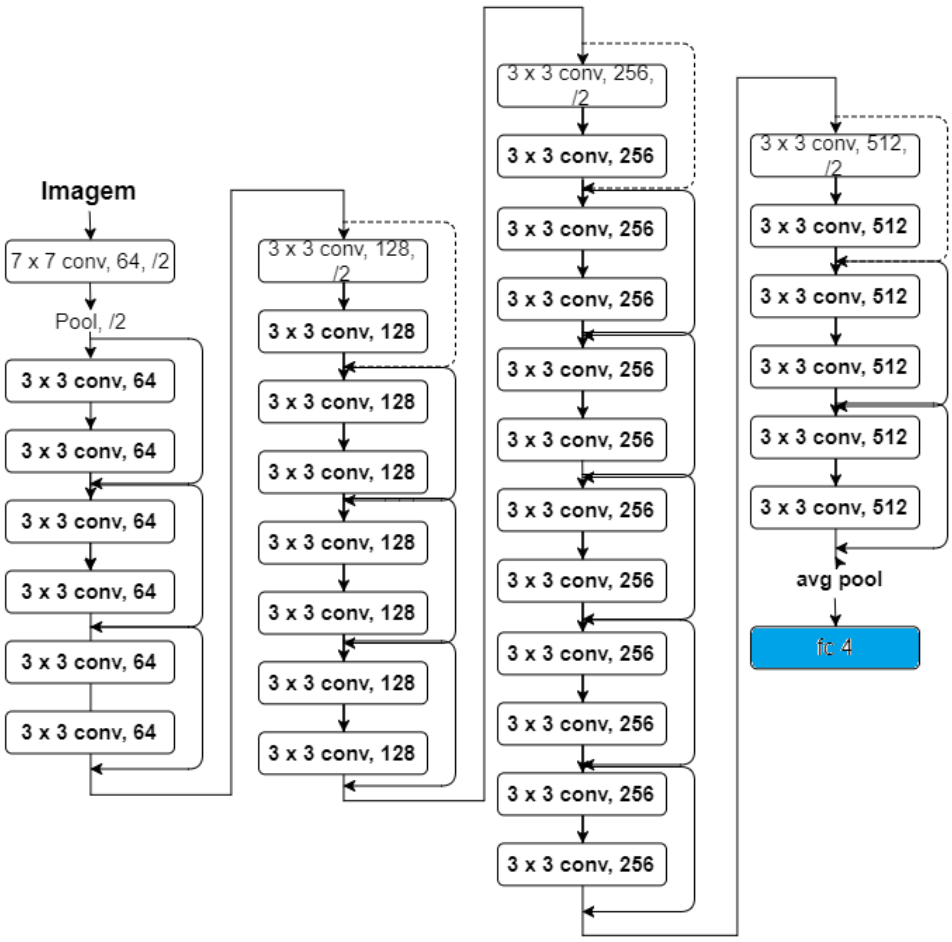
Outra técnica utilizada foi a Descida do Gradiente estocástico com Reinicializações (*Stochastic Gradient Descent with restarts* - SGDR). Durante o treinamento, é possível que a descida de gradiente fique presa em mínimos locais em vez do mínimo global. Ao aumentar subitamente a taxa de aprendizado, a taxa de aprendizado é redefinida no início de cada época para o valor original adotado como um parâmetro, a descida em gradiente pode sair de mínimos locais e voltar a procura pelo mínimo global.

Além das técnicas descritas acima, utilizamos, afim de evitar o *overfitting*, o *Dropout*, que é uma técnica de regularização que previne co-adaptações complexas nos dados de treinamento. O termo *dropout* refere-se ao abandono de unidades em uma rede neural.

Uma rede neural profunda com muitas camadas e grande quantidade de pesos em geral requer uma quantidade de dados de treinamento elevada, o que não é o caso para as bases que estamos considerando. Para lidar com esse desafio, consideramos a técnica de transferência de aprendizado, a qual consiste em utilizar uma rede pré-treinada numa base grande, idealmente num problema parecido, e adaptar algumas de suas camadas à base de interesse. A ideia é que as primeiras camadas, relacionadas aos atributos mais fundamentais e gerais, manter-se-iam quase as mesmas quando treinadas em bases diferentes porém similares.

Sendo assim, executamos novos testes utilizando a técnica de transferência de aprendizado com os valores dos pesos definidos para a ResNet de 34 camadas, treinada para a base de dados Imagenet, considerando um problema de classificação de imagens. Ainda que o problema em questão difira do de reconhecimento de imagens, observamos empiricamente que os atributos aprendidos são úteis para o problema de reconhecimento de emoções através da fala. Naturalmente, é de se esperar que a utilização de uma rede treinada numa base de voz (dedicada, por exemplo, ao reconhecimento de fala)

Figura 14: - ResNet de 34 camadas



produziria resultados ainda melhores quando adaptadas para reconhecimento de emoções. Essa possibilidade será investigada em trabalhos futuros.

Nesta dissertação, avaliamos dois modelos de transferência de aprendizado. Em um primeiro momento, testamos a rede após treinarmos apenas a camada totalmente conectada, responsável pela classificação propriamente dita e marcadas de azul na Figura 14, e mantendo fixos os valores dos pesos das camadas anteriores. Em seguida, produzimos um segundo modelo, onde treinamos a rede como um todo, com os pesos inicializados com os valores do pré-treino, porém usando taxas de aprendizado diferentes para cada camada, de forma que as camadas finais são modificadas mais intensamente do que as iniciais.

3.3.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (do Inglês, *Convolutional Neural Networks* – CNN) são uma categoria de redes neurais que possuem vastas aplicações em classificação

e reconhecimento de imagens e vídeos, segmentação, e geração artificial de imagens. Uma das primeiras arquiteturas de CNN desenvolvida foi a rede LeNet. Essa rede, criada por Yann LeCun, foi utilizada principalmente para a tarefa de reconhecimento de dígitos, com aplicação, inclusive, na leitura automática de cheques, ainda na década de 90. Houve muitas novas arquiteturas propostas nos últimos anos que são melhorias em relação a LeNet, mas que utilizam seus principais conceitos, e que se tornam relativamente mais fáceis de se entender se tivermos uma noção clara da primeira. O ano de 2012 foi muito importante para as redes convolucionais. Foi naquele ano que Alex Krizhevisky venceu uma competição chamada *Imagenet Large Scale Visual Recogniton Competition* (ILSVRC), uma espécie de Olimpíadas da área de Computação Visual, diminuindo a taxa de erro de classificação recorde da competição de 26% para 15%, uma melhoria surpreendente para época. Após este feito, muitos pesquisadores passaram a disputar e a vencer a ILSVRC dos anos seguintes com arquiteturas CNN. A importância de redes convolucionais não se limita ao mundo acadêmico. Uma série de empresas têm utilizado técnicas de aprendizagem profunda em seus serviços. Por exemplo, o reconhecimento de objetos em imagem é uma parte fundamental no desenvolvimento de carros autônomos.

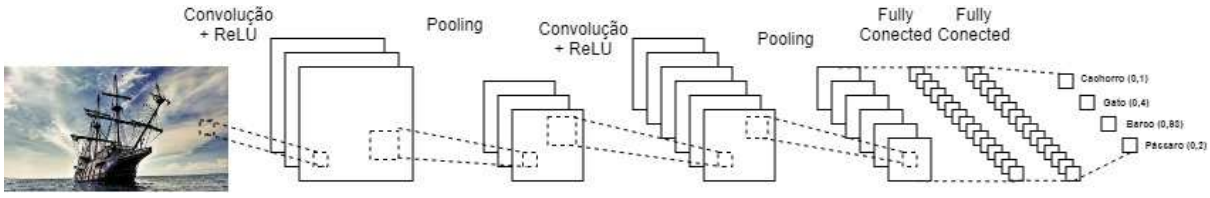
Nós humanos compartilhamos com as máquinas as habilidades de sermos capazes de reconhecer padrões rapidamente, generalizar a partir de conhecimento prévio, e se adaptar a diferentes ambientes, usando os dados dos sentidos, em especial a visão e audição. Diferentemente de nós, as máquinas exigem um aprendizado estruturado, com imagens de entrada e classes correspondentes anotadas. O treinamento consiste em determinar um mapeamento entre uma imagem na entrada do sistema e a probabilidade de que imagem pertença a cada uma das classes.

A forma de representação de uma imagem para um computador é realizada por meio de um conjunto de matrizes, onde cada elemento destas matrizes indica a intensidade de um pixel em determinado ponto da imagem. As dimensões das matrizes são definidas pela resolução da imagem e a profundidade do conjunto é determinada pelo esquema de cores utilizado. Para imagens RGB, o conjunto é formado por três matrizes, uma para cada cor; já uma imagem em escala de cinza possui apenas uma matriz. A cada matriz que representa uma imagem é associado um canal.

A seguir será desenvolvida uma intuição da composição da arquitetura de uma rede LeNet e de seu processo de aprendizagem para a tarefa de reconhecimento de imagens.

A rede neural apresentada na Figura 15 possui uma arquitetura similar à LeNet original, e pode classificar uma imagem em quatro categorias: cachorro, gato, barco ou pássaro. Como podemos ver na figura, a rede recebe como entrada uma imagem de um barco e na saída é atribuída uma porcentagem maior a classe barco, o que indica uma maior probabilidade de que a imagem de entrada pertença a essa classe. Da Figura 15 podemos destacar também os blocos principais que compõem a rede: Convolução, Não-Linearidade (ReLU), *Pooling* e Classificação (*Fully Conected Layer*). A seguir iremos descrever cada um desses principais blocos.

Figura 15: - Arquitetura de uma rede neural convolucional



Convolução

As Redes Neurais Convolucionais devem seu nome ao operador de convolução. O principal objetivo das camadas de convolução é extrair atributos invariantes a translação, que por sua vez são processados pelas camadas subsequentes, até que se obtenha uma representação abstrata, de alto nível, a partir da qual a classificação pode ser feita.

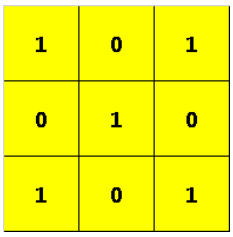
Como já foi visto anteriormente, cada imagem pode ser considerada uma matriz com valores de intensidade de pixels. A Figura 16 representa por meio de uma matriz uma imagem de tamanho 5x5.

Figura 16: - Representação matricial de uma imagem de entrada

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

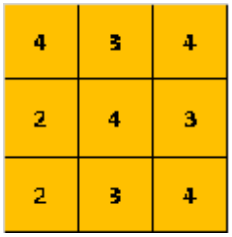
Já a Figura 17 representa uma matriz 3x3 que será o filtro, também conhecido como *kernel* ou detector de características na terminologia de redes convolucionais, a ser utilizado para ilustrar o funcionamento da convolução.

Figura 17: - Filtro de convolução



Assim, a convolução é realizada ao deslizarmos o filtro sobre a imagem de entrada, partindo da posição inicial onde o elemento do filtro correspondente a primeira linha e coluna fique sobre o mesmo elemento da matriz de entrada. É importante enfatizar que o número de *pixels* da imagem que o filtro será deslizado para chegar à próxima posição, que se chama passo, é um hiperparâmetro da rede que deve ser definido pelo projetista. Para cada posição são calculadas as multiplicações entre os elementos sobrepostos das duas matrizes e, após isso, os resultados dessas multiplicações são somados, gerando um resultado para cada posição do filtro sobre a entrada. Cada resultado do processo de convolução se torna um elemento de uma matriz chamada mapa de ativação ou mapa de características, que, utilizando a imagem de entrada e o filtro do exemplo com um passo igual a um pixel, está representado na Figura 18. Cabe destacar que cada elemento do filtro é um peso da rede neural e assim, um parâmetro que deve ser inicializado com valores aleatórios que serão otimizados na etapa de treinamento da rede.

Figura 18: - Mapa de ativação



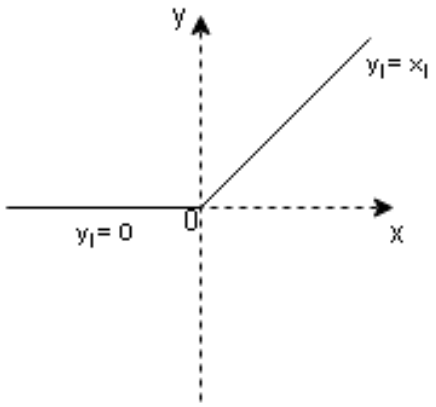
Unidade Linear Retificada

O objetivo da Unidade Linear Retificada (do Ingês, *Rectified Linear Unit* (ReLU)) é inserir uma não-linearidade no sistema, que até a etapa de convolução foi caracterizado somente por operações lineares de soma e multiplicação. A ReLU é a função de ativação mais utilizada em redes convolucionais. Outras funções comuns são a tangente hiperbólica e a função sigmoide.

Em redes convolucionais para o tratamento de imagens, as operações de ReLU são aplicadas a cada *pixel*, ou neurônio de uma camada à esquerda, e tem como efeito prático substituir os *pixels* (entrada) com valor negativo por zero. A Figura 19 apresenta a curva da equação (26), que é a representação matemática da função de ativação ReLU.

$$y = \max(zero, x) \tag{26}$$

Figura 19: - Gráfico da função ReLU



Pooling

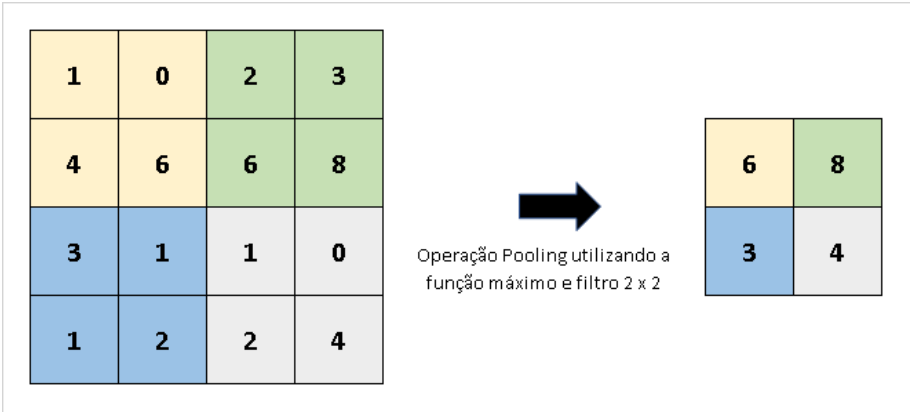
Também chamado de *undersampling* ou *downsampling*, essa operação objetiva reduzir a dimensionalidade da entrada, retendo as informações mais importantes. Os principais motivos para tal são a reduções da complexidade da rede, do custo computacional e do número de parâmetros.

