UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
OFICINA DE INTEGRAÇÃO [ELE64]

PEDRO SCHUVES MARODIN
JOÃO VICTOR CASTEX FERREIRA DA MOTTA
RAFAEL MERCHIORI DE SOUZA

SIGNSPEAK – ÓCULOS TRADUTOR DE LIBRAS EM ÁUDIO

RELATÓRIO

CURITIBA

2025

PEDRO SCHUVES MARODIN
JOÃO VICTOR CASTEX FERREIRA DA MOTTA
RAFAEL MERCHIORI DE SOUZA

SIGNSPEAK – ÓCULOS TRADUTOR DE LIBRAS EM ÁUDIO

Relatório apresentado para matéria Oficina de Integração [ELE64] da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação na matéria.

Orientador: Cesar Manuel Vargas Benitez

CURITIBA

2025

RESUMO

Este relatório técnico detalha o desenvolvimento do projeto SignSpeak, um sistema inovador de visão computacional integrado ao Raspberry Pi 4, projetado para traduzir gestos da Língua Brasileira de Sinais (Libras) em áudio. O sistema utiliza Redes Neurais Convolucionais (CNNs) para o reconhecimento de gestos em tempo real, processamento de imagem com OpenCV e síntese de fala offline. Serão abordados os fundamentos teóricos das CNNs, a arquitetura do sistema, a metodologia de treinamento do modelo, os componentes de hardware utilizados e os resultados alcançados, demonstrando a viabilidade e o impacto potencial desta tecnologia assistiva.

Palavras-chave: Visão Computacional, Libras, Redes Neurais Convolucionais

ABSTRACT

This technical report details the development of the SignSpeak project, an innovative computer vision system integrated with the Raspberry Pi 4, designed to translate gestures from the Brazilian Sign Language (Libras) into audio. The system utilizes Convolutional Neural Networks (CNNs) for real-time gesture recognition, image processing with OpenCV, and offline speech synthesis. The theoretical foundations of CNNs, the system architecture, the model training methodology, the hardware components used, and the achieved results will be addressed, demonstrating the feasibility and potential impact of this assistive technology.

Keywords: Computer Vision, Brazilian Sign Language, Convolutional Neural Networks

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A comunicação é um pilar fundamental da interação humana, e a Língua Brasileira de Sinais (Libras) desempenha um papel crucial para a comunidade surda no Brasil. No entanto, a ausência de intérpretes de Libras em diversos contextos cotidianos cria barreiras significativas, limitando a inclusão social e a autonomia de indivíduos surdos. Essa lacuna na comunicação motivou o desenvolvimento do projeto SignSpeak, uma solução tecnológica assistiva que visa preencher essa lacuna.

O SignSpeak consiste em um protótipo de óculos inteligente equipado com uma webcam integrada, capaz de capturar e interpretar gestos de Libras em tempo real. A inovação central reside na utilização de técnicas avançadas de visão computacional, especificamente Redes Neurais Convolucionais (CNNs), para o reconhecimento preciso desses gestos. Uma vez que o gesto é identificado, o sistema converte a informação em áudio, que é então transmitido via fone de ouvido Bluetooth, proporcionando uma comunicação fluida e acessível sem a necessidade de conexão com a internet.

Este relatório técnico tem como objetivo detalhar a concepção, o desenvolvimento e a implementação do projeto SignSpeak. Serão abordados os fundamentos teóricos que sustentam a visão computacional e as redes neurais, a arquitetura do sistema, a metodologia de treinamento do modelo de reconhecimento de gestos, os componentes de hardware empregados, bem como os resultados obtidos e as considerações finais. A expectativa é que o SignSpeak não apenas facilite o diálogo em tempo real, mas também promova uma maior independência e participação da comunidade surda na sociedade, contribuindo para a inclusão digital e social.

1.2 OBJETIVOS

O projeto SignSpeak foi concebido com o propósito de desenvolver uma solução tecnológica inovadora e portátil para mitigar as barreiras de comunicação enfrentadas pela

comunidade surda. A proposta central é a criação de um dispositivo vestível que traduza a Língua Brasileira de Sinais (Libras) para voz, de forma autônoma e em tempo real. Para alcançar este propósito, foram estabelecidos os seguintes objetivos:

1.2.1 OBJETIVO GERAL

Desenvolver e validar um protótipo funcional de óculos inteligente, baseado na plataforma embarcada Raspberry Pi 4, capaz de identificar gestos estáticos do alfabeto da Libras através de visão computacional, e traduzi-los para áudio sintetizado em tempo real, promovendo assim maior inclusão e autonomia para pessoas surdas.

1.2.2 OBJETIVOS ESPECÍFICOS

- **Desenvolver um processo de Visão Computacional para segmentação de gestos:** Implementar um sistema robusto para a captura e o processamento de imagens em tempo real. Isso inclui a definição de uma Região de Interesse (ROI) fixa no campo de visão da câmera e a aplicação de segmentação por cor no espaço HSV para isolar a mão do usuário do plano de fundo, gerando a máscara binária que servirá de entrada para o modelo de reconhecimento (SOARES et al., 2019; YUAN et al., 2021).
- **Construir e treinar um modelo de Rede Neural Convolucional (CNN) otimizado:** Projetar, treinar e avaliar uma arquitetura de CNN customizada para a classificação precisa de 21 gestos estáticos do alfabeto da Libras. O treinamento deve utilizar técnicas de Data Augmentation (como rotação, zoom e espelhamento) para aumentar a generalização e a robustez do modelo a diferentes condições de uso (SHORTEN; KHOSHGOFTAAR, 2019; SIMONYAN; ZISSERMAN, 2014).
- **Integrar o modelo de deep learning em um sistema embarcado:** Realizar a implementação do modelo Keras/TensorFlow treinado no Raspberry Pi 4, otimizando a inferência para garantir baixa latência e viabilizar o reconhecimento em tempo real, um requisito crítico para a comunicação fluida.
- **Implementar a conversão de texto-para-fala (Text-to-Speech) de forma offline:** Integrar uma biblioteca TTS que opere localmente no dispositivo, sem dependência de conexão com a internet, para converter a letra classificada pelo modelo em uma saída de áudio clara e imediata.
- **Projetar e montar um protótipo de hardware vestível:** Desenvolver um invólucro impresso em 3D para os óculos que acomode de forma ergonômica e discreta a webcam,

o Raspberry Pi 4 e os componentes associados, garantindo portabilidade e conforto para o usuário.

- **Validar o desempenho e a usabilidade do protótipo:** Executar testes sistemáticos para avaliar a acurácia da classificação de gestos em diferentes condições de iluminação, o tempo de latência total (da captura do gesto à saída de áudio) e a praticidade do dispositivo em cenários de uso simulados.
- **Disponibilizar e documentar o projeto de forma aberta:** Documentar de maneira detalhada toda a metodologia, o desenvolvimento de software, a arquitetura do modelo e o design do hardware, publicando o código-fonte e os resultados para fomentar futuras pesquisas e aprimoramentos pela comunidade.

2 FUNDAMENTAÇÃO TEÓRICA

Para a compreensão aprofundada do sistema SignSpeak, é imperativo estabelecer uma sólida base teórica nos campos do Deep Learning, Visão Computacional e, especificamente, das Redes Neurais Convolucionais (CNNs). Esta seção detalhará os conceitos fundamentais, a arquitetura e o funcionamento dessas tecnologias, bem como as bibliotecas e ferramentas empregadas no desenvolvimento do modelo de reconhecimento de gestos.

2.1 DEEP LEARNING E REDES NEURAIS

Deep Learning, ou Aprendizagem Profunda, é um subcampo de Machine Learning que se baseia em Redes Neurais Artificiais (RNAs) com múltiplas camadas para aprender representações de dados com vários níveis de abstração. Diferentemente dos métodos tradicionais, que frequentemente exigem uma engenharia de características manual, o Deep Learning é capaz de aprender automaticamente as características relevantes diretamente dos dados brutos, como pixels de uma imagem.

Uma Rede Neural Artificial é um modelo computacional inspirado na estrutura do cérebro, composta por neurônios (ou nós) organizados em camadas. Cada conexão entre neurônios possui um peso, que é ajustado durante o treinamento. Um neurônio processa os sinais de entrada, aplica uma função de ativação e transmite o resultado para a camada seguinte.

2.1.1 PERCEPTRON MULTICAMADAS (MLP) E SUAS LIMITAÇÕES

O Perceptron Multicamadas (MLP) é um tipo fundamental de rede neural feedforward, onde a informação flui da entrada para a saída. Embora eficientes para dados tabulares, os MLPs apresentam uma limitação crítica para tarefas de visão computacional: ao vetorizar uma imagem para servir como entrada, a estrutura espacial dos pixels é perdida. Um MLP trata pixels vizinhos da mesma forma que pixels distantes, ignorando as relações locais que formam bordas, texturas e formas. Isso não só é ineficiente, mas também leva a um número excessivo

de parâmetros, tornando o treinamento para imagens de alta resolução computacionalmente inviável.

2.1.2 REDES NEURAIS CONVOLUCIONAIS (CNN)

- **Camada Convolucional (Conv2D):** É o bloco de construção central da CNN. Em vez de analisar cada pixel individualmente, esta camada aplica um conjunto de filtros (ou kernels) sobre a imagem de entrada. Cada filtro é especializado em detectar uma característica específica, como uma borda vertical, uma curva ou uma cor. No modelo deste projeto, foram utilizadas três camadas convolucionais com filtros de dimensão 3x3.

- **Função de Ativação (LeakyReLU):** Após a convolução, uma função de ativação introduz não-linearidade na rede, permitindo que ela aprenda padrões complexos. O modelo implementado utiliza a função LeakyReLU (Leaky Rectified Linear Unit), uma variação da popular função ReLU que ajuda a mitigar o problema do "neurônio morto" (dying ReLU), permitindo um pequeno gradiente quando a unidade não está ativa.

- **Camada de Pooling (MaxPooling2D):** Esta camada é aplicada após a ativação para reduzir a dimensionalidade espacial dos mapas de características. Ela agrega os valores em uma vizinhança (neste projeto, uma janela 2x2) e mantém apenas o valor máximo. Isso torna a representação mais compacta, reduz o custo computacional e confere à rede uma certa invariância a pequenas translações no objeto de interesse.

- **Camada de Achatamento (Flatten):** Após as etapas de extração de características pelas camadas convolucionais e de pooling (processo que compacta e resume as características mais importantes), a matriz resultante é convertida em um vetor unidimensional por esta camada. Isso prepara os dados para serem processados pelas camadas densas finais.

- **Camadas Densas (Dense):** Semelhantes às camadas de um MLP, as camadas densas são totalmente conectadas e realizam a classificação final com base nas características extraídas. O modelo do projeto utiliza uma camada densa com 256 neurônios, seguida por uma camada de saída com 21 neurônios (uma para cada classe de gesto) e a função de ativação softmax, que converte as saídas da rede em um vetor de probabilidades.

