v7. n12.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

LIBRASFLOW: SISTEMA WEB PARA RECONHECIMENTO EM TEMPO REAL DE LETRAS DA LIBRAS COM VISÃO COMPUTACIONAL E APRENDIZADO DE MÁQUINA

LIBRASFLOW: WEB-BASED SYSTEM FOR REAL-TIME RECOGNITION OF LIBRAS LETTERS USING COMPUTER VISION AND MACHINE LEARNING

LIBRASFLOW: SISTEMA WEB PARA EL RECONOCIMIENTO EN TIEMPO REAL DE LETRAS DE LIBRAS MEDIANTE VISIÓN POR COMPUTADORA Y APRENDIZAJE AUTOMÁTICO

Gustavo Alves dos Santos[1], Carlos Henrique Moreira Gonçalves[1], Tardelli Ronan Coelho Stekel[2], Ana Paula Abrantes de Castro e Shiguemori[3]

https://doi.org/10.47820/recima21.v7i12.9762

PUBLICADO: 12/2026

RESUMO

A comunicação entre pessoas surdas e ouvintes ainda enfrenta barreiras de acessibilidade, especialmente na ausência de intérpretes da Língua Brasileira de Sinais (Libras). Este trabalho apresenta o desenvolvimento e a avaliação de um sistema de reconhecimento em tempo real de 21 configurações estáticas do alfabeto manual da Libras, combinando visão computacional, aprendizado de máquina e interface web. A versão com processamento no servidor utiliza *OpenCV, MediaPipe Hand*s, *Random Forest* e *Flask*, enquanto uma versão pública executa a detecção e a classificação diretamente no navegador por meio de uma rede perceptron multicamadas (MLP). Para reduzir oscilações entre letras semelhantes, foi empregado um mecanismo de confirmação temporal. A avaliação inicial revelou vazamento de dados causado por amostras quase duplicadas, motivando a adoção de particionamento por grupos geométricos. Nesse protocolo, o *Random Forest* obteve acurácia média de 97,01% ± 2,59% em dez repetições. Em um teste independente com 3.812 imagens inéditas de 15 classes, a acurácia foi de 71,2%, evidenciando limitações de generalização. Os resultados demonstram a viabilidade do sistema em tempo real, mas indicam a necessidade de ampliar a diversidade dos dados e a robustez do reconhecimento.

PALAVRAS-CHAVE: Língua Brasileira de Sinais. Visão Computacional. Aprendizado de Máquina. Reconhecimento do Alfabeto Manual. *Random Forest*. Acessibilidade Digital.

ABSTRACT

Communication between deaf and hearing people still faces accessibility barriers, especially when Brazilian Sign Language (Libras) interpreters are unavailable. This work presents the development and evaluation of a real-time recognition system for 21 static configurations of the Libras manual alphabet, combining computer vision, machine learning, and a web interface. The server-side version uses OpenCV, MediaPipe Hands, Random Forest, and Flask, while a public version performs detection and classification directly in the browser using a multilayer perceptron (MLP). A temporal confirmation mechanism was employed to reduce oscillations between similar letters. The initial evaluation revealed data leakage caused by near-duplicate samples, which motivated the adoption of group-based geometric partitioning. Under this protocol, the Random Forest classifier achieved a mean accuracy of 97.01% ± 2.59% over ten repetitions. In an independent test with 3,812 unseen images from 15 classes, accuracy was 71.2%, revealing limitations in generalization.

[1] Graduando em Análise e Desenvolvimento de Sistemas no IFSP, Campus Jacareí.
[2] Doutor em Geofísica Espacial pelo INPE e professor de Computação no IFSP.
[3] Doutora em Computação Aplicada pelo INPE e professora de Computação no IFSP.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

The results demonstrate the feasibility of real-time operation but indicate the need to increase data diversity and improve recognition robustness.

KEYWORDS: *Brazilian Sign Language. Computer Vision. Machine Learning. Manual Alphabet Recognition. Random Forest. Digital Accessibility.*

RESUMEN
La comunicación entre personas sordas y oyentes aún enfrenta barreras de accesibilidad, especialmente ante la ausencia de intérpretes de la Lengua Brasileña de Señas (Libras). Este trabajo presenta el desarrollo y la evaluación de un sistema de reconocimiento en tiempo real de 21 configuraciones estáticas del alfabeto manual de la Libras, combinando visión por computadora, aprendizaje automático e interfaz web. La versión con procesamiento en el servidor utiliza OpenCV, MediaPipe Hands, Random Forest y Flask, mientras que una versión pública realiza la detección y la clasificación directamente en el navegador mediante una red de perceptrón multicapa (MLP). Para reducir las oscilaciones entre letras similares, se empleó un mecanismo de confirmación temporal. La evaluación inicial reveló fuga de datos causada por muestras casi duplicadas, lo que motivó la adopción de una partición por grupos geométricos. Con este protocolo, el clasificador Random Forest obtuvo una exactitud media de 97,01 % ± 2,59 % en diez repeticiones. En una prueba independiente con 3.812 imágenes inéditas de 15 clases, la exactitud fue del 71,2 %, evidenciando limitaciones de generalización. Los resultados demuestran la viabilidad del sistema en tiempo real, pero indican la necesidad de ampliar la diversidad de los datos y mejorar la robustez del reconocimiento.

PALABRAS CLAVE: *Lengua Brasileña de Señas. Visión por Computadora. Aprendizaje Automático. Reconocimiento del Alfabeto Manual. Random Forest. Accesibilidad Digital.*

1. INTRODUÇÃO

A comunicação entre pessoas surdas e ouvintes ainda enfrenta barreiras de acessibilidade, especialmente quando não há intérpretes de Língua Brasileira de Sinais (Libras) disponíveis. A Libras desempenha papel central na inclusão social e no acesso à comunicação e à informação da comunidade surda (Buzolin; Costa; Flores, 2025). Apesar dos avanços em tecnologias assistivas (Sousa; Braz Junior, 2025) e em interfaces baseadas em gestos (Burdea; Coiffet, 2003), soluções de reconhecimento automático de sinais ainda podem apresentar limitações relacionadas ao custo computacional e à facilidade de utilização.

Entre as iniciativas voltadas à acessibilidade, destaca-se o VLibras, utilizado para tradução de conteúdos digitais para Libras, embora ainda apresente desafios técnicos e de usabilidade (Lima *et al.*, 2021). No reconhecimento automático de sinais, Redes Neurais Convolucionais aplicadas diretamente às imagens apresentam resultados promissores, mas podem demandar conjuntos de dados extensos e maior capacidade computacional (Pigou *et al.*, 2015). Alternativamente, abordagens baseadas em pontos de referência (*landmarks*) da mão reduzem a dimensionalidade dos dados e favorecem sistemas em tempo real com menor custo computacional (Halder; Tayade, 2021).

Além da eficiência computacional, a capacidade de generalização dos classificadores constitui um desafio relevante. Em conjuntos formados por imagens capturadas em sequência, amostras muito semelhantes podem ser distribuídas entre treinamento e teste, produzindo estimativas excessivamente otimistas. Assim, protocolos de avaliação devem considerar tanto a independência entre as amostras quanto o desempenho em imagens obtidas sob condições distintas das utilizadas no treinamento.

Neste contexto, este trabalho apresenta o LibrasFlow, um sistema web para reconhecimento em tempo real de 21 configurações estáticas do alfabeto manual da Libras, capturadas por uma webcam convencional. O escopo restringe-se às letras representáveis por configurações estáticas, excluindo H, J, K, X e Z, que envolvem movimento e exigem processamento temporal específico.