A operação consiste em aplicar a blocos de matrizes a operação escolhida, que podem ser máximo, média ou soma. Um exemplo desta operação está mostrado na Figura 20.

Fully connected Layer

Esta camada, normalmente inserida ao final do processamento, anteriormente à

Figura 20: - Exemplo da operação Pooling



classificação, consiste num *Multilayer Perceptron* tradicional, porém com a última camada tendo como função de ativação a função *softmax*, que computa probabilidades para cada classe. O bloco é chamado totalmente conectado (do Inglês, fully connected) em virtude de todos os neurônios de uma camada estarem conectados a todos os neurônios da camada subsequente.

O objetivo deste bloco de operação é classificar a entrada entre as classes do conjunto de treinamento. A intuição é a de que os neurônios correlacionam as características extraídas nos mapas com as classes.

A função *softmax* normaliza todas as saídas para valores entre zero e um, de forma que o resultado represente a probabilidade de uma determinada saída ser a classe correta para uma dada entrada. O treinamento desta rede é feito utilizando o algoritmo *backpropagation* e está melhor detalhado na subseção seguinte.

Treinamento da rede

O treinamento da rede utiliza o método de descida do gradiente e pode ser resumido em cinco passos, que serão descritos a seguir:

1. Inicialização de todos os filtros, parâmetros e pesos com valores aleatórios;

2. Realização da propagação direta da primeira amostra, isto é, realizar as operações de convolução, ReLU, pooling e a passagem pela camada totalmente conectada. Com isso, encontramos as probabilidades de saída de cada classe via função *softmax*;

3. Cálculo da função de custo a partir das probabilidades de saída obtidas;

4. Realização da propagação reversa para calcularmos os gradientes da função custo com relação aos pesos;

5. Atualização dos pesos por meio do método do gradiente descendente;

6. Repetição das etapas 2 a 4 para todo o conjunto de treinamento.

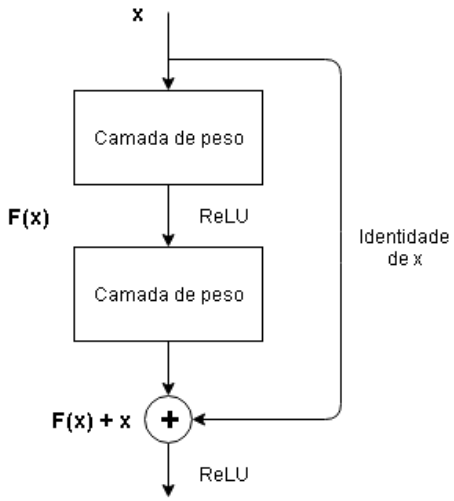
O processamento acima ainda precisa ser repetido inúmeras vezes até que os pesos convirjam para o mínimo da função custo (ou algum de seu mínimo local). O número de iterações é um hiperparâmetro e na literatura é usualmente designado como épocas (*epochs*).

***Residual Network* – ResNet**

O desenvolvimento de pesquisas focadas em redes neurais convolucionais profundas levaram a uma série de avanços na tarefa de classificação de imagens. Essas redes naturalmente integram parâmetros de baixo, médio e alto-nível a classificadores em um modelo multicamada de ponta a ponta. O desafio *ImageNet* é uma evidência que revela que a profundidade da rede é de importância crucial, e os principais resultados desse desafio exploraram modelos muito profundos. Baseados na importância da profundidade, o trabalho [63] se propôs a investigar se a única forma de desenvolver redes melhores é baseado somente em empilhar mais camadas. O resultado dessa pesquisa foi o desenvolvimento de uma versão de uma rede convolucional chamada ResNet, que introduziu o conceito de aprendizagem residual. O desenvolvimento desse conceito partiu de uma premissa de que, em arquiteturas planas, quanto maior a quantidade de camadas, maior o erro. Em vez de esperar que poucas camadas empilhadas se encaixem diretamente no mapeamento subjacente desejado, deixamos explicitamente que essas camadas se encaixem em um mapeamento residual. Formalmente, denotando o desejado mapeamento subjacente como $H(x)$, deixamos as camadas não-lineares empilhadas se ajustarem a outro mapeamento dado por $F(x) = H(x) - x$. O mapeamento original é então substituído por $F(x) + x$, e ao invés de mapearmos $H(x)$, mapeamos $F(x)$, que é o resíduo. Os autores supuseram aqui que é mais fácil otimizar o mapeamento residual do que otimizar o mapeamento original não referenciado. A formulação de $F(x) + x$ pode ser realizada por redes neurais *feedforward* utilizando "conexões de atalho", conforme apresentado na Figura 21, as quais permitem saltar uma ou mais camadas. No caso da ResNet, as conexões de atalho sim-

plesmente realizam o mapeamento de identidade e suas saídas são adicionadas às saídas das camadas empilhadas como também pode ser visto na Figura 21. Além disso, elas não adicionam nenhum parâmetro extra nem complexidade computacional. Toda a rede ainda pode ser treinada de ponta a ponta pelo método de descida do gradiente estocástico com retropropagação e pode também ser facilmente implementada usando bibliotecas comuns sem modificar os solucionadores.

Figura 21: - Aprendizado residual



Para verificar o desempenho dessa solução, os autores realizaram testes com duas redes planas, uma com 18 e outra com 34 camadas, e também com duas ResNets com os mesmos números de camadas. A Figura 22 apresenta as redes ResNet e plana com 34 camadas que foram utilizadas nesse experimento. Os modelos foram testados com o conjunto de dados de classificação do *ImageNet* 2012 que consiste em 1000 classes, 1,28 milhão de imagens de treinamento e 50 mil imagens de validação. A Tabela 6 apresenta as taxas de erro de validação top-5 para os quatro modelos, isto é, o percentual de vezes em que a classe correta estava fora das cinco classes mais prováveis geradas pela rede. Os resultados exibidos por ela mostram que a rede simples de 34 camadas tem um erro de validação mais alto do que a rede de 18 camadas. Já com o aprendizado residual a situação é invertida - a ResNet de 34 camadas é melhor que a ResNet de 18 camadas, uma diferença de 2,8% no desempenho, e ambas desempenharam melhor do que as suas respectivas redes planas.

Figura 22: - Arquiteturas das redes plana e residual com 34 camadas

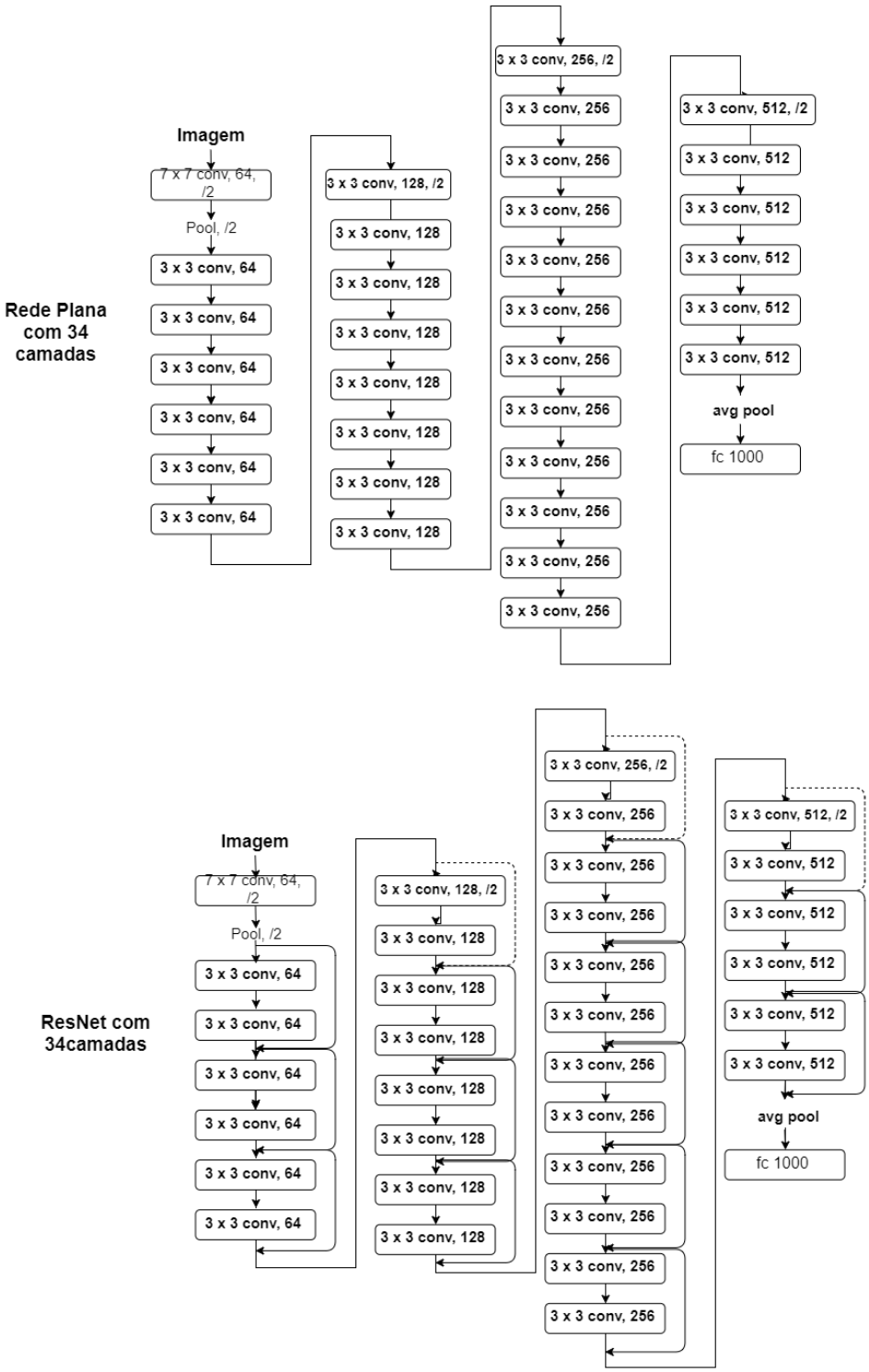


Tabela 6: - Resultados dos testes comparativos entre redes planas e ResNets, em termos de erro de classificação top-5 para desafio *ImageNet*.

	Rede plana	Rede residual
18 camadas	27,88	27,94
34 camadas	25,03	28,54

3.3.3 Transferência de Aprendizado

A efetividade do aprendizado profundo está condicionada a uma quantidade massiva de dados para treinamento em comparação com os métodos tradicionais de aprendizado de máquina, porque precisa de uma grande quantidade de dados para determinar a enorme quantidade de pesos que definem a rede neural. A escala do modelo e a quantidade necessária de dados tem uma relação quase sempre linear. Uma explicação aceitável é que, para um problema específico, o espaço expressivo do modelo deve ser grande o suficiente para descobrir os padrões contido nos dados [64]. As primeiras camadas de um modelo podem identificar padrões de nível mais baixo de abstração no conjunto de dados de treinamento, enquanto as camadas subsequentes identificam padrões de alto-nível, mais abstratos, e mais próximos das classes que se deseja identificar. Possuir dados de treinamento insuficientes é um problema inescapável em alguns casos especiais. A coleta de dados é complexa e cara, o que torna extremamente difícil construir um conjunto de dados com anotações de alta qualidade e larga escala. Devido a esse problema da quantidade de dados disponíveis, na prática, poucos pesquisadores treinam uma Rede Convolucional inteira com inicialização aleatória. Em vez disso, é comum pré-treinar uma rede convolucional com um conjunto de dados muito grande (por exemplo, a base de dados ImageNet [65], que contém 1,2 milhão de imagens com 1000 categorias), e usar a rede como uma inicialização ou um extrator de parâmetros fixo para a tarefa de interesse. Essa técnica é conhecida como transferência de aprendizado. Ela tenta transferir o aprendizado de um domínio fonte para um domínio alvo relaxando a hipótese de que os dados de treinamento devem ser independentes e identicamente distribuídos (i.i.d.) em relação aos dados de teste, o que nos motiva a usar a transferência de aprendizagem para o problema de dados insuficientes de treinamento. Além disso, o modelo no domínio alvo não precisa ser treinado do zero, o que pode reduzir significativamente a demanda de dados de treinamento e o tempo de treinamento no domínio alvo.

Na área de visão computacional, três métodos de transferência de aprendizado são largamente utilizados. No primeiro método, uma rede convolucional é treinada utilizando uma base de dados bem estabelecida, por exemplo a ImageNet, e, então, a camada totalmente conectada é removida e o restante da rede convolucional é tratado como um extrator de parâmetros fixo para a base de dados alvo. Após a extração dos parâmetros dos dados, um classificador linear é treinado para o novo conjunto de dados. Já a segunda estratégia consiste em não apenas substituir e treinar novamente o classificador da rede convolucional com o novo conjunto de dados, mas também ajustar os pesos da rede pré-treinada continuando a retropropagação. É possível fazer o ajuste fino de todas as camadas da rede, ou é possível manter algumas das camadas iniciais corrigidas (devido a problemas de *overfitting*) e apenas ajustar algumas partes superiores da rede. Esta técnica é motivada pela observação de que os parâmetros iniciais de uma rede convolucional contêm mais recursos genéricos (por exemplo, detectores de borda) que devem ser úteis para muitas tarefas, mas as camadas posteriores da rede se tornam progressivamente mais específicas para os detalhes das classes contidas no conjunto de dados original. No caso do ImageNet, por exemplo, que contém muitas raças de cães, uma parte significativa do poder de representação da rede pode ser dedicada a características que são específicas para diferenciar entre raças de cães. O último método leva em consideração que uma rede convolucional moderna leva de 2 a 3 semanas para treinar em várias GPUs utilizando o ImageNet. Devido a esse grande tempo de treinamento é comum ver pessoas compartilhando seus pontos de verificação finais da rede convolucional para o benefício de outras pessoas que podem usar as redes para o ajuste fino.

