Sistema de tradução simultanêa de Libras baseado em visão computacional e IA.

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

Autor: Alysson Carlos de Castro Cordeiro.
Orientador: Prof. Guilherme Cestari.
Instituição: Inteli - Instituto de Tecnologia e Liderança.
Ano: 2025.

Sumário

Introdução

- **A comunicação é um direito fundamental**, mas a inclusão de estudantes surdos no ensino médio brasileiro enfrenta barreiras significativas.

- Há cerca de 2,3 milhões de pessoas com deficiência auditiva severa ou total (IBGE, 2024), e menos de 1% da população ouvinte domina Libras (Senado Federal, 2019).

Problematização

- Escassez de intérpretes de Libras (especialmente em escolas públicas) e a baixa disseminação da língua entre ouvintes. Isso gera exclusão social e acadêmica.

Fonte: https://cienciahoje.org.br/artigo/como-ensinar-os-alunos-surdos-em-contextos-inclusivos/

Objetivo Geral

Desenvolver um sistema de tradução simultânea de Libras para texto baseado em visão computacional e IA, focado na inclusão educacional de estudantes surdos e ouvintes no ensino médio brasileiro, promovendo acessibilidade.

1. Desenvolver o modelo de visão computacional com MediaPipe.
2. Utilizar Long Short-Term Memory (LSTM) para criar memória artificial dos sinais dinâmicos em tempo real.
3. Garantir privacidade e segurança.
4. Otimizar para escalabilidade.
5. Garantir baixo custo e acessibilidade financeira

Hipótese

É viavel desenvolver um sistema de tradução de Libras que opere em tempo real, seja financeiramente acessível para escolas públicas e culturamente sensível utilizando técnicas de *Deep Learning* e visão computacional.

Fundamentação Teórica

- **Libras:** Uma língua visual-espacial complexa que envolve não apenas mãos, mas expressões faciais e corporais. (Hand Talk, 2024)

- **IA:** Uso de CNNs (Redes Neurais Convolucionais) para imagens estáticas e RNNs/LSTMs para sequências temporais (movimento)

- **Visão Computacional:** MediaPipe para detecção de landmarks das mãos e OpenCV para processamento de imagens, permitindo captura e interpretação de gestos.

Benchmark (Estado da Arte)

Projeto	Tecnologia	Acurácia/Precisão	Tempo de Resposta	Acessibilidade e Custo	Cobertura Linguística	Pontos Fortes
Sign-Language-Detection (CNN)	CNN + Dataset Kaggle	Alta (99%)	Tempo real	Gratuito, código aberto	Gestos básicos de ISL (Língua de Sinais Indiana)	Alta precisão para gestos estáticos
Sign Language Interpreter (Deep Learning)	TensorFlow + Keras + OpenCV	Boa (95%)	Tempo real	Gratuito, código aberto	44 caracteres da ASL (Língua de Sinais Americana)	Reconhecimento em tempo real
ASL - English Translation	MediaPipe PointNet + ThreeJS	Variável (CNNs: 95% estáticos; RNNs: 85-90% frases; Transformers: 80-95% contextos	Moderado (gestos simples: <1s; dinâmicos: 1-3s; complexos: 3-5s+)	Gratuito, requer hardware poderoso	ASL com expressões faciais	Tradução ASL-Inglês
Convolutional Neural Networks LIBRAS	CNN + OpenCV	Alta (dependente do dataset)	Tempo real	Gratuito, código aberto	Alfabeto em Libras	Foco no alfabeto de Libras
Libras Reader	TensorFlow + Teachable Machine + MediaPipe	Boa (dependente do dataset)	Tempo real	Gratuito, código aberto	Números em Libras	Integração com robótica
RAS-Libras	TensorFlow + OpenCV + PyQt5	Boa (dependente do dataset)	Tempo real	Gratuito, código aberto	Alfabeto em Libras	Aprendizado interativo de Libras

Quadro 1: Elaborada pelo autor

Benchmark (Estado da Arte)

- "Sign-Language-Detection" usa apenas CNN (foca só em imagens estáticas).
- "ASL - English Translation" é mais completo, mas exige hardware poderoso, o que viola sua premissa de acessibilidade. Outros são genéricos ("Free, open-source") mas não pensados para uma sala de aula pobre.
- Vários projetos na prometem "Real-time" (Tempo Real) ou respostas em menos de 1 segundo.