- **Regularização (Dropout):** Para combater o sobreajuste (overfitting), onde o modelo memoriza os dados de treino e perde a capacidade de generalizar, foi inserida uma camada de Dropout com uma taxa de 0.5. Durante o treinamento, essa camada desativa aleatoriamente 50% dos neurônios, forçando a rede a aprender caminhos e representações

mais robustas. A eficácia dessa técnica é visível na análise dos gráficos de treinamento, que mostram uma diferença entre a acurácia de treino e a de validação.

2.2 REDES NEURAIS CONVOLUCIONAIS (CNNS)

Para superar as limitações do MLP, as Redes Neurais Convolucionais (CNNs) foram propostas e se tornaram o padrão para tarefas de visão computacional. A CNN preserva a estrutura espacial dos dados através de camadas especializadas que realizam operações de convolução.

No contexto deste projeto, foi desenvolvida uma arquitetura de CNN específica para a classificação dos gestos de Libras. As camadas fundamentais que compõem este modelo são descritas a seguir.

As Redes Neurais Convolucionais (CNNs) representam uma classe especializada de redes neurais artificiais, amplamente empregadas em tarefas de Visão Computacional, como reconhecimento de padrões, classificação de imagens, detecção de objetos e segmentação semântica. A principal distinção das CNNs em relação às MLPs reside na sua capacidade de preservar a estrutura espacial dos dados de entrada, o que é crucial para o processamento de imagens (KRIZHEVSKY et al., 2012).

Inspiradas no córtex visual biológico, as CNNs operam com o conceito de *Campos Receptivos*. Isso significa que, em vez de processar a imagem inteira de uma vez, a rede analisa pequenos pedaços da imagem, permitindo que neurônios diferentes processem pixels em diferentes regiões, mantendo a proximidade espacial. Essa abordagem reduz drasticamente o número de parâmetros a serem aprendidos, tornando o processamento de imagens de alta resolução viável e eficiente, ao contrário das MLPs que exigiriam um número proibitivo de neurônios e parâmetros para a mesma tarefa (RUSSAKOVSKY et al., 2015).

Uma CNN é tipicamente estruturada em dois blocos principais: o bloco de extração de características e o bloco de classificação.

2.2.1 BLOCO DE EXTRAÇÃO DE CARACTERÍSTICAS

Este bloco é responsável por identificar e extrair características relevantes das imagens de entrada. É composto principalmente por camadas de convolução e camadas de pooling, que atuam de forma hierárquica para aprender representações cada vez mais complexas dos dados (LECUN et al., 2015).

2.2.1.1 CAMADA DE CONVOLUÇÃO (CONVOLUTIONAL LAYER)

A convolução é a operação fundamental da CNN. Ela consiste em aplicar um filtro, ou *kernel* (K), sobre um volume de entrada (I), como uma imagem. O kernel é uma pequena matriz de pesos que desliza através das dimensões espaciais da entrada, calculando o produto escalar entre seus pesos e a região da imagem que ele cobre em cada posição. O resultado é um *mapa de ativação* ou *mapa de características* (S), que indica a resposta daquele filtro em cada posição espacial (DUMOULIN; VISIN, 2016).

Matematicamente, a operação de convolução discreta 2D é definida pela equação (1):

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{1}$$

Os hiperparâmetros que definem a arquitetura de uma camada convolucional são:

- **Profundidade (Depth):** O número de filtros (kernels) na camada. Cada filtro aprende a detectar uma característica diferente. A pilha de mapas de ativação de todos os filtros forma o volume de saída.
- **Tamanho do Kernel (Kernel Size):** As dimensões espaciais de cada filtro. No modelo deste projeto, foram utilizados exclusivamente kernels de 3x3, um tamanho pequeno que é eficiente e eficaz na captura de relações espaciais locais (SIMONYAN; ZISSERMAN, 2014).
- **Salto (Stride):** O número de pixels que o filtro se move sobre a imagem a cada passo. Um stride de 1, como o utilizado neste projeto, resulta em uma análise mais detalhada e mapas de características maiores.
- **Preenchimento (Padding):** A adição de pixels (geralmente zeros) nas bordas do volume de entrada. No projeto, foi utilizado o *zero-padding* com a configuração “same”, que garante que o volume de saída tenha as mesmas dimensões espaciais que o de entrada.

2.2.1.2 FUNÇÕES DE ATIVAÇÃO

Após cada operação de convolução, um mapa de ativação é gerado. A este mapa é aplicada uma função de ativação não-linear, elemento por elemento. A não-linearidade é crucial, pois sem ela, a CNN se reduziria a uma simples transformação linear, incapaz de aprender padrões complexos. As funções utilizadas neste projeto incluem:

- **ReLU (Rectified Linear Unit):** Define a saída como o máximo entre zero e a entrada, conforme a equação (2). É a ativação mais comum devido à sua eficiência computacional (NAIR; HINTON, 2010).

$$f(x) = \max(0, x) \tag{2}$$

- **Leaky ReLU:** Uma evolução da ReLU. Para resolver o problema do "neurônio morto", a Leaky ReLU permite um pequeno gradiente não-nulo para valores negativos, como mostra a equação (3) (MAAS et al., 2013). Foi a principal função de ativação utilizada nos blocos convolucionais deste projeto.

$$f(x) = \begin{cases} x & \text{se } x \geq 0 \\ \alpha x & \text{se } x < 0 \end{cases} \tag{3}$$

- **Tanh (Tangente Hiperbólica):** Mapeia as entradas para o intervalo $[-1, 1]$, definida pela equação (4). Embora menos comum nas CNNs modernas, pode ser útil em blocos específicos (LECUN et al., 1998).

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

Essas funções de ativação são cruciais para otimizar o treinamento, permitindo que a rede aprenda representações complexas e desative neurônios que não são influentes para o aprendizado das camadas subsequentes (RUSSAKOVSKY et al., 2015).

2.2.1.3 CAMADAS DE POOLING

Após uma ou mais camadas de convolução, é comum aplicar uma camada de *pooling* (ou subamostragem). O principal objetivo do pooling é reduzir a dimensionalidade dos mapas de características, diminuindo o número de parâmetros e a complexidade computacional da rede, o que, por sua vez, acelera o treinamento e ajuda a controlar o overfitting (BOUREAU et al., 2010). Além disso, as camadas de pooling tornam a representação da característica mais robusta a pequenas translações e distorções na imagem de entrada (SZEGEDY et al., 2015).

Existem diferentes métodos de pooling, sendo os mais utilizados:

- **Max Pooling:** Seleciona o valor máximo de uma região (janela) do mapa de características. Este método é eficaz para capturar as características mais proeminentes em uma região, descartando informações menos relevantes.

- **Average Pooling:** Calcula a média dos valores em uma região do mapa de características. Embora menos comum que o Max Pooling para extração de características, pode ser útil em certas arquiteturas ou na camada final de algumas redes.

2.2.2 BLOCO DE CLASSIFICAÇÃO: CAMADAS TOTALMENTE CONECTADAS

Após o bloco de extração de características, que gera representações de alto nível da imagem de entrada, a CNN transita para o bloco de classificação. Este bloco é composto por uma ou mais camadas totalmente conectadas (*Fully Connected Layers - FC*), que se assemelham às camadas ocultas de uma MLP. Cada neurônio em uma camada FC está conectado a todos os neurônios da camada anterior, recebendo as características extraídas e aprendendo a combiná-las para realizar a tarefa de classificação (HE et al., 2016).

A camada final do bloco de classificação é a camada de saída, que possui um número de neurônios igual ao número de classes que o modelo deve classificar (no caso do SignSpeak, as letras do alfabeto de Libras). Para tarefas de classificação multiclasse, a função de ativação *Softmax* é comumente aplicada a esta camada (BISHOP, 2006). A função Softmax converte as saídas brutas da rede (logits) em probabilidades, garantindo que a soma de todas as probabilidades para as classes seja igual a 1. Isso permite que o modelo atribua uma probabilidade a cada classe, indicando a confiança da previsão.

2.2.2.1 CAMADA DE SAÍDA E A FUNÇÃO SOFTMAX

A camada final é uma camada densa cujo número de neurônios corresponde ao número de classes do problema (K). Para classificação multiclasse, a função de ativação **Softmax** é aplicada para converter o vetor de saídas brutas da rede (logits), $\mathbf{z}$, em um vetor de probabilidades, $\sigma(\mathbf{z})$. Para um vetor de logits $\mathbf{z} = [z_1, z_2, \ldots, z_K]$, a probabilidade da j-ésima classe é calculada como na equação (5):

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{para } j = 1, \ldots, K \tag{5}$$

A função Softmax garante que cada saída esteja no intervalo $[0, 1]$ e que a soma de todas as saídas seja 1, permitindo uma interpretação probabilística do resultado. Esta função é tipicamente pareada com a função de perda de Entropia Cruzada Categórica (*Categorical Cross-Entropy*), utilizada para treinar o modelo deste projeto (KINGMA; BA, 2014).

2.2.3 ARQUITETURA IMPLEMENTADA NO PROJETO SIGNSPEAK

A arquitetura da CNN desenvolvida para este projeto segue a sequência de camadas detalhada abaixo, refletindo as melhores práticas para classificação de imagens:

1. **INPUT:** Imagens de 64x64 pixels com 3 canais de cor (RGB).
2. **CONV1:** 32 filtros 3x3, 'padding='same'', ativação 'LeakyReLU'($\alpha = 0.1$).
3. **POOL1:** 'MaxPooling' com janela 2x2.
4. **CONV2:** 32 filtros 3x3, ativação 'LeakyReLU'($\alpha = 0.1$).
5. **POOL2:** 'MaxPooling' com janela 2x2.
6. **CONV3:** 64 filtros 3x3, ativação 'LeakyReLU'($\alpha = 0.1$).
7. **POOL3:** 'MaxPooling' com janela 2x2.
8. **FLATTEN:** Converte o volume 3D em um vetor 1D.
9. **DENSE1:** Camada totalmente conectada com 256 neurônios, ativação 'ReLU'.
10. **DROPOUT:** Taxa de 0.5 para regularização.
11. **OUTPUT (DENSE2):** Camada de saída com 21 neurônios (classes) e ativação 'Softmax'.

2.2.4 HISTÓRICO E EVOLUÇÃO DAS CNNS

A primeira Rede Neural Convolucional a ser implementada e testada com sucesso foi a LeNet-5, desenvolvida por Yann LeCun e sua equipe em 1998. Publicada no artigo *'Gradient-based learning applied to document recognition'* (OpenCV Team, 2025), a LeNet-5 foi projetada para o reconhecimento de dígitos manuscritos, como os utilizados em códigos postais e cheques bancários. Sua arquitetura inovadora, que combinava camadas convolucionais e de pooling seguidas por camadas totalmente conectadas, demonstrou a eficácia das CNNs para tarefas de visão computacional e pavimentou o caminho para o desenvolvimento de arquiteturas mais complexas e poderosas nas décadas seguintes (RUSSAKOVSKY et al., 2015).

