Revista Eletrônica de Iniciação Científica em Computação, 2026, 24:1
ISSN: 3085-8461 • doi: 10.5753/reic.2026.8428

Artigo de Pesquisa/Research Paper

Uso de Redes Neurais *BiLSTM* com Atenção para o Reconhecimento Automático de Gestos Isolados em Libras

BiLSTM Neural Networks with Attention for Automatic Recognition of Isolated Signs in Libras

Túlio Castro [Ibmec Belo Horizonte | *tulio.castro14@gmail.com*]
Pedro Calais [Ibmec Belo Horizonte| *pedro.calais@gmail.com*]

Ibmec Belo Horizonte, Belo Horizonte, MG, Brasil.

Resumo. Este trabalho apresenta o desenvolvimento e a validação de um sistema leve para reconhecimento de sinais isolados da Língua Brasileira de Sinais (Libras), utilizando *landmarks* das mãos extraídos pelo *MediaPipe* e uma arquitetura temporal baseada em *BiLSTM* com atenção. A entrada do sistema consiste em um vídeo do gesto, convertido em sequências de 126 características por quadro, correspondentes às coordenadas tridimensionais de até duas mãos. O modelo foi avaliado em 55 sinais e alcançou acurácia de 98,67%, com aproximadamente 1,04 milhão de parâmetros. Além da avaliação experimental, o modelo foi integrado a uma aplicação *web*, permitindo inferência em tempo real a partir da câmera do usuário. Os resultados indicam que representações baseadas em *landmarks* são suficientes para o reconhecimento eficiente de sinais isolados, viabilizando uma solução com latência interativa e custo computacional controlado.

Abstract. This work presents the development and validation of a lightweight web-based system for isolated sign recognition in Brazilian Sign Language (Libras). The system uses hand landmarks extracted with MediaPipe as input and a temporal BiLSTM-based architecture with attention for classification. Each video is converted into a sequence of 126 features per frame, corresponding to the three-dimensional coordinates of up to two hands. The model was evaluated on a dataset of 55 signs, achieving 98.67% accuracy with approximately 1.04 million parameters. In addition to the experimental evaluation, the trained model was integrated into a web application, enabling real-time inference from the user's camera. The results indicate that landmark-based representations are sufficient for efficient isolated sign recognition, supporting accessible applications with interactive latency and controlled computational cost.

Palavras-chave: Libras, reconhecimento de sinais, *BiLSTM*, *MediaPipe*, visão computacional

Keywords: Libras, *sign language recognition*, *BiLSTM*, *MediaPipe*, *computer vision*

Recebido/Received: 09 June 2026 • **Aceito/Accepted:** 12 June 2026 • **Publicado/Published:** 10 July 2026

1 Introdução

1.1 Contexto e motivação

A Língua Brasileira de Sinais (Libras) é um dos principais meios de comunicação da comunidade surda no Brasil, composta por um conjunto estruturado de gestos manuais, expressões faciais e movimentos corporais. Assim como outras línguas de sinais, a Libras possui gramática e léxico próprios, distintos da língua oral e escrita, exigindo conhecimento específico para sua interpretação. Entretanto, grande parte da população não domina Libras, o que cria barreiras significativas de comunicação entre pessoas surdas e ouvintes World Health Organization [2026]; World Federation of the Deaf [2026].

Segundo dados da Pesquisa Nacional de Saúde de 2019, aproximadamente 2,3 milhões de pessoas possuíam deficiência auditiva no Brasil Instituto Brasileiro de Geografia e Estatística [2019a]. No entanto, o conhecimento de Libras ainda é restrito dentro desse grupo: entre pessoas de cinco anos ou mais que declararam alguma dificuldade permanente para ouvir, apenas 1,8% afirmaram saber usar Libras; entre aquelas que não conseguiam ouvir de modo algum, esse percentual chegou a 35,8% Instituto Brasileiro de Geografia e Estatística [2019b, 2021]. Esses números evidenciam a lacuna existente entre a população que pode depender de recursos visuais de comunicação e a quantidade de pessoas capazes de utilizar Libras na comunicação cotidiana.

Nesse contexto, sistemas automáticos de reconhecimento de sinais podem atuar como ferramentas de apoio à inclusão, possibilitando a transcrição ou interpretação de gestos em tempo quase real. O avanço de técnicas de *Visão Computacional* e *Aprendizado Profundo* tem permitido extrair e processar características visuais a partir de imagens e vídeos com alta precisão Zhang and Jiang [2024]. Contudo, muitos modelos de alto desempenho dependem de arquiteturas complexas, alto custo computacional ou múltiplas modalidades de entrada, o que limita sua aplicação em dispositivos comuns e ambientes *web*.

Diante disso, este trabalho explora uma alternativa baseada exclusivamente nos *landmarks* das mãos. Em vez de processar o vídeo bruto diretamente, cada quadro é convertido em coordenadas tridimensionais extraídas pelo *MediaPipe*, reduzindo a dimensionalidade da entrada e preservando informações relevantes sobre configuração, posição e movimento das mãos. A partir dessas sequências, foi treinado um modelo temporal compacto, posteriormente integrado a uma aplicação *web* para execução em tempo real.

1.2 Objetivo

Este trabalho teve como objetivo desenvolver, treinar e disponibilizar um sistema de reconhecimento de sinais isolados de Libras em ambiente *web*, com foco em baixa latência, baixo custo computacional e viabilidade prática. Para isso, foi implementado um fluxo completo composto por extração de *landmarks* das mãos a partir de vídeo, organização das sequências temporais, treinamento de um modelo *BiLSTM* com atenção e integração do modelo a uma aplicação *web* para inferência em tempo real.

A solução proposta utilizou o *MediaPipe Hands* para detectar até duas mãos e extrair 21 pontos tridimensionais por mão, totalizando 126 características por quadro. Essas sequências foram processadas por uma arquitetura temporal compacta, capaz de capturar a dinâmica dos gestos sem depender do processamento direto dos quadros *RGB*. Dessa forma, o sistema buscou equilibrar desempenho preditivo e eficiência computacional, permitindo sua execução em cenários de uso acessíveis, como navegadores *web* e dispositivos convencionais.

2 Revisão de Literatura

O reconhecimento automático de línguas de sinais evoluiu de abordagens baseadas em sensores físicos para métodos baseados em visão computacional e aprendizado profundo. Inicialmente, eram comuns soluções com luvas instrumentadas e técnicas estatísticas, como Modelos de Markov Ocultos (*HMM*) e *Dynamic Time Warping* (*DTW*), capazes de modelar sequências temporais em ambientes controlados. Apesar de úteis, essas abordagens dependiam de dispositivos específicos, extração manual de características e apresentavam baixa escalabilidade para aplicações práticas Zhang and Jiang [2024].

Com o avanço do *deep learning*, redes convolucionais passaram a ser empregadas para extrair características espaciais de imagens e vídeos, enquanto *LSTMs*, arquiteturas híbridas *CNN-LSTM* e mecanismos de atenção foram utilizados para representar a dinâmica temporal dos gestos. Esses métodos permitiram ganhos relevantes de desempenho, principalmente por aprenderem automaticamente padrões visuais e temporais a partir dos dados Kumari and Anand [2024]; Zhang and Jiang [2024].

No contexto da Libras, trabalhos recentes exploram combinações de *CNNs*, *LSTMs* e representações baseadas em *landmarks*, destacando desafios como generalização entre bases, variação entre participantes e escassez de conjuntos públicos amplos de Avellar Sarmento and Ponti [2023]; Alves *et al*. [2024]. Entre as bases nacionais, destaca-se a MINDS-Libras, composta por sinais isolados capturados em condições controladas e com múltiplos participantes Rezende *et al*. [2021]; Rezende [2021]. Internacionalmente, bases de *ASL* como *MS-ASL*, *WLASL* e *ASL Citizen* contribuíram para o treinamento de modelos mais robustos, enquanto conjuntos como *How2Sign* e *OpenASL* impulsionaram estudos em sinais contínuos e tradução automática Vaezi Joze and Koller [2019]; Li *et al*. [2020]; Desai *et al*. [2023]; Duarte *et al*. [2021]; Shi *et al*. [2022].

De modo geral, a literatura indica que modelos baseados em visão computacional e aprendizado profundo têm elevado a precisão no reconhecimento de sinais. Entretanto, ainda há desafios relacionados à robustez em cenários reais, diversidade dos dados e custo computacional. Nesse sentido, abordagens baseadas em *landmarks* representam uma alternativa promissora, pois reduzem a dimensionalidade da entrada e preservam informações relevantes sobre posição, configuração e movimento das mãos.

3 Metodologia

O fluxo metodológico foi estruturado em três etapas: (i) coleta e representação dos dados; (ii) normalização e aumento de dados; (iii) modelagem, treinamento e avaliação do classificador.