- Arquitetura combinada (CNN para estáticos + LSTM para dinâmicos) rodando em hardware acessível. o problema dos gestos que se mexem sem precisar de um supercomputador, algo que os outros da lista não priorizaram da mesma forma.
- Teve uma latência de no máximo 2-3 segundos. *Arquitetura LSTM pura (janela de 30 frames) cria esse atraso.*
- *Foco é escolas públicas e baixo custo*

Metodologia e Arquitetura do sistema

Ferramentas e Pipeline

- Python, MediaPipe (Landmarks), TensorFlow/Keras.
- Desenvolvimento interativo *(Binário, Multiclasse; Hibrido)*

Modelo Dinâmico

- **Arquitetura:** LSTM (Long Short-Term Memory).
- **Configuração:** 2 Camadas (256 e 128 unidades) + *Batch Normalization.*
- **Regularização:** Dropout de 0.3 (para evitar overfitting).
- **Entrada:** Sequências de **30 frames** (Janela Temporal).

Modelo Estático

- **Arquitetura:** CNN (Rede Neural Convolucional).
- **Configuração**: 3 Camadas Convolucionais (32, 64, 128 filtros) + *Max Pooling.*
- **Entrada**: Imagens 64x64 pixels (Escala de Cinza).
- **Decisão Híbrida:** Script de integração prioriza confiança > **80%**

Metodologia e Arquitetura do sistema

Ferramentas e Pipeline

- Python, MediaPipe (Landmarks), TensorFlow/Keras.
- Desenvolvimento interativo *(Binário, Multiclasse; Hibrido)*

Modelo Dinâmico

- **Arquitetura:** LSTM (Long Short-Term Memory).
- **Configuração:** 2 Camadas (256 e 128 unidades) + *Batch Normalization.*
- **Regularização:** Dropout de 0.3 (para evitar overfitting).
- **Entrada:** Sequências de **30 frames** (Janela Temporal).

Modelo Estático

- **Arquitetura:** CNN (Rede Neural Convolucional).
- **Configuração**: 3 Camadas Convolucionais (32, 64, 128 filtros) + *Max Pooling.*
- **Entrada**: Imagens 64x64 pixels (Escala de Cinza).
- **Decisão Híbrida:** Script de integração prioriza confiança > **80%**

Coleta de Dados, Testes e Resultados

- Criação de datasets próprios,
- Coleta de vídeos para sinais dinâmicos ("H" com 691; "K" com 690 e "Outros" com 713) → 2041 sequências válidas após pré-processamento.
- Estáticos: 1600 imagens por letra ("A", "B", "outros"); 1200 por número (1-9) → Total 10.800 imagens.

Testes em Tempo Real: Alta precisão subjetiva em condições controladas, mas confusões entre "H"/"K" e sensibilidade a luz/distância (ideal: 30-50 cm). Testes com 8 participantes revelaram usabilidade razoável, mas latência de 1 a 3s e necessidade de tutoriais para iniciantes.

Resultados de Validação:

Métrica	LSTM (Dinâmicos)	CNN (Estáticos)
Acurácia Treino	97.55%	80.81%
Acurácia Validação	90.22%	86.03%
Perda Validação	0.3913	0.4953

Limitações, Restrições

- **O Grande Desafio (Latência):** Testes práticos revelaram um atraso de 2 a 3 segundos na tradução
- **Causa:** A "janela de processamento" de 30 frames necessária para a LSTM entender o movimento cria esse “delay”, o que impede uma tradução verdadeiramente "simultânea".
- **Sensibilidade:** O sistema mostrou-se muito sensível a variações de luz e distância da câmera.
- **Arquitetura**: Concluiu-se que usar apenas landmarks + LSTM é insuficiente para unir gestos estáticos e dinâmicos com eficiência.

Conclusão e Resposta à Hipótese

- A hipótese inicial foi **parcialmente refutada (**acurácia > 90% de validação**)**. Embora seja viável criar o sistema, a arquitetura escolhida inicialmente (LSTM pura) não suporta a "baixa latência" necessária para uma conversa fluida em tempo real

- O "Pivô" (Solução Proposta): A pesquisa apontou que o caminho correto é migrar para uma arquitetura **LRCN** (Long-term Recurrent Convolutional Network).

- Isso permitirá processar o vídeo diretamente **(análise espaço-temporal unificada),** resolvendo o problema da latência e melhorando a precisão.

- **Impacto Final:** O projeto evoluiu para uma proposta de plataforma de aprendizado de Libras, mantendo o foco na inclusão social e educacional.

Contribuições deste Trabalho

- Criação de Dataset Próprio e Diversificado;
- Diagnóstico de Latência em Arquiteturas LSTM;
- Proposta de Arquitetura Híbrida (LRCN);
- Foco na Realidade Escolar Brasileira;

Análise de Submissão para Eventos

- EduComp 2026 (Simpósio Brasileiro de Educação em Computação): O evento foca em como a computação pode apoiar a educação.
- CAIE (Congresso de Acessibilidade e Inclusão na Educação): Aderência muito alta. É o público-alvo exato do problema de pesquisa.

Referências

1.	
2.	
3.	
4.	