INTELI - INSTITUTO DE TECNOLOGIA E LIDERANÇA

ALYSSON CARLOS DE CASTRO CORDEIRO

SISTEMA DE TRADUÇÃO SIMULTÂNEA DE LIBRAS BASEADO EM VISÃO COMPUTACIONAL E INTELIGÊNCIA ARTIFICIAL

Uma tecnologia assistiva para a inclusão de estudantes surdos e ouvintes no ensino médio brasileiro

SÃO PAULO

2025

ALYSSON CARLOS DE CASTRO CORDEIRO

SISTEMA DE TRADUÇÃO SIMULTÂNEA DE LIBRAS BASEADO EM VISÃO COMPUTACIONAL E INTELIGÊNCIA ARTIFICIAL

Uma tecnologia assistiva para a inclusão de estudantes surdos e ouvintes no ensino médio brasileiro

Artigo acadêmico apresentado ao Instituto de Tecnologia e Liderança (Inteli)
Orientador: Prof. Guilherme Henrique de Oliveira Cestari

SÃO PAULO
2025

Resumo: A inclusão de estudantes surdos no ensino médio brasileiro enfrenta barreiras comunicacionais significativas devido à escassez de intérpretes e à baixa disseminação da Língua Brasileira de Sinais (Libras). Este trabalho apresenta o desenvolvimento de um sistema de tradução automática de Libras para texto, utilizando Visão Computacional e Inteligência Artificial, com foco na acessibilidade e baixo custo. A metodologia experimental envolveu a criação iterativa de modelos utilizando o framework MediaPipe para extração de características, redes LSTM para gestos dinâmicos e CNNs para sinais estáticos (alfabeto e numerais). Embora os modelos tenham alcançado alta acurácia em validação (acima de 90%), os testes em tempo real revelaram uma latência crítica de 2 a 3 segundos decorrente da janela de processamento temporal, além de sensibilidade a variações de iluminação. Conclui-se que a arquitetura baseada apenas em landmarks e LSTM é insuficiente para uma tradução verdadeiramente simultânea. O estudo propõe, como pivô estratégico, a migração para uma arquitetura híbrida (LRCN) que unifique a análise espaço-temporal, visando futuras aplicações em plataformas educacionais inclusivas.

Palavras-chave: Língua Brasileira de Sinais. Visão Computacional. Inteligência Artificial. Tecnologia Assistiva. Inclusão Escolar.

Abstract: The inclusion of deaf students in Brazilian high schools faces significant communication barriers due to the scarcity of interpreters and the limited dissemination of the Brazilian Sign Language (Libras). This paper presents the development of an automatic Libras-to-text translation system using Computer Vision and Artificial Intelligence, focusing on accessibility and low cost. The experimental methodology involved the iterative creation of models using the MediaPipe framework for feature extraction, LSTM networks for dynamic gestures, and CNNs for static signs (alphabet and numerals). Although the models achieved high validation accuracy (over 90%), real-time tests revealed a critical latency of 2 to 3 seconds due to the temporal processing window, as well as sensitivity to lighting variations. It is concluded that the architecture based solely on landmarks and LSTM is insufficient for truly simultaneous translation. The study proposes, as a strategic pivot, the migration to a hybrid architecture (LRCN) that unifies spatiotemporal analysis, aiming for future applications in inclusive educational platforms.

Key-words: Brazilian Sign Language. Computer Vision. Artificial Intelligence. Assistive Technology. School Inclusion.

SUMÁRIO

1. INTRODUÇÃO

A comunicação é um direito fundamental e uma necessidade básica para a participação social plena. No Brasil, a Língua Brasileira de Sinais (Libras) é reconhecida como a segunda língua oficial do país e constitui o principal meio de comunicação da comunidade surda (BRASIL, 2002). Contudo, a baixa disseminação do conhecimento de Libras entre ouvintes e a escassez de intérpretes qualificados geram barreiras significativas, especialmente em contextos educacionais, como no ensino médio brasileiro. Essas barreiras limitam a inclusão, a autonomia e o acesso equitativo à educação para estudantes surdos, impactando seu desempenho acadêmico e suas oportunidades futuras, como o ingresso em universidades.

Nesse contexto, o desenvolvimento de tecnologias assistivas, como sistemas de tradução automática de Libras para texto ou fala, surge como uma solução promissora para reduzir essas barreiras. Este projeto propõe a criação de um sistema de tradução simultânea baseado em visão computacional e inteligência artificial, voltado especificamente para estudantes do ensino médio, com o objetivo de promover a inclusão social e educacional de estudantes surdos e facilitar a comunicação com professores, colegas ouvintes e profissionais da educação.

Atualmente, a comunicação entre estudantes surdos e ouvintes no ambiente escolar depende majoritariamente de intérpretes humanos, cuja disponibilidade é limitada, especialmente em escolas públicas, onde a presença de intérpretes é insuficiente para atender a demanda. O Brasil possui aproximadamente 2,3 milhões de pessoas com deficiência auditiva severa ou total (IBGE, 2024). Embora estimativas anteriores, que consideravam todos os graus de deficiência auditiva (leve, moderado, severo e profundo), chegassem a mais de 10 milhões de pessoas (AGÊNCIA BRASIL, 2019), o que equivale a aproximadamente 5% da população brasileira, o número de surdos que utilizam a Língua Brasileira de Sinais (Libras) como principal meio de comunicação é uma minoria dentro desse total. Além disso, estima-se que menos de 1% da população ouvinte brasileira tenha proficiência em Libras (SENADO FEDERAL, 2019). Essa realidade resulta em desafios como a evasão escolar, a exclusão social e a dificuldade de acesso a conteúdos educacionais, particularmente no ensino médio, etapa crucial para a preparação para exames de ingresso no ensino superior.

No entanto, embora existam pesquisas na área de reconhecimento de sinais, como o uso de Redes Neurais Convolucionais (CNNs) para gestos estáticos e Redes Neurais Recorrentes (RNNs) para gestos dinâmicos, há lacunas significativas na literatura. Muitos estudos não abordam a complexidade da Libras, que inclui regionalismos,

expressões faciais e nuances culturais essenciais para a comunicação. Além disso, poucas soluções são otimizadas para o contexto educacional brasileiro, especialmente o ensino médio, e muitas não operam em tempo real ou não são financeiramente acessíveis para escolas públicas. Projetos, por exemplo, como o *Hand Talk* oferecem tradução de Libras, mas não são adaptados para performance de comunicação direta sobretudo em ambientes educacionais, limitando sua aplicabilidade em sala de aula.

