Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

Aplicação do modelo oculto de Markov e da rede neural convolucional em um sistema de reconhecimento de fala automático

Leonardo Castro Souza Marotta

João Monlevade, MG
2025

Leonardo Castro Souza Marotta

Aplicação do modelo oculto de Markov e da rede neural convolucional em um sistema de reconhecimento de fala automático

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.
Orientador: Prof. Orientador Glauco Ferreira Gazel Yared

Universidade Federal de Ouro Preto
João Monlevade
2025

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

M355a Marotta, Leonardo Castro Souza.
Aplicação do modelo oculto de Markov e da rede neural convolucional em um sistema de reconhecimento de fala automático. [manuscrito] / Leonardo Castro Souza Marotta. - 2025.
81 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Glauco Ferreira Gazel Yared.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia Elétrica .

1. Aprendizado do computador. 2. Markov, Processos de. 3. Processamento de sinais. 4. Redes Neurais Convolucionais. 5. Sistemas de processamento da fala. I. Yared, Glauco Ferreira Gazel. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.85

Bibliotecário(a) Responsável: Flavia Reis - CRB6/2431

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE ENGENHARIA ELETRICA

FOLHA DE APROVAÇÃO

Leonardo Castro Souza Marotta

Aplicação do modelo oculto de Markov e da Rede Neural Convolucional em um sistema de reconhecimento de fala automático

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica

Aprovada em 7 de outubro de 2025

Membros da banca

Doutor - Glauco Ferreira Gazel Yared - Orientador(a) - Universidade Federal de Ouro Preto
Doutora - Gilda Aparecida de Assis - Universidade Federal de Ouro Preto
Doutor - Marcelo Moreira Tiago - Universidade Federal de Ouro Preto

Glauco Ferreira Gazel Yared, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 13/11/2025

Documento assinado eletronicamente por **Glauco Ferreira Gazel Yared**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 13/11/2025, às 19:19, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1014872** e o código CRC **BECEB8FC**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0818 - www.ufop.br

À minha querida mãe.

Agradecimentos

À minha mãe, que não está mais presente entre nós, mas que sempre foi e será minha fonte de inspiração. Ao meu pai, que me ensina diariamente sobre superações, lutas e nunca deixar de lado a nossa essência.

À minha irmã, que é exemplo de persistência acadêmica, trabalho árduo e conquistas pessoais. Sua dedicação e desempenho profissional são fontes de inspiração que possuo sempre dentro de mim.

Aos meus familiares, pelo apoio constante nos momentos difíceis e transitórios da minha vida. Minha família é a base estrutural e educacional que tive, e serei eternamente grato por Deus me ter concedido uma família tão pura e unida.

À República São Jorge, pelos longos anos de vivência e aprendizado durante toda a minha graduação. Os amigos e irmãos que aqui criei serão lembrados para sempre.

Aos amigos feitos ao longo dessa jornada, pelos bons momentos, boas conversas e apoio nos estudos, dentro e fora da sala de aula. Os amigos que hoje considero parte da minha família, assim como sou considerado parte das suas famílias.

À Camila, pelo apoio diário e suporte em momentos de decisão. Sinônimo de leveza, que me traz uma luz constante em minha vida. Seguirei sempre ao seu lado.

Ao Professor Glauco, por todos os ensinamentos ao longo da minha caminhada acadêmica e pelo incentivo à especialização na área de Telecomunicações. O apoio e a persistência em momentos difíceis sempre foram muito importantes, e estarei sempre torcendo pelo seu sucesso pessoal e acadêmico.

A todos que me cercam e torcem por mim, o meu muito obrigado.

"Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça o que a maioria não faz."

Bill Gates

Resumo

O presente trabalho aborda a utilização das técnicas de Modelos Ocultos de Markov (HMM, do inglês *Hidden Markov Models*) e aprendizagem profunda através de Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*) aplicadas em sistemas de reconhecimento de fala. Foi desenvolvido um sistema de reconhecimento de comandos de fala dependente do locutor, onde serão avaliadas duas bases gravadas pelo mesmo locutor. Ambas as técnicas utilizadas partem do princípio de treinamento do modelo em cima das bases e depois a etapa de testes no reconhecimento através da extração de características. Para que estes sistemas tenham um reconhecimento assertivo, os modelos criados devem ser minimamente impactados por ruídos externos, o que depende em parte do pré-processamento e também da obtenção de modelos acústicos robustos. Para tal, neste trabalho é feita a remoção de silêncio a fim de aprimorar o reconhecimento com base na extração de características. Os resultados demonstraram que, enquanto o modelo CNN apresentou desempenho satisfatório com acurácia média entre 93% e 98%, o HMM obteve resultados superiores, alcançando acurácia média de 99% nas bases testadas, evidenciando maior capacidade de generalização e resistência a ruídos. Essas métricas confirmam a efetividade do aprendizado profundo na tarefa de reconhecimento automático de fala, especialmente em contextos de variação temporal e de pré-processamento otimizado.

Palavras-chave: Sistema de Reconhecimento de Fala, Modelos Ocultos de Markov, Aprendizado Profundo, Redes Neurais Convolucionais

Abstract

This paper addresses the use of Hidden Markov Model (HMM) and deep learning techniques through convolutional neural networks (CNN) applied to speech recognition systems. A speaker-dependent speech command recognition system were developed, where two databases recorded by the same speaker will be evaluated. Both techniques used are based on the principle of training the model on the databases and then the recognition testing stage through feature extraction. For these systems to have assertive recognition, the models created must be minimally impacted by external noise, which depends in part on preprocessing and also on obtaining robust acoustic models. To this end, this work removes noise and silence cutouts in order to improve recognition based on feature extraction. The results demonstrated that, while the CNN model performed satisfactorily with average accuracy between 93% and 98%, the HMM achieved superior results, achieving an average accuracy of 99% on the tested datasets, demonstrating greater generalization capacity and resistance to noise. These metrics confirm the effectiveness of deep learning in automatic speech recognition, especially in contexts with temporal variation and optimized preprocessing.

Keywords: Speech Recognition System, Hidden Markov Models, Deep Learning, Mel Cepstral Coefficients, Convolutional Neural Networks

Lista de Figuras

Lista de tabelas

Sumário

1 Introdução

1.1 Contextualização

A rápida evolução tecnológica tem impulsionado o desenvolvimento de Reconhecimento Automático de Fala (ASR, do inglês *Automatic Speech Recognition*), permitindo a interação entre humanos e computadores de uma forma mais natural e intuitiva. Essa tecnologia tem-se mostrado extremamente relevante em diferentes aplicações, oferecendo benefícios significativos, do ponto de vista da interação humano-computador e impactando positivamente a sociedade.

Um sistema de Reconhecimento de Fala Automático (ASR) é uma tecnologia que processa sinais de áudio para reconhecer e converter a linguagem falada em texto. Esses sistemas utilizam vários componentes, como unidades de detecção de fala, unidades de fornecimento de informações e unidades de seleção para otimizar a precisão e a eficiência do reconhecimento (HOMMA et al., 2019). Eles envolvem extração de características, modelos acústicos e análise de probabilidade de palavras-chave para reduzir a carga computacional e os requisitos de *hardware* (ZHAN; XIN, 2020).

Segundo (SINGH, 2019), existe um crescente otimismo em torno da futura integração da Interface Homem-Máquina (MMI, do inglês *Man-machine interface*) usando a tecnologia de fala, onde se destaca a mudança para sistemas de reconhecimento de fala já na fabricação de computadores.

A integração de sistemas de reconhecimento de voz em várias aplicações, como controle de acesso, segurança bancária e pagamento móvel, destaca sua importância em aumentar a segurança e a conveniência na vida diária (SINGH, 2019).

Diante dessa perspectiva, é importante compreender os ganhos proporcionados pela utilização de sistemas de reconhecimento de fala. Essa compreensão é fundamental para explorar o potencial dessa tecnologia e incentivar sua adoção em diversas aplicações.

1.2 Estado da arte

Os sistemas de Reconhecimento Automático de Fala(ASRs) que utilizam Modelos Ocultos de Markov (HMM, do inglês *Hidden Markov Models*) têm sido fundamentais no desenvolvimento de tecnologias de reconhecimento de fala. Estes sistemas baseados em HMM são projetados para modelar a variabilidade temporal da fala e têm sido usados com eficácia em vários idiomas e aplicações.

O desenvolvimento do primeiro sistema ASR para a linguagem Tulu empregou os modelos Modelo de Mistura Gaussiana (GMM, do inglês *Gaussian Mixture Model*) e

Redes Neurais Profundas (DNN, do inglês *Deep Neural Network*) compilando modelos híbridos GMM-HMM e DNN-HMM, revelando que os modelos monofônicos GMM-HMM tiveram um desempenho melhor com dados limitados em comparação com os modelos de trifone, que requerem conjuntos de dados mais extensos para um desempenho ideal (AMOOLYA et al., 2022).

No contexto de sistemas de fala interativos, Hamidi et al. (2021) um ASR baseado em HMM foi utilizado para reconhecer palavras, alfabetos e dígitos específicos na língua Amazigh, alcançando alto desempenho e aprimorando a capacidade do sistema de entender e processar comandos de voz com precisão. A pesquisa demonstrou que o sistema atinge uma alta taxa de reconhecimento de mais de 80% para usuários administradores autorizados, enquanto mantém uma baixa taxa de reconhecimento de menos de 5% para usuários não administradores, destacando efetivamente seus recursos de segurança (HAMIDI et al., 2021).

Os sistemas de reconhecimento de fala que utilizam HMM ganharam interesse significativo em vários trabalhos de pesquisa. E, além disso, os avanços na modelagem acústica, que são cruciais para sistemas baseados em HMMs, se concentram em aumentar a robustez ao ruído ambiental, às condições do canal e às variações dos alto-falantes, abordando variabilidade de pronúncia (ANUJA; AKSHATHA; JAYAPRAKASH, 2022).

Apesar da robustez dos sistemas tradicionais baseados em HMM, as abordagens modernas estão incorporando cada vez mais modelos de ponta a ponta. Métodos baseados em Redes Neurais Artificiais (ANN, do inglês *Artificial Neural Network*) combinados com técnicas de otimização aprimoram a comunicação, melhorando a precisão do reconhecimento e reduzindo ruídos indesejados, tornando a aplicação de pesquisa por voz mais confiável (ANUJA; AKSHATHA; JAYAPRAKASH, 2022). Esses avanços destacam o cenário em evolução das tecnologias ASR, em que os sistemas baseados em HMM continuam a desempenhar um papel vital, especialmente em cenários com dados limitados, enquanto os modelos mais novos ultrapassam os limites de desempenho e robustez em condições mais complexas e ruidosas.

As redes neurais artificiais (ANNs) e as Redes Neurais Profundas (DNN, do inglês *Deep Neural Network*) são fundamentais na evolução de modelos computacionais que imitam as funções do cérebro humano se baseando no funcionamento dos neurônios. As DNNs, um subconjunto das ANNs, surgiram com o avanço tecnológico e oferecem representações mais complexas de alto nível e têm transformado as mais diversas áreas da indústria e sociedade, especialmente os campos de visão computacional, processamento de sinais, reconhecimento de voz e processamento de linguagem natural (VANNESCHI; SILVA, 2022).

O aprendizado profundo, do inglês *Deep Learning*, é uma das formas mais conhecidas de se referir às DNNs. Com os recentes aprimoramentos das arquiteturas de processadores e placas gráficas, o maior poder computacional possibilitou o uso e desenvolvimento

mais intensivo de soluções utilizando as Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*). Estas podem ser vistas como um tipo especializado das redes neurais profundas (DNNs). Ao contrário das DNNs tradicionais, que dependem da multiplicação geral de matrizes, as CNNs utilizam uma operação matemática fundamental na rede conhecida como convolução em pelo menos uma de suas camadas.

No trabalho apresentado por How et al. (2022), a integração de modelos de aprendizado profundo com CNNs e DNNs foi explorada para reconhecer as emoções da fala, com as CNNs superando as DNNs em termos de precisão e função de perda, alcançando uma maior precisão de 76,50%.

As CNNs têm sido amplamente usadas em sistemas de reconhecimento de fala para detectar fonemas mal pronunciados e mostraram resultados notáveis em várias aplicações, com uma taxa de precisão de 91,81% e uma taxa de erro de palavras de 12,4%. (SOUNDARYA; KARTHIKEYAN; THANGARASU, 2023)

Este tipo de rede está impulsionando grandes avanços em visão computacional, que têm aplicações importantes em carros autônomos, robótica, drones, segurança, diagnósticos médicos e tratamentos para deficientes visuais.

1.3 Justificativa

O reconhecimento de fala é uma área multidisciplinar baseada em conhecimentos de processamento digital de sinais, aprendizado de máquina, estatística, dentre outros. Essa técnica está desempenhando um papel crucial em diversas aplicações, que envolvem desde assistentes virtuais em dispositivos móveis até sistemas de controle por voz em ambientes industriais.

A busca incessante por métodos mais eficazes e precisos no reconhecimento tem impulsionado o desenvolvimento de diversas técnicas de classificação ao longo das últimas décadas. Os sistemas de reconhecimento de fala ASRs são amplamente empregados atualmente e a precisão depende do método utilizado em combinação com a base de dados a ser comparada.

Para se definir o melhor modelo, o estudo atual apresenta uma comparação entre modelos a fim de identificar o método mais eficaz nas aplicações de reconhecimento de fala. Desta forma, o estudo visa aprimorar a interação humano-computador por meio do Reconhecimento Automático de Fala (ASR), que permite ao computador compreender as palavras faladas e transformá-las em formas textuais ou outras formas de interação.

1.4 Formulação do problema

Duas abordagens que emergiram no contexto de sistemas de reconhecimento de fala são os Modelos Ocultos de Markov (HMMs) e os modelos de aprendizado profundo. Essas

técnicas são baseadas em paradigmas diferentes e utilizadas separadamente de acordo com as limitações de dados, desempenho computacional e objetivos a serem alcançados pelo sistema de reconhecimento de fala.

Os métodos tradicionais de reconhecimento de fala, como HMMs, são usados há muito tempo, mas têm limitações em termos de precisão e eficiência. Recentemente, modelos de aprendizado profundo se mostraram mais eficazes em lidar com as complexidades das tarefas de reconhecimento de fala.

Os HMMs, que há muito tempo têm sido uma escolha padrão, oferecem uma estrutura robusta para modelar sequências temporais, especialmente na análise de características espectrais. Por outro lado, o avanço da tecnologia e aprimoramento do aprendizado profundo trouxeram métodos como DNNs e CNNs.

Considerando que o desempenho observado no reconhecimento depende diretamente do método de classificação empregado, este estudo visa explorar e comparar o desempenho dessas duas categorias de métodos no contexto do reconhecimento de fala.

1.5 Objetivos

O objetivo geral deste trabalho é comparar o desempenho entre métodos tradicionais de aprendizado de máquina baseados em HMM e modelos de aprendizado profundo baseados em CNN para determinar qual deles é o mais eficaz em sistemas de reconhecimento de fala.

1.5.1 Objetivos específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos devem ser cumpridos:

- Gravação de duas bases de áudios dependente de locutor;
- Implementação de um sistema de reconhecimento de fala utilizando o HMM e aprendizado de máquina;
- Implementação de um sistema de reconhecimento de fala utilizando CNN e aprendizado profundo;
- Treinamento dos modelos utilizando os conjunto de dados gravados pelo locutor;
- Testes e aferições com alterações nos parâmetros dos modelos a fim de encontrar a melhor eficiência de cada modelo;
- Realizar a validação experimental a fim de validar a eficácia dos modelos

1.6 Estrutura do trabalho

No Capítulo 1, foi feita uma contextualização dos ASRs e apresentado o estado da arte com algumas de suas aplicações, foi apresentada a justificativa e a formalização do problema, além de apresentar os objetivos do trabalho.

O Capítulo 2 apresenta toda a revisão de literatura apresentando os conceitos atrelados aos sistemas de reconhecimento de fala. Os modelos ocultos de Markov HMMs e redes neurais convolucionais foram detalhados com conceitos, estruturas e técnicas utilizadas nas aplicações dos mesmos. Conceitos básicos e necessários ao desenvolvimento do trabalho foram explicados para que possamos compreender melhor as discussões abordadas ao longo do trabalho.

O Capítulo 3 apresenta a metodologia utilizada na construção do modelo oculto de marvok (HMM) do presente trabalho, descrevendo todas as etapas do modelo, bem como os parâmetros a serem considerados na execução dos experimentos. Em uma segunda parte do Capítulo 3, foi apresentada a metodologia utilizada na construção da rede neural convolucional (CNN) do presente trabalho e detalhados todos os parâmetros e etapas de construção desta rede.

O Capítulo 4 apresenta uma série de experimentos realizados para o treinamento e validação dos modelos HMM e CNN, os quais foram conduzidos de forma comparativa e estruturada em diferentes etapas de análise. Para o modelo HMM, foram avaliados o limiar do recorte de silêncio aplicado aos dados de entrada, a variação no número de Gaussianas utilizadas na modelagem e o desempenho do modelo final após o ajuste dos parâmetros. Já para o modelo CNN, os experimentos compreenderam a análise do limiar de recorte de silêncio, a aplicação de técnicas de distensão temporal nos dados de entrada e a avaliação do modelo final de rede convolucional. Todos os experimentos foram aplicados a duas bases de dados distintas, permitindo comparar o comportamento e a eficiência de cada abordagem sob diferentes condições de pré-processamento e configuração dos modelos.

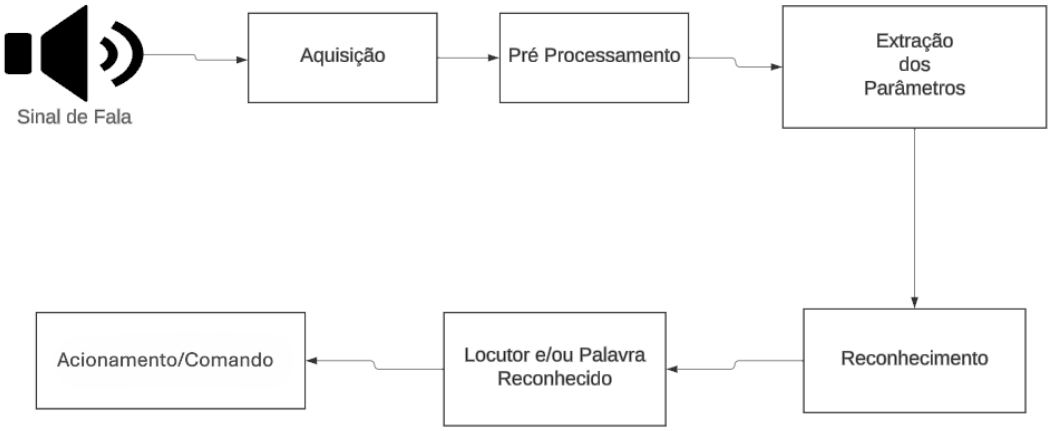
2 Revisão bibliográfica

2.1 Sistema de Reconhecimento de Fala

O campo do ASR evoluiu significativamente nas últimas décadas, marcado por fases distintas e avanços tecnológicos. Os primeiros estudos surgiram com o intuito de criar máquinas capazes de reconhecer e entender a fala de qualquer alto-falante em qualquer ambiente, uma meta que impulsiona pesquisas há mais de 70 anos (RABINER; JUANG, 2007).

A representação de um modelo ASR pode ser vista na Figura 1. A principal função dos sistemas ASRs é transformar um sinal de entrada acústica da fala em uma sequência simbólica correspondente (fonemas, trifones, palavras, etc).

Figura 1 – Sistema de Reconhecimento de Fala.



Fonte: Do autor.

Um sistema ASR, genericamente, consiste em uma parte frontal de processamento de sinal e uma parte de modelagem e reconhecimento. A tarefa principal da parte frontal é analisar o sinal acústico de entrada e extrair os eventos acústicos relevantes que identificam características específicas da fala, como a posição e o movimento da língua do falante. Essas informações devem ser representadas em termos de um conjunto compacto e eficiente de parâmetros de fala.

A etapa subsequente utiliza essas características para analisar e reconhecer o conteúdo fonético do sinal de fala de entrada. Para melhorar o desempenho em ambientes ruidosos, alguns sistemas incorporam uma unidade de medição de ruído que ajusta os padrões de rejeição com base nos níveis de ruído ambiente, garantindo um reconhecimento mais preciso ao rejeitar resultados não confiáveis (SAKOE, 1978).

2.2 Coeficientes Mel-Cepstrais

Nos sistemas de reconhecimento de fala, a etapa do pré-processamento consiste na limpeza e remoção de ruídos do sinal de entrada. Ao combinar o pré-processamento de sinais de fala e a extração de recursos do Coeficientes Mel-Cepstrais (MFCC, do inglês *Mel-frequency cepstral coefficients*), os modelos em estudo derivam um algoritmo de reconhecimento de fala.

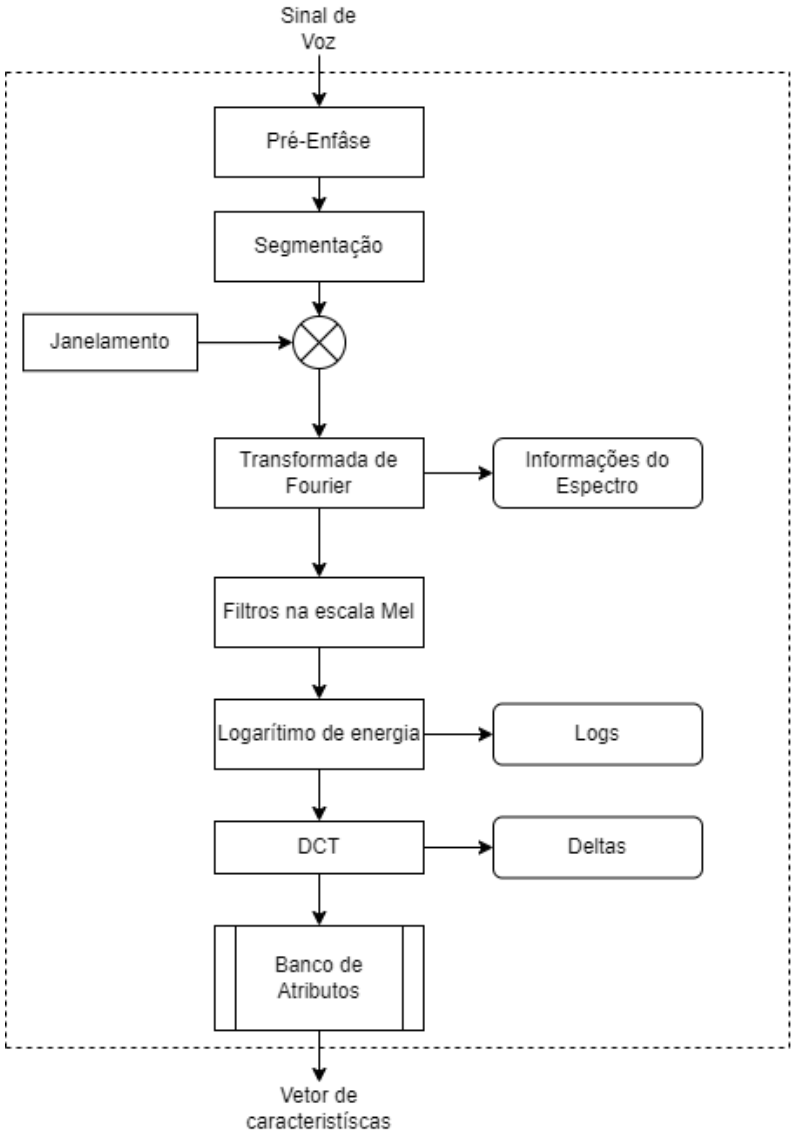
Os MFCC são usados para representar aspectos importantes dos sinais de fala originais, que são então usados como entrada para os modelos dos sistemas de reconhecimento de fala. Ao converter os dados brutos de áudio em um conjunto de coeficientes, os MFCCs fornecem uma representação parametrizada do sinal de fala que o HMM e o CNN podem processar com eficácia (HUANG; ZHU; GUO, 2020).

Os MFCCs são derivados da escala Mel, que imita a resposta do ouvido humano a diferentes frequências, tornando-os altamente eficazes para análise de áudio. . Conforme demonstrador por Wassner e Chollet (1996), as taxas de erro foram reduzidas eme 50% no reconhecimento de palavras conectadas.

A Figura 2 mostra detalhadamente o processo de extração de MFCCs, composto pelas etapas de pré-ênfase, enquadramento, janelas, Transformada Rápida de Fourier (FFT), processamento do banco de filtros Mel e Transformada Discreta de Cosseno (DCT).

- Pré-ênfase: Esta etapa envolve a passagem do sinal de fala por um filtro que enfatiza frequências mais altas, o que ajuda a equilibrar o espectro e melhorar a relação sinal/ruído.
- Janelamento: O sinal de áudio é dividido em pequenos segmentos chamados de quadros, geralmente com duração de 20ms a 30ms. Essa etapa é crucial para considerar a natureza não estacionária do sinal de fala, uma vez que as propriedades acústicas podem variar rapidamente. Cada quadro é multiplicado por uma função de janela, como a janela Hamming, para minimizar as descontinuidades nas bordas dos quadros.
- Transformada Rápida de Fourier (FFT): Os dados em janela são transformados no domínio da frequência usando o FFT.
- Filtros na escala Mel: O espectro de potência obtido da FFT é passado por um conjunto de filtros triangulares espaçados de acordo com a escala Mel, o que ajuda a enfatizar as frequências perceptualmente importantes.
- Logaritmo: Calcula-se a energia na saída de cada filtro triangular do banco.

Figura 2 – Coeficientes Cepstrais de Frequência Mel (MFCCs).



Fonte: Do autor.

- Transformada de cosseno discreta (DCT): A DCT é aplicada sobre o conjunto de valores de energia obtidos em cada filtro do banco, de modo que o resultado deste cálculo fornece os coeficientes MFCCs.

Os MFCCs são amplamente usados porque fornecem uma boa discriminação entre os diferentes sons da fala, têm baixa correlação e capturam características fonéticas importantes, tornando-os altamente eficazes para tarefas de reconhecimento de fala. De acordo com Huang, Zhu e Guo (2020), os MFCCs são usados como um vetor de características em combinação com modelos ocultos de Markov (HMMs).

2.3 Modelos Ocultos de Markov - HMM

Os sistemas de reconhecimento de fala demandam uma modelagem sofisticada capaz de incorporar informações relevantes representadas no espaço de características acústicas, e também a variabilidade temporal.

Um modelo oculto de Markov (HMM) é um modelo estatístico usado para representar sistemas que são considerados um processo de Markov com estados ocultos, o que significa que o sistema passa por transições de um estado para outro, mas os estados em si não são diretamente visíveis para o observador (PING, 2021).

No contexto deste trabalho, o HMM é usado para reconhecimento de fala e este modelo é projetado levando-se em consideração tanto o número de estados quanto o número de gaussianas por estado.

O número de estados em um HMM representa diferentes segmentos do sinal de fala que o modelo tenta capturar. O número de gaussianos por estado se refere à complexidade da distribuição da probabilidade de emissão para cada estado. A partir das misturas gaussianas, o modelo pode identificar melhor a variabilidade dentro de cada estado.

A relação entre o número de gaussianas e os estados é crucial porque determina a capacidade do modelo de representar com precisão as características acústicas da fala. Mais estados podem fornecer uma segmentação mais precisa do sinal de fala, enquanto mais gaussianas por estado podem oferecer uma representação mais detalhada da variabilidade acústica dentro de cada estado.

Em um HMM, existem dois componentes principais: Os estados e observações. Os estados são as partes ocultas do modelo que não podem ser identificadas de forma direta, enquanto as observações são as saídas visíveis que podem ser medidas ou registradas.

As probabilidades de transição representam as chances de passar de um estado para outro. Essas probabilidades ajudam a determinar a probabilidade de transição entre os estados no modelo e ajudam a prever a sequência de estados ao longo do tempo.