3.1 Coleta e Representação dos Dados

A entrada do processo de extração é um vídeo contendo a execução de um sinal isolado de Libras, capturado por câmera ou proveniente da base original. Cada vídeo é processado quadro a quadro com o *MediaPipe Hands* (*Hand Landmarker*)[1], que detecta até duas mãos e estima 21 *landmarks* tridimensionais por mão. Cada ponto é representado por coordenadas normalizadas (x, y, z). A Figura 1 ilustra a indexação desses pontos de referência.

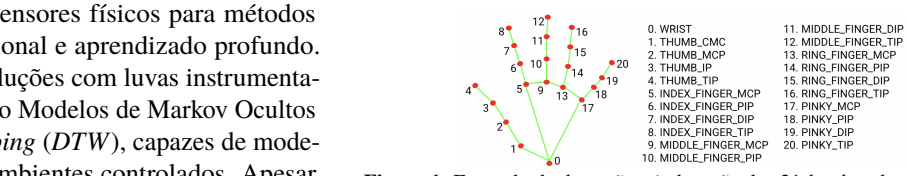


Figura 1. Exemplo de detecção e indexação dos 21 *landmarks* pelo *MediaPipe Hands* Google [2024].

Para uma mão detectada no quadro f, isto é, no f-ésimo quadro da sequência temporal extraída do vídeo, o vetor de pontos é definido como:

$$\mathbf{L}_f = \big[(x_0, y_0, z_0), \ldots, (x_{20}, y_{20}, z_{20})\big]. \tag{1}$$

Nessa representação, $\mathbf{L}_f$ é o conjunto de *landmarks* de uma mão no quadro f; cada tripla (x_j, y_j, z_j) representa as coordenadas normalizadas do ponto anatômico j, com j variando de 0 a 20 conforme a indexação do *MediaPipe*.

Como o modelo considera até duas mãos, os vetores das mãos detectadas são concatenados:

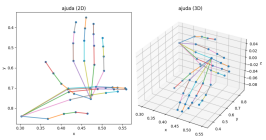
$$\mathbf{V}_f = [\mathbf{L}_f^{(\mathrm{D})} \parallel \mathbf{L}_f^{(\mathrm{E})}], \tag{2}$$

em que $\mathbf{V}_f$ é o vetor final de características do quadro f; $\mathbf{L}_f^{(\mathrm{D})}$ e $\mathbf{L}_f^{(\mathrm{E})}$ representam, respectivamente, os *landmarks* da mão direita e da mão esquerda. Dessa forma, cada quadro é convertido em um vetor de dimensão fixa com $126 = 2 \times 21 \times 3$ atributos. Quando apenas uma mão é detectada, aplica-se *zero-padding* ao vetor da mão ausente para manter a dimensionalidade constante. Exemplos de extração 2D/3D para três gestos são apresentados na Figura 2.

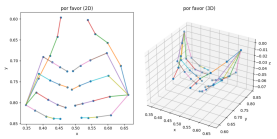
A sequência completa é representada por uma matriz:

$$X = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_T]^\top \in \mathbb{R}^{T \times 126}, \tag{3}$$

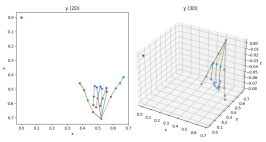
[1] Disponível em: `https://developers.google.com/mediapipe`



(a) Ajuda



(b) Por favor



(c) Letra Y

Figura 2. Exemplos de representação 2D/3D dos *landmarks* das mãos para diferentes gestos.

em que X é a matriz temporal que representa uma execução completa do sinal, $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_T$ são os vetores de características dos quadros 1 até T,onde T é o número de quadros utilizados para representar o sinal e $\mathbb{R}^{T\times 126}$ indica que X pertence ao espaço das matrizes reais com T linhas e 126 colunas. Nas coletas complementares, cada execução foi armazenada como uma sequência de $T = 32$ quadros. Essas matrizes são salvas como amostras individuais e organizadas em diretórios separados por classe, de modo que cada sinal possui uma pasta própria contendo suas respectivas execuções. Essa organização facilita o carregamento supervisionado dos dados, associando automaticamente cada amostra ao rótulo correspondente.

3.2 Base de Dados

A base principal utilizada foi a MINDS-Libras Rezende [2021], composta por vídeos padronizados de sinais isolados e metadados associados, como sinal e participante. Essa estrutura permite realizar particionamento controlado por classe e por grupo, reduzindo o risco de que execuções de um mesmo participante apareçam simultaneamente nos conjuntos de treinamento e validação.

Além da base principal, foi construído um conjunto complementar com sinais do alfabeto manual, pronomes e expressões frequentes, visando ampliar o vocabulário avaliado e incluir gestos curtos, discretos e de uso recorrente. Para esse conjunto adicional, voluntários gravaram vídeos seguindo um padrão visual semelhante ao da base MINDS-Libras, com sinais isolados, enquadramento frontal e foco nas mãos durante a execução do gesto. As gravações foram realizadas com diferentes dispositivos, incluindo iPhone 12, iPhone 16, iPad 10 e Lenovo IdeaPad 3, introduzindo variações moderadas de câmera e captura. Em seguida, os vídeos foram processados pelo mesmo procedimento de extração de *landmarks*, convertendo cada execução em uma matriz temporal de coordenadas das mãos. A Tabela 1 resume os sinais utilizados, indicando origem, quantidade de mãos envolvidas e natureza do gesto, classificada como sinal estático ou sinal com movimento.

Tabela 1: Sinais e gestos utilizados, origem, quantidade de mãos e natureza.

Item	Origem	Mãos	Nat.
Acontecer	MINDS	Duas	Mov.
Aluno	MINDS	Uma	Est.
Amarelo	MINDS	Uma	Mov.

Continued on next page

Tabela 1: Sinais e gestos utilizados, origem, quantidade de mãos e natureza. (Continued)

Item	Origem	Mãos	Nat.
América	MINDS	Duas	Est.
Aproveitar	MINDS	Uma	Mov.
Bala	MINDS	Uma	Mov.
Banco	MINDS	Uma	Mov.
Banheiro	MINDS	Duas	Mov.
Barulho	MINDS	Uma	Mov.
Cinco	MINDS	Uma	Est.
Conhecer	MINDS	Uma	Mov.
Espelho	MINDS	Uma	Mov.
Esquina	MINDS	Duas	Mov.
Filho	MINDS	Uma	Mov.
Maçã	MINDS	Uma	Mov.
Medo	MINDS	Uma	Mov.
Ruim	MINDS	Uma	Mov.
Sapo	MINDS	Duas	Mov.
Vacina	MINDS	Uma	Mov.
Vontade	MINDS	Uma	Mov.
a	Adic.	Uma	Est.
ajuda	Adic.	Duas	Mov.
b	Adic.	Uma	Est.
c	Adic.	Uma	Est.
d	Adic.	Uma	Est.
e	Adic.	Uma	Est.
eu	Adic.	Uma	Est.
f	Adic.	Uma	Est.
g	Adic.	Uma	Est.
h	Adic.	Uma	Est.
i	Adic.	Uma	Est.
j	Adic.	Uma	Mov.
k	Adic.	Uma	Est.
l	Adic.	Uma	Est.
m	Adic.	Uma	Est.
n	Adic.	Uma	Est.
não	Adic.	Uma	Mov.
o	Adic.	Uma	Est.
p	Adic.	Uma	Est.
por favor	Adic.	Duas	Mov.
q	Adic.	Uma	Est.
qual	Adic.	Uma	Mov.
quer	Adic.	Uma	Mov.
r	Adic.	Uma	Est.
s	Adic.	Uma	Est.
sim	Adic.	Uma	Mov.
t	Adic.	Uma	Est.
tudo bem	Adic.	Duas	Mov.
u	Adic.	Uma	Est.
v	Adic.	Uma	Est.
você	Adic.	Uma	Mov.
w	Adic.	Uma	Mov.

Continued on next page

Tabela 1: Sinais e gestos utilizados, origem, quantidade de mãos e natureza. (Continued)

Item	Origem	Mãos	Nat.
x	Adic.	Uma	Mov.
y	Adic.	Uma	Mov.
z	Adic.	Uma	Mov.

Abreviações: Nat. = natureza do sinal; Mov. = sinal com movimento; Est. = sinal estático; Adic. = sinal adicionado ao conjunto original.

3.3 Normalização e Aumento de Dados

As sequências foram padronizadas por *z-score* usando estatísticas calculadas apenas no conjunto de treinamento:

$$\hat{x}_{i,d} = \frac{x_{i,d} - \mu_d}{\sigma_d}. \tag{4}$$

Nessa equação, $x_{i,d}$ é a característica d da observação i, μ_d e σ_d são a média e o desvio padrão dessa dimensão no treino, e $\hat{x}_{i,d}$ é o valor normalizado.

