



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
ENGENHARIA ELÉTRICA - ELETRÔNICA

**TRADUÇÃO AUTOMÁTICA DE LIBRAS PARA TRIAGEM
HOSPITALAR: UMA ABORDAGEM COM VISÃO COMPUTACIONAL E
APRENDIZADO PROFUNDO DE MÁQUINA**

Thiago Patrick Tavares Costa

MANAUS-AM

2025

Thiago Patrick Tavares Costa

TRADUÇÃO AUTOMÁTICA DE LIBRAS PARA TRIAGEM HOSPITALAR: UMA
ABORDAGEM COM VISÃO COMPUTACIONAL E APRENDIZADO PROFUNDO
DE MÁQUINA

Monografia apresentada à Coordenação do
Curso de Engenharia Elétrica - Eletrônica
da Universidade Federal do Amazonas, como
parte dos requisitos necessários à obtenção
do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Manuel Augusto Pinto Cardoso

MANAUS-AM

2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

- C837t Costa, Thiago Patrick Tavares
Tradução automática de LIBRAS para triagem hospitalar: uma abordagem com visão computacional e aprendizado profundo de máquina / Thiago Patrick Tavares Costa. - 2025.
54 f. : il., color. ; 31 cm.
- Orientador(a): Manuel Augusto Pinto Cardoso.
Trabalho de Conclusão de Curso (graduação) - Universidade Federal do Amazonas, Faculdade de Tecnologia, Curso de Engenharia Elétrica, Manaus, 2025.
1. Acessibilidade . 2. Deep Learning. 3. GRU. 4. LIBRAS. 5. Visão Computacional. I. Cardoso, Manuel Augusto Pinto. II. Universidade Federal do Amazonas. Faculdade de Tecnologia. Curso de Engenharia Elétrica. III. Título
-

Dedico este trabalho aos meus pais, Carlos Costa e Adriana Tavares. A meu pai, que na juventude abdicou de noites de sono, dividindo-se entre trabalho e estudos, para que eu pudesse sonhar com tranquilidade. À minha mãe, que construiu estradas firmes com coragem e amor, para que eu pudesse engatinhar sem ralar os joelhos. A eles, que sempre transformaram pedras em caminhos com o suor do esforço, para que eu pudesse trilhar uma estrada de luz. Ao meu avô, Augusto Tavares, que não pôde ver seu neto se formar, mas cuja memória me acompanha em cada conquista. Aos meus amigos, que me ensinaram o valor do companheirismo e estiveram comigo em cada etapa desta caminhada. E, com todo o meu coração, a Deus — por cada novo dia, por cada oportunidade e por nunca me deixar só.

Agradecimentos

Quero expressar minha imensa felicidade ao agradecer primeiramente a Deus por me conceder a conclusão deste trabalho e do curso de Engenharia Elétrica. À minha família, que sempre esteve ao meu lado, proporcionando apoio e condições ideais para que eu me dedicasse completamente aos estudos, meu profundo reconhecimento.

Aos amigos que fiz durante esse percurso, minha sincera gratidão. Suas contribuições e apoio foram essenciais para que eu chegasse até aqui. Em especial, quero destacar os amigos da turma de 2019/1, cujo espírito de equipe tornou essa jornada mais leve e significativa.

Ao Prof. Manuel Cardoso, agradeço por aceitar me orientar nesta jornada e por compartilhar seus ensinamentos. À banca examinadora e a todos os professores que contribuíram para o meu crescimento acadêmico, meus sinceros agradecimentos.

“Imagination is more important than knowledge. For knowledge is limited to all we now know and understand, while imagination embraces the entire world, and all there ever will be to know and understand.”

(Albert Einstein)

Resumo

Esta monografia apresenta o desenvolvimento e a validação de um sistema de tradução automática da Língua Brasileira de Sinais (LIBRAS) para a língua portuguesa, focado especificamente no contexto de triagem médica hospitalar. A barreira comunicativa entre pacientes surdos e profissionais de saúde ouvintes compromete a qualidade do atendimento e a autonomia do paciente. Para mitigar esse problema, propõe-se uma arquitetura de visão computacional baseada em aprendizado profundo (*Deep Learning*) capaz de interpretar sinais dinâmicos em tempo real utilizando câmeras convencionais. A metodologia emprega a biblioteca *MediaPipe Holistic* para a extração de vetores de características baseados em coordenadas esqueléticas de pose, face e mãos, eliminando a dependência de processamento de imagem bruta e garantindo eficiência computacional. A classificação temporal dos sinais é realizada por uma Rede Neural Recorrente do tipo GRU (*Gated Recurrent Unit*), selecionada por sua superioridade em relação à LSTM nos testes preliminares. Para validar a robustez e a capacidade de generalização do sistema, foi criado um conjunto de dados exclusivo com a participação de 30 voluntários, incluindo surdos e deficientes auditivos. A avaliação utilizou o rigoroso protocolo estatístico *Leave-One-Subject-Out Cross-Validation* (LOSO-CV), simulando um cenário real onde o sistema opera com usuários desconhecidos. Os resultados obtidos demonstram uma acurácia média global de 94% e um F1-Score médio de 0,94. O sistema demonstrou viabilidade prática para ser implantado em hospitais utilizando hardware acessível e de fácil aquisição, contribuindo para a democratização da tecnologia assistiva.

Palavras-chave: Acessibilidade, *Deep Learning*, GRU, LIBRAS, Visão Computacional.

Abstract

This monograph presents the development and validation of an automatic translation system from Brazilian Sign Language (LIBRAS) to Portuguese, specifically focused on the context of hospital medical triage. The communication barrier between deaf patients and hearing healthcare professionals compromises the quality of care and patient autonomy. To mitigate this issue, a computer vision architecture based on Deep Learning is proposed to interpret dynamic signs in real-time using conventional cameras. The methodology employs the MediaPipe Holistic library for the extraction of feature vectors based on skeletal coordinates of the pose, face, and hands, eliminating the dependence on raw image processing and ensuring computational efficiency. The temporal classification of signs is performed by a Gated Recurrent Unit (GRU) Neural Network, selected for its superiority over LSTM in preliminary tests. To validate the robustness and generalization capability of the system, a unique dataset was created with the participation of 30 volunteers, including deaf and hard-of-hearing individuals. The evaluation used the rigorous statistical protocol Leave-One-Subject-Out Cross-Validation (LOSO-CV), simulating a real-world scenario where the system operates with unknown users. The obtained results demonstrate a global average accuracy of 94% and an average F1-Score of 0.94. The system demonstrated practical viability for deployment in hospitals using accessible and easily available hardware, contributing to the democratization of assistive technology.

Keywords: Accessibility, Computer Vision, Deep Learning, GRU, LIBRAS.

Lista de Figuras

1.1	Fluxograma ilustrando a barreira de comunicação e o risco associado.	2
1.2	Visão geral da solução proposta: do sinal visual à tradução textual.	3
2.1	Neurônio biológico	7
2.2	Representação gráfica das portas lógicas	8
2.3	Representação lógica de um neurônio artificial	8
2.4	Representação do Perceptron	9
2.5	Rede neural artificial	10
2.6	Ilustração do Forward Pass e Backward Pass	12
2.7	Córtex visual	16
4.1	Diagrama de fluxo do sistema de interpretação de LIBRAS.	27
4.2	Pipeline detalhada do processamento de dados do sistema.	27
4.3	Extração dos keypoints com MediaPipe	29
5.1	Curvas de aprendizado do modelo LSTM.	38
5.2	Curvas de aprendizado do modelo GRU.	38
5.3	Resultados preliminares LSTM.	38
5.4	Resultados preliminares GRU.	38
5.5	Sistema em operação durante a fase de testes e coleta de dados.	39
5.6	Representação do método de treinamento LOSO-CV	42
5.7	Matriz de Confusão Global acumulada (Soma das 30 rodadas LOSO).	44
5.8	Distribuição da acurácia individual nos 30 folds do experimento LOSO.	45

Lista de Tabelas

4.1	Classificação metodológica das etapas do desenvolvimento.	36
4.2	Fonte - Autor.	36
5.1	Categorização dos erros observados no teste piloto (Modelo Mono-Usuário).	40
5.2	Métricas detalhadas por classe após validação LOSO-CV (Média de 30 Rodadas).	42
5.3	Comparativo de Desempenho: Piloto (Estimado) vs. Validação LOSO.	43

Lista de Abreviaturas e Siglas

LIBRAS	Língua Brasileira de Sinais
LSTM	Long Short-Term Memory (Memória de Longo e Curto Prazo)
GRU	Gated Recurrent Unit (Unidade Recorrente com Portas)
LOSO-CV	Leave-One-Subject-Out Cross-Validation
CNN	Convolutional Neural Network (Rede Neural Convolutacional)
RGB	Red, Green, Blue (Espaço de cor)
FPS	Frames Per Second (Quadros por Segundo)
API	Application Programming Interface
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico)
HDF5	Hierarchical Data Format version 5
FACS	Facial Action Coding System
TFLite	TensorFlow Lite

Lista de Símbolos

N	Número total de participantes (30)
n	Tamanho da amostra estatística
t	Passo de tempo (<i>time step</i>) na sequência de vídeo
x_t	Vetor de características de entrada no instante t
h_t	Estado oculto (<i>hidden state</i>) da rede recorrente
σ	Função de ativação sigmoide
\tanh	Função de ativação tangente hiperbólica
TP	Verdadeiros Positivos (<i>True Positives</i>)
TN	Verdadeiros Negativos (<i>True Negatives</i>)
FP	Falsos Positivos (<i>False Positives</i>)
FN	Falsos Negativos (<i>False Negatives</i>)
$F1$	Pontuação F1-Score (Média harmônica)

Sumário

1	Introdução	1
1.1	Objetivo Geral	3
1.2	Objetivos Específicos	3
1.3	Estrutura do Trabalho	4
2	Referencial Teórico	6
2.1	O Modelo do Cérebro Artificial: Um Breve Histórico do Aprendizado Profundo	6
2.1.1	A Inspiração Biológica: O Neurônio	6
2.1.2	O Primeiro Neurônio Artificial: O Modelo de McCulloch-Pitts	7
2.1.3	A Evolução para o Aprendizado: O Perceptron de Rosenblatt	9
2.1.4	Limitações e o Primeiro “Inverno da IA”	10
2.1.5	O Problema do Treinamento e a Solução Oculta na Astronomia	10
2.1.6	O Renascimento com a Retropropagação	11
2.1.7	Desafios das Redes Profundas e a Ascensão de Outros Métodos	12
2.1.8	A Revolução do Aprendizado Profundo	13
2.2	Contexto histórico da LIBRAS	14
2.3	Cenário Atual das LIBRAS no Brasil	15
2.4	Fundamentação técnica	16
2.4.1	Visão Computacional	16
2.4.2	Aprendizado Profundo para Processamento de Sequências	18
2.4.2.1	A Arquitetura das Redes Neurais Recorrentes (RNNs)	19
2.4.2.2	O Desafio dos Gradientes em Sequências Longas	19
2.4.2.3	Componentes Essenciais: Funções de Ativação Sigmoide e Tangente Hiperbólica	20

2.4.2.4	A Arquitetura LSTM e a Solução para o Fluxo do Gradiente	20
2.4.2.5	A Arquitetura GRU: Uma Alternativa Eficiente	21
3	Revisão de Trabalhos Correlatos	23
3.1	Abordagens para Sinais Estáticos e Isolados	23
3.2	Arquiteturas Híbridas para Sinais Dinâmicos	24
3.3	Sistemas em Tempo Real e a Lacuna no Contexto de Uso	24
4	Metodologia	26
4.1	Visão Geral do Sistema	26
4.2	Coleta e Preparação dos Dados	28
4.2.1	Definição do Vocabulário e Gravação	28
4.2.2	Normalização Temporal e Justificativa Técnica	28
4.3	Extração de Características com MediaPipe	29
4.3.1	Justificativa da Biblioteca MediaPipe	29
4.3.2	Vetor de Características e Formato HDF5	29
4.4	Arquitetura e Treinamento do Modelo LSTM	30
4.4.1	Justificativa da API Keras e da Arquitetura	30
4.4.2	Justificativa dos Parâmetros de Treinamento	31
4.5	Sistema de Inferência e Avaliação de Desempenho	31
4.5.1	Lógica de Inferência	32
4.5.2	Definição e Justificativa das Métricas de Avaliação	32
4.5.2.1	Acurácia (Accuracy)	32
4.5.2.2	Precisão (Precision)	33
4.5.2.3	Sensibilidade (Recall)	33
4.5.2.4	F1-Score	33
4.6	Ferramentas e Ambiente de Desenvolvimento	34
4.7	Definição do Vocabulário e Restrições Iniciais	34
4.7.1	Normalização Temporal e Pré-processamento	35
4.7.2	Caracterização do Dataset	35
4.8	Caracterização da Pesquisa	35
5	Resultados e Discussão	37
5.1	Fase 1: Seleção de Modelo e Treinamento Inicial	37

5.1.1	Comparativo LSTM vs. GRU	37
5.2	Fase 2: Teste de Campo Piloto e Análise Qualitativa	38
5.2.1	Observações de Campo e Falhas de Generalização	39
5.3	Fase 3: Validação Robusta (LOSO-CV)	40
5.3.1	Justificativa Estatística do Tamanho da Amostra	41
5.3.2	Ambiente Computacional	41
5.3.3	Resultados Quantitativos do LOSO	41
5.3.4	Discussão da Evolução do Sistema	44
5.4	Conclusão dos Resultados	45
6	Considerações Finais	46
6.1	Trabalhos Futuros	47
	Referências Bibliográficas	49

Capítulo 1

Introdução

Uma sociedade inclusiva depende de mecanismos que permitam a todos os seus membros participar ativamente das atividades cotidianas; contudo, no Brasil, mais de dez milhões de pessoas convivem com algum grau de surdez, sendo aproximadamente 2,7 milhões com perda auditiva severa ou profunda ([Universidade de São Paulo, 2023](#); [Agência Brasil, 2022](#)). Ainda que a *Lei n.º 10.436/2002* reconheça oficialmente a Língua Brasileira de Sinais (LIBRAS) ([Brasil, 2002](#)) e que iniciativas governamentais celebrem quase duas décadas de sua promulgação ([Casa Civil da Presidência da República, 2021](#)), a falta de difusão dessa língua entre ouvintes mantém barreiras comunicativas que limitam o acesso a educação, saúde e mercado de trabalho para a comunidade surda.