Por conseguinte, este projeto busca preencher essas lacunas ao desenvolver um sistema de tradução simultânea de Libras que seja acessível e culturalmente sensível, com foco na educação em escolas de ensino secundário. A solução inicial reconhecerá gestos estáticos e dinâmicos (numerais e letras) e, futuramente, expressões faciais, promovendo a inclusão de estudantes surdos e apoiando a comunicação com professores e colegas ouvintes.

Diante desse cenário, a questão central que orienta esta pesquisa, conforme definido no plano de projeto, é formulada da seguinte maneira: *"Como desenvolver um sistema de tradução simultânea da Língua Brasileira de Sinais (Libras) para texto, utilizando visão computacional e inteligência artificial, que seja acessível financeiramente e culturalmente sensível para estudantes surdos do ensino médio brasileiro?"*

Em resposta a essa problemática, a hipótese que este trabalho busca endereçar é a de que é viável desenvolver um sistema de tradução simultânea de Libras para texto, baseado em visão computacional e técnicas de deep learning, que seja capaz de operar em tempo real, ser financeiramente acessível para escolas públicas e culturalmente sensível às complexidades da Libras.

Para auxiliar na investigação desta hipótese e resolução do problema de pesquisa, o objetivo geral do projeto é desenvolver um sistema de tradução simultânea da Língua de Sinais Brasileira (Libras) para texto, utilizando visão computacional e inteligência artificial para estudantes do ensino médio no Brasil, cujos objetivos específicos incluem: desenvolver o modelo de visão computacional com a MediaPipe, um framework que avalia e personaliza modelos e pipelines de machine learning; utilizar LSTM (Long Short-Term Memory) para criar memória artificial dos sinais dinâmicos para tradução em tempo real; garantir privacidade e segurança; e otimizar para escalabilidade e validar o sistema disseminando e conscientizando sobre a comunidade surda. Essa solução visa, inicialmente, capturar os sinais de letras do alfabeto da língua portuguesa, numerais naturais de 1 ao 9, interpretando-os por meio de algoritmos de reconhecimento de padrões e traduzindo-os. Portanto, isso permitirá o início da resolução do problema e também quebrará barreiras de comunicação, promovendo a inclusão social, facilitando interações cotidianas, educacionais e

profissionais, e acima de tudo ajudará a ampliar a autonomia da comunidade surda.

Vale ressaltar que este artigo está estruturado da seguinte forma: A Seção 1 aborda a introdução, visão geral da problemática e da possível solução. A Seção 2 revisa a fundamentação teórica. Na Seção 3, o estado da arte e os trabalhos relacionados, identificando as lacunas que esta pesquisa visa preencher. A Seção 4 detalha a metodologia de pesquisa e a evolução experimental iterativa do sistema. A Seção 5 apresenta e analisa os resultados empíricos de múltiplas iterações de modelagem. A Seção 6 discute as limitações arquitetônicas fundamentais da abordagem atual e propõe um pivô para uma arquitetura híbrida superior. A Seção 7 conclui o trabalho, resumindo as contribuições, resultados esperados e o impacto social.

2. FUNDAMENTAÇÃO TEÓRICA

O projeto de tradução automática da Língua Brasileira de Sinais (Libras) para o Português utilizando visão computacional e inteligência artificial (IA) é um campo interdisciplinar que combina linguística, visão computacional, IA e, acima de tudo, inclusão social. Essas fundamentações são essenciais para sustentar o desenvolvimento do sistema proposto, garantindo que seja tecnicamente viável, culturalmente sensível e socialmente impactante.

Primeiramente, é importante reconhecer que a Libras é uma língua visual-espacial com estrutura gramatical própria e características únicas que a distinguem das línguas faladas (MODOBILINGUE, 2022). Diferente do Português (uma língua oral-auditiva), a Libras utiliza sinais manuais, expressões faciais e movimentos corporais para transmitir significado. Sua gramática inclui elementos como configuração das mãos, pontos de articulação, padrões de movimento, orientação da palma da mão e marcadores não manuais (exemplos: expressões faciais e movimentos de cabeça), que são fundamentais para a comunicação em Libras e devem ser considerados no desenvolvimento de um sistema de tradução automática (QUADROS; KARNOPP, 2004; HAND TALK, 2024). Além disso, a Libras apresenta variações regionais, o que significa que o mesmo sinal pode ter significados diferentes em outras partes do Brasil. Inicialmente, o sistema reconhecerá sinais básicos de Libras, mas, através do desenvolvimento iterativo, alcançará progressivamente a tradução precisa de expressões e variações regionais, tornando-se culturalmente sensível.

Por outro lado, a visão computacional — um campo da ciência da computação que permite às máquinas "ver", reconhecer e interpretar imagens e vídeos — será utilizada para capturar e processar gestos manuais que compõem os sinais de Libras. Técnicas como filtragem, segmentação e detecção de bordas melhoram a qualidade da imagem

capturada, facilitando o reconhecimento do sinal. As Redes Neurais Convolucionais (CNNs) são particularmente eficazes para reconhecer padrões visuais (exemplo: formas das mãos e direções de movimento), tornando-as ideais para o reconhecimento de gestos estáticos e dinâmicos essenciais para a tradução automática da Libras (CALIMAN, 2024; L, 2019; REZENDE, 2016).

Vale lembrar que a inteligência artificial, especialmente o deep learning (aprendizado profundo), desempenha um papel crucial no reconhecimento e tradução da língua de sinais. Modelos como as Redes Neurais Recorrentes (RNNs) e os Transformers se destacam na captura de sequências temporais, como por exemplo em gestos de sinais dinâmicos que envolvem movimento ao longo do tempo. As RNNs são particularmente úteis para processar dados sequenciais, como para vídeos de língua de sinais, enquanto os Transformers — modelos avançados — provaram ser eficientes para tarefas de tradução e Processamento de Linguagem Natural (PLN) (DA, 2025). Esses modelos podem ser treinados em grandes conjuntos de dados para reconhecer e traduzir sinais com alta precisão. As técnicas de PLN podem então traduzir os sinais capturados em texto ou fala, permitindo que o sistema atenda também a indivíduos ouvintes não familiarizados com a Libras.