O sistema foi implementado em duas arquiteturas: uma versão com processamento no servidor, baseada em *MediaPipe Hands* e *Random Forest*, e uma versão pública com inferência diretamente no navegador utilizando uma rede Multilayer Perceptron (MLP). A avaliação considera particionamento aleatório, particionamento por grupos geométricos e um conjunto independente de imagens, permitindo investigar o risco de vazamento de dados e a capacidade de generalização dos classificadores.

A principal contribuição deste trabalho consiste na integração do reconhecimento de configurações manuais a uma interface web, associada a uma avaliação experimental dos efeitos do particionamento dos dados e das limitações de generalização dos modelos.

2. REFERENCIAL TEÓRICO

2.1. Língua Brasileira de Sinais

A Língua Brasileira de Sinais (Libras) é uma língua visuoespacial reconhecida oficialmente pela Lei nº 10.436/2002 e utilizada pela comunidade surda brasileira como meio de comunicação e expressão (Brasil, 2002). Entre seus recursos está o alfabeto manual, empregado principalmente na soletração de nomes próprios e termos sem sinal correspondente (Quadros; Karnopp, 2004).

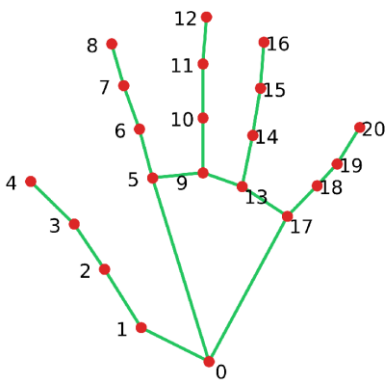
O alfabeto manual representa apenas uma parcela da Libras, pois muitos sinais dependem de movimentos das mãos e do corpo, expressões faciais e contexto. Por essa razão, o presente estudo restringe-se ao reconhecimento de configurações estáticas do alfabeto manual.

2.2. Visão computacional e extração de landmarks

Neste trabalho, o *MediaPipe Hands* é utilizado para extrair 21 pontos de referência (landmarks) da mão, representados pelas coordenadas (x), (y) e (z) (Zhang *et al.*, 2020), conforme ilustrado na Figura 1. As coordenadas (x) e (y) indicam posições normalizadas na

imagem, enquanto (z) corresponde a uma estimativa de profundidade relativa obtida a partir da câmera RGB (Google AI Edge, 2026).

Figura 1. Pontos de referência da mão (landmarks) detectados pelo MediaPipe Hands



0. Punho
1. Polegar – CMC
2. Polegar – MCF
3. Polegar – IF
4. Polegar – ponta
5. Indicador – MCF
6. Indicador – IFP
7. Indicador – IFD
8. Indicador – ponta
9. Médio – MCF
10. Médio – IFP
11. Médio – IFD
12. Médio – ponta
13. Anelar – MCF
14. Anelar – IFP
15. Anelar – IFD
16. Anelar – ponta
17. Mínimo – MCF
18. Mínimo – IFP
19. Mínimo – IFD
20. Mínimo – ponta

CMC: carpometacarpal
MCF: metacarpofalângica
IF: interfalângica
IFP/IFD: interfalângica proximal/distal

Fonte: Adaptado de Google AI Edge (2026).

Esses pontos permitem representar a configuração da mão por vetores de atributos geométricos, reduzindo a dimensionalidade em relação ao processamento direto dos pixels e possibilitando o emprego de classificadores de menor complexidade computacional (Halder; Tayade, 2021). Neste estudo, são avaliadas representações com 42 atributos ((x,y)) e 63 atributos ((x,y,z)).

2.3. Aprendizado de máquina: Random Forest e perceptron multicamadas

O *Random Forest* é um método de aprendizado supervisionado baseado na combinação de múltiplas árvores de decisão, cuja agregação tende a reduzir a variância e o sobreajuste em relação a árvores individuais (Breiman, 2001). Esse algoritmo é empregado como classificador principal na versão do LibrasFlow com processamento no servidor.

A versão pública executada no navegador utiliza uma rede do tipo perceptron multicamadas (MLP), capaz de modelar relações não lineares entre os atributos de entrada e as classes de saída por meio de treinamento baseado em retropropagação (Rumelhart; Hinton; Williams, 1986). Redes MLP de pequeno porte podem apresentar baixo custo de inferência, característica adequada à execução em ambientes web (Goodfellow; Bengio; Courville, 2016; Smilkov *et al.*, 2019).

2.4. Arquitetura cliente-servidor, streaming de vídeo e inferência no navegador

Na arquitetura cliente-servidor, o processamento é realizado no servidor e os resultados são enviados ao navegador por HTTP (Fielding; Nottingham; Reschke, 2022). No LibrasFlow, essa implementação utiliza *Flask* para processar os quadros capturados e transmitir continuamente as imagens resultantes ao cliente (Pallets Projects, 2024; Grinberg, 2014).

Como alternativa, a versão pública executa a detecção e a classificação diretamente no navegador, utilizando *MediaPipe Hand Landmarker* e *WebAssembly*. Essa abordagem evita a transmissão contínua das imagens ao servidor, embora seu desempenho dependa das características do dispositivo, do navegador e do modelo empregado (Haas *et al.*, 2017).

3. METODOLOGIA

Esta seção descreve os procedimentos adotados para o desenvolvimento e a avaliação do sistema LibrasFlow, organizados em três etapas: (i) preparação dos dados e desenvolvimento do classificador; (ii) avaliação experimental do desempenho e da capacidade de generalização; e (iii) implementação e análise das arquiteturas de execução em tempo real.

Na arquitetura inicial, as imagens capturadas por uma *webcam* são processadas pelo *MediaPipe Hands*, responsável pela identificação de 21 pontos de referência (*landmarks*) da mão. As coordenadas bidimensionais (x, y) desses pontos são extraídas e normalizadas, formando um vetor de 42 atributos utilizado como entrada do classificador *Random Forest*. A classe estimada é submetida a um mecanismo de confirmação temporal para reduzir oscilações entre predições consecutivas, antes de ser apresentada ao usuário.

A Figura 2 apresenta o fluxo de processamento dessa arquitetura. Posteriormente, foi desenvolvida uma segunda versão, com inferência diretamente no navegador, que utiliza uma rede neural do tipo *perceptron* multicamadas (*Multilayer Perceptron* — MLP). Como as duas implementações utilizam classificadores e representações de atributos distintos, seus procedimentos são descritos separadamente nas Subseções 3.5 e 3.6.

Figura 2. Fluxo de processamento do LibrasFlow com Random Forest no servidor



Fonte: Elaborado pelos autores (2026).

3.1. Ambiente experimental e ferramentas

O sistema foi desenvolvido em Python com OpenCV, MediaPipe Hands, Scikit-Learn, NumPy, Pandas e Flask. Os experimentos foram executados em um notebook Intel Core i5-13420H, com 8 GB de RAM, Windows 10 e webcam 720p, sem GPU dedicada. Foram utilizados Python 3.11.9, MediaPipe 0.10.14, OpenCV 4.11.0, Flask 3.0.0 e Scikit-Learn 1.9.0. A avaliação interna foi reproduzida em Linux, com resultados idênticos para o *Random Forest* e diferenças inferiores a 0,2 ponto percentual para a MLP. Os principais parâmetros são apresentados na Tabela 1.

Tabela 1. Principais parâmetros utilizados nos experimentos

Etapa ou componente	Parâmetros
MediaPipe Hands	Imagem estática; 1 mão; confiança ≥ 0,5
Normalização	Relativa ao punho; escala pelo maior valor absoluto
Random Forest	100 árvores; profundidade 15; mín. 2 amostras/folha
Particionamento aleatório	80% treino / 20% teste; estratificado
Particionamento por grupos	DBSCAN/classe; ε = 0,05; *min_samples* = 1; ~20% teste
Repetições	Sementes 0-9
MLP (versão pública)	63 → 128 → 64 → 21; ReLU; Adam; ≤ 300 épocas
Confirmação temporal	30 quadros (servidor); 12 quadros + confiança ≥ 0,75 (navegador)
Botão virtual “LIMPAR”	25 quadros
Medição de latência	300 quadros consecutivos, *webcam* de 720p

Fonte: Elaborado pelos autores (2026).