Para aumentar a robustez a pequenas variações de detecção, aplicou-se ruído gaussiano aos pontos das mãos:

$$\mathbf{p}'_{t,j} = \mathbf{p}_{t,j} + \boldsymbol{\epsilon}_{t,j}, \quad \boldsymbol{\epsilon}_{t,j} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{aug} I), \quad \sigma_{aug} = 0{,}01. \tag{5}$$

Nessa formulação, $\mathbf{p}_{t,j} = (x_{t,j}, y_{t,j}, z_{t,j})$ é o *landmark* j no quadro t, $\mathbf{p}'_{t,j}$ é o ponto aumentado, $\boldsymbol{\epsilon}_{t,j}$ é o vetor de ruído e $\sim \mathcal{N}$ indica amostragem de uma distribuição normal multivariada. O termo σ^2_{aug} representa a variância do ruído em cada coordenada, enquanto I é a matriz identidade.

O espelhamento horizontal, usado para simular variações entre mão direita e esquerda, foi definido por:

$$\mathcal{M}(x_{t,j}, y_{t,j}, z_{t,j}) = (1 - x_{t,j}, y_{t,j}, z_{t,j}), \tag{6}$$

em que $\mathcal{M}$ é a transformação de espelhamento e $(x_{t,j}, y_{t,j}, z_{t,j})$ são as coordenadas normalizadas do ponto. Após centralização no punho, essa operação equivale a inverter o eixo horizontal relativo, $x \leftarrow -x$; em sinais de uma mão, também se troca o bloco direito/esquerdo para preservar a entrada $[\mathrm{D} \,\|\, \mathrm{E}]$.

3.4 Arquitetura *BiLSTM* com Atenção

O modelo proposto utiliza uma arquitetura *BiLSTM* com mecanismo de atenção para representar a evolução temporal dos sinais. A *LSTM* é uma rede recorrente projetada para lidar com dependências temporais, utilizando uma memória interna controlada por portões. Em cada instante t, a célula recebe a entrada x_t, o estado oculto anterior h_{t-1} e o estado de célula anterior c_{t-1}. Seu funcionamento pode ser resumido por:

$$\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
h_t &= o_t \odot \tanh(c_t).
\end{aligned} \tag{7}$$

Nessas equações, t é o índice temporal da sequência; x_t é o vetor de entrada no instante t; h_{t-1} e h_t são, respectivamente, os estados ocultos anterior e atual; c_{t-1} e c_t são os estados de célula anterior e atual; i_t, f_t e o_t representam os portões de entrada, esquecimento e saída; e $\tilde{c}_t$ é o candidato a novo conteúdo da célula. As matrizes W_i, W_f, W_o e W_c ponderam a entrada atual, enquanto U_i, U_f, U_o e U_c ponderam o estado oculto anterior. Os termos b_i, b_f, b_o e b_c são vieses treináveis; σ é a função sigmoide; $\tanh$ é a tangente hiperbólica; e $\odot$ indica multiplicação elemento a elemento. O portão de entrada controla a incorporação de novas informações à memória; o portão de esquecimento regula quais informações anteriores são preservadas ou descartadas; e o portão de saída determina quanto do estado de célula será exposto como estado oculto. Dessa forma, a *LSTM* consegue preservar informações relevantes ao longo da sequência, reduzindo limitações de redes recorrentes simples em sequências temporais.

Diferentemente de uma *LSTM* unidirecional, que processa a sequência apenas do primeiro ao último quadro, a *BiLSTM* analisa o movimento nos dois sentidos temporais. Assim, para cada instante t, a representação final combina informações do passado e do futuro da sequência:

$$h_t^{Bi} = [h_t^{\rightarrow}; h_t^{\leftarrow}], \tag{8}$$

em que h_t^{Bi} é a representação bidirecional no instante t, $h_t^{\rightarrow}$ é o estado oculto produzido pela *LSTM* no sentido direto, $h_t^{\leftarrow}$ é o estado produzido pela *LSTM* no sentido reverso. Essa característica é importante para o reconhecimento de sinais em Libras, pois a classe de um gesto não depende apenas da configuração da mão em um quadro isolado, mas da trajetória completa do movimento. Por exemplo, sinais como "I" e "J" podem apresentar configurações iniciais semelhantes, mas se diferenciam pela dinâmica temporal: o movimento executado, a transição entre os quadros e a finalização do gesto. Dessa forma, a modelagem bidirecional permite que cada instante da sequência seja interpretado considerando tanto os quadros anteriores quanto os posteriores, favorecendo a distinção entre sinais visualmente parecidos em determinados momentos da execução.

Na configuração final, denominada neste trabalho como *BiLSTM* com atenção, a entrada do modelo é uma sequência com $T = 32$ quadros e $F = 126$ atributos por quadro. Antes da classificação, os *landmarks* são representados no formato centrado no punho (*wrist-centered*), reduzindo a dependência da posição absoluta da mão na imagem e destacando a configuração relativa dos dedos e o movimento ao longo do tempo.

A arquitetura final utiliza duas camadas *BiLSTM* empilhadas, com 256 e 192 unidades, respectivamente. Após cada bloco recorrente, aplica-se normalização por camada, com o objetivo de estabilizar as ativações internas durante o treinamento. Essa combinação permite que as primeiras camadas capturem padrões temporais mais amplos, enquanto a segunda camada refina movimentos sutis e diferenças entre sinais visualmente próximos.

Para complementar a representação recorrente, foi utilizado um mecanismo de autoatenção multi-cabeças. Essa etapa permite que o modelo atribua maior importância aos quadros mais informativos da sequência. A atenção é calcu-

lada por:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \quad (9)$$

Nessa equação, Q, K e V representam, respectivamente, as matrizes de consultas (*queries*), chaves (*keys*) e valores (*values*) obtidas a partir da sequência de estados da ***BiLSTM***; $K^\top$ é a transposta da matriz de chaves; d_k é a dimensão das chaves; $\sqrt{d_k}$ é usado para escalar os produtos internos; e a função softmax converte os escores de atenção em pesos normalizados. A melhor configuração emprega atenção multi-cabeças com 8 cabeças e dimensão de chave $d_k = 64$, sem conexão residual. Nos experimentos de ablação, essa escolha apresentou melhor equilíbrio entre capacidade de representação e estabilidade. Em seguida, utiliza-se *Global Max Pooling* para converter a sequência temporal em um vetor fixo, preservando os quadros mais discriminativos do gesto. O vetor resultante passa por uma camada densa compacta com 64 unidades, sem *dropout* adicional, e a camada final utiliza *softmax* para estimar a probabilidade de cada uma das $K = 55$ classes:

$$p(y = k|x) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}. \quad (10)$$

Nessa formulação, $p(y = k|x)$ representa a probabilidade estimada de a entrada x pertencer à classe k; y é o rótulo de classe; z_k é o logito produzido pelo modelo para a classe k; z_j é o logito associado à classe genérica j; e K é o número total de classes.

A configuração final do modelo foi definida a partir dos estudos de ablação, que avaliaram diferentes combinações de profundidade recorrente, número de unidades, mecanismos de atenção, normalização, conexões residuais e regularização na cabeça de classificação. A arquitetura selecionada busca combinar a capacidade das camadas ***BiLSTM*** de modelar dependências temporais bidirecionais com o mecanismo de atenção multi-cabeças, responsável por destacar padrões relevantes ao longo da sequência de *landmarks*. Após a extração temporal, a agregação por máximo global resume as ativações mais discriminativas antes da etapa final de classificação por *softmax*.

De modo geral, os resultados dos estudos de ablação indicaram melhor desempenho para uma arquitetura recorrente mais profunda, com normalização por camada, atenção multi-cabeças e sem conexão residual ou *dropout* na cabeça de classificação. Os principais componentes e hiperparâmetros do modelo final são apresentados na Tabela 2.

Tabela 2: Configuração final do melhor modelo ***BiLSTM*** com atenção.