A escolha do método a ser utilizado é uma função de vários fatores, mas os dois mais importantes são o tamanho do novo conjunto de dados (pequeno ou grande) e sua similaridade com o conjunto de dados original. Tendo em mente que os parâmetros de uma rede convolucional são mais genéricos nas camadas iniciais e mais específicos do conjunto de dados original nas camadas posteriores, existem algumas regras comuns para navegar pelos quatro principais cenários. No primeiro caso, o conjunto de dados alvo é pequeno e semelhante ao conjunto de dados fonte. Como a quantidade de dados é pequena, não é uma boa ideia ajustar a rede convolucional devido ao problema de *overfitting*. No entanto, os dados são semelhantes aos dados originais, então, esperamos que os recursos de nível superior na rede também sejam relevantes para esse conjunto de dados. Portanto,

a melhor ideia pode ser treinar um classificador linear utilizando os parâmetros extraídos pela CNN treinada utilizando a base de dados fonte. Para o caso de termos um conjunto de dados alvo grande e semelhante ao conjunto de dados fonte, como temos mais dados, podemos ter mais confiança de que não causaremos *overfitting* se tentarmos ajustar a rede inteira. Em um outro cenário, podemos ter um conjunto de dados alvo pequeno e muito diferente do conjunto de dados original. Com poucos dados, provavelmente é melhor treinar apenas um classificador linear no final da rede. Além disso, considerando que os dados dos dois conjuntos são diferentes, não será efetivo treinarmos o classificador linear após as últimas camadas da rede, visto que essas camadas contém os parâmetros mais específicos do conjunto de dados. Em vez disso, pode funcionar melhor treinar o classificador em algum ponto inicial da rede. Resta-nos o cenário onde o conjunto de dados alvo é grande e muito diferente do conjunto de dados original. Como o conjunto de dados é muito grande, podemos esperar que possamos treinar uma CNN a partir do zero. No entanto, na prática, muitas vezes ainda é benéfico inicializar com pesos de um modelo pré-treinado. Nesse caso, teríamos dados e confiança suficientes para realizar um ajuste fino em todas as camadas da rede.

4 RESULTADOS

4.1 Base de Dados EmoDb

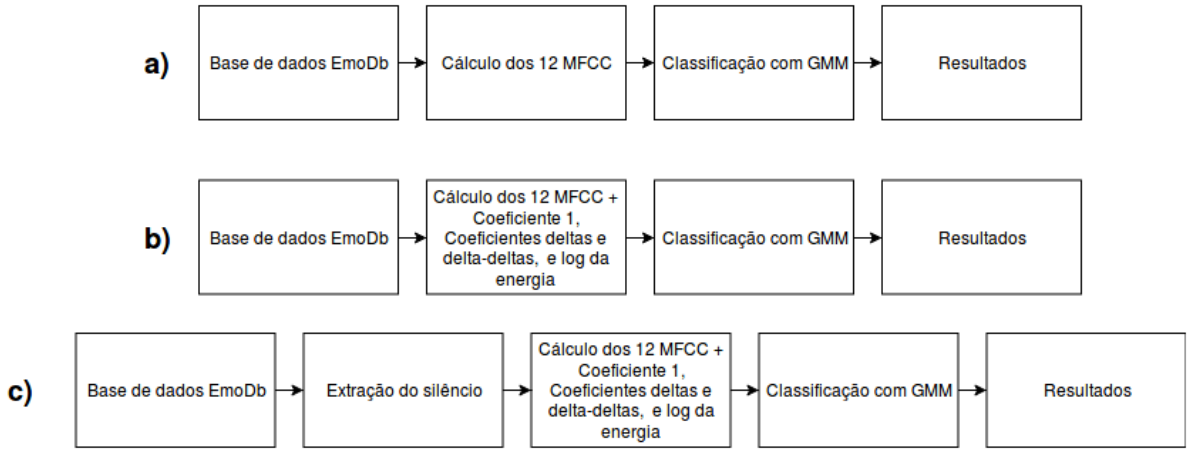
Iniciamos nossos experimentos com a base EmoDB. Por ser uma base pequena, realizamos apenas experimentos com os classificadores GMM e PNN, visto que as redes neurais profundas tipicamente necessitam de uma grande quantidade de amostras para serem treinadas. Conforme explicado na metodologia, nosso interesse nesses experimentos foi verificar a eficácia do aumento do número de parâmetros e também da extração dos períodos de silêncio das amostras de áudio nos resultados desses classificadores na tarefa de reconhecimento de emoções.

4.1.1 Modelo de Mistura de Gaussianas

A Figura 23 apresenta os procedimentos adotados em cada experimento realizado com a combinação da base EmoDb e o classificador GMM. Nela, os diagramas a) e b) apresentam os experimentos sem o pré-processamento das amostras de áudio; e o diagrama c), o experimento com as amostras da base EmoDb sem os períodos de silêncio. Realizamos os experimentos nessa ordem pois, primeiramente, o nosso interesse foi de avaliar os resultados do classificador em relação a quantidade de parâmetros. Para esta avaliação, comparamos os resultados do primeiro com o segundo experimento. Em seguida, avaliamos também a eficácia da extração do silêncio na classificação utilizando o GMM por meio da comparação do segundo experimento com o terceiro.

No primeiro experimento, apresentado na Figura 23a), usamos como *features* os doze coeficientes mel-cepstrais das amostras de áudio sem pré-processamento. Já no segundo e no terceiro, apresentados na Figura 23b) e na Figura 23c), além dos doze MFCC, usamos também o coeficiente mel-cepstral de número um, o logaritmo da energia dos quadros dos sinais de áudio, os coeficientes deltas e os coeficientes delta-deltas. Com o cálculo desses novos parâmetros, aumentamos o número de parâmetros para quarenta e dois. Além disso, em todos os três experimentos foi utilizado o algoritmo de agrupamento Figueiredo-Jain. A Tabela 7 apresenta o número médio de Gaussianas encontradas pelo algoritmo para cada classe em cada experimento. Importante ressaltar que a variação do número de curvas se apresentou inversamente proporcional a variação de parâmetros.

Figura 23: - Procedimentos adotados em cada experimento realizado com a combinação da base EmoDb e o classificador GMM



Ainda, no terceiro experimento todas as classes foram representadas com Gaussianas simples.

Tabela 7: - Relação da quantidade média de curvas Gaussianas utilizadas para representar cada classe em cada experimento

Classe	Quantidade média de curvas Gaussianas		
	Experimento 1	Experimento 2	Experimento 3
Felicidade	20	1	1
Neutra	21	2	1
Raiva	21	2	1
Tristeza	19	2	1

Devido à aleatoriedade da escolha das amostras utilizadas nas etapas de treinamento e teste dos experimentos, foram realizadas dez simulações dessas etapas em cada experimento, e os resultados das etapas de teste de cada experimento foram organizados em uma matriz de confusão cuja finalidade é exibir a quantidade de amostras pertencentes a cada classe real que foram classificadas em cada classe prevista. A partir desses resultados, foram calculados os valores médios de todas as posições dessas matrizes e, então, por meio desses valores, calculamos as matrizes de confusão percentuais de cada experimento.

Realizando, primeiramente, uma análise relativa ao efeito de aumentar a quantidade de atributos no desempenho do classificador, podemos verificar com base nas matrizes do primeiro experimento, apresentada na Tabela 8, e do segundo experimento, apresentada na Tabela 9, que, de forma geral, esse aumento provocou uma melhora na classificação geral em 3,85 pontos percentuais. No primeiro experimento, o GMM classi-

Tabela 8: - Matriz confusão do primeiro experimento com a base EmoDb e o GMM

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	8,18%	0,00%	0,00%	0,00%
	Neutra	0,91%	50,83%	0,00%	0,00%
	Raiva	89,55%	2,08%	100,00%	0,00%
	Tristeza	1,36%	47,08%	0,00%	100,00%
Geral		**69,23%**			

ficou corretamente 69,23% das amostras, enquanto que no segundo 73,08% das amostras foram classificadas corretamente. Além do resultado geral, é interessante também compararmos os resultados dos experimentos para cada classe de forma separada. Assim, podemos observar que para as classes Felicidade e Neutra, o GMM teve um resultado melhor com o aumento do número de parâmetros. Já para as classes Raiva e Tristeza, esse resultado piorou.

Tabela 9: - Matriz confusão do segundo experimento com a base EmoDb e o GMM

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	48,64%	2,08%	28,97%	0,00%
	Neutra	8,64%	80,42%	0,26%	3,16%
	Raiva	42,27%	0,42%	70,77%	0,00%
	Tristeza	0,45%	17,08%	0,00%	96,84%
Geral		**73,08%**			

Conforme exposto anteriormente, nesta subseção avaliamos também o efeito da extração do silêncio das amostras no desempenho do GMM. Essa avaliação se deu por meio da comparação do resultado do segundo experimento com o do terceiro, apresentado na Tabela 10. Vale relembrar que nesses dois experimentos foram utilizados quarenta e dois atributos. Conforme apresentado acima, o segundo experimento teve 73,08% das amostras classificadas corretamente. Já o resultado geral do terceiro experimento foi de 85,77%, uma diferença de 12,69 pontos percentuais. Dessa análise podemos afirmar que quando realizamos a extração do silêncio das amostras da base EmoDb, do que quando esse pré-processamento não foi realizado. Ao analisarmos também os resultados do classificador para cada classe separadamente, podemos afirmar que a técnica de extração do silêncio utilizada no experimento acarretou um resultado melhor para todas as classes.

Outro fato que chamou a atenção nos resultados de todos os três experimentos foi

Tabela 10: - Matriz confusão do terceiro experimento com a base EmoDb e o GMM

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	53,64%	0,00%	5,64%	0,00%
	Neutra	14,09%	99,17%	3,59%	4,21%
	Raiva	32,27%	0,00%	90,77%	0,00%
	Tristeza	0,00%	0,83%	0,00%	95,79%
Geral		**85,77%**			

a grande quantidade de amostras da classe Felicidade que foram classificadas pelo GMM como se pertencessem a classe Raiva.

4.1.2 Redes Neurais Probabilísticas

Também foram realizados testes com a base de dados EmoDb e uma rede neural probabilística. Nesta etapa, executamos três experimentos com o mesmo objetivo da subseção anterior: avaliar se o aumento do número de amostras e a extração do silêncio da base EmoDb acarretam em um aumento no desempenho da rede probabilística na tarefa de reconhecimento de emoções. Além disso, a escolha dos parâmetros de cada experimento também foi igual. No primeiro, utilizamos os doze coeficientes mel-cepstrais. Já no segundo e terceiro experimentos foram calculados os doze coeficientes MFCC padrão, o coeficente MFCC de número um, o logaritmo da energia do quadro da amostra de áudio, os coeficientes deltas e os coeficientes delta-deltas como parâmetros da rede. Entretanto, no terceiro experimento foram extraídos os períodos de silêncio das amostras de áudio da base EmoDb.

Tabela 11: - Matriz confusão do primeiro experimento com a base EmoDb e a PNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	0,00%	0,00%	0,00%	0,00%
	Neutra	0,00%	12,50%	0,00%	0,00%
	Raiva	100,00%	4,17%	100,00%	0,00%
	Tristeza	0,00%	83,33%	0,00%	100,00%
Geral		**53,13%**			

No primeiro experimento, obtivemos o resultado exposto na Tabela 11 que representa a matriz confusão percentual do experimento. Analisando essa matriz, podemos

perceber que essa rede não conseguiu reconhecer as amostras pertencentes a classe Felicidade, todas elas foram classificadas como se pertencessem a categoria Raiva. A rede também apresentou uma acurácia baixa para a classe Neutra, uma vez que apenas 12,50% das amostras desse grupo foram classificadas corretamente. Além disso, a rede apresentou uma confusão dessas amostras com a classe Tristeza. Entretanto, a rede classificou corretamente todos os exemplos de teste das classes Raiva e Tristeza. A acurácia geral da rede nesse experimento foi de 53,13%.

Quando aumentamos o número de atributos utilizados como parâmetros pela rede, a classificação das amostras das classes Felicidade e Neutra aumentaram para 22,73% e 95,83%, respectivamente, conforme apresentado na Tabela 12. Esses resultados proporcionaram um aumento de 23,30 pontos percentuais na acurácia geral da rede, quando comparado ao primeiro experimento. Entretanto, a maior parte das amostras da classe Felicidade continuou a ser confundida pela rede como se pertencessem ao grupo Raiva.

Tabela 12: - Matriz confusão do segundo experimento com a base EmoDb e a PNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	22,73%	0,00%	0,00%	0,00%
	Neutra	4,55%	95,83%	0,00%	0,00%
	Raiva	72,73%	0,00%	100,00%	0,00%
	Tristeza	0,00%	4,17%	0,00%	100,00%
Geral		**79,64%**			

Diferente do observado nos experimentos com o GMM, para a nossa rede probabilística, a técnica de extração de silêncio não melhorou o resultado da classificação pela rede. Podemos confirmar isto observando a Tabela 13, que é a matriz confusão do experimento no qual utilizamos essa técnica. Da figura, podemos perceber que as amostras de teste das classes Felicidade, Raiva e Tristeza foram mais confundidas pela rede como se pertencessem a outras classes do que no experimento anterior. Um fato a se destacar foram os 16,36% de amostras da classe Felicidade que foram confundidas como se pertencessem a classe Neutra. Nos experimentos anteriores não foi observada essa confusão tão acentuada.

Tabela 13: - Matriz confusão do terceiro experimento com a base EmoDb e a PNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	14,55%	0,00%	0,00%	0,00%
	Neutra	16,36%	98,33%	16,92%	4,21%
	Raiva	69,09%	0,00%	83,08%	2,11%
	Tristeza	0,00%	1,67%	0,00%	93,68%
Geral		**74,04%**			

4.2 Base de Dados IEMOCAP

A IEMOCAP é uma base de dados que possui, somadas as quatro classes utilizadas nesse trabalho, 4490 amostras distribuídas da seguinte forma: 595 amostras da classe Felicidade, 1708 da classe Neutra, 1103 da classe Raiva e 1084 da classe Tristeza. Dado esse grande número de amostras, escolhemos utilizar essa base para realizar testes com uma rede neural convolucional e verificar a influência das técnicas de aumento da base e de transferência de aprendizado no desempenho da classificação de emoções na fala por uma rede convolucional. Para termos uma base de comparação para os resultados da rede convolucional, iniciamos os testes com essa base de dados utilizando o GMM como classificador.

4.2.1 Modelo de Mistura de Gaussianas

De posse dos resultados da seção anterior, onde o GMM alcançou um resultado melhor do que a rede probabilística, foi definido que esse classificador também seria utilizado para experimentos com os dados da base IEMOCAP a fim de termos um parâmetro de comparação para os resultados dos experimentos que utilizaram a rede convolucional.

Nestes experimentos utilizamos também o algoritmo Figueiredo-Jain para encontrar a máxima verossimilhança dos parâmetros das misturas de Gaussianas. Como discutido anteriormente, este algoritmo calcula de maneira automática a quantidade de curvas utilizadas para representar uma determinada classe. Na Tabela 14 são apresentados os números médios de curvas Gaussianas utilizadas em cada experimento.