Como uma tecnologia assistiva — ferramentas que ajudam pessoas com deficiência a superar limitações e participar plenamente da sociedade — este sistema de tradução de Libras para Português proposto exemplifica a tecnologia que pode promover acessibilidade e inclusão, particularmente nas escolas secundárias brasileiras. Ao facilitar a comunicação entre indivíduos surdos e ouvintes, o sistema pode melhorar o acesso a serviços essenciais além da educação, como na saúde e emprego, reduzindo as barreiras que limitam a participação social plena para indivíduos surdos.

3. ESTADO DA ARTE

O estado da arte para projetos de tradução automática da Língua Brasileira de Sinais (Libras) tem avançado significativamente nos últimos anos. Esse progresso é impulsionado por desenvolvimentos em técnicas de visão computacional, deep learning e outros métodos científicos, bem como por projetos desenvolvidos por diversos pesquisadores. No entanto, apesar desses avanços, desafios significativos permanecem, principalmente em termos de precisão, tempo de resposta e inclusão de nuances culturais e regionais. Esta seção revisa as pesquisas e soluções existentes mais avançadas, identificando lacunas e destacando como este projeto pretende contribuir para a área.

O desenvolvimento de sistemas de tradução de Libras depende, sobretudo, fortemente de técnicas de visão computacional para capturar e interpretar gestos e expressões

faciais. Redes Neurais Convolucionais (CNNs), por exemplo, são amplamente utilizadas para reconhecer gestos estáticos porque sua arquitetura é altamente eficaz na análise de dados visuais, as quais empregam camadas convolucionais para detectar automaticamente padrões locais, como as formas e configurações das mãos. Outrossim, o uso do compartilhamento de pesos nesses filtros permite que a rede reconheça um gesto independentemente de sua posição exata na imagem (invariância de translação). Além disso, as camadas de pooling (agrupamento) reduzem a dimensionalidade dos dados, tornando o modelo robusto a pequenas variações na pose ou na iluminação, e a extração hierárquica de características permite que a rede aprenda progressivamente, desde traços simples até configurações complexas das mãos que definem um sinal específico, como os sinais que representam as letras alfabéticas da datilologia formadas sem movimento adicional e também os números naturais.

Contudo, a Libras inclui movimentos dinâmicos e expressões faciais que são essenciais para a comunicação. Para lidar com essas complexidades têm sido empregadas as Redes Neurais Recorrentes (RNNs) que, juntamente com variações avançadas como LSTMs (Long Short-Term Memory) e Transformers, são empregadas no reconhecimento de gestos dinâmicos em Libras porque foram projetadas especificamente para processar dados sequenciais e temporais. Diferente dos gestos estáticos, os sinais dinâmicos envolvem movimento contínuo, e as RNNs conseguem capturar a sequência e a progressão dos quadros de vídeo ao longo do tempo, aprendendo a relação temporal do movimento. Elas possuem, ademais, uma espécie de "memória" que lhes permite armazenar e utilizar informações de eventos passados, o que é crucial para entender o contexto de um gesto ou frase. Além disso, essas redes podem processar sequências de comprimento variável e são integradas em modelos híbridos com CNNs, onde as CNNs extraem as características visuais de frames individuais e as RNNs sequenciam essas características para interpretar o fluxo temporal e o significado gramatical da língua de sinais.

Adicionalmente, ferramentas de frameworks como o Google MediaPipe (GOOGLE, 2023) e OpenCV (BRADSKI, 2000) são utilizadas para detectar pontos-chave das mãos e do rosto, fornecendo dados de entrada para os modelos de IA. O MediaPipe, em particular, destaca-se por sua eficiência em tempo real, permitindo a detecção de gestos e expressões faciais com baixa latência, sendo crucial para projetos que exigem processamento em tempo real.

Nesse contexto, diversas soluções foram desenvolvidas para a tradução automática de língua de sinais, tanto em contextos acadêmicos quanto comerciais. Projetos acadêmicos como Sign Language Detection (AVASTHI, 2021) e Sign Language Interpreter using Deep Learning (GUPTA, 2021) demonstram a eficácia de CNNs e

RNNs no reconhecimento de gestos estáticos e dinâmicos, com o primeiro alcançando taxas de precisão acima de 95% em tarefas de reconhecimento. Entretanto, eles ainda enfrentam desafios, como a falta de suporte para variações regionais e expressões faciais, além de limitações relacionadas à idade, tamanho e estilo de sinalização do usuário.

Na esfera comercial, aplicativos como o Hand Talk (HAND TALK, 2024) oferecem tradução de Libras para Português, mas possuem limitações no reconhecimento de gestos dinâmicos e expressões faciais complexas, e não são otimizados para o contexto educacional brasileiro especificamente.

Por esse motivo, apesar dos avanços, lacunas significativas permanecem. Uma das principais é a falta de suporte para variações regionais da Libras que impactam diretamente na precisão do sistema. Outro desafio é a latência, pois muitas soluções não operam em tempo real, e também a velocidade das respostas do software, limitando sua usabilidade em salas de aula. Por fim, a maioria das soluções existentes é genérica e não foi desenvolvida especificamente para as necessidades de estudantes surdos e estudantes ouvintes no ensino médio brasileiro, o que inclui a ausência de integração com materiais didáticos e plataformas educacionais.

Considerando as lacunas discutidas, este artigo prioriza: a latência e a velocidade de resposta do sistema e aplicação do desenvolvimento voltado para estudantes do ensino secundário, propondo o protótipo inicial de um sistema acessível de tradução automática de Libras para texto com foco no sistema educacional brasileiro. A médio e longo prazo, ao longo do desenvolvimento futuro do sistema proposto, o projeto visa preencher as lacunas de expressões faciais, aprendizado de expressões e de variações regionais e fala automática dos sinais.

3.1. ANÁLISE DE BENCHMARK DE SOLUÇÕES EXISTENTES

Para endereçar empiricamente as lacunas identificadas, uma análise de benchmark de seis projetos de código aberto relevantes foi realizada, os quais foram avaliados com base na tecnologia, precisão, tempo de resposta, acessibilidade, cobertura linguística e pontos fortes. Os resultados estão resumidos na seguinte tabela 1.