Cada estado no HMM tem uma distribuição de probabilidade associada às observações possíveis (probabilidades de emissão). A modelagem dos eventos de fala, como o início de um fonema, por exemplo, ocorre através destas distribuições de probabilidades dos estados, e a duração desses eventos é modelada através de probabilidades de emissão e transição de estado. Sendo assim, o HMM é capaz de observar as variações temporais entre diferentes amostras de uma mesma palavra.

Para inicializar o modelo, gera-se a distribuição do estado inicial, considerando um número fixo de Gaussianas que especifica a probabilidade de o sistema começar em cada estado possível. Essas probabilidades (Gaussianas) definem o ponto de partida para o modelo e são essenciais para inicializar os cálculos em algoritmos como o algoritmo de Viterbi.

Os HMMs exigem um treinamento do modelo. Este treinamento envolve o processo de estimação dos parâmetros do modelo (probabilidades de transição, probabilidades de

emissão e distribuição de estado inicial) a partir de um conjunto de dados observados.

Após ter a base de palavras descritas por seus modelos, o HMM realiza o processo de decodificação onde tem-se por objetivo encontrar a sequência mais provável de estados ocultos dada uma sequência de observações. O algoritmo de Viterbi é comumente usado para esse propósito, fornecendo uma maneira eficiente de determinar a melhor sequência de estados que explica os dados observados (PING, 2021).

Cada palavra da base de dados é representada por uma sequência de estados e dessa forma foi criado um modelo HMM para cada palavra. Os modelos acústicos baseados em HMM podem ser representados pela forma compacta, sendo π as probabilidades iniciais, A a matriz de covariância e B as probabilidades de transição de estados.

$$\lambda = (A,B,\pi) \tag{2.1}$$

Após obter os modelos através do treinamento, a etapa de reconhecimento faz a comparação de um sinal de áudio e verifica qual dos modelos treinados produz a maior verossimilhança, de modo a se determinar qual é a palavra produzida.

2.4 Algoritmo de Baum-Welch

O Algoritmo Baum-Welch (BWA, do inglês *Baum-Welch algorithm*) é um algoritmo de aprendizado de máquina utilizado na fase de treinamento de Modelos Ocultos de Markov (HMMs). É frequentemente aplicado em sistemas de reconhecimento de fala para ajustar os parâmetros do modelo HMM com base em dados de treinamento.

Para estimar os parâmetros do HMMs, o algoritmo funciona de forma iterativa para melhorar as estimativas destes parâmetros. Para isso, uma estimativa inicial dos parâmetros é utilizada e, em seguida, refina repetidamente essas estimativas para maximizar a probabilidade dos dados observados. Esse processo iterativo continua até que a probabilidade de encontrar os dados observados não aumente mais significativamente ou atinja um valor de verossimilhança desejado (ANNAS; OUZINEB; BENYACOUB, 2022).

Durante o treinamento, o BWA executa duas etapas de maximização da verossimilhança, com o objetivo de encontrar estes parâmetros desconhecidos de um modelo oculto de Markov. As etapas são divididas em Etapa de expectativa (E-*Step*) e Etapa de maximização (M-*Step*).

Na etapa de expectativa, o algoritmo calcula o número esperado de vezes que cada transição de estado ocorre e o número esperado de vezes que cada estado é visitado, dadas as estimativas dos parâmetros atuais. Essas expectativas são calculadas usando as probabilidades para frente e para trás e dispõem informações sobre os estados ocultos do HMM.

Na etapa de maximização, o algoritmo atualiza as estimativas dos parâmetros para maximizar a probabilidade esperada calculada na etapa anterior. Isso envolve atualizar

as probabilidades de transição entre os estados e as probabilidades de observar cada ocorrência em cada estado. Nesta etapa, os parâmetros HMM são atualizados com o objetivo de maximizar a verossimilhança dos dados observados.

Os passos E-*Step* e M-*Step* são repetidos iterativamente até que os parâmetros convirjam para uma solução ótima ou até que um critério de parada seja alcançado. A convergência do algoritmo garante que os parâmetros não mudem significativamente entre iterações sucessivas. Com isso, ao final das iterações, os parâmetros do HMM estão ajustados para melhor se adequar aos dados de treinamento, melhorando a representação de uma determinada sequência de fala.

2.5 Algoritmo de Viterbi

Na etapa de decodificação, o algoritmo Viterbi é empregado para se determinar a sequência de estados mais provável que resultam na identificação de palavras faladas a partir de um determinado conjunto de observações.

O algoritmo Viterbi funciona calculando iterativamente a probabilidade do caminho mais provável para cada estado em cada etapa de tempo, usando as probabilidades dos estados anteriores e as probabilidades de transição entre os estados (PING, 2021).

O algoritmo de Viterbi atua na etapa de decodificação dentro do HMM. Para isso, o algoritmo remonta do estado final ao estado inicial, seguindo o caminho que maximizou as probabilidades em cada etapa. Essa etapa envolve retroceder pelos estados para encontrar a sequência que levou à maior probabilidade no estado final e, assim, reconhecer qual foi a palavra falada.

2.6 Redes Neurais Convolucionais - CNN

As redes neurais convolucionais (CNNs) são redes neurais artificiais profundas que podem ser usadas para classificar imagens, agrupá-las por similaridade e realizar reconhecimento de padrões.

As CNNs também podem realizar o reconhecimento óptico de caracteres para digitalizar textos e tornar possível o processamento de linguagem natural em documentos analógicos e manuscritos, onde as imagens são símbolos a serem transcritos. As CNNs também podem ser aplicadas a arquivos de áudio quando estes são representados visualmente como um espectrograma.

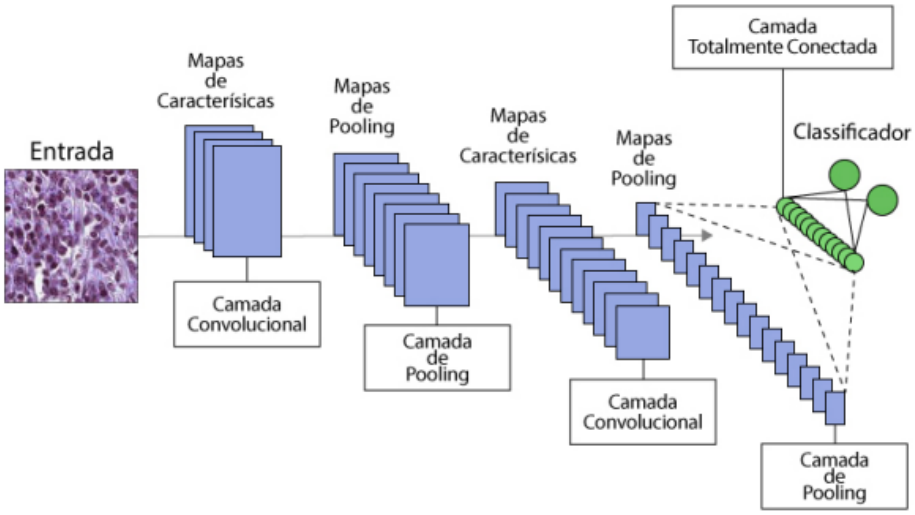
O processamento nessas redes ocorre em vários blocos de construção, como camadas de convolução, camadas de agrupamento e camadas totalmente conectadas.

Um método chave usado em CNNs é a operação de convolução. Isso envolve aplicar um filtro (ou *kernel*) à imagem de entrada, deslizando-o por pequenas seções da imagem. Esse processo gera um mapa de ativação, que destaca a presença de características espe-

cíficas na imagem. A operação de convolução é fundamental para a capacidade das CNNs de detectar padrões e características em imagens.

As CNNs são compostas por várias camadas, incluindo camadas convolucionais, camadas de agrupamento e camadas totalmente conectadas. Essas camadas trabalham juntas para extrair e aprender recursos dos dados de entrada. (MUNIR; KONG; QURESHI, 2023). A Figura 3 exemplifica uma CNN composta por duas camadas convolucionais, duas camadas de agrupamento(*pooling*) e uma camada totalmente conectada. A saída final é reduzida a um único vetor de classificação.

Figura 3 – Ilustração da arquitetura de uma CNN com duas camadas convolucionais, duas de *pooling*, uma totalmente conectada e a de saída



Fonte: Retirado de Renesio (2019).

2.6.1 Camadas convolucionais

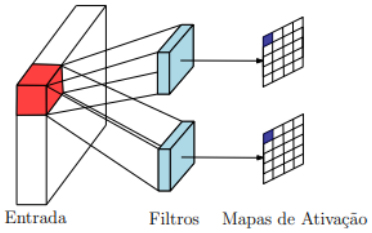
A camada convolucional geralmente aparece mais de uma vez dentro da rede neural, sendo sempre a primeira camada da rede. A camada convolucional foi projetada para detectar e extrair as características (*features*) dos dados de entrada.

A camada aplica um conjunto de filtros aos dados de entrada. Cada filtro desliza sobre os dados de entrada, executando um produto ponto a ponto entre o filtro e uma pequena região da entrada. Essa operação é conhecida como convolução e resulta em um mapa de características que representa a presença de recursos específicos nos dados de entrada.

A Figura 4 exemplifica a aplicação de dois filtros a um dado de entrada, resultando em dois mapas de características. A posição (1,1) está marcada para identificar a primeira operação da convolução. A varredura acontece ponto a ponto, onde o filtro passa por todas as posições até que seja formado o mapa de características final.

Cada filtro resulta em um mapa de características que identifica uma característica específica e, posteriormente, cada mapa passa por uma função de ativação para gerar um mapa de ativação. Na Figura 4 temos duas características sendo levantadas por dois filtros, e cada mapa é responsável por identificar uma dessas características.

Figura 4 – Ilustração da operação realizada pela camada convolucional. Dois filtros são aplicados à entrada, resultando em seus respectivos mapas de ativação.



Fonte: Retirado de Kovaleski (2018).

Os filtros podem ser ajustados para identificar padrões específicos e ajustados conforme a complexidade do modelo e da rede a ser aplicada. O primeiro parâmetro a ser ajustado é o tamanho do filtro convolucional e este é usualmente chamado de *kernel size*. A Figura 5 exemplifica um filtro de tamanho 3x3 (kernel 3) e as equações (2.2) e (2.3) exemplificam o processo de convolução e deslizamento deste filtro sob os dados de entrada.

$$O_{1,1} = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} F_{i,j} *_I i,j \tag{2.2}$$

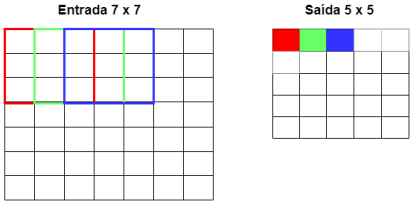
$$O_{1,2} = \sum_{i=1}^{i=3} \sum_{j=2}^{j=4} F_{i,j} *_I i,j \tag{2.3}$$

na qual `O` são as saídas dos mapas de características, `i` e `j` são as dimensões da matriz para convolução, `F` o filtro aplicado aos dados de entrada definidos por `I`.

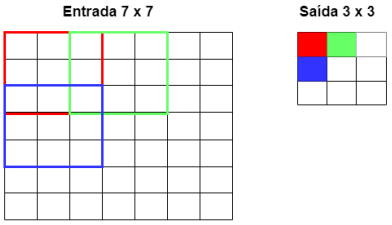
Além do tamanho do filtro como parâmetro ajustável, é possível definir o padrão de deslocamento do filtro sobre os dados, a quantidade de valores considerados em cada operação e inserir novos valores ao conjunto a ser analisado. Os parâmetros de passo, do inglês *stride*, e preenchimento , do inglês *padding*, são os mais frequentemente utilizados para realizar tais definições. (KOVALESKI, 2018).

O passo controla como o filtro fará as convoluções em torno do dado de entrada. A Figura 5(a) exemplifica o deslocamento gerado com o passo sendo 1 e resulta no deslocamento do filtro de uma amostra por vez. O passo é normalmente definido de forma que a dimensão da saída seja um número inteiro e não uma fração. A Figura 5(b) mostra o que acontece com a saída quando se altera o valor do passo para 2 no mesmo filtro de tamanho 3 x 3 e o salto passa a ser 2. O deslocamento deste filtro ocorre coluna a coluna e depois linha a linha a fim de analisar todo o dado de entrada.

Figura 5 – Filtro de tamanho 3 x 3 na camada de convolução



(a) Passo(*stride*) de valor 1.



(b) Passo (*stride*) de valor 2.

Fonte: Do autor.

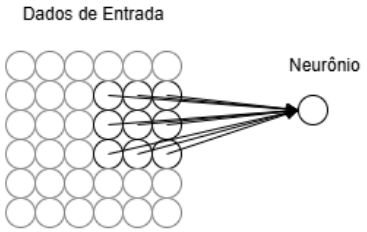
O passo define o movimento do filtro, ou seja, com um passo de tamanho 1 o filtro caminha uma amostra por vez. Quanto maior o valor do passo, menor será a dimensão da saída e isso não é o desejado em uma camada de convolução. Especificamente nas primeiras camadas da rede, devem ser preservadas o máximo de informações sobre o dado de entrada original para que sejam extraídos o maior número de características.

Uma técnica que auxilia na preservação dos dados é o preenchimento zero, do inglês *zero padding*, nessa camada. O preenchimento zero preenche o volume de entrada com zeros ao redor da borda e faz com que a entrada e a saída possuam o mesmo tamanho.

O conjunto destes parâmetros é definido como os hiperparâmetros e pode variar de acordo com o tamanho, complexidade, tipo de tarefa de processamento e objetivo de aplicação da rede neural com essa camada. Ao analisar um conjunto de dados, a escolha dos hiperparâmetros deve ser feita levando-se em consideração os objetivos específicos da rede neural. As primeiras camadas convolucionais detectam características de baixos níveis de complexidade, mas a rede deve ser projetada para que sejam detectadas características de altos níveis.

A Figura 6 mostra os dados de entradas divididos por posição.Cada neurônio da camada seguinte estará conectado a uma pequena região da camada de entrada. Na figura em questão, tem-se 3 x 3 conexões feitas ao neurônio, um total de 9 dados de entrada.

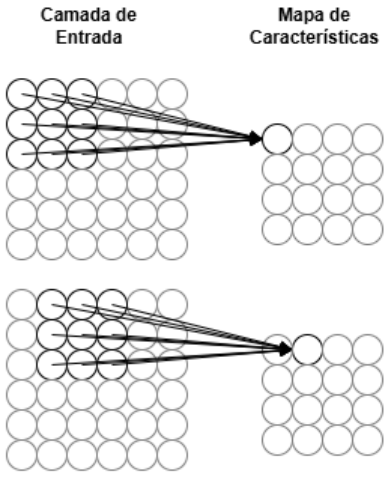
Figura 6 – Campo receptivo local de um filtro 3 x 3



Fonte: Do autor.

A região em negrito da Figura 6 é chamada de campo receptivo local para o neurônio oculto. É uma pequena janela em cima dos dados de entrada e esta deve ser deslocada por toda a imagem de entrada para realizar a operação de convolução. Para cada campo receptivo local, existe um neurônio oculto diferente na primeira camada oculta. A Figura 7 mostra o deslocamento da primeira janela para o seu neurônio de referência.

Figura 7 – Deslocamento do campo receptivo local para criação do mapa de características



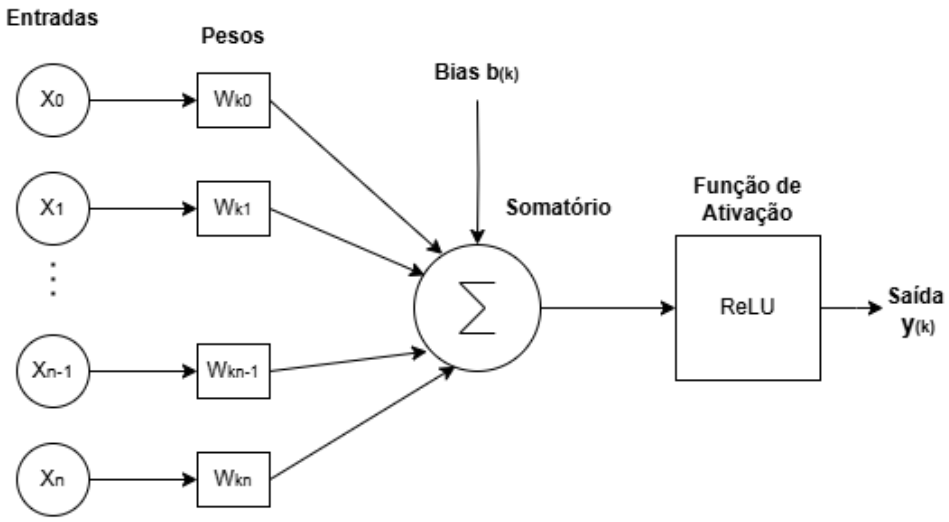
Fonte: Do autor.

O movimento de deslizamento continua até que todos os neurônios ocultos sejam definidos e seja construído o mapa de características referentes a esse filtro. No exemplo acima, temos uma entrada de 6 x 6 e, utilizando um filtro de tamanho 3 x 3, haverá 4 x 4 neurônios no mapa de características, um total de 16 neurônios para o exemplo em questão.

Cada neurônio aprende um peso a ser aplicado em cada uma de suas conexões criadas e o neurônio oculto também aprende um viés, do inglês, *bias* geral que entra como uma constante no somatório aplicado. A Figura 8 demonstra o funcionamento do neurônio

e os pesos aplicados aos dados de entrada para se definir o valor na saída de cada um dos neurônios que reflete no mapa de características.

Figura 8 – Função somátoria com pesos e bias de um neurônio em uma CNN.



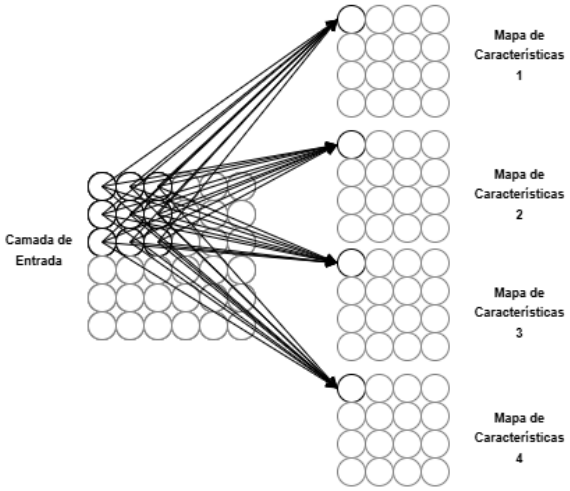
Fonte: Do autor.

Cada neurônio tem um *bias* e pesos conectados ao seu campo receptivo local. Para um determinado mapa de características, todos os neurônios compartilham os mesmos pesos e mesmos *bias*. No exemplo anterior, os 16 neurônios que compõem o mapa de características são treinados para determinar um tipo de característica específica daquele mapa e compartilham os mesmos pesos e viés.

Para redes neurais de complexidade maior, torna-se desejável a projeção de mais de um mapa, onde cada mapa irá representar uma característica específica. Por exemplo, em uma detecção de imagem, um mapa pode ser treinado para detectar bordas, outro para detectar linhas verticais, outro para detectar linhas horizontais e um quarto para detectar preenchimento.

A Figura 9 mostra as diferentes conexões feitas para cada mapa de características, onde cada mapa possui seus respectivos pesos e vieses compartilhados.

Figura 9 – Mapas de características formados a partir de dados de entrada

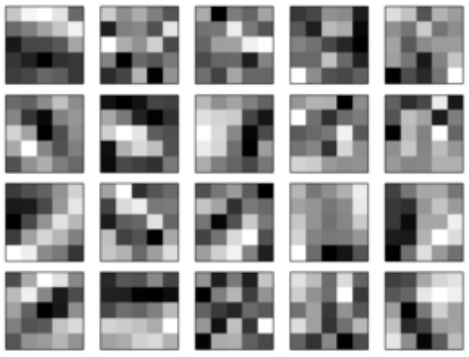


Fonte: Do autor.

No exemplo, foram utilizados 4 mapas de características, possuindo 16 neurônios cada um e representando uma camada oculta de 64 neurônios no total. Na prática, as CNNs de complexidade mais elevadas possuem cerca de 20 a 40 mapas de características.

O compartilhamento de pesos e *bias* dentro de cada mapa traz uma economia de processamento e definição de parâmetros. Por exemplo, na Figura 10, são utilizados mapas de tamanho 5 x 5, totalizando 25 neurônios por mapa e, considerando 20 mapas, um total de 500 neurônios. Se fossem aplicados filtros 5 x 5, 12.500 pesos seriam utilizados para determinar e 500 *bias*. Mas com o compartilhamento de pesos e *bias*, tem-se um total de 500 pesos e 25 *bias* nessa camada da rede neural em questão.

Figura 10 – Exemplo de 20 mapas de caracteríscitas de uma CNN



Fonte: Retirado de Data Science Academy (2022).

2.6.2 Camadas ReLU (Unidades Lineares Retificadas)

Após cada camada convolução, é comum fazer a aplicação de uma camada não linear, ou camada de ativação, imediatamente depois. A função de ativação é aplicada em

cima de cada neurônio do mapa de características e transforma este mapa em um mapa de ativação para que o neurônio seja ativo ou não quando identificar a característica que ele está treinado para reconhecer.

O propósito desta camada é introduzir a não linearidade a um sistema que realizou operações lineares durante as camadas de convolução, permitindo que a rede aprenda padrões mais complexos.

A camada ReLU aplica a função

$$f(x) = max(0, x), \tag{2.4}$$

e significa que ela emite a entrada diretamente se for positiva, caso contrário, ela gera zero. Esse limite simples em zero traz uma eficiência computacional para a rede e permite um treinamento mais rápido dos modelos.

Esta camada também ajuda a aliviar o problema do gradiente de desaparecimento, que é o problema em que as camadas inferiores da rede treinam muito lentamente devido à diminuição do gradiente através das camadas. Isso ocorre porque o gradiente do ReLU é zero ou um, garantindo que os gradientes não diminuam à medida que se propagam pela rede.

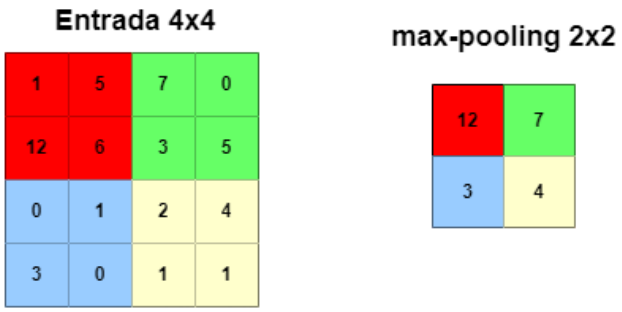
2.6.3 Camadas de agrupamento

A camada de agrupamento, do inglês, *pooling layer*, tem como objetivo reduzir a quantidade de parâmetros da rede através da compressão dos dados de saída da camada de convolução anterior a ela. Estas camadas de agrupamento costumam vir logo após uma camada de convolução para que seja reduzida a complexidade computacional.

Nessa categoria, existem várias opções de técnicas a serem aplicadas para a realização do agrupamento. A aplicação da função de máxima, ou *max-pooling* é a mais popular e consegue abranger as exigências específicas dos sistemas de reconhecimento de fala.

A Figura 11 exemplifica o agrupamento feito através da função *max-pooling*. Foi aplicado um filtro de tamanho 2x2 e um passo de 2. Dessa forma, ele atua nos dados de entrada da camada anterior e gera o número máximo em cada sub-região em torno da qual o filtro faz a convolução, resultando em uma redução do mapa de características em 75%. Com isso, atende-se ao primeiro propósito principal de uma rede convolucional, que é reduzir o custo computacional.

Figura 11 – Funcionamento de uma camada de agrupamento com *max-pooling*.



Fonte: Do autor.

Além deste, a camada de agrupamento com *max-pooling* atende o segundo propósito de uma rede que é evitar o *overfitting*. Esse termo se refere a quando um modelo é tão ajustado aos exemplos de treinamento que não é capaz de generalizar os conjuntos de validação e teste. Uma rede sofrendo de *overfitting* não consegue perceber as variações e só consegue trabalhar com os dados perfeitos e idênticos aos do treinamento, o que não é desejado para um sistema de reconhecimento de fala que possui variações.

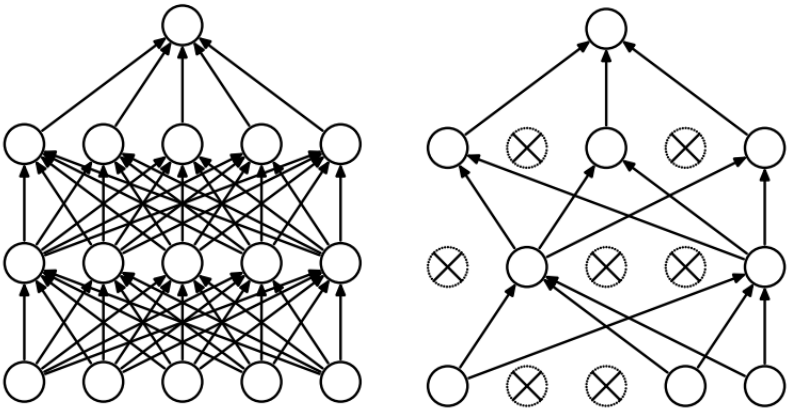
2.6.4 Camadas de abandono

As camadas de abandono, ou camadas de *dropout*, têm uma função muito específica em redes neurais e afetam significativamente o desempenho da rede nos sistemas de reconhecimento de fala. Na última seção, foi apresentado o problema de *overfitting*, onde, após o treinamento, ao se aplicar o modelo aos novos dados de teste, obtém-se um desempenho insatisfatório.

Esta camada desativa aleatoriamente elementos individuais (neurônios) dentro dos mapas de ativação na camada que a antecede, definindo-os como zero. Com isso, a rede é forçada a fornecer a classificação ou saída correta para um exemplo específico, mesmo que alguns dos neurônios sejam descartados. Isso garante que a rede não esteja ficando muito presa aos dados de treinamento e, portanto, ajuda a aliviar o problema de *overfitting* (HINTON et al., 2014).

O termo *dropout* refere-se a uma técnica de regularização utilizada em redes neurais, cujo objetivo é reduzir o sobreajuste (*overfitting*) e aumentar a capacidade de generalização do modelo. Ele consiste na desativação temporária de neurônios durante o processo de treinamento, feita de forma aleatória a partir de uma probabilidade fixa p. Quando um neurônio é desativado, ele deixa de contribuir com o cálculo da saída e tem todas as suas conexões de entrada e saída momentaneamente ignoradas. A Figura 12(a) ilustra uma rede neural completa, com todos os neurônios ativos, enquanto a Figura 12(b) apresenta a mesma rede com uma fração de neurônios desativados, considerando $p = 0{,}5$.

Figura 12 – Camada de abandono



(a) Rede neural completa (b) Rede neural após aplicação do *dropout*

Fonte: Retirado de Hinton(2014)

É importante destacar que o dropout é aplicado exclusivamente durante o treinamento. A cada iteração, uma fração definida dos neurônios (por exemplo, 20% ou 50%) é removida, o que força a rede a não depender excessivamente de conexõs ou unidades específicas. Já durante a fase de inferência (teste ou uso prático do modelo), todos os neurônios permanecem ativos, e os valores de saída são ajustados de forma a compensar a ausência do dropout no treinamento, preservando a coerência estatística.

2.6.5 Camadas totalmente conectadas

Esta é a última camada de uma rede neural convolucional(CNN) e ela atua como a principal camada na classificação devido à junção de todas as características anteriores a ela.

Foi visto anteriormente que os filtros na primeira camada convolucional são projetados para identificar características de baixo nível, por exemplo, reconhecer um sinal de áudio. Porém, para a rede determinar e reconhecer uma palavra através do sinal de fala, a rede precisa ser capaz de reconhecer características de nível mais alto, como formantes, frequência fundamental (*pitch*) e duração dos sons, por exemplo.