Bloco	**Configuração**
Entrada	Sequência com $T = 32$ quadros e $F = 126$ características
Características	2 mãos $\times$ 21 *landmarks* $\times$ 3 coordenadas (x, y, z)
Normalização	*Wrist-centered*: subtração do *landmark* 0 de cada mão
Camada recorrente 1	***BiLSTM*** com 256 unidades
Normalização 1	*LayerNormalization*

Continued on next page

Tabela 2: Configuração final do melhor modelo ***BiLSTM*** com atenção. (Continued)

Bloco	**Configuração**
Camada recorrente 2	***BiLSTM*** com 192 unidades
Normalização 2	*LayerNormalization*
Atenção	*MultiHeadAttention* com 8 cabeças e `key_dim = 64`
Conexão residual	Não utilizada
Pooling	*GlobalMaxPooling1D*
Camada densa	*Dense* com 64 neurônios
Dropout	0,0
Saída	*softmax* com 55 classes
Aumento de dados	*Jitter* gaussiano $\sigma = 0{,}01$; espelh. horiz. D/E; rotação 2D $\pm 8°$; escala 0,9–1,1×; *temporal dropout* com $p = 0{,}05$; *time masking* de 10% dos passos temporais
Otimizador	*AdamW*
Taxa de aprendizado	3×10^{-4}
Weight decay	1×10^{-4}
Label smoothing	$\epsilon = 0{,}10$
Pesos de classe	Não utilizados
EarlyStopping	Monitoramento de `val_loss` com `patience = 15`

3.5 Treinamento e Validação

O treinamento foi realizado como um problema de classificação multiclasse, no qual cada sequência de entrada $X \in \mathbb{R}^{T\times 126}$ está associada a um rótulo y pertencente a uma das $K = 55$ classes. O particionamento dos dados foi feito de forma estratificada por classe, preservando a proporção dos sinais nos conjuntos de treinamento e validação. Além disso, adotou-se isolamento por grupo (*signer*/sessão), evitando que amostras de um mesmo participante ou sessão aparecessem simultaneamente nos dois conjuntos. Essa estratégia reduz o risco de vazamento de informação e fornece uma estimativa mais realista da capacidade de generalização do modelo. A validação foi fixada em 20% do conjunto total.

Antes do treinamento, todas as sequências foram ajustadas para um comprimento temporal fixo de $T = 32$ quadros por meio de *padding* ou *cropping*. Em seguida, as entradas foram normalizadas com estatísticas calculadas exclusivamente no conjunto de treinamento. Dessa forma, a mesma média e o mesmo desvio padrão foram aplicados às amostras de validação, evitando que informações do conjunto de validação influenciassem o processo de treinamento.

A função de perda utilizada foi a entropia cruzada categórica com *label smoothing*. Para uma amostra com vetor de probabilidades previstas $\hat{y}$ e rótulo verdadeiro suavizado y^{LS}, a perda é dada por:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k^{LS} \log(\hat{y}_k). \quad (11)$$

Nessa equação, $\mathcal{L}$ representa o valor da função de perda para a amostra avaliada, k é o índice da classe, K é o número

total de classes, y_k^{LS} é o valor do rótulo verdadeiro suavizado para a classe k, $\hat{y}_k$ é a probabilidade prevista pelo modelo para essa classe.

O *label smoothing* substitui o alvo *one-hot* por uma distribuição suavizada:

$$y_k^{LS} = (1-\alpha)y_k + \frac{\alpha}{K}, \tag{12}$$

em que y_k^{LS} é o rótulo suavizado da classe k, y_k é o rótulo original codificado em formato *one-hot*, α é o fator de suavização e K é o número de classes. Na melhor configuração, utilizou-se $\alpha = 0{,}10$ e não foram aplicados pesos de classe. Essa técnica reduz a confiança excessiva do modelo nas classes corretas durante o treinamento, favorecendo melhor generalização.

A otimização dos parâmetros foi realizada com *AdamW*, uma variação do *Adam* que desacopla o decaimento de pesos da atualização adaptativa dos gradientes. O *Adam* estima médias móveis do primeiro e segundo momentos do gradiente:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2, \tag{13}$$

em que t é o índice da iteração de otimização, g_t é o gradiente no instante t, m_t e v_t representam, respectivamente, estimativas do primeiro e do segundo momento, m_{t-1} e v_{t-1} são essas estimativas na iteração anterior, e β_1 e β_2 são coeficientes de decaimento exponencial usados pelo *Adam*. Após a correção de viés, a atualização com *AdamW* pode ser escrita como:

$$\theta_t = \theta_{t-1} - \eta \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_{t-1} \right), \tag{14}$$

em que θ_t é o vetor de parâmetros após a atualização, θ_{t-1} é o vetor de parâmetros antes da atualização, η é a taxa de aprendizado, $\hat{m}_t$ e $\hat{v}_t$ são as estimativas corrigidas de viés do primeiro e do segundo momento, ϵ evita divisão por zero e λ é o coeficiente de *weight decay*. Neste trabalho, utilizou-se taxa de aprendizado inicial de 3×10^{-4} e *weight decay* de 10^{-4}. O uso do *AdamW* contribui para controlar o crescimento dos pesos e reduzir sobreajuste, mantendo a adaptação individual da taxa de aprendizado para cada parâmetro.

Também foi aplicado *gradient clipping* por norma, limitando a magnitude dos gradientes durante a otimização:

$$g_t \leftarrow g_t \cdot \min\left(1, \frac{c}{\|g_t\|_2}\right), \tag{15}$$

em que g_t é o gradiente antes do ajuste, $\|g_t\|_2$ é sua norma euclidiana, c é o limite máximo permitido para a norma do gradiente. Neste trabalho, utilizou-se $c = 1.0$. Essa técnica é especialmente útil em redes recorrentes, pois reduz instabilidades associadas a gradientes explosivos.

A métrica monitorada durante o treinamento foi a acurácia categórica, definida como:

$$\text{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i), \tag{16}$$

em que Acc é a acurácia categórica, N é o número de amostras avaliadas, i é o índice da amostra, $\hat{y}_i$ é a classe prevista para a amostra i, y_i é a classe verdadeira correspondente e $\mathbb{I}$ é a função indicadora, que assume valor 1 quando a predição está correta e 0 caso contrário. Ao final, o modelo com melhor desempenho de validação foi utilizado nas avaliações reportadas na seção de resultados. As sementes aleatórias, as estatísticas de normalização e as configurações de treinamento foram registradas para garantir reprodutibilidade. O código-fonte do sistema, os *scripts* utilizados nos experimentos, as configurações dos modelos avaliados e os resultados completos dos estudos de ablação foram disponibilizados no repositório público Tulio-CS/CTIC-CSBC-Libras, no *GitHub* Silva [2026].

4 Resultados

4.1 Acurácia e Perda

A acurácia categórica, definida na Equação 16, representa a proporção de amostras corretamente classificadas pelo modelo em relação ao total de amostras avaliadas. Assim, valores mais altos indicam maior capacidade de associar cada sequência de *landmarks* ao sinal correto. Já a perda quantifica o erro de otimização usado para ajustar os parâmetros da rede; neste trabalho, ela corresponde à entropia cruzada categórica com *label smoothing*, cuja formulação foi apresentada anteriormente na Equação 11. Valores menores de perda indicam que as probabilidades previstas pelo modelo estão mais alinhadas aos rótulos esperados.

A Figura 3 apresenta a evolução da acurácia, enquanto a Figura 4 apresenta a evolução da função de perda nos conjuntos de treinamento e validação para o modelo *BiLSTM* com atenção e para a *LSTM* unidirecional. Essa análise permite observar o comportamento de convergência dos modelos ao longo das épocas, além de indicar possíveis sinais de sobreajuste ou instabilidade durante o treinamento.

Ambos os modelos apresentam crescimento rápido da acurácia nas primeiras épocas, indicando que as representações baseadas em *landmarks* fornecem informações discriminativas para o reconhecimento dos sinais. Ainda assim, a *BiLSTM* com atenção alcança patamar superior e mantém maior proximidade entre as curvas de treinamento e validação.

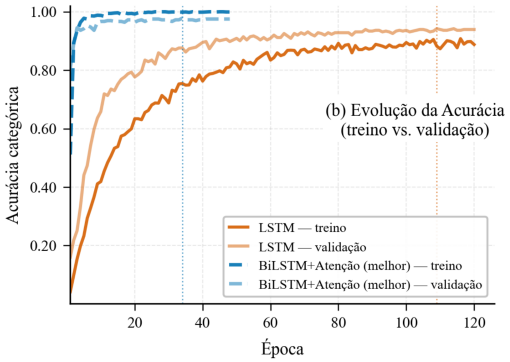


Figura 3. Evolução da acurácia ao longo das épocas para os modelos *BiLSTM* com atenção e *LSTM* unidirecional.

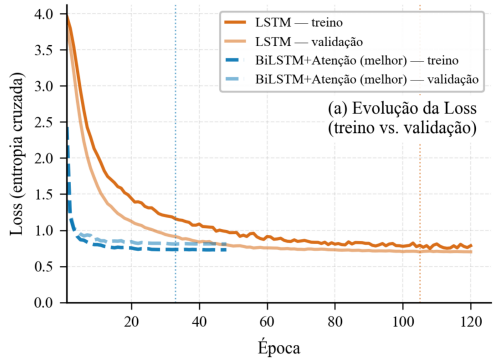


Figura 4. Evolução da perda ao longo das épocas para os modelos *BiLSTM* com atenção e *LSTM* unidirecional.