3.2. Coleta e preparação dos dados

Foi utilizado o conjunto de dados público “Libras” (Oliveira, 2022), disponível na plataforma Kaggle, contendo imagens organizadas em diretórios correspondentes às letras do alfabeto manual.

Como a classificação é realizada sobre quadros individuais, foram selecionadas 21 das 26 letras do alfabeto, excluindo-se H, J, K, X e Z, que envolvem movimento ou trajetória não representados por uma única imagem estática.

O processamento das imagens foi realizado utilizando o *MediaPipe Hands*, configurado para imagens estáticas, detecção de uma única mão e confiança mínima de detecção de 0,5. Para cada imagem em que uma mão foi identificada, foram extraídas as coordenadas dos 21 pontos de referência (*landmarks*), incluindo o punho, as articulações e as extremidades dos dedos.

Embora o *MediaPipe* forneça três coordenadas por ponto (x, y, z), o classificador principal utiliza apenas x e y, de modo que cada imagem é convertida em um vetor de 42 atributos numéricos.

3.2.1. Normalização dos atributos

Para reduzir a influência da posição da mão e de variações uniformes de escala, as coordenadas do punho (ponto 0) foram subtraídas das de todos os pontos de referência, e os valores resultantes foram divididos pelo maior valor absoluto do vetor. Essa operação não garante invariância às rotações, aos efeitos de perspectiva ou às mudanças na orientação da mão em relação à câmera. Para os pontos de índices i = 0, ..., 20, a normalização é expressa pelas Equações 1 e 2.

$$x_i^* = (x_i - x_0) / s \quad (1)$$

$$y_i^* = (y_i - y_0) / s \quad (2)$$

em que x_0 e y_0 representam as coordenadas do punho e s corresponde ao maior valor absoluto entre todas as coordenadas relativas x e y.

Os vetores normalizados foram armazenados em formato tabular (CSV), contendo 42 atributos numéricos e o rótulo correspondente à letra representada. Ao final do processamento, o conjunto de dados utilizado no treinamento e na avaliação do classificador foi composto por 34.374 amostras, distribuídas entre as 21 classes selecionadas.

Durante a análise exploratória, foram identificadas amostras geometricamente muito semelhantes, o que motivou a adoção de um protocolo baseado em agrupamento geométrico, além da divisão aleatória convencional.

3.3. Desenvolvimento do modelo de classificação

O reconhecimento das configurações estáticas da mão foi formulado como um problema de classificação supervisionada multiclasse, no qual cada vetor de 42 atributos deve ser associado a uma das 21 letras consideradas no estudo. Para essa tarefa, foi utilizado o algoritmo *Random Forest* (Breiman, 2001), implementado por meio da biblioteca *Scikit-Learn* (Pedregosa *et al.*, 2011), com os hiperparâmetros apresentados na Tabela 1.

Após os procedimentos de avaliação, o classificador utilizado na aplicação foi treinado novamente com todas as 34.374 amostras rotuladas e serializado em formato binário (*pickle*).

Esse modelo final não participou da avaliação interna e foi utilizado posteriormente na avaliação com o conjunto independente.

3.4. Protocolos de avaliação experimental

A avaliação considerou três procedimentos principais: particionamento aleatório estratificado, particionamento por grupos geométricos e teste independente com imagens inéditas. Adicionalmente, foram realizados dois experimentos complementares: a análise da influência da coordenada de profundidade estimada pelo *MediaPipe* (Seção 3.4.4) e a avaliação da arquitetura MLP empregada na versão pública (Seção 3.4.5).

3.4.1. Particionamento aleatório estratificado

Inicialmente, o conjunto de dados foi dividido aleatoriamente em 80% das amostras para treinamento e 20% para teste, com estratificação por classe. Esse procedimento foi utilizado como referência, embora a presença de amostras geometricamente muito semelhantes possa produzir estimativas excessivamente otimistas de desempenho.

3.4.2. Particionamento por grupos geométricos

Para reduzir o risco de compartilhamento de amostras quase duplicadas entre treinamento e teste, as amostras de cada classe foram agrupadas pelo algoritmo *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN), com raio de vizinhança $\varepsilon = 0{,}05$ no espaço de atributos normalizados e parâmetro *min_samples* = 1.

A escolha do raio $\varepsilon = 0{,}05$ baseou-se em dois critérios: agrupar configurações praticamente idênticas sem fundir poses efetivamente distintas e preservar número suficiente de grupos por classe para manter aproximadamente 20% das amostras no conjunto de teste. Esses critérios foram avaliados por análise de sensibilidade com ε igual a 0,02; 0,03; 0,05; 0,075; 0,10 e 0,15, considerando o número de grupos, o menor número de grupos por classe, a fração efetiva de teste e a acurácia nas dez repetições.

Para verificar se o agrupamento reduziu o compartilhamento de amostras muito semelhantes entre treinamento e teste, calculou-se, para cada amostra de teste, a distância euclidiana até a amostra de treinamento mais próxima da mesma classe, no espaço de atributos normalizados. Após a identificação dos grupos, cada grupo foi tratado como uma unidade indivisível, destinada exclusivamente ao treinamento ou ao teste, reservando-se aproximadamente 20% das amostras de cada classe para avaliação. Esse procedimento reduz a presença de configurações quase duplicadas entre os conjuntos, mas não garante independência entre usuários, sessões de aquisição ou condições ambientais.

Os protocolos aleatórios e por grupos foram executados dez vezes, com sementes aleatórias de 0 a 9 para o particionamento e para a inicialização do classificador. As métricas

(acurácia e *F1-score* macro e, por classe, precisão e revocação) foram calculadas em cada repetição e sintetizadas por média, desvio-padrão amostral, valor mínimo e valor máximo.

3.4.3. Avaliação com imagens inéditas

Para investigar a capacidade de generalização, o modelo final (Subseção 3.3) foi avaliado uma única vez, sem novo treinamento, em um conjunto independente de imagens, previamente processadas pelo *MediaPipe*.

As 4.411 imagens utilizadas pertencem ao conjunto público *Brazilian Sign Language Alphabet* (Passos; Fernandes; Comunello, 2020), composto por imagens de 200 × 200 pixels de 15 letras estáticas. Segundo seus autores, essas imagens foram selecionadas do conjunto *ASL Alphabet*, disponível na plataforma Kaggle, e são compostas por sinais do alfabeto manual da Língua de Sinais Americana (ASL) com representação considerada equivalente na Libras.

Esse conjunto possui origem distinta da do conjunto de treinamento e condições de captura diferentes. Entretanto, a documentação dos dois conjuntos não informa o número de sinalizantes, o que impede caracterizar a avaliação como uma validação entre usuários.

A inspeção dos dados revelou diferenças de configuração nas letras M e N entre os dois conjuntos e variação também na representação da letra E. Por esse motivo, os resultados foram apresentados considerando as 15 classes, 13 classes sem M e N e 12 classes sem M, N e E.

A distribuição das imagens entre as classes é desbalanceada. Por isso, além da acurácia global, foram calculadas a taxa de acerto por classe e a acurácia balanceada. A incerteza das estimativas foi expressa por intervalos de confiança de 95%, calculados pelo método de Wilson.

Os classificadores foram avaliados sobre as mesmas detecções, permitindo comparações pareadas pelo teste exato de McNemar. Adicionalmente, calculou-se a distância euclidiana entre cada amostra externa e a amostra de treinamento mais próxima no espaço de 42 atributos.