Quando passa por outra camada convolucional, a saída da primeira camada convolucional se torna a entrada da segunda camada convolucional. Essa entrada é dada pelos mapas de características de baixo nível que resultam da primeira camada. Na segunda camada convolucional, quando é aplicado um novo conjunto de filtros em cima dessa entrada, a saída será um mapa de características que representa recursos de nível mais alto. Ao avançar pela rede, passa-se por mais camadas convolucionais, obtendo mapas de características que representam recursos cada vez mais complexos.

Após a aplicação de camadas de convolução, seguidas por camadas de ReLU, agrupamento e abandono, têm-se os mapas de características que representam recursos de alto nível. E, para finalizar a rede, é adicionada uma camada totalmente conectada formada por neurônios individuais, e cada um deles conecta-se a todas as características da camada anterior.

Cada neurônio em uma camada totalmente conectada é um nó individual que recebe como entrada todas as ativações da camada anterior. Se a camada anterior tem, por exemplo, 64 características, e a primeira camada densa possui 32 neurônios, então cada neurônio da FC1 tem 64 pesos (um para cada entrada). No total, a FC1 teria $32 \times (64 + 1) = 2080$ parâmetros treináveis.

Em redes classificadoras, a última camada totalmente conectada utiliza a saída das camadas anteriores como entrada e gera um vetor dimensional N como saída, sendo N o número de classes que a rede tem para classificar.

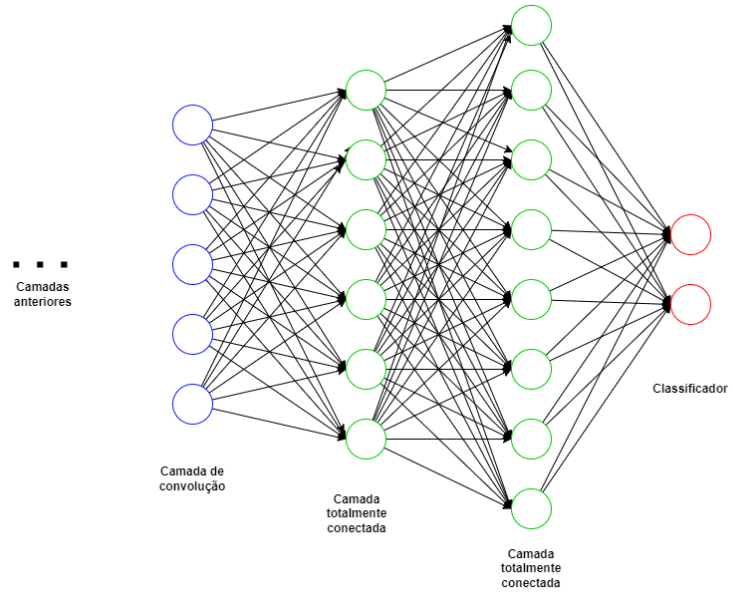
Em Munir, Kong e Qureshi (2023) apresentam um programa de classificação de dígitos através de imagem, N seria 10, pois existem 10 dígitos. Cada número nesse vetor dimensional N representa a probabilidade de uma determinada classe. Por exemplo, se o vetor resultante para um programa de classificação de dígitos for

$$[0,0.1,0.1,0.75,0,0,0,0,0,0.05], \tag{2.5}$$

isso representa uma probabilidade de 10% de que a imagem seja o número 1, uma probabilidade de 10% de que a imagem seja o número 2, uma probabilidade de 75% de que a imagem seja o número 3 e uma probabilidade de 5% de que a imagem seja o número 9.

A Figura 13 mostra duas camadas totalmente conectadas e suas conexões ao final de uma rede neural convolucional. Estas duas camadas possuem, respectivamente, 6 neurônios na primeira camada e 8 neurônios na segunda.

Figura 13 – Conexões de duas camadas totalmente conectadas



Fonte: Do autor.

As conexões da camada totalmente conectada são feitas uma a uma para cada neurônio dos mapas de características da camada anterior. Ao final, a camada totalmente conectada se resume aos neurônios de classificação. Na Figura 13 tem-se a classificação de apenas 2 classes.

2.7 Taxa de erro de palavras

A Taxa de Erro de Palavras (WER, do inglês *Word Error Rate*) é uma métrica de avaliação comumente utilizada em tarefas de reconhecimento de fala e sistemas de processamento de linguagem natural. Ela é empregada para definir a precisão entre a transcrição automática de um sistema e a transcrição de referência, medida em termos da taxa de erro de palavras.A fórmula básica para calcular a WER é dada por (2.6).

$$\mathrm{WER} = \frac{\mathrm{S} + \mathrm{D} + \mathrm{I}}{N}, \tag{2.6}$$

sendo S o número de palavras substituídas, D o número de palavras deletadas, I o número de palavras inseridas e N o número total de palavras na transcrição de referência.

A métrica WER considera a soma dos três tipos de erros, sendo eles a quantidade de Substituições (S), a quantidade de Deleções (D) e a quantidade de inserções (I). Em performance de sistemas de reconhecimento de fala, a acurácia de palavras dada por (WAcc) é mais comumente utilizada e ela é dada por (2.7).

$$W_{Acc} = 1 - \mathrm{WER} = 1 - \frac{\mathrm{S} + \mathrm{D} + \mathrm{I}}{N}, \tag{2.7}$$

Para sistemas onde não se tem a remoção e a inserção de palavras, a taxa de assertividade é dada pelo número de acertos dividido pelo número de dados de teste. Reescrevendo a equação (2.7), pode-se definir WAcc como

$$W_{Acc} = 1 - \frac{S}{N} = \frac{N - S}{N}. \tag{2.8}$$

2.8 Validação Cruzada como método de avaliação

A validação cruzada K-*Fold* é uma técnica usada para avaliar o desempenho de modelos de aprendizado de máquina, dividindo os dados em k subconjuntos.

Essa técnica envolve particionar o conjunto de dados em k subconjuntos, treinar o modelo em k-1 subconjuntos e validá-lo no subconjunto restante, iterando esse processo k vezes para garantir que cada subconjunto sirva como conjunto de validação uma vez.

Uma separação comumente utilizada e que será utilizada neste trabalho são de 70% dos dados para o treinamento e 30% para os testes. Neste trabalho, os dados foram divididos entre subconjunto de treinamento e teste, de forma aleatória. Este processo foi repetido iterativamente, de modo que, ao término de cada iteração, foi possível avaliar o desempenho do sistema de reconhecimento. Ao final, pode-se obter o desempenho médio, a partir dos valores encontrados em cada iteração.

3 Desenvolvimento do trabalho

3.1 Introdução

A partir das técnicas descritas no capítulo anterior, foi possível implementar um sistema de reconhecimento de comandos de fala baseado em HMM e um outro baseado em CNN. Sendo assim, esse capítulo expõe os métodos aplicados ao longo do desenvolvimento do trabalho com a finalidade de descrever as técnicas e detalhar os parâmetros aplicados a elas.

Para o trabalho atual foram utilizadas duas bases de palavras com locutor único, sendo a primeira focada no reconhecimento de palavra isolada e a segunda com foco no reconhecimento de frases de comando. Cada base é composta de 150 gravações de um único locutor, sendo 30 repetições de cada classe das seguintes bases:

- Base 1 - Palavras Isoladas
  - Sala;
  - Cozinha;
  - Quarto;
  - Acender;
  - Apagar.
- Base 2 - Frases de Comando
  - Acender luz quarto;
  - Apagar luz quarto;
  - Ligar ar quarto;
  - Desligar ar quarto;
  - Desligar TV Sala.

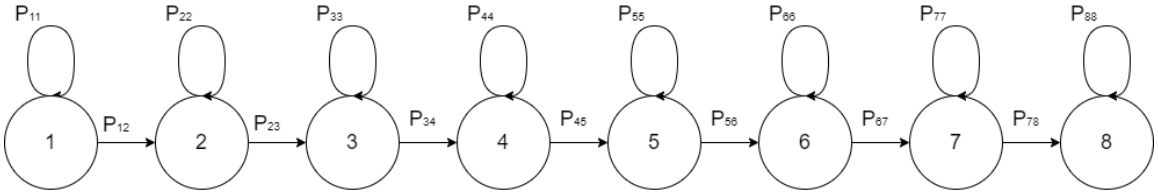
Cada uma destas bases foi utilizada separadamente aos dois modelos propostos. A primeira parte deste capítulo detalha a aplicação ao modelo oculto de Markov (HMM) considerando as etapas de pré-processamento, treinamento e testes. A segunda etapa do trabalho foi feita aplicando as mesmas bases em uma rede neural convolucional (CNN) projetada especificamente para um sistema de reconhecimento de fala.

Ambas as técnicas utilizam a extração de características dos sinais de fala e realizam a classificação dos mesmos a fim de reconhecer a palavra pronunciada. O reconhecimento acontece de forma separada em cada um dos modelos e, ao final, é feita a comparação de eficácia dos modelos em cada experimento.

3.2 Modelo oculto de Markov

A Figura 14 detalha a estrutura do HMM utilizado neste trabalho modelada com 8 estados. Essa estrutura de 8 estados foi definida com base nos experimentos para atingir o objetivo deste trabalho. Foi iniciado em 5 mas os melhores desempenhos encontrados foram com 8 estados.

Figura 14 – Estrutura HMM do presente trabalho.



Fonte: Do autor.

Para o presente trabalho, foi construído um modelo de reconhecimento de fala baseado no HMM através do *software Matlab*. A ferramenta, juntamente com a *toolbox* de *Machine Learning*, presente no *software*, simplificando os processos de treinamento e comparação dos modelos baseados em aprendizado de máquina.

No presente trabalho, o treinamento é executado utilizando o algoritmo Baum-Welch, que ajusta iterativamente os parâmetros do modelo baseado na maximização da verossimilhança (Maximum-Likelihood Estimation).

3.2.1 Pré-processamento

Os sinais gravados foram pré-processados com o objetivo de reduzir as perturbações e ressaltar as informações úteis, pois até mesmo os melhores sistemas de reconhecimento sofrem substancial degradação de seu desempenho quando trabalham com sinais de fala corrompidos por ruídos.

Para o presente trabalho, os sinais de áudio foram adquiridos através do microfone de um computador, o qual não garante perfeição e remoção dos ruídos. Estas aquisições foram feitas utilizando o Matlab e, sob elas, foram aplicados filtros passa-altas a fim de remover os ruídos de baixa frequência. Além destes filtros, todos os sinais foram submetidos a uma normalização do sinal de áudio para que todos fossem processados em mesma amplitude, reduzindo dessa forma as variações resultantes de diferentes volumes de gravação do microfone.

Por fim, foi realizado o recorte de silêncio dos sinais de áudio gravados que se encontram antes e após o sinal acústico. O custo computacional da máquina a realizar o processamento também é reduzido com o corte de silêncio.

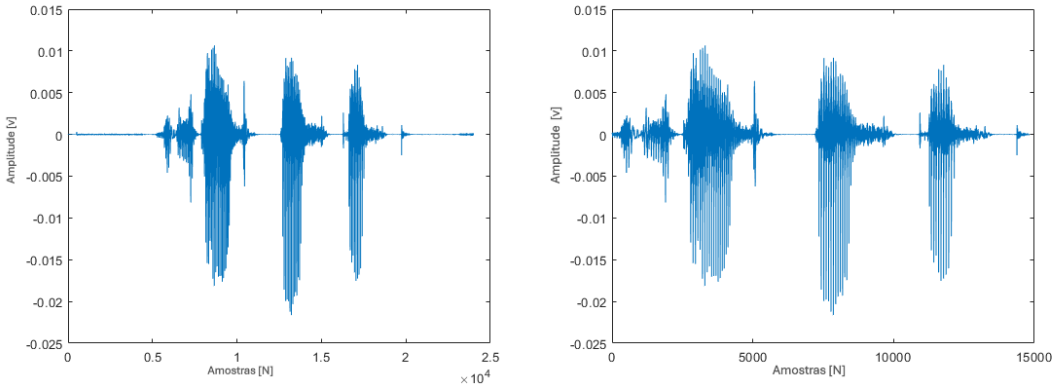
O recorte de silêncio tem como base a função de energia de um sinal de áudio onde ela é calculada utilizando uma janela deslizante que se move ao longo do sinal. Para cada

posição da janela, a energia é calculada somando-se os quadrados dosvalores das amostras dentro da janela. Em seguida, um limiar é aplicado a essa energia para determinar se a região é considerada como silêncio ou não.

Para o presente trabalho, foram identificadas regiões do sinal onde a energia, ou a magnitude ao quadrado do sinal, é menor que um determinado limiar fixo. Essas regiões identificadas como "silêncio"foram removidas, resultando em um sinal de áudio com menos ruído de fundo. O melhor resultado encontrado foi utilizando-se um limiar fixo de corte de ruído com amplitude de 0.29614.

Na Figura 15 observa-se o sinal de áudio de um comando antes e após a etapa de pré-processamento. Pode ser observada a remoção de períodos de silêncio ou ruído indesejado do sinal captado.

Figura 15 – Recorte de silêncio aplicado durante a etapa de pré processamento do HMM sedo (a) Sem recorte de silêncio e (b) Com recorte de silêncio.



(a) Sem recorte de silêncio. (b) Com recorte de silêncio.

Fonte: Do autor.

3.2.2 Extração de características

A função desta etapa é criar uma representação do sinal de fala através de de um conjunto de características representativas de cada segmento do sinal contido em uma janela temporal.

Para a tratativa de cada sinal de áudio, o sinal de entrada foi dividido em quadros de 160 amostras, considerando-se uma frequência de amostragem de 8 kHz, com uma duração de 20 ms, sendo os quadros adjacentes separados por 80 amostras. Esta divisão foi de 20 ms para que o quadro em análise seja considerado um sinal quase estacionário,

Conforme visto na seção 2.2, o janelamento é crucial na etapa de extração de características. Neste trabalho, utilizou-se uma janela de Hamming com o intuito de reduzir a distorção espectral associada ao efeito do janelamento dos dados.

A obtenção dos MFCCs é o método mais difundido para extração de características e é utilizado para seleção das informações dinâmicas.

No presente trabalho, foi utilizado um vetor de parâmetros com dimensão de 39 coeficientes mel. Esses coeficientes capturam informações relevantes sobre as características espectrais da fala. A divisão destes 39 coeficientes e a forma como foram obtidos neste trabalho são apresentadas abaixo:

- 12 Parâmetros Mel-Cepstrais (MFCC) - Representam as amplitudes das componentes espectrais do sinal de áudio e foram calculadas na escala mel através da utilização de um banco de 18 filtros passa-faixa triangulares e cálculo logaritmo da saída deste banco de banco de filtros. Por fim, aplica-se a DCT sobre o vetor contendo os 18 valores de energia correspondentes a cada filtro do banco, com o intuito de se comprimir esta informação para apenas 12 parâmetros.
- Derivada Primeira (Delta-Mel-Cepstrais) - Capturam a taxa de variação dos parâmetros mel-cepstrais ao longo do tempo, fornecendo informações dinâmicas.
- Derivada Segunda (Delta-Delta-Mel-Cepstrais) - Representam a taxa de variação das derivadas primeiras, oferecendo uma segunda ordem de dinâmica nas características espectrais
- 1 Parâmetro de Energia - Reflete a quantidade total de energia no sinal de áudio, em cada quadro.
- 1 Derivada Primeira (Delta-Energia) - Indica a variação na energia do sinal ao longo do tempo.
- 1 Derivada Segunda (Delta-Delta-Energia) - Representa a taxa de variação da derivada primeira da energia, adicionando uma dimensão extra de dinâmica.

Os coeficientes MFCCs e suas respectivas derivadas de primeira e segunda ordem assim como o parâmetro de energia e suas respectivas derivadas, foram obtidos a partir de cada janela. Dessa forma, foi gerado um vetor de saída composto de 39 parâmetros para cada janela de 20 ms do sinal.

3.2.3 Treinamento do HMM

Conforme descrito no início deste capítulo, o sistema de reconhecimento de fala dependente de locutor em aplicação foi inicializado com gravação de 30 sinais de áudio de cada uma das 5 palavras em cada base. Destes 30 sinais de áudio, 22 foram separados para a etapa de treinamento e 8 foram para a etapa de teste, que será descrita na próxima seção.

A metodologia para treinamento do modelo oculto de Markov (HMM) utilizada envolveu um processo iterativo que utiliza o algoritmo BWA, também conhecido como

-*Expectation-Maximization*, para ajustar os parâmetros do modelo com base nos dados observados.

Os parâmetros do modelo HMM foram inicializados e foram ajustados ao longo da etapa de treinamento. Estes incluem probabilidades iniciais, matriz de transição de estados e distribuições de probabilidade de emissão formando cada estado de Markov.

O treinamento ocorreu de forma iterativa, aplicando-se o algoritmo de Expectation-Maximization (EM), apresentado no Capítulo 2, a fim de chegar a uma condição de convergência onde se tenha o maior número de iterações ou uma pequena variação nos parâmetros daquele modelo. Após o treinamento, o modelo resultante foi avaliado usando conjuntos de dados de teste para verificar seu desempenho, e estes serão detalhados na próxima seção.

3.2.4 Etapas de teste e aferição dos resultados

Após o treinamento do HMM, deve-se realizar os testes usando conjuntos de dados distintos dos utilizados no treinamento. O processo de teste envolve avaliar o desempenho do modelo na classificação ou predição de sinais de áudio que não foram utilizados durante a etapa de treinamento. Conforme visto na seção anterior, 8 sinais de áudio de cada palavra foram separados para esta etapa de teste.

Assim como no treinamento, os dados de teste devem passar por um processo de pré-processamento e extração de características, tal como ocorreu com os dados utilizados na etapa de treinamento. No caso deste trabalho, buscou-se realizar o recorte de silêncio e realizar a extração de características conforme descrito na 3.2.2.

Os parâmetros ajustados durante o treinamento foram carregados da etapa de teste. Com o auxílio do algoritmo Viterbi, descrito no Capítulo 2, foi determinada a sequência mais provável de estados dada a sequência de observação durante a etapa de decodificação.

Por fim, para análise de desempenho durante a fase de testes, foram comparadas a sequência de estados identificada pelo sistema com o sinal de áudio original utilizado como entrada nos testes. Com base nos valores de probabilidade produzidos pelos modelos acústicos associados a cada comando de fala, esta etapa da decodificação identifica o modelo que fornece a maior verossimilhança, o qual está associado a uma determinada palavra.

Para realizar a validação dos resultados, foram calculadas as taxas de erro e assertividade do sistema através de métricas de desempenho descritas no Capítulo 2. Através da validação cruzada *5-fold*, o conjunto de dados foi dividido em 5 grupos e foram calculados resultados específicos para cada um destes 5 conjuntos. Esta divisão em 5 *folds* faz com que todas as gravações passem pela etapa de teste.

3.3 Redes neurais convolucionais

As redes neurais convolucionais (CNNs) avançaram significativamente no campo do reconhecimento de fala, aprimorando a capacidade de aprender características complexas dos sinais de fala. Esses avanços foram aplicados a vários aspectos do reconhecimento de fala, incluindo reconhecimento automático de fala, análise de dependências temporais na fala e interpretação de dados de fala complexos, para melhorar a precisão e a eficiência.

Os ASRs baseados em CNN necessitam de uma etapa de pré-processamento do sinal falado acompanhada da etapa de extração de características a fim de preparar a base de dados para que a rede neural seja capaz de realizar as etapas de treinamento e a etapa de testes.

Para o presente trabalho foi construído um modelo de reconhecimento de fala baseado no CNN através do ambiente de código aberto *Google Colab*. A ferramenta, juntamente com as bibliotecas disponíveis, em específico a biblioteca *PyTorch*, escrita na linguagem de *Python*, permite a construção de modelos de aprendizado profundo que exigem um maior poder computacional. Devido ao aumento dessa complexidade de processamento, os experimentos das CNNs foram realizados em *Python* e nao no *Matlab*.

3.3.1 Pré-processamento

Os sinais gravados passaram por pré-processamento com o objetivo de reduzir as perturbações e ressaltar as informações úteis. Assim como no modelo anterior, a primeira etapa é vista como um tratamento da base de dados a ser utilizada a fim de neutralizar os erros dos sinais de fala corrompidos por ruídos.

Para o presente trabalho, os sinais de áudio foram adquiridos através do microfone de um computador, o qual não garante perfeição e remoção dos ruídos, e foi aplicada a técnica de remoção de silêncio. O recorte de silêncio tem como base a função de energia de um sinal de áudio onde ela é calculada para cada janela do sinal. Para cada posição da janela, a energia é calculada somando-se os quadrados dos valores das amostras dentro da janela. Em seguida, o limiar é aplicado a essa energia para determinar se a região é considerada como silêncio ou não.

A esta rede neural, durante alguns experimentos, foi aplicada a técnica de distensão temporal com o objetivo de aumentar a base de dados e analisar o desempenho do sistema de reconhecimento de fala com uma variação do sinal de entrada. Esta técnica não altera a informação contida no áudio.

A distensão temporal é uma técnica usada em vários campos, do processamento de áudio à imagem óptica, para manipular as características temporais de sinais ou dados. Esse processo envolve estender ou comprimir o tempo de duração de um sinal sem alterar seu tom ou outras características essenciais. Para o presente trabalho, foi aplicado um

fator de variação aleatório nos testes realizados entre 0.8 e 1.2 de forma uniforme em todo o sinal de entrada.

Por fim, a técnica de adição de zeros ao sinal de áudio foi aplicada a todos os sinais para garantir que todos tenham o mesmo tamanho antes de serem tratados.

3.3.2 Extração de características

Os sinais de áudio foram convertidos para representações espectrais adequadas à entrada da rede convolucional onde foram utilizadas duas representações complementares: o espectrograma Mel (*Mel spectrogram*) e os Coeficientes Mel-Cepstrais (MFCC, do inglês *Mel-frequency cepstral coefficients*). Ambas as transformações foram aplicadas a cada sinal de áudio e, em seguida, concatenadas ao longo da dimensão da frequência, formando a matriz final de entrada da CNN.

O espectrograma Mel é uma representação do espectro de frequências em uma escala perceptiva baseada na forma como o ouvido humano percebe o som. Nesta implementação, foram utilizadas 80 bandas Mel, definidas de forma empírica, resultando em uma matriz de dimensões (80, T), onde T representa o número de quadros temporais. Cada elemento dessa matriz representa a energia do sinal de áudio em determinada banda de frequência em instante de tempo.

Os MFCCs são extraídos a partir do espectrograma Mel por meio de operações de logaritmo e Transformada Discreta de Cosseno (DCT), com o objetivo de obter uma representação mais compacta do envelope espectral do sinal. Foram extraídos 12 coeficientes para cada quadro temporal, gerando uma matriz de dimensão (12, T).

As duas matrizes (espectrograma Mel e MFCC) foram então concatenadas ao longo da dimensão de frequência, formando um único tensor de dimensão (92, T) por amostra de áudio. Dessa forma, os 92 canais de entrada da CNN correspondem à combinação de 80 bandas de frequência na escala Mel (valores de energia) e 12 coeficientes cepstrais (MFCCs) por quadro temporal.

A dimensão temporal representa a evolução do sinal ao longo de janelas sucessivas. Neste trabalho, o número de quadros temporais resultantes foi 241, após normalização da duração do áudio por *zero-padding* e truncamento.

Essa representação combinada permite que a rede convolucional capture tanto informações espectrais detalhadas (via espectrograma Mel), quanto padrões acústicos mais globais e robustos (via MFCCs). A CNN é, então, capaz de aprender a partir desses 92 vetores características ao longo do tempo, promovendo uma classificação eficiente dos sinais de fala.

É essa representação que será usada como entrada para a rede neural e, é dada por um tensor. A primeira dimensão é uma dimensão unitária para ser tratado apenas um sinal de áudio por vez. A segunda dimensão é composta pelas 92 características extraídas e a terceira é a dimensão temporal definida pelo tamanho total de 241 amostras de cada

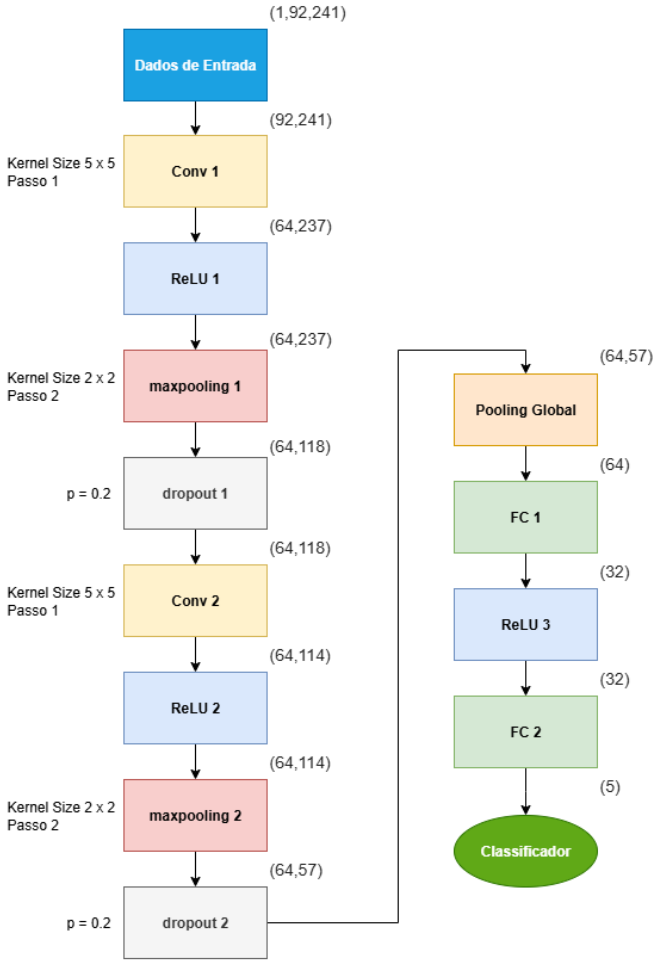
sinal. Os dados de entrada da CNN do presente trabalho foram definidos como um tensor de medidas 1 x 92 x 241.

3.3.3 Estrutura da CNN

Para o presente trabalho, tem-se o objetivo de fazer o reconhecimento de um sinal de áudio através da classificação do mesmo entre as classes pré-determinadas.

Para atingir o objetivo, a rede neural convolucional foi construída utilizando 13 camadas, definidas a partir de experimentos e ela pode ser visto na Figura 16.

Figura 16 – Estrutura da rede neural convolucional do presente trabalho.



Fonte: Do autor.

O sistema foi desenvolvido em Python, utilizando principalmente a biblioteca PyTorch, que oferece recursos completos para criação, treinamento e validação de redes neurais. Também foram empregadas bibliotecas complementares como torchaudio, voltada ao processamento e análise de dados de áudio, *scikit-learn* para avaliação de desempenho do modelo e *matplotlib* para geração de gráficos e visualizações. Essas bibliotecas foram escolhidas por serem amplamente utilizadas em pesquisa e desenvolvimento de soluções

com inteligência artificial, possuírem código aberto e documentação acessível, permitindo total transparência e reprodutibilidade dos resultados apresentados neste trabalho.

A rede convolucional implementada neste trabalho segue uma arquitetura sequencial, composta por camadas convolucionais, funções de ativação, camadas de agrupamento (*pooling*), regularização via abandono (*dropout*) e camadas totalmente conectadas (*fully conected*). A entrada de cada amostra na rede possui formato (1,92,241), onde:

- o valor 1 refere-se ao canal da amostra (mono),
- o valor 92 representa os canais de características, compostos por 80 bandas espectrais da transformada de Mel e 12 coeficientes cepstrais (MFCCs),
- e o valor 241 corresponde ao número de quadros temporais extraídos do sinal de áudio.

Antes de ser processada pela primeira camada convolucional, a entrada é reduzida para (92, 241) por meio da operação *.squeeze(1)*, eliminando a dimensão adicional do canal. A operação *.squeeze(1)* em Python, é utilizada na bibliotecas PyTorch, ela é usada para remover uma dimensão de tamanho 1 de um array ou tensor ao longo de um eixo específico. O 1 em squeeze(1) indica que a operação tem como alvo a segunda dimensão que é indicada pelo índice 1.