As curvas de perda seguem a mesma tendência: há queda acentuada nas primeiras épocas e estabilização progressiva em ambos os modelos, com valores finais menores para a *BiLSTM*. Em conjunto, acurácia e perda indicam convergência estável, sem sinais evidentes de sobreajuste severo, e reforçam a escolha da arquitetura bidirecional para capturar transições e finalizações relevantes dos gestos em Libras.

4.2 Curvas e Métricas Complementares

Além das métricas agregadas, foram analisadas a comparação direta entre métricas macro, histogramas de confiança e a distribuição do *F1-score* por categoria de sinal. Essas análises complementares permitem avaliar não apenas o desempenho médio do modelo, mas também sua estabilidade entre classes e diferentes tipos de gesto.

Antes da comparação gráfica, é importante definir as métricas utilizadas. A acurácia já foi definida anteriormente pela Equação 16 e oferece uma visão geral do desempenho do classificador. A precisão indica, entre as amostras atribuídas a uma classe, quantas realmente pertencem a essa classe; portanto, valores altos de precisão sugerem menor ocorrência de falsos positivos. A revocação, também denominada *recall*, mede, entre as amostras que pertencem de fato a uma classe, quantas foram corretamente recuperadas pelo modelo; assim, valores altos indicam menor ocorrência de falsos negativos. O *F1-score* resume o equilíbrio entre precisão e revocação por meio da média harmônica, sendo útil quando se deseja avaliar simultaneamente a capacidade do modelo de evitar falsos positivos e falsos negativos, especialmente em cenários com possível desbalanceamento entre classes.

Considerando verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN), as demais métricas podem ser expressas como:

$$\begin{aligned} \text{Precisao} &= \frac{VP}{VP+FP}, \\ \text{Revocacao} &= \frac{VP}{VP+FN}, \\ \text{F1} &= 2 \cdot \frac{\text{Precisao} \cdot \text{Revocacao}}{\text{Precisao} + \text{Revocacao}}. \end{aligned} \tag{17}$$

No caso multiclasse, essas quantidades são calculadas por classe, tratando a classe analisada como positiva e as demais como negativas. As métricas macro reportadas neste trabalho correspondem à média dessas medidas entre as classes, atribuindo o mesmo peso a cada sinal independentemente de sua frequência no conjunto avaliado.

A Figura 5 resume a comparação entre *BiLSTM* com atenção e *LSTM* unidirecional em acurácia, precisão, revocação e *F1-score*. Observa-se vantagem consistente do modelo bidirecional com atenção em todas as métricas avaliadas.

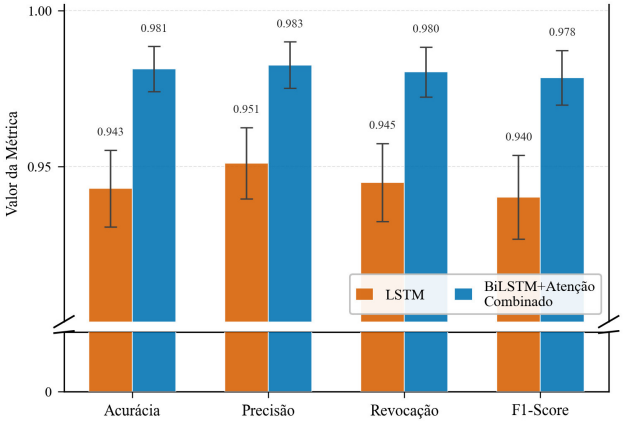


Figura 5. Comparação de acurácia, precisão, revocação e *F1-score* entre *BiLSTM* com atenção e *LSTM* unidirecional.

A Figura 6 apresenta a distribuição da confiança máxima atribuída pelo *softmax* para predições corretas e incorretas no modelo *LSTM* unidirecional, enquanto a Figura 7 apresenta a mesma análise para o modelo *BiLSTM* com atenção. Essa análise permite comparar não apenas a taxa de acerto dos modelos, mas também o grau de segurança associado às suas predições.

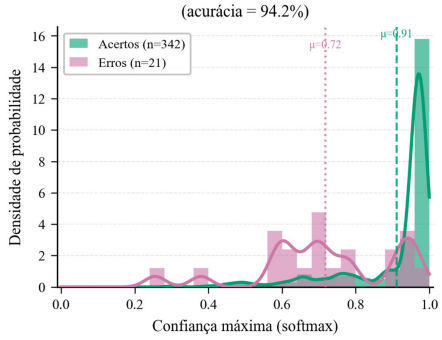


Figura 6. Histograma de confiança para acertos e erros no modelo *LSTM* unidirecional.

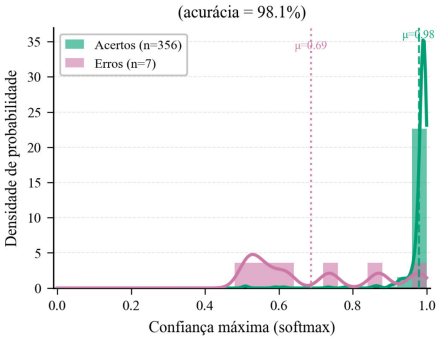


Figura 7. Histograma de confiança para acertos e erros no modelo *BiLSTM* com atenção.

Observa-se que o modelo *LSTM* apresenta acertos com confiança média de aproximadamente 0,91 e erros com média próxima de 0,72, além de uma distribuição mais dispersa das predições corretas. Já o *BiLSTM* com atenção apresenta acertos com confiança média de aproximadamente 0,98 e erros com média próxima de 0,69, além de menor quantidade de erros no conjunto avaliado. Essa comparação sugere que o modelo bidirecional com atenção não apenas melhora a taxa de acerto, mas também produz predições corretas com maior confiança média, enquanto seus erros ocorrem, em média, com confiança ligeiramente menor do que no modelo *LSTM*. Ainda assim, a presença de erros em faixas intermediárias e altas de confiança indica que a saída do *softmax* deve ser interpretada como um indicador interno de confiança do modelo, e não necessariamente como uma probabilidade calibrada.

Para investigar a consistência do modelo entre diferentes tipos de gestos, analisou-se também a distribuição do *F1-score* por quantidade de mãos e por natureza do sinal. A Figura 8 indica que o modelo apresenta *F1-score* elevado em todas as categorias avaliadas. Observa-se uma leve tendência de melhor desempenho em sinais realizados com uma mão em comparação aos sinais com duas mãos, assim como em sinais estáticos em relação aos sinais dinâmicos. Esse comportamento é esperado, pois sinais com uma única mão e menor variação temporal tendem a apresentar menor complexidade espacial e temporal. No entanto, a diferença entre as distribuições não é acentuada, indicando que o modelo também mantém desempenho elevado em sinais com duas mãos e sinais dinâmicos. De forma geral, os resultados sugerem boa estabilidade entre categorias distintas, com reduções pontuais associadas a sinais de maior semelhança visual ou maior variação de execução.

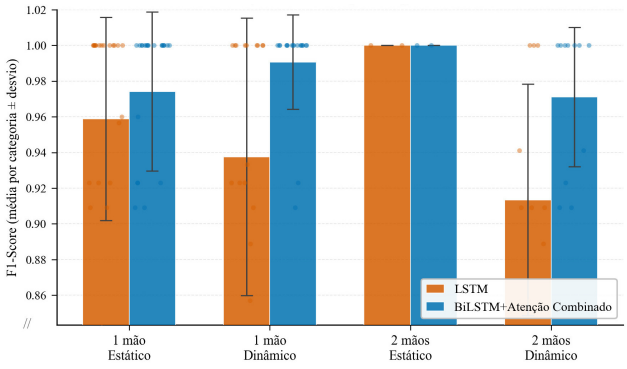


Figura 8. Distribuição do *F1-score* por categoria de sinal: uma mão estático, uma mão dinâmico, duas mãos estático e duas mãos dinâmico.

4.3 Comparação com trabalhos anteriores

A Tabela 3 compara o melhor modelo treinado e avaliado no MINDS-Libras com trabalhos prévios.

Tabela 3. Resultados no conjunto de dados MINDS-Libras e comparação com trabalhos prévios.

Modelo / trabalho	Entrada	*Acc.* (%)	*F1*	Prec.	Rec.
BiLSTM + atenção	*Landmarks* das mãos	97,8	0,977	0,978	0,978
Alves et al. Alves *et al*. [2024]	Mãos, corpo e face	93,0	0,930	0,940	0,930
De Castro Rezende [2021]	Vídeo *RGB*	91,0	0,900	–	–
Passos et al. Stefano *et al*. [2021]	Vídeo *RGB*	85,0	–	–	–

A comparação direta entre a acurácia deste sistema e os resultados de trabalhos anteriores deve ser interpretada com cautela, pois os estudos externos podem diferir quanto à divisão dos dados, participantes, classes, condições de captura, pré-processamento e critério de validação. Assim, a Tabela 3 contextualiza o desempenho, mas não estabelece uma hierarquia definitiva: o resultado sugere que o uso apenas de *landmarks* das mãos pode ser competitivo frente a entradas com mãos, corpo e face Alves *et al*. [2024] ou vídeo *RGB* Rezende [2021]; Stefano *et al*. [2021], mantendo menor dimensionalidade. A Tabela 4 compara o modelo treinado apenas com o MINDS-Libras e a versão treinada com o conjunto combinado, formado por MINDS-Libras e sinais adicionados.