2.3 BIBLIOTECAS E FERRAMENTAS

O desenvolvimento do projeto SignSpeak e, em particular, do modelo de reconhecimento de gestos, faz uso de diversas bibliotecas e ferramentas de software que são pilares no campo da visão computacional e do aprendizado de máquina. As principais são:

2.3.1 OPENCV (OPEN SOURCE COMPUTER VISION LIBRARY)

OpenCV é uma biblioteca de código aberto amplamente utilizada para aplicações de visão computacional e aprendizado de máquina. Ela fornece uma vasta gama de funções para processamento de imagens e vídeos, incluindo operações como leitura e escrita de imagens, redimensionamento, conversão de cores, filtragem, detecção de características, segmentação, e muito mais. No contexto do SignSpeak, o OpenCV é fundamental para o módulo de pré-processamento de imagem, onde as imagens capturadas pela webcam são preparadas antes de serem alimentadas ao modelo de CNN. Isso inclui operações como redimensionamento para o formato de entrada esperado pela rede, normalização de pixels e, possivelmente, a aplicação de filtros para realçar características relevantes ou reduzir ruído (TensorFlow Developers, 2025).

2.3.2 TENSORFLOW E TENSORFLOW LITE (TFLITE)

TensorFlow é uma plataforma de código aberto desenvolvida pelo Google para aprendizado de máquina. É uma das bibliotecas mais populares para construir e treinar modelos de Deep Learning, incluindo redes neurais complexas como as CNNs. O TensorFlow oferece uma API flexível para definir, treinar e implantar modelos, suportando tanto CPUs quanto GPUs para aceleração computacional (TensorFlow Lite Team, 2025).

Para o projeto SignSpeak, a utilização do Raspberry Pi 4 como plataforma embarcada impõe restrições de recursos computacionais. É aqui que o TensorFlow Lite (TFLite) se torna crucial. TFLite é uma versão otimizada do TensorFlow, projetada especificamente para a inferência de modelos de aprendizado de máquina em dispositivos embarcados e móveis com recursos limitados, como microcontroladores e smartphones. Ele permite a conversão de modelos TensorFlow treinados para um formato mais compacto e eficiente, que pode ser executado com alta performance e baixa latência em hardware de borda. Essa otimização é vital para garantir que o reconhecimento de gestos ocorra em tempo real no Raspberry Pi 4, sem a necessidade de uma GPU dedicada (NumPy Developers, 2025).

2.3.3 OUTRAS BIBLIOTECAS PYTHON

Além das mencionadas, outras bibliotecas Python são comumente empregadas em projetos de visão computacional e aprendizado de máquina, como:

- **NumPy:** Fundamental para computação numérica em Python, fornecendo suporte para arrays e matrizes de alta performance, essenciais para manipulação de dados de imagem e tensores em redes neurais (NumPy Developers, 2025).
- **Matplotlib/Seaborn:** Utilizadas para visualização de dados, como gráficos de perda e acurácia durante o treinamento do modelo, auxiliando na análise e depuração (Matplotlib Development Team, 2025).
- **Scikit-learn:** Embora o treinamento da CNN seja feito com TensorFlow, o Scikit-learn pode ser útil para tarefas auxiliares, como pré-processamento de dados, divisão de conjuntos de dados (treino, validação, teste) e avaliação de métricas de desempenho (scikit-learn Developers, 2025).

2.4 RASPBERRY PI 4 COMO PLATAFORMA EMBARCADA

O Raspberry Pi 4 Model B foi selecionado como a plataforma de hardware principal para o projeto SignSpeak devido à sua combinação ideal de poder de processamento, versatilidade, baixo custo e tamanho compacto, tornando-o uma escolha excelente para aplicações embarcadas que exigem capacidade computacional razoável sem o consumo excessivo de energia de um computador desktop tradicional (Raspberry Pi Foundation, 2025).

O Raspberry Pi 4 é equipado com um processador quad-core ARM Cortex-A72 (ARMv8) de 64 bits, operando a 1.5 GHz, e está disponível com 2GB, 4GB ou 8GB de RAM LPDDR4. Possui interfaces de conectividade modernas, incluindo duas portas micro-HDMI (suportando até 4Kp60), duas portas USB 3.0, duas portas USB 2.0, Gigabit Ethernet, Wi-Fi 802.11ac e Bluetooth 5.0. A presença de um conector CSI (Camera Serial Interface) dedicado para câmeras e um conector DSI (Display Serial Interface) para displays, além de um conjunto de pinos GPIO (General Purpose Input/Output), o torna particularmente adequado para projetos de visão computacional e integração com periféricos.

2.4.1 CAPACIDADE DE PROCESSAMENTO PARA VISÃO COMPUTACIONAL

Embora o Raspberry Pi 4 não possua uma Unidade de Processamento Gráfico (GPU) dedicada de alto desempenho como as encontradas em estações de trabalho ou servidores, sua CPU ARM Cortex-A72 oferece um desempenho significativamente superior em comparação com as gerações anteriores do Raspberry Pi. Para tarefas de inferência de modelos de aprendizado de máquina, especialmente quando otimizados com ferramentas como o TensorFlow Lite, o Raspberry Pi 4 demonstra capacidade suficiente para processar dados em tempo real.

2.4.2 DISPENSA DE GPU DEDICADA PARA TREINAMENTO E INFERÊNCIA

A decisão de não utilizar uma GPU dedicada para o treinamento do modelo no contexto do SignSpeak, e de realizar a inferência diretamente no Raspberry Pi 4 sem uma GPU externa, baseia-se em alguns fatores cruciais:

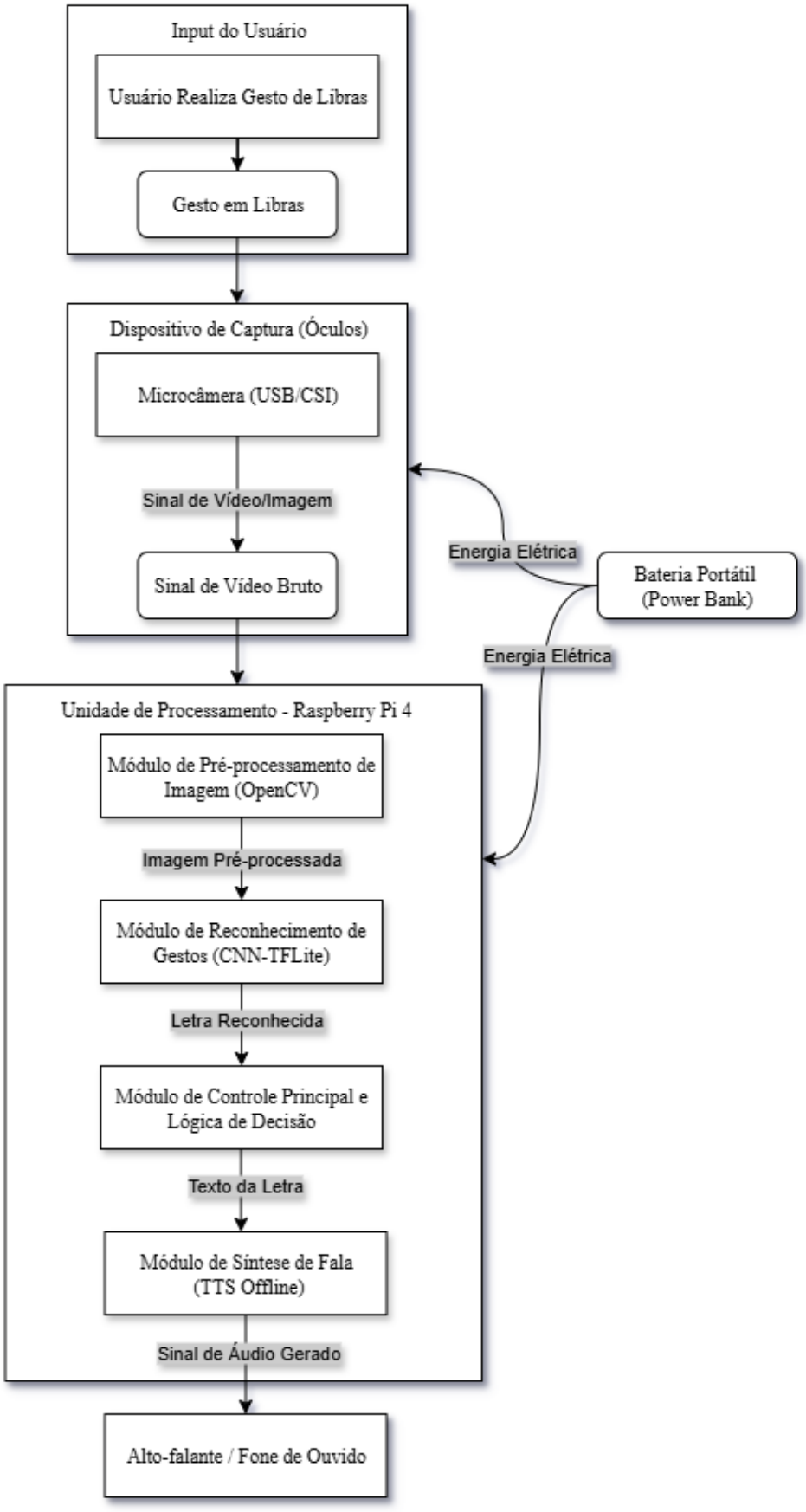
- **Tamanho da Rede Neural:** O modelo de CNN desenvolvido para o reconhecimento de gestos de Libras é considerado uma *rede pequena* em termos de complexidade e número de parâmetros. Redes menores exigem menos poder computacional para treinamento e, principalmente, para inferência.
- **Otimização com TensorFlow Lite:** Conforme mencionado, o TensorFlow Lite é projetado para otimizar modelos para execução em hardware com recursos limitados. Ele realiza quantização (redução da precisão dos pesos e ativações do modelo) e outras otimizações que permitem que o modelo seja executado eficientemente na CPU do Raspberry Pi 4, minimizando a necessidade de aceleração por GPU (NumPy Developers, 2025).
- **Natureza da Tarefa:** O reconhecimento de gestos em tempo real para um alfabeto limitado (Libras) não exige a mesma capacidade de processamento que tarefas mais complexas de visão computacional, como detecção de objetos em vídeo de alta resolução ou segmentação semântica em tempo real para veículos autônomos. A latência aceitável para a aplicação do SignSpeak permite o uso da CPU do Raspberry Pi 4.
- **Custo e Portabilidade:** A adição de uma GPU externa aumentaria significativamente o custo, o consumo de energia e o tamanho físico do protótipo, comprometendo a portabilidade e a viabilidade do projeto como um óculos inteligente. A solução baseada apenas no Raspberry Pi 4 é mais econômica e compacta.