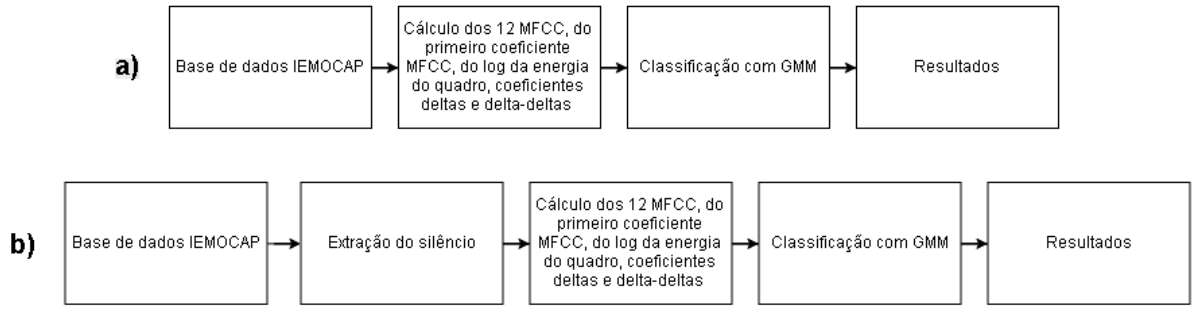
Conforme mostra a Figura 24, nessa subseção já iniciamos os testes utilizando os doze coeficientes cepstrais, o primeiro coeficiente MFCC, o valor logaritmos da energia de cada quadro, os coeficientes deltas e delta-deltas. Essa escolha foi feita devido a esse con-

Tabela 14: - Relação da quantidade média de curvas Gaussianas utilizadas para representar cada classe em cada experimento com a base de dados IEMOCAP

Classe	Quantidade média de curvas Gaussianas	
	Experimento 1	Experimento 2
Felicidade	1	1
Neutra	2	2
Raiva	2	2
Tristeza	2	2

junto de parâmetros ter obtido um resultado melhor nos experimentos com o EmoDb. No segundo experimento, utilizamos os mesmos coeficientes, mas antes de realizar a extração dos parâmetros, processamos os áudios retirando os períodos de silêncio.

Figura 24: - Procedimentos adotados em cada experimento realizado com a combinação da base IEMOCAP e o classificador GMM



Entretanto, diferente do resultado apresentado pelos experimentos realizados com os dados da base EmoDb, os experimentos utilizando GMM para a base IEMOCAP não apresentaram um melhor desempenho de classificação do experimento com os áudios sem os períodos de silêncio. Comparando a Tabela 15 com a Tabela 16, podemos perceber que o segundo experimento teve resultados ligeiramente melhores na classificação das amostras de teste das classes Felicidade - que ainda assim ficou com apenas 6,78% de acerto - e Raiva, com 86,59% das amostras classificadas corretamente. Já no primeiro experimento o classificador acertou a classificação de um número maior de amostras de teste das classes Neutra e Tristeza. Comparando com o resultado do segundo experimento para essas duas emoções, a diferença foi de 11,02% e 23,67% pontos percentuais, respectivamente. No geral, o primeiro e segundo experimentos atingiram um resultado de 66,83% e 56,28% de acerto na classificação das amostras de teste, respectivamente.

Tabela 15: - Matriz confusão do primeiro experimento com a base IEMOCAP e o GMM

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	0,34%	0,06%	0,00%	0,00%
	Neutra	31,02%	60,69%	7,58%	2,40%
	Raiva	6,61%	1,26%	83,41%	0,16%
	Tristeza	62,03%	37,98%	9,01%	97,44%
Geral		**66,83%**			

Tabela 16: - Matriz confusão do segundo experimento com a base IEMOCAP e o GMM

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	6,78%	1,04%	0,19%	2,81%
	Neutra	37,80%	49,67%	16,85%	27,57%
	Raiva	1,19%	0,59%	86,59%	0,00%
	Tristeza	54,80%	53,18%	1,78%	73,77%
Geral		**56,28%**			

4.2.2 Redes Profundas Convolucionais

Conforme apresentado anteriormente, para esta etapa de testes, utilizamos duas bases de dados, a IEMOCAP original e outra com todos os dados originais da primeira, mais três amostras novas para cada arquivo de áudio original. Na primeira amostra, realizamos o deslocamento do pitch da fala; na segunda, o aumento da velocidade do áudio; e na terceira, a extração do silêncio. Nosso objetivo nessa subseção foi verificar a eficácia das técnicas de aumento de dados e do mecanismo de Transferência de Aprendizado.

A Figura 25 apresenta os experimentos realizados nesta subseção. Como podemos observar, foram realizados quatro experimentos, a diferença entre eles consiste na utilização da técnica de transferência de aprendizado e no aumento da base de dados. Nos dois primeiros, foram realizados testes sem a transferência de aprendizado, já nos outros dois esta técnica foi utilizada. Os primeiros experimentos de cada dupla de testes foi com a base IEMOCAP original, os seguintes utilizaram a base aumentada. Em todos os quatro, foi gerado o espectrograma de cada amostra de áudio para o treinamento e teste da nossa ResNet.

No primeiro experimento, treinamos a ResNet com os pesos inicializados de forma aleatória, e geramos a curva de custo por taxa de aprendizado, que pode ser vista na

Figura 25: - Procedimentos adotados em cada experimento realizado com a combinação da base IEMOCAP e o classificador CNN (ResNet).

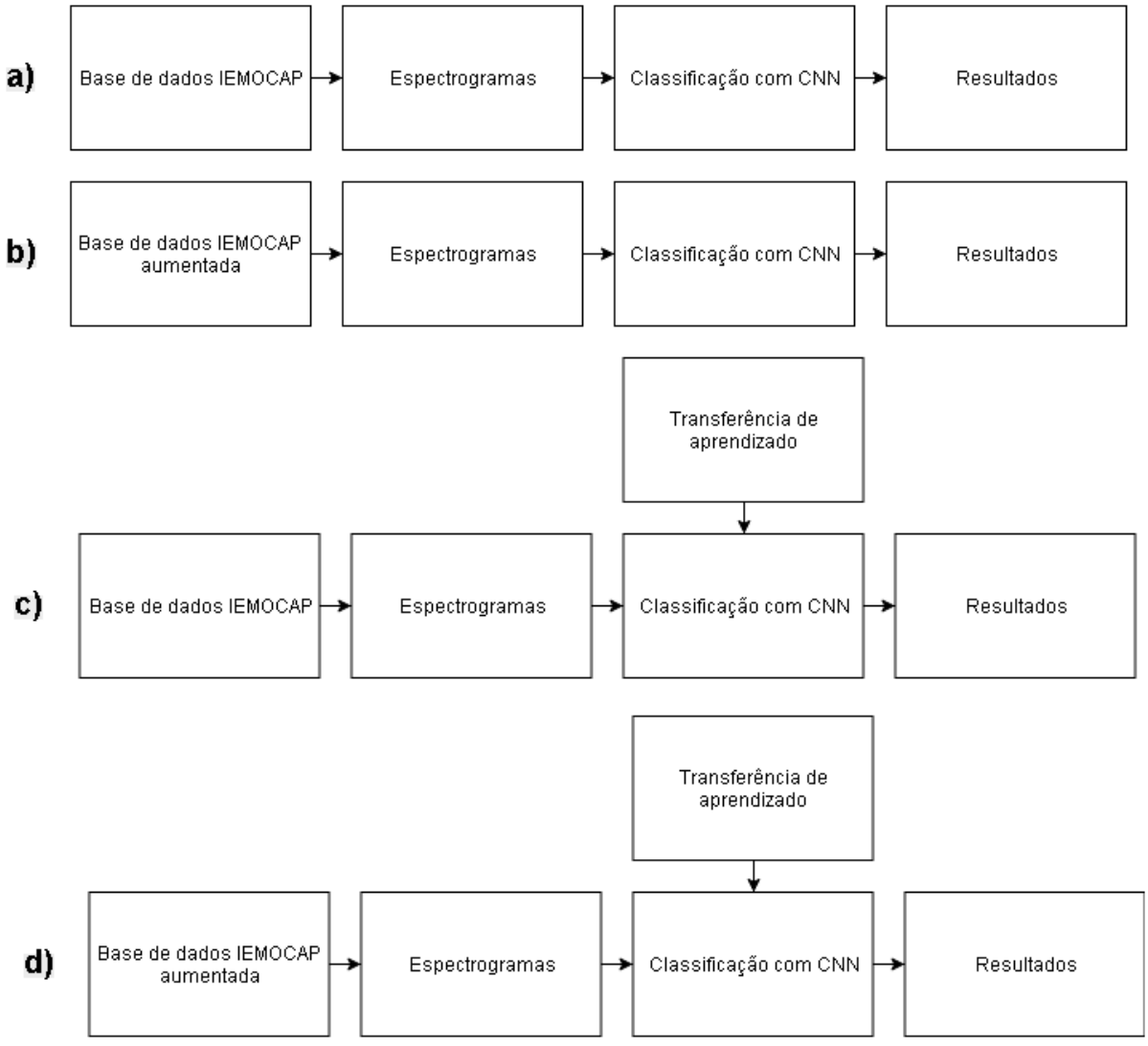
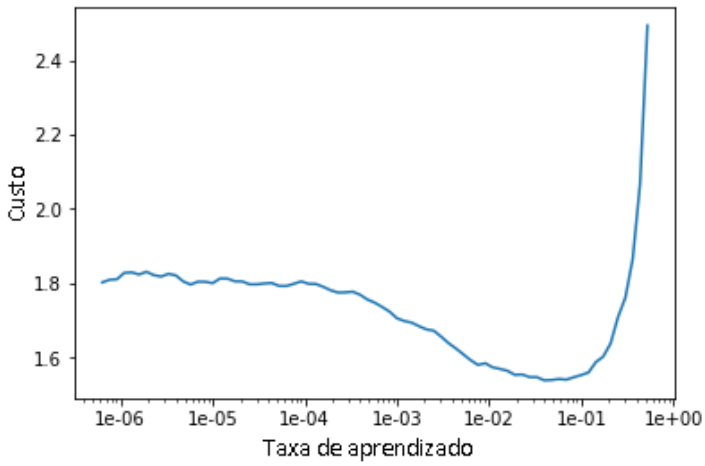


Figura 26, seguindo o procedimento descrito na metodologia para determinação da taxa de aprendizagem. Com o auxílio deste gráfico, definimos a taxa de aprendizado da rede como 10^{-2}. Após essa definição, realizamos o treinamento e a validação da rede com essa taxa em quinze épocas. O resultado dessa etapa está apresentado na Tabela 17, onde podemos verificar que a acurácia geral da rede não variou muito entre as épocas, e seu valor final foi de 0,598608. Além disso, o resultado do custo de validação apresenta *overtraining* desde o início do treinamento desta rede. Para analise dos resultados de cada classe, foi gerada a matriz confusão do experimento. Desta matriz, apresentada na Tabela 18, podemos afirmar que a classe com o pior desempenho foi a Felicidade (assim como nos experimentos com GMM), que ficou com 16,10% de acerto. Esse resultado é devido a essa classe possuir a menor quantidade de amostras. Além disso, todas as outras classes apresentaram uma acurácia maior que 60%, onde a classe de melhor resultado foi a Raiva, com 71,05% das amostras de teste classificadas corretamente. Vale destacar que o resultado geral de classificação pela rede convolucional (59,86%) não foi melhor do que o resultado geral do primeiro experimento com o GMM para a base de dados IEMOCAP.

Figura 26: - Gráfico para encontrar a taxa de aprendizado da rede no experimento 1



Já no segundo experimento, conforme mencionado anteriormente, executamos a técnica de aumento da base de dados IEMOCAP em quatro vezes o seu tamanho original, e seguimos as mesmas etapas do experimento anterior. Geramos o gráfico - apresentado na Figura 27 - para encontrarmos a taxa de aprendizado para essa rede. Com o auxílio desse gráfico, definimos a taxa como 10^{-2}.

Com essa taxa de aprendizado definida, executamos a rede em 15 épocas, assim

Tabela 17: - Resultado da classificação do primeiro experimento com a taxa de aprendizado 10^{-2}

Época	Custo de Treino	Custo de validação	Acurácia
1	1.286629	1.027866	0.590487
2	1.088441	1.033717	0.589327
3	0.967714	1.044087	0.605568
4	0.911319	1.188079	0.596288
5	0.853787	1.182761	0.609049
6	0.797502	1.080468	0.604408
7	0.700675	1.098917	0.588167
8	0.616391	1.178904	0.569606
9	0.511176	1.250570	0.598608
10	0.423265	1.456313	0.588167
11	0.329858	1.369980	0.597448
12	0.257469	1.426652	0.585847
13	0.203169	1.504504	0.596288
14	0.159058	1.507039	0.598608
15	0.137549	1.505375	0.598608

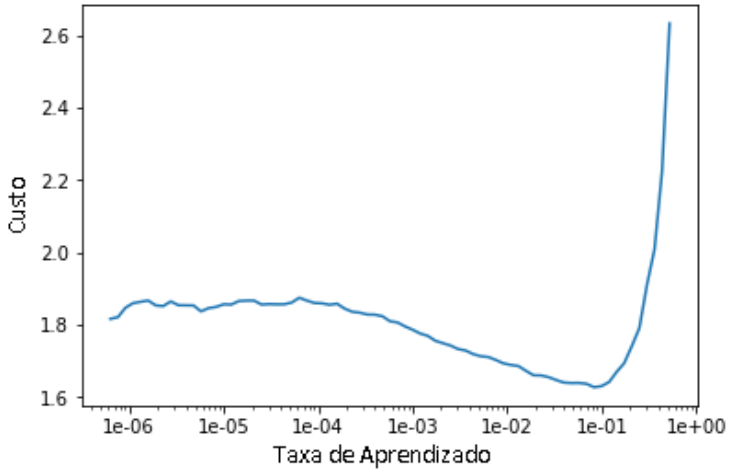
Tabela 18: - Matriz confusão do primeiro experimento com a base IEMOCAP e a CNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	16,10%	7,42%	5,26%	4,61%
	Neutra	54,24%	67,66%	16,84%	29,03%
	Raiva	9,32%	10,98%	71,05%	4,61%
	Tristeza	20,34%	13,95%	6,84%	61,75%
Geral		**59,86%**			

Tabela 19: - Resultado da classificação do segundo experimento com a taxa de aprendizado 10^{-2}

Época	Custo de treino	Custo de validação	Acurácia
1	1.102713	0.961955	0.616370
2	0.988591	0.945531	0.620267
3	0.902019	0.868427	0.651448
4	0.829157	0.829457	0.668708
5	0.744950	0.815851	0.673998
6	0.720630	0.811695	0.678452
7	0.720989	0.795426	0.693207
8	0.610956	0.820782	0.685134
9	0.449865	0.820122	0.693764
10	0.292121	0.940352	0.687082
11	0.151875	0.979785	0.728285
12	0.090401	1.135558	0.718820
13	0.035021	1.104437	0.730512
14	0.019320	1.114583	0.734410
15	0.013764	1.146373	0.735802

Figura 27: - Gráfico para encontrar a taxa de aprendizado da rede no experimento 2



como no experimento anterior. Os resultados da acurácia e custo de treino e de validação dessas épocas podem ser vistos na Tabela 19. Dela podemos afirmar que já na primeira época a rede começou com uma acurácia maior do que o experimento anterior terminou. Além disso, a acurácia aumentou em todas as épocas, atingindo um valor de 0.735802 na última. Este fato demonstra que a técnica de aumento de dados é bastante útil na tarefa de reconhecimento de emoções por uma rede neural convolucional. Com a intenção de identificarmos quais classes contribuíram para esse resultado geral, geramos a matriz confusão desse experimento que é apresentada na Tabela 20. Dela, podemos verificar que todas as classes tiveram um desempenho melhor se comparado ao experimento anterior, onde podemos destacar a classe Felicidade associada a uma melhora de 22,80 pontos percentuais.