Em uma CNN, a dimensão de profundidade indica quantos mapas de características (*feature maps*) estão sendo processados em determinada camada. A camada `Conv1` recebe como entrada os 92 mapas de características e aplica 64 filtros convolucionais (ou neurônios), cada um capaz de convoluir toda a profundidade de entrada. Como resultado, essa camada gera 64 novos mapas de saída, cada um representando um padrão espectro-temporal aprendido pela rede. Essa transformação permite que a CNN extraia combinações entre bandas Mel e coeficientes cepstrais na dimensão de profundidade, enfatizando os aspectos mais relevantes para a tarefa de classificação, sem perda de informação.

A dimensão temporal da saída da primeira convolução é calculada pela equação abaixo:

$$\text{Saída} = \left(\frac{(W - F)}{S} + 1\right) = \left(\frac{(241 - 5)}{1} + 1\right) = 237 \tag{3.1}$$

Considerando o tamanho da entrada W=241, o tamanho do filtro F=5 e o passo (stride) S=1, a saída da `Conv1` apresenta dimensões (64,237). A essa camada é aplicada uma função de ativação `ReLU`, que introduz não linearidade ao modelo, seguida por uma camada de agrupamento máximo (`MaxPooling1D`) com filtro de tamanho 2×2 e passo 2. Essa etapa reduz pela metade a dimensão temporal, resultando em uma saída de (64,118), além de contribuir para a redução computacional e mitigação de *overfitting*. Em seguida, uma camada de abandono (`Dropout`) com taxa de 20% é aplicada, sem alteração nas dimensões.

Na segunda etapa da rede, aplica-se novamente uma convolução (`Conv2`) com os mesmos parâmetros da anterior. Porém, ao contrário da primeira camada convolucional, que opera diretamente sobre a entrada bruta, essa convolução atua sobre os mapas de ativação produzidos anteriormente, extraindo padrões mais abstratos e específicos. A entrada dessa camada possui dimensões (64,118), sendo reduzida para (64,114) após a aplicação dos filtros 5x5. Em seguida, repetem-se as camadas `ReLU`, `MaxPooling` e `Dropout`, resultando em uma saída com 64 mapas de características e 57 quadros temporais (64,57).

Posteriormente, é aplicada uma operação de pooling global (`GlobalAveragePooling1D`), que reduz a dimensão temporal ao calcular a média ao longo do tempo para cada mapa de característica. O resultado é um vetor unidimensional com 64 valores, representando a consolidação final dos padrões aprendidos ao longo do espectro e do tempo.

Esse vetor é transferido para a primeira camada totalmente conectada (`FC1`), composta por 32 neurônios. Cada neurônio nesta camada está completamente conectado a todos os 64 elementos da saída anterior, o que permite ao modelo associar combinações específicas de características extraídas para diferentes padrões de fala. Em seguida, aplica-se uma função de ativação `ReLU`, resultando em um vetor de ativação com 32 valores.

A etapa final do processo de classificação ocorre na camada totalmente conectada `FC2`, que recebe a saída dos 32 neurônios da camada FC1 e se conecta aos 5 neurônios da camada FC2. É gerado um novo vetor com cinco valores, cada um correspondente a uma das classes previstas pelo modelo. Esses valores indicam a ativação da rede em relação a cada classe possível. Ao final do processamento, a rede neural escolhe a classe associada ao valor mais alto e a define como a predição para a amostra de áudio analisada.

A Tabela 1 resume as entradas e saídas da rede CNN projetada no presente trabalho.

Etapa	Camada	Tipo	Entrada	Saída	Observações
1	Entrada	Pré-processamento	(1, 92, 241)	(92, 241)	Canal removido com .squeeze(1)
2	Conv1	Convolucional 1D	(92, 241)	(64, 237)	64 filtros, kernel=5, stride=1, sem padding
3	ReLU1	Ativação	(64, 237)	(64, 237)	Função não linear aplicada ponto a ponto
4	MaxPool1	Pooling 1D	(64, 237)	(64, 118)	Kernel=2, stride=2, reduz temporalidade pela metade
5	Dropout1	Regularização	(64, 118)	(64, 118)	Dropout com p = 0,2 durante o treino
6	Conv2	Convolucional 1D	(64, 118)	(64, 114)	Novo kernel=5, stride=1, sem padding
7	ReLU2	Ativação	(64, 114)	(64, 114)	Função ReLU aplicada novamente
8	MaxPool2	Pooling 1D	(64, 114)	(64, 57)	Redução temporal adicional
9	Dropout2	Regularização	(64, 57)	(64, 57)	Dropout adicional com p = 0,2
10	Pooling Global	Redução com média	(64, 57)	64	Média ao longo da dimensão temporal
11	FC1	Camada densa	64	32	64 → 32 neurônios totalmente conectados
12	ReLU3	Ativação	32	32	Ativação aplicada antes da saída final
13	FC2	Camada de saída	32	5	5 neurônios para classificação de 5 classes

Fonte: Do autor.

Tabela 1 – Estrutura da rede projetada

3.3.4 Treinamento da rede convolucional

Assim como no modelo oculto de Markov, o modelo de rede neural convolucional foi aplicado em cima de uma divisão de *k-folds* para ser feita a validação cruzada ao fim do teste. Com isso, temos a certeza de que toda a base de dados foi avaliada e sem a repetição dos mesmos sinais de áudios utilizados no treinamento para o teste. Para tal, as *folds* e números de amostras foram divididas da seguinte forma:

- Base I - Palavras
  - *Fold 1*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 2*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 3*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 4*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 5*: Amostras de treino: 110, Amostras de teste: 40
- Base II - Frases
  - *Fold 1*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 2*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 3*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 4*: Amostras de treino: 110, Amostras de teste: 40
  - *Fold 5*: Amostras de treino: 110, Amostras de teste: 40

No aprendizado profundo, uma época do inglês, *epoch* é uma passagem completa de todo o conjunto de dados de treinamento por meio de um algoritmo de aprendizagem. O número de épocas é um hiperparâmetro que determina quantas vezes o modelo passará por todos os dados de treinamento. O número de épocas para o presente trabalho utilizado foi 15 e a cada época o modelo fez o treinamento em cima das 150 amostras de treino.

Uma época é composta de lotes de dados, também conhecidos como *batch*. O tamanho do lote é definido pelo número de amostras que são aplicadas à rede neural de uma só vez e o indicador *batch/s* é definido como a quantidade de amostras que o modelo, projetado no presente trabalho, conseguiu analisar em 1 segundo.

Durante o treinamento, os pesos e *bias* aplicados aos filtros são atualizados por meio de retropropagação, permitindo que a rede aprenda os recursos mais relevantes para a tarefa em questão. A cada época, a acurácia do modelo ficará mais precisa devido aos ajustes de pesos provenientes dessa retropropagação e o resultado final de acurácia é definido após a última época ter sido executada.

3.3.5 Etapas de teste e aferição dos resultados

Cada base de dados foi dividida em 5 *folds* onde cada *fold* abrange um certo número de amostras para testes e estas não serão utilizadas novamente como dados de teste na *fold* seguinte.

Após o treinamento, durante a fase de testes, os dados separados para testes são classificados entre as 5 classes pré-definidas de cada uma das bases. A classificação assertiva significa que o neurônio da camada de classificação com o maior valor probabilístico é o neurônio que classifica a entrada corretamente.

Pode-se citar um exemplo utilizando o classificador do presente trabalho. Dada uma entrada da classe "Apagar", se um neurônio classificador final mais ativo for da classe "Apagar", a rede tem sucesso na classificação. Mas se o neurônio classificador ativo para a mesma entrada for o neurônio que classifica a palavra "Quarto", a rede não tem sucesso na classificação.

Para cada *fold* específica, a rede neural convolucional projetada no presente trabalho teve 15 épocas de treinamento e testes executadas a fim de melhorar a acurácia do modelo. Ao final das 15 épocas, o modelo retorna a acurácia final que foi melhorada para cada época. Ao final das 5 *folds*, foi feita a média de acurácia de cada fold para ser definido a acurácia total do modelo.

Conforme visto no seção 2.7, acurácia de cada fold e, para cada modelo, foi calculada com base na taxa de erro e o desempenho de cada modelo foi definido com base nessa taxa. O modelo com zero erros, tem uma taxa de acurácia de 100% e o melhor desempenho no reconhecimento de fala.

4 Resultados

4.1 Modelo Oculto de Markov - HMM

No presente trabalho, as classes de palavras (Base I) e de frases(Base II) tiveram suas características extraídas conforme descrito na subseção 3.2.2. O formato da extração de característica não varia em função do modelo. Essas informações são usadas em conjunto com o HMM para realizar a tarefa de classificação e, consequentemente, o reconhecimento do sinal de fala.

Nesta seção serão descritos os resultados encontrados para os diferentes experimentos realizados com base no HMM. Foram variados o limiar do recorte de silêncio e o número de gaussianas a fim de encontrar o modelo com maior índice de acertos.

4.1.1 Limiar do recorte do silêncio

Como descrito no Capítulo 2, o recorte do silêncio é uma etapa importante para o modelo oculto de Markov(HMM) projetado no presente trabalho.

O recorte de silêncio foi feito com o auxílio da função de energia do sinal e sob esta, foi aplicado um limiar de valor fixo durante toda a amostra do sinal para remover o silêncio. Esse limiar foi determinado de forma empírica e todo dado do sinal que ficou abaixo desse limiar foi removido antes da extração de características.

Para o experimento de variação do limiar de recorte do silêncio, o modelo foi inicializado com um valor alto do limiar de energia no recorte de silêncio a fim de mensurar as informações contidas no sinal de áudio de entrada. Posteriormente, de maneira empírica, foi diminuído este valor até que o sistema encontrasse a melhor taxa de acerto. A partir do momento em que o sistema começou a cometer mais erros com a diminuição do limiar, significa que o limiar do silêncio não estava removendo somente o silêncio e ruídos do sinal. Uma vez que segmentos do sinal acústico associados ao silêncio estão presentes em todas as gravações, as informações neles contidas podem ser incorporadas aos modelos, contribuindo para a ocorrência de erros de classificação ao removê-las.

No presente trabalho, utilizou-se da validação cruzada de 5 *folds* onde se tem toda a varredura dos sinais de áudio da base de dados. Conforme visto no Capítulo 3, para cada palavra ou frase, são utilizados 22 sinais de áudio para o treinamento e os demais 8 dados são utilizados para os testes.

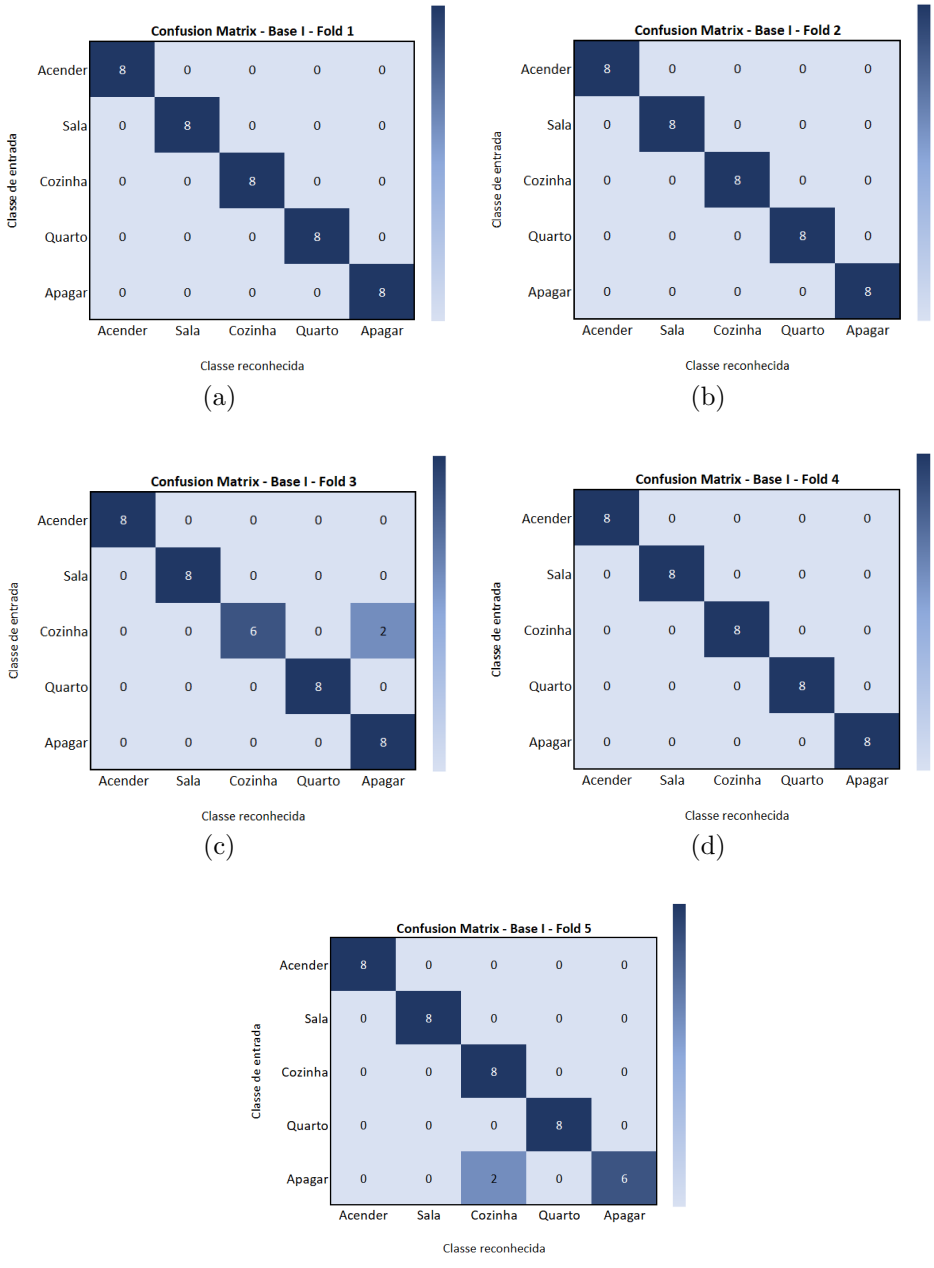
Para seguir com outros experimentos, apenas o parâmetro do limiar de recorte de silêncio foi alterado sendo aumentado a cada experimento de forma empírica. Os demais parâmetros ficaram constantes ao longo deste experimento, a fim de se validar o impacto da variação do limiar de recorte de silêncio.

4.1.1.1 Limiar do recorte do silêncio - Base I

Para o primeiro experimento, foi utilizado um limiar de energia fixo de 2,9614 $\times$ 10^{-02} a fim de recortar o silêncio existente antes e depois da palavra contida na elocução.

A partir dos dados de entrada, o sistema fez a classificação de 40 locuções durante a etapa de testes de cada *fold*, sendo 8 locuções de cada classe. A Figura 17 mostra todas as classificações feitas pelo modelo para cada sinal de entrada. A matriz confusão enumera os erros cometidos pelo modelo ao comparar o sinal de entrada com o sinal predito (sinal de saída).

Figura 17 – Matriz confusão para o experimento do HMM com remoção de silêncio e limiar de recorte 2,9614 × 10^{-02} - Base I - Palavras.



(a) (b) (c) (d) (e)

Fonte: Do autor.

Como pode ser observado nas Figuras 19(c) e 17(e), o sistema fez a confusão na classificação das palavras "Cozinha"e "Apagar"nas *folds* 2 e 5. Um total de 4 erros foi identificado no sistema e a taxa de acerto final do modelo foi de 98%. A Tabela 2 mostra os resultados obtidos para cada *fold* deste experimento com remoção de silêncio aplicada na base de palavras.

Tabela 2 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-02}$ - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de Acerto**
1	40	0	1,000
2	40	0	1,000
3	38	2	0,950
4	40	0	1,000
5	38	2	0,950
Total	**196**	**4**	**0,980**

Fonte: Do autor.

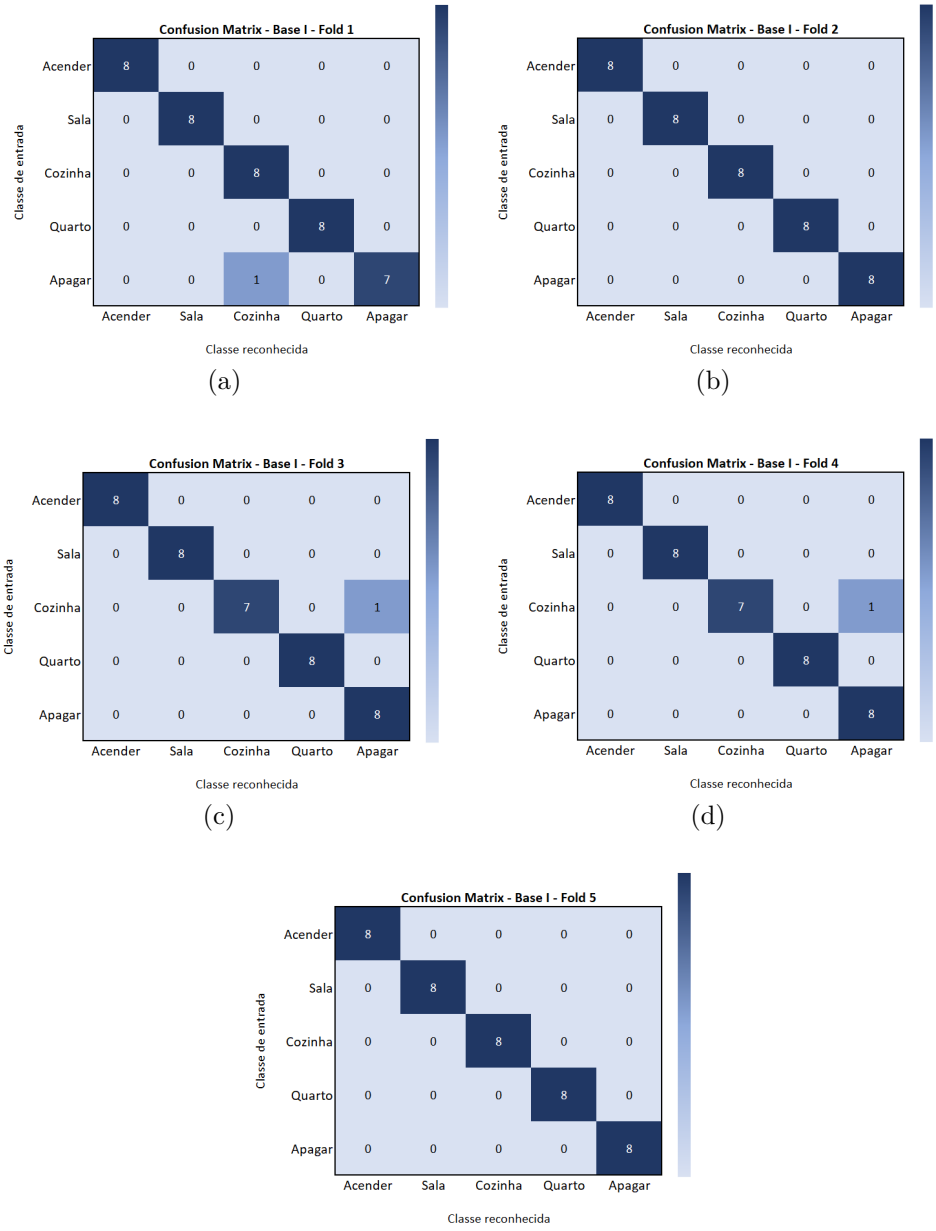
Para os experimentos seguintes, o limiar de recorte de silêncio foi diminuído em 100 vezes o valor do experimento anterior. A Tabela 3 mostra a taxa de acerto do modelo para cada *fold* no segundo experimento com o limiar de recorte sendo $2{,}9614 \times 10^{-04}$. A taxa de acerto final do modelo neste experimento encontrada foi de 98,5%. A Figura 18 detalha a matriz de confusão desse experimento.

Tabela 3 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-04}$ - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	39	1	0,975
2	40	0	1,000
3	39	1	0,975
4	39	1	0,975
5	40	0	1,000
Total	**197**	**3**	**0,985**

Fonte: Do autor.

Figura 18 – Matriz confusão para o experimento do HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-04}$ - Base I - Palavras.



(a) (b) (c) (d) (e)

Fonte: Do autor.

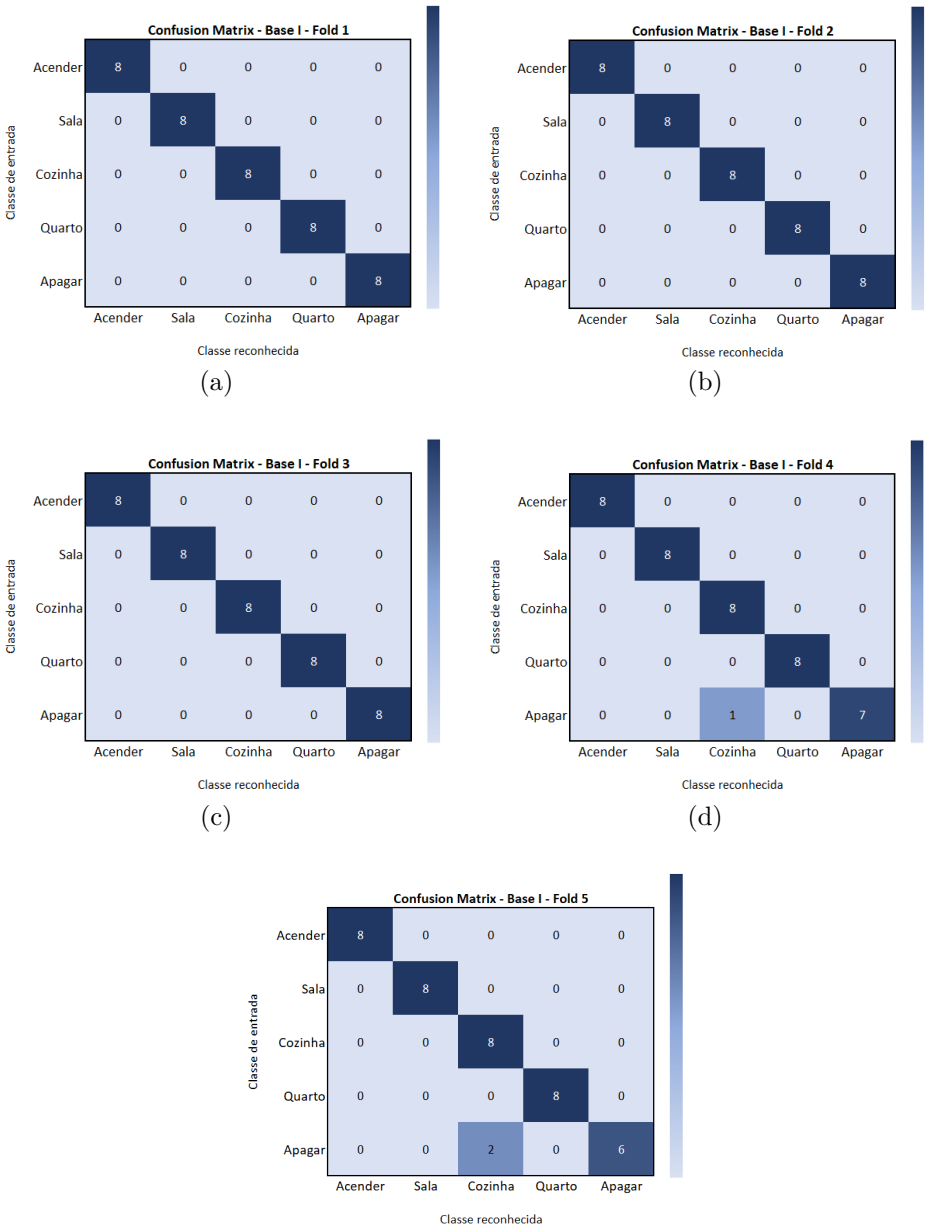
Para o terceiro experimento, tem-se a diminuição do limiar para $2{,}9614 \times 10^{-06}$. A Tabela 4 mostram as taxa de acertos do sistema para o terceiro experimento com este limiar. A taxa de acerto final do modelo neste experimento encontrada foi de 99,5%. A matriz de confusão desse experimento é plotada em sequência na figura Figura 19.

Tabela 4 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-06}$ - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	40	0	1,000
3	40	0	1,000
4	39	1	0,975
5	40	0	1,000
Total	**199**	**1**	**0,995**

Fonte: Do autor.

Figura 19 – Matriz confusão para o experimento do HMM com remoção de silêncio e limiar de recorte 2,9614 × 10^{-06} - Base I - Palavras.



Fonte: Do autor.

A partir do quarto experimento, o limiar se tornou tão baixo que a taxa de acerto do modelo HMM começou a decair bruscamente e notou-se que o limiar não estava removendo o silêncio e ruídos que impactaram a extração de características e, consequentemente, a classificação do modelo.

Assim como no primeiro experimento, a matriz de confusão foi montada para se analisar os erros do sistema. A Figura 18 mostra a matriz de confusão resultante do segundo experimento e podemos observar que um fator comum às três *folds* é o erro na classificação entre as palavras 'Cozinha' e 'Apagar'. A Figura 19 mostra a única *fold* que

teve erro no terceiro experimento e podemos notar que o erro acontece quando ocorre confusão entre as classes 'Cozinha' e 'Apagar' também observada nos dois experimentos anteriores.

Com a diminuição do limiar do silêncio em 100 vezes o seu valor a cada experimento, o modelo oculto de Markov do presente trabalho obteve um acréscimo de 0,5% no segundo experimento e de 1% no terceiro experimento para a Base I. A taxa de acerto final de 99,5% define o melhor limiar de recorte de silêncio.

Apesar da diminuição do limiar do recorte do silêncio, o sistema de reconhecimento projetado continuou a confundir as classes de 'Apagar' e 'Cozinha' mas com taxas de erros inferiores a 1,5% e demonstrou um excelente desempenho para a base de palavras.