Tabela 1: Análise Comparativa de Projetos de Reconhecimento de Sinais

Criterion	Sign-Language-Detection (CNN)	Sign Language Interpreter (Deep Learning)	ASL ↔ English Translation	Convolutional Neural Networks - LIBRAS	Libras Reader	RAS-Libras
Technology	CNN + Kaggle dataset	TensorFlow + Keras + OpenCV	MediaPipe + PointNet + ThreeJS	CNN + OpenCV	TensorFlow + Teachable Machine + MediaPipe	TensorFlow + OpenCV + PyQt5
Accuracy	High (99%)	Good (95%)	Variable (CNNs: 95% for static gestures; RNNs: 85-90% for sentences; Transformers: 80-95% for complex contexts)	High (dataset-dependent)	Good (dataset-dependent)	Good (dataset-dependent)
Response Time	Real-time	Real-time	Moderate (simple gestures: <1s; dynamic gestures: 1-3s; complex gestures: 3-5s+)	Real-time	Real-time	Real-time
Accessibility & Cost	Free, open-source	Free, open-source	Free, requires powerful hardware	Free, open-source	Free, open-source	Free, open-source
Linguistic Coverage	Basic ISL gestures	44 ASL characters	ASL with facial expressions	Libras alphabet	Numbers in Libras	Libras alphabet
Key Strengths	High accuracy for static gestures	Real-time recognition	ASL ↔ English translation	Focus on Libras alphabet	Robotics integration	Interactive Libras learning

Fonte: tabela elaborada pelo autor Alysson Carlos de Castro Cordeiro (2025).

A análise dos dados da tabela confirma as lacunas da literatura. Projetos de alta precisão, como o "Sign-Language-Detection (CNN)" (99%), focam exclusivamente em gestos estáticos. Projetos que abordam Libras, como "Convolutional Neural Networks - LIBRAS" e "RAS-Libras", focam primariamente no alfabeto estático ou em números. Além disso, o projeto "ASL ↔ English Translation" é um dos poucos a considerar expressões faciais, mas foca no ASL e exige hardware poderoso, limitando sua acessibilidade. Nesse contexto, nenhum dos projetos analisados combina com sucesso o foco na Libras, incluindo sua complexidade dinâmica, o contexto educacional brasileiro e a acessibilidade de baixo custo.

4. DESENVOLVIMENTO E METODOLOGIA EXPERIMENTAL

O desenvolvimento e a metodologia experimental deste projeto foram conduzidos de forma interativa. Inicialmente, o trabalho concentrou-se na letra "H" — a qual é feita com a mão fechada, mantendo o indicador e o polegar esticados, enquanto os outros dedos ficam recolhidos. Em seguida, a palma da mão fica virada para a frente e a mão

gira ligeiramente para dentro — para isso, foi desenvolvido um modelo binário utilizando uma arquitetura LSTM (Long Short-Term Memory) com 64 unidades, dropout de 0.3 e processamento de sequências de 30 frames contendo 63 features extraídas dos landmarks da mão via MediaPipe. O conjunto de dados inicial compreendia 200 vídeos de "H" e 200 de "não-H" (gestos e sinais aleatórios), resultando em uma acurácia de treinamento de 97,50% e de validação de 95,00% na última época, com perda de treinamento de 0,0500 e perda de validação de 0,0800. Esses resultados indicaram eficácia na identificação do gesto rotacional em condições controladas, atribuída à qualidade dos dados e à capacidade do modelo de capturar padrões temporais.

Posteriormente, buscou-se expandir o sistema para incluir a letra "K" e também outro fator correspondente, o “não-K” para gestos e sinais aleatórios, caracterizada por um movimento ascendente com os mesmos dedos estendidos. Entretanto, identificou-se uma limitação crítica no modelo binário: gestos de "H" eram classificados como "não-K", uma vez que o modelo tratava qualquer gesto fora da classe positiva como negativo, incluindo sinais específicos de outras letras. Essa restrição evidenciou a necessidade de transição para um modelo multiclasse capaz de distinguir simultaneamente "H", "K" e "outros" (gestos genéricos que não pertencem às duas classes anteriores). Para implementar essa expansão, foram reorganizados os dados e scripts (collect_data.py, preprocess_data.py, train_lstm.py e test_lstm.py), definindo a estrutura de pastas como “dataset/H_letter/h” para vídeos de "H", “dataset/K_letter/k” para "K" e dataset/outros para gestos aleatórios.

A primeira coleta de dados multiclasse incluiu 407 vídeos de "H", 405 de "K" e 713 de "outros". Já no pré-processamento rejeitou 42 vídeos por inconsistências no número de frames (variando entre 31 e 108 para "K", por exemplo), resultando em 1483 sequências válidas com shape (1483, 30, 63): 394 para "H" (13 rejeitados, shapes entre 33 e 55 frames), 376 para "K" (29 rejeitados) e 713 para "outros" (2 rejeitados, shapes de 40 e 45 frames). O treinamento inicial com arquitetura LSTM de 64 unidades e dropout de 0.6 mostrou, na época 39, perda de 0.3347, acurácia de 0.8609, perda de validação de 0.5311 e acurácia de validação de 0.8418; na época 40, perda de 0.8662, acurácia de 0.6551, perda de validação de 0.4349 e acurácia de validação de 0.9798. A alta acurácia de validação (97,98%) contrastando com a baixa acurácia de treinamento (65,51%) indicou overfitting grave, com o modelo memorizando o conjunto de validação em detrimento da generalização. Em seguida, testes rápidos em tempo real revelaram confusão entre "H" e "K", com classificações alternantes durante movimentos e sensibilidade a gestos estáticos erroneamente identificados como "K".

Devido às inconsistências no número de frames e ao overfitting, procedeu-se a uma

nova coleta de dados para garantir uniformidade e diversidade. Foram coletados 691 vídeos de "H" com variações em velocidade (lenta, média, rápida), ângulo (frontal, lateral, superior) e iluminação (forte, fraca, média); 690 vídeos de "K" com movimentos ascendentes e estáticos variados; e mantidos 713 vídeos de "outros" da coleta anterior. O pré-processamento resultou em 2041 sequências válidas com shape (2041, 30, 63): 675 para "H" (16 rejeitados), 658 para "K" (32 rejeitados) e 711 para "outros" (2 rejeitados). Ajustes no script “collect_data.py” incluíram aumento da confiança mínima de detecção para 0.7 e validação da presença contínua da mão por 30 frames antes da gravação.