Tabela 20: - Matriz confusão do segundo experimento com a base IEMOCAP e a CNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	38,90%	4,79%	2,24%	2,88%
	Neutra	32,98%	79,00%	14,94%	15,42%
	Raiva	12,05%	6,71%	79,29%	3,57%
	Tristeza	16,07%	9,50%	3,53%	78,14%
Geral		**73,58%**			

No terceiro experimento, começamos a testar a rede com a técnica de transferência de aprendizado. Nele, alimentamos a rede, já pré-treinada, com as amostras de treino da base de dados IEMOCAP original, realizando o ajuste apenas da última camada da rede,

que é responsável pela classificação propriamente dita.

Ao aplicarmos o procedimento para gerar o gráfico que relaciona o custo e a taxa de aprendizado, obtivemos o gráfico exibido na Figura 28. Nela, observamos que para a taxa de 10^{-1} temos o menor valor de erro. Por convenção, devemos utilizar um valor 10 vezes menor que esse valor para nossa taxa de aprendizado, que neste caso é igual a 10^{-2}, dando-lhe uma taxa mais cautelosa durante a retropropagação na etapa de treinamento.

Figura 28: - Gráfico para encontrar a taxa de aprendizado da rede no experimento 3

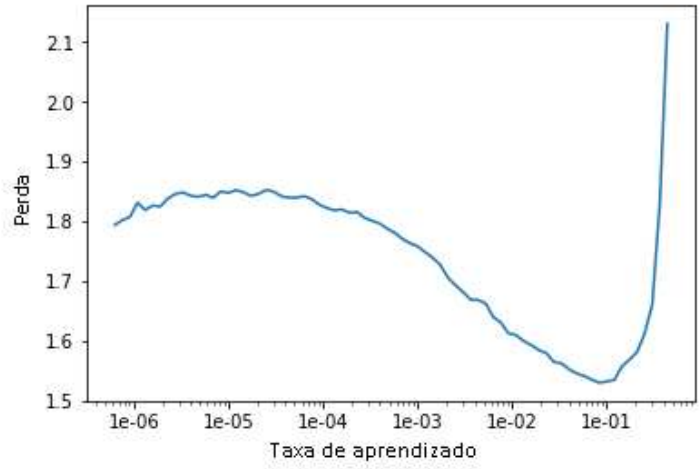


Tabela 21: - Resultado da classificação com a taxa de aprendizado 10^{-2} no terceiro experimento

Épocas	Custo de treino	Custo de validação	Acurácia
1	1.241078	1.034500	0.605568
2	1.060820	0.968925	0.592807
3	0.936409	0.973663	0.631090
4	0.795626	0.885393	0.660093
5	0.650062	0.892627	0.655452

Após definirmos a taxa de aprendizado na rede, realizamos o treinamento e a validação da mesma em cinco épocas e obtivemos o resultados listados na Tabela 21. Dela podemos observar que após a quinta época a acurácia geral da rede foi de 0.655452 com custo de treino igual a 0.650062 e custo de validação igual 0.892627.

Nosso próximo passo foi verificar uma segunda abordagem da técnica de transferência de aprendizado. Para isso, treinamos a rede como um todo utilizando as amostras da nossa base de dados. Então, executamos o procedimento para encontrar a nova taxa de aprendizado - onde foi gerado o novo gráfico, apresentado na Figura 29 - e verificamos

Figura 29: - Gráfico para encontrar a nova taxa de aprendizado da rede no experimento 3

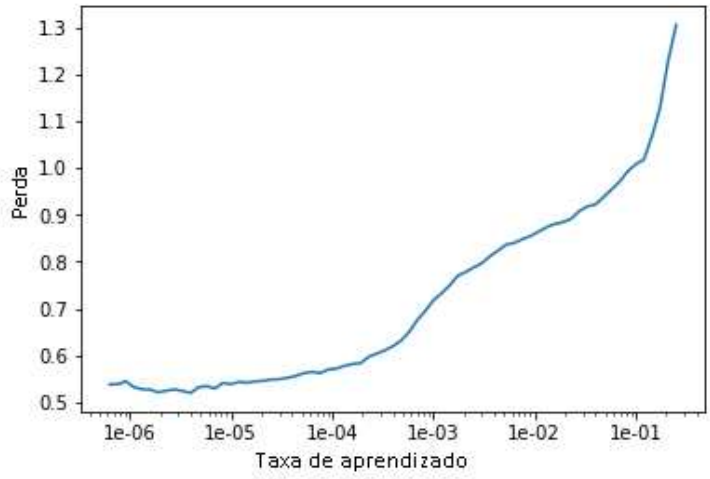


Tabela 22: - Resultado da classificação com a taxa de aprendizado 10^{-7} no terceiro experimento

Época	Custo de treino	Custo de validação	Acurácia
1	0.548789	0.923870	0.658933
2	0.480139	1.044008	0.649652
3	0.381412	1.220775	0.612529
4	0.286155	1.400898	0.621810
5	0.177816	1.486815	0.635731
6	0.092419	1.593256	0.660093
7	0.052476	1.635659	0.664733
8	0.030814	1.654233	0.654292
9	0.019481	1.674734	0.650812
10	0.013692	1.634277	0.653132

que a nova taxa de aprendizado é igual a 10^{-7}. De posse desta taxa, treinamos a rede em dez épocas. Os resultados dessa etapa estão detalhados na Tabela 22. Dela, podemos verificar que ocorreu *overtraining* para essa rede. Comparando o resultado da última época, 0.653132, com o resultado da acurácia antes de treinarmos todas as camadas da rede, que foi de 0.655452, podemos afirmar que esse procedimento não melhorou o nosso resultado.

Para verificarmos como foi o desempenho no reconhecimento de cada classe, geramos a matriz confusão desse experimento que está exibida na Tabela 23. Nela, podemos verificar que a classe Raiva obteve o melhor desempenho, 76,39%; e a Felicidade teve o pior, com apenas 23,73% de acerto. Outro ponto relevante foi a classificação de 40,68% das amostras de teste da classe Felicidade, que foram classificadas como Neutra. Essa foi

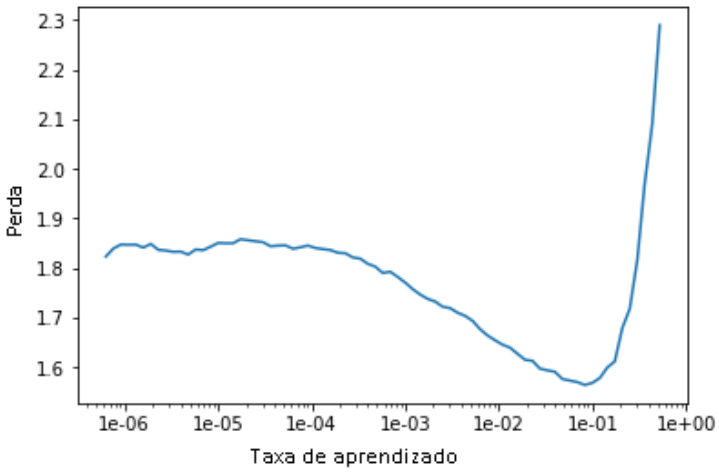
a maior confusão realizada por esse classificador. A segunda foi a classificação de 25,35% das amostras da categoria Tristeza, que foram classificadas também como Neutra.

Tabela 23: - Matriz confusão do terceiro experimento com a base IEMOCAP e a CNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	23,73%	8,36%	3,24%	5,07%
	Neutra	40,68%	72,03%	16,67%	25,35%
	Raiva	11,86%	6,11%	76,39%	2,30%
	Tristeza	23,73%	13,50%	3,70%	67,28%
Geral		**65,31%**			

O quarto e último experimento com a CNN consistiu em alimentar a rede com os espectrogramas gerados da base IEMOCAP aumentada. Nele, utilizamos a mesma ResNet pré-treinada do terceiro experimento. Após gerarmos os espectrogramas, definimos a taxa de aprendizado da rede como 10^{-2} por meio do procedimento que gera a curva que relaciona essa taxa com o custo da rede pré-treinada exibida na Figura 30. De posse desse valor, treinamos a camada *Fully connected* em cinco épocas. Após termos a rede treinada ela passou pelo processo de validação. Os resultados dessa etapa e da etapa de treinamento podem ser verificados na Tabela 24. Nela podemos ver que após a quinta época já alcançamos uma acurácia maior do que no experimento anterior, 0.671492.

Figura 30: - Gráfico para encontrar taxa de aprendizado da rede no experimento 4



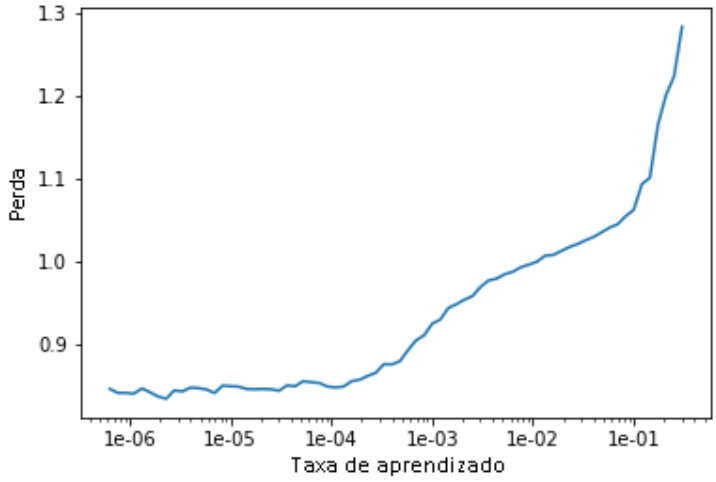
Nesse experimento, novamente executamos o procedimento de treinar a rede como um todo utilizando as amostras da nossa base de dados. A seguir, geramos o gráfico

Tabela 24: - Resultado da classificação com a taxa de aprendizado igual a 10^{-2} no quarto experimento

Época	Custo de treino	Custo de validação	Acurácia
1	1.201774	1.545130	0.483296
2	1.352582	1.126734	0.553174
3	1.144411	1.012508	0.619432
4	0.976383	0.899688	0.643653
5	0.873673	0.836139	0.671492

apresentado na Figura 31 para encontrar a nova taxa de aprendizado, que foi definida como 10^{-7}.

Figura 31: - Gráfico para encontrar a nova taxa de aprendizado da rede no experimento 4



Após essa etapa, treinamos e testamos a rede. Os valores para o custo de treinamento, validação e acurácia da rede podem ser vistos na Tabela 25. Após a décima época, conforme podemos observar na tabela, a rede conseguiu alcançar uma acurácia de 0.812639 na classificação, demonstrando a eficácia do método de aumento da base de dados e da transferência de dados que foram aplicados.

Com o objetivo de avaliar as classes de maneira separada, geramos a matriz confusão dos testes realizados nesse experimento. Da Tabela 26 podemos destacar que foi obtido o melhor resultado do reconhecimento da classe Felicidade em nosso trabalho. Como vimos anteriormente, essa é a classe com o menor número de amostras e o aumento do número de amostras se mostrou uma técnica eficaz para o aumento da capacidade da rede em reconhecer as amostras dessa classe. Esse resultado contribuiu de forma impor-

Tabela 25: - Resultado da classificação com a taxa de aprendizado igual a 10^{-7} no quarto experimento

Época	Custo de treino	Custo de validação	Acurácia
1	0.878089	0.836610	0.669543
2	0.891139	1.045950	0.588530
3	0.905717	1.353459	0.586860
4	0.809399	0.880154	0.692929
5	0.703324	0.818643	0.704343
6	0.554468	0.702610	0.739699
7	0.382100	0.650087	0.771715
8	0.210504	0.705921	0.791481
9	0.107573	0.711677	0.811804
10	0.075634	0.740088	0.812639

tante para o resultado geral da rede.

Tabela 26: - Matriz confusão do quarto experimento com a base IEMOCAP e a CNN

		Classe Real			
		Felicidade	Neutra	Raiva	Tristeza
Classe Prevista	Felicidade	63,00%	4,57%	1,65%	2,53%
	Neutra	23,89%	82,07%	10,35%	9,32%
	Raiva	5,71%	5,79%	84,82%	1,73%
	Tristeza	7,40%	7,57%	3,18%	86,42%
Geral		**81,26%**			

CONCLUSÃO

Neste trabalho foram investigadas diversas técnicas de aprendizagem de máquina aplicadas ao problema de reconhecimento de emoções a partir da fala. Em particular, comparamos técnicas clássicas baseadas em GMM e redes neurais probabilísticas com abordagens mais recentes tais como redes convolucionais profundas (ResNet), explorando a transferência de aprendizagem e o aumento artificial dos dados. Restringimos os procedimentos de treinamento e teste às classes Felicidade, Neutra, Raiva e Tristeza contidas nas bases de dados IEMOCAP e EmoDb para treino e teste dos sistemas propostos.

Observando os resultados do classificador GMM com ambas as bases de dados, podemos perceber que a extração do silêncio não melhorou o resultado no GMM para a base IEMOCAP, mas apresentou uma melhora para a base EmoDb. Isso nos leva a concluir que para a base EmoDb os períodos de silêncio não carregam o conteúdo emocional contidos nas sentenças ditas. Já na base IEMOCAP esses períodos de silêncio foram importantes para a classificação das emoções pelo GMM. Além disso, comparando os resultados desse classificador utilizando os dados das bases EmoDb e IEMOCAP, verificamos também que esse classificador obteve um resultado melhor quando treinado e testado com um conjunto de dados menor, a base de dados EmoDb.