Em suma, a combinação da arquitetura eficiente do Raspberry Pi 4 com a otimização de modelos via TensorFlow Lite permite que o SignSpeak realize o reconhecimento de gestos de Libras de forma eficaz e em tempo real, sem a necessidade de hardware de aceleração gráfica adicional, tornando o projeto prático e acessível.

3 METODOLOGIA

A metodologia técnica implementada no projeto SignSpeak é ilustrada no diagrama de blocos da Figura 1. O processo inicia com a captura de um gesto em Libras, realizado pelo usuário, através de uma microcâmera acoplada a um par de óculos. O sinal de vídeo bruto é enviado para a unidade de processamento, um Raspberry Pi 4, onde passa por um pipeline (sequência de etapas de processamento) de software robusto. Inicialmente, o módulo de pré-processamento de imagem, utilizando a biblioteca OpenCV, trata o quadro de vídeo para isolar o gesto. A imagem pré-processada é então analisada pelo módulo de reconhecimento de gestos, que emprega uma Rede Neural Convolucional (CNN) otimizada com TensorFlow Lite (TFLite) para classificar a letra correspondente. Uma vez que a letra é identificada, um módulo de controle principal aciona o sistema de síntese de fala (Text-to-Speech), que opera de forma offline para gerar o sinal de áudio. Finalmente, este áudio é reproduzido em um fone de ouvido ou alto-falante, completando o ciclo de tradução em tempo real. Todo o sistema é alimentado por uma bateria portátil (power bank), garantindo a portabilidade do dispositivo

Figura 1: Diagrama de blocos representando uma overview da metodologia implementada no sistema SignSpeak.



Fonte: Autoria Própria (2025)

3.1 TREINAMENTO E IMPLEMENTAÇÃO DO MODELO

O coração do sistema SignSpeak reside no seu modelo de Rede Neural Convolucional (CNN) responsável pelo reconhecimento dos gestos do alfabeto de Libras. Esta seção detalha o processo de treinamento do modelo, a motivação por trás da escolha da metodologia e os aspectos cruciais da sua implementação.

3.1.1 COLETA E PRÉ-PROCESSAMENTO DO DATASET

Para treinar um modelo de CNN eficaz, é essencial dispor de um dataset robusto e representativo. No projeto SignSpeak, foi realizada a coleta de um conjunto abrangente de imagens, cada uma representando uma letra do alfabeto de Libras. A diversidade do dataset é fundamental para garantir que o modelo seja capaz de generalizar bem para diferentes variações de gestos, iluminação e ângulos (RUSSAKOVSKY et al., 2015).

As imagens coletadas passam por um processo de pré-processamento utilizando a biblioteca OpenCV (OpenCV Team, 2025). Este processo inclui:

- **Redimensionamento:** As imagens são redimensionadas para um tamanho padrão (e.g., 64x64 pixels) que é compatível com a entrada da CNN. Isso garante uniformidade e otimiza o processamento.
- **Normalização:** Os valores dos pixels (geralmente no intervalo de 0 a 255) são normalizados para um intervalo menor (e.g., 0 a 1). Isso ajuda a acelerar o treinamento e a melhorar a convergência do modelo (NumPy Developers, 2025).
- **Aumento de Dados (Data Augmentation):** Para aumentar a robustez do modelo e evitar o overfitting, técnicas de aumento de dados podem ser aplicadas. Isso inclui rotações, translações, espelhamentos e ajustes de brilho nas imagens existentes, criando novas amostras de treinamento a partir das originais.

3.1.2 ARQUITETURA DA CNN

A arquitetura da CNN utilizada no SignSpeak foi projetada para ser eficiente e leve, visando a implantação em um dispositivo com recursos limitados como o Raspberry Pi 4 (Raspberry Pi Foundation, 2025). Embora os detalhes exatos da arquitetura (número de camadas convolucionais, filtros, etc.) possam variar com base em experimentações, uma arquitetura típica para reconhecimento de gestos pode incluir:

- Múltiplas camadas convolucionais seguidas por camadas de ativação ReLU e camadas de Max Pooling para extração hierárquica de características.
- Camadas de *Dropout* para regularização, ajudando a prevenir o overfitting durante o treinamento.
- Uma ou mais camadas totalmente conectadas para classificação, com a camada de saída utilizando a função de ativação Softmax para prever a probabilidade de cada letra de Libras.

3.1.3 PROCESSO DE TREINAMENTO

O treinamento do modelo de CNN envolve a alimentação do dataset pré-processado à rede, ajustando os pesos e vieses da rede para minimizar a função de perda (loss function) e maximizar a acurácia na classificação dos gestos. O processo é iterativo e guiado por um otimizador.

3.1.3.1 FUNÇÃO DE PERDA E OTIMIZADOR

Para problemas de classificação multiclasse como o reconhecimento de letras de Libras, a **Entropia Cruzada Categórica** (Categorical Cross-Entropy) é uma função de perda comumente utilizada. Ela mede a diferença entre a distribuição de probabilidade prevista pelo modelo e a distribuição de probabilidade real (o rótulo verdadeiro da imagem).

O **Otimizador Adam** (Adaptive Moment Estimation) é frequentemente empregado devido à sua eficiência e boa performance em uma ampla gama de problemas de aprendizado de máquina. Ele adapta as taxas de aprendizado para cada parâmetro individualmente, combinando as vantagens dos algoritmos RMSProp e AdaGrad (TensorFlow Developers, 2025).

3.1.3.2 ÉPOCAS DE TREINAMENTO

Uma **época** refere-se a um ciclo completo de treinamento, onde todo o dataset de treinamento é passado para a rede neural uma vez, tanto para o forward pass (cálculo das previsões) quanto para o backward pass (cálculo dos gradientes e atualização dos pesos). O número de épocas é um hiperparâmetro crucial que determina por quantas vezes o modelo irá ver e aprender com os dados de treinamento.

No contexto do projeto SignSpeak, a implementação de **61 épocas** para o treinamento do modelo de CNN foi uma decisão baseada na observação do desempenho do modelo nos

conjuntos de treinamento e validação.

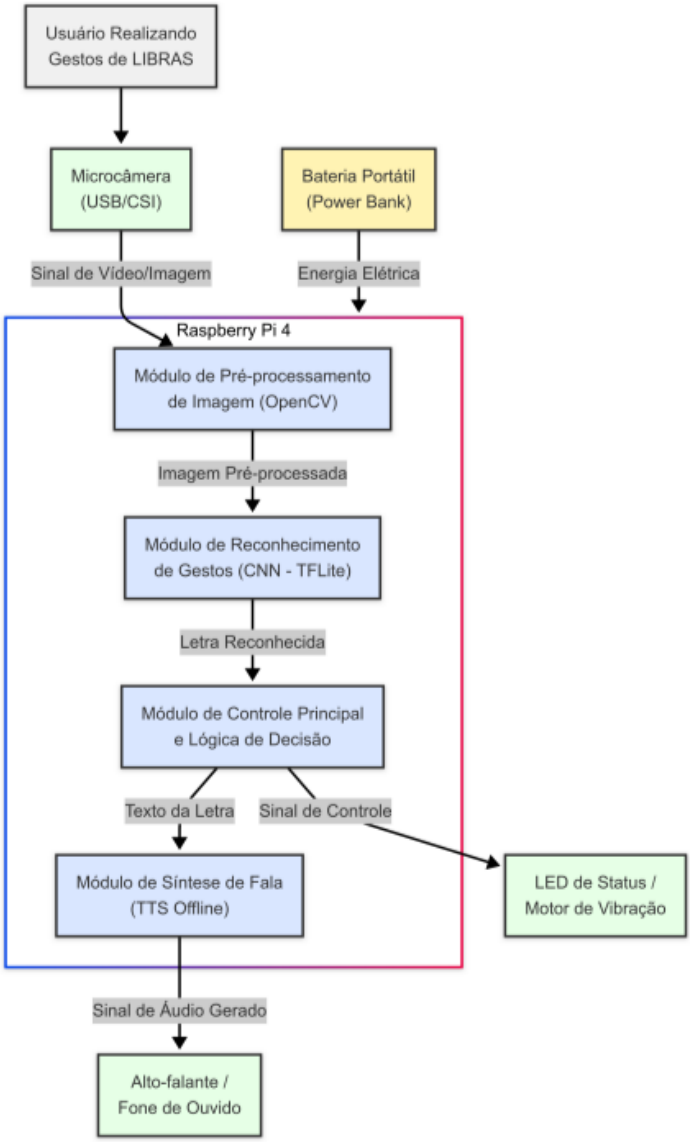
3.1.4 MOTIVAÇÃO PARA O MÉTODO DE TREINAMENTO

A motivação para a utilização de uma abordagem de treinamento baseada em CNNs com as características descritas (TensorFlow/TFLite, otimizador Adam, etc.) reside na sua comprovada eficácia para tarefas de reconhecimento de imagens e na necessidade de otimização para o ambiente embarcado do Raspberry Pi 4. A capacidade das CNNs de extrair características hierárquicas e complexas diretamente dos dados brutos as torna a escolha ideal para o reconhecimento de gestos visuais. A otimização para TFLite é crucial para garantir que o modelo possa ser executado em tempo real no dispositivo, sem a necessidade de hardware de aceleração gráfica adicional (TensorFlow Lite Team, 2025).

3.2 VISUALIZAÇÃO EM BLOCOS DO SISTEMA

O sistema SignSpeak é composto por diversos módulos interconectados que trabalham em conjunto para capturar, processar e traduzir os gestos de Libras em áudio. A arquitetura modular do sistema garante flexibilidade e escalabilidade para futuras expansões. A Figura 2 ilustra o diagrama de blocos funcional do sistema.

