

ARTIGO DE PESQUISA/RESEARCH PAPER

# Visão Computacional Flexível para Autonomia Assistiva: Uma Solução Edge AI Não-Invasiva de Baixo Custo

## Flexible Computer Vision for Assistive Autonomy: A Low-Cost, Non-Invasive Edge AI Solution

Luís Lourenço   [Universidade Federal de São Carlos | [luislourenco@estudante.ufscar.br](mailto:luislourenco@estudante.ufscar.br) ]

Luciano Neris  [Universidade Federal de São Carlos | [lneris@ufscar.br](mailto:lneris@ufscar.br) ]

 Departamento de Computação, Universidade Federal de São Carlos, Rod. Washington Luís, s/n, Monjolinho, São Carlos, SP, 13565-905, Brasil.

**Resumo.** No Brasil, milhões de pessoas enfrentam barreiras diárias para manter sua independência devido a limitações motoras. Embora a Visão Computacional e a Inteligência Artificial (IA) tenham avançado o campo da tecnologia assistiva, a maioria das soluções disponíveis no mercado é de alto custo, inacessível e carece de padronização para adaptação individualizada. Diante deste cenário, este trabalho propõe o desenvolvimento e a avaliação de um Dispositivo Assistivo Flexível (DAF) de baixo custo, utilizando Edge AI e Visão Computacional. O DAF visa promover a autonomia de indivíduos com severas limitações motoras, permitindo o controle de dispositivos externos através da detecção e interpretação de gestos ou expressões faciais. O grande diferencial metodológico reside na arquitetura flexível, que permite a customização individualizada por meio de *fine-tuning* remoto. O cuidador utiliza o DAF (baseado na placa Sipeed MAix Bit/Maixduino) para capturar imagens automaticamente rotuladas. Em seguida, essas imagens são enviadas a uma Plataforma Web para que o servidor realize o *fine-tuning* do modelo pré-treinado (MobileNet 2.5). Os novos pesos são então devolvidos por e-mail para serem atualizados no dispositivo. Em comparação com tecnologias existentes, o DAF demonstrou grandes diferenciais, sendo uma solução não-invasiva, de baixo custo e que não exige um host (computador auxiliar), facilitando sua modularização e transporte. O sistema apresentou um tempo de resposta satisfatório (1 a 2 segundos), princípio para uso prático. O trabalho atende, assim, à necessidade de dispositivos flexíveis, acessíveis e adaptáveis às necessidades específicas de cada usuário.

**Abstract.** In Brazil, millions of people face daily barriers to maintaining their independence due to motor limitations. Although Computer Vision and Artificial Intelligence (AI) have advanced the field of assistive technology, most solutions available on the market are high-cost, inaccessible, and lack standardization for individualized adaptation. Given this scenario, this work proposes the development and evaluation of a low-cost Flexible Assistive Device (DAF), utilizing Edge AI and Computer Vision. The DAF aims to promote the autonomy of individuals with severe motor limitations, allowing for the control of external devices through the detection and interpretation of gestures or facial expressions. The major methodological differential lies in the flexible architecture, which enables individualized customization through remote fine-tuning. The caregiver uses the DAF (based on the Sipeed MAix Bit/Maixduino board) to capture automatically labeled images. Subsequently, these images are sent to a Web Platform for a server to perform the fine-tuning of the pre-trained model (MobileNet 2.5). The new weights are then returned via email to be updated on the device. Compared to existing technologies, the DAF demonstrated significant advantages, being a non-invasive solution, low-cost, and one that does not require a host (auxiliary computer), facilitating its modularity and transport. The system presented a satisfactory response time (1 to 2 seconds), suitable for practical use. The work thus addresses the need for flexible, accessible, and adaptable devices for the specific needs of each user.

**Palavras-chave:** Tecnologia Assistiva, Visão Computacional, Edge AI, Fine-Tuning

**Keywords:** Assistive Technology, Computer Vision, Edge AI, Fine-Tuning

**Recebido/Received:** 16 October 2025 • **Aceito/Accepted:** 16 January 2026 • **Publicado/Published:** 23 January 2026

## 1 Introdução

Em 2022, cerca de 14,4 milhões de brasileiros possuíam algum tipo de deficiência. Esse número corresponde a 7,3% da população acima de dois anos de idade no Brasil. Apesar de não haver um número exato de pessoas com deficiências motoras paralisantes, percebe-se que 4% da população geral (7,9 milhões de pessoas) possuíam alguma dificuldade permanente para andar ou subir degraus, ou dificuldade permanente em pegar pequenos objetos ou abrir e fechar tampas de garrafas [Instituto Brasileiro de Geografia e Estatística, 2022]. Embora esses números englobem uma gama de dificuldades, as

pessoas que se encaixam nesse espectro, principalmente nos graus mais severos, enfrentam barreiras diárias para manter sua independência.