A escassez de intérpretes qualificados agrava esse cenário: auditorias em ambientes escolares, hospitalares e institucionais revelam a persistência de atendimentos improvisados, frequentemente mediados por familiares ou cuidadores não habilitados, comprometendo a qualidade da informação transmitida ([Ribeiro et al., 2019](#)). Quando presentes, intérpretes costumam atender simultaneamente múltiplas demandas, resultando em sobrecarga profissional e em filas de espera que atrasam consultas, matrículas ou procedimentos essenciais ([Barbosa and Oliveira, 2024](#)).

A dimensão tecnológica também reflete desigualdades, pois grande parte dos sistemas digitais — sejam interfaces de serviços públicos, sejam aplicativos privados — foi concebida prioritariamente para usuários ouvintes. Sem recursos de acessibilidade adequados, pessoas surdas acabam recorrendo à escrita em português, sua segunda língua, ou à presença eventual de intérpretes, perpetuando dependência e isolamento ([Ribeiro et al., 2019](#)).

Nos últimos anos, avanços em visão computacional e aprendizado profundo (*deep learning*) demonstraram potencial para reduzir esse hiato: modelos baseados em redes neurais profundas já exibem desempenho expressivo no reconhecimento de sinais (de Avellar Sarmiento and de Avellar Sarmiento, 2023) e na tradução automática de língua de sinais em diversos países, impulsionados por revisões sistemáticas e melhorias arquiteturais (Zhang and Jiang, 2024).

Frameworks otimizados, como o *MediaPipe*, possibilitam a segmentação robusta de mãos, rosto e corpo em tempo real, integrando-se a modelos recorrentes e *Transformers* para capturar a dinâmica temporal de frases completas em LIBRAS (Sitorus and Siregar, 2023). No setor privado, soluções comerciais já exploram reconhecedores bidirecionais baseados em inteligência artificial — um exemplo notável é a plataforma Hand Talk, que vem popularizando tradutores virtuais de sinais para línguas orais e vice-versa (Hand Talk, 2024).

Diante desse panorama, este Trabalho de Conclusão de Curso propõe o desenvolvimento de um interpretador de LIBRAS baseado em visão computacional e redes neurais recorrentes. A solução utiliza câmeras convencionais para capturar a execução de sinais e traduzi-los automaticamente para texto em português, viabilizando uma comunicação fluida em tempo quase real. A arquitetura do sistema baseia-se na extração de pontos de articulação (*keypoints*) via *MediaPipe Holistic*, processados por uma rede GRU capaz de modelar a dinâmica temporal dos sinais.

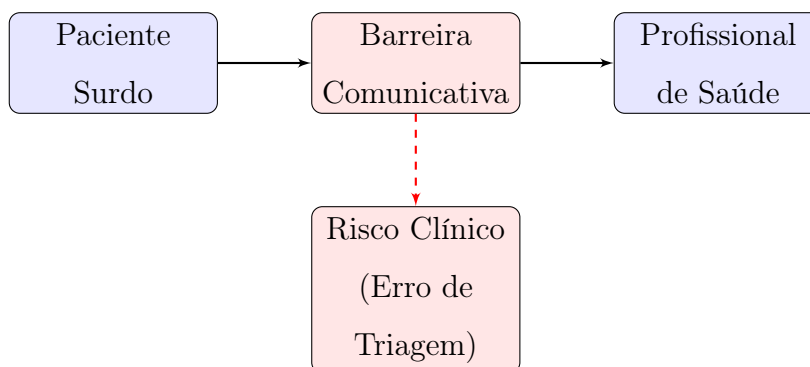


Figura 1.1: Fluxograma ilustrando a barreira de comunicação e o risco associado.

Fonte – Autor.

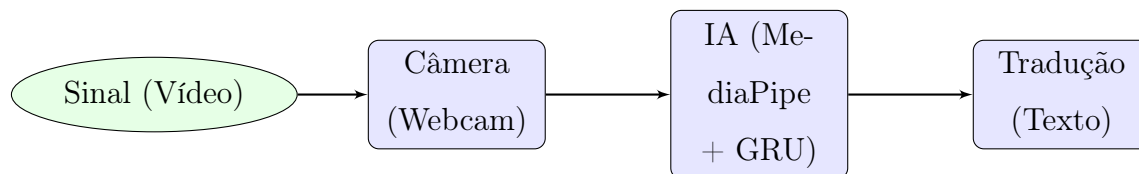


Figura 1.2: Visão geral da solução proposta: do sinal visual à tradução textual.

Fonte – Autor.

Diferentemente de sistemas de tradução genéricos encontrados no estado da arte, que muitas vezes focam em vocabulários amplos, porém superficiais, este trabalho apresenta uma abordagem original ao restringir o domínio para a *triagem médica*. Essa especificidade permite tratar com maior rigor a dificuldade crítica que surdos enfrentam ao tentar comunicar sintomas de dor e desconforto, cenários onde a precisão da informação é vital e o erro de interpretação pode ter consequências graves.

Espera-se que o protótipo resultante, disponibilizado como código aberto, fomente colaborações acadêmicas e industriais, além de contribuir efetivamente para reduzir desigualdades no acesso a serviços essenciais, ampliar a autonomia da pessoa surda e sensibilizar a sociedade quanto à necessidade de tecnologias verdadeiramente inclusivas.

1.1 Objetivo Geral

Desenvolver e validar um interpretador automático de sinais de LIBRAS baseado em visão computacional e *deep learning*, focado em um vocabulário específico para triagem médica hospitalar, capaz de traduzir os sinais para texto em língua portuguesa em tempo quase real, visando mitigar barreiras de comunicação entre pacientes surdos e profissionais de saúde.

1.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Revisar criticamente o estado da arte em reconhecimento de sinais e tradução de línguas de sinais, com ênfase em abordagens que utilizem redes neurais profundas e frameworks de captura de pose.

- Construir um conjunto de dados (*dataset*) próprio, contendo gravações de 30 voluntários (surdos e deficientes auditivos) executando sinais relacionados a sintomas médicos, garantindo variabilidade de biotipos e estilos de sinalização.
- Projetar e implementar uma arquitetura de sistema que combine a extração de características visuais via MediaPipe com modelagem temporal de sequências.
- Realizar uma análise comparativa de desempenho entre diferentes arquiteturas de redes recorrentes (LSTM e GRU) para determinar a mais eficiente para o problema proposto.
- Validar a capacidade de generalização do modelo através da análise comparativa entre treinamentos mono-usuário e a metodologia estatística *Leave-One-Subject-Out Cross-Validation* (LOSO-CV).

1.3 Estrutura do Trabalho

Este documento está organizado em seis capítulos, estruturados da seguinte forma para facilitar a compreensão do desenvolvimento e dos resultados obtidos:

O **Capítulo 2** (Referencial Teórico) apresenta o embasamento necessário para a compreensão do projeto. São abordados aspectos fundamentais da LIBRAS, conceitos de Redes Neurais Artificiais (com foco em arquiteturas recorrentes como LSTM e GRU) e o funcionamento da biblioteca MediaPipe para extração de características biométricas.

O **Capítulo 3** (Revisão de Trabalhos Correlatos) discute o estado da arte na área, analisando abordagens anteriores para o reconhecimento de sinais, desde técnicas para sinais estáticos até sistemas dinâmicos em tempo real, destacando as lacunas que este trabalho busca preencher.

O **Capítulo 4** (Metodologia) detalha a engenharia do sistema proposto. Descreve-se o processo de coleta e normalização do dataset médico, a pipeline de pré-processamento de dados, a arquitetura das redes neurais implementadas e a estratégia de treinamento adotada.

O **Capítulo 5** (Resultados e Discussão) expõe a avaliação experimental do sistema. Apresenta-se o comparativo entre as arquiteturas testadas, a análise qualitativa dos testes de campo e a validação estatística robusta realizada através do método LOSO-CV com

30 participantes, discutindo as métricas de desempenho obtidas.

Por fim, o **Capítulo 6** (Considerações Finais) sintetiza as conclusões do estudo, reafirmando a viabilidade da solução proposta e sugerindo caminhos para trabalhos futuros visando a expansão e aprimoramento da tecnologia desenvolvida.

Capítulo 2

Referencial Teórico

2.1 O Modelo do Cérebro Artificial: Um Breve Histórico do Aprendizado Profundo

A busca pela criação de uma inteligência artificial tem suas raízes na tentativa de modelar o dispositivo mais complexo conhecido: o cérebro humano. Embora filósofos como Gottfried von Leibniz já no século XVII idealizassem um “cálculo de raciocínio” capaz de deduzir verdades universais (Wooldridge, 2021), foi no século XX que a convergência entre a neurociência e a matemática deu origem ao primeiro modelo computacional do cérebro.

2.1.1 A Inspiração Biológica: O Neurônio

Na década de 1940, o conhecimento sobre o sistema nervoso já estabelecia o neurônio como sua unidade fundamental de processamento de informações (Kandel et al., 2013). Embora existam diversos tipos de neurônios, sua estrutura básica é composta por: *dendritos*, que recebem sinais de outras células; o *corpo celular (soma)*, que processa esses sinais; e um *axônio*, que transmite o sinal de saída para os terminais, onde a comunicação com outros neurônios ocorre por meio de sinapses químicas (Gidon et al., 2020).

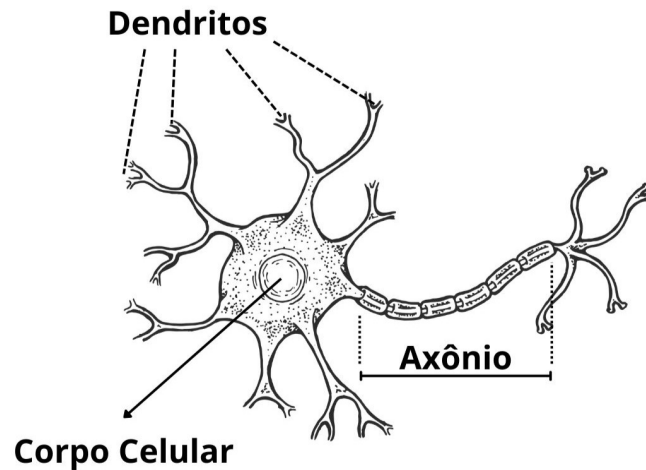


Figura 2.1: Neurônio biológico

Fonte – Autor.

Um único neurônio pode integrar informações de milhares de outras células. Apenas quando a soma dos sinais recebidos em seus dendritos atinge um determinado limiar é que o neurônio “dispara”, propagando um pulso elétrico ao longo de seu axônio. Era a interação entre bilhões dessas unidades que, para pesquisadores como o neuropsiquiatra Warren McCulloch, dava origem à cognição e à inteligência humana ([Anderson and Rosenfeld, 1998](#)).

2.1.2 O Primeiro Neurônio Artificial: O Modelo de McCulloch-Pitts

Motivado a criar um modelo matemático que capturasse a essência do processamento neuronal, McCulloch, em colaboração com o jovem lógico Walter Pitts, publicou em 1943 o artigo seminal “A logical calculus of the ideas immanent in nervous activity” ([McCulloch and Pitts, 1943](#)). Nele, foi definido o primeiro neurônio artificial: um modelo simplificado que recebe múltiplas entradas binárias (0 ou 1). Esses valores são somados e, se o resultado ultrapassar um limiar pré-determinado, o neurônio ativa e produz uma saída de valor 1; caso contrário, a saída é 0.

McCulloch e Pitts demonstraram que, apesar de sua simplicidade, esse neurônio era capaz de implementar operações lógicas fundamentais, como **AND**, **OR** e **NOT**.

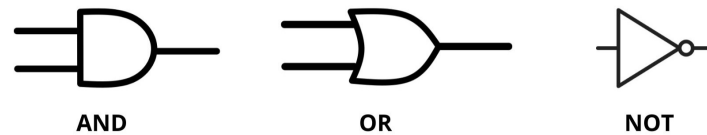


Figura 2.2: Representação gráfica das portas lógicas

Fonte - Autor.

- **AND:** Uma operação que retorna verdadeiro (1) somente se todas as suas entradas forem verdadeiras. Em um neurônio artificial com duas entradas, isso pode ser implementado definindo o limiar de ativação como 2, exigindo que ambas as entradas sejam 1 para que ele dispare.
- **OR:** Uma operação que retorna verdadeiro (1) se pelo menos uma de suas entradas for verdadeira. Isso pode ser implementado com um limiar igual a 1.
- **NOT:** Uma operação que inverte a entrada recebida, o que pode ser modelado através de conexões inibitórias.

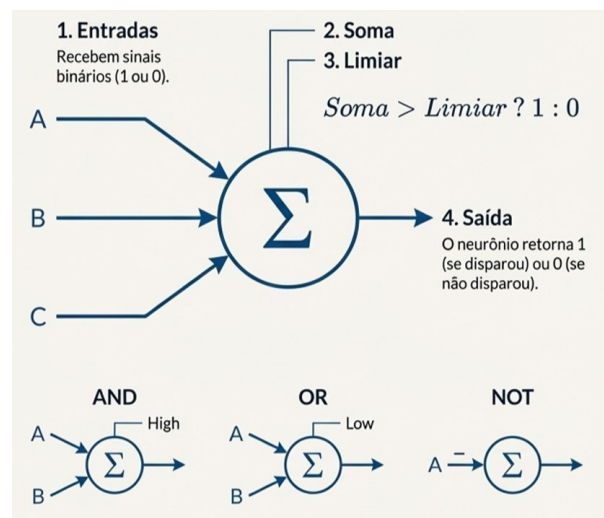


Figura 2.3: Representação lógica de um neurônio artificial

Fonte – Adaptado do ChatGpt.

A tese central era que a combinação de bilhões dessas operações lógicas poderia, em teoria, explicar a racionalidade humana. Contudo, o modelo era estático e desconsiderava um aspecto fundamental da inteligência: **o aprendizado**. Como sugeriu Alan Turing, para se criar uma máquina com inteligência de um adulto, seria necessário primeiro construir uma com a mente de uma criança e dotá-la da capacidade de aprender (Wooldridge, 2021).