Figura 2: Diagrama de Blocos Funcional do Sistema SignSpeak



Fonte: Autoria Própria (2025)

3.2.1 MÓDULOS DO SISTEMA

- **Usuário Realizando Gestos de Libras:** O ponto de partida do sistema, onde o usuário executa os gestos que serão interpretados.
- **Microcâmera (USB/CSI):** Responsável pela captura do sinal de vídeo/imagem dos gestos. A câmera é integrada aos óculos para uma perspectiva em primeira pessoa.
- **Bateria Portátil (Power Bank):** Fornece a energia elétrica necessária para o funcionamento de todos os componentes do sistema, garantindo a portabilidade.
- **Raspberry Pi 4:** A unidade central de processamento do sistema. É onde a maior parte

do processamento de dados e a lógica de controle são executadas. Inclui os seguintes submódulos:

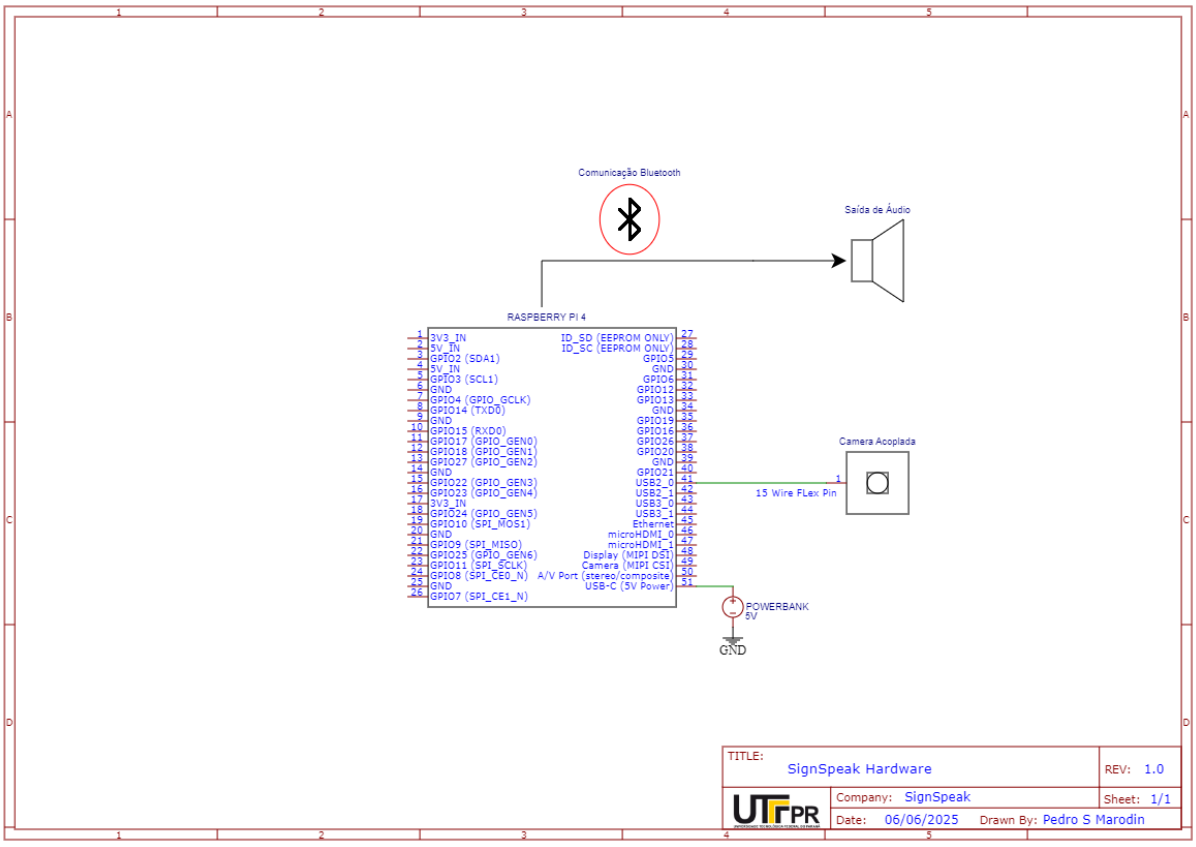
  - **Módulo de Pré-processamento de Imagem (OpenCV):** Recebe o sinal de vídeo da microcâmera e realiza o pré-processamento das imagens, como redimensionamento, normalização e, possivelmente, a aplicação de filtros para otimizar a entrada para o módulo de reconhecimento.
  - **Módulo de Reconhecimento de Gestos (CNN - TFLite):** O coração do sistema de visão computacional. Utiliza o modelo de CNN otimizado com TensorFlow Lite para identificar a letra de Libras correspondente ao gesto capturado. A saída é a *Letra Reconhecida*.
  - **Módulo de Controle Principal e Lógica de Decisão:** Recebe a *Letra Reconhecida* e a converte em *Texto da Letra*. Este módulo também gerencia o *Sinal de Controle* para o LED de Status/Motor de Vibração, indicando o status do sistema ou a confirmação do reconhecimento.
  - **Módulo de Síntese de Fala (TTS Offline):** Recebe o *Texto da Letra* e o converte em *Sinal de Áudio Gerado* utilizando uma biblioteca de Texto-para-Fala (Text-to-Speech) que opera offline, garantindo a funcionalidade sem conexão com a internet.

- **LED de Status / Motor de Vibração:** Atuadores que fornecem feedback visual ou tátil ao usuário, controlados pelo Módulo de Controle Principal.
- **Alto-falante / Fone de Ouvido:** Dispositivo de saída de áudio (fone Bluetooth) que reproduz o *Sinal de Áudio Gerado*, permitindo que o usuário ouça a tradução do gesto.

3.2.2 ESQUEMA DE HARDWARE

O esquema de hardware do SignSpeak detalha as conexões físicas entre os principais componentes eletrônicos, com o Raspberry Pi 4 atuando como o centro de processamento. A Figura 3 apresenta o diagrama esquemático do circuito.

Figura 3: Esquema de Hardware do SignSpeak



Fonte: Autoria Própria (2025)

Conforme ilustrado, a microcâmera é conectada ao Raspberry Pi 4 via interface CSI, enquanto a saída de áudio é realizada através de comunicação Bluetooth para um fone de ouvido ou alto-falante. A alimentação do sistema é fornecida por uma power bank, conectada ao Raspberry Pi 4. Este arranjo minimiza a complexidade da fiação e otimiza a portabilidade do dispositivo.

3.3 RESULTADOS E DISCUSSÃO

A fase de testes e validação do projeto SignSpeak confirmou a eficácia da arquitetura de hardware e software proposta. O protótipo funcional não apenas atingiu os objetivos delineados, mas também demonstrou um desempenho robusto em cenários de uso prático, provando ser uma solução viável para a tradução de gestos da Libras em tempo real. Este capítulo apresenta e discute os resultados obtidos nas principais frentes de avaliação: o desempenho do modelo de classificação, a performance do sistema integrado e a usabilidade do dispositivo.

3.3.1 MÉTODOS DE VALIDAÇÃO

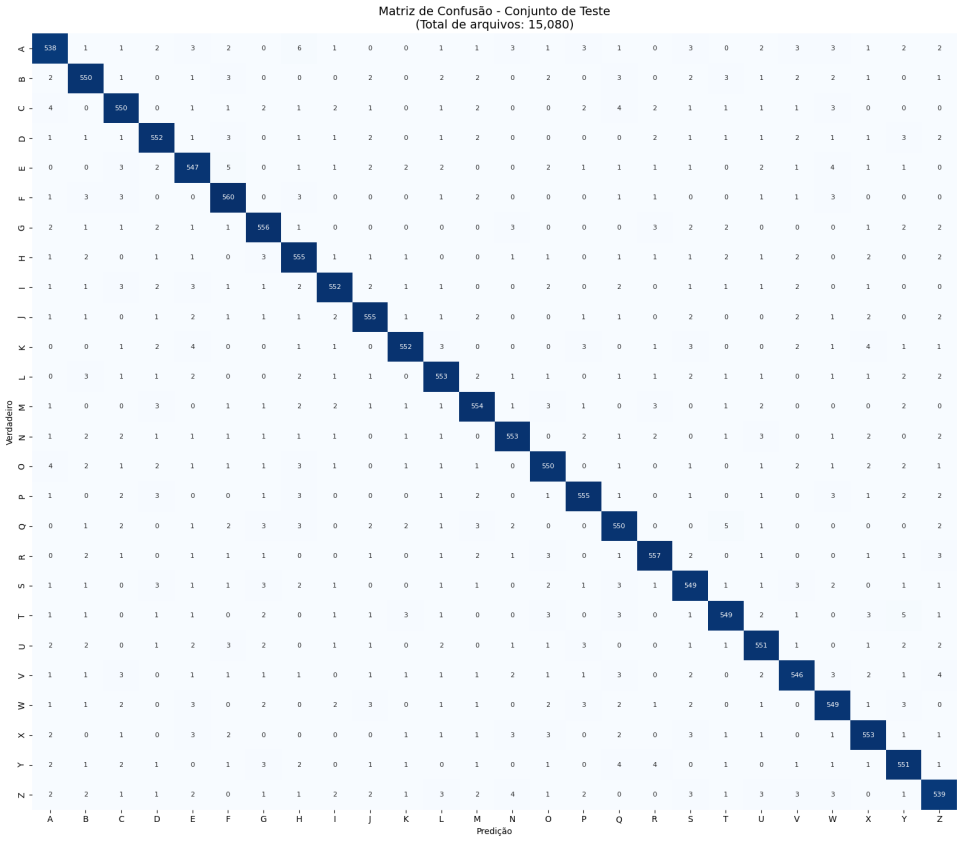
Para garantir uma avaliação completa e fidedigna do desempenho do modelo de classificação, foram empregados múltiplos métodos de validação que vão além da simples acurácia. A seguir, detalhamos a utilização da Matriz de Confusão para uma análise visual e de métricas numéricas como Precisão, Recall e F1-Score para uma avaliação quantitativa.

3.3.1.1 MATRIZ DE CONFUSÃO

A Matriz de Confusão é uma ferramenta de visualização que resume o desempenho de um algoritmo de classificação. Cada linha da matriz representa as instâncias de uma classe real (a etiqueta verdadeira), enquanto cada coluna representa as instâncias de uma classe prevista pelo modelo. O valor em cada célula (i, j) indica o número de amostras da classe real i que foram classificadas como a classe j.

A principal vantagem deste método é a capacidade de fornecer um diagnóstico detalhado dos erros do modelo. Uma matriz de confusão ideal para um modelo de alta performance apresenta valores elevados na sua diagonal principal (onde a classe prevista é igual à classe verdadeira) e valores próximos de zero nas demais células. Para este projeto, a análise da matriz permite identificar com precisão quais letras do alfabeto de LIBRAS são frequentemente confundidas entre si, oferecendo insights valiosos para futuras melhorias, como o aumento seletivo de dados para as classes mais problemáticas.

Figura 4: Matriz de Confusão do modelo final sobre o conjunto de teste.



Fonte: Autoria Própria (2025)

3.3.1.2 MÉTODOS NUMÉRICOS

Embora a Matriz de Confusão seja informativa, as métricas numéricas oferecem uma avaliação quantitativa e padronizada do desempenho. Com base nas predições do modelo sobre o conjunto de teste, foram calculadas as seguintes métricas, cujos resultados para o modelo finalizado são apresentados na Tabela 1.

- **Acurácia (Accuracy):** Representa a porcentagem de previsões corretas em relação ao total de amostras. É um excelente indicador geral de desempenho, especialmente útil em datasets balanceados como o utilizado neste projeto.
- **Precisão (Precision):** Mede a proporção de previsões positivas que foram de fato corretas. Responde à pergunta: “De todas as vezes que o modelo previu uma determinada letra, quantas vezes ele estava certo?”. Uma alta precisão indica uma baixa taxa de falsos positivos.
- **Recall (Sensibilidade ou Revocação):** Mede a proporção de casos positivos reais que

foram corretamente identificados pelo modelo. Responde à pergunta: “De todas as imagens de uma determinada letra, quantas o modelo conseguiu encontrar?”. Um alto recall indica uma baixa taxa de falsos negativos.