Tabela 4. Comparação entre modelos treinados no MINDS-Libras e no conjunto combinado.

Modelo	Dados	Cls.	*Acc.* (%)	*F1*	Prec.	Rec.
BiLSTM + atenção	MINDS	20	97,8	0,977	0,978	0,978
BiLSTM + atenção	MINDS + adic.	55	98,7	0,986	0,987	0,987

4.4 Comparativo direto: *BiLSTM* vs. *LSTM* unidirecional

A Tabela 5 resume as métricas agregadas para a comparação entre a *LSTM* unidirecional e o *BiLSTM* com atenção em sua melhor configuração. Essa configuração corresponde ao modelo descrito na metodologia, com duas camadas *BiLSTM*, normalização por camada, atenção multi-cabeças e agregação por máximo global. O modelo apresenta maior desempenho global e converge em menos épocas, indicando vantagem da bidirecionalidade e do mecanismo de atenção em padrões temporais sutis.

Tabela 5. Resumo comparativo entre *LSTM* e *BiLSTM* com atenção.

Modelo	*Acc.* (%)	Prec. (%)	Rec. (%)	*F1* (%)	Ép.
LSTM	93,9	94,7	94,1	94,1	241
BiLSTM + atenção	97,3	97,7	97,2	97,3	97

O *BiLSTM* com atenção convergiu em apenas 97 épocas, contra 241 épocas da *LSTM* unidirecional, utilizando *EarlyStopping* com paciência=15. O *EarlyStopping* é uma estratégia de parada antecipada que monitora uma métrica de

validação durante o treinamento, neste caso a perda de validação, e interrompe o processo quando não há melhora após um número definido de épocas consecutivas. Assim, a paciência igual a 15 significa que o treinamento só é encerrado depois de 15 épocas seguidas sem redução da perda de validação, evitando interrupções prematuras por pequenas oscilações e, ao mesmo tempo, reduzindo treinamento desnecessário e risco de sobreajuste.

4.5 Estudo de Ablação

Para isolar o efeito das principais escolhas de arquitetura e regularização, foram conduzidos estudos de ablação sobre o modelo temporal baseado em *landmarks*. No artigo, são apresentados apenas os resultados mais relevantes para a análise comparativa; o material complementar disponível no repositório público Tulio-CS/CTIC-CSBC-Libras contém as tabelas completas, as variações arquiteturais avaliadas e métricas adicionais obtidas durante os experimentos Silva [2026]. A Tabela 6 resume as variações arquiteturais avaliadas.

Tabela 6. Ablação arquitetural do modelo *BiLSTM* com atenção.

Aspecto	Variação avaliada	Melhor escolha	Ganho (Acc)
Arquit.	*LSTM*; *BiLSTM*; *BiLSTM* + atenção	*BiLSTM* + atenção	+0,034 vs. *LSTM*
Profund.	1, 2 ou 3 camadas *BiLSTM*	2 camadas	+ 0,011 vs. 1
Unidades	Pequeno; médio; grande	Grande [256, 192]	+0,008
Cab. atenção	1, 2, 4 ou 8 cabeças	8 cabeças	+0,008 vs. 4
Dim. atenção	16, 32 ou 64	64	+0,005 vs. 32
Camada densa	Pequena [64]; média [192, 128]; grande [256, 128]	Pequena [64]	+0,008
Agreg. temporal	Média; máximo; último estado	Máximo	marginal
Norm.	Com ou sem *LayerNorm*	Com *LayerNorm*	+0,008
Residual	Com ou sem conexão residual	Sem residual	+0,008

A Tabela 7 apresenta os resultados de regularização, incluindo variações de *dropout*, suavização de rótulos e uso de pesos de classe.

Tabela 7. Ablação de regularização.

Aspecto	Variação avaliada	Melhor escolha	Ganho (Acc)
Dropout	0,0; 0,2; 0,4; 0,6	0,0	+0,022
Dropout recorr.	0,00; 0,10; 0,15; 0,30	0,15	–
Suavização de rótulos	0,00; 0,05; 0,10; 0,20	0,10	+0,006
Pesos de classe	Com ou sem pesos	Sem pesos	marginal

Na ablação de regularização, a remoção de *dropout* apresentou o maior ganho individual de acurácia (+0,0220), sugerindo que o conjunto de treinamento e a arquitetura já forneciam regularização suficiente para o cenário avaliado. A suavização de rótulos com valor 0,10 também contribuiu positivamente, enquanto o uso de pesos de classe teve efeito marginal.

Por fim, a análise de sinais confundíveis avaliou as confusões mais frequentes no conjunto de teste. O par i $\leftrightarrow$ j foi o mais confundido: a letra "i" é estática, enquanto a letra "j" utiliza a mesma configuração manual inicial, mas adiciona um movimento curvo. O modelo *BiLSTM* com atenção ampliado confundiu j$\rightarrow$i com confiança de 98,41%, indicando que o movimento dinâmico de "j" é especialmente difícil de distinguir no início da sequência. Letras com configurações manuais similares, como h/u e o/c, também apareceram entre as confusões mais frequentes.

4.6 Eficiência Computacional

O *benchmark* de eficiência foi executado em *CPU*, com *batch*=1 e 200 execuções por modelo. A Tabela 8 compara a *LSTM* unidirecional e o *BiLSTM* com atenção em termos de tamanho, custo computacional e latência.

Tabela 8. *Benchmark* de eficiência em *CPU*.

Métrica	*LSTM*	*BiLSTM* + atenção
Parâmetros	195.383	1.041.655
FP32	0,76 MB	3,97 MB
GFLOPs/seq.	0,0058	0,0309
Latência média	64,26 ms	304,32 ms
p95	73,03 ms	320,85 ms
Pipeline web	62,90 ms	317,97 ms

A Tabela 9 detalha o tempo médio e o percentil 95 das principais etapas do *pipeline* combinado.

Tabela 9. *Breakdown* do *pipeline* combinado em *CPU* (ms).

Etapa	Média	p95
Feature transform	0,038	0,052
Normalização	0,040	0,055
Inferência	321,35	330,71
EMA	0,023	0,032

Embora o *BiLSTM* com atenção seja mais custoso, com 1.041.655 parâmetros, 3,97 MB em *FP32* e 0,0309 *GFLOPs* por sequência, o *pipeline web* manteve latência média de 317,97 ms, indicando viabilidade de uso interativo sem *GPU* dedicada.

5 Sistema *Web* e Inferência em Tempo Real

Além da avaliação experimental, o modelo treinado foi integrado a uma aplicação *web* funcional, disponível em `https://libras.tremdigital.com.br`. O objetivo dessa etapa foi validar o uso prático do classificador em um fluxo próximo ao cenário real, no qual o usuário executa um sinal diante da câmera e recebe a predição diretamente no navegador.

A Figura 9 resume esse funcionamento, mostrando o caminho desde a captura do gesto pela câmera até a resposta final apresentada ao usuário.

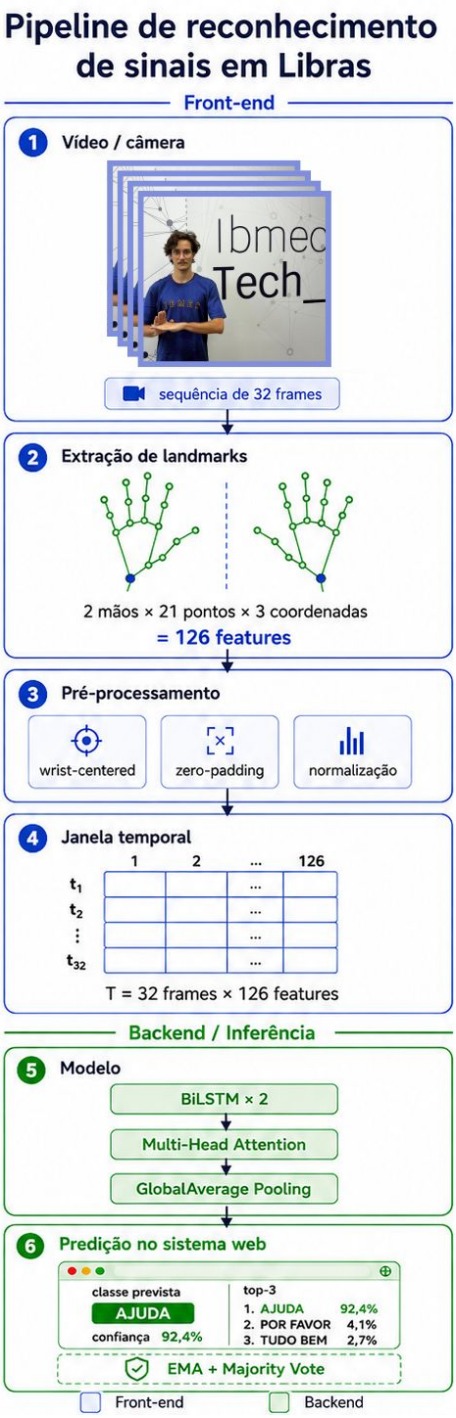


Figura 9. Fluxo de inferência do sistema *web*. O gesto é capturado no navegador, convertido em pontos das mãos, organizado em uma sequência de 32 quadros e enviado ao modelo para identificação do sinal.