2.1.3 A Evolução para o Aprendizado: O Perceptron de Rosenblatt

Uma década mais tarde, o psicólogo Frank Rosenblatt foi o primeiro a propor um neurônio artificial capaz de aprender com a experiência. Inspirado pela teoria do neuropsicólogo Donald Hebb, que postulava que o aprendizado ocorria pelo fortalecimento das conexões entre neurônios que se ativam simultaneamente (Hebb, 1949), Rosenblatt desenvolveu o **Perceptron** (Rosenblatt, 1958). Em 1958, ele apresentou sua implementação física, o Mark I Perceptron, uma máquina que, à época, foi anunciada pela imprensa como um “cérebro elétrico” capaz de aprender a reconhecer imagens de forma autônoma (Anderson and Rosenfeld, 1998).

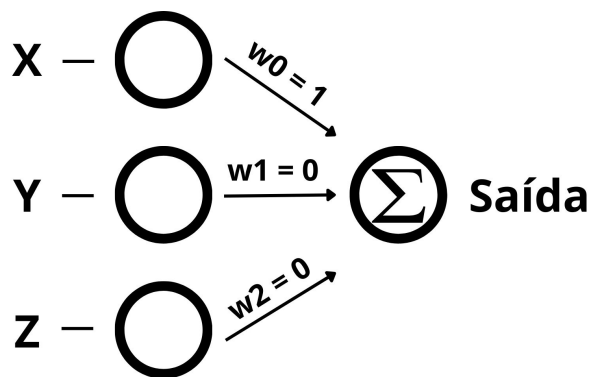


Figura 2.4: Representação do Perceptron

Fonte - Autor.

O neurônio de Rosenblatt era uma evolução do modelo de McCulloch-Pitts. Suas principais inovações foram:

1. **Entradas Ponderadas:** Cada entrada do neurônio foi associada a um peso (um parâmetro numérico), representando a “força” daquela conexão sináptica.
2. **Bias (Viés):** Foi adicionado um parâmetro de viés, uma constante que ajusta a facilidade com que o neurônio atinge seu limiar de ativação.

Nesse modelo, o neurônio calcula uma soma ponderada de suas entradas, adiciona o viés e, se o resultado final exceder o limiar, ele é ativado. O aprendizado consiste em um algoritmo que ajusta iterativamente os pesos e o viés com base nos erros de classificação (Bishop and Bishop, 2024).

2.1.4 Limitações e o Primeiro “Inverno da IA”

Apesar do entusiasmo inicial, o Perceptron possuía uma limitação crítica: seu algoritmo de aprendizado só era capaz de encontrar uma solução para problemas que fossem linearmente separáveis (Abu-Mostafa et al., 2012). Ou seja, ele só conseguia “desenhar” uma única linha reta para separar duas classes de dados. Para problemas mais complexos, o modelo falhava.

Rosenblatt estava ciente dessa limitação e propôs que a solução seria organizar múltiplos Perceptrons em camadas, formando uma **rede neural artificial**. No entanto, ele não conseguiu desenvolver um algoritmo para treinar os pesos de uma rede com múltiplas camadas, pois o método de correção de erros não se propagava eficientemente para as camadas internas. Essas dificuldades, somadas à crítica formal publicada por Marvin Minsky e Seymour Papert em seu livro de 1969, *Perceptrons*, levaram a uma drástica redução no financiamento e no interesse acadêmico pela área, inaugurando um período que ficou conhecido como o primeiro “inverno da Inteligência Artificial” (Minsky and Papert, 1969; Schmidhuber, 2015).

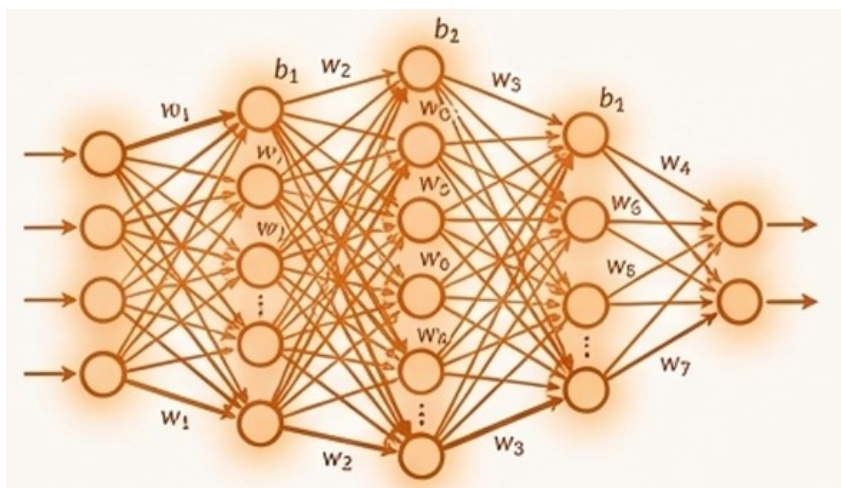


Figura 2.5: Rede neural artificial

Fonte – Adaptado do ChatGpt.

2.1.5 O Problema do Treinamento e a Solução Oculta na Astro-nomia

O desafio de treinar redes neurais profundas se resumia a um problema de otimização: como ajustar milhões de pesos interdependentes para minimizar o erro total da rede?

Sem que a comunidade de IA soubesse, as ferramentas matemáticas para resolver esse problema já existiam, mas em domínios completamente diferentes.

Uma analogia histórica ilustra perfeitamente o problema. Em 1801, o astrônomo Giuseppe Piazzi descobriu um novo corpo celeste, que nomeou de Ceres. Após 42 dias de observação, Ceres desapareceu atrás do Sol, e os astrônomos não conseguiam prever sua trajetória para reencontrá-lo (Teets and Whitehead, 1999). O desafio foi solucionado pelo jovem matemático Carl Friedrich Gauss, que tratou o problema como uma questão de minimização de erro.

Gauss definiu uma **função de erro**: a soma dos quadrados das distâncias entre as posições observadas e as posições previstas por uma dada elipse candidata. A tarefa se reduziu, então, a encontrar os parâmetros da elipse que produziam o menor erro possível (Tennenbaum and Director, 1997). Para resolver isso, Gauss utilizou um método precursor da **descida de gradiente**. O gradiente é um vetor que, em qualquer ponto de uma superfície de erro, aponta na direção de maior crescimento. Consequentemente, a direção oposta ao gradiente aponta para o caminho de descida mais íngreme, ou seja, o caminho mais rápido para um ponto de erro mínimo (Ananthaswamy, 2024).

A conexão com as redes neurais é direta: treinar uma rede é análogo a encontrar o planeta perdido. O objetivo é ajustar os milhões de pesos para encontrar o ponto mais baixo na vasta superfície da função de erro. A solução crucial, um algoritmo para calcular eficientemente o gradiente em sistemas complexos, foi publicada em 1960 por Henry J. Kelley, um engenheiro aeroespacial que o utilizou para otimizar trajetórias de voo (Kelley, 1960). Seu método era funcionalmente uma versão inicial da **retropropagação** (*backpropagation*). No entanto, por ter sido desenvolvido em um contexto diferente, levaria mais de uma década até que essa ideia fosse redescoberta e popularizada por pesquisadores como Paul Werbos, e mais tarde por Rumelhart, Hinton e Williams, resolvendo o problema de treinamento e encerrando o “inverno da IA” (Rumelhart et al., 1986; Schmidhuber, 2015).

2.1.6 O Renascimento com a Retropropagação

A retropropagação do erro, ou *backpropagation*, elegantemente popularizada por Rumelhart, Hinton e Williams em 1986, foi o algoritmo que finalmente permitiu treinar Redes Neurais com múltiplas camadas (*Multi-Layer Perceptrons* - MLPs) de forma eficiente. A sua lógica consiste em um processo de duas fases:

1. **Passagem Direta (*Forward Pass*):** Os dados de entrada (por exemplo, os pixels de uma imagem) são inseridos na primeira camada da rede. Cada neurônio processa os sinais recebidos e os passa para a camada seguinte. Esse processo continua até que a última camada produza um resultado final.
2. **Passagem Inversa (*Backward Pass*):** O resultado obtido é comparado com o resultado esperado, e o erro é calculado. É aqui que a “mágica” acontece: o algoritmo utiliza o cálculo diferencial (especificamente, a regra da cadeia) para propagar esse sinal de erro de volta, da última camada para a primeira. Em cada camada, ele calcula o quanto cada peso contribuiu para o erro total — o seu gradiente. Com essa informação, os pesos são ajustados minimamente na direção oposta ao seu gradiente, reduzindo o erro geral da rede.

Ao repetir esse ciclo milhares de vezes com diferentes exemplos de dados, a rede neural gradualmente “aprende” a mapear as entradas para as saídas corretas, superando a limitação de separabilidade linear do Perceptron e reacendendo o interesse da comunidade científica na área ([Rumelhart et al., 1986](#)).

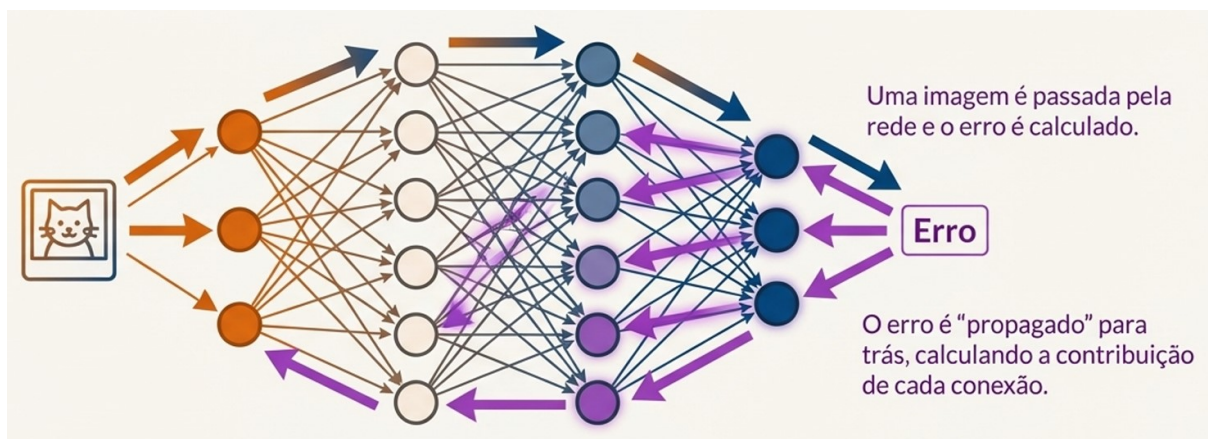


Figura 2.6: Ilustração do Forward Pass e Backward Pass

Fonte – Adaptado do ChatGpt.

2.1.7 Desafios das Redes Profundas e a Ascensão de Outros Métodos

Apesar do avanço, durante as décadas de 1990 e 2000, os pesquisadores enfrentaram novas barreiras ao tentar construir redes neurais cada vez mais profundas. O principal

obstáculo era o **problema do desaparecimento do gradiente** (*vanishing gradient*). Durante a retropropagação em redes com muitas camadas, o sinal de erro tendia a diminuir exponencialmente à medida que se aproximava das camadas iniciais. Na prática, isso significava que os pesos das primeiras camadas quase não eram atualizados, impedindo que a rede aprendesse características fundamentais dos dados (Hochreiter et al., 2001).

Devido a essa e outras dificuldades de treinamento, o campo da inteligência artificial viu a ascensão de outras técnicas de aprendizado de máquina que, na época, se mostravam mais robustas e fáceis de treinar. Dentre elas, destacaram-se as **Máquinas de Vetores de Suporte** (*Support Vector Machines - SVMs*), um método poderoso com uma base matemática sólida que se tornou a abordagem padrão para muitos problemas de classificação (Cortes and Vapnik, 1995).

2.1.8 A Revolução do Aprendizado Profundo

O cenário começou a mudar drasticamente por volta de 2010, quando uma “tempestade perfeita” de três fatores convergiu para dar início à era moderna do Aprendizado Profundo (*Deep Learning*):

- **Grandes Conjuntos de Dados (*Big Data*):** A proliferação da internet e a digitalização de informações levaram à criação de conjuntos de dados massivos. Um marco foi o surgimento do **ImageNet**, um banco de dados com milhões de imagens rotuladas, que forneceu o “combustível” necessário para treinar modelos de visão computacional em uma escala sem precedentes (Deng et al., 2009).
- **Poder Computacional (Hardware):** O treinamento de redes neurais profundas é uma tarefa computacionalmente intensiva. A grande virada veio com a adaptação de **Unidades de Processamento Gráfico (GPUs)** para fins de computação geral. Originalmente projetadas para renderizar gráficos de videogames, as GPUs se mostraram ideais para realizar os cálculos matriciais massivamente paralelos exigidos pelo *deep learning*, acelerando o tempo de treinamento de semanas para horas (Krizhevsky et al., 2012).
- **Inovações Algorítmicas:** Avanços cruciais nas próprias arquiteturas de rede foram desenvolvidos. Em 2012, a rede neural convolucional **AlexNet** venceu a competição de reconhecimento de imagem do ImageNet com uma margem de erro drástica-

mente menor que a dos seus concorrentes, provando a superioridade da abordagem de aprendizado profundo (Krizhevsky et al., 2012). A AlexNet popularizou o uso da função de ativação **ReLU** (*Rectified Linear Unit*), que ajudou a mitigar o problema do desaparecimento do gradiente, e introduziu técnicas de regularização como o *dropout*, que melhoraram a capacidade de generalização dos modelos.

Essa combinação de dados massivos, hardware poderoso e algoritmos mais sofisticados permitiu que as redes neurais não apenas se tornassem viáveis, mas dominassem o campo da inteligência artificial, alcançando desempenho sobre-humano em diversas tarefas e impulsionando a revolução tecnológica que vivemos hoje.