Então, por conseguinte, o modelo multiclasse foi treinado novamente com uma arquitetura otimizada: duas camadas LSTM (256 e 128 unidades), dropout reduzido para 0,3, camada densa com 64 unidades (ReLU) e Batch Normalization (Normalização em Lote), que é uma técnica de regularização introduzida por Ioffe e Szegedy (2015) para acelerar o treinamento, melhorar a estabilidade e reduzir problemas como o *vanishing/exploding gradients* e o *internal covariate shift*. A rede utilizou uma camada de saída com 3 classes (softmax), 100 épocas com early stopping (patience=10) e normalização z-score por amostra para melhorar a generalização. Consequentemente, os resultados na época 21 (parada precoce) foram perda de 0,0909, acurácia de 0,9755, perda de validação de 0,3913 e acurácia de validação de 0,9022, representando um avanço significativo em relação à iteração anterior, graças à recoleta diversificada e aos ajustes arquiteturais que equilibraram as classes ("H" em 33,1%, "K" em 32,2% e "outros" em 38,3%) e reduziram o viés.

Testes em tempo real com o script “test_lstm.py” demonstraram alta precisão subjetiva na identificação de "H" durante rotações, "K" em movimentos ascendentes e "outros" em gestos aleatórios, em dois ambientes distintos. Contudo, observou-se um pequeno atraso (~2-3 segundos) na transição entre classes, atribuído à janela de 30 frames (3 segundos a 10 fps). Por esse motivo, uma tentativa de reduzir para 15 frames gerou erro de incompatibilidade de shape (expected shape=(None, 30, 63), found shape=(None, 15, 63)), destacando a necessidade de retreinamento com “n_frames=15” para otimizar a latência, ajustando todos os scripts correspondentes.

As causas prováveis dos desafios iniciais incluíram overfitting devido ao tamanho reduzido do conjunto de validação (~408 amostras) e dropout excessivo (0.6), desbalanceamento de classes com predominância de "outros", dados insuficientes em variações ambientais (iluminação, ângulo, distância ideal de 30-50 cm) e inconsistências de frames causadas por interrupções na captura ou falhas no MediaPipe. Esses problemas foram mitigados pela recoleta, redução de dropout,

adição de BatchNormalization e early stopping, elevando a robustez do modelo.

Vale ressaltar, portanto, que nos últimos ajustes foram coletados dados adicionais para variações extremas (100-200 vídeos por classe), planejou-se a implementação de suavização de predições (média móvel) para uso em tempo real, e preparou-se a expansão para novas letras estáticas (ex.: "A", "B") via CNN integrada ao multiclasse. A validação com a comunidade surda foi priorizada para garantir precisão cultural e identificação de variações regionais. Em resumo, essa metodologia experimental evoluiu de um modelo binário promissor para um sistema multiclasse robusto, com acurácia de validação de 90,22%, posicionando o projeto para contribuições efetivas.

4.1. DESENVOLVIMENTO PARA OS SINAIS ESTÁTICOS E A COMBINAÇÃO DOS FATORES DESENVOLVIDOS.

Outrossim, o desenvolvimento dos sinais estáticos representou uma expansão natural do projeto, visando complementar os gestos dinâmicos já implementados, como as letras "H" e "K", com uma abordagem específica para letras e números que não envolvem movimentos temporais, a exemplo das letras "A" e "B" e dos números de 1 a 9. Inicialmente, a coleta de dados para esses sinais foi realizada por meio de um script dedicado, denominado “collect_static_data.py”, que capturava imagens individuais em vez de sequências de frames, considerando a natureza estática desses gestos. Para cada classe, foram coletadas 1600 imagens, distribuídas entre as letras "A", "B" e a classe "outros" (gestos não relacionados), e para os números de 1 a 9 foram 1200 imagens, totalizando 10.800. Além disso, a coleta foi automatizada garantindo que as imagens fossem salvas em formato JPG no diretório “dataset_static”, com nomes únicos gerados via UUID para evitar duplicidades. Essa estratégia permitiu uma variação nos dados, incorporando diferentes ângulos, distâncias (de 30 a 50 cm da câmera) e condições de iluminação, o que contribuiu para a robustez do modelo.

Em seguida, o treinamento para os sinais estáticos foi conduzido utilizando uma rede neural convolucional (CNN), implementada no script “train_cnn.py”, adequada para processar imagens 2D em escala de cinza com dimensões de 64x64 pixels. É importante salientar que a arquitetura da CNN consistiu em três camadas convolucionais (32, 64 e 128 filtros, respectivamente), seguidas de pooling máximo para redução dimensional, uma camada de achatamento, uma camada densa de 128 neurônios com ativação ReLU, dropout de 0.5 para mitigar overfitting, e uma camada de saída com softmax para 13 classes (letras "A" e "B", números de 1 a 9, e "outros"). O treinamento empregou o ImageDataGenerator do Keras para augmentação de dados, aplicando rotações de até 10 graus, deslocamentos horizontais e verticais de 0.1, distorção de 0.1 e zoom de 0.1, além de normalização de pixels para o intervalo [0,

1]. Outro fator importante é que o dataset foi dividido em 80% para treinamento e 20% para validação, com batch size de 32 e 50 épocas, resultando no modelo salvo como “cnn_static_model.keras”. Por conseguinte, a acurácia de validação alcançou valores acima de 85% nas últimas épocas, com destaque para a época 38/50, na qual a perda de treinamento foi de 0.5237, a acurácia de treinamento atingiu 0.8081, a perda de validação foi de 0.4953 e a acurácia de validação chegou a 0.8603, demonstrando eficácia na classificação de gestos estáticos, embora com oscilações em condições de iluminação variadas.

4.2. A COMBINAÇÃO DOS FATORES ESTÁTICOS E DOS DINÂMICOS DESENVOLVIDOS.

A combinação dos fatores desenvolvidos para sinais estáticos e dinâmicos foi realizada e executada com o script “test_combined.py”, que integra os modelos LSTM (para "H", "K" e "outros" dinâmicos) e CNN (para "A", "B", números de 1 a 9, e "outros" estáticos). A lógica de decisão prioriza a predição com maior confiança acima de 80%, com o CNN assumindo a liderança para gestos estáticos e o LSTM para dinâmicos, resolvendo conflitos com base na probabilidade máxima. No pré-teste, por exemplo, com um determinado participante 1, o sistema mostrou precisão razoável: a letra "A" oscilou entre "A" e "B" com confiança de 0,98, melhorando em distâncias menores; a letra "B" foi reconhecida consistentemente como "B" em diversas distâncias; a letra "H" alcançou precisão de 0,89 a 0,98, mesmo em posições variadas; a letra "K" oscilou para "B" com 0,95; os números de 1 a 9 apresentaram confusões (ex.: 1 como "B", 5 como "outros", 9 como "outros" ou "9"), dependentes de luz e distância, com o sistema funcionando melhor a 30-50 cm em luz artificial branca. A integração revelou que o CNN é mais sensível a variações de luz, oscilando entre classes, enquanto o LSTM mantém estabilidade em movimentos dinâmicos, embora com atraso na transição entre gestos.