Na aplicação, o *front-end* corresponde à parte com a qual o usuário interage diretamente. Ele é responsável por abrir a página no navegador, solicitar acesso à câmera, mostrar o vídeo em tempo real e desenhar sobre a imagem os pontos detectados nas mãos. Essa etapa também organiza os pontos capturados para que eles possam ser enviados ao restante do sistema. Em outras palavras, o *front-end* funciona como a interface visual e como a primeira etapa de preparação do gesto executado pelo usuário.

O *back-end*, por sua vez, corresponde à parte executada no servidor. Ele recebe os pontos das mãos enviados pelo navegador, aplica os mesmos ajustes usados durante o treinamento e executa o modelo treinado para identificar qual sinal foi realizado. Depois disso, o servidor devolve ao navegador a classe prevista, o nível de confiança e as opções mais prováveis. Essa divisão mantém a interação simples para o usuário: o navegador cuida da captura e da visualização, enquanto o servidor realiza a classificação com o modelo neural.

O fluxo de inferência em produção segue quatro etapas principais: captura do vídeo no navegador, extração dos *landmarks* das mãos, envio das características ao servidor e classificação temporal pelo modelo. Embora o sistema utilize componentes específicos para comunicação e implantação, a lógica principal pode ser entendida como uma troca contínua: o navegador envia uma representação compacta do gesto e o servidor retorna a interpretação mais provável.

Antes da predição, os pontos extraídos no navegador passam pelas mesmas transformações aplicadas aos dados de treinamento. Os *landmarks* são organizados em uma sequência temporal, centralizados em relação ao punho (*wrist-centered*) e ajustados ao formato esperado pelo modelo. O sistema acumula 32 quadros consecutivos de *landmarks* antes de realizar uma predição. Após a formação da janela, a sequência é enviada ao modelo *BiLSTM* com atenção, que retorna a distribuição de probabilidade sobre as $K = 55$ classes. Além da predição principal, o servidor também retorna a confiança e as três classes mais prováveis.

Para tornar a saída mais estável em tempo real, foi aplicado pós-processamento sobre as predições sucessivas, combinando média móvel exponencial (*Exponential Moving Average*, *EMA*) e voto majoritário (*Majority Vote*). A *EMA* suaviza oscilações abruptas nas probabilidades previstas, enquanto o voto majoritário reduz trocas rápidas de classe entre janelas consecutivas. A classe final exibida ao usuário é, portanto, obtida a partir da combinação entre a probabilidade atual do modelo e a consistência das predições recentes.

Seja $\mathbf{p}_t \in \mathbb{R}^K$ o vetor de probabilidades retornado pelo modelo na janela temporal t. A suavização por *EMA* atualiza o vetor suavizado $\tilde{\mathbf{p}}_t$ como:

$$\tilde{\mathbf{p}}_t = \alpha \mathbf{p}_t + (1-\alpha)\tilde{\mathbf{p}}_{t-1}, \quad 0 < \alpha \leq 1, \tag{18}$$

em que α controla o peso da predição atual em relação ao histórico: valores maiores tornam a resposta mais sensível à predição recente, enquanto valores menores produzem maior suavização temporal. A classe associada ao vetor suavizado pode ser obtida por:

$$\hat{y}_t^{EMA} = \arg \max_{k \in \{1,\ldots,K\}} \tilde{p}_{t,k}. \tag{19}$$

Em seguida, considerando uma janela com as W predições suavizadas mais recentes, o voto majoritário seleciona a classe mais frequente:

$$\hat{y}_t^{MV} = \arg \max_{k \in \{1,\ldots,K\}} \sum_{i=0}^{W-1} \mathbb{I}\left(\hat{y}_{t-i}^{EMA} = k\right), \tag{20}$$

em que $\mathbb{I}(\cdot)$ é a função indicadora, que assume valor 1 quando a condição é verdadeira e 0 caso contrário. Assim, a *EMA* reduz oscilações nas probabilidades, enquanto o voto

majoritário reforça a consistência da classe exibida ao longo de janelas consecutivas.

A adoção de *landmarks*, em vez dos quadros *RGB* brutos, reduz substancialmente a dimensionalidade da entrada e o custo computacional da comunicação. Na prática, o celular executa a detecção dos pontos no navegador e envia apenas 126 valores por quadro ao servidor via *WebSocket*. O servidor realiza a inferência com o modelo *BiLSTM* em *CPU* e retorna a predição, a confiança e o *top-3* à interface. Esse fluxo permite atualizar a interface com latência interativa, demonstrando que o modelo pode ser utilizado em um sistema de reconhecimento de sinais isolados de Libras em tempo quase real.

6 Discussão

Os resultados obtidos indicam que a combinação de *landmarks* das mãos com uma arquitetura *BiLSTM* com atenção é uma alternativa eficiente para o reconhecimento de sinais isolados de Libras. O uso do *MediaPipe* permitiu representar cada quadro por 126 atributos, correspondentes às coordenadas tridimensionais de até duas mãos, reduzindo a dimensionalidade da entrada em comparação com abordagens baseadas diretamente em vídeo *RGB*. Essa escolha contribui para diminuir o custo computacional e facilita a execução em tempo real.

Em comparação com abordagens que utilizam imagens ou vídeos brutos, a representação por *landmarks* tende a ser menos dependente de características visuais como fundo, iluminação, cor de pele e textura da imagem. Isso é especialmente relevante para aplicações práticas, pois o sistema deve operar em condições variadas de captura. Trabalhos anteriores, como Sarmento e Ponti de Avellar Sarmento and Ponti [2023], também observaram que representações baseadas em *landmarks* podem apresentar maior robustez do que entradas *RGB* em cenários de variação entre bases.

No contexto do MINDS-Libras, os resultados devem ser interpretados pelo equilíbrio entre desempenho e simplicidade da entrada. O modelo alcançou 97,75% de acurácia e *F1-score* de 0,9774 nesse cenário e 98,67% de acurácia e *F1-score* de 0,9864 no conjunto combinado, sugerindo que, para os sinais isolados avaliados, a dinâmica temporal preservada pelos *landmarks* das mãos fornece informação discriminativa sem exigir o processamento completo dos quadros de vídeo.

A implantação em uma aplicação *web* complementa essa análise ao avaliar a solução em um fluxo mais próximo do uso real. Nesse caso, a contribuição não está apenas na métrica final do classificador, mas na integração entre captura pela câmera, extração dos pontos das mãos, envio das características e retorno da predição ao usuário. Essa validação prática mostra que a abordagem pode ser incorporada a uma interface acessível, mantendo o foco em sinais isolados e em uma arquitetura de baixo custo computacional.

Apesar dos resultados promissores, algumas limitações devem ser consideradas. O sistema foi avaliado em sinais isolados, não contemplando ainda frases contínuas, coarticulação entre sinais ou elementos não manuais, como expressões faciais e movimentos corporais. Além disso, embora o conjunto complementar tenha introduzido variações de dispositivos e participantes, avaliações futuras devem incluir cenários mais diversos, com diferentes condições de iluminação, enquadramento e perfis de usuários.

7 Conclusão

Este trabalho apresentou o desenvolvimento, treinamento e implantação *web* de um sistema para reconhecimento de sinais isolados de Libras. A solução utiliza *landmarks* das mãos extraídos pelo *MediaPipe* como entrada e uma arquitetura *BiLSTM* com atenção para modelar a dinâmica temporal dos gestos. Com essa abordagem, foi possível reduzir a complexidade da entrada, evitando o processamento direto dos quadros *RGB*, e manter um modelo compacto, com aproximadamente 1,04 milhão de parâmetros.

O principal resultado experimental foi obtido no conjunto combinado, no qual o modelo alcançou acurácia de 98,67% e *F1-score* de 0,9864 para 55 sinais. Considerando também o tamanho reduzido da entrada e a quantidade controlada de parâmetros, esse desempenho indica que a estratégia baseada em *landmarks* oferece um caminho promissor para sistemas leves de reconhecimento de sinais isolados.