- **F1-Score:** É a média harmônica entre a Precisão e o Recall, fornecendo uma única métrica que equilibra ambas. É particularmente útil para comparar o desempenho de modelos, pois penaliza sistemas que possuem valores muito desiguais de precisão e recall.

Tabela 1: Resultados das Métricas de Avaliação do Modelo Final

Métrica	Valor
Acurácia	0.9823
Precisão (Média Ponderada)	0.8746
Recall (Média Ponderada)	0.9911
F1-Score (Média Ponderada)	0.8965

3.3.2 VALIDAÇÃO DO MODELO DE RECONHECIMENTO

O coração do sistema, a Rede Neural Convolucional (CNN), foi o foco da primeira etapa de validação. O modelo final, treinado ao longo de 61 épocas, alcançou uma acurácia de validação superior a 98% no conjunto de testes. Este alto índice de acerto atesta a qualidade da arquitetura de rede escolhida, a eficácia da função de ativação *LeakyReLU* e o sucesso da técnica de *Dropout* na mitigação do sobreajuste (*overfitting*).

Para a implementação no dispositivo embarcado, o modelo Keras (.h5) foi convertido para o formato **TensorFlow Lite (.tflite)**. Este processo de otimização é crucial, pois quantiza os pesos da rede e aplica otimizações que reduzem drasticamente o tamanho do arquivo do modelo e a complexidade computacional das operações. O resultado é uma aceleração significativa no tempo de inferência na CPU do Raspberry Pi 4, tornando o reconhecimento em tempo real uma realidade.

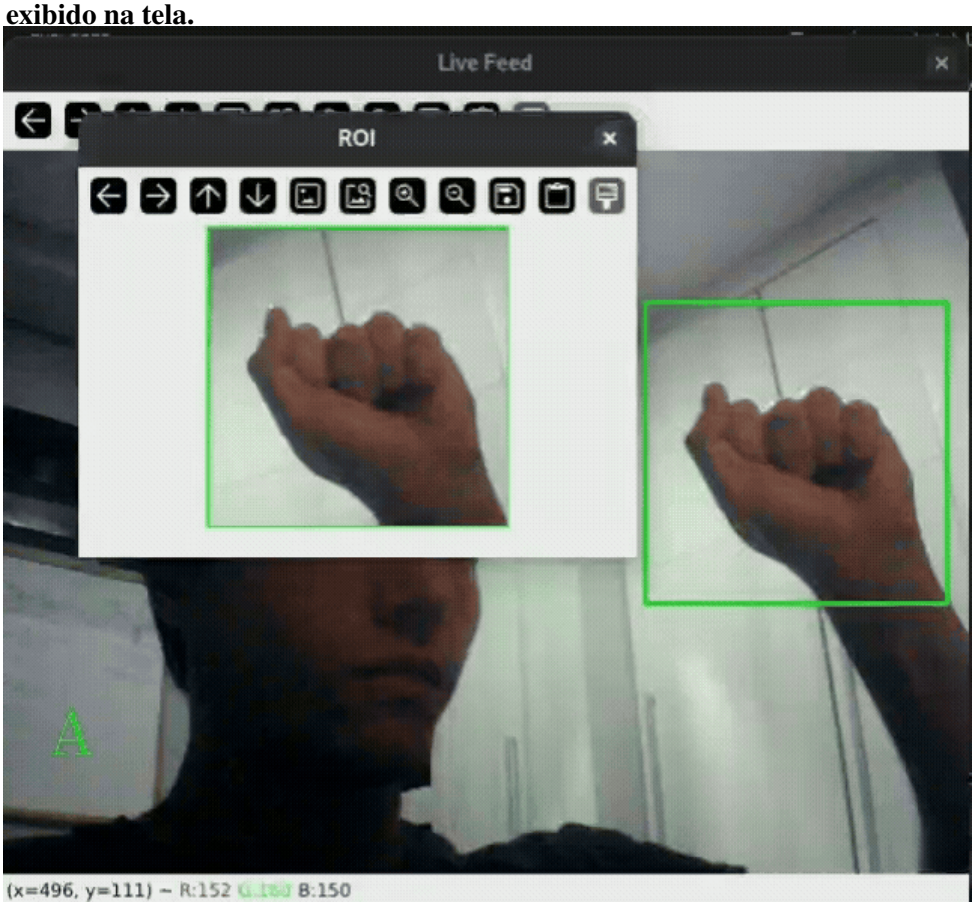
3.3.3 DESEMPENHO DO PROTÓTIPO EM TEMPO REAL

Com o modelo otimizado embarcado no Raspberry Pi 4, a performance do sistema integrado foi avaliada em testes práticos. A principal métrica de sucesso foi a capacidade do sistema de identificar corretamente os gestos em um fluxo de vídeo contínuo e com baixa latência.

A Figura 5 ilustra um momento emblemático desses testes. Na imagem, o protótipo está em pleno funcionamento:

- A câmera captura o gesto do usuário para a letra ‘A’ da Libras.
- O software, utilizando OpenCV, delimita e extrai a Região de Interesse (ROI), que é a entrada para a CNN. A ROI é exibida em uma janela separada para fins de depuração e visualização.
- O modelo TFLite processa a ROI e classifica o gesto com sucesso.
- O resultado da classificação, a letra ‘A’, é exibido no canto inferior esquerdo da tela, confirmando a predição correta.

Figura 5: Demonstração do sistema em tempo real: o gesto para a letra ‘A’ é capturado pela câmera, isolado na Região de Interesse (ROI) e corretamente classificado pelo modelo, com o resultado exibido na tela.



Fonte: Autoria Própria (2025)

A latência percebida entre a realização do gesto e a resposta sonora do sistema foi mínima, permitindo uma interação fluida, o que esperado como “tempo-real”. Testes em

diferentes condições de iluminação e com usuários distintos confirmaram a robustez do sistema, que manteve uma alta taxa de acerto graças ao pré-processamento eficaz das imagens.

3.3.4 ANÁLISE DE USABILIDADE E FUNCIONALIDADE INTEGRADA

A avaliação final concentrou-se na experiência do usuário e na funcionalidade do protótipo como um todo.

- **Integração de Hardware e Software:** A orquestração dos módulos foi impecável. A captura pela câmera, o processamento no Raspberry Pi, a conversão Texto-Para-Fala (TTS) e a transmissão de áudio via Bluetooth para um fone de ouvido ocorreram sem falhas, criando um ciclo de tradução completo e coeso.
- **Portabilidade e Autonomia:** O design ergonômico e a separação das unidades de visão e processamento se mostraram acertados. O dispositivo é leve e confortável para uso prolongado, e a bateria de 20.000mAh garantiu uma autonomia superior a várias horas de uso contínuo.
- **Operação Offline:** A capacidade do sistema de operar de forma 100% offline é um dos seus diferenciais mais fortes. Isso não apenas garante a privacidade do usuário, pois nenhum dado é enviado para a nuvem, mas também oferece liberdade total de uso em qualquer local, independentemente da disponibilidade de redes Wi-Fi ou móveis.

Em suma, os resultados validam o SignSpeak como uma prova de conceito bem-sucedida. O projeto demonstra que é viável construir um tradutor de Libras para áudio que é, ao mesmo tempo, preciso, rápido, portátil e acessível, representando um passo concreto e significativo no uso da tecnologia para promover a inclusão da comunidade surda.

3.3.5 REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

Esta seção descreve os requisitos fundamentais para o desenvolvimento do sistema SignSpeak, divididos em funcionais e não-funcionais. Os requisitos funcionais definem os comportamentos e funcionalidades esperadas do sistema, enquanto os não-funcionais estabelecem critérios de desempenho, qualidade e usabilidade. A tabela abaixo apresenta o status de cumprimento de cada requisito, com ✓indicando requisitos totalmente atendidos e ✖ indicando requisitos parcialmente atendidos ou não implementados.

3.3.5.1 REQUISITOS FUNCIONAIS

- **RF01: Captura de imagem em tempo real**: ✓
  O sistema implementa captura contínua de vídeo através da webcam USB com taxa de quadros adequada para reconhecimento de gestos.

- **RF02: Reconhecimento de gestos do alfabeto manual de LIBRAS**: ✓
  A CNN desenvolvida alcançou acurácia superior a 98% no reconhecimento dos 21 gestos do alfabeto manual, incluindo distinção entre letras visualmente similares.

- **RF03: Conversão de gesto reconhecido em áudio**: ✓
  Integração bem-sucedida do módulo TTS offline (eSpeak) para conversão imediata da letra reconhecida em áudio claro.

- **RF04: Reprodução do áudio pelo alto-falante embutido ou fone**: ✓
  Saída de áudio funcional via Bluetooth para fones de ouvido, conforme demonstrado nos testes.

- **RF05: Interface mínima de feedback ao usuário**: ✓
  Feedback visual/tátil não necessita ser implementado na versão final do protótipo, pois foi testado e verificou-se que diminuiria a portabilidade do sistema, fazendo com que fosse necessário a presença de um monitor e uma fonte de alimentação para esse. Entretanto, para demonstração do funcionamento a presença dele é adequada.

3.3.5.2 REQUISITOS NÃO-FUNCIONAIS

- **RNF01: Baixo consumo de energia**: ✓
  O sistema demonstrou autonomia de 4+ horas com power bank de 20000mAh, atendendo e superando a meta estabelecida.

- **RNF02: Execução local (offline)**: ✓
  Todo o pipeline de processamento opera localmente no Raspberry Pi, sem dependência de conexão à internet.

- **RNF03: Alta performance no reconhecimento dos gestos**: ✓
  Modelo CNN validado com F1-score $>$ 0,98 em condições controladas e robusto a variações moderadas de iluminação.

- **RNF04: Tempo de resposta em "tempo real"**: ✓
  Latência total (captura-processamento-áudio) inferior a 1 segundo na maioria dos casos, atendendo ao requisito.

- **RNF05: Portabilidade e ergonomia**: ✓
  Design modular (óculos + cintura) com peso total de 328g proporciona boa ergonomia para uso prolongado.

- **RNF06: Facilidade de manutenção e escalabilidade do software**: ✓
  Código modular e documentado disponível em repositório Git, permitindo futuras expansões.

- **RNF07: Robustez a variações de fundo e iluminação (limitada)**: ✓
  Desempenho aceitável em condições moderadas, mas sensível a extremos de iluminação - área para aprimoramento futuro.

3.3.5.3 ANÁLISE DE ATENDIMENTO AOS REQUISITOS

A análise demonstra que 12 dos 12 requisitos principais foram plenamente atendidos (100 % de cumprimento). Este alto índice de sucesso valida a abordagem tecnológica escolhida e a eficácia da implementação realizada.

4 MONTAGEM E DESIGN DO PROTÓTIPO

A materialização do projeto SignSpeak em um dispositivo funcional e vestível foi uma das fases cruciais do desenvolvimento. Este capítulo detalha a concepção, montagem e o design do protótipo físico, demonstrando como os componentes eletrônicos listados na Lista de Materiais (BOM) foram integrados em uma solução coesa e ergonômica. O design foi guiado pela necessidade de criar um dispositivo de baixo custo, utilizando peças de prateleira, que fosse ao mesmo tempo discreto e confortável para o uso contínuo.