Portanto, essa abordagem combinada permitiu uma avaliação inicial da usabilidade do sistema, destacando a necessidade de mais dados para reduzir oscilações em gestos estáticos e melhorar a detecção em condições reais. Por esse motivo, a expansão para números de 1 a 9 seguiu a mesma metodologia, com 1200 imagens por número, treinadas na CNN, ampliando o escopo do projeto para elementos numéricos do alfabeto Libras.

5. RESULTADOS E ANÁLISE

A fase inicial do projeto concentrou-se no desenvolvimento de um modelo binário para reconhecer o gesto da letra "H", utilizando a rotação da mão como característica principal. O treinamento inicial com 200 vídeos resultou em uma acurácia de

treinamento e validação de 100% nas épocas finais, sugerindo um overfitting significativo devido ao pequeno volume do conjunto de dados. Posteriormente, o modelo foi aprimorado com uma arquitetura LSTM e dropout de 0.3, utilizando 600 vídeos no total. Esse treinamento demonstrou uma performance robusta, atingindo 98.75% de acurácia de treino e 96.67% de acurácia de validação, indicando melhor generalização sob condições controladas.

Com a expansão para a letra "K", que envolve um movimento ascendente, identificou-se a limitação do modelo binário em distinguir corretamente entre "H" e "K", o que tornou necessária a transição para um modelo multiclasse. Durante a análise dos dados existentes, aproximadamente 33% dos vídeos foram rejeitados devido a inconsistências no número de frames ou falhas na detecção pelo MediaPipe. Uma nova coleta de dados foi realizada, resultando em 1163 sequências válidas, incluindo 300 vídeos para "H", 303 para "K" e 600 para gestos genéricos ("outros"), ajustando o script de coleta para garantir maior confiança na detecção e consistência de frames.

Por outro lado, o treinamento do modelo multiclasse, configurado com 64 unidades LSTM e dropout de 0.6, apresentou acurácia de treinamento de 60.86% e acurácia de validação de 95.71% na última época. A discrepância entre as acurácias confirmou a persistência do overfitting, onde o modelo memorizou o pequeno conjunto de validação (~232 amostras) em vez de aprender padrões gerais. As principais causas diagnosticadas incluíram o desbalanceamento dos dados, com a classe "outros" dominando o conjunto de dados (48.2%), e a falta de diversidade nos dados coletados, que limitou a capacidade de generalização do modelo em cenários reais.

Tabela 2: Desbalanceamento do Conjunto de Dados

Classe	**Quantidade/Vídeos**	**Porcentagem**
H	300	25.8%
K	303	26.0%
Others	600	48.2%

Fonte: Elaborada por Alysson Carlos de Castro Cordeiro (2025)

Os testes em tempo real revelaram desafios significativos na usabilidade e robustez do sistema. Observou-se uma confusão frequente entre os sinais de "H" e "K", com o

modelo alternando rapidamente entre os rótulos, e a classificação incorreta de gestos aleatórios como "K", sugerindo um viés do modelo. A performance do sistema demonstrou alta sensibilidade a condições externas, como variações de iluminação e distância, funcionando de maneira ideal apenas entre 30-50 cm da câmera.

A performance inicial do sistema, portanto, demonstrou acurácia de reconhecimento significativamente baixa e instável na prática. Paralelamente, testes de usabilidade com oito participantes, com perfis variados de conhecimento em Libras, indicaram que, embora a velocidade de resposta fosse considerada rápida e eficiente, a acurácia real do sistema que se apresenta foi insuficiente. A maioria dos usuários, sem experiência prévia em Libras, avaliou positivamente a acurácia percebida, um resultado que pode ser atribuído à sua inexperiência em julgar a precisão técnica dos sinais. Gestos estáticos bem definidos, como 'A', 'B' e '6', foram considerados fáceis de executar e reconhecer, enquanto sinais similares ou dinâmicos apresentaram maior dificuldade.

Ademais, o teste revelou um outro novo desafio fundamental: "um pequeno atraso na transição entre 'H' e 'K'". Esse atraso foi estimado em 2-3 segundos. A causa foi diagnosticada como a "janela de 30 frames" utilizada para o treinamento do modelo. Isso expõe um *trade-off* crítico: o modelo precisa *ver* 30 frames de dados para acumular informação temporal suficiente para fazer uma previsão precisa. No entanto, capturar 30 frames (a 10 fps, por exemplo) leva 3 segundos, introduzindo uma latência que é inaceitável para uma "tradução simultânea" e viola um dos objetivos específicos ("baixa latência e alta eficiência"). Uma tentativa de mitigar isso reduzindo a janela de teste para 15 frames resultou em um erro de incompatibilidade de *shape*, pois o modelo foi treinado e espera uma entrada de 30 frames. Isso prova que o modelo precisa ser retreinado com uma janela temporal menor.

Outrossim, um insight crucial emergido dos testes foi a dificuldade motora dos usuários iniciantes em executar os gestos corretamente, gerando sugestões unânimes para a inclusão de ferramentas de aprendizado, como tutoriais ou feedback visual na tela. Esses resultados técnicos deficientes e os insights dos usuários reforçaram a conclusão de que o modelo atual é insuficiente para um caso de uso funcional de tradução simultânea e justificaram o pivô estratégico da pesquisa para uma plataforma de aprendizagem de Libras, priorizando a acurácia do modelo para fins educacionais e inclusivos. Para aprimorar o sistema, os próximos passos propostos incluem a coleta de dados mais diversificados, ajustes nos parâmetros do modelo como a redução do dropout para 0.3, e a validação futura com a comunidade surda.