2.2 Contexto histórico da LIBRAS

A história da LIBRAS no Brasil caminha da primeira escola de surdos inaugurada em 1857 à consolidação de um marco legal que garante hoje o direito à comunicação em língua de sinais. Ao mesmo tempo, os dados censitários revelam que milhões de brasileiros ainda enfrentam barreiras de acesso à educação, à empregos e até mesmo saúde, em boa parte por falta de intérpretes. Frente a esse déficit, tecnologias assistivas baseadas em visão computacional e aprendizado profundo despontam como caminho viável para ampliar a inclusão, desde que dialoguem com a realidade sociolinguística brasileira e sejam integradas aos serviços públicos. A trajetória da Língua Brasileira de Sinais (LIBRAS) tem início formal em 1857, quando o professor francês Édouard Huet fundou, no Rio de Janeiro, o Imperial Instituto de Surdos-Mudos com o apoio de Dom Pedro II. A combinação dos sinais franceses com gestualidades já usadas pelos surdos locais lançou as bases de uma língua própria, que se difundiu pelas escolas especiais do país ao longo da segunda metade do século XIX (Huet, Édouard, 1857; Quadros, 1997). O final do século XIX e as primeiras décadas do XX foram marcados pela influência do Congresso de Milão (1880), que vetou oficialmente o uso de línguas de sinais em sala de aula. No Brasil, o Instituto Nacional de Educação de Surdos (INES) oscilou entre metodologias oralistas e gestuais, o que resultou em décadas de marginalização do uso público dos sinais e na perda de prestígio linguístico da LIBRAS (Strobel, 2008; Quadros, 1997). A partir de meados da década de 1970, movimentos da comunidade surda impulsionaram o reconhecimento cultural da LIBRAS, articulando militância política e pesquisa linguística. Estudos aca-

dêmicos demonstraram sua gramática completa e legitimaram a língua no meio científico, fortalecendo a identidade surda e contestando políticas assimilacionistas (Strobel, 2008; Quadros, 1997). Esse processo culminou na Lei 10.436/2002, que reconheceu a LIBRAS como meio legal de comunicação e expressão, e no Decreto 5.626/2005, que regulamentou o ensino da língua em cursos de licenciatura e a presença de intérpretes nos serviços públicos, estabelecendo um marco jurídico decisivo para a inclusão da comunidade surda brasileira (Brasil, 2002, 2005).

2.3 Cenário Atual das LIBRAS no Brasil

Segundo o Censo 2010 do IBGE, aproximadamente 10 milhões de brasileiros apresentam algum grau de deficiência auditiva; entre eles, cerca de 2,1 milhões possuem perda severa ou total (Instituto Brasileiro de Geografia e Estatística (IBGE), 2011). No entanto, o Censo Escolar 2023 registrou apenas 61,5 mil matrículas de estudantes surdos, evidenciando que a maioria ainda não encontra ambiente educacional plenamente acessível (INEP, 2024). Estudos sobre empregabilidade mostram que barreiras comunicacionais e a escassez de intérpretes limitam a inserção de surdos em ocupações de maior qualificação, contribuindo para rendas até 40% inferiores à média nacional (Pires and Almeida, 2022). A insuficiência de profissionais de interpretação, sobretudo fora dos grandes centros urbanos, agrava essa desigualdade. Nesse cenário, tecnologias assistivas de tradução automática — por exemplo, reconhecimento de LIBRAS por visão computacional — surgem como solução escalável para reduzir a dependência exclusiva de intérpretes humanos. Tais ferramentas podem oferecer suporte em aulas remotas, telemedicina e atendimentos públicos, ampliando a autonomia da pessoa surda (Frazão et al., 2015; Pires and Almeida, 2022). Projetos como o VLibras, que traduz texto escrito em português para animações 3-D em LIBRAS, e pesquisas recentes em reconhecimento automático de sinais brasileiros indicam altos índices de acurácia, mas apontam a necessidade de bases de dados mais robustas e de adaptação aos dialetos regionais da língua de sinais (Frazão et al., 2015; Rezende and Ludermir, 2021). Investir em soluções baseadas em aprendizado profundo é, portanto, estratégico para cumprir a legislação vigente e promover inclusão social de forma ampla.

2.4 Fundamentação técnica

2.4.1 Visão Computacional

A percepção visual humana é um processo computacional complexo. Inicia-se com a captura de luz pelos olhos, que atuam como sensores biológicos. A informação captada é transmitida ao cérebro, onde o córtex visual a processa em múltiplos estágios para extrair significado, como a identificação de formas, movimentos e profundidade. A visão, portanto, não é apenas “ver”, mas “compreender” o que é visto (Marr, 1982).

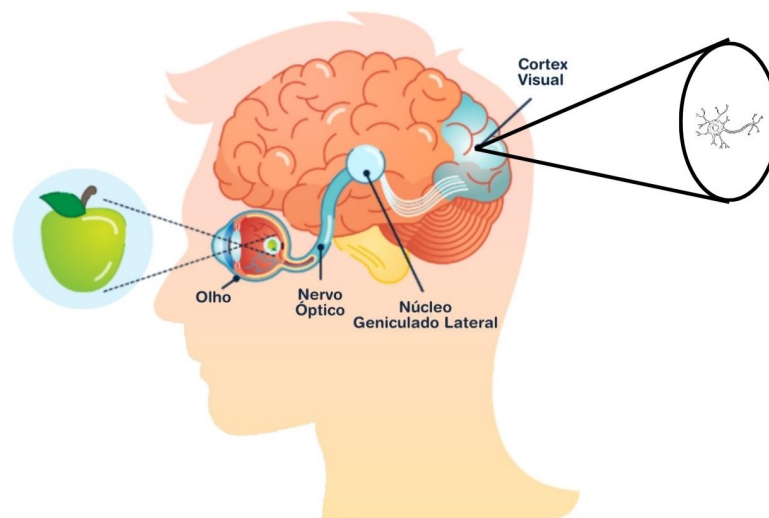


Figura 2.7: Córtex visual

Fonte – Autor.

De forma análoga, o “olho” de um sistema computacional é a câmera. Seus sensores digitais (como CCD ou CMOS) capturam a luz ambiente e a convertem em um sinal elétrico, que é então digitalizado e estruturado como uma grande matriz de valores numéricos. Essa matriz, conhecida como imagem digital, é composta por pixels, onde cada pixel armazena informações de intensidade e cor (Gonzalez and Woods, 2018). Até este ponto, o computador possui apenas os dados brutos; ele “viu” a imagem, mas ainda não a “compreendeu”.

É precisamente neste ponto que a **Visão Computacional** se inicia. Ela é a área que estuda métodos e algoritmos para que as máquinas possam interpretar essa matriz de pixels e “compreender” o mundo visual, indo além da simples captura de dados. O seu objetivo é extrair informação significativa de imagens e vídeos para realizar tarefas como detecção e reconhecimento de objetos, segmentação, estimativa de pose e rastrea-

mento. Diferente de simples processamento de imagem, a visão computacional combina modelos geométricos, estatísticos e de aprendizado de máquina para inferir estrutura e significado a partir de dados ruidosos, muitas vezes em tempo real. **Essa capacidade é fundamental para este projeto de interpretação de LIBRAS, pois permite transformar gestos complexos em representações computacionais.** Em um intérprete de língua de sinais, isso implica compreender padrões espaço-temporais distribuídos no corpo (mãos, braços, rosto e tronco) e convertê-los em representações simbólicas úteis para etapas linguísticas posteriores (Szeliski, 2022; Forsyth and Ponce, 2012).

Contudo, a eficácia desses métodos está sujeita às limitações físicas da câmera:

- **Resolução espacial:** limita a detecção de detalhes finos (por exemplo, configurações de dedos em LIBRAS);
- **Taxa de quadros (FPS):** afeta a captura de movimentos rápidos característicos de sinais;
- **Ângulo de visão:** restringe o enquadramento necessário para capturar o corpo inteiro;
- **Sensibilidade luminosa:** dificulta a operação em ambientes com iluminação subótima;
- **Distorções ópticas:** podem alterar trajetórias de movimento críticas para o significado.

Esses fatores de hardware interagem com o conteúdo visual e o ambiente (iluminação, fundo, sombras), impondo requisitos de pré-processamento, calibração e desenho de modelos robustos (Szeliski, 2022; Gonzalez and Woods, 2018).

Ao trabalhar com projetos de visão computacional, Python destaca-se como linguagem de programação ideal devido ao seu ecossistema maduro. A aquisição e o pré-processamento de quadros e sequências são comumente realizados via OpenCV-Python, que oferece operações eficientes de captura, filtragem, detecção e rastreamento; a manipulação numérica é sustentada por *arrays* do NumPy; transformações e rotinas de análise complementares são encontradas no *scikit-image*. Para estimar *keypoints* de mãos, face e corpo, frameworks de percepção como o MediaPipe fornecem pipelines prontos e

otimizados; e, para modelagem de alto nível, bibliotecas de aprendizado profundo como TensorFlow e PyTorch permitem treinar e servir modelos modernos com aceleração por GPU/TPU (Bradski, 2000; Harris et al., 2020; Van der Walt et al., 2014; Lugaresi et al., 2019; Abadi et al., 2016; Paszke et al., 2019).

A importância da visão computacional para este projeto decorre de sua capacidade de **superar limitações técnicas por meio de abordagens algorítmicas**: capturar tanto componentes manuais (configuração e trajetória das mãos) quanto não manuais (expressões faciais, inclinação de cabeça, postura do tronco), todos essenciais para gerar sentido das frases em LIBRAS. Evidências na literatura mostram que pipelines baseados em detecção de mãos/face, estimativa de pose e extração de características visuais, integrados a modelos temporais, **compensam deficiências de hardware** e melhoram substancialmente a precisão em tarefas de classificação e tradução de sinais, especialmente em contextos de uso contínuo (*continuous sign language*) (Rastgoo et al., 2020; Camgöz et al., 2018; Cao et al., 2017). Além disso, módulos de visão podem ser otimizados para dispositivos embarcados de baixa latência via quantização e arquiteturas eficientes — **onde a representação por *keypoints* minimiza o impacto das limitações da câmera ao trabalhar com dados geométricos normalizados em vez de pixels brutos**, reduzindo a sensibilidade a variações de iluminação e fundos, facilitando a generalização entre ambientes (Jacob et al., 2018; Howard et al., 2017; Sandler et al., 2018; Szeliski, 2022).

2.4.2 Aprendizado Profundo para Processamento de Sequências

Enquanto a visão computacional fornece as ferramentas para “ver” e extrair dados de imagens, o aprendizado de máquina (*Machine Learning* - ML) oferece os meios para interpretar esses dados e tomar decisões. O ML é essencial para traduzir os padrões visuais complexos dos sinais em LIBRAS para os símbolos linguísticos que eles representam, uma tarefa inviável com programação tradicional baseada em regras explícitas (Bishop and Bishop, 2024).

Diferente de abordagens de ML tradicional, como *Support Vector Machines* (SVMs) ou *Random Forests*, que frequentemente dependem de uma etapa manual de extração de características (*feature engineering*), o Aprendizado Profundo (*Deep Learning*) utiliza redes neurais com múltiplas camadas para aprender representações hierárquicas dos dados

de forma automática. Para o reconhecimento de sinais, isso significa que a rede pode aprender a identificar desde características simples, como a orientação dos dedos, até combinações complexas que formam um sinal completo, diretamente dos dados brutos ou de representações como *keypoints* (Goodfellow et al., 2016).

2.4.2.1 A Arquitetura das Redes Neurais Recorrentes (RNNs)

Para tarefas que envolvem dados sequenciais, como o reconhecimento de LIBRAS, a arquitetura de rede neural padrão não é suficiente, pois trata cada entrada de forma independente. As Redes Neurais Recorrentes (RNNs) foram projetadas para superar essa limitação. A sua característica fundamental é a presença de um “laço” (*loop*), que permite que a informação persista ao longo do tempo.

A anatomia básica de uma RNN consiste em uma célula que processa a entrada de um passo de tempo (x_t) e o estado oculto do passo de tempo anterior (h_{t-1}) para produzir um novo estado oculto (h_t). Este novo estado, então, serve como entrada para o próximo passo de tempo, junto com a nova entrada x_{t+1} . Matematicamente, essa recorrência pode ser expressa como:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

Onde W_{hh} e W_{xh} são as matrizes de pesos para as conexões recorrentes e de entrada, respectivamente, b_h é o vetor de viés e f é uma função de ativação não-linear. Crucialmente, os mesmos pesos (W) e viés (b) são compartilhados em todos os passos de tempo, permitindo que a rede aplique o mesmo aprendizado a diferentes posições na sequência.

2.4.2.2 O Desafio dos Gradientes em Sequências Longas

Apesar de sua elegância conceitual, o treinamento de RNNs com o algoritmo de retropropagação no tempo (*Backpropagation Through Time* - BPTT) revelou dois problemas graves que dificultam o aprendizado de dependências de longo prazo:

- **Explosão do Gradiente (*Exploding Gradient*):** O problema de Explosão do Gradiente ocorre se os valores da matriz de pesos recorrente forem grandes (norma maior que 1). Nesse caso, o gradiente cresce exponencialmente, resultando em atualizações de peso massivas e instáveis que podem levar os valores dos pesos a se tornarem ‘NaN’ (*Not a Number*), colapsando o processo de treinamento. Embora

este problema seja mais fácil de detectar e mitigar (usando técnicas como *gradient clipping*), ele evidencia a instabilidade fundamental do treinamento de RNNs simples (Goodfellow et al., 2016).

- **Desaparecimento do Gradiente (*Vanishing Gradient*):** Durante a retropropagação, o gradiente do erro é propagado para trás através do laço da rede. Pela regra da cadeia, isso envolve multiplicações sucessivas pela matriz de pesos recorrente (W_{hh}). Se os valores dessa matriz forem pequenos (norma menor que 1), o gradiente diminuirá exponencialmente à medida que se afasta no tempo. Para sequências longas, o gradiente pode se tornar tão infinitesimalmente pequeno que os pesos das camadas iniciais da sequência deixam de ser atualizados, impedindo que a rede aprenda a correlação entre eventos distantes (Hochreiter et al., 2001).

2.4.2.3 Componentes Essenciais: Funções de Ativação Sigmoides e Tangente Hiperbólica

Para entender como as arquiteturas LSTM e GRU resolvem esses problemas, é fundamental primeiro compreender duas funções de ativação que são seus blocos de construção:

- **Função Sigmoides (σ):** Esta função “esmaga” qualquer valor de entrada para um intervalo entre 0 e 1. Matematicamente, $\sigma(x) = \frac{1}{1+e^{-x}}$. Devido a essa propriedade, ela é utilizada nos “portões” (*gates*) das LSTMs e GRUs para atuar como um filtro ou controlador. Um valor de saída próximo de 0 significa “não deixe passar nenhuma informação”, enquanto um valor próximo de 1 significa “deixe toda a informação passar”.
- **Função Tangente Hiperbólica (\tanh):** Semelhante à sigmoide, a *tanh* também esmaga os valores de entrada, mas para um intervalo entre -1 e 1. Matematicamente, $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. Ela é usada para regular o “conteúdo” ou a “substância” da informação que flui pela rede, mantendo os valores normalizados e centrados em zero.