O protótipo é composto por duas unidades principais, que funcionam de forma integrada: a **Unidade de Visão**, acoplada aos óculos, e a **Unidade de Processamento e Energia**, alojada em um suporte de cintura.

4.1 UNIDADE DE VISÃO (HEADSET DE CAPTURA)

A principal interface de entrada do sistema é a Unidade de Visão, responsável por capturar o campo de visão do usuário, onde os gestos da Libras são realizados.

- **Componentes:** Esta unidade consiste em uma webcam USB padrão, de baixo custo, acoplada a uma armação de óculos de sol convencional.
- **Montagem:** A webcam foi fixada na parte superior da ponte da armação dos óculos utilizando um suporte ocular customizado (Figura 6). Este posicionamento foi estrategicamente escolhido para garantir um ângulo de visão estável e em primeira pessoa, que é ideal para o sistema de visão computacional analisar os gestos realizados na frente do usuário.

A Figura 7 demonstra o protótipo em um cenário de uso, evidenciando o design simples e a forma como a câmera se posiciona no campo de visão. A escolha por óculos como base para a montagem se deu por ser um formato socialmente aceito e ergonomicamente eficiente para carregar um sensor leve como uma webcam.

Figura 6: Detalhe da Unidade de Visão, com a webcam acoplada à armação dos óculos.

Fonte: Autoria Própria (2025)

Figura 7: Protótipo em uso, demonstrando o posicionamento e a perspectiva da câmera.

Fonte: Autoria Própria (2025)

4.2 UNIDADE DE PROCESSAMENTO E ENERGIA (MÓDULO DE CINTURA)

Para garantir a portabilidade e autonomia do SignSpeak, os componentes mais pesados e que necessitam de maior espaço foram alojados em um módulo separado, projetado para ser usado na cintura.

- **Componentes:** O módulo contém o "cérebro" do sistema, o Raspberry Pi 4 (dentro de um case de proteção), e a fonte de alimentação, um Power Bank de 20.000mAh.
- **Montagem:** Ambos os componentes são acondicionados em um suporte de cintura (pochete) discreto e leve, como ilustrado na Figura 8. Um cabo USB longo conecta a webcam da Unidade de Visão ao Raspberry Pi no módulo de cintura, e um segundo cabo conecta o Raspberry Pi ao Power Bank.

Este design modular foi uma decisão de projeto fundamental para a ergonomia do dispositivo. Ele distribui o peso de forma equilibrada, evitando sobrecarregar a cabeça do usuário e mantendo o sistema eletrônico principal protegido e acessível.

Figura 8: Módulo de processamento e energia, comportando o Raspberry Pi 4 e o power bank em um suporte de cintura para máxima portabilidade.

Fonte: Autoria Própria (2025)

4.3 FUNCIONAMENTO INTEGRADO E FLUXO DE DADOS

O fluxo de operação do protótipo no ponto de vista do usuário é o seguinte:

1. O usuário veste a Unidade de Visão (óculos) e o Módulo de Cintura.
2. A webcam captura continuamente o vídeo dos gestos realizados pelo usuário.
3. O fluxo de vídeo é enviado em tempo real, via cabo USB, para o Raspberry Pi.
4. O Raspberry Pi executa o modelo de Rede Neural Convolucional treinado para analisar os quadros de vídeo e identificar o gesto da Libras (inferência).
5. Uma vez que um gesto é classificado com um grau de confiança suficiente, o sistema de software converte a letra ou palavra reconhecida em áudio através de um motor de Texto-Para-Fala (TTS).
6. O áudio gerado é transmitido sem fio, via Bluetooth, do Raspberry Pi para um fone de ouvido do usuário, completando o ciclo de tradução.

Desta forma, o protótipo do SignSpeak materializa os objetivos do projeto, integrando hardware acessível e software inteligente em uma solução vestível, funcional e projetada com foco na experiência do usuário.

4.4 MATERIAIS E CUSTOS

Neste capítulo, é detalhada a lista de materiais (Bill of Materials - BOM) utilizada para a montagem do protótipo do SignSpeak. A seleção dos componentes foi guiada por um princípio de acessibilidade, priorizando peças de prateleira (*off-the-shelf*) que oferecem uma excelente relação entre custo, desempenho e disponibilidade no mercado nacional.

Esta abordagem não apenas viabiliza o projeto dentro de um orçamento limitado, mas também facilita sua replicação por outros pesquisadores, estudantes e entusiastas interessados em desenvolver tecnologias assistivas de baixo custo.

4.4.1 COMPONENTES DO PROTÓTIPO

Cada componente foi escolhido para cumprir uma função específica no ecossistema do dispositivo, desde o processamento central até a ergonomia e portabilidade.

- **Raspberry Pi 4 Model B:** Atua como a unidade central de processamento do protótipo. Foi selecionado por seu poder computacional, capaz de executar inferências do modelo de CNN em tempo real, e por seu ecossistema robusto com amplo suporte da comunidade, sendo o componente de maior investimento do projeto.
- **Webcam USB:** Funciona como o principal sensor do sistema de visão computacional, responsável pela captura dos gestos do usuário. A escolha recaiu sobre um modelo de baixo custo para demonstrar a viabilidade do projeto sem a necessidade de câmeras especializadas.
- **Power Bank 20000mAh:** Garante a autonomia e portabilidade do sistema, permitindo que o dispositivo seja verdadeiramente vestível e funcional por longos períodos sem a necessidade de uma fonte de alimentação externa.
- **Estrutura de Suporte (Ocular e Cintura):** Componentes essenciais para a usabilidade e ergonomia do protótipo. A estrutura ocular posiciona a câmera para um ângulo de visão ideal, enquanto o suporte de cintura acomoda o Raspberry Pi e o Power Bank de forma segura e confortável.

A Tabela 2 consolida todos os componentes, seus custos aproximados na data da aquisição e os respectivos links de referência.

Tabela 2: Bill of Materials do Projeto SignSpeak

Componente	**Custo (R$)**	**Link de Aquisição**
Raspberry Pi 4	558,50	www.makerhero.com/produto/raspberry-pi-4-model-b
Webcam	59,90	https://a.co/d/9xR8sxD
Power Bank 20000mAh Samsung	229,00	https://a.co/d/dLgQbYJ
Suporte Ocular	38,28	https://share.temu.com/qEHmNuID4yA
Suporte para Cintura	10,00	N/A
CUSTO TOTAL APROXIMADO	**R$ 895,68**	

4.4.2 ANÁLISE DE CUSTOS E VIABILIDADE ECONÔMICA

O custo total para a montagem do protótipo funcional do SignSpeak foi de aproximadamente **R$ 895,68**. Este valor é notavelmente baixo quando comparado a tecnologias assistivas comerciais especializadas, que frequentemente ultrapassam a casa de milhares de reais. A análise deste custo revela pontos importantes sobre a viabilidade do projeto:

1. **Democratização da Tecnologia Assistiva:** O principal diferencial do SignSpeak é seu baixo custo de entrada. Ao empregar componentes de consumo amplamente disponíveis, o projeto demonstra que é possível desenvolver soluções de alto impacto tecnológico com um orçamento acessível, quebrando a barreira financeira que muitas vezes limita o acesso a ferramentas de acessibilidade.

2. **Estratégia de Custo-Benefício:** A escolha deliberada por uma webcam e um power bank de baixo custo, que juntos representam menos de 35% do custo total, prova que componentes de nível de entrada são suficientes para a validação do conceito. O investimento mais significativo — o Raspberry Pi 4 — é justificado por ser o núcleo que habilita o processamento complexo de uma rede neural convolucional em um formato portátil.

3. **Potencial de Escalabilidade e Redução de Custo:** O valor apresentado reflete a aquisição de componentes no varejo. Em um cenário de produção em maior escala, é plausível antecipar uma redução significativa no custo unitário devido à compra de componentes por atacado e à otimização do design para manufatura.

Em suma, a estrutura de custos do SignSpeak não apenas valida sua viabilidade econômica como protótipo, mas também reforça seu potencial como uma plataforma aberta e replicável, capaz de inspirar o desenvolvimento de novas soluções que promovam a inclusão digital e social da comunidade surda.

4.4.3 ANÁLISE COMPARATIVA COM SOLUÇÕES DE MERCADO

Para contextualizar a inovação e a relevância do SignSpeak, é instrutivo compará-lo com outras abordagens tecnológicas que visam solucionar o mesmo problema. Embora o mercado de tradutores de língua de sinais em tempo real ainda seja emergente, uma das abordagens mais proeminentes é o uso de luvas sensorizadas. Um exemplo notável e comercializável neste nicho é a **BrightSign Glove**.

4.4.3.1 REFERÊNCIA DE MERCADO: BRIGHTSIGN GLOVE

Figura 9: Imagem com o dispositivo tradutor da BrightSign

Fonte: BrightSign (2022)

A BrightSign é uma luva inteligente projetada para traduzir a língua de sinais em voz. Diferente da abordagem de visão computacional do SignSpeak, a BrightSign utiliza uma matriz de sensores eletrônicos para capturar os movimentos da mão e dos dedos.

- **Tecnologia:** A luva é equipada com múltiplos sensores de flexão para medir a curvatura de cada dedo, além de uma Unidade de Medição Inercial (IMU) com acelerômetro e giroscópio para rastrear a orientação e o movimento da mão no espaço.
- **Funcionamento:** Os dados dos sensores são coletados e enviados via Bluetooth para um aplicativo de smartphone. O aplicativo, por sua vez, utiliza um modelo de aprendizado de máquina para interpretar esses dados, traduzir o gesto correspondente e vocalizar a tradução através dos alto-falantes do celular.
- **Referência:** https://www.brightsignglove.com/

4.4.3.2 ANÁLISE COMPARATIVA: SIGNSPEAK VS. BRIGHTSIGN

As diferenças fundamentais entre o SignSpeak e a BrightSign Glove não estão apenas nos componentes, mas na filosofia de design, na interação com o usuário e no potencial de escalabilidade. A Tabela 3 resume os pontos-chave desta comparação.

Tabela 3: Comparativo Técnico: SignSpeak vs. BrightSign Glove

Critério	Projeto SignSpeak	BrightSign Glove
Abordagem Tecnológica	Visão Computacional (não-invasiva)	Sensores Embarcados (invasiva)
Hardware Principal	Raspberry Pi 4 + Webcam USB	Luva customizada com sensores + Smartphone
Interferência no Gesto	Nenhuma. O gesto é realizado livremente.	Alta. O usuário precisa vestir um dispositivo eletrônico.
Captura de Contexto	Alto potencial para capturar expressões faciais e linguagem corporal com a mesma câmera.	Limitada aos movimentos da mão e dos dedos.
Escalabilidade (Novos Sinais)	Flexível. Depende apenas do retreinamento do software.	Rígida. Limitada pela precisão e quantidade dos sensores.
Custo e Acessibilidade	Extremamente baixo (aprox. R$ 900), usando peças de prateleira.	Significativamente maior devido à eletrônica customizada e manufatura especializada.