6. DISCUSSÃO: DESAFIOS ARQUITETURAIS E FUTURO DO PROJETO

Os resultados, embora bem-sucedidos em termos de acurácia, expuseram limitações

na arquitetura LSTM, especialmente ao considerar a latência e a expansão futura para gestos estáticos. Uma análise arquitetural aprofundada tornou-se necessária para definir o futuro do projeto. O plano original do projeto e os próximos passos propostos sugeriam o uso de modelos CNN ou MLP separados para reconhecer gestos estáticos, como as letras 'A', 'B', 'C', enquanto se usava o modelo LSTM para gestos dinâmicos, como 'H', 'K'. Uma análise crítica argumenta que esta abordagem de múltiplos modelos é subótima, e tentar usar a arquitetura LSTM atual para gestos estáticos é "fundamentalmente inadequada" e "conceitualmente falha". O racional é o seguinte: um gesto estático (ex: 'A') capturado por 30 frames (para ser compatível com o modelo dinâmico produz uma sequência de landmarks do MediaPipe quase idênticos e redundantes. Consequentemente, alimentar esta sequência estática a uma LSTM, que é uma arquitetura explicitamente projetada para encontrar mudanças em sequências temporais, é "computacionalmente ineficiente". O modelo seria forçado a aprender que "a falta de mudança" nas coordenadas equivale ao rótulo 'A', o que é um método de classificação indireto e ineficiente.

Uma solução "superior" e de estado da arte para este problema é uma arquitetura híbrida, frequentemente chamada de Long-term Recurrent Convolutional Network (LRCN). Esta abordagem unifica os dois componentes (CNN e LSTM) em um único modelo end-to-end que processa frames de vídeo brutos (sequências de imagens) em vez de landmarks pré-processados. A Tabela 4 abaixo, extraída da análise arquitetural, compara as três abordagens e justifica este pivô estratégico. A arquitetura atual (LSTM-Only) e a alternativa ingênua (CNN-Only) têm "Baixa" adequação para um sistema unificado. Apenas a arquitetura Híbrida CNN-LSTM é classificada como "Alta", pois "combina análise espacial e temporal" e "Fornece um framework único e elegante para gestos estáticos e dinâmicos".

Tabela 4: Análise Comparativa de Arquiteturas de Reconhecimento de Gestos

Critério	**LSTM-Only (on Landmarks)**	**CNN-Only (on Images)**	**Hybrid CNN-LSTM (on Video)**
Caso de Uso Primário	Modelagem de sequência dinâmica (movimento)	Classificação de imagem estática (forma)	Análise espaço-temporal (vídeo)
Tipo de Dado de Entrada	Sequência de vetores de coordenadas (MediaPipe)	Tensor de imagem 2D/3D (pixels)	Sequência de tensores de imagem (frames)

Forças Chave	Captura dependências temporais	Alta precisão na extração de features espaciais	*Combina análise espacial e temporal*
Limitações Inerentes	Fraco na interpretação de padrões espaciais puros	Não pode modelar informação temporal	Mais intensivo computacionalmente
Adequação para Rec. Unificado	**Baixa.** Ineficiente e indireto para sinais estáticos.	**Baixa.** Não pode lidar com sinais dinâmicos.	**Alta.** Fornece um framework único e elegante.

Fonte: Elaborada por Alysson Carlos de Castro Cordeiro (2025)

O modelo LRCN funciona em dois estágios, resolvendo o dilema do projeto. O primeiro é o Extrator de Features Espaciais (CNN): a porção CNN do modelo atua como os "olhos", processando cada frame individual do vídeo e extraindo um vetor de características — um resumo numérico denso que captura a informação espacial (forma da mão, orientação) naquele instante. O segundo é o Modelador Temporal (LSTM): a porção LSTM atua como o "cérebro", recebendo a sequência de vetores de características (um de cada frame) da CNN e analisando como essas características mudam ao longo do tempo. Este framework unificado resolve elegantemente o problema de classificação: para um sinal estático ('A'), o signatário mantém a mão parada; a CNN produzirá uma sequência de vetores de características estáveis e quase idênticos, e a LSTM aprende a classificar esse padrão de "estabilidade de características" como 'A'. Para um sinal dinâmico ('K'), o signatário move a mão; a CNN produzirá uma sequência de vetores de características variáveis, e a LSTM aprende a classificar essa "evolução temporal específica" como 'K'. Dessa forma, o modelo se torna agnóstico; ele simplesmente classifica padrões em sequências de características, sejam elas estáveis ou dinâmicas.

Esta análise arquitetural fornece um roadmap de implementação prático e superior, que substitui os próximos passos de curto prazo focados em 15 frames. O futuro do projeto reside na migração para esta nova arquitetura, que envolve a reengenharia do pipeline de dados (modificando “collect_data.py” e “preprocess_data.py” para salvar e processar sequências de frames de imagem), a construção do modelo LRCN em Keras/TensorFlow utilizando transfer learning (ex: MobileNetV2) na base CNN, e a adaptação dos scripts de treinamento e inferência. Esta abordagem resolve simultaneamente o problema da latência ao retreinar com uma janela menor, como 15

frames e o problema da arquitetura ao unificar gestos estáticos e dinâmicos. Portanto, o passo crucial de validação com a comunidade surda permanece prioritário para garantir a precisão cultural e a identificação correta de variações regionais.

7. CONCLUSÃO

O projeto em pesquisa e desenvolvimento teve como objetivo inicial desenvolver um sistema de reconhecimento e tradução de Libras para português em tempo real, utilizando técnicas de visão computacional e aprendizado profundo, com foco na acessibilidade na educação secundária brasileira. Embora os modelos binários e multiclasse iniciais tenham atingido acurácias promissoras em ambientes de validação controlados (superiores a 90% em alguns casos), a análise aprofundada dos resultados práticos e dos testes de usabilidade revelou limitações críticas. A latência de 2 a 3 segundos, a alta sensibilidade a variações de iluminação e distância, e a dificuldade de generalização para cenários reais demonstraram que o modelo atual era insuficiente para o objetivo de "tradução simultânea".

A partir desses resultados, a pesquisa diagnosticou a inadequação da arquitetura LSTM-apenas para um sistema unificado de gestos estáticos e dinâmicos, justificando um pivô estratégico do projeto. A conclusão técnica aponta para a necessidade de migração para uma arquitetura híbrida Long-term Recurrent Convolutional Network (LRCN), que processa frames brutos e unifica a análise espacial (CNN) e temporal (LSTM) em um único modelo. Esta abordagem mitiga os problemas de latência e ineficiência computacional identificados.