2.4.2.4 A Arquitetura LSTM e a Solução para o Fluxo do Gradiente

A arquitetura *Long Short-Term Memory* (LSTM) foi projetada especificamente para combater os problemas de gradiente, introduzindo um **estado da célula** (C_t) que atua

como uma via expressa para o fluxo de informação e gradientes (Hochreiter and Schmidhuber, 1997). A LSTM controla essa via através de seus três portões:

1. **Portão de Esquecimento** (f_t): Controlado por uma função sigmoide, este portão determina qual fração da informação do estado da célula anterior (C_{t-1}) será mantida.
2. **Portão de Entrada** (i_t): Decide quais novas informações, geradas a partir da entrada atual (x_t) e do estado oculto anterior (h_{t-1}), serão adicionadas ao estado da célula.
3. **Portão de Saída** (o_t): Filtra o estado da célula para produzir o novo estado oculto (h_t), que é a saída da célula para o próximo passo de tempo.

A chave para a resolução do problema do gradiente está na atualização do estado da célula, expressa pela equação:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

Onde \odot representa a multiplicação elemento a elemento e \tilde{C}_t são os novos valores candidatos gerados por uma camada *tanh*. A natureza **aditiva** desta operação é fundamental. Durante a retropropagação, o gradiente pode passar pela primeira parte da equação ($f_t \odot C_{t-1}$) sem ser repetidamente multiplicado por uma matriz de pesos. Se o portão de esquecimento (f_t) aprender a produzir um vetor de “uns”, a informação e o gradiente do estado anterior (C_{t-1}) passam diretamente para o estado atual (C_t), criando um caminho ininterrupto. Isso permite que o gradiente flua por longas distâncias no tempo sem desaparecer ou explodir, capacitando a rede a aprender dependências de longo prazo.

2.4.2.5 A Arquitetura GRU: Uma Alternativa Eficiente

A *Gated Recurrent Unit* (GRU) é uma simplificação da LSTM que também se mostrou altamente eficaz (Cho et al., 2014). Ela combina o estado da célula e o estado oculto e utiliza apenas dois portões:

1. **Portão de Atualização** (z_t): Este único portão controla tanto o esquecimento quanto a adição de novas informações. Ele decide a proporção com que a unidade

atualizará seu estado com a nova informação candidata versus manter a informação do estado anterior.

2. **Portão de Reinicialização (r_t):** Determina o quão permeável a célula está à informação do estado anterior ao calcular o novo conteúdo. Se este portão estiver próximo de 0, a célula efetivamente descarta a informação do passado para gerar sua nova proposta de informação.

A equação de atualização da GRU é:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

Assim como na LSTM, a estrutura aditiva da GRU — balanceando diretamente o estado anterior (h_{t-1}) com o novo candidato (\tilde{h}_t) — cria um caminho direto para o fluxo do gradiente. O portão de atualização (z_t) regula esse fluxo, mas não o multiplica repetidamente por uma matriz de pesos fixa, resolvendo eficazmente os problemas de gradiente das RNNs simples, porém com menos parâmetros e complexidade computacional.

Capítulo 3

Revisão de Trabalhos Correlatos

Este capítulo apresenta uma revisão do estado da arte em reconhecimento automático da Língua Brasileira de Sinais (LIBRAS), com foco em abordagens baseadas em visão computacional e aprendizado profundo. A análise dos trabalhos a seguir busca identificar as metodologias consolidadas, os resultados alcançados e, principalmente, as limitações e lacunas que justificam a proposta deste Trabalho de Conclusão de Curso.

3.1 Abordagens para Sinais Estáticos e Isolados

Uma parcela significativa da pesquisa inicial em reconhecimento de LIBRAS concentrou-se em problemas de escopo reduzido, como a classificação de sinais estáticos. Um exemplo representativo dessa abordagem é o trabalho de Costa & Oliveira (2020), que desenvolveu um classificador para o alfabeto datilológico da LIBRAS. Por se tratar de poses estáticas das mãos, a metodologia pôde se concentrar exclusivamente na análise espacial, utilizando uma Rede Neural Convolutiva (CNN) leve, a MobileNetV2, para classificar as imagens. A abordagem obteve altíssima acurácia, superando 98%, e demonstrou viabilidade para uso em tempo real (Costa and Oliveira, 2020).

Apesar do sucesso, a principal limitação de trabalhos como este é sua aplicabilidade restrita. A comunicação fluente em LIBRAS é majoritariamente composta por sinais dinâmicos, que envolvem movimentos complexos, trajetórias e expressões faciais. Portanto, embora a classificação de sinais estáticos seja um problema resolvido com alta eficácia, a metodologia não é transferível para o desafio mais amplo do reconhecimento de sinais dinâmicos.

3.2 Arquiteturas Híbridas para Sinais Dinâmicos

Para abordar a natureza temporal dos sinais dinâmicos, a literatura convergiu para o uso de arquiteturas híbridas, que combinam modelos espaciais e temporais. Um trabalho exemplar desta linha é o de Souza & Silva (2019), que propôs um sistema para classificar 150 sinais isolados da LIBRAS. A metodologia empregada consistiu em duas etapas: primeiramente, uma CNN pré-treinada (InceptionV3) foi utilizada para extrair um vetor de características de cada quadro do vídeo; em seguida, a sequência de vetores foi fornecida como entrada para uma Rede Neural Recorrente do tipo LSTM, responsável por modelar a dinâmica temporal e realizar a classificação final do sinal (Souza and Silva, 2019).

Essa abordagem alcançou resultados robustos, com acurácia superior a 94%, consolidando-se como um método padrão na área. Contudo, o sucesso de tais sistemas geralmente está atrelado a condições controladas. O trabalho de Souza & Silva (2019), assim como muitos outros, utilizou um dataset gravado em laboratório, com fundo uniforme e iluminação ideal. Além disso, o foco permaneceu na classificação de **sinais isolados**, onde o início e o fim de cada gesto são claramente definidos, um cenário que difere significativamente da sinalização contínua e fluida do mundo real.

3.3 Sistemas em Tempo Real e a Lacuna no Contexto de Uso

Com a evolução das técnicas de extração de características, surgiram abordagens mais eficientes, visando a aplicação em tempo real. Nascimento et al. (2021) desenvolveram um tradutor para um vocabulário básico de 20 sinais de uso geral. Em vez de processar os pixels brutos com uma CNN pesada, os autores utilizaram a biblioteca MediaPipe para extrair *keypoints* das mãos e do corpo. Essa representação esquelética, mais leve e invariante a ruídos de fundo, serviu como entrada para uma rede GRU, que se mostrou eficiente para aprender os padrões temporais mesmo com um dataset customizado e de tamanho reduzido (Nascimento et al., 2021).

O sistema alcançou performance em tempo real com boa acurácia (91%), validando a metodologia de *keypoints* + GRU/LSTM como uma arquitetura moderna e viável. No entanto, a principal limitação apontada é a natureza genérica do vocabulário. A aplicação

se concentra em saudações e expressões cotidianas, sem atender a um domínio específico.

É precisamente nesta lacuna que o presente trabalho se insere. Enquanto as pesquisas demonstram a viabilidade técnica do reconhecimento de LIBRAS, uma carência notável persiste no desenvolvimento de sistemas aplicados a contextos de nicho, onde a comunicação clara é crítica. O foco deste TCC é utilizar uma metodologia moderna e eficiente, semelhante à de Nascimento et al. (2021), para resolver um problema prático e de alto impacto: a barreira de comunicação entre pacientes surdos e profissionais de saúde. Ao desenvolver um intérprete para um vocabulário específico de sintomas e condições médicas, este projeto não apenas valida uma abordagem técnica, mas também propõe uma solução direcionada a uma necessidade social urgente e pouco explorada na literatura acadêmica.

Capítulo 4

Metodologia

Este capítulo detalha a metodologia empregada para o desenvolvimento do interpretador de sinais da Língua Brasileira de Sinais (LIBRAS) focado no contexto médico. A abordagem segue um fluxo de trabalho estruturado, partindo da criação de um dataset customizado até a implementação de um sistema de inferência em tempo real. Cada etapa, desde o tratamento dos dados brutos até a avaliação do modelo final, foi projetada para construir um sistema robusto e funcional, capaz de traduzir sinais complexos em frases contextuais.

4.1 Visão Geral do Sistema

O sistema proposto implementa uma pipeline completa de visão computacional e aprendizado profundo. O processo inicia com a captura de vídeo, que é subsequentemente transformado em uma sequência de dados numéricos representando a pose do usuário. Essa sequência é então analisada por um modelo de rede neural recorrente treinado para classificar o sinal. Em paralelo, uma lógica de detecção de apontamento identifica para onde o usuário aponta. Finalmente, as saídas de ambos os módulos são combinadas para gerar uma tradução textual contextualizada, como “dor de cabeça”. O fluxo geral do processo pode ser visualizado na Figura 4.1.

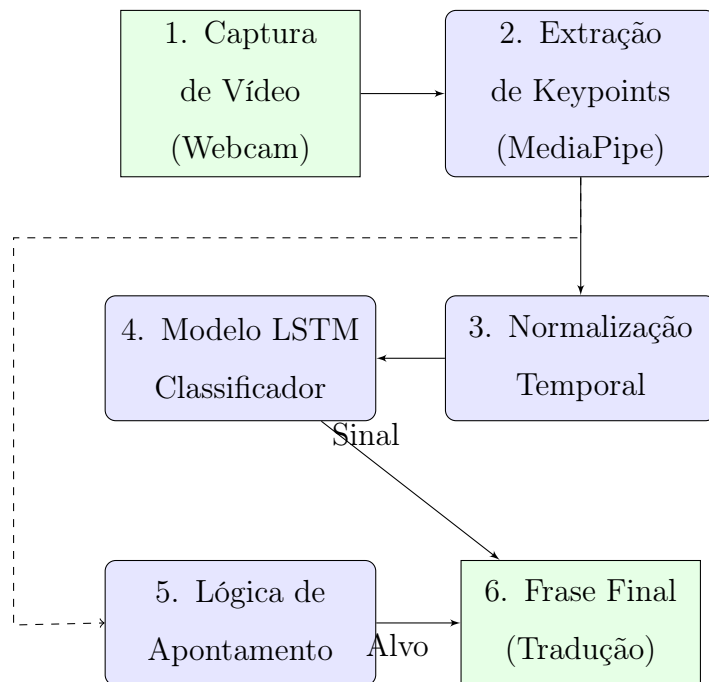


Figura 4.1: Diagrama de fluxo do sistema de interpretação de LIBRAS.

Fonte – Autor.

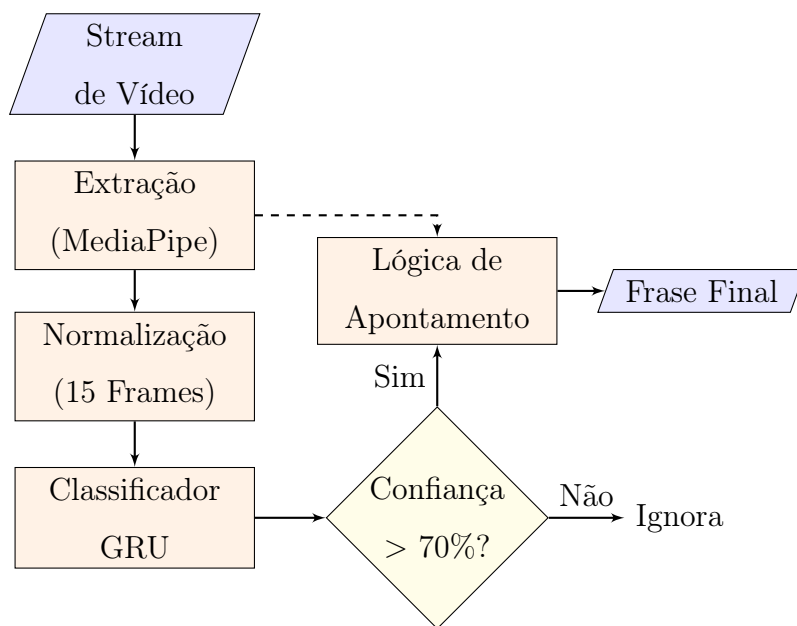


Figura 4.2: Pipeline detalhada do processamento de dados do sistema.

Fonte – Autor.

4.2 Coleta e Preparação dos Dados

A fundação de qualquer sistema de aprendizado de máquina é um conjunto de dados robusto e bem estruturado. Para este projeto, foi criado um dataset customizado, focado em um vocabulário específico para o domínio da saúde.

4.2.1 Definição do Vocabulário e Gravação

O vocabulário foi definido para cobrir queixas comuns em um ambiente de triagem hospitalar, consistindo nos oito sinais: **dor de cabeça**, **dor no ouvido**, **dor na garganta**, **dor no peito**, **dor de barriga**, **coriza**, **febre** e **estou doente**. As amostras de vídeo para cada sinal foram capturadas utilizando uma webcam Logitech c920s, com o auxílio do script `src/capture/captura_samples.py`. Este script automatiza a gravação ao detectar a presença de mãos em quadro, garantindo a captura de sequências de sinais relevantes.

4.2.2 Normalização Temporal e Justificativa Técnica

Modelos de redes neurais recorrentes exigem que as sequências de entrada tenham um comprimento fixo. Para padronizar a duração variável dos sinais, o script `src/processing/normalize_frames.py` foi desenvolvido para normalizar todas as amostras de vídeo para um comprimento fixo de **15 frames**, conforme definido em `src/config.py`.

A escolha de 15 frames representa um balanço técnico fundamental. Conforme discutido na fundamentação teórica, a captura de movimentos rápidos é limitada pela taxa de quadros (FPS) da câmera. Uma sequência muito curta (e.g., 5 frames) poderia não capturar a dinâmica completa de um sinal mais longo, enquanto uma sequência muito longa (e.g., 60 frames) aumentaria drasticamente o custo computacional do treinamento e da inferência, tornando a aplicação em tempo real inviável em hardware convencional. O valor de 15 frames foi determinado empiricamente como um ponto ótimo que preserva a informação temporal essencial dos sinais do vocabulário definido, mantendo a carga computacional gerenciável. Para amostras com menos de 15 frames, foi aplicada uma interpolação linear para gerar frames intermediários. Para amostras com mais de 15 frames, foi realizado um processo de subamostragem.

4.3 Extração de Características com MediaPipe

Optou-se por uma representação baseada em pontos-chave (*keypoints*) em vez de utilizar os pixels brutos dos vídeos. Esta decisão foi motivada pela necessidade de criar um sistema robusto a variações de ambiente (iluminação, fundo) e de reduzir a dimensionalidade dos dados de entrada.