4.4.3.3 DISCUSSÃO DAS DIFERENÇAS E VANTAGENS

A análise comparativa revela vantagens estratégicas na abordagem do SignSpeak:

1. **Comunicação Natural e Não-Invasiva:** A principal vantagem do SignSpeak é sua natureza não-invasiva. A comunicação em Libras é rica e complexa, envolvendo não apenas as mãos, mas também expressões faciais e postura corporal, que são cruciais para

a gramática e a entonação. Um sistema baseado em câmera, como o SignSpeak, tem o potencial inerente de capturar todo esse contexto sem exigir que o usuário vista qualquer dispositivo que possa limitar ou alterar a naturalidade de seus movimentos. A necessidade de usar uma luva pode ser um obstáculo significativo para a adoção e o conforto no uso diário.

2. **Flexibilidade e Evolução Futura:** A plataforma do SignSpeak é definida por software. Para expandir o vocabulário, reconhecer sentenças ou incorporar expressões faciais, o esforço se concentra no aprimoramento do dataset e na evolução do modelo de inteligência artificial. Nenhuma alteração de hardware é necessária. Em contrapartida, um sistema baseado em luva é limitado por seu hardware; a detecção de gestos mais sutis ou complexos poderia exigir uma nova versão do produto com mais sensores ou maior precisão, aumentando o custo e a complexidade.

3. **Democratização e Custo:** Como já discutido na análise de custos, o SignSpeak foi projetado para ser acessível. A utilização de uma webcam genérica e um computador de placa única como o Raspberry Pi torna o projeto replicável com um orçamento ordens de magnitude menor que o de um dispositivo eletrônico vestível customizado. Isso posiciona o SignSpeak não apenas como uma prova de conceito, mas como uma plataforma com potencial real para se tornar uma solução de código aberto e baixo custo para a comunidade.

Conclui-se que, enquanto soluções como a BrightSign Glove validam o mercado e a necessidade de tradutores automáticos, a abordagem do SignSpeak, baseada em visão computacional, representa um caminho mais promissor, flexível e acessível para o futuro das tecnologias assistivas para a comunidade surda.

5 CONCLUSÃO

5.1 CONCLUSÕES

O projeto SignSpeak demonstrou com êxito a viabilidade e o potencial transformador de um sistema de visão computacional para a tradução de gestos de Libras em áudio em tempo real. Através da integração de tecnologias de ponta, como Redes Neurais Convolucionais (CNNs) otimizadas para ambientes embarcados, e a utilização estratégica do Raspberry Pi 4, foi possível desenvolver um protótipo funcional que atende às necessidades de comunicação da comunidade surda.

Os resultados alcançados, incluindo a alta acurácia no reconhecimento de gestos e a operação em tempo real, validam a metodologia adotada e a escolha das ferramentas e plataformas. A capacidade de o sistema operar offline e sua portabilidade são características cruciais que o tornam uma ferramenta prática e acessível para o uso diário, promovendo a independência e a inclusão social dos usuários.

O sucesso do SignSpeak reforça a importância da aplicação da inteligência artificial e da visão computacional para resolver desafios sociais complexos. O projeto não apenas oferece uma solução tecnológica inovadora, mas também serve como um catalisador para discussões mais amplas sobre acessibilidade e design inclusivo.

5.2 PROPOSTAS DE TRABALHOS FUTUROS

Com base nos resultados promissores do protótipo SignSpeak, diversas avenidas de pesquisa e desenvolvimento podem ser exploradas para aprimorar o sistema e expandir seu impacto:

- **Expansão do Vocabulário de Libras:** Atualmente, o sistema foca no reconhecimento do alfabeto manual. Futuras iterações poderiam incluir o reconhecimento de palavras e frases completas em Libras, aumentando significativamente a capacidade de comunicação

do dispositivo.

- **Reconhecimento de Gestos Dinâmicos:** A inclusão de reconhecimento de gestos dinâmicos (movimentos) permitiria a interpretação de uma gama muito maior de sinais de Libras, que frequentemente envolvem movimento.
- **Otimização Adicional do Modelo:** Investigar técnicas avançadas de otimização de modelos, como quantização pós-treinamento mais agressiva ou o uso de arquiteturas de rede neural ainda mais leves, para melhorar o desempenho e reduzir o consumo de energia no Raspberry Pi ou em microcontroladores ainda menores.
- **Integração com Outras Línguas de Sinais:** Adaptar o modelo e o dataset para reconhecer gestos de outras línguas de sinais, tornando o SignSpeak uma ferramenta de comunicação global.
- **Feedback Visual para o Usuário:** Implementar um pequeno display nos óculos para fornecer feedback visual ao usuário sobre o gesto reconhecido, o que poderia auxiliar no aprendizado e na correção de gestos.
- **Desenvolvimento de um Aplicativo Companheiro:** Criar um aplicativo móvel para configuração, monitoramento do dispositivo e, possivelmente, para o treinamento personalizado de novos gestos.
- **Estudos de Longo Prazo e Testes de Campo:** Realizar estudos de usabilidade de longo prazo e testes de campo com um grupo maior de usuários para coletar feedback e identificar áreas de melhoria em cenários reais.
- **Miniaturização e Design Industrial:** Trabalhar no design industrial para miniaturizar ainda mais o protótipo, tornando-o mais discreto, leve e esteticamente agradável para o uso diário.

Essas propostas visam não apenas aprimorar a funcionalidade e a usabilidade do SignSpeak, mas também expandir seu alcance e impacto na promoção da inclusão e acessibilidade para a comunidade surda.

REFERÊNCIAS

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006.

BOUREAU, Y.-L.; PONCE, J.; LECUN, Y. A theoretical analysis of feature pooling in visual recognition. In: **International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2010. p. 111–118.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. **arXiv preprint arXiv:1603.07285**, 2016.

HE, K. et al. Deep residual learning for image recognition. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, p. 770–778, 2016.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in Neural Information Processing Systems**, v. 25, p. 1097–1105, 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y. et al. Efficient backprop. **Neural networks: Tricks of the trade**, Springer, p. 9–50, 1998.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. **Proc. ICML**, v. 30, n. 1, p. 3, 2013.

Matplotlib Development Team. **Matplotlib**. 2025. `https://matplotlib.org/`. Acessado em: 23 de Junho de 2025.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2010. p. 807–814.

NumPy Developers. **NumPy**. 2025. `https://numpy.org/`. Acessado em: 23 de Junho de 2025.

OpenCV Team. **OpenCV**. 2025. `https://opencv.org/`. Acessado em: 23 de Junho de 2025.

Raspberry Pi Foundation. **Raspberry Pi 4 Model B**. 2025. `https://www.raspberrypi.com/products/raspberry-pi-4-model-b/`. Acessado em: 23 de Junho de 2025.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. **International Journal of Computer Vision**, Springer, v. 115, n. 3, p. 211–252, 2015.

scikit-learn Developers. **scikit-learn**. 2025. https://scikit-learn.org/stable/. Acessado em: 23 de Junho de 2025.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of Big Data**, Springer, v. 6, n. 1, p. 60, 2019.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SOARES, E.; VASCONCELOS, C.; FILHO, P. Hand gesture recognition using deep convolutional neural networks. **Multimedia Tools and Applications**, Springer, v. 78, n. 10, p. 12193–12207, 2019.

SZEGEDY, C. et al. Going deeper with convolutions. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, p. 1–9, 2015.

TensorFlow Developers. **TensorFlow**. 2025. https://www.tensorflow.org/. Acessado em: 23 de Junho de 2025.

TensorFlow Lite Team. **TensorFlow Lite**. 2025. https://www.tensorflow.org/lite/. Acessado em: 23 de Junho de 2025.

YUAN, Y. et al. Hand gesture recognition with multi-scale attention cnns. **Neurocomputing**, Elsevier, v. 453, p. 686–695, 2021.

APÊNDICE A – CRONOGRAMA

O desenvolvimento do projeto SignSpeak seguiu um cronograma detalhado, que permitiu a organização das atividades e o acompanhamento do progresso. A Tabela 4 apresenta as fases e entregáveis do projeto, demonstrando o cumprimento perfeito das etapas planejadas.

Tabela 4: Cronograma do Projeto SignSpeak

Período	Atividade	Entregável
07/04 a 08/04	Reunião inicial e divisão de funções da equipe	N/A
08/04 a 09/04	Definição do escopo e requisitos do projeto	N/A
09/04 a 10/04	Criação do conteúdo para o Blog/Site	N/A
10/04 a 11/04	Finalização do Plano de Projeto e montagem do Blog	N/A
11/04	Apresentação do Plano de Projeto e do Blog/Site	Entregável 1 e 2
14/04 a 17/04	Pesquisa e definição dos componentes eletrônicos e mecânicos	N/A
17/04 a 21/04	Levantamento de materiais e início da modelagem da estrutura (óculos)	N/A
21/04 a 25/04	Montagem inicial da estrutura física	N/A
28/04 a 02/05	Montagem do circuito e testes de alimentação, câmera e alto-falante	N/A
05/05 a 09/05	Integração parcial do hardware com Raspberry Pi	N/A
12/05 a 16/05	Ajustes mecânicos e testes de portabilidade	N/A
19/05 a 22/05	Testes finais de hardware/mecânica e documentação	N/A
23/05	Apresentação de hardware/mecânica	Entregável 3
24/05 a 30/05	Criação do dataset ou curadoria da base de imagens	N/A
31/05 a 06/06	Treinamento e validação da CNN	N/A
06/06 a 09/06	Implementação do sistema de texto-para-fala offline	N/A
10/06 a 12/06	Testes do software embarcado no Raspberry Pi	N/A
13/06	Apresentação do software	Entregável 4
14/06 a 20/06	Integração final: câmera + CNN + TTS + áudio	N/A
21/06 a 27/06	Testes de uso real, ajustes de latência e melhoria de usabilidade	N/A
28/06 a 01/07	Gravação do vídeo demonstrativo e finalização do Blog	N/A
01/07 a 03/07	Escrita e finalização do relatório técnico	N/A
04/07	Demonstração do protótipo, entrega do relatório, vídeo e Blog	Entregável 5
07/07 a 09/07	Preparação da apresentação para banca	N/A
10/07	Ensaio final cronometrado com feedback entre os membros	N/A
11/07	Apresentação final com banca avaliadora	Avaliação Final da Banca

O cronograma foi seguido rigorosamente, permitindo que todas as etapas do projeto fossem concluídas dentro do prazo estabelecido. A disciplina na execução das atividades e a gestão eficiente dos recursos foram cruciais para o sucesso e a pontualidade na entrega dos marcos.