3.4.4. Análise da influência da coordenada de profundidade

Para investigar o efeito da coordenada z estimada pelo *MediaPipe*, a representação bidimensional com 42 atributos (x, y) foi comparada com uma representação de 63 atributos (x, y, z), ambas centralizadas em relação ao punho e normalizadas pelo maior valor absoluto do vetor.

Para isolar o efeito da coordenada z, as duas representações foram avaliadas com exatamente as mesmas partições de treinamento e teste e as mesmas dez sementes. As acurácias pareadas foram comparadas pelo teste de postos sinalizados de Wilcoxon, com nível de significância de 5%. No teste com imagens inéditas, as predições pareadas foram comparadas pelo teste de McNemar.

3.4.5. Avaliação da arquitetura MLP

A arquitetura MLP empregada na versão pública, cujos parâmetros são apresentados na Tabela 1, foi avaliada nas mesmas partições e sementes utilizadas nos demais experimentos. Um modelo treinado com todas as amostras também foi avaliado no conjunto independente. Essa avaliação caracteriza a arquitetura MLP, mas não corresponde aos pesos exatos do modelo disponibilizado publicamente. A latência da inferência no navegador não foi medida.

3.5. Arquitetura Web com processamento no servidor

Na versão com processamento no servidor, o Flask disponibiliza a interface web e transmite ao navegador os quadros processados. Cada quadro capturado pela webcam é analisado pelo *MediaPipe Hands*, que extrai os 21 pontos de referência da mão; as coordenadas x e y são normalizadas conforme a Subseção 3.2.1 e classificadas pelo *Random Forest*.

Para reduzir oscilações entre predições consecutivas, foi empregado um mecanismo de confirmação temporal, cujos parâmetros estão apresentados na Tabela 1. A interface exibe o vídeo processado, os pontos de referência da mão, a letra reconhecida, o texto construído, o indicador “AO VIVO”, o botão virtual “LIMPAR” e um guia do alfabeto manual. Nessa arquitetura, a inferência é executada no servidor, enquanto o navegador atua como meio de visualização e interação.

3.6. Versão pública com inferência no navegador

Uma segunda versão do LibrasFlow foi desenvolvida para executar a detecção da mão e a classificação diretamente no dispositivo do usuário, dispensando o processamento remoto das imagens. Nessa implementação, o MediaPipe Hand Landmarker é executado no navegador por meio de WebAssembly, com possibilidade de aceleração por GPU.

A classificação utiliza uma rede MLP, cujos parâmetros estão apresentados na Tabela 1, alimentada pelas coordenadas x, y e z dos 21 pontos de referência normalizados. A classe de maior saída é utilizada como predição, e o respectivo valor é empregado apenas como indicador de confiança, uma vez que a calibração probabilística do modelo não foi avaliada. O mecanismo de confirmação temporal também utiliza os parâmetros apresentados na Tabela 1.

Como essa versão difere da arquitetura com processamento no servidor quanto ao classificador, à dimensionalidade dos atributos e ao mecanismo de confirmação, os resultados obtidos com o *Random Forest* não podem ser diretamente extrapolados para ela. A arquitetura MLP foi avaliada sob o protocolo interno e no conjunto de imagens inéditas, mas sua latência no navegador não foi medida; por esse motivo, a versão pública é apresentada como uma demonstração funcional.

3.7. Avaliação do desempenho computacional

Para investigar a viabilidade da execução em tempo real da versão baseada em *Random Forest*, mediu-se a latência média das principais etapas ao longo de 300 quadros consecutivos, com a mão do usuário visível à câmera: captura do quadro, detecção dos pontos de referência pelo *MediaPipe,* classificação pelo *Random Forest*, medida dentro do próprio laço de processamento, e não em lote, e fluxo completo, da captura até a classificação, sem incluir a exibição do quadro. A taxa de quadros por segundo (FPS) equivalente foi calculada como o inverso da latência média, e também foi registrada a taxa efetiva de processamento de ponta a ponta.

A medição foi realizada em uma única execução no equipamento descrito na Seção 3.1 e caracteriza apenas as condições específicas desse experimento. A versão pública não foi submetida ao mesmo procedimento, de modo que não foram realizadas comparações de latência entre as duas arquiteturas.

4. RESULTADOS E DISCUSSÃO

4.1. Influência do protocolo de particionamento no desempenho do classificador

Na divisão aleatória estratificada (80% das amostras para treinamento e 20% para teste; 6.875 amostras de teste), o *Random Forest* com 42 atributos obteve, em dez repetições independentes, acurácia média de 99,96% ± 0,02% (mínimo de 99,91% e máximo de 99,99%) e *F1-score* macro de 99,96% ± 0,02%, com valores de precisão e revocação próximos de 1,00 para praticamente todas as 21 classes.

A acurácia próxima de 100%, associada à reduzida variabilidade entre as repetições, motivou uma investigação sobre dependências entre as amostras de treinamento e teste, que identificou amostras geometricamente muito semelhantes no conjunto de 34.374 amostras, indicando risco de vazamento de dados (*data leakage*). O agrupamento por densidade (DBSCAN, $\varepsilon = 0,05$) identificou 6.270 grupos geométricos, com aproximadamente 5,5 amostras por grupo, em média, embora cada grupo não corresponda necessariamente a uma pose fisicamente distinta ou a uma aquisição independente.

Com o particionamento por grupos geométricos (Seção 3.4.2), a acurácia média em dez repetições foi de 97,01% ± 2,59% (mínimo de 92,69% e máximo de 99,90%), com *F1-score* macro de 96,74% ± 2,99% e conjuntos de teste entre 6.879 e 6.890 amostras.

A verificação das distâncias entre as amostras confirmou o efeito do particionamento sobre o compartilhamento de configurações semelhantes. No particionamento aleatório, 83,68% ± 0,33% das amostras de teste possuíam, no conjunto de treinamento, uma amostra da mesma classe a uma distância inferior a 0,05, e a mediana dessa distância foi de 0,031. No

particionamento por grupos, essa proporção foi nula em todas as repetições, e a mediana da distância ao vizinho de treinamento mais próximo aumentou para 0,085. Como o agrupamento mantém no mesmo grupo todas as amostras conectadas por distâncias inferiores a ε, o resultado confirma que o procedimento eliminou os pares de quase duplicatas entre treinamento e teste, aumentando a distância típica entre os dois conjuntos em aproximadamente 2,7 vezes.

A Tabela 2 apresenta a redução progressiva da acurácia com o aumento de ε. Valores inferiores a 0,05 produziram agrupamentos muito próximos do particionamento aleatório, enquanto valores superiores reduziram excessivamente o número de grupos de algumas classes e alteraram a proporção de teste. Assim, ε = 0,05 foi adotado por representar o maior raio avaliado que preservou ao menos 16 grupos por classe e aproximadamente 20% das amostras no conjunto de teste.

Tabela 2. Sensibilidade do particionamento por grupos ao raio ε do DBSCAN (Random Forest com 42 atributos; média ± desvio-padrão em dez repetições)

ε	Grupos	Amostras por grupo	Mínimo de grupos por classe	Teste (%)	Acurácia (%)
0,02	30.266	1,14	912 (B)	20,0	99,97 ± 0,02
0,03	19.413	1,77	303 (B)	20,0	99,95 ± 0,03
0,05	6.270	5,48	16 (B)	20,0	97,01 ± 2,59
0,075	2.133	16,12	5 (B)	21,1	93,65 ± 3,17
0,10	961	35,77	4 (B)	22,8	88,13 ± 2,21
0,15	347	99,06	2 (L)	33,4	58,22 ± 0,99

Fonte: Elaborado pelos autores (2026).