Nos experimentos realizados com a Rede Neural Probabilística, observamos que assim como aconteceu nos experimentos com a GMM, essa rede classificou grande parte das amostras de teste da classe Felicidade como Raiva. Entretanto, quando comparado com o GMM, a rede obteve uma acurácia menor para a classe Felicidade em todos os experimentos propostos. Além disso, observamos nesses testes que a técnica de extração do silêncio não se mostrou eficiente para o resultado da classificação com a PNN.

Ao final dos experimentos, verificamos que a nossa ideia de que o uso da técnica de transferência de aprendizado aplicada em uma rede convolucional resultaria em uma melhora no resultado da classificação das emoções contidas na fala se mostrou correta. No primeiro experimento com essa técnica (onde utilizamos os dados da base IEMOCAP original) obtivemos uma acurácia de 65,31%, contra 59,86% da rede onde não foi utilizada a técnica e foi alimentada pela mesma base. Além disso, nos experimentos com a base aumentada em quatro vezes, a rede pré treinada obteve uma acurácia de 81,26% - o melhor resultado para os experimentos com a rede convolucional - contra 73,58% atingido

pelo experimento onde não foi utilizada a técnica de transferência de aprendizado. Esses resultados nos mostram que a técnica de aumento da base também contribui para um aumento do desempenho da rede. Um caso que deixa isso bem claro é o da classe Felicidade, que é classe da base de dados IEMOCAP que possui o menor número de amostras, e que apresentou o maior ganho de acurácia com essa técnica.

Ainda com relação a redes neurais profundas aplicada ao problema de reconhecimento de emoções em fala, podemos concluir que:

- Uma base de dados relativamente pequena não é empecilho para uso de redes neurais profundas, desde que se explore uma rede pré-treinada;
- Ainda que a rede tenha sido pré-treinada com imagens, em vez de sinais de fala, a transferência de aprendizagem ainda é efetiva;
- O uso do espectrograma, consagrado em aplicações diversas, incluindo o reconhecimento de fala, é também útil para o reconhecimento de emoções.

Possíveis trabalhos futuros são listados abaixo:

- Testar o uso de outras bases de dados com a rede convolucional treinada neste trabalho;
- Estudar o resultado da utilização de uma rede com um maior número de camadas convolucionais;
- Testar a utilização de outras técnicas para o aumento da base de dados;
- Utilizar uma rede pré-treinada em sinais de fala para realizar a transferência de aprendizagem;
- Combinar outras partes da base IEMOCAP, relativo à video e texto para uma classificação mais robusta explorando multi-modalidade.

REFERÊNCIAS

[1] BUSSO, C. et al. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, Springer, v. 42, n. 4, p. 335, 2008.

[2] AYADI, M. E.; KAMEL, M. S.; KARRAY, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, Elsevier, v. 44, n. 3, p. 572–587, 2011.

[3] BANSE, R.; SCHERER, K. R. Acoustic profiles in vocal emotion expression. *Journal of personality and social psychology*, US: American Psychological Association, v. 70, n. 3, p. 614, 1996.

[4] KLEINGINNA, P. R.; KLEINGINNA, A. M. A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and emotion*, Springer, v. 5, n. 4, p. 345–379, 1981.

[5] SCHULLER, B. W. Speech emotion recognition: two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, ACM, v. 61, n. 5, p. 90–99, 2018.

[6] EKMAN, P.; SORENSON, E. R.; FRIESEN, W. V. Pan-cultural elements in facial displays of emotion. *Science*, American Association for the Advancement of Science, v. 164, n. 3875, p. 86–88, 1969.

[7] GUNES, H.; SCHULLER, B. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image and Vision Computing*, Elsevier, v. 31, n. 2, p. 120–136, 2013.

[8] BRADLEY, M. M.; LANG, P. J. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, Elsevier, v. 25, n. 1, p. 49–59, 1994.

[9] WILLIAMS, C. E.; STEVENS, K. N. Vocal correlates of emotional states. *Speech evaluation in psychiatry*, Grune & Stratton New York, p. 221–240, 1981.

[10] CAHN, J. E. The generation of a ect in synthesized speech. *Journal of the American Voice I/O Society*, Citeseer, v. 8, p. 1–19, 1990.

[11] LISCOMBE, J. J. *Prosody and speaker state: paralinguistics, pragmatics, and proficiency*. [S.l.]: Citeseer, 2007.

[12] DEVILLERS, L.; VIDRASCU, L.; LAMEL, L. Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks*, Elsevier, v. 18, n. 4, p. 407–422, 2005.

[13] NWE, T. L.; FOO, S. W.; SILVA, L. C. D. Speech emotion recognition using hidden markov models. *Speech communication*, Elsevier, v. 41, n. 4, p. 603–623, 2003.

[14] BURKHARDT, F. et al. A database of german emotional speech. In: *Ninth European Conference on Speech Communication and Technology*. [S.l.: s.n.], 2005.

[15] SLANEY, M.; MCROBERTS, G. Baby ears: a recognition system for affective vocalizations. In: IEEE. *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. [S.l.], 1998. v. 2, p. 985–988.

[16] HANSEN, J. H.; BOU-GHAZALE, S. E. Getting started with susas: A speech under simulated and actual stress database. In: *Fifth European Conference on Speech Communication and Technology*. [S.l.: s.n.], 1997.

[17] RINGEVAL, F. et al. Introducing the recola multimodal corpus of remote collaborative and affective interactions. In: IEEE. *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. [S.l.], 2013. p. 1–8.

[18] LEE, C. M. et al. Emotion recognition based on phoneme classes. In: *Eighth International Conference on Spoken Language Processing*. [S.l.: s.n.], 2004.

[19] KWON, O.-W. et al. Emotion recognition by speech signals. In: *Eighth European Conference on Speech Communication and Technology*. [S.l.: s.n.], 2003.

[20] BREAZEAL, C.; ARYANANDA, L. Recognition of affective communicative intent in robot-directed speech. *Autonomous robots*, Springer, v. 12, n. 1, p. 83–104, 2002.

[21] SCHULLER, B. Towards intuitive speech interaction by the integration of emotional aspects. In: IEEE. *Systems, Man and Cybernetics, 2002 IEEE International Conference on*. [S.l.], 2002. v. 6, p. 6–pp.

[22] PIERRE-YVES, O. The production and recognition of emotions in speech: features and algorithms. *International Journal of Human-Computer Studies*, Elsevier, v. 59, n. 1-2, p. 157–183, 2003.

[23] NICHOLSON, J.; TAKAHASHI, K.; NAKATSU, R. Emotion recognition in speech using neural networks. *Neural computing & applications*, Springer, v. 9, n. 4, p. 290–296, 2000.

[24] PETRUSHIN, V. A. Emotion recognition in speech signal: experimental study, development, and application. In: *Sixth International Conference on Spoken Language Processing*. [S.l.: s.n.], 2000.

[25] HENDY, N. A.; FARAG, H. Emotion recognition using neural network: A comparative study. In: WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY (WASET). *Proceedings of World Academy of Science, Engineering and Technology*. [S.l.], 2013. p. 791.

[26] SHAMI, M. T.; KAMEL, M. S. Segment-based approach to the recognition of emotions in speech. In: IEEE. *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. [S.l.], 2005. p. 4–pp.

[27] ZHANG, S. et al. Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching. *IEEE Transactions on Multimedia*, IEEE, v. 20, n. 6, p. 1576–1590, 2018.

[28] TZIRAKIS, P.; ZHANG, J.; SCHULLER, B. W. End-to-end speech emotion recognition using deep neural networks. In: IEEE. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2018. p. 5089–5093.

[29] REYNOLDS, D. A.; QUATIERI, T. F.; DUNN, R. B. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, Academic Press, v. 10, n. 1-3, p. 19–41, 2000.

[30] SCHULLER, B.; RIGOLL, G.; LANG, M. Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In: IEEE. *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. [S.l.], 2004. v. 1, p. I–577.

[31] FRANCE, D. J. et al. Acoustical properties of speech as indicators of depression and suicidal risk. *IEEE transactions on Biomedical Engineering*, IEEE, v. 47, n. 7, p. 829–837, 2000.

[32] JIN, H.; YANG, L. T.; TSAI, J. J.-P. *Ubiquitous Intelligence and Computing: Third International Conference, UIC 2006, Wuhan, China, September 3-6, 2006, Proceedings*. [S.l.]: Springer, 2006.

[33] LANJEWAR, R. B.; MATHURKAR, S.; PATEL, N. Implementation and comparison of speech emotion recognition system using gaussian mixture model (gmm) and k-nearest neighbor (k-nn) techniques. *Procedia Computer Science*, Elsevier, v. 49, p. 50–57, 2015.

[34] SATT, A.; ROZENBERG, S.; HOORY, R. Efficient emotion recognition from speech using deep learning on spectrograms. *Proc. Interspeech 2017*, p. 1089–1093, 2017.

[35] DELLER, J. R.; HANSEN, J. H.; PROAKIS, J. G. Discrete-time processing of speech signals. IEEE New York, NY, USA:, 2000.

[36] PROAKIS, J.; DELLER, J.; HANSEN, J. Discrete-time processing of speech signals. *New York, Macrnillan Pub. Co*, 1993.

[37] MCLOUGHLIN, I. *Applied speech and audio processing: with Matlab examples*. [S.l.]: Cambridge University Press, 2009.

[38] GARGOURI, D.; KAMMOUN, M. A.; HAMIDA, A. B. A comparative study of formant frequencies estimation techniques. In: *Proceedings of the 5th WSEAS. International Conference on Signal Processing*. [S.l.: s.n.], 2006. p. 15–19.

[39] RABINER, L. R.; SCHAFER, R. W. et al. Introduction to digital speech processing. *Foundations and Trends® in Signal Processing*, Now Publishers, Inc., v. 1, n. 1–2, p. 1–194, 2007.

[40] ROSENBERG, A. E. Effect of glottal pulse shape on the quality of natural vowels. *The Journal of the Acoustical Society of America*, ASA, v. 49, n. 2B, p. 583–590, 1971.

[41] DAVIS, S.; MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, IEEE, v. 28, n. 4, p. 357–366, 1980.

[42] HUANG, X. et al. *Spoken language processing: A guide to theory, algorithm, and system development*. [S.l.]: Prentice hall PTR Upper Saddle River, 2001.

[43] HIRSCH, H.-G.; PEARCE, D. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In: *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*. [S.l.: s.n.], 2000.

[44] KLASMEYER, G. *Akustische Korrelate des stimmlich emotionalen Ausdrucks in der Lautsprache*. [S.l.]: Th. Hector, 1999.

[45] SENDLMEIER, W. ''phonetische reduktion und elaboration bei emotionaler sprechweise". *Von Sprechkunst und Normphonetik*, p. 169–177, 1997.

[46] VLASENKO, B. et al. Frame vs. turn-level: emotion recognition from speech considering static and dynamic processing. In: SPRINGER. *International Conference on Affective Computing and Intelligent Interaction*. [S.l.], 2007. p. 139–147.

[47] EYBEN, F.; SCHULLER, B.; RIGOLL, G. Improving generalisation and robustness of acoustic affect recognition. In: ACM. *Proceedings of the 14th ACM international conference on Multimodal interaction*. [S.l.], 2012. p. 517–522.

[48] ESKIMEZ, S. E. et al. Emotion classification: how does an automated system compare to naive human coders? In: IEEE. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2016. p. 2274–2278.

[49] SCHERER, K. R.; WALLBOTT, H. G.; SUMMERFIELD, A. B. *Experiencing emotion: A cross-cultural study*. [S.l.]: Editions de la Maison des Sciences de l'Homme, 1986.

[50] FIGUEIREDO, M. A. T.; JAIN, A. K. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, v. 24, n. 3, p. 381–396, 2002.

[51] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

[52] ANTONIOU, A.; LU, W.-S. *Practical Optimization: Algorithms and Engineering Applications*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2007. ISBN 0387711066, 9780387711065.

[53] DELLAERT, F. *The expectation maximization algorithm*. [S.l.], 2002.

[54] SOLTANE, M. Figueiredo-jain (fj) tune algorithm for gaussian mixture modal (gmm) based face and signature multi-modal biometric verification fusion systems. *Journal of Computational Intelligence and Electronic Systems*, American Scientific Publishers, v. 4, n. 1, p. 27–36, 2015.

[55] NANDAKUMAR, K. et al. Likelihood ratio-based biometric score fusion. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 30, n. 2, p. 342–347, 2008.

[56] PAALANEN, P. Bayesian classification using gaussian mixture model and em estimation: Implementations and comparisons. *Information Technology Project*, 2004.

[57] PALO, H. K.; MOHANTY, M. N. Comparative analysis of neural networks for speech emotion recognition. *International Journal of Engineering & Technology*, v. 7, n. 4.39, p. 112–116, 2018.

[58] MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.

[59] HAYKIN, S. S. et al. *Neural networks and learning machines/Simon Haykin.* [S.l.]: New York: Prentice Hall,, 2009.

[60] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

[61] SPECHT, D. F. Probabilistic neural networks. *Neural networks*, Elsevier, v. 3, n. 1, p. 109–118, 1990.

[62] ZEINALI, Y.; STORY, B. A. Competitive probabilistic neural network. *Integrated Computer-Aided Engineering*, IOS press, v. 24, n. 2, p. 105–118, 2017.

[63] HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

[64] TAN, C. et al. A survey on deep transfer learning. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2018. p. 270–279.

[65] RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, Springer, v. 115, n. 3, p. 211–252, 2015.

[66] OPPENHEIM, A. V. *Discrete-time signal processing*. [S.l.]: Pearson Education India, 1999.

[67] DRIEDGER, J.; MÜLLER, M. A review of time-scale modification of music signals. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 6, n. 2, p. 57, 2016.

[68] FLANAGAN, J. L.; GOLDEN, R. Phase vocoder. *Bell System Technical Journal*, Wiley Online Library, v. 45, n. 9, p. 1493–1509, 1966.

[69] PORTNOFF, M. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 24, n. 3, p. 243–248, 1976.

[70] MÜLLER, M. *Fundamentals of music processing: Audio, analysis, algorithms, applications*. [S.l.]: Springer, 2015.

[71] GRIFFIN, D.; LIM, J. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 32, n. 2, p. 236–243, 1984.

[72] HAGHPARAST, A.; PENTTINEN, H.; VÄLIMÄKI, V. Real-time pitchshifting of musical signals by a time-varying factor using normalized filtered correlation time-scale

modification (nfc-tsm). In: CITESEER. *Proceedings of the International Conference on Digital Audio Effects (DAFx), Bordeaux, France*. [S.l.], 2007. p. 10–15.

[73] SCHÖRKHUBER, C.; KLAPURI, A.; SONTACCHI, A. Audio pitch shifting using the constant-q transform. *Journal of the Audio Engineering Society*, Audio Engineering Society, v. 61, n. 7/8, p. 562–572, 2013.