4.3.1 Justificativa da Biblioteca MediaPipe

A biblioteca **MediaPipe** foi escolhida por ser uma solução de código aberto, altamente otimizada e que oferece modelos pré-treinados de alta performance para a detecção de marcos corporais. Sua solução **Holistic** permite a extração simultânea e em tempo real de *keypoints* de pose, face e mãos, o que é ideal para a captura da riqueza de informações presentes em LIBRAS. O script `src/processing/create_keypoints.py` implementa essa extração.



Figura 4.3: Extração dos keypoints com MediaPipe

Fonte - Adaptado do ChatGpt.

Dessa forma, ao em vez de processar pixels de vídeo, o que é lento e sensível à iluminação, o MediaPipe extrai um “esqueleto digital” do indivíduo.

4.3.2 Vetor de Características e Formato HDF5

Para cada frame, a função `extract_keypoints` extraiu um vetor numérico unificado e concatenado de **1.662 características** (pose, face e ambas as mãos). As sequências resultantes (15 vetores por amostra) foram salvas em arquivos no formato **HDF5**

(**Hierarchical Data Format 5**). Este formato foi escolhido por sua alta eficiência no armazenamento e manipulação de grandes volumes de dados numéricos. O HDF5 permite acesso rápido a subconjuntos de dados sem a necessidade de carregar o arquivo inteiro na memória, o que é extremamente vantajoso durante o processo de treinamento do modelo.

4.4 Arquitetura e Treinamento do Modelo LSTM

O núcleo do sistema de reconhecimento de sinais é um modelo de aprendizado profundo, definido e treinado no script `src/train/train_model.py`.

4.4.1 Justificativa da API Keras e da Arquitetura

O modelo foi implementado utilizando a API **Keras** do **TensorFlow**. Keras foi selecionada por ser uma API de alto nível que permite a prototipagem rápida e a construção de arquiteturas de rede complexas de forma intuitiva e modular. A arquitetura da Rede Neural Recorrente foi projetada para capturar eficientemente as dependências temporais dos dados:

- **Duas Camadas LSTM:** A utilização de duas camadas empilhadas permite que a rede aprenda hierarquias de características temporais. A primeira camada (com 64 unidades e `return_sequences=True`) processa a sequência de entrada e passa uma nova sequência de estados ocultos para a camada seguinte. A segunda camada LSTM (com 64 unidades) recebe essa sequência e a resume em um único vetor que encapsula a informação de todo o sinal.
- **Camadas Dropout:** Duas camadas de *Dropout* com taxa de 20% foram inseridas após cada camada LSTM. Esta é uma técnica de regularização essencial para combater o *overfitting*. Durante o treinamento, o Dropout zera aleatoriamente uma fração das saídas dos neurônios, forçando a rede a aprender representações mais robustas e a não depender excessivamente de neurônios específicos.
- **Camada Densa (ReLU):** Uma camada Densa com 64 unidades e ativação ReLU é utilizada para aprender combinações não-lineares das características extraídas pelas camadas LSTM, aumentando a capacidade de discriminação do modelo.

- **Camada de Saída (Softmax):** A camada final possui 8 neurônios (um para cada sinal) e a função de ativação **Softmax**. Softmax transforma os valores de saída brutos da rede (logits) em uma distribuição de probabilidade, onde a soma de todas as saídas é 1. Isso permite interpretar a saída do modelo como o nível de confiança para cada uma das classes possíveis.

4.4.2 Justificativa dos Parâmetros de Treinamento

A compilação do modelo envolveu a escolha de um otimizador e de uma função de perda, ambos cruciais para a eficácia do treinamento:

- **Otimizador Adam:** O **Adam** (Adaptive Moment Estimation) foi escolhido por ser um algoritmo de otimização de gradiente descendente que adapta a taxa de aprendizado para cada parâmetro do modelo de forma individual. Ele combina as vantagens de outros otimizadores (como AdaGrad e RMSProp), sendo computacionalmente eficiente e robusto para uma vasta gama de problemas de aprendizado profundo.
- **Função de Perda Sparse Categorical Crossentropy:** Esta função de perda foi selecionada por ser especificamente projetada para problemas de classificação multiclasse onde os rótulos são fornecidos como números inteiros (0, 1, 2, ..., 7), em vez de vetores *one-hot*. Isso torna o processo mais eficiente em termos de memória e computação, sem perda de desempenho.

O treinamento foi executado por **100 épocas**, utilizando um *callback* `ModelCheckpoint` para salvar apenas a melhor versão do modelo, evitando assim o *overfitting* que pode ocorrer nas épocas finais.

4.5 Sistema de Inferência e Avaliação de Desempenho

A aplicação final do modelo foi implementada no script [src/inference/combined_symptom_detector2.py](#), que combina a classificação de sinais com a lógica de apontamento para uso em tempo real.

4.5.1 Lógica de Inferência

O sistema opera em tempo real, mantendo um buffer dos *keypoints* extraídos dos últimos frames da webcam. Quando um sinal é concluído, a sequência é normalizada e enviada ao modelo LSTM. Se a previsão atinge um limiar de confiança de 70%, o sinal é reconhecido. Em paralelo, o sistema calcula a distância euclidiana entre a ponta do dedo indicador e alvos corporais pré-definidos (cabeça, peito, etc.). Se um sinal como “dor” é reconhecido junto a um apontamento para a “cabeça”, o sistema gera a frase contextual “dor de cabeça”.

4.5.2 Definição e Justificativa das Métricas de Avaliação

A avaliação de modelos de redes neurais, como LSTMs e GRUs, requer a seleção de métricas alinhadas à natureza do problema. Em tarefas de *regressão*, onde o objetivo é prever valores contínuos (como a temperatura de um paciente ou o preço de uma ação), utilizam-se métricas baseadas na magnitude do erro, como o Erro Médio Absoluto (MAE), o Erro Quadrático Médio (MSE), a Raiz do Erro Quadrático Médio (RMSE) ou o coeficiente de determinação (R^2).

No entanto, o problema abordado neste trabalho é de natureza **classificatória** (Classificação Multiclasse), onde o objetivo é categorizar uma entrada (sequência de *keypoints*) em um rótulo discreto (um dos 8 sintomas). Neste contexto, métricas de regressão não são aplicáveis. Portanto, a avaliação de desempenho baseou-se nas métricas fundamentais de classificação: Acurácia, Precisão, Sensibilidade (*Recall*) e *F1-Score*. A seguir, detalha-se a definição e a importância de cada uma para o contexto médico deste estudo.

4.5.2.1 Acurácia (Accuracy)

A acurácia é a métrica mais intuitiva, representando a proporção de previsões corretas em relação ao total de amostras avaliadas. Embora forneça uma visão geral do desempenho, ela pode ser enganosa em *datasets* desbalanceados. No entanto, como o protocolo de coleta deste trabalho buscou balancear as classes (número similar de repetições para cada sinal), a acurácia mantém-se como um indicador válido de performance global.

4.5.2.2 Precisão (Precision)

A precisão responde à pergunta: “De todas as vezes que o modelo previu a classe X, quantas vezes ele estava certo?”. Matematicamente, é a razão entre os Verdadeiros Positivos (VP) e a soma de Verdadeiros Positivos e Falsos Positivos (FP):

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (4.1)$$

No contexto de triagem hospitalar, uma alta precisão é desejável para evitar “alarmes falsos” que poderiam sobrecarregar o sistema de atendimento ou gerar ansiedade desnecessária no paciente.

4.5.2.3 Sensibilidade (Recall)

Também conhecida como Revocação, a sensibilidade responde à pergunta: “De todos os casos reais da classe X, quantos o modelo foi capaz de detectar?”. É a razão entre os Verdadeiros Positivos (VP) e a soma de Verdadeiros Positivos e Falsos Negativos (FN):

$$\text{Recall} = \frac{VP}{VP + FN} \quad (4.2)$$

Em aplicações médicas, o *Recall* é frequentemente a métrica mais crítica. Um baixo *Recall* implicaria em não detectar um sintoma que o paciente realmente está sentindo (falso negativo), o que poderia levar a uma triagem incorreta e riscos à saúde.

4.5.2.4 F1-Score

O *F1-Score* é a média harmônica entre a Precisão e o *Recall*. Diferente da média aritmética simples, a média harmônica penaliza valores extremos.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (4.3)$$

Esta métrica é essencial para demonstrar a robustez do modelo. Um *F1-Score* alto indica que o sistema não está apenas “chutando” a classe majoritária (o que aumentaria o *Recall* mas derrubaria a Precisão) nem sendo excessivamente conservador (o que aumentaria a Precisão mas derrubaria o *Recall*). Ele representa o equilíbrio ideal do classificador.

4.6 Ferramentas e Ambiente de Desenvolvimento

O projeto foi desenvolvido inteiramente na linguagem **Python 3**. As principais bibliotecas utilizadas foram:

- **TensorFlow** e **Keras**: Para a construção e treinamento do modelo de aprendizado profundo.
- **OpenCV**: Para captura e manipulação de vídeo em tempo real.
- **MediaPipe**: Para a extração de *keypoints* de corpo, face e mãos.
- **NumPy** e **Pandas**: Para manipulação de dados numéricos e estruturação dos *keypoints*.

A estrutura do projeto foi modularizada, com a separação de responsabilidades em diferentes pacotes e o uso de um arquivo `config.py` para centralizar os parâmetros globais do sistema.

4.7 Definição do Vocabulário e Restrições Iniciais

O vocabulário final do sistema foi consolidado em oito sinais de sintomas médicos. Inicialmente, tentou-se uma abordagem com um vocabulário mais extenso, utilizando um modelo treinado exclusivamente com dados do próprio autor (abordagem mono-usuário). No entanto, testes preliminares de campo mostraram que, ao aumentar a quantidade de classes sem a devida variabilidade de dados de treinamento (diferentes executores), o modelo sofria de *overfitting* severo, decorando o estilo de sinalização do autor e falhando ao generalizar para terceiros.

Para garantir a robustez do sistema na validação final com os 30 voluntários, optou-se por restringir o escopo para os oito sinais mais críticos de triagem. Essa decisão permitiu focar a coleta de dados na variabilidade dos sujeitos (30 pessoas) em vez da quantidade de classes, priorizando a qualidade da generalização em um cenário real sobre a amplitude do dicionário.

4.7.1 Normalização Temporal e Pré-processamento

Para adequar as sequências de vídeo de duração variável à entrada fixa da rede neural (15 frames), aplicaram-se técnicas de normalização temporal:

- **Interpolação Linear:** Utilizada quando a amostra original possuía menos de 15 frames. O algoritmo calcula coordenadas intermediárias entre dois frames adjacentes, criando novos quadros sintéticos que preenchem a lacuna temporal sem alterar a velocidade percebida do sinal.
- **Subamostragem Uniforme:** Aplicada quando a amostra excedia 15 frames. O algoritmo seleciona quadros em intervalos regulares (ex: a cada 2 ou 3 frames), preservando a estrutura macroscópica do movimento e descartando redundâncias.

4.7.2 Caracterização do Dataset

O conjunto de dados final (*dataset*) foi construído a partir das gravações dos 30 voluntários. Cada participante executou cada um dos 8 sinais uma única vez, de modo a não atrapalhar as atividades normais de cada um. A distribuição das classes foi mantida balanceada para evitar viés no treinamento, garantindo que o modelo aprendesse a identificar todos os sintomas com a mesma prioridade.

4.8 Caracterização da Pesquisa

O presente trabalho caracteriza-se como uma pesquisa de natureza **aplicada** e de caráter **exploratório**. É aplicada pois objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos — neste caso, a barreira de comunicação na triagem hospitalar. É exploratória pois visa proporcionar maior familiaridade com o problema, envolvendo levantamento bibliográfico e testes práticos em um domínio onde a aplicação de Deep Learning para LIBRAS (especificamente em contexto médico) ainda carece de soluções definitivas.

Dada a complexidade do sistema, que envolve desde a fundamentação teórica até a validação com seres humanos, adotou-se uma abordagem metodológica mista. O desenvolvimento foi estruturado em quatro etapas distintas, cada uma exigindo uma estratégia de

investigação específica para garantir o rigor científico e a reprodutibilidade dos resultados. A Tabela 4.1 apresenta a classificação metodológica adotada para cada fase do projeto.

Tabela 4.1: Classificação metodológica das etapas do desenvolvimento.

Etapa do Projeto	Tipo de Pesquisa	Descrição da Atividade
Revisão do Estado da Arte	Bibliográfica e Exploratória	Levantamento de trabalhos correlatos em bases como IEEE Xplore e Google Scholar.
Construção do Dataset	Experimental (Campo)	Coleta de dados com 30 voluntários em ambiente controlado.
Desenvolvimento do Modelo	Experimental (Laboratório)	Treinamento e comparação de arquiteturas (LSTM vs GRU) em ambiente Python.
Validação do Sistema	Quantitativa / Estatística	Aplicação do método <i>Leave-One-Subject-Out</i> (LOSO) para aferição de métricas.

Tabela 4.2: Fonte - Autor.

Capítulo 5

Resultados e Discussão

Este capítulo descreve a trajetória experimental do projeto, dividida em três fases estratégicas: (1) Seleção da arquitetura de rede neural e treinamento inicial com dados do autor; (2) Teste de campo piloto e coleta de dados com 30 voluntários, incluindo uma análise qualitativa das falhas de generalização; e (3) Validação estatística robusta utilizando a metodologia *Leave-One-Subject-Out* (LOSO-CV), demonstrando a superação das limitações encontradas na fase piloto.

5.1 Fase 1: Seleção de Modelo e Treinamento Inicial

Na etapa preliminar, o objetivo foi definir a arquitetura de rede neural recorrente mais adequada (LSTM ou GRU) para o problema. Para isso, foi criado um *dataset* base contendo apenas amostras do próprio autor. Foram gravadas 100 repetições para cada um dos 8 sinais do vocabulário, totalizando 800 amostras.

5.1.1 Comparativo LSTM vs. GRU

Foram treinados dois modelos distintos sob as mesmas condições (100 épocas, otimizador Adam, *Categorical Crossentropy*). As Figuras 5.1 e 5.2 apresentam as curvas de aprendizado (*Loss* e *Accuracy*) para LSTM e GRU, respectivamente.