A redução progressiva da acurácia com o aumento de ε, assim como a maior variabilidade em relação à divisão aleatória, indica que a estimativa interna depende do grau de separação imposto entre as configurações de treinamento e de teste, o que é compatível com a hipótese de que a proximidade geométrica entre amostras contribuiu para os resultados quase perfeitos observados inicialmente. Como o particionamento por grupos não garante independência entre usuários ou sessões de aquisição, a generalização foi investigada também com imagens inéditas (Seção 4.2).

A análise por classe nas dez repetições mostra que a variabilidade observada se concentra em poucas letras. Enquanto 16 das 21 classes apresentaram revocação média igual ou superior a 98,7%, as letras G (81,06% ± 37,79%), T (85,63% ± 22,02%), R (88,85% ± 16,63%), F (90,09%

± 15,32%) e U (94,15% ± 10,75%) apresentaram médias inferiores e desvios elevados, com mínimos de 9,09% (G) e 53,56% (T). As confusões mais frequentes, somadas nas dez repetições, foram G→L (18,45% das amostras de G), T→F (14,37%), R→U (11,15%), F→T (9,76%) e U→R (5,00%), pares de configurações próximas que diferem principalmente pela orientação da mão ou pela sobreposição dos dedos. Uma possível explicação é que, nessas letras, alguns grupos concentram parcela expressiva das amostras (até 26,7% na letra G); quando um desses grupos é destinado ao teste, configurações sem equivalente no treinamento passam a ser atribuídas à letra geometricamente mais próxima.

4.2. Avaliação da capacidade de generalização em imagens inéditas

Das 4.411 imagens do conjunto independente (Seção 3.4.3), 599 (13,6%) não produziram detecção válida da mão pelo *MediaPipe*, restando 3.812 imagens para a avaliação do modelo final, treinado com as 34.374 amostras. As falhas de detecção concentraram-se nas letras N (125 de 156 imagens; 80,1%), M (142 de 197; 72,1%), O (57 de 150; 38,0%) e S (47 de 151; 31,1%); nas demais letras, a proporção de falhas não ultrapassou 12%.

A verificação de independência não identificou imagens repetidas entre os dois conjuntos: nenhuma amostra externa coincidiu com uma amostra de treinamento, e nenhuma apresentou vizinho da mesma classe a distância inferior a 0,05, limiar adotado no particionamento por grupos. A mediana da distância ao vizinho de treinamento mais próximo foi de 0,428, aproximadamente cinco vezes a observada no particionamento por grupos (0,085; Seção 4.1), o que confirma que o conjunto externo é composto por configurações distintas das utilizadas no treinamento.

A acurácia obtida pelo *Random Forest* com 42 atributos, configuração adotada na versão com processamento no servidor, foi de 71,22% (2.715 acertos em 3.812 imagens; intervalo de confiança de 95%: 69,8% a 72,6%), substancialmente inferior às acurácias medidas dentro do próprio conjunto de dados. Embora o protocolo por grupos tenha reduzido o compartilhamento de configurações quase duplicadas entre treinamento e teste, o desempenho observado nesse protocolo não se reproduziu no conjunto externo.

Cabe ressaltar que esse valor se refere exclusivamente às imagens com detecção válida da mão e às 15 classes presentes no conjunto externo, não representando, isoladamente, a taxa de reconhecimento sobre todas as 4.411 imagens nem o desempenho para todo o alfabeto considerado no treinamento. Além disso, como a documentação do conjunto externo não descreve os sinalizantes nem as condições de aquisição, não é possível determinar com precisão quais fatores contribuíram predominantemente para a redução observada.

4.3. Influência da coordenada de profundidade e análise dos erros

Para investigar o efeito da coordenada z, a representação de 42 atributos (x, y) foi comparada com a de 63 atributos (x, y, z). Em uma análise preliminar, com os grupos recalculados no espaço de 63 atributos, a configuração tridimensional obteve acurácia média de 99,73% ± 0,47%. Entretanto, nesse espaço o DBSCAN produziu 12.841 grupos, contra 6.270 na representação bidimensional, o que torna o particionamento menos restritivo e impede a comparação direta. Por esse motivo, as duas representações foram avaliadas novamente com exatamente as mesmas partições (Seção 3.4.4), cujos resultados são apresentados na Tabela 3.

Tabela 3. Desempenho dos classificadores nas mesmas partições por grupos (média ± desvio-padrão em dez repetições)

Modelo	Atributos	Acurácia (%)	Mínimo–máximo (%)	*F1-score* macro (%)
Random Forest	42 (x, y)	97,01 ± 2,59	92,69–99,90	96,74 ± 2,99
Random Forest	63 (x, y, z)	97,84 ± 1,86	94,42–99,87	97,71 ± 2,04
MLP	63 (x, y, z)	96,39 ± 2,69	91,25–99,94	95,56 ± 3,52

Fonte: Elaborado pelos autores (2026).

Nas mesmas partições, não houve diferença significativa entre as representações com 42 e 63 atributos ($p = 0,85$), indicando que a inclusão da coordenada z não produziu ganho consistente na avaliação interna. A vantagem observada na análise preliminar decorreu, em grande parte, da diferença de granularidade dos agrupamentos, e não da inclusão da profundidade.

A MLP também apresentou desempenho interno semelhante ao dos modelos *Random Forest*, sem diferenças estatisticamente significativas em relação às configurações com 63 atributos ($p = 0,28$) e 42 atributos ($p = 0,43$). No conjunto de imagens inéditas, entretanto, sua acurácia foi de 59,39% (IC 95%: 57,8% a 60,9%), significativamente inferior à dos dois modelos *Random Forest* ($p < 0,001$), evidenciando que a equivalência observada internamente não se manteve diante de novas condições de captura.

No mesmo conjunto externo, o *Random Forest* com 63 atributos obteve acurácia de 69,39% (IC 95%: 67,9% a 70,8%), contra 71,22% (IC 95%: 69,8% a 72,6%) para a representação com 42 atributos. A comparação pareada pelo teste de McNemar indicou diferença significativa ($p < 0,001$), com vantagem para a representação bidimensional. Assim, os resultados não sustentam a inclusão da coordenada z como atributo adicional nas condições avaliadas. Uma

possível explicação envolve a sensibilidade dessa coordenada, estimada a partir de uma câmera RGB convencional, às condições de aquisição, embora os dados disponíveis não permitam confirmar essa hipótese.

A Figura 3 apresenta a matriz de confusão da configuração com 42 atributos, adotada na versão com processamento no servidor, e não do classificador MLP empregado na versão pública.