4.1.1.2 Limiar do recorte do silêncio - Base II

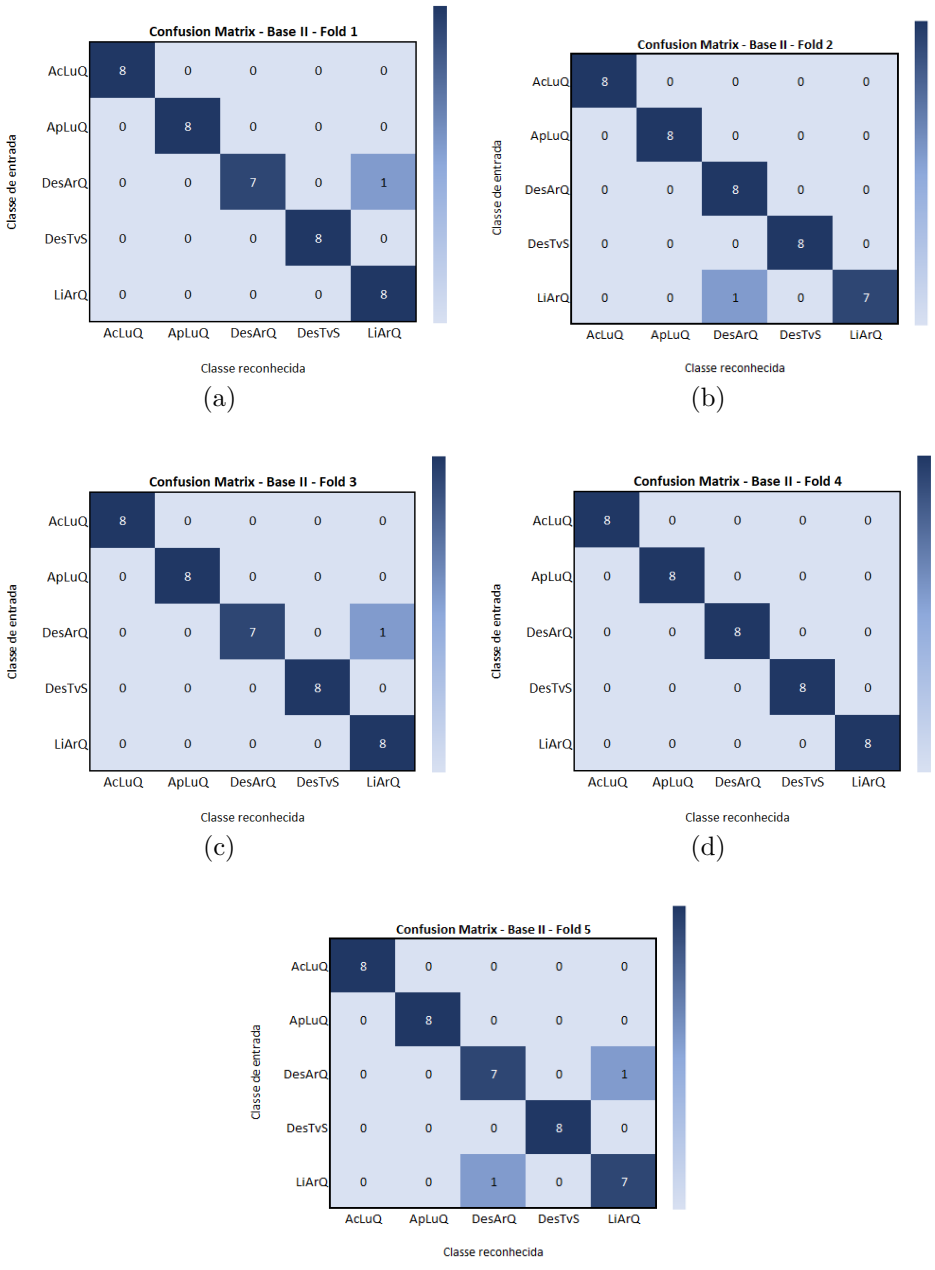
Para a base II, foram classificadas 5 frases e, para fins explicativos dos experimentos, utilizaremos as seguintes abreviaturas:

- AcLuQ - Acender Luz Quarto
- ApLuQ - Apagar Luz Quarto
- DesArQ - Desligar Ar Quarto
- DesTvS - Desligar TV Sala
- LiArQ - Ligar Ar Quarto

Para o primeiro experimento, utilizou-se o mesmo limiar de energia fixo de $2{,}9614 \times 10^{-02}$ a fim de recortar o silêncio existente antes e depois da palavra contida na elocução.

A partir dos dados de entrada, o sistema foi utilizado para classificar 40 locuções durante a etapa de testes de cada *fold*. A Figura 20 mostra todas as classificações feitas pelo modelo para cada sinal de entrada e as confusões do sistema entre as classes dos áudios de entrada e predições realizadas.

Figura 20 – Matriz confusão para o experimento do HMM com remoção de silêncio e limiar de recorte 2,9614 × 10^{-02} - Base II - Frases.



Fonte: Do autor.

Como pode ser observado na Figura 20, com exceção da *fold* 4 neste experimento, todas as demais *folds* houve confusão entre as classes de frases 'Desligar Ar Quarto' e 'Ligar Ar Quarto' Um total de 5 erros foram identificados no sistema e a taxa de acerto final do modelo foi de 97,5%. A Tabela 5 mostra os resultados obtidos para cada *fold* deste experimento com remoção de silêncio aplicada na base de palavras.

Tabela 5 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-02}$ - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	39	1	0,975
2	39	1	0,975
3	39	1	0,975
4	40	0	1,000
5	38	2	0,950
Total	**195**	**5**	**0,975**

Fonte: Do autor.

Para os experimentos seguintes, limiar de recorte de silêncio foi reduzido do experimento anterior até que se atingisse uma taxa de acerto ideal para os objetivos deste modelo. Foram realizados mais experimentos, um com o limiar de recorte definido como $2{,}9614 \times 10^{-04}$, outro como $2{,}9614 \times 10^{-06}$ e o último como $2{,}9614 \times 10^{-07}$. A taxa de acerto final destes modelos foi 98%, 98,5% e 100%, respectivamente. As Tabela 6, Tabela 7 e Tabela 8 mostram os resultados obtidos para cada *fold* destes experimentos com a remoção de silêncio variando sob a Base II.

Tabela 6 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-04}$ - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	38	2	0,950
2	40	0	1,000
3	40	0	1,000
4	38	2	0,950
5	40	0	1,000
Total	**196**	**4**	**0,980**

Fonte: Do autor.

Tabela 7 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-06}$ - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	39	1	0,975
3	38	2	0,950
4	40	0	1,000
5	40	0	1,000
Total	**197**	**3**	**0,985**

Fonte: Do autor.

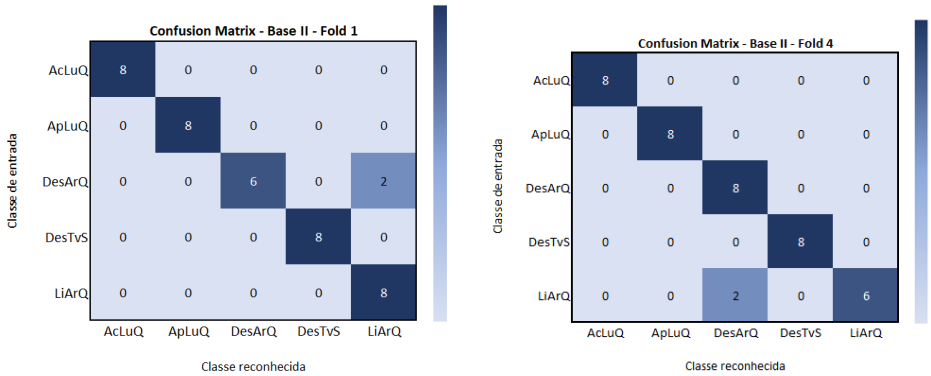
Tabela 8 – Taxa de acerto do modelo HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-07}$ - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	40	0	1,000
3	40	0	1,000
4	40	0	1,000
5	40	0	1,000
Total	**200**	**0**	**1,000**

Fonte: Do autor.

Os experimentos variando o limiar de recorte do silêncio mostraram que a diminuição do limiar para a base II resultou no aumento da taxa de acerto do modelo. As Figura 21 e Figura 22 mostram as *folds* que tiveram erros na classificação. O resumo dessa matriz de confusão para o segundo e terceiro experimento com limiares de $2{,}9614 \times 10^{-04}$ e $2{,}9614 \times 10^{-06}$ evidencia que o modelo HMM projetado comete os mesmos erros do primeiro experimento dessa base. Todos eles fazem a confusão entre as classes 'Ligar Ar Quarto' e 'Desligar Ar Quarto'.

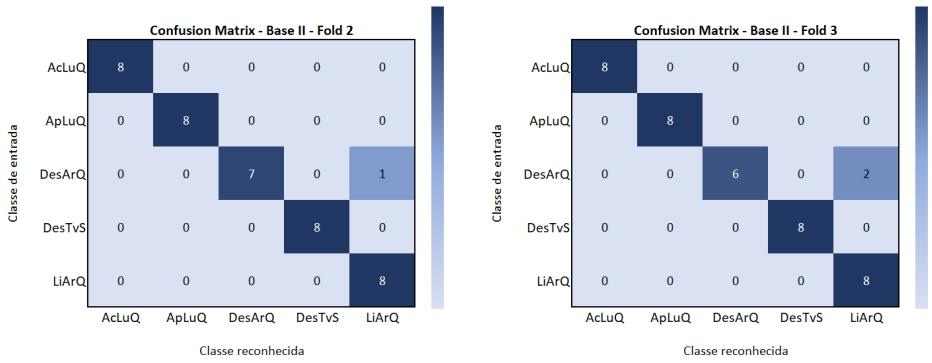
Figura 21 – *Folds* com erros na classificação para o experimento do HMM com remoção de silêncio e limiar de recorte $2{,}9614 \times 10^{-04}$ - Base II - Frases.



(a) (b)

Fonte: Do autor.

Figura 22 – *Folds* com erros na classificação para o experimento do HMM com remoção de silêncio e limiar de recorte 2,9614 × 10^{-06} - Base II - Frases.



(a) (b)

Fonte: Do autor.

O modelo chegou à taxa de acerto ideal de 100% para a classe de frases definida na Base II e o recorte de silêncio aplicado ao sinal de áudio original fez com que apenas informações importantes à classificação fossem ressaltadas para o modelo. Quando se trata de frases, o sinal de áudio de entrada possui uma alta quantidade de informações variantes dentro do mesmo comprimento de áudio.

4.1.2 Variação no número de gaussianas

Como descrito no Capítulo 2, a variação no número de gaussianas está relacionado à complexidade do modelo acústico. A variação no número de gaussianas pode melhorar ou impactar o desempenho do sistema de reconhecimento de fala.

Assim como nos experimentos com variação do limiar do recorte de silêncio , o experimento ocorreu através da validação cruzada de 5 *folds* e cada *fold* teve 22 sinais de áudio utilizados para o treinamento e os demais 8 dados foram utilizados para os testes.

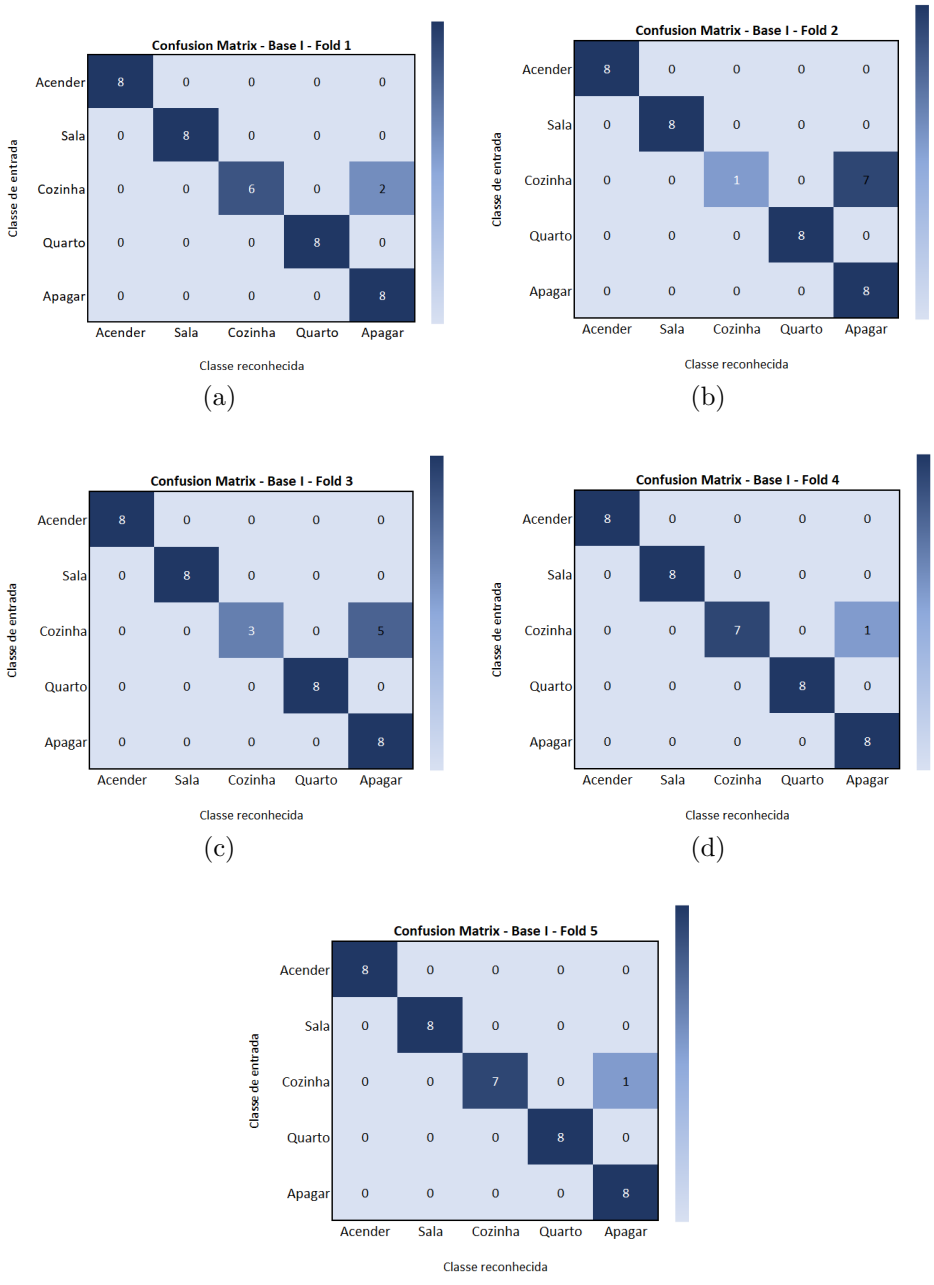
Para estes experimentos, o limiar do recorte de silêncio foi fixado no valor que resultou na melhor taxa de acerto entre os experimentos de variação do limiar da seção anterior dado por 2,9614 × 1006 . Apenas o número de gaussianas foi variado a fim de mensurar o impacto na taxa de acerto do sistema.

4.1.2.1 Variação no número de gaussianas - Base I

Para o primeiro experimento, foi utilizada apenas 1 gaussiana a fim de modelar a distribuição de probabilidades do modelo oculto de Markov (HMM).

A partir dos dados de entrada, o sistema fez a classificação de 40 locuções durante a etapa de testes de cada *fold*. A Figura 23 ilustra todas as classificações feitas pelo modelo para cada sinal de entrada e plota a matriz confusão.

Figura 23 – Matriz confusão para o experimento do HMM com 1 gaussiana- Base I - Palavras.



Fonte: Do autor.

Observa-se na Figura 23 que o modelo HMM projetado com 1 gaussiana comete um total de 16 erros e fez a confusão na classificação das palavras 'Cozinha' e 'Apagar' em todas as *folds*. O baixo número de gaussianas não permite ao modelo se ajustar de forma precisa às classes e isso ocasiona a confusão entre classes, fazendo com que o sistema seja um sistema de reconhecimento de fala de baixa complexidade. A Tabela 9 mostra o resultado de cada *fold* e a taxa de acerto final deste modelo com 1 gaussiana foi de 92%

Tabela 9 – Taxa de acerto do modelo HMM com 1 gaussiana - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	38	2	0,950
2	33	7	0,825
3	35	5	0,875
4	39	1	0,975
5	39	1	0,975
Total	**184**	**16**	**0,920**

Fonte: Do autor.

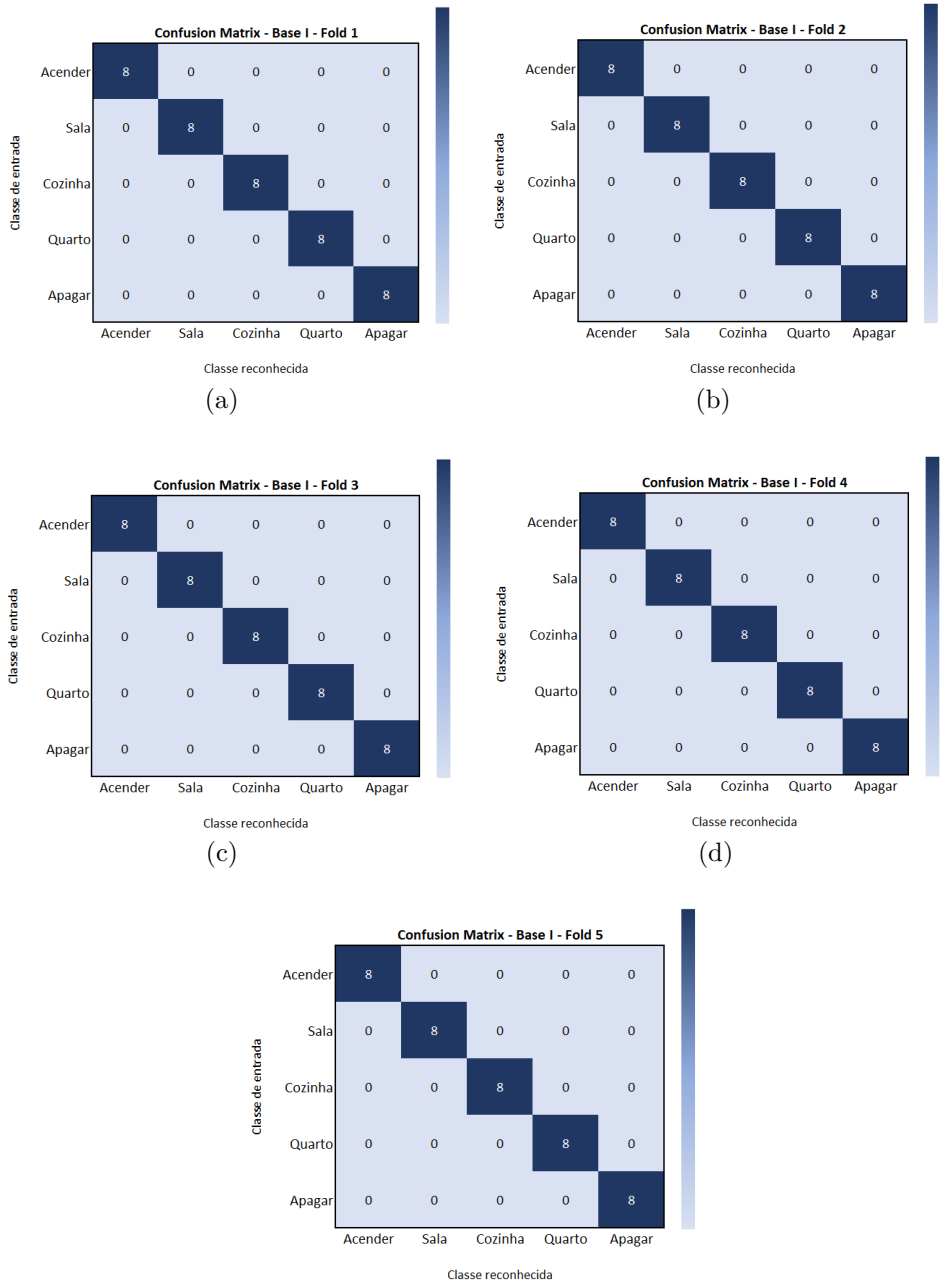
Para os experimentos seguintes, o número de gaussianas foi elevado a fim de aumentar a complexidade de reconhecimento e melhoria de taxa de acerto. Não existe o número perfeito para se definir a quantidade de gaussianas do HMM e para o segundo experimento, realizou-se um ensaio com com 15 gaussianas. A complexidade do modelo se elevou ao ponto de ser atingida a taxa de acerto de 100% no ASR projetado. A Tabela 10 e a Figura 24 detalham o resultado dessa taxa de acerto para cada *fold*.

Tabela 10 – Taxa de acerto do modelo HMM com 15 gaussianas - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	40	0	1,000
3	40	0	1,000
4	40	0	1,000
5	40	0	1,000
Total	**200**	**0**	**1,000**

Fonte: Do autor.

Figura 24 – Matriz confusão para o experimento do HMM com 15 gaussianas- Base I - Palavras.



Fonte: Do autor.

Porém, atingir essa taxa de acerto de 100%, o aumento do número de gaussianas fez com que o custo computacional aumentasse. O tempo de execução do modelo de HMM com 15 gaussianas teve um aumento em 817% em comparação ao modelo com 1 gaussiana, passando de um tempo de execução 13,23 segundos para 108,09 segundos.

No presente trabalho, um dos objetivos do sistema ASR projetado é otimizar o custo computacional mantendo a taxa de acerto alta. Portanto, foram executados novos experimentos de forma emírica, aumentando de 5 em 5 o número de gaussianas até se

alcançar o número de gaussianas ideal para atender à tarefa de reconhecimento de fala.

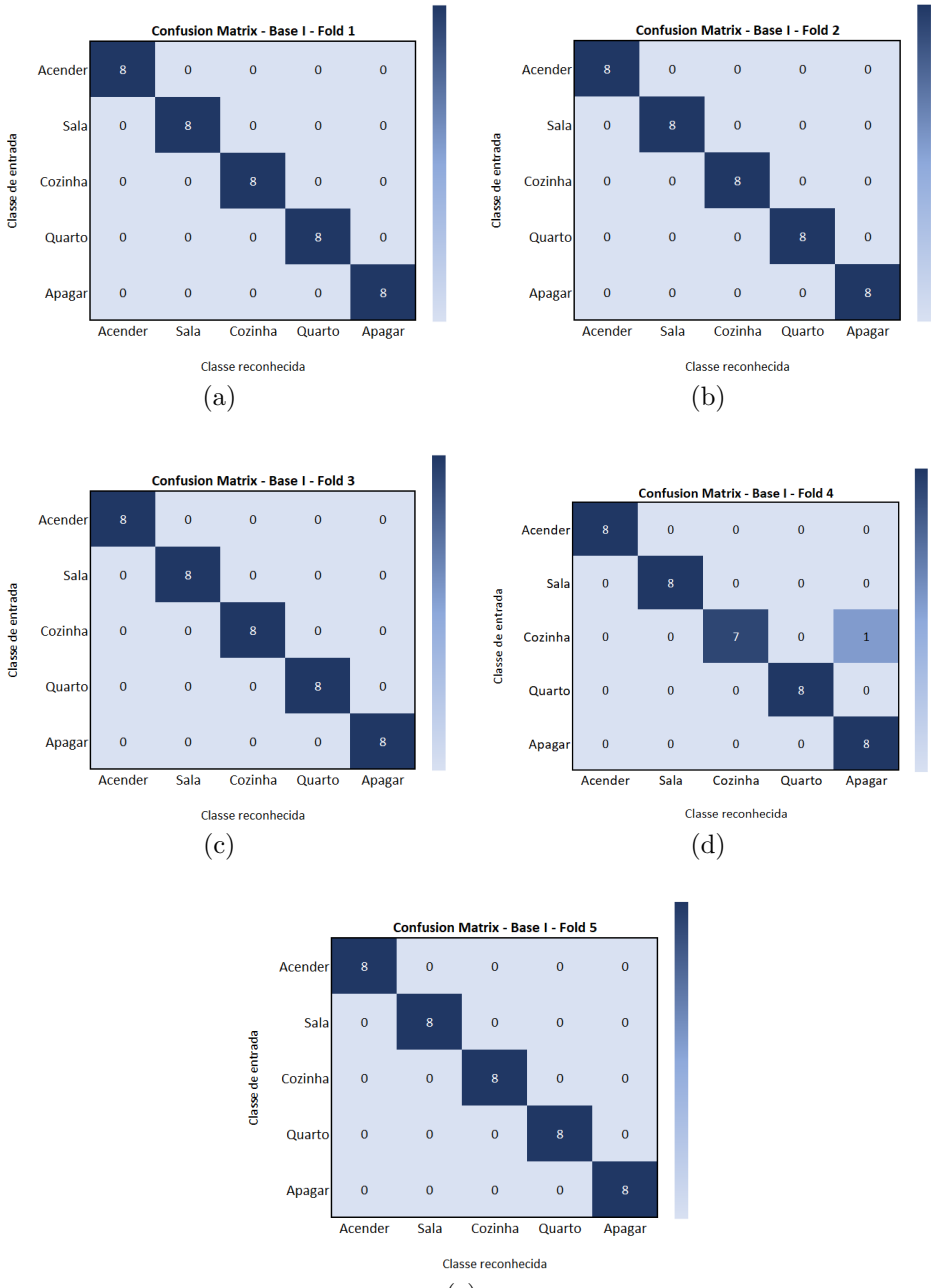
O último experimento na variação do número de gaussianas definiu como sendo 5 a quantidade mais apropriada para ser utilizada, neste problema, no modelo HMM. A Tabela 11 mostra os resultados obtidos para cada *fold* com este número de gaussianas e a Figura 25 mostra a matriz de confusão com o único erro que o sistema cometeu. Ao final, foi obtida uma taxa de acerto de 99,5% para a base I de palavras com 5 gaussianas.

Tabela 11 – Taxa de acerto do modelo HMM com 5 gaussianas - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	40	0	1,000
3	40	0	1,000
4	39	1	0,975
5	40	0	1,000
Total	**199**	**1**	**0,995**

Fonte: Do autor.

Figura 25 – Matriz confusão para o experimento do HMM com 5 gaussianas- Base I - Palavras.

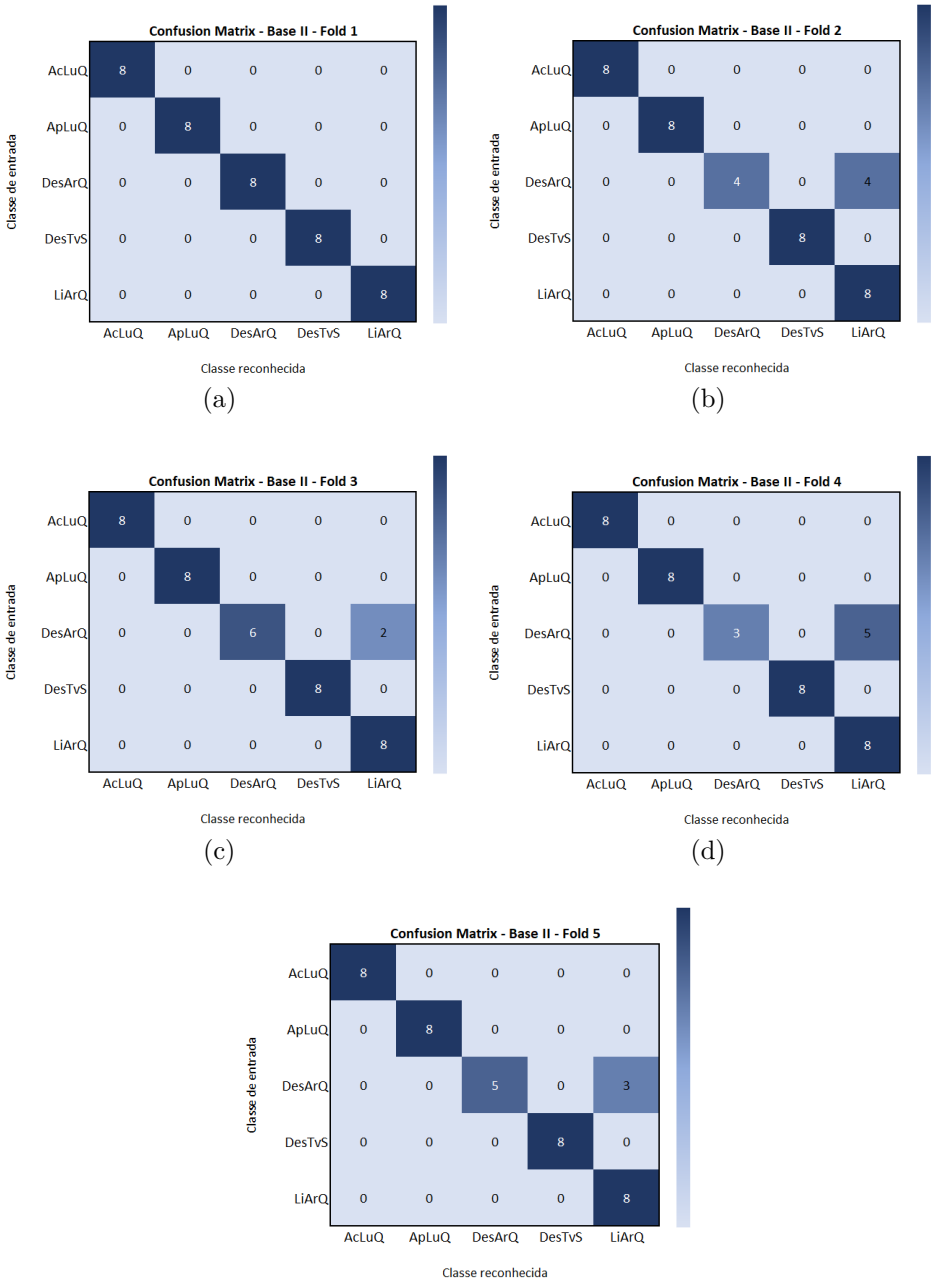


Fonte: Do autor.