APÊNDICE A - ESPECTROGRAMA

A Transformada de Fourier de tempo curto (do Inglês, *Short-Time Fourier Transform*, STFT) é a base para uma ampla gama de sistemas de análise, codificação e síntese da fala [39], e é definida como

$$X_{\hat{n}}(e^{j\hat{\omega}}) = \sum_{m=-\infty}^{\infty} x[m]\omega[\hat{n} - m]e^{-j\hat{\omega}m} \tag{27}$$

Por definição, para o tempo de análise fixo $\hat{n}$, a STFT é a transformada de Fourier de tempo discreto (DTFT) do sinal $x_{\hat{n}}[m] = x[m]\omega[\hat{n} - m]$, isto é, a DTFT do sinal selecionado e ponderado em amplitude pela janela deslizante $\omega[\hat{n} - m]$. Portanto, o STFT é uma função de duas variáveis: $\hat{n}$ - o índice de tempo discreto denotando a posição da janela - e $\hat{\omega}$ representando a frequência de análise. Como (27) é uma sequência de DTFTs, a função bidimensional $X_{\hat{n}}(e^{j\hat{\omega}})$ no tempo discreto $\hat{n}$ é uma função periódica da frequência radial contínua $\hat{\omega}$ com o período igual a 2π.

Além disso, a STFT pode ser expressa em termos de uma operação de filtragem linear. Por exemplo, a equação (27) pode ser expressa como a convolução discreta

$$X_{\hat{n}}(e^{j\hat{\omega}}) = (x[n]e^{-j\hat{\omega}n}) * \omega[n]|_{n=\hat{n}} \tag{28}$$

ou

$$X_{\hat{n}}(e^{j\hat{\omega}}) = (x[n] * (\omega[n]e^{j\hat{\omega}n})|_{n=\hat{n}} \tag{29}$$

Uma janela típica, como uma janela de Hamming, quando vista como uma resposta ao impulso de um filtro linear, tem uma resposta em baixa frequência com a frequência de corte variando inversamente com o comprimento da janela [39]. Isso significa que, para um valor fixo de $\hat{\omega}$, $X_{\hat{n}}(e^{j\hat{\omega}})$ varia lentamente em função de $\hat{n}$ [66].

Como definido em (27), a STFT é uma função de uma frequência de análise contínua $\hat{\omega}$. A STFT se torna uma ferramenta prática tanto para análise quanto para aplicações quando implementada com uma janela de duração finita movida em passos de $R > 1$ amostras no tempo e calculada em um conjunto discreto de frequências como em [39]

$$X_{rR}[k] = \sum_{m=rR-L+1}^{rR} x[m]\omega[rR-m]e^{-j(2\pi k/N)m} \quad k = 0, 1, ..., N-1 \tag{30}$$

onde N é o número de frequências uniformemente espaçadas no intervalo $0 \leq \hat{\omega} < 2\pi$, e L é o comprimento da janela, em amostras. Observe que assumimos que $\omega[m]$ é causal e diferente de zero apenas no intervalo $0 \leq m \leq L-1$ de modo que o segmento janelado $x[m]\omega[rR-m]$ seja diferente de zero sobre $rR - L + 1 \leq m \leq rR$. Para ajudar na interpretação, é útil escrever (30) na forma equivalente:

$$X_{rR}[k] = \tilde{X}_{rR}[k]e^{-j(2\pi k/N)rR} \quad k = 0, 1, ..., N-1 \tag{31}$$

onde

$$\tilde{X}_{rR}[k] = \sum_{m=0}^{L-1} x[rR-m]w[m]e^{j(2\pi k/N)m} \quad k = 0, 1, ..., N-1 \tag{32}$$

Como assumimos, por especificidade, que $\omega[m] \neq 0$ apenas no intervalo $0 \leq m \leq L-1$, a forma alternativa, $\tilde{X}_{rR}[k]$, tem a interpretação da transformada discreta de Fourier de N pontos da sequência $x[rR-m]\omega[m]$, que, devido à definição da janela, é diferente de zero no intervalo $0 \leq m \leq L-1$. Em (32), o tempo de análise rR é deslocado para o tempo de origem do cálculo da DFT, e o segmento do sinal de fala é a sequência reversa no tempo das L amostras que precedem o tempo de análise. A exponencial complexa $e^{-j(2\pi k/N)rR}$ em (31) resulta do deslocamento para o tempo de origem.

Para obter a STFT discreta, calcula-se $\tilde{X}_{rR}[k]$ da seguinte forma [39]:

1. Obtém-se a sequência $x_{rR}[m] = x[rR-m]\omega[m]$, para $m = 0, 1, ..., L-1$.

2. Calcula-se o conjugado complexo da DFT de N pontos da sequência $x_{rR}[m]$. (Isso pode ser feito eficientemente com um algoritmo FFT) [39]

3. A multiplicação por $e^{-j(2\pi k/N)rR}$ pode ser feita se necessário, mas frequentemente pode ser omitida

4. Move-se a posição da janela de R amostras (ou seja, $r \rightarrow r+1$) e retorna ao passo 1.

A STFT discreta é especificada em função do período de amostragem temporal, R, e do número de frequências uniformemente espaçadas, N. Pode-se facilmente mostrar que R e N são determinados inteiramente pela largura de tempo e largura de banda de frequência da janela passa-baixa, $\omega[m]$, usada para calcular a STFT, dando as seguintes restrições em R e N [39]:

1. $R \leq L/(2C)$ onde C é uma constante que é dependente da largura de banda da janela; $C = 2$ para uma janela de Hamming e $C = 1$ para uma janela retangular.

2. $N \leq L$, onde L é o comprimento da janela em amostras.

A restrição (1) acima está relacionada à amostragem do STFT no tempo a uma taxa de duas vezes a largura de banda da janela, a fim de eliminar o efeito de *aliasing* na frequência, e a restrição (2) está relacionada à amostragem em frequência a uma taxa de duas vezes a duração da janela para garantir que não haja *aliasing* no domínio do tempo.

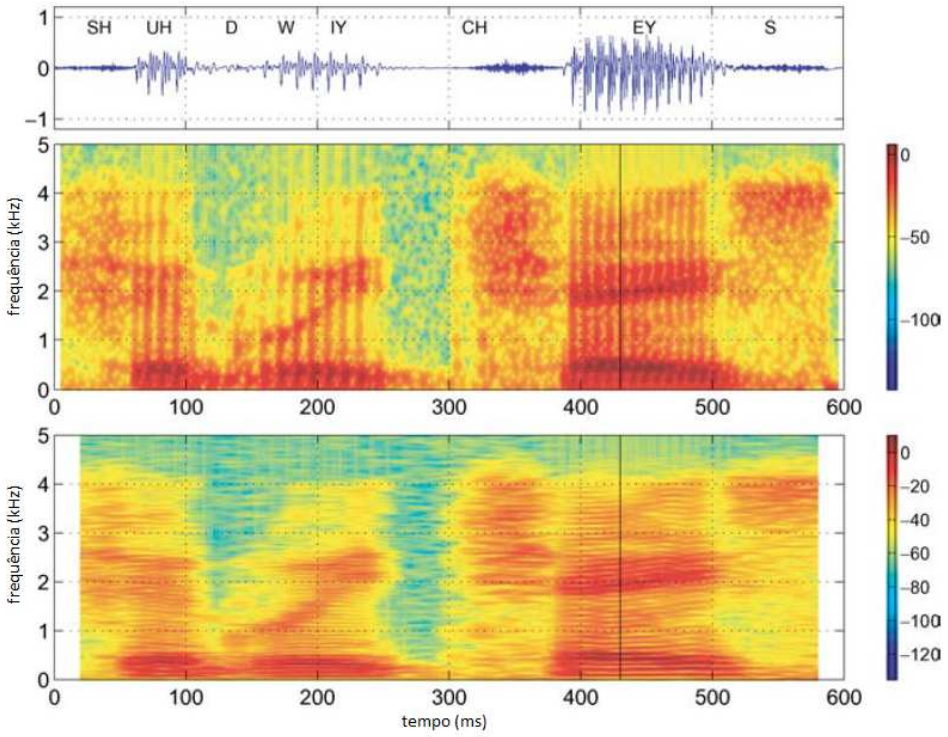
O espectrograma do sinal sonoro é uma ferramenta básica para entender como os sons da fala são produzidos e como a informação fonética está codificada no sinal da fala [39]. Ele permite a visualização gráfica do módulo da STFT de sinais de fala. A Figura 32 apresenta um sinal de fala e o seus espectrogramas correspondentes a equação

$$S(t_r, f_k) = 20log_{10}|\tilde{X}_{rR}[k]| = 20log_{10}|X_{rR}[k]| \tag{33}$$

onde os eixos do gráfico são rotulados em termos de tempo e frequência por meio das relações $t_r = rRT$ e $f_k = k/(NT)$, onde T é o período de amostragem do sinal em tempo discreto $x[n] = x_a(nT)$. Para produzir as imagens de aparência suave como as da Figura 32, o valor de R é geralmente bem pequeno comparado ao comprimento de janela L e a quantidade de amostras de frequência, N, que pode ser muito maior que o comprimento de janela L. A função de duas variáveis pode ser plotada em uma superfície bidimensional como uma imagem em escala de cinza ou em cores. A Figura 32 mostra a forma de onda temporal no topo e dois espectrogramas com diferentes janelas de análise. As barras à direita calibram o mapa de cores em decibéis. Uma interpretação cuidadosa da equação (32) e as imagens de espectrograma correspondentes levam a informações valiosas sobre a natureza do sinal de fala. Primeiro observe que a sequência da janela $w[m]$ é diferente de zero apenas no intervalo $0 \leq m \leq L - 1$. O comprimento da janela tem um efeito

importante na imagem do espectrograma. O espectrograma superior na Figura 32 foi calculado com um comprimento de janela de $L = 101$ amostras, correspondendo a uma duração de 10 ms. Esse comprimento de janela é da ordem do comprimento de um período de *pitch* da forma de onda durante intervalos vozeados [39]. Como resultado, em intervalos sonoros, o espectrograma exibe estrias orientadas verticalmente. Quando temos uma janela de análise curta cada período de *pitch* individual é bem representada na dimensão de tempo, mas a resolução de frequência é baixa. Por esta razão, se a janela de análise é curta, o espectrograma é chamado de banda larga. Isto é consistente com a interpretação de filtragem linear do STFT, já que um filtro de análise curto tem uma faixa larga. Por outro lado, quando o comprimento da janela é longo, o espectrograma é de banda estreita, que é caracterizado por boa resolução de frequência e baixa resolução de tempo.

Figura 32: - Um sinal de fala e seus respectivos espectrogramas



Fonte: [39]

APÊNDICE B - MODIFICAÇÃO DE PITCH E DE ESCALA DE TEMPO

Os procedimentos de modificação de escala do tempo são métodos de processamento digital de sinais para reduzir a duração de um determinado sinal de áudio [67]. Idealmente, o sinal modificado em escala de tempo deve soar como se o conteúdo do sinal original fosse realizado em um ritmo diferente, preservando propriedades como o *pitch* e o timbre. Para atingir esse objetivo, muitos procedimentos de modificação de escala do tempo seguem uma estratégia comum cuja ideia principal é decompor o sinal de entrada em quadros curtos. Tendo um comprimento fixo, geralmente na faixa de 50 a 100 milissegundos de material de áudio, cada quadro captura o conteúdo de *pitch* local do sinal. Os quadros são então realocados no eixo de tempo para obter a modificação da escala de tempo real, enquanto, ao mesmo tempo, preserva o *pitch* do sinal.

O primeiro passo do procedimento consiste em dividir o sinal x em pequenos quadros de análise x_m, cada um deles tendo um comprimento de N amostras. Os quadros são espaçados por um salto de análise H_a:

$$x_m(r) = \begin{cases} x(r + mH_a), & \text{se } r \in [-\frac{N}{2}; \frac{N}{2} - 1] \\ 0, & \text{caso contrário.} \end{cases} \tag{34}$$

Em uma segunda etapa, esses quadros são realocados no eixo de tempo em relação a um salto de síntese específico H_s. Essa realocação é responsável pela modificação real da escala de tempo do sinal de entrada por um fator de esticamento $\alpha = H_s/H_a$. Uma vez que frequentemente é desejável ter uma sobreposição específica dos quadros realocados, o salto de síntese H_s é frequentemente fixo (escolhas comuns são $H_s = N/2$ ou $H_s = N/4$) [67] enquanto que o salto de análise é dado por $H_a = H_s/\alpha$. No entanto, a simples sobreposição dos quadros recolocados sobrepostos levaria a artefatos indesejados, como descontinuidades de fase nos limites do quadro e flutuações de amplitude. Portanto, antes da reconstrução do sinal, os quadros de análise são adequadamente adaptados para formar quadros de síntese y_m. Na etapa final, os quadros de síntese são sobrepostos para reconstruir o sinal de saída modificado na escala de tempo $y : (Z) \to \mathbb{R}$ do procedimento de modificação de escala do tempo:

$$y(r) = \sum_{m \in (Z)} y_m(r - mH_s). \tag{35}$$

Embora essa estratégia fundamental pareça direta à primeira vista, há muitas armadilhas e escolhas de design que podem influenciar fortemente a qualidade perceptual do sinal de saída modificado na escala de tempo. A questão mais óbvia é como adaptar os quadros de análise x_m para formar os quadros de síntese y_m. Há muitas maneiras de abordar essa tarefa, levando a procedimentos de modificação de escala do tempo conceitualmente diferentes. A seguir, introduziremos uma estratégia de modificação de escala do tempo que trabalha no domínio da frequência e com uma técnica conhecida como *phase vocoder* [68] [69].

A principal ideia dos procedimentos de modificação de escala do tempo que trabalham no domínio da frequência é preservar a periodicidade de todos os componentes do sinal [67]. Esses procedimentos interpretam cada quadro de análise como uma soma ponderada de componentes senoidais com frequência e fase conhecidas. Com base nesses parâmetros, cada um desses componentes é manipulado individualmente para evitar artefatos de salto de fase em todas as frequências do sinal reconstruído. Uma ferramenta fundamental para a análise de frequência do sinal de entrada é a transformada de Fourier de tempo curto. No entanto, dependendo dos parâmetros de discretização escolhidos, as estimativas de frequência resultantes podem ser imprecisas. Para este fim, a técnica conhecida como *phase vocoder* é usada para melhorar as estimativas de frequências grosseiras da transformada de Fourier de tempo curto, derivando frequências instantâneas de componentes senoidais. Nos procedimentos de modificação de escala do tempo baseados no *phase vocoder*, essas estimativas aprimoradas são usadas para atualizar as fases dos componentes senoidais de um sinal de entrada em um processo conhecido como propagação de fase.