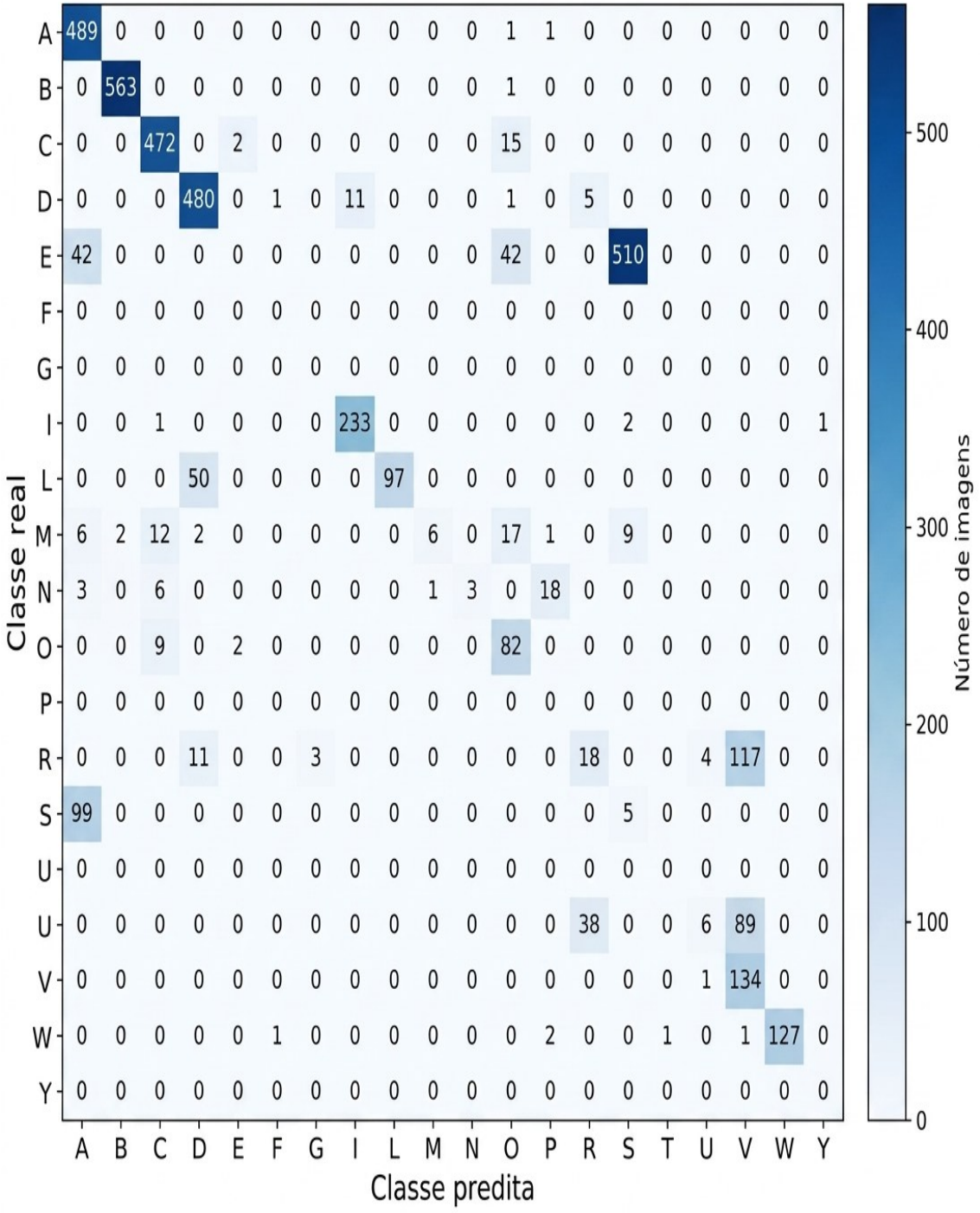
Os principais erros concentraram-se em pares de configurações visualmente semelhantes. A letra E foi classificada predominantemente como S; também ocorreram confusões relevantes entre S/A, R/V, U/V, U/R, L/D e N/P. Em contrapartida, B e V foram reconhecidas corretamente em todas as imagens avaliadas, enquanto A, I e W apresentaram altas taxas de acerto.

No caso da letra E, a inspeção das imagens revelou diferença de configuração entre os conjuntos: no conjunto externo, os dedos aparecem mais flexionados sobre o polegar, aproximando-se de um punho fechado, enquanto no treinamento predomina uma forma mais aberta. As duas representações são compatíveis com variantes da letra E descritas para a Libras (Diniz, 2020), mas apenas uma delas está adequadamente representada no conjunto de treinamento. Essa diferença contribui para a classificação frequente das imagens externas de E como S.

A acurácia balanceada da configuração com 42 atributos foi de 58,9%, inferior à acurácia global de 71,22%, evidenciando a influência do desbalanceamento entre as classes. As maiores dificuldades ocorreram nas letras E, U, S, N, M e R. Na configuração com 63 atributos, a acurácia balanceada foi ainda menor, 55,6%, com padrão de erros semelhante, indicando que as principais dificuldades de reconhecimento persistiram independentemente da inclusão da coordenada z.

Quando M e N, cujas configurações diferem entre os conjuntos, foram excluídas, a acurácia da representação com 42 atributos aumentou para 72,62%, contra 70,99% para a representação com 63 atributos. Com a exclusão adicional da letra E, os valores atingiram 85,26% e 83,30%, respectivamente. Os erros remanescentes concentraram-se principalmente em R, S e U, que diferem de outras classes por detalhes geométricos finos e dependem de pontos de referência parcialmente ocultos ou mais sujeitos a erro de detecção. Esses resultados indicam que a queda de desempenho externo decorre tanto de diferenças de configuração entre os conjuntos quanto das condições de captura das imagens.

Figura 3. Matriz de confusão do classificador Random Forest com 42 atributos (x, y) no teste independente com imagens inéditas (acurácia de 71,22%)



Fonte: Elaborado pelos autores (2026). Linhas sem amostras correspondem às letras não representadas entre as imagens com detecção válida da mão.

4.4. Desempenho computacional do sistema

A Tabela 4 apresenta a latência média e o FPS equivalente de cada etapa, obtidos em uma única execução de 300 quadros, com o *hardware* descrito na Seção 3.1.

Tabela 4. Latência média e FPS equivalente por etapa do pipeline de reconhecimento

Etapa	Latência média (ms)	FPS equivalente
Captura de quadro (*OpenCV*)	3,98	251,1
Detecção de *landmarks* (*MediaPipe Hands*)	16,72	59,8
Classificação (*Random Forest*)	25,57	39,1
***Pipeline* completo (captura + *MediaPipe* + classificação)**	**40,37**	**24,8**

Fonte: Elaborado pelos autores (2026). Valores obtidos em uma única execução de 300 quadros, no hardware descrito na Seção 3.1.

A etapa de classificação apresentou latência média de 25,57 ms (mediana de 28,15 ms; percentil 95 de 31,61 ms), e o *pipeline* completo, de 40,37 ms, correspondente a aproximadamente 24,8 FPS; a taxa efetivamente observada de ponta a ponta foi de 19,5 FPS.

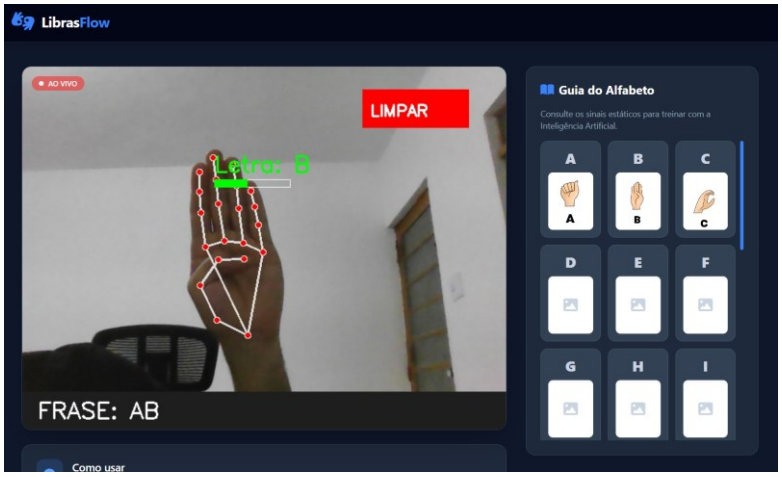
Ressalta-se que as latências médias das etapas não são diretamente somáveis: a latência de classificação foi calculada apenas sobre os quadros em que a mão foi detectada, enquanto as demais etapas e o *pipeline* completo consideram todos os 300 quadros. Além disso, o tempo do *pipeline* completo foi medido do início da captura até o término da classificação, sem incluir a exibição do quadro; a diferença entre o FPS equivalente (24,8) e a taxa efetivamente observada (19,5 FPS) decorre principalmente dessa exibição e das demais operações do laço não cronometradas individualmente.