4.1.2.2 Variação no número de gaussianas - Base II

Para o primeiro experimento, foi utilizada apenas 1 gaussiana a fim de modelar a distribuição de probabilidades do modelo oculto de Markov (HMM). A Figura 26 mostra todas as classificações feitas pelo modelo para cada sinal de entrada e define a matriz de confusão para este modelo.

Figura 26 – Matriz confusão para o experimento do HMM com 1 gaussiana- Base II - Frases.



Fonte: Do autor.

Observa-se na Figura 26 que o modelo HMM projetado com 1 gaussiana comete um total de 14 erros, resultando em uma taxa de acerto final de 93%. Todas as 14 classificações incorretas foram entre as classes 'Desligar Ar Quarto' e 'Ligar Ar Quarto'. A proximidade destas frases exige um número maior de gaussianas para separar as distribuições probabilísticas. A Tabela 12 mostra as taxa de acertos obtidas em cada *fold* do experimento.

Tabela 12 – Taxa de acerto do modelo HMM com 1 gaussiana - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	38	2	0,950
2	33	7	0,825
3	35	5	0,875
4	39	1	0,975
5	39	1	0,975
Total	**186**	**14**	**0,930**

Fonte: Do autor.

A abordagem utilizando apenas 1 gaussiana não foi eficiente para diferenciar as duas classes que possuem todos os fonemas em comum, com exceção do fonema "Des"na frase 'Desligar Ar Quarto'. Com isso, foram aumentados os números de gaussianas para os experimentos seguintes. Foram realizados mais dois experimentos apenas, o primeiro com 5 gaussianas e o segundo com 15 gaussianas. As Tabela 13 e a Tabela 14 mostram que ambos os modelos atingiram a taxa de acerto final de 100%.

Tabela 13 – Taxa de acerto do modelo HMM com 5 gaussianas - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	40	0	1,000
3	40	0	1,000
4	40	0	1,000
5	40	0	1,000
Total	**200**	**0**	**1,000**

Fonte: Do autor.

Tabela 14 – Taxa de acerto do modelo HMM com 15 gaussianas - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	40	0	1,000
3	40	0	1,000
4	40	0	1,000
5	40	0	1,000
Total	**200**	**0**	**1,000**

Fonte: Do autor.

O modelo com 5 gaussianas foi escolhido como o mais apropriado para o sistema de reconhecimento de fala ASR projetado no presente trabalho. Apesar do modelo com 15 gaussianas também ter atingido a taxa de acerto de 100%, este teve um custo computacional aumentado em 250% quando comparado com o modelo de 5 gaussianas. Portanto, com o objetivo de aplicar o sistema de reconhecimento da fala em atividades e áreas da

sociedade que exigem uma iteração mais rápida, o modelo com 5 gaussianas se torna o mais ideal.

4.1.3 Modelos finais e comparações HMM

4.1.3.1 Modelos finais HMM - Base I

Podemos perceber pela Tabela 15 que o sistema encontra uma taxa de acerto de 100%, mas, conforme visto durante os experimentos, o custo computacional foi alto e não condiz com o objetivo do trabalho.

O modelo mais adequado do sistema de reconhecimento de fala com HMM encontrado foi o modelo com 5 gaussianas e limiar do recorte de silêncio como $2{,}9614 \times 10^{-04}$ onde a taxa de acerto final foi de 99,5%, encontrando apenas 1 erro dentro das 200 classificações realizadas. Este é o melhor resultado visto que não teria como o sistema ter 0,5 erro e que ter zero erros tem um custo operacional alto.

Tabela 15 – Taxa de acerto dos modelos HMM - Base I - Palavras.

	1 gaussiana	**5 gaussianas**	**15 gaussianas**
$2{,}9614 \times 10^{-02}$	-	98%	-
$2{,}9614 \times 10^{-04}$	-	98,5%	-
$2{,}9614 \times 10^{-06}$	92%	99,5%	100%
$2{,}9614 \times 10^{-07}$	-	98%	-

Fonte: Do autor.

Vale ressaltar que todos os erros do sistema, independentemente da variação dos parâmetros citados, ocorreram na confusão de classificação entre as palavras 'Cozinha' e 'Apagar' o que pode indicar uma má qualidade nos sinais gravados destas duas classes. O resultado de classificação assertiva de 99,5% com os erros concentrados em uma mesma classificação se mostrou um resultado satisfatório no reconhecimento da fala para o modelo HMM com a base I.

4.1.3.2 Modelos finais HMM - Base II

Para a segunda base de dados, de frases, os experimentos também possibilitaram a análise com relação aos impactos do limiar de recorte do silêncio e da variação do número de gaussianas. Para este modelo, a complexidade do sinal reconhecido foi elevada ao ser produzidos mais fonemas do que a base de palavras.

Podemos perceber pela Tabela 16 que o sistema encontra a taxa de acerto de 100% em dois dos experimentos realizados. Conforme observado durante os experimentos, o custo computacional foi alto para o modelo com 15 gaussianas e não condiz com o objetivo do trabalho.

O modelo ideal do sistema de reconhecimento de fala com HMM encontrado foi o modelo com 5 gaussianas e limiar do recorte de silêncio como $2{,}9614 \times 10^{-07}$, onde a taxa de acerto final também foi de 100%, mas com custo operacional aceitável.

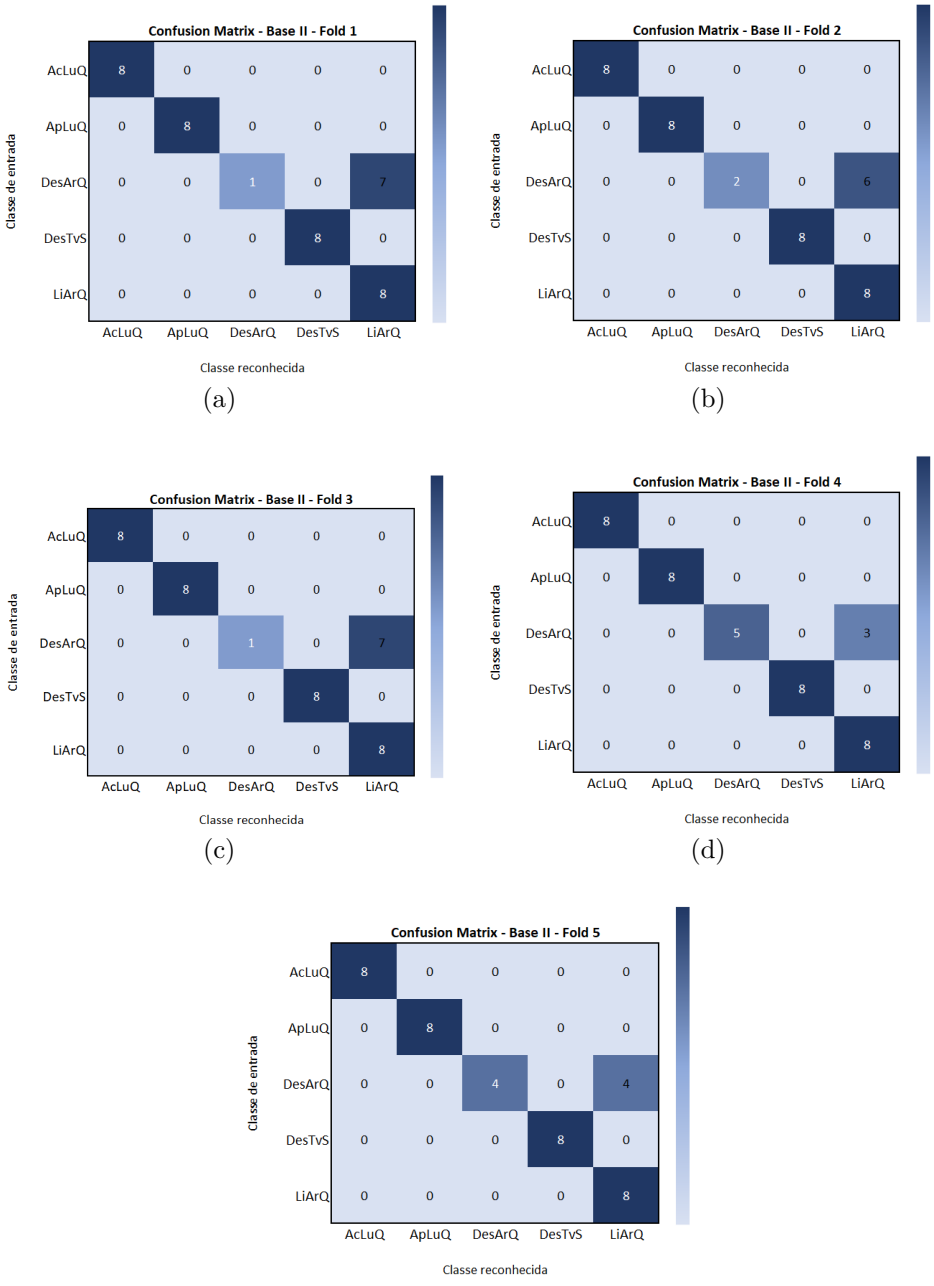
Tabela 16 – Taxa de acerto dos modelos HMM - Base II - Frases.

	1 gaussiana	**5 gaussianas**	**15 gaussianas**
$2{,}9614 \times 10^{-02}$	86,5%	97,5%	-
$2{,}9614 \times 10^{-04}$	-	98%	-
$2{,}9614 \times 10^{-06}$	-	98,5%	-
$2{,}9614 \times 10^{-07}$	93%	100%	100%

Fonte: Do autor.

Para esta base de dados, foi realizado um experimento final com os piores parâmetros encontrados nos experimentos anteriores. O resultado dessa taxa de acerto foi de 86,5% onde o sistema cometeu 27 erros na classificação. Como podemos observar na matriz de confusão plotada para este experimento na Figura 27, todas as classificações erradas foram entre as classes 'Desligar Ar Quarto' e 'Ligar Ar Quarto'.

Figura 27 – Matriz confusão para o experimento do HMM com os piores parâmetros - Base II - Frases.



Fonte: Do autor.

Esta classificação errada é a única encontrada em todos os experimentos realizados no HMM com a Base II. A similaridade dos fonemas das duas classes é a origem dos erros de classificação para o modelo. Porém, ao se aumentar o número de gaussianas, o modelo foi capaz de classificar estas classes mesmo com alta complexidade. A taxa de acerto de 100% encontrada no HMM ajustada com o menor recorte de silêncio e 5 gaussianas atinge todos os objetivos propostos de reconhecimento de fala para a base II.

4.2 Redes Neurais Convolucionais - CNN

Nesta seção serão descritos e discutidos os resultados encontrados para os diferentes experimentos realizados com base no modelo criado utilizando a CNN construída para a tarefa de reconhecimento automático da fala. Em todos os experimentos que serão discutidos nesta seção, a extração de características foi feita após a etapa de pré-processamento dos dados e anterior à rede convolucional projetada.

4.2.1 Limiar do recorte do silêncio

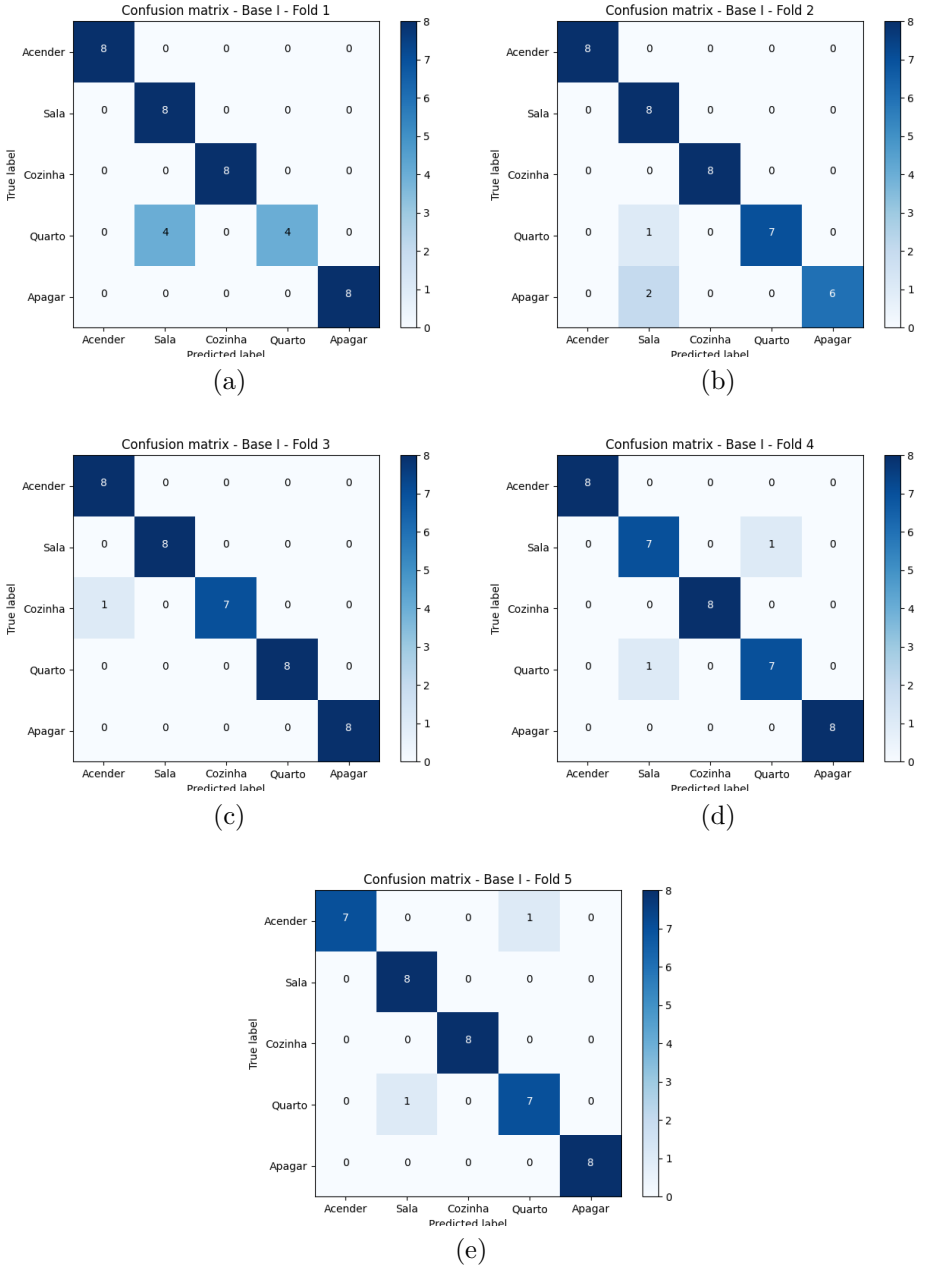
Para o modelo de redes neurais convolucionais, foi aplicada a técnica de remoção do silêncio com o limiar fixo, utilizando-se o mesmo limiar que obteve melhor resultados nos experimentos com HMM. Esta remoção foi aplicada aos sinais áudios da base de dados durante a fase de pré-processamento e preparação dos áudios antes da extração de características e inserção dos sinais da entrada na rede neural convolucional projetada.

4.2.1.1 Limiar do recorte do silêncio - Base I

Para os testes na base I, realizou-se a validação cruzada mencionada anteriormente e os dados foram separados em 5 *folds* para que todos os dados passassem pelos testes, garantindo assim a exclusão do fator aleatoriedade, uma vez que todos os sinais de áudios da base I foram utilizados nos testes. Conforme visto anteriormente, em cada *fold* os dados utilizados no treinamento não se repetem nos testes.

Para cada *fold*, foi gerado uma matriz de confusão a fim de identificar os erros de classificação que o sistema obteve. As linhas desta matriz indicam a classe à qual o sinal de entrada do teste pertence e as colunas indicam a classe à qual a rede neural faz a classificação. A Figura 28 apresenta os resultados obtidos no reconhecimento do sinal de fala para a base de palavras.

Figura 28 – Matriz confusão para o experimento com remoção de silêncio - Base I - Palavras.



Fonte: Do autor.

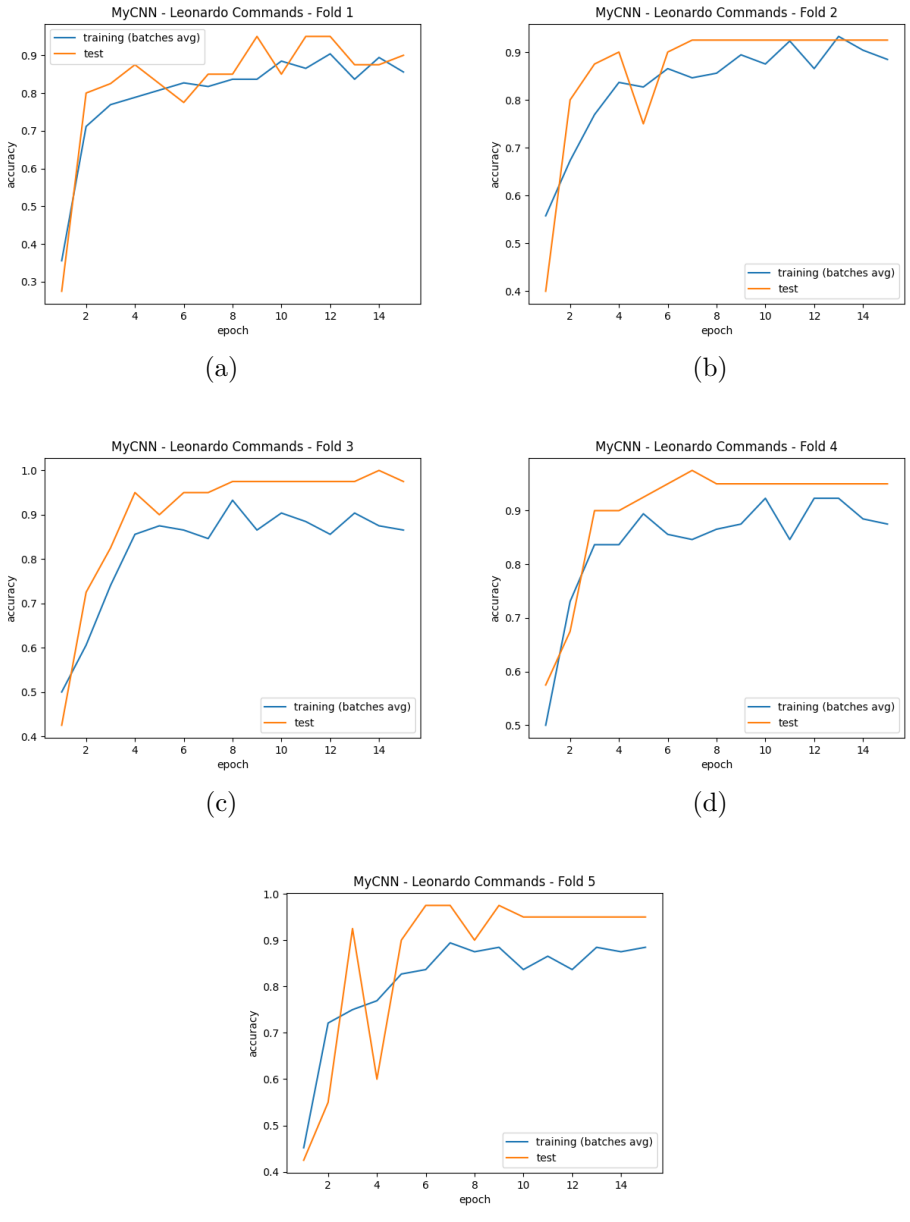
A partir desta matriz de confusão, é possível observar que o maior erro de classificação da rede foi com as palavras 'Quarto' e 'Sala'. Como pode ser visto na Figura 28(a), em uma mesma *fold*, a rede neural cometeu o mesmo erro 4 vezes durante a fase de testes. No total, a rede classificou erroneamente estas classes 8 vezes, representando um total de 67% de todos os erros para esta base e com a aplicação da remoção de silêncio.

Foram executadas 15 épocas à etapa de treinamento para que a rede neural aprendesse conforme o avanço das épocas e leituras dos áudios de entrada. Os neurônios da rede tiveram seus pesos ajustados para classificações com base na retropropagação da

rede neural.

A Figura 29 apresenta o desempenho da rede neural convolucional do presente trabalho ao longo das épocas e à melhoria desta rede.

Figura 29 – Taxa de acerto do modelo CNN com a remoção de silêncio - Base I - Palavras.



Fonte: Do autor.

Como pode ser observado, um fator comum a todas as *folds* é a melhora de desempenho da rede a cada época avançada durante os primeiros treinamentos. O sistema executa o primeiro treinamento sem nenhum preparo ou sem ter tido contato com a base de dados em estudo, e a taxa de acerto se aproxima de 0. Com o passar das épocas, a taxa de acerto do modelo caminha para perto de 1. Isso se deve ao ajuste dos parâmetros

da CNN, que ocorre durante a fase de treinamento, pela retropropagação.

Para o experimento da CNN com a remoção de silêncio para a Base I, a *fold* 1 teve o pior resultado de taxa de acerto, conforme pode ser visto pela matriz de confusão na Figura 28(a). Mas, como pode ser observado na Figura 29(a), a rede chegou a atingir a taxa de acerto de 1 entre as épocas 9 e 12, tendo uma queda posterior.

A Tabela 17 mostra os resultados obtidos em cada *fold* e a taxa de acerto média do modelo com a remoção do silêncio para esta base de palavras foi de 94%.

Tabela 17 – Taxa de acerto do modelo com recorte de silêncio - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	36	4	0,900
2	37	3	0,925
3	39	1	0,975
4	38	2	0,950
5	38	2	0,950
Total	**188**	**12**	**0,940**

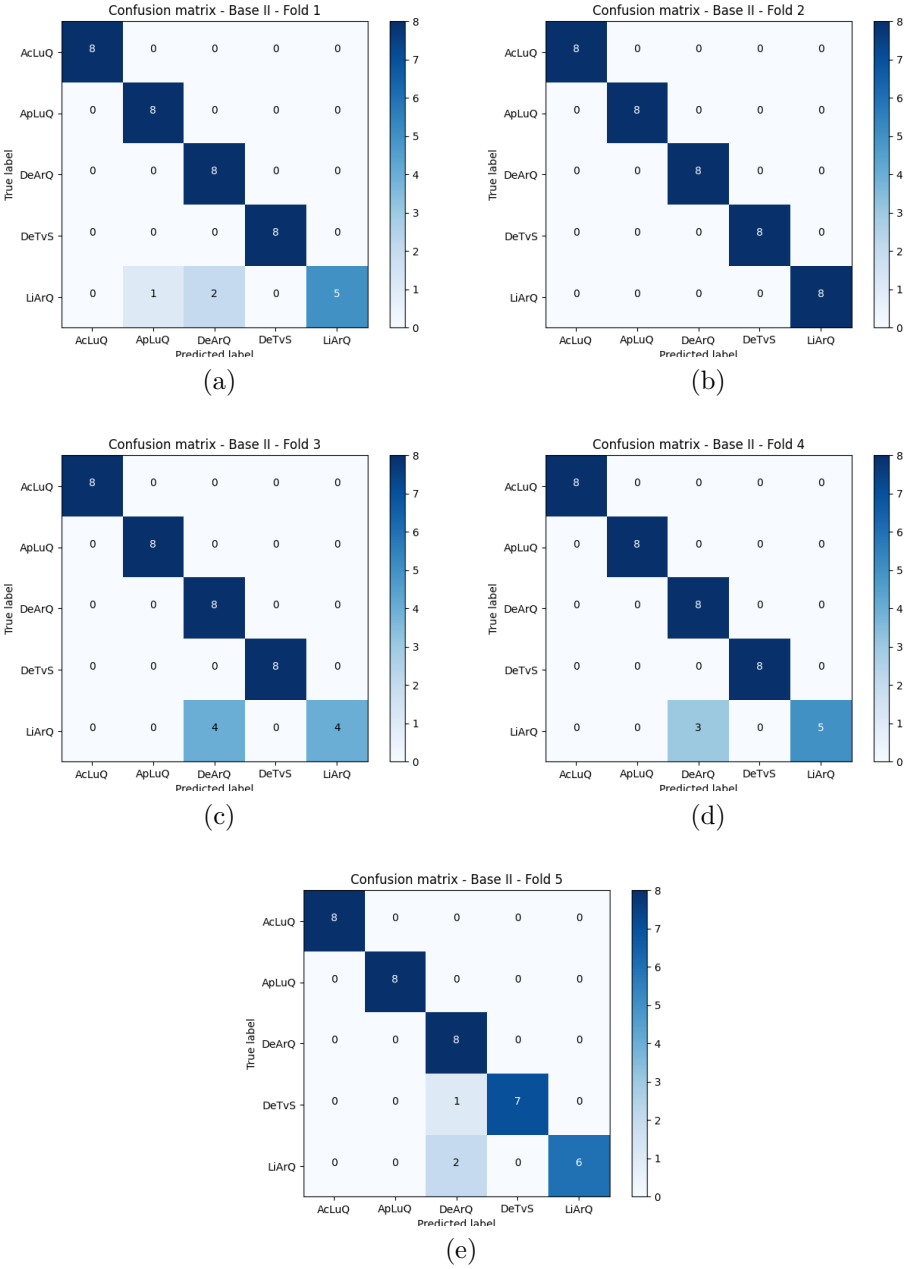
Fonte: Do autor.

4.2.1.2 Limiar do recorte do silêncio - Base II

Para os testes na Base II, também foi feita a validação cruzada de 5 *folds* com o mesmo objetivo de varredura de toda a base de áudio de entrada.

Para cada *fold* foi plotada uma matriz de confusão a fim de identificar os erros de classificação que o sistema obteve. A Figura 30 mostra os resultados obtidos no reconhecimento do sinal de fala para a base de frases com a remoção do silêncio.

Figura 30 – Matriz confusão para o experimento com remoção de silêncio - Base II - Frases.



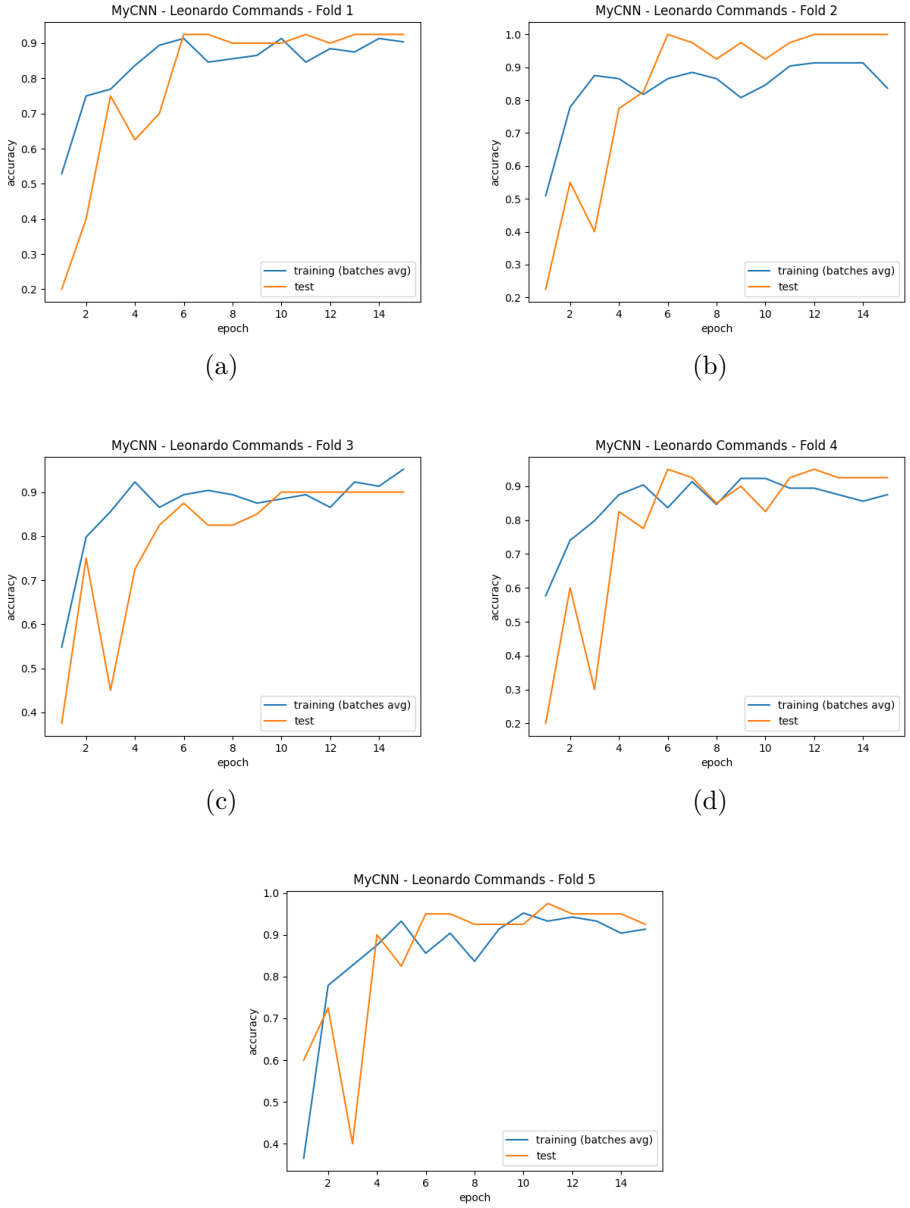
Fonte: Do autor.