A integração *web* fecha o ciclo entre pesquisa e aplicação, pois transforma o modelo treinado em um sistema utilizável a partir da câmera do usuário. Dessa forma, a contribuição do trabalho não se limita ao treinamento do classificador, mas inclui um fluxo completo de captura, preparação dos dados, inferência e apresentação da resposta, com potencial de apoio a ferramentas de acessibilidade, ensino de Libras e comunicação assistiva.

Como trabalhos futuros, pretende-se extrapolar o reconhecimento de sinais isolados e avançar para sinais em contexto, considerando sequências contínuas de Libras e relações temporais entre gestos. Essa etapa envolve desafios como segmentação automática, coarticulação, dependência semântica entre sinais e integração de informações não manuais, como expressões faciais e movimentos corporais. Além disso, pretende-se investigar a criação de *embeddings* para representar sinais em um espaço vetorial, permitindo medir similaridades entre gestos e capturar relações visuais, temporais e semânticas. A longo prazo, esses avanços podem apoiar o desenvolvimento de modelos linguísticos específicos para Libras, capazes de interpretar sequências de sinais em contexto e contribuir para sistemas mais completos de tradução, ensino e comunicação assistiva.

Declarações complementares

Agradecimentos

Os autores agradecem ao Centro Universitário Ibmec BH pelo apoio institucional ao desenvolvimento deste trabalho, bem como aos professores e colaboradores que contribuíram com orientações, discussões técnicas e sugestões ao longo da pesquisa. Agradecem, em especial, à Prof.ª Gisele Tessari Santos, coordenadora do curso de Ciência de Dados e Inteligência Artificial, e à Prof.ª Clarissa Ana Zambiasi, coordenadora do curso de Engenharia de Produção, pelo apoio, incentivo e acompanhamento ao longo das diferentes etapas do projeto, bem como pelo estímulo à sua realização no contexto acadêmico e institucional da faculdade. Os autores agradecem também aos participantes envolvidos nas coletas complementares, cuja colaboração foi essencial para a avaliação do sistema proposto.

Financiamento

Este trabalho contou com apoio institucional do Centro Universitário Ibmec BH, incluindo suporte à supervisão do projeto e à viabilização da participação dos autores no evento.

Contribuições dos autores

Túlio Castro contribuiu para a concepção do estudo, implementação do sistema, preparação dos dados, treinamento dos modelos, análise dos resultados e redação do manuscrito. Pedro Calais contribuiu para a orientação, revisão crítica e aprimoramento do manuscrito. Todos os autores leram e aprovaram a versão final.

Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

Disponibilidade de dados e materiais

O código-fonte do sistema, os *scripts* utilizados nos experimentos, as configurações dos modelos avaliados e os resultados completos dos estudos de ablação estão disponíveis no repositório público Tulio-CS/CTIC-CSBC-Libras, no *GitHub* Silva [2026]. No artigo, são apresentados apenas os resultados mais relevantes para a análise comparativa, enquanto o material complementar disponibilizado no repositório contém as tabelas completas, as variações arquiteturais avaliadas e as métricas adicionais obtidas durante os experimentos. Os conjuntos de dados gerados e/ou analisados durante o estudo atual serão disponibilizados mediante solicitação aos autores, respeitando eventuais restrições de privacidade dos participantes.

Outras informações relevantes

As coletas complementares envolveram registros de sinais isolados em Libras para fins exclusivamente acadêmicos e de pesquisa. Todas as etapas relacionadas à coleta e ao uso dos dados foram conduzidas mediante consentimento dos participantes e aprovação ética e institucional, observando princípios de privacidade, confidencialidade e uso restrito das informações no contexto do estudo. Não houve necessidade de identificação pública dos participantes, uma vez que os registros foram utilizados apenas para avaliação e aprimoramento do modelo proposto.

Referências

Alves, C. E. G. R., Boldt, F. d. A., and Paixão, T. M. (2024). Enhancing Brazilian Sign Language recognition through skeleton image representation. arXiv preprint arXiv:2404.19148.

de Avellar Sarmento, A. H. and Ponti, M. A. (2023). A cross-dataset study on the Brazilian Sign Language translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2816–2820. DOI: 10.1109/ICCVW60793.2023.00300.

Desai, A., Berger, L., Minakov, F., Milano, N., Singh, C., Pumphrey, K., Ladner, R. E., Daumé III, H., Lu, A. X., Caselli, N., and Bragg, D. (2023). ASL citizen: A community-sourced dataset for advancing isolated sign language recognition. In *Advances in Neural Information Processing Systems*, volume 36. Versão publicada no NeurIPS 2023 Datasets and Benchmarks Track. DOI: 10.48550/arXiv.2304.05934.

Duarte, A., Palaskar, S., Ventura, L., Ghadiyaram, D., DeHaan, K., Metze, F., Torres, J., and Giró-i Nieto, X. (2021). How2Sign: A large-scale multimodal dataset for continuous American Sign Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2735–2744. DOI: 10.1109/CVPR46437.2021.00276.

Google (2024). MediaPipe: Cross-platform, customizable ML solutions for live and streaming media. Project website. Disponível em: mediapipe.dev. Acesso em: 15 maio 2026.

Instituto Brasileiro de Geografia e Estatística (2019a). Pesquisa nacional de saúde 2019: Tabela 8217 – pessoas com deficiência auditiva, por sexo e situação do domicílio. Sistema IBGE de Recuperação Automática (SIDRA). Disponível em: sidra.ibge.gov.br. Acesso em: 15 maio 2026.

Instituto Brasileiro de Geografia e Estatística (2019b). Pesquisa nacional de saúde 2019: Tabela 8223 – pessoas de cinco anos ou mais de idade que referiram dificuldade permanente para ouvir, por conhecimento da língua brasileira de sinais – libras e grau de dificuldade para ouvir. Sistema IBGE de Recuperação Automática (SIDRA). Disponível em: sidra.ibge.gov.br. Acesso em: 16 maio 2026.

Instituto Brasileiro de Geografia e Estatística (2021). Um em cada quatro idosos tinha algum tipo de deficiência em 2019. Agência IBGE Notícias. Publicado em 26 ago. 2021. Disponível em: agenciadenoticias.ibge.gov.br. Acesso em: 16 maio 2026.

Kumari, D. and Anand, R. S. (2024). Isolated video-based sign language recognition using a hybrid CNN-LSTM framework based on attention mechanism. *Electronics*, 13(7):1229. DOI: 10.3390/electronics13071229.

Li, D., Rodriguez, C., Yu, X., and Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1459–1469. DOI: 10.1109/WACV45572.2020.9093512.

Rezende, T. M. (2021). *Reconhecimento Automático de Sinais da Libras: Desenvolvimento da Base de Dados MINDS-Libras e Modelos de Redes Convolucionais*. Tese (doutorado em engenharia elétrica), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil. Disponível em: hdl.handle.net/1843/39785. Acesso em: 15 maio 2026.

Rezende, T. M., Almeida, S. G. M., and Guimarães, F. G. (2021). Development and validation of a Brazilian Sign Language database for human gesture recognition. *Neural Computing and Applications*, 33(16):10449–10467. DOI: 10.1007/s00521-021-05802-4.

Shi, B., Brentari, D., Shakhnarovich, G., and Livescu, K. (2022). Open-domain sign language translation learned from online video. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6379, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. DOI: 10.18653/v1/2022.emnlp-main.427.

Silva, T. C. (2026). CTIC-CSBC-Libras: Código-fonte, scripts experimentais e resultados complementares. Repositório público no GitHub. Tulio-CS/CTIC-CSBC-Libras. Disponível em: github.com/Tulio-CS/CTIC-CSBC-Libras. Acesso em: 15 maio 2026.

Stefano, G. d. S. L., Passos, W. L., Gois, J. N., Araujo, G. M., and de Lima, A. A. (2021). Um sistema de reconhecimento de sinais em Libras usando CNN e LSTM. In *Anais do XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2021)*, Fortaleza, CE, Brasil. Sociedade Brasileira de Telecomunicações. DOI: 10.14209/sbrt.2021.1570727292.

Vaezi Joze, H. R. and Koller, O. (2019). MS-ASL: A large-scale data set and benchmark for understanding American Sign Language. arXiv preprint arXiv:1812.01053.

World Federation of the Deaf (2026). Our work. Institutional website. Disponível em: wfdeaf.org. Acesso em: 15 maio 2026.

World Health Organization (2026). Deafness and hearing loss. Fact sheet. Disponível em: who.int. Publicado em 3 mar. 2026. Acesso em: 15 maio 2026.

Zhang, Y. and Jiang, X. (2024). Recent advances on deep learning for sign language recognition. *Computer Modeling in Engineering & Sciences*, 139(3):2399–2450. DOI: 10.32604/cmes.2023.045731.