4.5. Funcionamento da aplicação e interface Web

Nos testes funcionais da aplicação *web* com processamento no servidor, o mecanismo de confirmação temporal reduziu as oscilações na inclusão das letras, e o botão virtual “LIMPAR” permitiu apagar o texto por meio de um gesto do usuário. Esses testes demonstram o funcionamento dos mecanismos de interação, mas não constituem uma avaliação quantitativa de usabilidade ou de robustez em diferentes condições de operação.

A Figura 4 apresenta a interface em funcionamento, com o indicador “AO VIVO”, a área de vídeo com o botão virtual “LIMPAR”, a barra de texto reconhecido e o guia de referência do alfabeto manual.

Figura 4. Interface web do sistema LibrasFlow em operação, na versão com processamento no servidor



Fonte: Elaborado pelos autores (2026).

Na versão pública (Seção 3.6), o processamento é executado no navegador, dispensando a transmissão dos quadros para inferência remota. Entretanto, como essa implementação utiliza um classificador MLP com 63 atributos, cujo desempenho no conjunto de imagens inéditas foi inferior ao do *Random Forest* (Seção 4.3) e cuja latência no navegador não foi medida, os resultados de desempenho temporal obtidos com o *Random Forest* não podem ser extrapolados para ela, e sua caracterização permanece como etapa a ser realizada.

4.6. Discussão geral dos resultados

A diferença entre a acurácia de 99,96% obtida no particionamento aleatório, a média de 97,01% observada no particionamento por grupos geométricos e os 71,2% do teste com imagens inéditas evidencia a influência da estratégia de divisão dos dados sobre as estimativas de desempenho: a separação por similaridade geométrica, embora relevante, não é suficiente para caracterizar integralmente a robustez do classificador diante de novas condições. A análise de sensibilidade do raio ε e o experimento com a coordenada z reforçam essa interpretação, pois a acurácia interna diminui à medida que se exige maior separação entre treinamento e teste, e a inclusão da profundidade, sem ganho significativo na avaliação interna, esteve associada a desempenho inferior no conjunto externo.

A representatividade dos conjuntos de dados constitui uma limitação importante para a interpretação desses resultados. O conjunto utilizado no treinamento compreende 21 configurações estáticas, enquanto o conjunto externo contém apenas 15 dessas classes e é derivado, em parte, de imagens do alfabeto manual da ASL. Além disso, foram identificadas

diferenças de configuração nas letras M e N e, em menor grau, na letra E. A ausência de informações sobre o número de sinalizantes, as sessões de aquisição e as condições de captura nos dois conjuntos impede avaliar de forma controlada a influência da variabilidade entre usuários. Assim, o desempenho obtido no conjunto externo deve ser interpretado como evidência de generalização para imagens de origem distinta, mas não como uma validação entre sinalizantes nem como uma estimativa definitiva do desempenho do sistema em situações reais de uso.

Do ponto de vista da implementação, o LibrasFlow demonstra a possibilidade de integrar extração de pontos de referência, classificação supervisionada e interação gestual em uma interface *web*; contudo, a disponibilidade funcional da aplicação e as métricas de classificação constituem dimensões distintas de avaliação.

Também é importante distinguir os resultados obtidos pelos dois classificadores empregados nas implementações do LibrasFlow. Sob o protocolo interno, a arquitetura MLP apresentou desempenho próximo ao *Random Forest*, sem diferença estatisticamente significativa. Entretanto, no conjunto de imagens inéditas, a MLP apresentou acurácia inferior, indicando que a equivalência observada na avaliação interna não se manteve diante de novas condições de captura. Dessa forma, os resultados de classificação e de desempenho temporal obtidos para a versão baseada em *Random Forest* não devem ser extrapolados para a versão pública baseada em MLP, cuja latência no navegador também não foi avaliada experimentalmente.

Além disso, embora os testes funcionais tenham demonstrado o funcionamento técnico da aplicação, não foi realizada uma avaliação sistemática com usuários reais, incluindo pessoas surdas ou sinalizantes com diferentes perfis de uso da Libras. Consequentemente, os resultados apresentados não permitem inferir aspectos como usabilidade, aceitabilidade, facilidade de interação ou desempenho em situações reais de comunicação. Estudos futuros devem contemplar protocolos com múltiplos sinalizantes, identificação das sessões de aquisição e avaliação com usuários, permitindo caracterizar de forma mais abrangente a robustez e a aplicabilidade do sistema.

Portanto, a principal contribuição experimental do estudo está na investigação das diferenças entre protocolos de avaliação, na identificação das limitações de generalização dos classificadores e na demonstração da importância de considerar a representatividade dos dados e a independência entre usuários e condições de aquisição na avaliação de sistemas de reconhecimento de configurações manuais.

5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento e a avaliação do sistema LibrasFlow, voltado ao reconhecimento automático, em tempo real, de 21 configurações estáticas do alfabeto manual da Libras, com base em 42 atributos geométricos extraídos pelo *MediaPipe Hands* e no algoritmo *Random Forest*. Os resultados demonstraram a viabilidade dessa representação, mas evidenciaram limitações importantes na capacidade de generalização do modelo.

Uma das principais contribuições metodológicas do estudo foi a investigação da influência do particionamento dos dados sobre as estimativas de desempenho: em dez repetições, a acurácia média passou de 99,96% ± 0,02% na divisão aleatória estratificada para 97,01% ± 2,59% no particionamento por grupos geométricos, o que evidencia o efeito de configurações quase duplicadas. O agrupamento reduziu esse compartilhamento entre treinamento e teste, embora não garanta independência entre usuários, sessões ou condições de aquisição.

No teste independente, a acurácia foi de 71,22% (IC 95%: 69,8% a 72,6%) em 3.812 imagens com detecção válida da mão, pertencentes a 15 das 21 classes, valor substancialmente inferior ao das avaliações internas e condicionado ao sucesso da detecção pelo *MediaPipe*; desconsideradas as letras cuja configuração difere entre os conjuntos (E, M e N), a acurácia foi de 85,26% (IC 95%: 84,0% a 86,4%). Com as mesmas partições, a inclusão da coordenada z não produziu diferença significativa na avaliação interna (97,84% ± 1,86%, contra 97,01% ± 2,59%; $p = 0,85$) e reduziu significativamente a acurácia no teste independente (69,39%; $p < 0,001$ pelo teste de McNemar), o que não sustenta a adoção da profundidade estimada como atributo adicional.

O trabalho também demonstrou a integração das etapas de detecção da mão, extração de atributos, classificação e confirmação temporal em uma aplicação *web* com processamento no servidor e em uma versão pública com inferência local no navegador. A arquitetura MLP desta última não diferiu significativamente do *Random Forest* no protocolo interno (96,39% ± 2,69%), mas apresentou desempenho inferior no conjunto de imagens inéditas (59,39%) e não teve seu desempenho temporal medido, sendo considerada uma demonstração funcional.

Entre as principais limitações do estudo, destacam-se a restrição a configurações estáticas; a abrangência do conjunto externo, composto por apenas 15 das 21 classes e derivado de imagens do alfabeto manual da ASL, no qual as letras M e N e a forma da letra E diferem das representadas no conjunto de treinamento; a ausência de documentação sobre os sinalizantes dos dois conjuntos; e a concentração dos erros remanescentes em pares de letras que diferem por detalhes finos, especialmente S/A, R/V, U/V e L/D.

Como trabalhos futuros, recomenda-se ampliar e diversificar os conjuntos de dados, incluindo diferentes usuários, condições de iluminação, distâncias e ângulos de câmera; adotar protocolos de avaliação que considerem a independência entre usuários ou sessões de aquisição; realizar uma avaliação externa com imagens de Libras produzidas por sinalizantes identificados; investigar a menor capacidade de generalização do modelo MLP e seu desempenho em diferentes dispositivos; e estender o sistema a sinais dinâmicos, por meio de modelos temporais, como redes LSTM ou *Transformers*, aplicados a sequências de pontos de referência.