A partir desta matriz de confusão é possível observar que o maior erro de classificação da rede foi com as frases 'Desligar Ar Quarto' e 'Ligar Ar Quarto'. A rede neural cometeu esse erro em diferentes *folds* somando 11 erros para essa classificação e todos eles sendo a entrada como 'Ligar Ar Quarto'. Estes erros representam um total de 85 % de todos os erros para a base II com a aplicação da remoção de silêncio.

Foram executadas 15 épocas à etapa de treinamento para que a rede neural aprendesse conforme o avanço das épocas e leituras dos áudios de entrada. A Figura 31 mostra o desempenho da rede neural convolucional do presente trabalho em comparação ao avanço

das épocas e à melhora dessa rede.

Figura 31 – Taxa de acerto do modelo CNN com a remoção de silêncio - Base II - Frases.



(a) (b) (c) (d) (e)

Fonte: Do autor.

Como pode ser observado, todas as *folds* da base de frases (Base II) levaram aproximadamente 6 épocas para atingirem uma taxa de acerto próxima a 90%, enquanto os experimentos da base de palavras (Base I) levaram aproximadamente 3 épocas para atingirem esse mesmo nível de taxa de acerto. Esse resultado evidencia o aumento de complexidade ao se passar de uma classificação de palavras para uma classificação de frases, onde o volume de dados ao longo do tempo é maior.

Para o experimento da CNN com a remoção de silêncio para a Base II, a *fold* 2 teve

o melhor resultado de taxa de acerto, conforme pode ser visto pela matriz de confusão na Figura 30(b). Na Figura 31(b), a rede atinge a taxa de acerto de 1 na época 12 e mantém esse resultado até o final das épocas, definindo a taxa de acerto dessa *fold* como 1.

A Tabela 18 mostra os resultados obtidos em cada *fold* e a taxa de acerto média do modelo com a remoção do silêncio para esta base foi de 93,5%.

Tabela 18 – Taxa de acerto do modelo com recorte de silêncio - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	37	3	0,925
2	40	0	1,000
3	36	4	0,900
4	37	3	0,925
5	37	3	0,925
Total	**187**	**13**	**0,935**

Fonte: Do autor.

4.2.2 Distensão Temporal

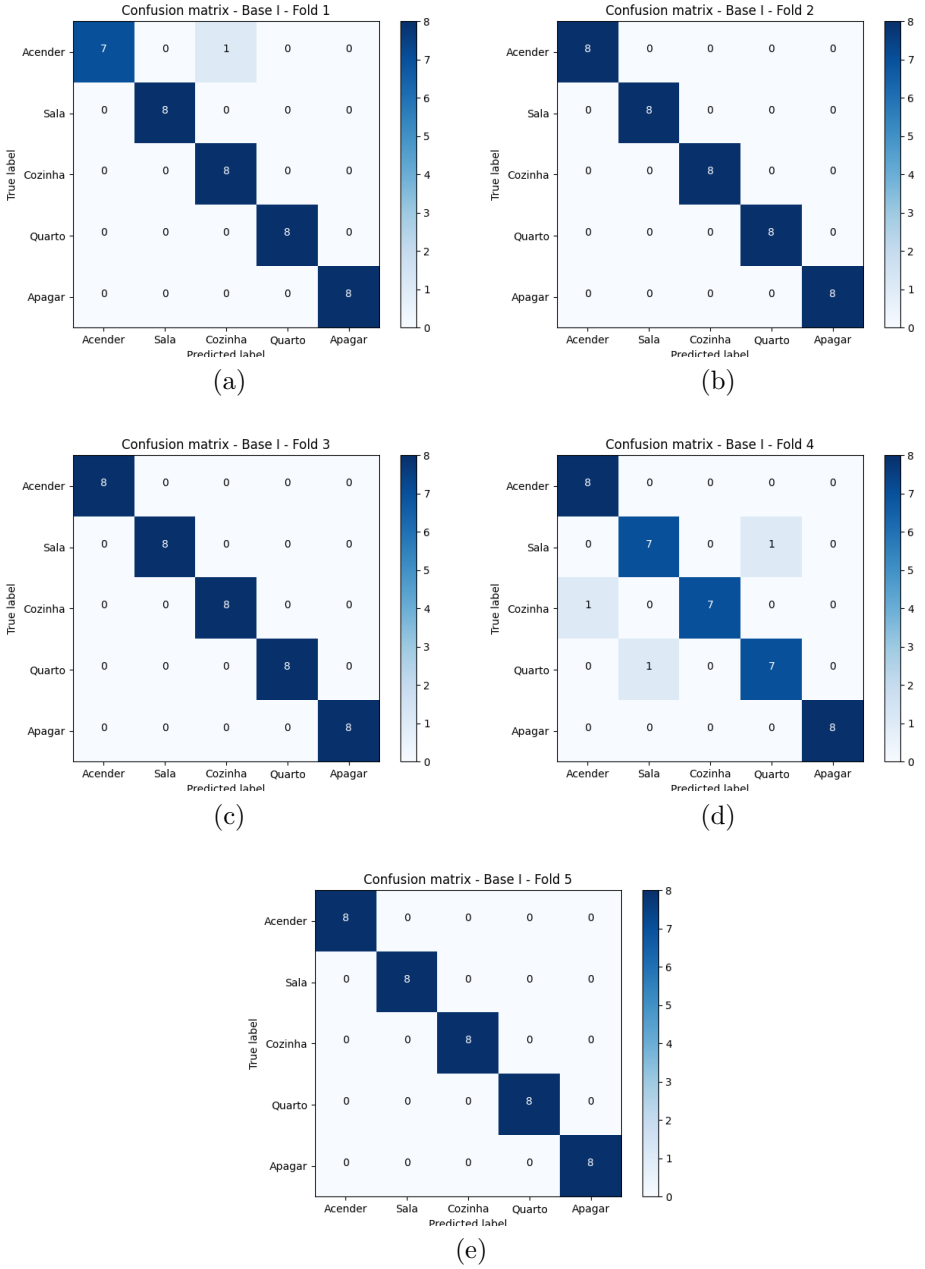
Neste experimento, foi aplicada somente a técnica de distensão temporal, do inglês *time stretch* com um fator de variação aleatório entre 0.8 e 1.2. Esta distensão foi aplicada aos sinais de áudio da base de dados durante a fase de pré-processamento e preparação dos áudios antes da extração de características e inserção dos áudios na rede neural convolucional projetada. Ela foi variada aleatoriamente para cada áudio, de forma que o tamanho do áudio não fosse alterado, apenas acelerado e desacelerado e o objetivo dessa distenção é aumentar a base de dados para a rede.

4.2.2.1 Distensão Temporal - Base I

Para os testes de distensão temporal na base I, realizou-se a validação cruzada com 5 *folds*. Todos os dados foram utilizados tanto na etapa de treinamento quanto na etapa de testes, sem que um mesmo áudio fosse utilizado nas duas etapas dentro de uma mesma *fold*.

Para cada *fold* foi plotada uma matriz de confusão a fim de identificar os erros de classificação que o sistema obteve. A Figura 32 mostra os resultados obtidos no reconhecimento do sinal de fala para a base de frases com a remoção do silêncio.

Figura 32 – Matriz confusão para o experimento com distensão temporal - Base I - Palavras.



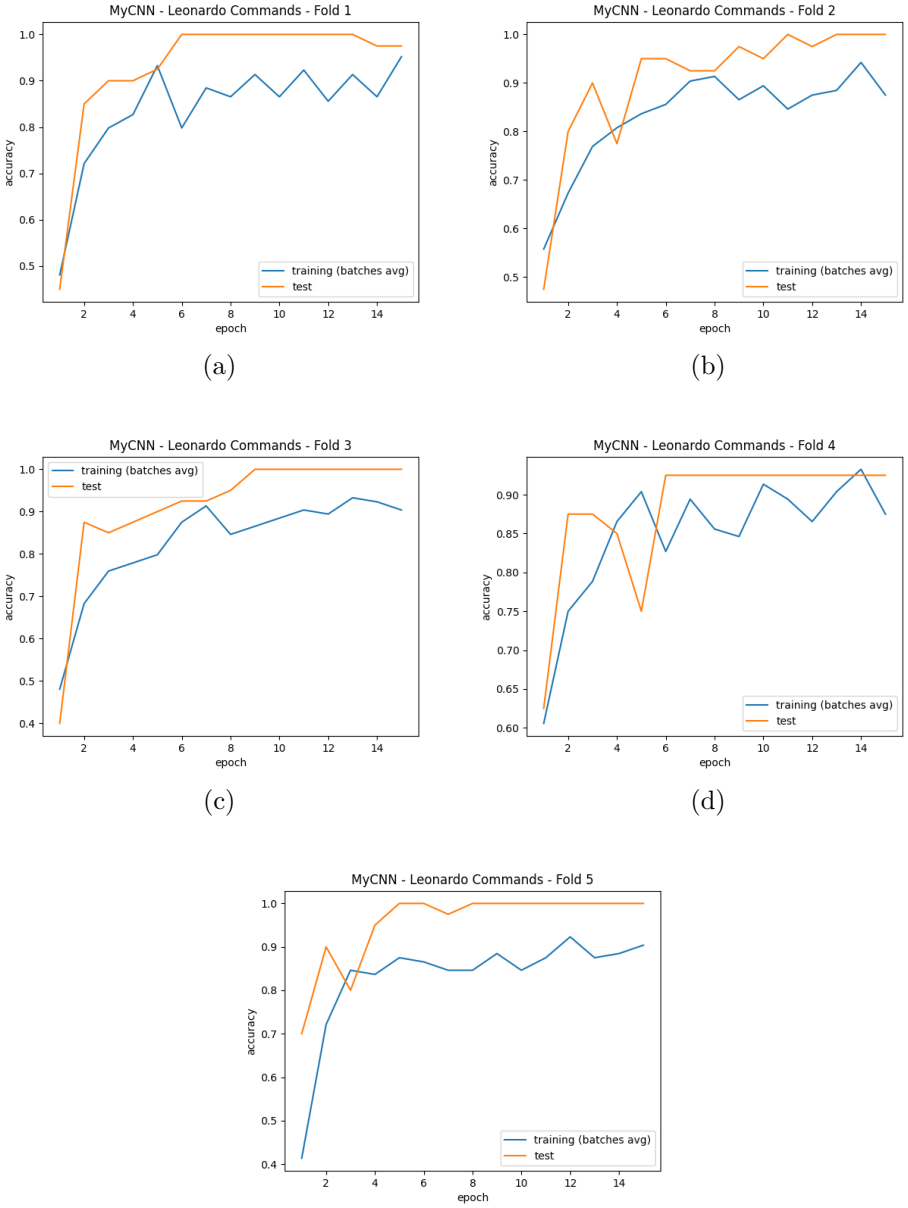
Fonte: Do autor.

A partir desta matriz de confusão é possível observar que os erros de classificação ficaram divididos entre a confusão entre 'Sala' e 'Quarto' e a confusão entre 'Cozinha' e 'Acender' onde cada confusão ocorreu 2 vezes e representam 50% dos 4 erros. Ressalta-se que, no teste com remoção de silêncio para a Base I, o maior prejuízo para a taxa de acerto da rede foi a confusão entre 'Sala' e 'Quarto'.

Foram executadas 15 épocas à etapa de treinamento para que a rede neural aprendesse conforme o avanço das épocas e reconhecimento de palavras a partir dos áudios de entrada. A Figura 33 mostra o desempenho da rede neural convolucional do presente

trabalho em comparação ao avanço das épocas e melhora desta rede.

Figura 33 – Taxa de acerto do modelo CNN com a distensão temporal - Base I - Palavras.



(a) (b) (c) (d) (e)

Fonte: Do autor.

Como pode ser observado nas Figuras 33(b), 33(c) e 33(e), as 3 *folds* alcançaram a taxa de acerto 1 no sistema com o avanço das épocas e assim se mantiveram até o final. Podemos observar que entre as épocas 4 e 6, a rede neural atingiu uma taxa de acerto próxima de 90%, mostrando um leve atraso da rede para atingir uma taxa de acerto próxima daquela atingida no experimento de remoção de ruído. Mas este experimento mostra uma melhor taxa de acerto ao final das épocas.

A Tabela 19 mostra os resultados obtidos em cada *fold* e a taxa de acerto média

do modelo com a distensão temporal para esta base foi de 98%.

Tabela 19 – Distensão temporal - Base I - Palavras.

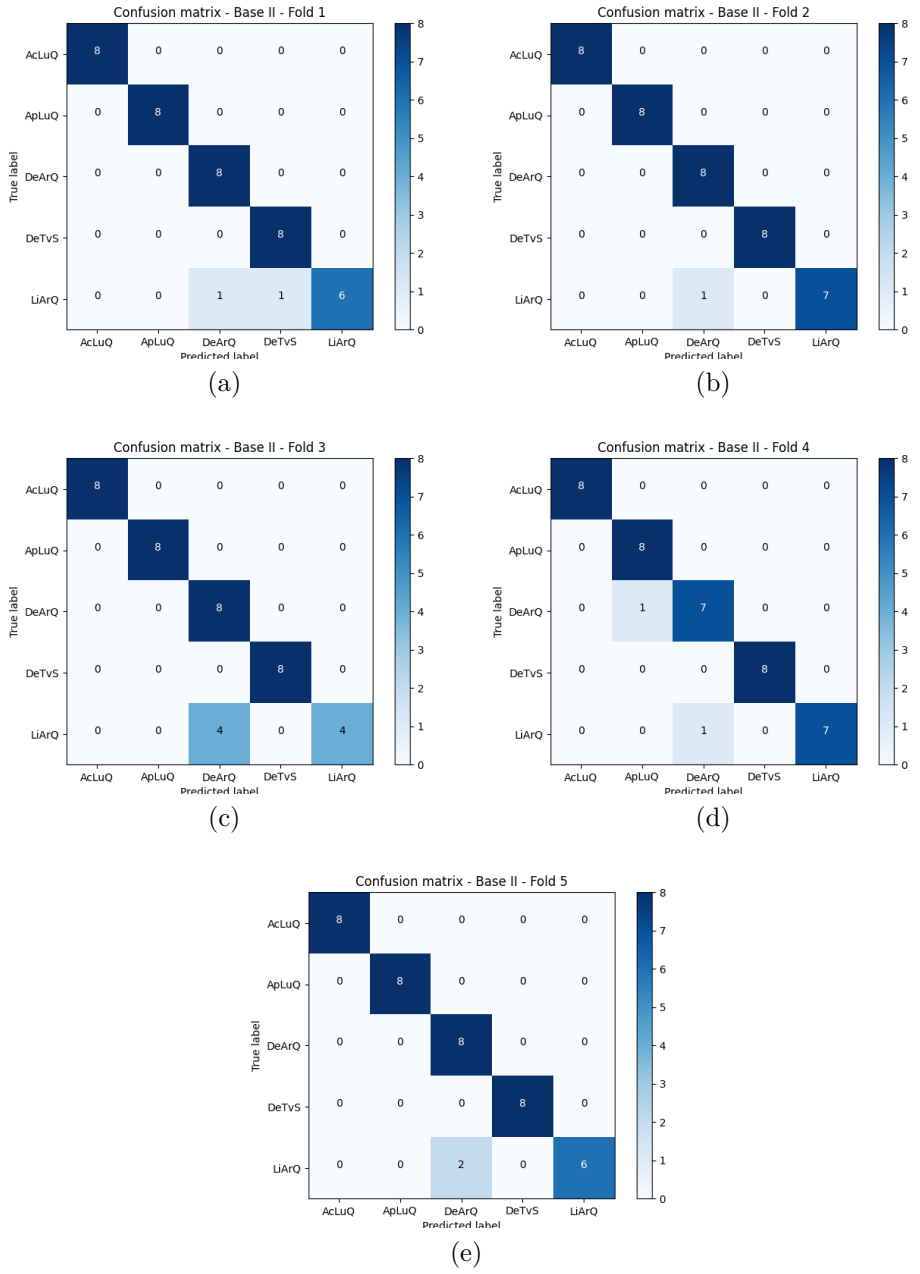
Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	39	1	0,975
2	40	0	1,000
3	40	0	1,000
4	37	3	0,925
5	40	0	1,000
Total	**196**	**4**	**0,980**

Fonte: Do autor.

4.2.2.2 Distensão Temporal - Base II

Para os testes de distensão temporal na base II, também foi feita a validação cruzada com 5 *folds*. Para cada *fold* foi plotada uma matriz de confusão a fim de identificar os erros de classificação que o sistema obteve. A Figura 34 mostra os resultados obtidos no reconhecimento do sinal de fala para a base de frases com a remoção do silêncio.

Figura 34 – Matriz confusão para o experimento com distensão temporal - Base II - Frases.

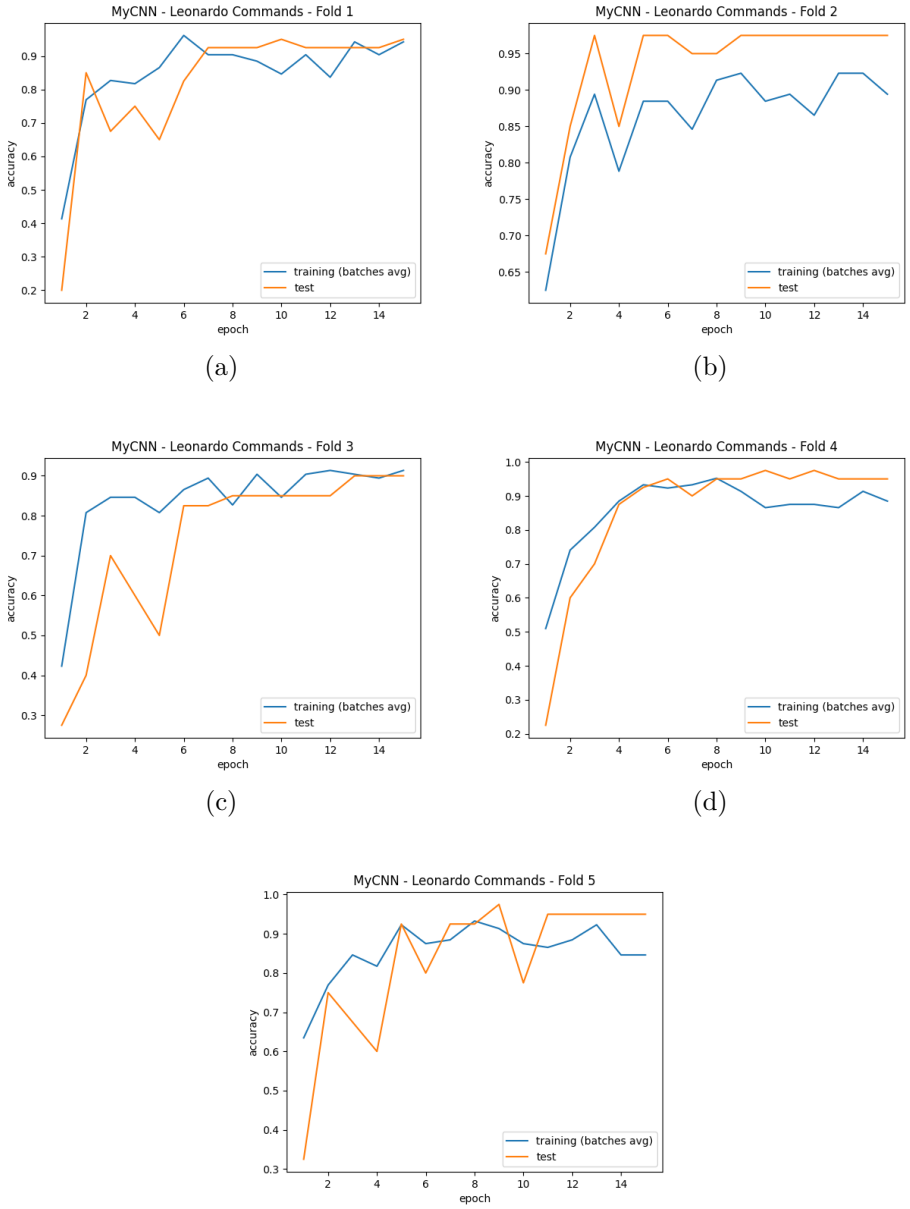


Fonte: Do autor.

Assim como no experimento de remoção de silêncio, a rede apresentou um baixo desempenho com um número elevado de erros na confusão entre as classes 'Ligar Ar Quarto' e 'Desligar Ar Quarto'. Neste experimento, a confusão em questão apresentou um total de 9 erros. Também podemos observar que a confusão teve a mesma origem do sinal de áudio de entrada, sendo a classe de 'Ligar Ar Quarto'.

Foram executadas 15 épocas na etapa de treinamento para que a rede neural melhorasse o seu desempenho com a retropropagação. A Figura 35 mostra o desempenho da rede neural convolucional do presente trabalho em comparação ao avanço das épocas e à melhora desta rede.

Figura 35 – Taxa de acerto do modelo CNN com a distensão temporal - Base II - Frases.



(a) (b) (c) (d) (e)

Fonte: Do autor.

Como pode ser observado na Figura 35(c), a *fold* 3 obteve o pior resultado para o experimento de distensão temporal aplicado à Base II. Nesta *fold*, a rede levou 13 épocas para atingir uma taxa de acerto de 0,9 na tarefa de classificação e reconhecimento do sinal da fala. Esse atraso específico dessa *fold* também pode ser observado no experimento de remoção de silêncio com as mesmas confusões na classificação.

A Tabela 20 mostra os resultados obtidos em cada *fold* e a taxa de acerto média do modelo com a distensão temporal para esta base de frases foi de 94,5%.

Tabela 20 – Distensão temporal - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	38	2	0,950
2	39	1	0,975
3	36	4	0,900
4	38	2	0,950
5	38	2	0,950
Total	**189**	**11**	**0,945**

Fonte: Do autor.

4.2.3 CNN com remoção do silêncio e distensão temporal

Conforme mencionado no início desta seção, a extração de características é realizada após o tratamento de áudio durante a etapa de pré-processamento. No experimento da remoção do silêncio, estas características foram extraídas do sinal sem o silêncio e, no experimento de distensão temporal, as características foram extraídas do sinal alterado temporalmente.

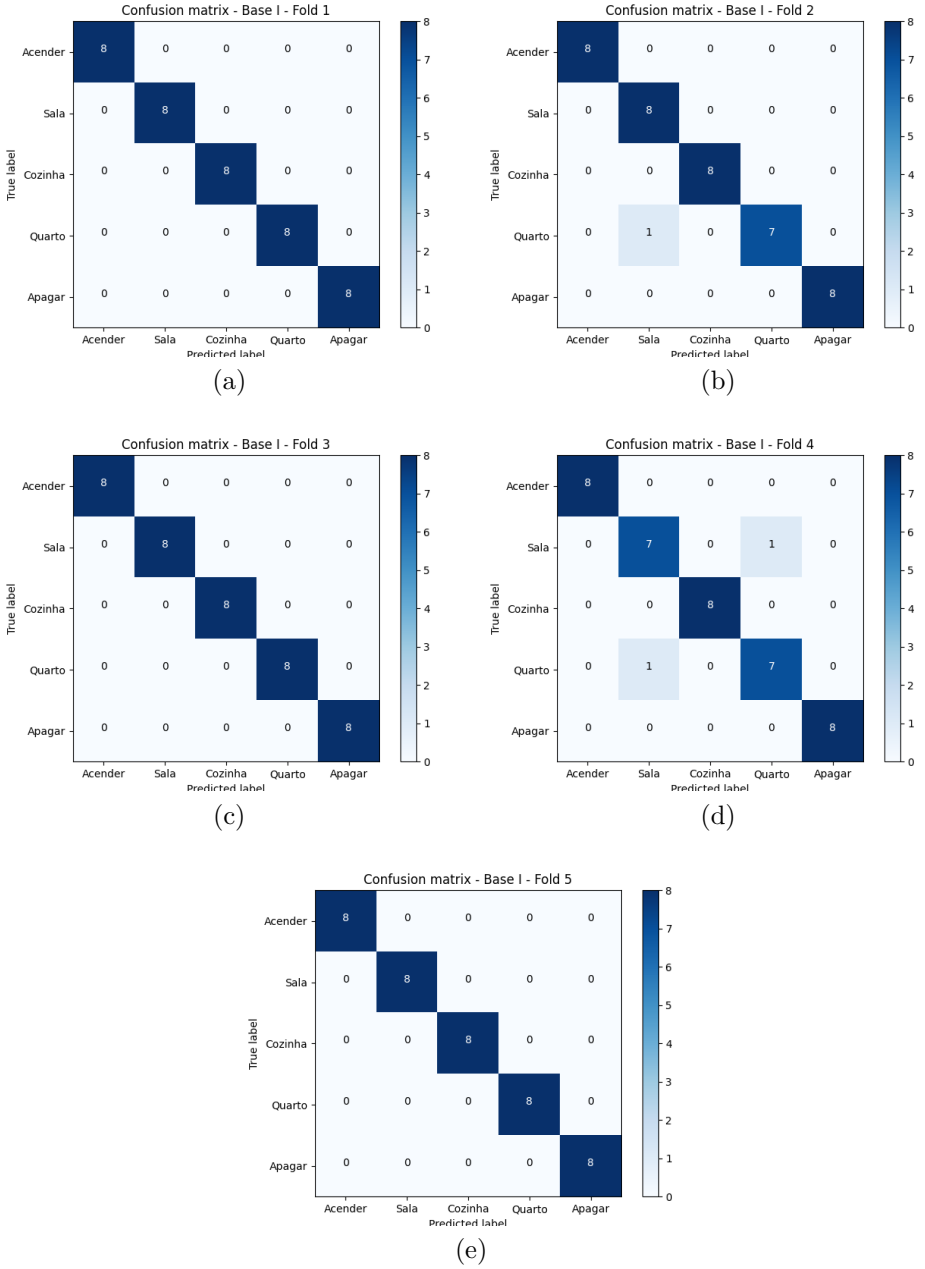
Estas características extraídas formam a entrada na primeira camada convolucional, ou a entrada na rede neural convolucional uma vez que a camada conv1 é a primeira camada da rede. Diferentemente dos primeiros experimentos em que o sinal original foi alterado por apenas uma técnica, neste experimento, a rede neural atuou com a aplicação de ambas as técnicas combinadas. Primeiramente, foi aplicada a distensão temporal para aumentar a base e depois removido o silêncio dos sinais.

4.2.3.1 CNN com remoção do silêncio e distensão temporal - Base I

Para os testes na base I, foi feita a validação cruzada mencionada anteriormente e os dados foram separados em 5 *folds*.Para cada *fold* foi plotada uma matriz de confusão a fim de identificar os erros de classificação que o sistema obteve.