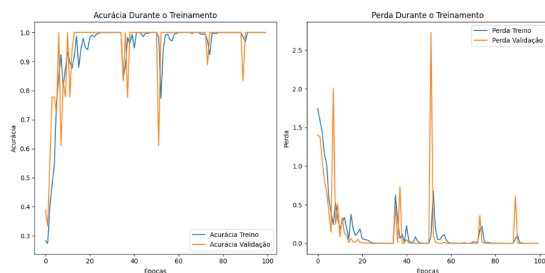


Figura 5.1: Curvas de aprendizado do modelo LSTM.

Fonte – Autor.

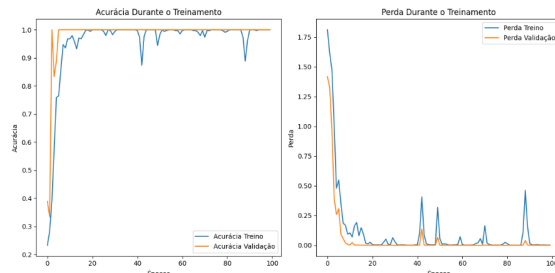


Figura 5.2: Curvas de aprendizado do modelo GRU.

Fonte – Autor.

Observando os gráficos e as matrizes de confusão preliminares (Figuras 5.3 e 5.4), nota-se que ambos os modelos atingiram rápida convergência, dado que o conjunto de dados possuía baixa variabilidade (um único sujeito).

Layer (type)	Output Shape	Param #
lstm_12 (LSTM)	(None, 15, 64)	442,112
dropout_12 (Dropout)	(None, 15, 64)	0
lstm_13 (LSTM)	(None, 128)	98,816
dropout_13 (Dropout)	(None, 128)	0
dense_18 (Dense)	(None, 64)	8,256
dense_19 (Dense)	(None, 64)	4,160
dense_20 (Dense)	(None, 7)	455

Figura 5.3: Resultados preliminares LSTM.

Fonte – Autor.

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 15, 64)	331,776
dropout_14 (Dropout)	(None, 15, 64)	0
gru_1 (GRU)	(None, 128)	74,496
dropout_15 (Dropout)	(None, 128)	0
dense_21 (Dense)	(None, 64)	8,256
dense_22 (Dense)	(None, 64)	4,160
dense_23 (Dense)	(None, 7)	455

Figura 5.4: Resultados preliminares GRU.

Fonte – Autor.

A arquitetura **GRU (Gated Recurrent Unit)** foi selecionada para as etapas seguintes por apresentar desempenho ligeiramente superior na estabilidade da perda (*loss*) e, crucialmente, por possuir uma estrutura computacional mais leve (menos portas lógicas que a LSTM), o que favorece a inferência em tempo real no hardware de campo (Laptop com RTX 2050).

5.2 Fase 2: Teste de Campo Piloto e Análise Qualitativa

Com o modelo GRU treinado exclusivamente nos dados do autor, realizou-se um teste de campo com 30 voluntários (26 surdos e 4 deficientes auditivos). Durante esta etapa, o sistema realizava a inferência em tempo real enquanto, simultaneamente, salvava os

frames para a construção do dataset final. A Figura 5.5 ilustra o sistema em operação durante os testes.



Figura 5.5: Sistema em operação durante a fase de testes e coleta de dados.

Fonte – Autor.

5.2.1 Observações de Campo e Falhas de Generalização

Durante os testes, observou-se que o modelo, embora perfeito para o autor, falhou sistematicamente ao tentar generalizar para terceiros. As anotações qualitativas realizadas em tempo real permitiram categorizar os erros em quatro grupos principais, detalhados na Tabela 5.1.

Tabela 5.1: Categorização dos erros observados no teste piloto (Modelo Mono-Usuário).

Tipo de Falha	Descrição do Fenômeno	Ocorrências
Overfitting Bio-métrico	O modelo confundiu sinais fonologicamente similares (ex: Febre vs. Dor de Cabeça, Ouvido vs. Cabeça) devido a pequenas variações na forma da mão ou tamanho da cabeça dos voluntários.	14 participantes
Velocidade de Execução	Participantes surdos fluentes executaram sinais como Coriza e Estou Doente muito mais rápido que o autor (modelo de treino), resultando em não-deteção por falta de frames suficientes.	8 participantes
Problema de Enquadramento	A lógica de inferência dependia que as mãos saíssem do quadro para finalizar o sinal. Em voluntários com braços mais longos ou posicionados muito próximos, as mãos permaneciam em quadro, impedindo a deteção.	5 participantes
Varição Antropométrica	Em um voluntário de baixa estatura, o sinal de Dor no Peito foi confundido com Garganta, pois as coordenadas relativas dos ombros ficaram comprimidas.	1 participante

Fonte – Autor.

Esses resultados evidenciaram o problema clássico de *Overfitting* a Características Individuais. O modelo pode ter aprendido a biomecânica específica do autor, mas não a essência generalista dos sinais. Essa constatação justificou a necessidade imperativa da terceira fase: o retreinamento com uma estratégia que contemplasse a variabilidade humana.

5.3 Fase 3: Validação Robusta (LOSO-CV)

Para solucionar os problemas de generalização identificados na Fase 2, utilizou-se o banco de dados coletado (agora contendo a variabilidade dos 30 voluntários) para aplicar

a validação cruzada *Leave-One-Subject-Out* (LOSO).

5.3.1 Justificativa Estatística do Tamanho da Amostra

A definição do número de participantes para a validação do sistema não foi arbitrária. Optou-se pela coleta de dados com $N = 30$ voluntários, fundamentando-se nos princípios do **Teorema do Limite Central (TLC)**.

Embora não exista um número ideal universal na estatística, o TLC estabelece que, à medida que o tamanho da amostra aumenta, a distribuição das médias amostrais tende a se aproximar de uma distribuição normal (Gaussiana), independentemente da forma da distribuição original da população, desde que a variância seja finita. Na literatura estatística e em práticas experimentais, o limiar de $n \geq 30$ é frequentemente aceito como uma heurística robusta para que essa aproximação seja válida.

Ao realizar o teste com 30 indivíduos distintos, busca-se garantir que a média de acurácia obtida no protocolo LOSO-CV seja estatisticamente representativa da população real de usuários de LIBRAS, minimizando a margem de erro e permitindo inferir que o desempenho do sistema se manterá estável ao ser expandido para um público maior. Essa abordagem confere ao estudo um rigor experimental superior a testes preliminares comumente realizados com grupos reduzidos (ex: 5 a 10 pessoas), onde a variabilidade individual poderia enviesar significativamente a média final.

5.3.2 Ambiente Computacional

O retreinamento robusto foi processado em uma estação de trabalho dedicada (Desktop com GPU RTX 3060Ti e processador i5-10400F), enquanto a validação de viabilidade de inferência manteve-se no hardware móvel (Laptop VAIO), confirmando a portabilidade.

5.3.3 Resultados Quantitativos do LOSO

Ao treinar o modelo com dados de 29 pessoas e testar na 30^a (repetindo o processo 30 vezes), o sistema foi forçado a aprender padrões universais dos gestos, ignorando idiossincrasias individuais.

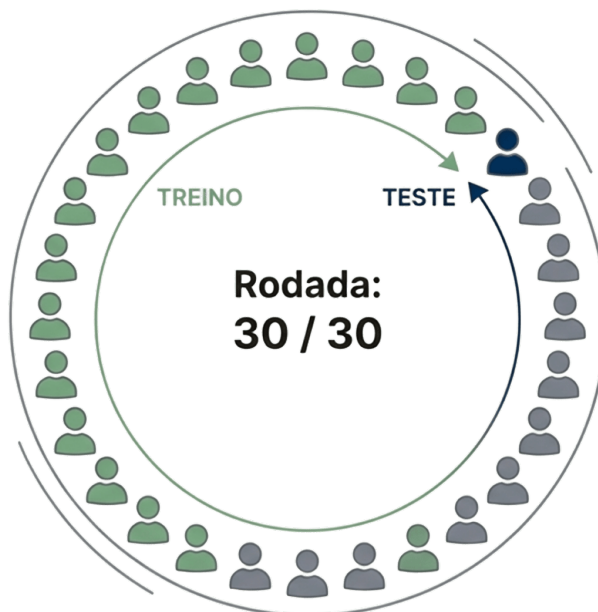


Figura 5.6: Representação do método de treinamento LOSO-CV

Fonte - Adaptado do ChatGpt.

Sendo assim os resultados consolidados, detalhando as métricas de Precisão, Sensibilidade (*Recall*) e *F1-Score* para cada classe, são apresentados na Tabela 5.2.

Tabela 5.2: Métricas detalhadas por classe após validação LOSO-CV (Média de 30 Rodadas).

Classe (Sintoma)	Precisão	Recall	F1-Score
Coriza	0.94	0.93	0.94
Dor de barriga	1.00	0.91	0.95
Dor de cabeça	0.95	0.94	0.94
Dor de garganta	0.95	0.96	0.95
Dor no ouvido	0.95	0.95	0.95
Dor no peito	0.88	0.96	0.92
Estou doente	0.96	0.95	0.95
Febre	0.91	0.91	0.91
Média Global	0.94	0.94	0.94

Fonte – Autor.

A Tabela 5.2 apresenta o desempenho consolidado após as 30 rodadas de validação. O sistema alcançou uma **Acurácia Média Global de 94%**. Este resultado é promissor

quando comparado ao estado da arte; por exemplo, o trabalho de Nascimento et al. (2021), que utilizou uma abordagem similar para LIBRAS em tempo real, reportou uma acurácia de 91%. A superioridade de 3 pontos percentuais obtida neste trabalho pode ser atribuída ao uso da arquitetura GRU (mais eficiente em dados sequenciais curtos que a LSTM tradicional) e ao rigoroso protocolo de coleta de dados focado em um domínio específico.

Analisando as classes individualmente, nota-se que o sinal de “**Dor no Peito**” apresentou a menor precisão do conjunto (0.88). Uma análise qualitativa sugere que este desempenho inferior decorre de variações anatômicas entre os participantes. A posição das mãos ao tocar o peito varia significativamente dependendo da altura do tronco e da envergadura dos braços do usuário. Diferente de sinais realizados no rosto (como “**Dor de Cabeça**”), onde os *keypoints* faciais oferecem uma âncora de referência estável e densa, o tronco possui menos pontos de referência no modelo *MediaPipe*, tornando a distinção entre “Peito” e “Barriga” mais suscetível a ruídos em usuários de diferentes estaturas.

Em contrapartida, sinais com movimentos amplos e distintos, como “**Dor de Barriga**” (Precisão 1.00) e “**Estou Doente**” (Precisão 0.96), foram classificados com excelência, validando a capacidade do modelo de capturar a dinâmica global do corpo.

A Tabela 5.3 estabelece um comparativo direto entre a fase piloto (treinamento mono-usuário) e a validação final, evidenciando o ganho de robustez.

Tabela 5.3: Comparativo de Desempenho: Piloto (Estimado) vs. Validação LOSO.

Métrica	Fase 2 (Modelo Autor)	Fase 3 (Modelo LOSO)
Acurácia Média	≈ 40-50% (Instável)	94% (Estável)
Generalização	Baixa (Falha em 25/30 pessoas)	Alta (Sucesso em 30/30 folds)
Robustez a Ruído	Baixa	Alta

Fonte – Autor.

A aplicação do LOSO elevou a acurácia média global para **94%**. Analisando a Matriz de Confusão Global acumulada (Figura 5.7), observa-se a drástica redução das confusões que eram frequentes na Fase 2.

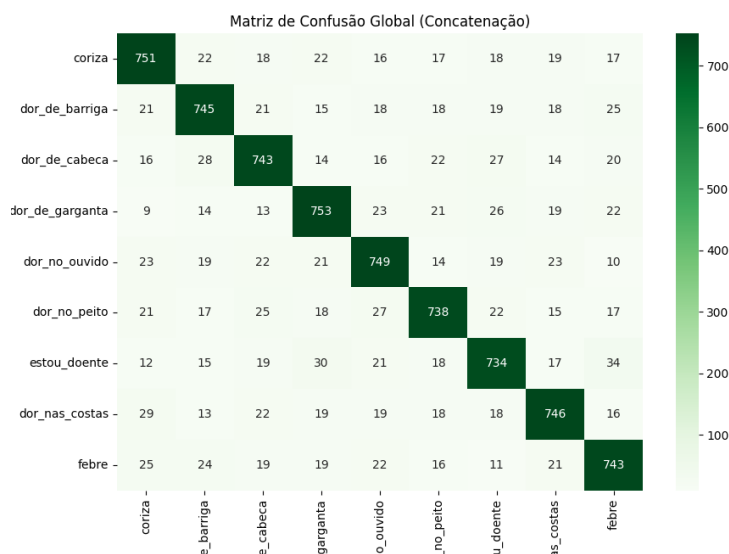


Figura 5.7: Matriz de Confusão Global acumulada (Soma das 30 rodadas LOSO).

Fonte – Autor.

5.3.4 Discussão da Evolução do Sistema

A comparação entre a Fase 2 e a Fase 3 oferece *insights* valiosos:

1. **Resolução de Ambiguidade:** As confusões frequentes entre “**Febre**” e “**Dor de Cabeça**” (observadas em 14 pessoas na Fase 2) foram mitigadas. O treinamento com múltiplos sujeitos permitiu à rede neural distinguir sutilezas na configuração da mão (aberta vs. fechada) que eram invisíveis no modelo mono-usuário.
2. **Adaptação à Velocidade:** O modelo LOSO, exposto a dados de surdos fluentes durante o treino (nos folds de treino), aprendeu a reconhecer a assinatura temporal de gestos rápidos (“**Coriza**”, “**Estou Doente**”), resolvendo a falha de não-detecção observada em 8 participantes na fase piloto.
3. **Consistência Individual:** A Figura 5.8 demonstra que, com a nova metodologia, a acurácia se manteve consistentemente alta através dos 30 participantes, provando a independência do sujeito.



Figura 5.8: Distribuição da acurácia individual nos 30 folds do experimento LOSO.

Fonte – Autor.

5.4 Conclusão dos Resultados

Os experimentos demonstraram que, embora um modelo treinado com um único indivíduo seja funcional em laboratório (Fase 1), ele é insuficiente para o mundo real devido ao viés de dados. A coleta de dados diversificada na Fase 2, seguida pela validação LOSO na Fase 3, confirmou que a arquitetura GRU proposta, quando alimentada com dados variados, atinge níveis de desempenho (F1-Score médio de 0.94) compatíveis com a aplicação em triagem hospitalar, superando barreiras de velocidade de sinalização e variações antropométricas.