Em síntese, a abordagem desenvolvida constitui uma base computacional para o reconhecimento de configurações manuais em tempo real e evidencia a importância de protocolos experimentais rigorosos para estimar a capacidade de generalização de classificadores de sinais.

DISPONIBILIDADE DO CÓDIGO-FONTE

O código-fonte e os materiais computacionais associados ao desenvolvimento e à avaliação do sistema LibrasFlow estão disponíveis publicamente no repositório https://github.com/SantosGAlves/LibrasFlow (Santos; Gonçalves, 2026). O repositório reúne as rotinas de extração e normalização de atributos, treinamento, particionamento por grupos geométricos, avaliação interna e externa, experimentos complementares (análise de sensibilidade do raio ε, verificação de vazamento, comparação entre representações e avaliação da arquitetura MLP) e medição de latência, além dos códigos da aplicação *web*, dos dados tabulares processados e dos modelos treinados.

A versão pública de demonstração, com inferência executada localmente no navegador, pode ser acessada em https://librasflow.ifsp.dev (LibrasFlow, 2026). Essa implementação utiliza um classificador MLP, distinto do *Random Forest* avaliado quantitativamente neste estudo.

REFERÊNCIAS

BRASIL. Lei nº 10.436, de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais - Libras e dá outras providências. **Diário Oficial da União**: seção 1, Brasília, DF, 25 abr. 2002.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5-32, 2001. DOI: 10.1023/A:1010933404324.

BURDEA, G.; COIFFET, P. **Virtual reality technology**. 2. ed. New Jersey: John Wiley & Sons, 2003.

BUZOLIN, A. A.; COSTA, A. da; FLORES, T. C. A importância da Libras no processo de ensino aprendizagem. **Revista Acadêmica da Lusofonia**, [s. l.], v. 2, n. 7, p. 1-14, 2025. DOI: 10.69807/2966-0785.2025.108. Disponível em: https://revistaacademicadalusofonia.com/index.php/lusofonia/article/view/108. Acesso em: 30 out. 2026.

DINIZ, H. G. Descrição da variação fonético-fonológica na letra manual “E” em Libras. **Revista Linguíʃtica**, v. 16, n. 3, p. 452-459, 2020. DOI: 10.31513/linguistica.2020.v16n3a37488. Disponível em: https://revistas.ufrj.br/index.php/rl/article/view/37488. Acesso em: 30 out. 2026.

FIELDING, R. T.; NOTTINGHAM, M.; RESCHKE, J. **HTTP semantics**. RFC 9110. [S. l.]: IETF, 2022. DOI: 10.17487/RFC9110. Disponível em: https://www.rfc-editor.org/rfc/rfc9110. Acesso em: 27 out. 2026.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge: MIT Press, 2016.

GOOGLE AI EDGE. **Hand landmarks detection guide**. Google for Developers, 2026. Disponível em: https://developers.google.com/edge/mediapipe/solutions/vision/hand_landmarker. Acesso em: 30 jun. 2026.

GRINBERG, M. **Video streaming with Flask**. 2014. Disponível em: https://blog.miguelgrinberg.com/post/video-streaming-with-flask. Acesso em: 27 out. 2026.

HAAS, A.; ROSSBERG, A.; SCHUFF, D. L.; TITZER, B. L.; HOLMAN, M.; GOHMAN, D.; WAGNER, L.; ZAKAI, A.; BASTIEN, J. F. Bringing the web up to speed with WebAssembly. In: ACM SIGPLAN CONFERENCE ON PROGRAMMING LANGUAGE DESIGN AND IMPLEMENTATION, 38., 2017, Barcelona. **Proceedings** [...]. New York: ACM, 2017. p. 185-200. DOI: 10.1145/3062341.3062363.

HALDER, A.; TAYADE, A. Real-time vernacular sign language recognition using MediaPipe and machine learning. **International Journal of Research Publication and Reviews**, v. 2, n. 5, p. 9-17, 2021. Disponível em: https://ijrpr.com/uploads/V2ISSUE5/IJRPR462.pdf. Acesso em: 30 out. 2026.

LIBRASFLOW. **LibrasFlow**: reconhecimento do alfabeto em LIBRAS em tempo real pela webcam. Jacareí: Instituto Federal de São Paulo, 2026. Disponível em: https://librasflow.ifsp.dev. Acesso em: 19 out. 2026.

LIMA, C. J. de; LIMA, E. G. de O.; SILVA, N. C. da; CORREIA NETO, J. da S. Assistive technology and translation into Libras: challenges of the VLibras video machine translation tool. **Research, Society and Development**, v. 10, n. 12, p. e385101220720, 2021. DOI: 10.33448/rsd-v10i12.20720.

OLIVEIRA, W. **Libras**. Kaggle, 2022. Disponível em: https://www.kaggle.com/datasets/williansoliveira/libras. Acesso em: 12 ago. 2026.

PALLETS PROJECTS. **Flask**: web development, one drop at a time. [S. l.]: Pallets, 2024. Disponível em: https://flask.palletsprojects.com. Acesso em: 30 jun. 2026.

PASSOS, B. T.; FERNANDES, A. M. da R.; COMUNELLO, E. **Brazilian Sign Language Alphabet**. Versão 5. [S. l.]: Mendeley Data, 2020. DOI: 10.17632/k4gs3bmx5k.5. Disponível em: https://data.mendeley.com/datasets/k4gs3bmx5k. Acesso em: 29 out. 2026.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, É. Scikit-learn: machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825-2830, 2011.

PIGOU, L.; DIELEMAN, S.; KINDERMANS, P.-J.; SCHRAUWEN, B. Sign language recognition using convolutional neural networks. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 13., 2014, Zurich. **Computer vision – ECCV 2014 workshops**: proceedings, part I. Cham: Springer, 2015. p. 572-578. DOI: 10.1007/978-3-319-16178-5_40.

QUADROS, R. M. de; KARNOPP, L. B. **Língua de sinais brasileira**: estudos linguísticos. Porto Alegre: Artmed, 2004.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533-536, 1986. DOI: 10.1038/323533a0.

SANTOS, G. A. dos; GONÇALVES, C. H. M. **LibrasFlow**: código-fonte, dados e modelos do sistema de reconhecimento de letras da LIBRAS. GitHub, 2026. Disponível em: https://github.com/SantosGAlves/LibrasFlow. Acesso em: 19 out. 2026.

SMILKOV, D.; THORAT, N.; ASSOGBA, Y.; YUAN, A.; KREEGER, N.; YU, P.; ZHANG, K.; CAI, S.; NIELSEN, E.; SOERGEL, D.; BILESCHI, S.; TERRY, M.; NICHOLSON, C.; GUPTA, S. N.; SIRAJUDDIN, S.; SCULLEY, D.; MONGA, R.; CORRADO, G.; VIÉGAS, F. B.; WATTENBERG, M. TensorFlow.js: machine learning for the web and beyond. **Proceedings of Machine Learning and Systems**, v. 1, p. 309-321, 2019.

SOUSA, C. H. S.; BRAZ JUNIOR, G. Tecnologias assistivas e inteligência artificial para tradução e ensino de Libras. **Revista Contemporânea**, v. 5, n. 2, p. e7594, 2025. DOI: 10.56083/RCV5N2-131.

ZHANG, F.; BAZAREVSKY, V.; VAKUNOV, A.; TKACHENKA, A.; SUNG, G.; CHANG, C.-L.; GRUNDMANN, M. MediaPipe Hands: on-device real-time hand tracking. **arXiv**, 2020. DOI: 10.48550/arXiv.2006.10214. Disponível em: https://arxiv.org/abs/2006.10214. Acesso em: 27 out. 2026. Trabalho apresentado no CVPR Workshop on Computer Vision for Augmented and Virtual Reality, Seattle, 2020.