A Figura 36 mostra os resultados obtidos no reconhecimento do sinal de fala para a base de palavras com ambas as técnicas combinadas.

Figura 36 – Matriz confusão para o experimento da rede com remoção do silêncio e distensão temporal - Base I - Palavras.



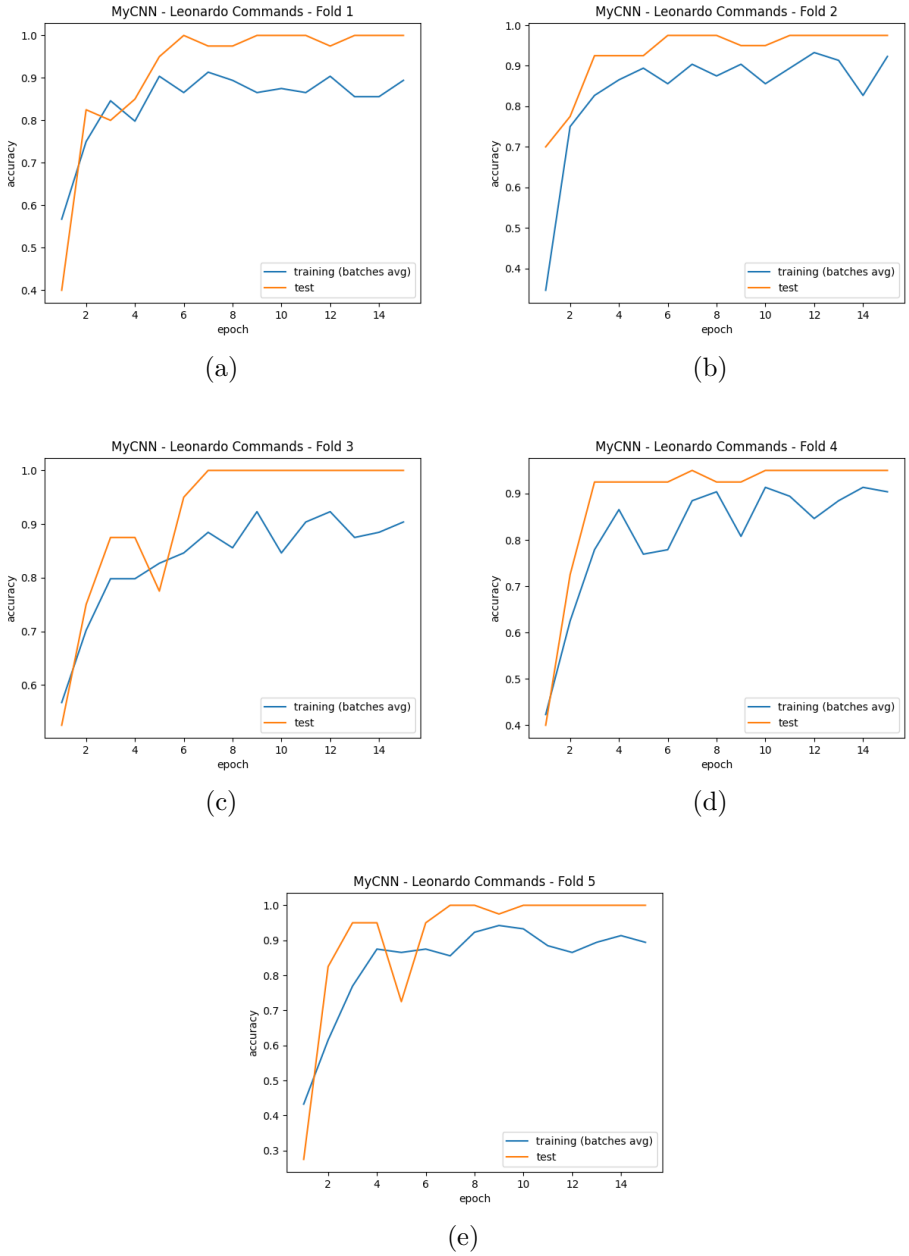
Fonte: Do autor.

Como pode ser visto nas Figuras 36(b) e 36(d) os únicos erros que a rede neural convolucional cometeu foram a mesma confusão observada nos experimentos anteriores. A classificação errada entre as palavras 'Sala' e 'Quarto' representou aqui 100% dos erros do sistema. Ao considerar todas as *folds*, dos 200 testes realizados, o sistema cometeu este erro um total de 3 vezes. Isso pode identificar uma proximidade em algumas das características que compõem estas classes, uma vez que nenhuma outra confusão ocorreu no experimento sob os dados crus.

Foram executadas 15 épocas na etapa de treinamento para que a rede neural

melhorasse o seu desempenho com a retropropagação. A Figura 37 mostra o desempenho da rede neural convolucional com ambas técnicas aplicadas sob dados de entrada.

Figura 37 – Taxa de acerto do modelo CNN scom remoção do silêncio e distensão temporal - Base I - Palavras.



(a) (b) (c) (d) (e)

Fonte: Do autor.

Como pode ser observado na Figura 37, a rede neural projetada sob os dados alcança uma taxa de acerto aproximadamente de 0,8 em poucas épocas e em 3 das 5 *folds* o sistema atinge a taxa de acerto de 1 por volta da décima época. Isso indica uma rápida adequação dos neurônios sob os dados tratados. Após remoção de silêncio e distensão temporal aplicadas, a rede, através da retropropagação, consegue fazer um rápido ajuste

dos filtros (neurônios da rede) para que os mesmos aprendam a classificar corretamente as classes do modelo.

A Tabela 21 mostra os resultados obtidos em cada *fold* e a taxa de acerto média do modelo com a classificação sob os dados tratados para esta base foi de 98,5%.

Tabela 21 – CNN com remoção do silêncio e distensão temporal - Base I - Palavras.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	39	1	0,975
3	40	0	1,000
4	38	2	0,950
5	40	0	1,000
Total	**197**	**3**	**0,985**

Fonte: Do autor.

Durante os experimentos realizados neste trabalho, observou-se que o modelo de rede neural convolucional obteve seu melhor desempenho ao ser treinado com espectrogramas construídos a partir de sinais de áudio com a aplicação de remoção de silêncio e distensão temporal. A taxa de acerto atingiu 98,5% nesse cenário, superando os resultados obtidos com as técnicas de pré-processamento sendo aplicadas separadamente, que variaram entre 93,5% e 94,5%.

Os resultados indicados na Tabela 22 que a combinação das técnicas de remoção de silêncio e distensão temporal levou ao melhor desempenho da rede convolucional. A remoção do silêncio contribuiu para eliminar trechos com baixa relevância acústica, reduzindo redundâncias e aumentando a relação sinal-ruído, enquanto a distensão temporal atuou como uma forma de *data augmentation*, ampliando a diversidade do conjunto de treinamento e tornando o modelo mais robusto a variações na velocidade de fala entre diferentes locutores. Assim, a aplicação conjunta dessas técnicas produziu entradas mais consistentes e informativas para a CNN, favorecendo a extração de padrões espectro-temporais mais discriminativos e, consequentemente, resultando em uma maior taxa de acerto na tarefa de classificação.

Tabela 22 – Taxa de acerto final do modelo - Base I - Palavras.

Experimento	**Taxa de acerto do modelo**
CNN com Remoção do silêncio	0,940
CNN com Distensão temporal	0,980
CNN com remoção do silêncio e distensão temporal	0,985

Fonte: Do autor.

Este modelo de rede foi projetado com os hiperparâmetros ajustados para fazer a classificação de um número baixo de classes e uma base de dados relativamente pequena

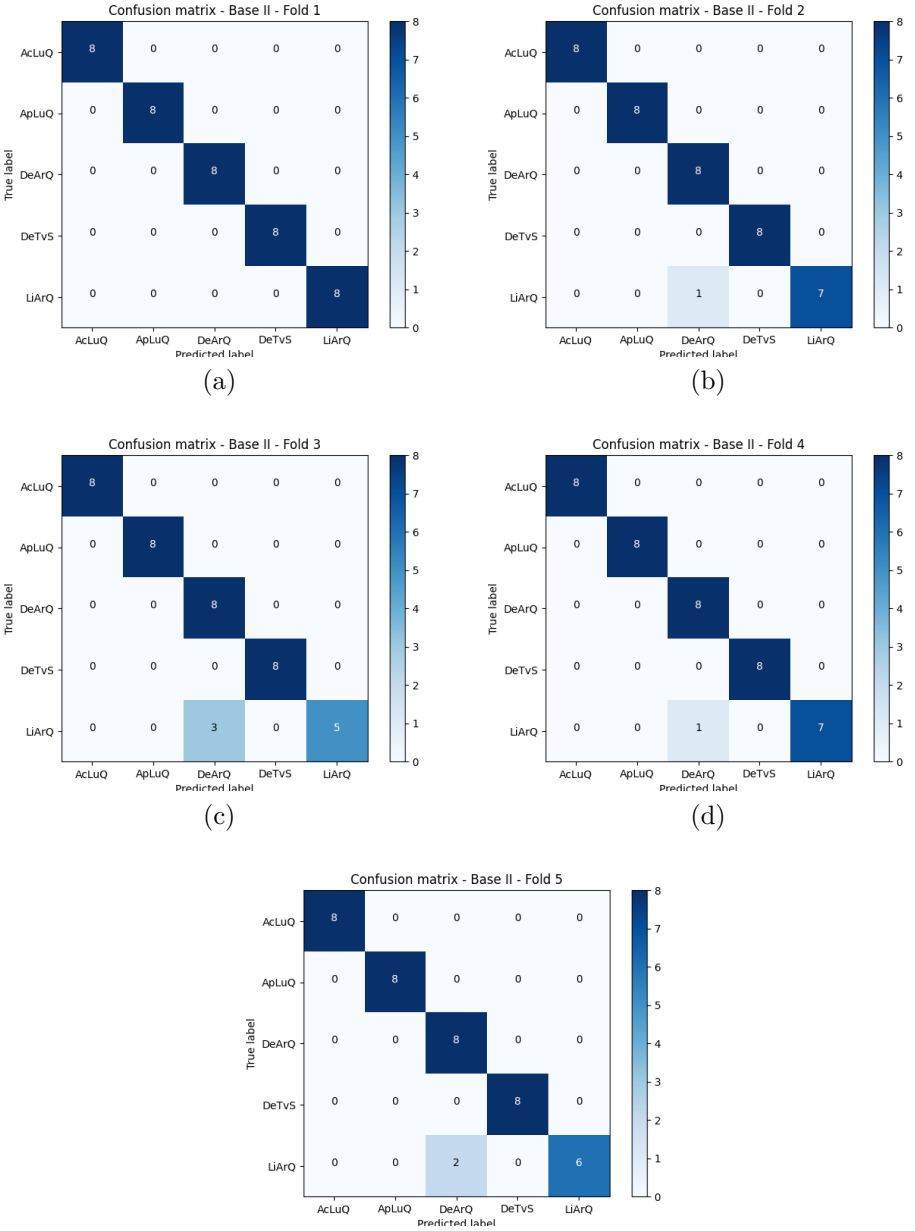
de 150 sinais de áudio. A rápida adequação da rede ao treinamento para chegar em taxa de acertos elevadas mostra um resultado da rede satisfatório na classificação de palavras e frases.

4.2.3.2 CNN com remoção do silêncio e distensão temporal - Base II

Para os testes na Base II, os dados tratados foram separados em 5 *folds* para realizar a validação cruzada. Para cada *fold* foi plotada uma matriz de confusão a fim de identificar os erros de classificação que o sistema obteve.

A Figura 38 mostra os resultados obtidos no reconhecimento do sinal de fala para a base de palavras com remoção do silêncio e distensão temporal.

Figura 38 – Matriz confusão para o experimento da rede com remoção do silêncio e distensão temporal - Base II - Frases.



(a) (b) (c) (d) (e)

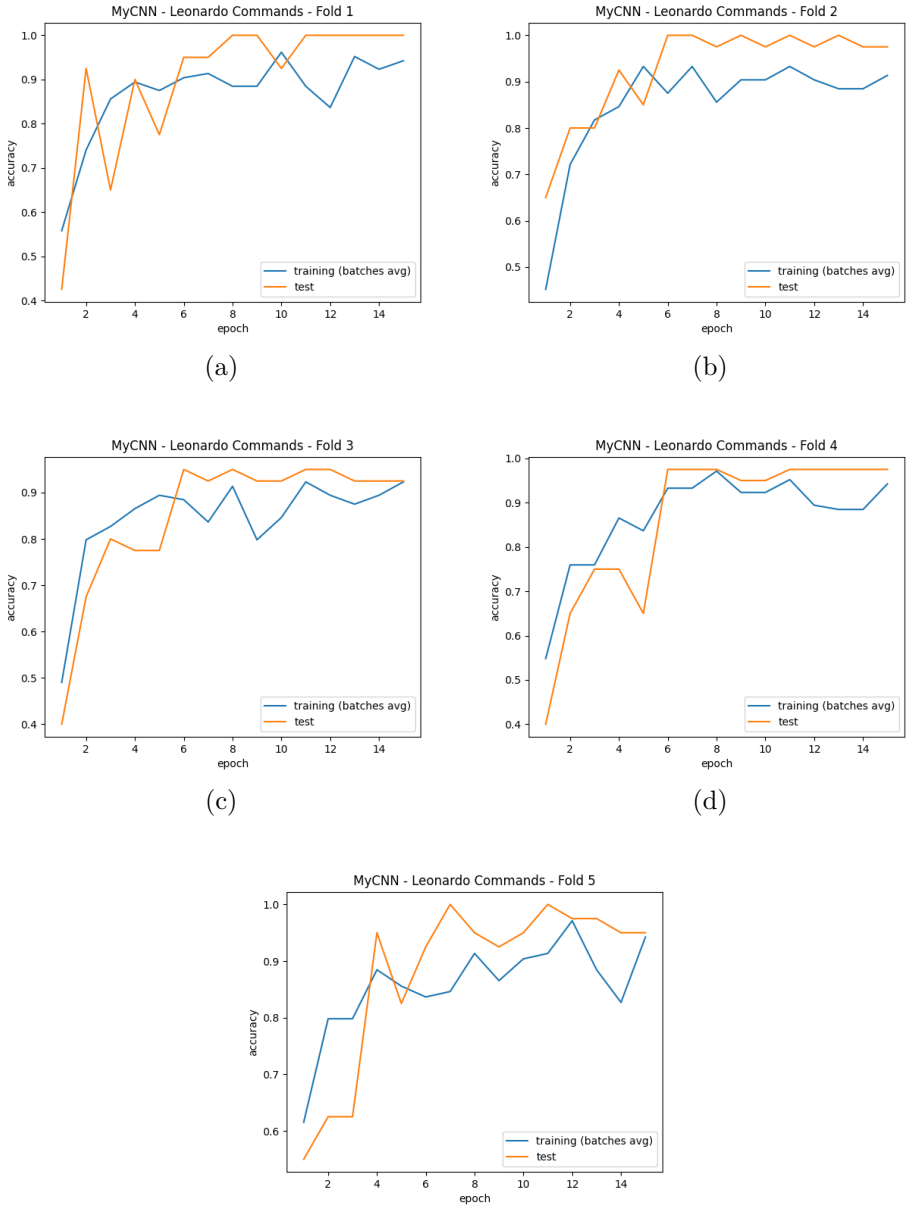
Fonte: Do autor.

Como pode ser visto na Figura 38, com exceção da *fold* 1, todas as *folds* cometeram a confusão entre a entrada da classe "Ligar Ar Quarto"pela classe "Desligar Ar Quarto". No total, 7 erros com esta confusão foram evidenciados ao longo dos 200 treinamentos. Assim como nos demais experimentos, esta confusão pode ser explicada pela similaridade entre as classes e, consequentemente, pela similaridade entre as características extraídas dos sinais de áudio destas frases.

Foram executadas 15 épocas à etapa de treinamento para que a rede neural melhorasse o seu desempenho com a retropropagação sob os dados tratados. A Figura 39

mostra o desempenho da rede neural convolucional com tratamento dos dados de entrada.

Figura 39 – Taxa de acerto do modelo CNN com remoção do silêncio e distensão temporal - Base II - Frases.



(a) (b) (c) (d) (e)

Fonte: Do autor.

Como pode ser observado na Figura 39, assim como nos experimentos anteriores, a rede neural projetada sob os dados tratados precisa de um número maior de épocas para atingir uma mesma taxa de acerto quando se compara o desempenho da base I e da base II. A complexidade de uma frase é maior que a de uma palavra e, com isso, os neurônios da rede necessitam de mais treinamento para identificar as características de mais alto nível.

A Tabela 23 mostra os resultados obtidos em cada *fold* e a taxa de acerto média do modelo atuando sob os dados tratados para a Base II foi de 96,5%.

Tabela 23 – CNN com remoção do silêncio e distensão temporal - Base II - Frases.

Fold	**Acertos**	**Erros**	**Taxa de acerto**
1	40	0	1,000
2	39	1	0,975
3	37	3	0,925
4	39	1	0,975
5	38	2	0,950
Total	**193**	**7**	**0,965**

Fonte: Do autor.

Este modelo de rede atuando sob os dados após a aplicação das técnicas de remoção do silêncio e distensão temporal mostrou a melhor taxa de acerto para a base de palavras (Base II) com um aumento de 3% em relação ao experimento de remoção de silêncio e um aumento de 2% em relação ao experimento de distensão temporal. A Tabela 24 apresenta uma comparação da taxa de acerto da rede neural convolucional para a base de frases e evidencia o melhor resultado para o modelo atuando sob os dados com ambas técnicas aplicadas aos sinais de áudio.

Tabela 24 – Taxa de acerto final do modelo - Base II - Frases.

Experimento	**Taxa de acerto do modelo**
CNN com Remoção do silêncio	0,935
CNN com Distensão temporal	0,945
CNN com remoção do silêncio e distensão temporal	0,965

Fonte: Do autor.

Assim como para a base I, o modelo de rede foi projetado com os hiperparâmetros ajustados para fazer a classificação de um número baixo de classes e uma base de dados de 150 sinais de áudio, onde 110 vão para o treinamento e 40 para a etapa de testes. Após separar 26% dos dados para teste, a rede neural tem um baixo número de amostras para realizar o treinamento e aprender os padrões da rede, visto que cada sinal de áudio foram gravados somente 30 elocuções.

Para o reconhecimento de frases, a complexidade e nível das classes se tornam mais altos para que a rede faça a classificação. Em comparação à base I, a rede apresentou um melhor desempenho com a classificação de palavras devido ao nível de características (menor número de fonemas, duração de fala menor, menos pitches) ser de menor complexidade.

A rápida adequação da rede ao treinamento para chegar em taxa de acertos elevadas mostra um resultado satisfatório da rede para reconhecimento de palavras, com

exceção dos erros de frases com alta verossimilhança. A classificação com uma taxa de acerto de 96,5% em cima de dados tratados mostrou um bom desempenho do sistema.

4.3 Considerações parciais

Comparando-se os resultados obtidos uma comparação entre o modelo oculto de Markov (HMM) e o modelo rede neural convolucional (CNN), pode-se perceber a adequação de cada modelo às bases I e II e como os modelos se comportaram para a base de palavras e a base de frases.

A Tabela 25 mostra as taxa de acertos obtidas em cada modelo para cada base de palavras. Como podemos perceber, o modelo HMM se mostrou mais eficaz para as classes de palavras e para a classe de frases, contidas na Base I e na Base II, respectivamente.

Tabela 25 – Comparativo entre os modelos HMM e CNN.

	HMM	**CNN**
Base I	99,5%	98,5%
Base II	100%	96,5%

Fonte: Do autor.

O modelo HMM obteve apresentou melhores resultados e classificação para a base de frases em comparação à base de palavras. A Base II pode ser considerada mais complexa pelo maior número de fonemas e similaridade de alguns dentro dessa classe. Já o modelo CNN obteve a melhor taxa de acerto para a base de palavras e uma menor taxa de acerto para a base de frases.

As matrizes de confusão obtidas nos experimentos da base I mostraram que o modelo HMM cometeu todos os erros na classificação entre 'Apagar' e 'Cozinha' enquanto o modelo CNN cometeu todos os seus erros de classificação na confusão entre as classes 'Sala' e 'Quarto'. Uma vez que a mesma base de dados foi utilizada nos dois modelos, não podemos atrelar os erros do sistema à base de áudios gravados. O modelo HMM se mostrou mais eficaz, com uma taxa de acerto de 99,5%.

Para os experimentos realizados com a base II, as matrizes de confusão de ambos os modelos HMM e CNN identificaram todos os erros na confusão entre as mesmas classes de 'Ligar Ar Quarto' e 'Desligar Ar Quarto'. Uma vez que o modelo HMM atingiu a taxa de acerto de 100% para a base II e essa foi a mesma utilizada no modelo CNN, não podemos atrelar os erros do sistema à base de áudios gravada. O modelo HMM se mostrou novamente mais eficaz no reconhecimento de fala no presente trabalho.

5 Conclusão e trabalhos futuros

No presente trabalho foram realizados experimentos em dois modelos de reconhecimento da fala a fim de apresentar um estudo comparativo entre os mesmos. Foi implementado um sistema de reconhecimento de fala baseado no HMM através do aprendizado de máquina e um sistema de reconhecimento baseado nas redes neurais CNN através do aprendizado profundo.

Para os experimentos de classificação utilizando o HMM, a remoção de silêncio se mostrou crucial ao funcionamento do modelo e foi observado que o número de gaussianas deste modelo define o comportamento dele no reconhecimento tanto na base I que é composta de palavras quanto na base II composta por frases curtas. Para uma base de áudios de menor complexidade, o número de 5 gaussianas foi utilizado, levando em consideração a não necessidade de alta complexidade e alto custo operacional.

Para os experimentos de classificação utilizando o CNN, a remoção do ruído e a distensão temporal foram utilizadas, mas o melhor desempenho encontrado foi em cima dos dados que têm as duas técnicas combinadas. Após a inserção dos sinais de áudio na rede neural, a rede mostrou uma rápida adaptação dos parâmetros dos neurônios durante as etapas de treinamento.

Por fim, os resultados obtidos pelos experimentos apresentados no Capítulo 4 mostraram que ambos os modelos atingiram uma taxa de acerto média de 98,65%. Isso significa ter, em média, 3 erros de classificação a cada 200 elocuções inseridas nos sistemas. Isso o torna eficaz e aplicável aos sistemas de reconhecimento de fala propostos. O modelo HMM se mostrou mais eficaz do que o modelo CNN para ambas as bases e chegou a atingir a acurácia de 100% com custo operacional baixo para a Base II.

5.1 Etapas futuras

No sistema de reconhecimento dependente de locutor, todo treinamento deve ser realizado estritamente com sinais de áudio provenientes deste locutor. Tal fato limita algumas aplicações quando se deseja atrelar uma ação ou comando a qualquer indivíduo com acesso a esta aplicação. Como proposta para os próximos trabalhos, a aplicação destes sistemas em reconhecimento de fala independente de locutor.

Uma segunda proposta de continuidade é a aplicação dos modelos a bases de dados maiores para verificação do fenômeno de *overfitting*. A utilização de uma base maior pode obter uma acurácia mais elevada, evidenciando a influência do *overfitting* nos modelos aplicados.

Por fim, a possível integração do classificador treinado a um sistema prático que acione dispositivos com comandos reais de voz valida a aplicação em contexto real.

Referências

AMOOLYA, G. et al. Automatic speech recognition for tulu language using gmm-hmm and dnn-hmm techniques. *International Conference on Advanced Computing Technologies and Applications (ICACTA)*, 2022. doi: 10.1109/ICACTA54488.2022.9753319.

ANNAS, M. E.; OUZINEB, M.; BENYACOUB, B. Hidden markov models training using hybrid baum welch - variable neighborhood search algorithm. *Statistics, Optimization and Information Computing*, v. 10, n. 1, p. 160–170, 2022. doi: 10.19139/soic-2310-5070-1213.

ANUJA, V.; AKSHATHA; JAYAPRAKASH. Voice to text using asr and hmm. *International Journal For Science Technology And Engineering*, 2022. doi: 10.22214/ijraset.2022.44803.

HAMIDI, M. et al. Interactive administration service based on hmm speech recognition system. *International Journal of Computer Aided Engineering and Technology*, 2021. doi: 10.1504/ijcaet.2022.120819.

HINTON, G. et al. Dropout: A simple way to prevent neural networks from overfitting. *Department of Computer Science University of Toronto*, 2014. doi: 10.55041/ijsrem30472.

HOMMA, T. et al. Speech recognition apparatus and speech recognition system. *Science and Technical Research Laboratories NHK (Nippon Hoso Kyokai; Japan Broadcasting Corp.)*, 2019.

HOW, C. K. et al. Development of audio-visual speech recognition using deep-learning technique. *Mekatronika*, 2022. doi: 10.15282/mekatronika.v4i1.8625.

HUANG, C.; ZHU, Z.; GUO, J. Investigations of hmm-based speech recognition technology. *IEEE*, p. 74–77, 2020. doi: 10.1109/IWECAI50956.2020.00021.

KOVALESKI, P. de A. Implementação de redes neurais profundas para reconhecimento de ações em vídeo. *Universidade Federal do Rio de Janeiro*, 2018.

MUNIR, A.; KONG, J.; QURESHI, M. A. Overview of convolutional neural networks. *Wiley-IEEE Press*, 2023. doi: 10.1002/9781394171910.ch2.

PING, L. English speech recognition method based on hmm technology. *IEEE*, 2021. doi: 10.1109/ICITBS53129.2021.00164.

RABINER, L. R.; JUANG, B.-H. *Historical Perspective of the Field of ASR/NLU*. [S.l.]: Pearson Education, 2007. 193 p. doi: 10.1007/978-3-540-49127-9$_2$6.

SAKOE, H. Speech recognition system. *Journal of the Acoustical Society of America*, 1978. doi: 10.1121/1.391520.

SINGH, S. The role of speech technology in biometrics, forensics and man-machine interface. *International Journal of Electrical and Computer Engineering*, 2019. doi: 10.11591/IJECE.V9I1.PP281-288.

SOUNDARYA, M.; KARTHIKEYAN, P.; THANGARASU, G. Automatic speech recognition trained with convolutional neural network and predicted with recurrent neural network. *International Conference on Electrical Engineering and System*, 2023. doi: 10.1109/ICEES57979.2023.10110224.

VANNESCHI, L.; SILVA, S. Artificial neural networks. *Natural computing series*, 2022. doi: 10.1007/978-3-031-17922-8$_7$.

WASSNER, H.; CHOLLET, G. *New time-frequency derived cepstral coefficients for automatic speech recognition*. 1996.

ZHAN, C.; XIN, Q. Speech recognition method and device and speech recognition system. *Science and Technical Research Laboratories NHK (Nippon Hoso Kyokai; Japan Broadcasting Corp.)*, 2020.

MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Colegiado do Curso de Engenharia Elétrica

TERMO DE RESPONSABILIDADE

O texto do trabalho de conclusão de curso intitulado Aplicação do modelo oculto de Markov e da rede neural convolucional em um sistema de reconhecimento de fala automático é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico ou qualquer outro material pertencente a terceiros sem a devida citação ou consentimento dos referidos autores.

João Monlevade, 11 de janeiro de 2026.

Leonardo Castro Souza Marotta

MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Colegiado do Curso de Engenharia Elétrica

DECLARAÇÃO DE CONFERÊNCIA DA VERSÃO FINAL

Declaro que conferi a versão final a ser entregue pelo aluno Leonardo Castro Souza Marotta, autor do trabalho de conclusão de curso intitulado Aplicação do modelo oculto de Markov e da rede neural convolucional em um sistema de reconhecimento de fala automático quanto à conformidade nos seguintes itens:

1. A monografia corresponde a versão final, estando de acordo com as sugestões e correções sugeridas pela banca e seguindo as normas ABNT;
2. A versão final da monografia inclui a ata de defesa (Anexo IV), a ficha catalográfica e o termo de responsabilidade (ANEXO X) devidamente assinado.

João Monlevade, 11 de janeiro de 2026.

Prof. Orientador Glauco Ferreira Gazel Yared