A ferramenta mais importante da modificação da escala do tempo utilizando o *phase vocoder* é a STFT, que aplica a Transformada de Fourier para cada quadro de análise de um dado sinal de entrada. A STFT X de um sinal x é dado por

$$X(m,k) = \sum_{r=-\frac{N}{2}}^{\frac{N}{2}-1} x_m(r)\omega(r)\exp(-2\pi ikr/N) \tag{36}$$

onde $m \in \mathbb{Z}$ é o índice dos quadros, $k \in [0 : N-1]$ é o índice das frequências, N é o comprimento do quadro, x_m é o m-ésimo quadro de x e ω é uma função janela. Dada a frequência de amostragem F_s do sinal, o índice m dos quadros de $X(m,k)$ é associado ao tempo físico

$$T_{coef}(m) = \frac{mH_a}{F_s} \tag{37}$$

dado em segundos, e o índice k relativo as frequências corresponde a frequência física

$$F_{coef}(k) = \frac{kF_s}{N} \tag{38}$$

dada em Hertz. O número complexo $X(m,k)$ denota o k-ésimo coeficiente de Fourier para o m-ésimo quadro de análise. Ele pode ser representado por uma magnitude $|X(m,k)| \in \mathbb{R}^+$ e uma fase $\varphi(m,k) \in [0,1)$ como

$$X(m,k) = |X(m,k)| \exp(2\pi\varphi(m,k)) \tag{39}$$

No contexto da modificação da escala do tempo utilizando o *phase vocoder*, é necessário reconstruir o sinal de saída y de uma STFT de um sinal modificado X^{Mod}. Note que um STFT modificado é em geral não inversível [70]. Em outras palavras, pode não existir um sinal y que possua um X^{Mod} como sua STFT. No entanto, existem métodos que visam reconstruir um sinal y de X^{Mod} cujo STFT está próximo de X^{Mod} com relação a alguma medida de distância. Seguindo o procedimento descrito em [71], primeiro calculamos os quadros x_m^{Mod} no domínio do tempo usando a transformada inversa de Fourier.

$$x_m^{Mod}(r) = \frac{1}{N} \sum_{k=0}^{N-1} X^{Mod}(m,k) \exp(2\pi ikr/N). \tag{40}$$

A partir desses quadros, derivamos quadros de síntese

$$y_m(r) = \frac{\omega(r) x_m^{Mod}(r)}{\sum_{n\in\mathbb{Z}} \omega(r - nH_s)^2} \tag{41}$$

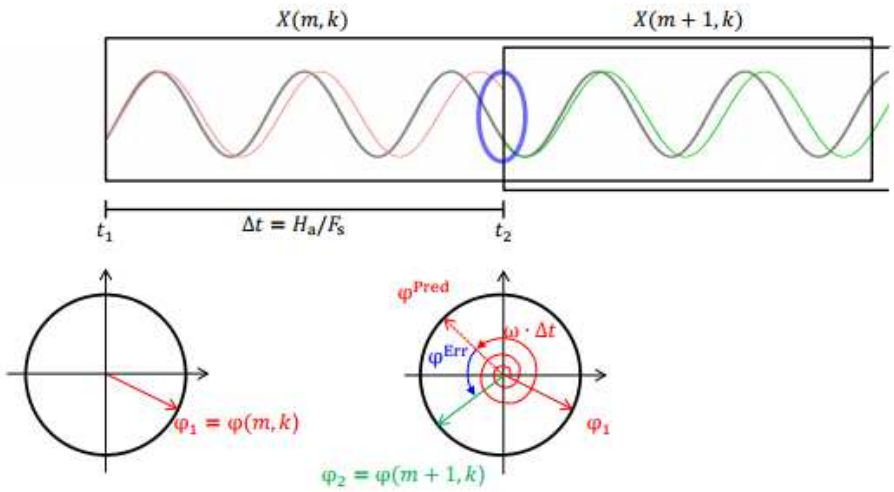
e reconstruímos o sinal de saída y com a equação (35).

Phase Vocoder

Cada $X(m,k)$ de um STFT pode ser interpretado como um componente senoidal

com amplitude $|X(m,k)|$ e fase $\varphi(m,k)$ que contribui para o m-ésimo quadro de análise do sinal de entrada x. No entanto, a transformada de Fourier produz coeficientes apenas para um conjunto discreto de frequências que são amostradas linearmente no eixo da frequência, veja a equação (38). A resolução de frequência do STFT, portanto, não é suficiente para atribuir um valor de frequência preciso a esse componente senoidal. O vocoder de fase é uma técnica que refina a estimativa de frequência grosseira do STFT explorando as informações de fase fornecidas.

Figura 33: - Princípio do *Phase Vocoder*



Fonte: [67]

Para entender o *phase vocoder*, vamos dar uma olhada no cenário mostrado na Figura 33. Suponha que recebamos duas estimativas de fase $\varphi_1 = \varphi(m,k)$ e $\varphi_2 = \varphi(m+1,k)$ nas instâncias de tempo $t_1 = T_{coef}(m)$ e $t_2 = T_{coef}(m+1)$ de um componente senoidal para o qual temos apenas uma estimativa de frequência grosseira $\omega = F_{coef}(k)$. Nosso objetivo é estimar a frequência instantânea "real" da senoide $IF(\omega)$. A Figura 33 mostra este componente senoidal (cinza), bem como duas senoides que têm uma frequência de ω (vermelho e verde). Além disso, também vemos representações de fase nas instâncias de tempo t_1 e t_2. A senoide vermelha tem uma fase de φ_1 em t_1 e a senoide verde uma fase de φ_2 em t_2. Pode-se ver que a frequência ω das senoides vermelha e verde é ligeiramente menor que a frequência da senoide cinza. Intuitivamente, enquanto as fases das senoides cinza e vermelho coincidem em t_1, elas divergem ao longo do tempo, e podemos observar uma considerável discrepância após $\Delta t = t_2 - t_1$ segundos (oval azul). Uma vez que

conhecemos a frequência da senoide vermelho, podemos calcular o seu avanço de fase não desembrulhado, ou seja, o número de oscilações que ocorrem ao longo de Δt segundos:

$$\omega \Delta t \tag{42}$$

Sabendo que sua fase em t_1 é φ_1, podemos prever sua fase após Δt segundos:

$$\varphi^{Pred} = \varphi_1 + \omega \Delta t \tag{43}$$

Em t_2, novamente temos uma estimativa precisa da fase φ_2 para a senoide cinza. Podemos, portanto, calcular o erro de fase φ^{Err} entre a fase realmente medida em t_2 e a fase prevista ao assumir uma frequência de ω:

$$\varphi^{Err} = \Psi(\varphi_2 - \varphi^{Pred}) \tag{44}$$

onde Ψ é a principal função de argumento que mapeia uma determinada fase no intervalo $[-0,5, 0,5]$. Observe que mapeando φ^{Err} nesta faixa, assumimos que o número de oscilações das senoides cinza e vermelho diferem em no máximo meio período. No contexto da estimação de frequência instantânea, isso significa que a estimativa de frequência grosseira ω precisa estar próxima da frequência real da senoide, e que o intervalo Δt deve ser pequeno.O avanço de fase desembrulhado da senoide cinza pode então ser computado pela soma do avanço de fase desembrulhado da senoide vermelha com a frequência ω (seta espiral vermelha) e o erro de fase (seta curva azul):

$$\omega \Delta t + \varphi^{Err}. \tag{45}$$

Isso nos dá o número de oscilações da senoide cinza ao longo de Δt segundos. A partir disso, podemos derivar a frequência instantânea da senoide cinza por

$$IF(\omega) = \frac{\omega \Delta t + \varphi^{Err}}{\Delta t} = \omega + \frac{\varphi^{Err}}{\Delta t} \tag{46}$$

A frequência $\varphi^{Err}/\Delta t$ pode ser interpretada como o pequeno deslocamento da frequência real da senoide cinza a partir da estimativa de frequência aproximada ω.

Podemos usar essa abordagem para refinar a resolução de frequência grosseira do STFT calculando estimativas de frequência instantânea $F^{IF}_{coef}(m, k)$ para todos $X(m, k)$:

$$F_{coef}^{IF}(m,k) = IF(\omega) = \omega + \frac{\Psi(\varphi_2 - (\varphi_1 + \omega\Delta t))}{\Delta t} \tag{47}$$

com $\omega = F_{coef}(k)$, $\Delta t = H_a/F_s$, $\varphi_1 = \varphi(m,k)$ e $\varphi_2 = \varphi(m+1,k)$.

O princípio da modificação de escala do tempo usando *Phase Vocoder* pode ser visualizado na Figura 34. Dado um sinal de áudio de entrada x, o primeiro passo é calcular o STFT X. A Figura 34a mostra os dois espectros de frequência sucessivos do m-ésimo quadro de análise e seu posterior, denotados por $X(m)$ e $X(m+1)$, respectivamente. Nosso objetivo é calcular um STFT modificado X^{Mod} com fases ajustadas φ^{Mod} a partir das quais podemos reconstruir um sinal modificado na escala de tempo sem artefatos de salto de fase:

$$X^{Mod}(m,k) = |X(m,k)| \exp(2\pi i \varphi^{Mod}(m,k). \tag{48}$$

Calculamos as fases ajustadas φ^{Mod} em um processo iterativo conhecido como propagação de fase. Suponha que as fases do m-ésimo quadro já tenham sido modificadas (veja a fase da senoide vermelha na Figura 34b sendo diferente de sua fase na Figura 34a). Como indicado pela Figura 34b, a sobreposição do m-ésimo quadro e seu posterior no salto de síntese H_s pode levar a saltos de fase. Conhecendo as frequências instantâneas F_{coef}^{IF} derivadas pelo *phase vocoder*, podemos predizer as fases dos componentes senoidais no quadro m após um intervalo de tempo correspondente a amostras H_s. Para este fim, ajustamos $\varphi_1 = \varphi^{Mod}(m,k)$, $\omega = F_{coef}^{IF}(m,k)$, e $\Delta t = H_s/F_s$ na equação (43). Isso nos permite substituir as fases do quadro na posição $m+1$ com a fase prevista:

$$\varphi^{Mod}(m+1,k) = \varphi^{Mod}(m,k) + F_{coef}^{IF}(m,k)\frac{H_s}{F_s} \tag{49}$$

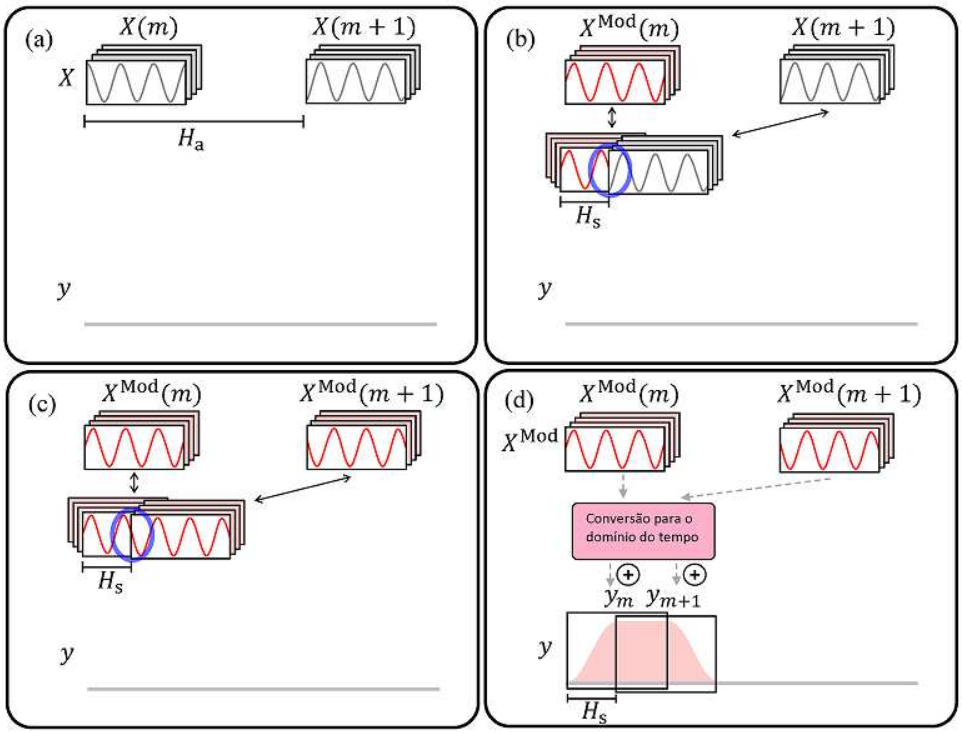
para $k \in [0 : N-1]$. Assumindo que as estimativas das frequências instantâneas F_{coef}^{IF} estão corretas, não há mais saltos de fase quando a sobreposição dos espectros modificados no salto de síntese H_s (Figura 34c). Na prática, iniciamos a propagação da fase iterativa com o índice de quadros $m = 0$ e definimos

$$\varphi^{Mod}(0,k) = \varphi(0,k), \tag{50}$$

para todo $k \in [0 : N-1]$. Finalmente, o sinal de saída y pode ser calculado usando o

procedimento de reconstrução de sinal utilizando o procedimento com a STFT descrito anteriormente (Figura 34d).

Figura 34: - Princípio da modificação de escala do tempo usando *Phase Vocoder*



Fonte: Adaptado de [67]

Mudança de Pitch

A mudança de *pitch* é a tarefa de alterar a afinação de uma gravação de áudio sem alterar seu comprimento [67]. Embora existam procedimentos especializados de mudança de *pitch* [72] [73], também é possível abordar o problema combinando a técnica de modificação da escala do tempo com a técnica de reamostragem. A principal observação é que reamostrar um determinado sinal e reproduzi-lo na taxa de amostragem original altera o comprimento e o *pitch* do sinal ao mesmo tempo. Para alterar o *pitch* de um determinado sinal, ele primeiro deve ser reamostrado e depois a escala de tempo dele deve ser modificada para compensar a mudança no comprimento. Mais precisamente, um sinal de áudio, amostrado a uma taxa de $F_s^{(in)}$, é primeiramente reamostrado e passa a ter uma nova taxa de amostragem $F_s^{(out)}$. Ao reproduzir o sinal em sua taxa de amostragem $F_s^{(in)}$ original, esta operação altera o comprimento do sinal por um fator de $\frac{F_s^{(out)}}{F_s^{(in)}}$ e dimensiona suas frequências pelo termo inverso. Para compensar a mudança de comprimento, o sinal

precisa ser esticado por um fator de $\alpha = \frac{F_s^{(in)}}{F_s^{(out)}}$, usando um procedimento de modificação da escala do tempo.