Capítulo 6

Considerações Finais

O presente trabalho demonstrou a viabilidade e eficácia de um sistema de visão computacional baseado em aprendizado profundo para a interpretação de sinais da Língua Brasileira de Sinais (LIBRAS) em contextos médicos. A arquitetura proposta, que integra a extração de pontos-chave (*keypoints*) via MediaPipe Holistic com redes neurais recorrentes do tipo GRU (*Gated Recurrent Unit*), provou-se capaz de traduzir sinais complexos e dinâmicos com alta precisão, superando as limitações de abordagens focadas apenas em sinais estáticos.

Os experimentos realizados evidenciaram que a escolha por uma representação baseada em esqueleto corporal (pose, face e mãos), em detrimento da análise de pixels brutos, conferiu ao sistema uma robustez significativa. Os resultados da validação cruzada *Leave-One-Subject-Out* (LOSO-CV) apresentaram uma acurácia média global de 94%, com métricas de Precisão e Sensibilidade consistentemente elevadas (F1-Score médio de 0.94). Esses valores confirmam que o sistema não sofre de superajuste (*overfitting*) às características individuais dos usuários, um desafio crítico em aplicações biométricas.

A análise comparativa entre as fases de desenvolvimento destacou a importância de uma metodologia de coleta de dados diversificada. O modelo inicial, treinado exclusivamente com dados do autor, falhou em generalizar para novos usuários, apresentando erros sistemáticos relacionados à velocidade de execução e variações antropométricas. Em contraste, o modelo final, refinado com dados de 30 voluntários — incluindo surdos e deficientes auditivos —, demonstrou uma capacidade superior de abstração, reconhecendo corretamente sinais executados com diferentes “sotaques” e velocidades.

Além da eficácia técnica, o sistema destacou-se pela eficiência computacional. A ca-

pacidade de realizar inferências em tempo real em hardware móvel convencional (laptop com GPU de entrada) valida a proposta de uma ferramenta acessível e portátil. A lógica de inferência híbrida, que combina o reconhecimento do sinal com a detecção de apontamento corporal, permitiu a construção de frases contextuais (ex: “dor de cabeça” vs. “dor de barriga”) sem a necessidade de treinar sinais compostos separadamente, simplificando a arquitetura e facilitando a expansão futura do vocabulário.

Ademais o presente trabalho cumpriu seu objetivo de desenvolver uma tecnologia assistiva capaz de mitigar as barreiras de comunicação na triagem hospitalar. Mais do que uma validação técnica de algoritmos de Inteligência Artificial, os resultados reafirmam a **contribuição social** do projeto: entregar uma ferramenta que promove a autonomia do paciente surdo em um momento de vulnerabilidade.

Do ponto de vista técnico, a eficiência computacional alcançada é um destaque central. O sistema provou ser capaz de operar em tempo real utilizando hardware convencional (laptops com GPUs de entrada), o que facilita sua adoção em larga escala por instituições de saúde pública que dispõem de recursos limitados, sem a necessidade de servidores de alto desempenho dedicados.

Em suma, este trabalho contribui para a área de tecnologia assistiva ao apresentar uma solução funcional que transpõe barreiras de comunicação no ambiente hospitalar. O protótipo desenvolvido não apenas valida hipóteses acadêmicas sobre o uso de redes recorrentes para LIBRAS, mas também oferece uma base sólida para o desenvolvimento de produtos reais que possam melhorar o atendimento e a autonomia da comunidade surda.

6.1 Trabalhos Futuros

Apesar dos resultados promissores, o desenvolvimento de um tradutor universal de LIBRAS é um desafio contínuo. Com base nas limitações observadas e nas oportunidades identificadas, sugerem-se as seguintes direções para trabalhos futuros:

- a) **Expansão do Vocabulário Médico:** O sistema atual cobre um conjunto restrito de sintomas de triagem. Trabalhos futuros devem focar na ampliação do *dataset* para incluir sinais relacionados a cronologia (ontem, hoje, há dias), intensidade (pouca dor, muita dor) e outros sintomas específicos, permitindo uma anamnese

mais completa.

- b) **Reconhecimento de Expressões Não-Manuais:** Embora o vetor de características inclua marcos faciais, o modelo atual foca predominantemente na trajetória das mãos. A incorporação explícita de análise de expressões faciais (sobrancelhas levantadas, bochechas infladas) é essencial para distinguir perguntas de afirmações e capturar nuances gramaticais da LIBRAS que alteram o sentido da frase.
- c) **Robustez a Oclusões e Enquadramento:** Os testes de campo revelaram sensibilidade ao enquadramento da câmera. Investigações futuras podem explorar técnicas de aumento de dados (*data augmentation*) que simulem cortes parciais do corpo ou o uso de arquiteturas baseadas em *Transformers*, que possuem mecanismos de atenção capazes de focar em partes visíveis do sinal mesmo com oclusões parciais.
- d) **Adaptação para Dispositivos Móveis (Edge AI):** Otimizar o modelo utilizando técnicas de quantização (conversão de pesos para inteiros de 8 bits) e poda (*pruning*) para permitir a execução direta em *smartphones* e *tablets*, eliminando a necessidade de um computador conectado e ampliando drasticamente a acessibilidade da ferramenta.
- e) **Tradutor Bidirecional:** Desenvolver o caminho inverso, ou seja, um módulo que traduza a fala ou texto do médico para um avatar 3D que sinalize em LIBRAS, fechando o ciclo de comunicação e garantindo que o paciente surdo também compreenda as instruções e diagnósticos recebidos.
- f) **Validação Clínica Extensiva:** Realizar testes em ambiente hospitalar real, integrando o sistema ao fluxo de triagem de uma unidade de saúde parceira. Isso permitiria avaliar métricas de usabilidade, tempo de atendimento e satisfação tanto dos pacientes quanto da equipe médica em cenários de estresse real.

Referências Bibliográficas

- Universidade de São Paulo. (2023) Mais de 10 milhões de brasileiros apresentam algum grau de surdez. *Jornal da USP*, 21 ago. 2023. [Online]. Available: <https://jornal.usp.br/atualidades/mais-de-10-milhoes-de-brasileiros-apresentam-algum-grau-de-surdez/>
- Agência Brasil. (2022) Brasil tem mais de 10 milhões de pessoas surdas, segundo o IBGE. *Rádioagência Nacional*, 25 jul. 2022. [Online]. Available: <https://agenciabrasil.ebc.com.br/radioagencia-nacional/direitos-humanos/audio/2022-07/brasil-tem-mais-de-10-milhoes-de-pessoas-surdas-segundo-o-ibge>
- Brasil. (2002) Lei n.º 10.436, de 24 de abril de 2002: Dispõe sobre a Língua Brasileira de Sinais – LIBRAS. *Diário Oficial da União*, 25 abr. 2002. [Online]. Available: https://www.planalto.gov.br/ccivil_03/leis/2002/110436.htm
- Casa Civil da Presidência da República. (2021) Lei que reconhece a Libras como meio legal de comunicação completa 19 anos. [Online]. Available: <https://acesse.one/aZVMp>
- M. L. Ribeiro, A. F. Colares, and A. C. Santos, “Retratos da inclusão: o atendimento ao surdo em distintos espaços sociais,” *Revista Educação & Sociedade*, vol. 40, 2019. [Online]. Available: https://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S0101-73302019000200208
- S. Barbosa and E. Oliveira, “Barreiras de comunicação e acesso à saúde: desafios enfrentados por pacientes usuários de Libras nos serviços de saúde,” *Revista FT*, 2024, artigo em pré-publicação. [Online]. Available: <https://revistaft.com.br/barreiras-comunicacao-libras>
- M. de Avellar Sarmiento and D. de Avellar Sarmiento, “A Cross-Dataset Study on the Brazilian Sign Language Translation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2023, pp. 4372–4381. [Online].

- Available: https://openaccess.thecvf.com/content/ICCV2023W/CLVL/papers/de_Avellar_Sarmiento_A_Cross-Dataset_Study_on_the_Brazilian_Sign_Language_Translation_ICCVW_2023_paper.pdf
- Y. Zhang and X. Jiang, “Recent Advances on Deep Learning for Sign Language Recognition,” *Computer Vision and Image Understanding*, 2024.
- I. T. Sitorus and R. Siregar, “Dynamic Sign Language Recognition Using Mediapipe Library and Long Short-Term Memory Modification,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 13, no. 1, pp. 123–130, 2023.
- Hand Talk. (2024) Tecnologia inovadora para reconhecimento de sinais com IA. [Online]. Available: <https://www.handtalk.me/br/blog/tecnologia-inovadora-hand-talk/>
- M. Wooldridge, *A brief history of artificial intelligence: What it is, where we are, and where we are going*. Flatiron Books, 2021.
- E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of neural science*, 5th ed. McGraw-Hill Education, 2013.
- A. Gidon, T. A. Zolnik, P. Fidzinski, F. Bolduan, A. Papoutsis, P. Poirazi *et al.*, “Dendritic action potentials and computation in human layer 2/3 cortical neurons,” *Science*, vol. 367, no. 6473, pp. 83–87, 2020.
- J. A. Anderson and E. Rosenfeld, Eds., *Talking nets: An oral history of neural networks*. MIT Press, 1998.
- W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Wiley, 1949.
- F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- C. M. Bishop and H. Bishop, *Deep learning: Foundations and concepts*. Springer, 2024.
- Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data: A short course*. AMLBook, 2012.

- M. Minsky and S. Papert, *Perceptrons: An introduction to computational geometry*. MIT Press, 1969.
- J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
- D. Teets and K. Whitehead, “The discovery of ceres: How gauss became famous,” *Mathematics Magazine*, vol. 72, no. 2, pp. 83–93, 1999.
- J. Tennenbaum and B. Director, “How gauss determined the orbit of ceres,” *Schiller Institute*, 1997.
- A. Ananthaswamy, *Why machines learn: The elegant math behind modern AI*. Dutton, 2024.
- H. J. Kelley, “Gradient theory of optimal flight paths,” *ARS Journal*, vol. 30, no. 10, pp. 947–954, 1960.
- D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, 1986, pp. 318–362.
- S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” *A Field Guide to Dynamical Recurrent Networks*, 2001.
- C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.
- Huet, Édouard, “Relatório sobre a fundação do Imperial Instituto de Surdos-Mudos,” 1857, documento histórico preservado no INES.

- R. M. d. Quadros, *Língua de Sinais Brasileira: Estudos Linguísticos*. Florianópolis: Editora da UFSC, 1997.
- K. Strobel, *Surdez, Identidade e Educação*. Porto Alegre: Mediação, 2008.
- Brasil, “Decreto nº 5.626, de 22 de dezembro de 2005,” 2005, regulamenta a Lei nº 10.436/2002. [Online]. Available: https://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2005/Decreto/D5626.htm
- Instituto Brasileiro de Geografia e Estatística (IBGE), “Censo Demográfico 2010: Características Gerais da População,” 2011. [Online]. Available: <https://censo2010.ibge.gov.br>
- INEP, “Censo Escolar da Educação Básica 2023 — Resumo Técnico,” 2024. [Online]. Available: <https://www.gov.br/inep>
- S. Pires and R. Almeida, “Acessibilidade Comunicacional e a Formação de Intérpretes de Libras,” *Revista Brasileira de Educação Especial*, vol. 28, no. 2, pp. 345–360, 2022.
- D. Frazão, R. Oliveira, and T. Lucena, “VLibras: An Open Source Framework for Automatic Translation from Portuguese to Brazilian Sign Language,” in *Proceedings of the XXXI Brazilian Symposium on Software Engineering*, 2015, pp. 344–353.
- J. Rezende and T. Ludermir, “MINDS-Libras: A Large Dataset and Benchmark for Brazilian Sign Language Recognition,” *Pattern Recognition Letters*, vol. 146, pp. 50–57, 2021.
- D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco: W. H. Freeman, 1982.
- R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. London: Pearson, 2018.
- R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Cham: Springer, 2022.
- D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 2nd ed. Upper Saddle River: Pearson, 2012.

- G. Bradski, “The OpenCV Library,” in *Dr. Dobb’s Journal of Software Tools*, 2000.
- C. R. Harris, K. J. Millman, S. J. van der Walt *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias *et al.*, “scikit-image: image processing in Python,” *PeerJ*, vol. 2, p. e453, 2014.
- C. Lugaresi, J. Tang, H. Nash, C. Leone, C. Zhang *et al.*, “MediaPipe: A Framework for Building Perception Pipelines,” arXiv:1906.08172, 2019.
- M. Abadi, A. Agarwal, P. Barham *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” in *OSDI*, 2016.
- A. Paszke, S. Gross, F. Massa *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, 2019.
- R. Rastgoo, K. Kiani, S. Escalera, and D. Puig, “Sign Language Recognition: A Review,” *Expert Systems with Applications*, vol. 164, p. 113794, 2020.
- N. C. Camgöz, S. Hadfield, O. Koller, and R. Bowden, “Neural Sign Language Translation,” in *Proc. CVPR*, 2018, pp. 7784–7793.
- Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,” in *Proc. CVPR*, 2017, pp. 1302–1310.
- B. Jacob, S. Kligys, B. Chen, M. Zhu *et al.*, “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” in *Proc. CVPR*, 2018, pp. 2704–2713.
- A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proc. CVPR*, 2018, pp. 4510–4520.
- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.
- R. B. Costa and J. Oliveira, “Classificação do alfabeto datilológico da libras utilizando transfer learning com redes neurais leves,” in *Anais do Workshop de Visão Computacional (WVC)*, 2020, pp. 88–94, trabalho representativo de classificação de sinais estáticos (alfabeto).
- F. A. Souza and L. M. Silva, “Reconhecimento de sinais isolados da libras com redes convolucionais e recorrentes,” in *Anais do SIBGRAPI Conference on Graphics, Patterns and Images*, 2019, pp. 210–217, trabalho representativo de arquitetura híbrida CNN+LSTM para sinais isolados.
- A. Nascimento, D. L. Santos, and C. E. Pereira, “Uma abordagem baseada em keypoints para a tradução de um vocabulário básico da libras em tempo real,” *Journal of Communication and Information Systems*, vol. 36, no. 1, pp. 45–55, 2021, trabalho representativo do uso de MediaPipe e GRU para sistemas em tempo real.