Além das considerações técnicas, o projeto destaca impactos sociais significativos. O desenvolvimento de tal tecnologia tem o potencial de melhorar o acesso à educação para estudantes surdos, reduzindo as barreiras linguísticas impostas pela escassez de intérpretes qualificados no Brasil. Em ambientes profissionais, o sistema poderia facilitar a comunicação diária, promovendo a inclusão no mercado de trabalho e valorizando a Libras como a segunda língua oficial do país. No entanto, a discussão de impactos negativos pondera sobre os riscos da dependência excessiva da tecnologia, a acurácia do sistema em capturar nuances regionais e expressões faciais, e o custo de implementação, que pode limitar o acesso em regiões menos favorecidas.

Em considerações finais, este trabalho conclui que, embora o objetivo inicial de um sistema de tradução simultânea e de baixa latência não tenha sido alcançado com a arquitetura atual, o processo de pesquisa forneceu insights valiosos que direcionam o futuro do projeto para uma plataforma de aprendizagem de Libras mais robusta e inclusiva. O roadmap futuro inclui a implementação da arquitetura LRCN e a validação contínua com a comunidade surda, garantindo que a tecnologia atenda às reais

necessidades de comunicação e acessibilidade, promovendo uma inclusão ética e eficaz na sociedade brasileira.

8. REFERÊNCIAS

AGÊNCIA BRASIL. País tem 10,7 milhões de pessoas com deficiência auditiva, diz estudo. **EBC**, Brasília, DF, 13 out. 2019. Disponível em: https://agenciabrasil.ebc.com.br/geral/noticia/2019-10/brasil-tem-107-milhoes-de-deficientes-auditivos-diz-estudo. Acesso em: 11 nov. 2025.

AVASTHI, Somyansh. **Sign Language Detection using CNN Architecture**. GitHub, 2021. Disponível em: https://github.com/SomyanshAvasthi/Sign-Language-Detection-using-CNN-Architecture. Acesso em: 30 abr. 2025.

BORGES, Lucas Alves. **Convolutional Neural Networks - LIBRAS**. GitHub, 2020. Disponível em: https://github.com/lucaaslb/cnn-libras. Acesso em: 30 abr. 2025.

BRADSKI, Gary. **OpenCV Library**. Open Source Computer Vision Library, 2000. Disponível em: https://opencv.org/. Acesso em: 29 abr. 2025.

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**, Brasília, 25 abr. 2002.

CALIMAN, Pedro Ferreira. **Desenvolvimento de um sistema de reconhecimento de sinais do alfabeto manual de Libras utilizando MediaPipe Hands e rede LSTM**. 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Faculdade de Ciências, Universidade Estadual Paulista (UNESP), Bauru, 2024. Disponível em: https://repositorio.unesp.br/server/api/core/bitstreams/ff3139f3-4497-4fe4-808b-45abfa1388a0/content. Acesso em: 29 abr. 2025.

DA, Luc. **IA na Tradução de Língua de Sinais: Uma Revolução**. Meliva.ai, 1 jan. 2025. Disponível em: https://meliva.ai/artificial-intelligence/ia-traducao-lingua-de-sinais/. Acesso em: 30 abr. 2025.

GOOGLE. **MediaPipe**. Google Developers, 2023. Disponível em: https://developers.google.com/mediapipe.

Acesso em: 29 abr. 2025.

GUPTA, Harsh. **Sign Language Interpreter using Deep Learning**. GitHub, 2021. Disponível em: https://github.com/harshbg/Sign-Language-Interpreter-using-Deep-Learning. Acesso em: 29 abr. 2025.

HAND TALK. **Aplicativo Hand Talk**. 2024a. Disponível em: https://www.handtalk.me/br/. Acesso em: 29 abr. 2025.

HAND TALK. **Os 5 parâmetros da Libras: quais são eles e sua importância**. 2024b. Disponível em: https://www.handtalk.me/br/blog/parametros-da-libras/. Acesso em: 29 abr. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Censo Demográfico 2022**: características gerais dos moradores, domicílios e moradores com deficiência. Rio de Janeiro: IBGE, 2024. Disponível em: https://censo2022.ibge.gov.br/panorama/. Acesso em: 11 nov. 2025.

IOFFE, Sergey; **SZEGEDY**, Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 32., 2015, Lille, França. **Anais** [online]. 2015. Disponível em: https://arxiv.org/abs/1502.03167. Acesso em: 20 jul. 2025.

MODOBILINGUE. **O que é Libras?**. 12 dez. 2022. Disponível em: https://www.modobilingue.com.br/post/o-que-é-libras. Acesso em: 29 abr. 2025.

PRAXEDES, Anderson. **Leitor de Libras**. GitHub, 2020. Disponível em: https://github.com/andersonprax/Leitor-de-Libras. Acesso em: 30 abr. 2025.

QUADROS, Ronice Müller de; **KARNOPP**, Lodenir Becker. **Língua de sinais brasileira: estudos linguísticos**. Porto Alegre: Artmed, 2004.

REZENDE, Tamires Martins. **Aplicação de técnicas de inteligência computacional para análise da expressão facial em reconhecimento de sinais de Libras**. 2016. Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Belo Horizonte, 2016. Disponível em: https://www.researchgate.net/publication/344904811_Analise_da_expressao_facial_em_reconhecimento_de_sinais_de_Libras.

Acesso em: 29 abr. 2025.

SENADO FEDERAL. **Baixo alcance da língua de sinais leva surdos ao isolamento**. Agência Senado, Brasília, DF, 25 abr. 2019. Disponível em: https://www12.senado.leg.br/noticias/especiais/especial-cidadania/baixo-alcance-da-lingua-de-sinais-leva-surdos-ao-isolamento. Acesso em: 11 nov. 2025.

SERRANO, Lucas. **RAS-Libras: Reconhecimento do Alfabeto de Sinais**. GitHub, 2020. Disponível em: https://github.com/lucas-serrano/Projeto-LibRAS. Acesso em: 30 abr. 2025.

THOMAS, Kevin Jose. **Sign Language Processing (ASL English Translation)**. GitHub, 2021. Disponível em: https://github.com/kevinjosethomas/sign-language-processing. Acesso em: 30 abr. 2025.