O desenvolvimento de tecnologias assistivas tem desempenhado um papel crucial na melhoria da qualidade de vida de pessoas com deficiências motoras e cognitivas. Com o avanço da Inteligência Artificial (IA) e da Visão Computacional, novas soluções vêm sendo exploradas para permitir maior autonomia e comunicação para esses indivíduos. Tecnologias como rastreamento ocular, sensores mioelétricos e dispositivos de interação assistida têm sido aplicadas para facilitar

atividades diárias e promover a inclusão social de pessoas com limitações motoras severas.

No entanto, apesar dos avanços tecnológicos, muitos dos dispositivos assistivos disponíveis no mercado são de alto custo e apresentam barreiras de acessibilidade para grande parte da população que deles necessita [Ariza and Pearce, 2022]. Agravando este cenário, a falta de padronização e a necessidade de adaptação individualizada e customizada dificultam a ampla adoção dessas tecnologias, além de encarregar ainda mais o processo. Isso evidencia a necessidade de dispositivos flexíveis, acessíveis e adaptáveis às necessidades específicas de cada usuário, promovendo maior independência e integração social [Naves *et al.*, 2012].

Diante desse cenário, este trabalho propõe o desenvolvimento e a avaliação de um dispositivo assistivo flexível baseado em *Edge AI* e Visão Computacional, voltado para promover a autonomia de indivíduos com severas limitações motoras, permitindo sua interação com sistemas computacionais e ambientes físicos de forma eficiente, intuitiva e acessível.

Os objetivos específicos incluem:

- Desenvolver uma solução de baixo custo utilizando hardware e software de código aberto, buscando acessibilidade e reprodutibilidade;
- Implementar algoritmos de Visão Computacional e IA embarcada para a detecção e interpretação de gestos ou expressões faciais, a fim de controlar dispositivos através desses gestos;
- Criar uma plataforma web simples e intuitiva para receber imagens a fim de realizar *fine-tuning* no modelo e retornar os pesos atualizados;
- Avaliar a eficácia da solução por meio de testes experimentais controlados, considerando métricas como tempo de resposta, precisão na detecção e facilidade de uso;
- Comparar o desempenho e a aplicabilidade da solução proposta com tecnologias assistivas existentes descritas na literatura, com foco em flexibilidade, custo e adaptabilidade a diferentes perfis de usuário.

## 2 Trabalhos Relacionados

É notável que, nos últimos anos, o interesse pelo desenvolvimento de tecnologias assistivas tem aumentado, produzindo diversas abordagens, métodos e resultados diferentes. Alguns dos trabalhos na literatura focam no movimento ocular. O trabalho de [Wang *et al.*, 2018] apresenta um sistema robótico que utiliza rastreamento ocular tridimensional para permitir que pacientes com quadriplegia consigam interagir com objetos ao seu redor, como pode ser visto em Fig. 1. O projeto combina reconhecimento de objetos e planejamento de trajetória para possibilitar uma interação intuitiva. Além disso, a solução obteve uma ótima acurácia nos testes iniciais. Contudo, o sistema contém um alto custo de produção, utilizando itens como óculos de rastreamento ocular, Microsoft Kinect e braço robótico.

Outro trabalho que utiliza o movimento ocular como foco de estudo é o de [Pandey *et al.*, 2018]. O sistema apresentado se utiliza tanto de movimentos da íris quanto de movimentos da pálpebra para produzir um método de comunicação para pacientes com paralisia. O projeto também apresentou uma

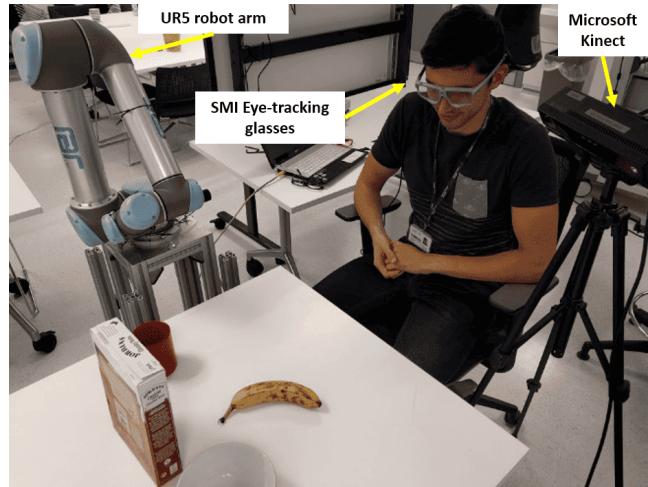


Figura 1. Sistema proposto por [Wang *et al.*, 2018].



Figura 2. Usuário utilizando os sensores mioelétricos de [Naves *et al.*, 2012]

boa acurácia e, ao contrário do trabalho anterior, utiliza itens de baixo custo, como uma webcam, sem necessidade de dispositivos especializados. Porém, essa solução necessita de um *host*, isto é, um computador, que deverá executar o código conectado a uma webcam, dificultando sua modularização.

No campo da visão computacional aplicada à mobilidade, [Rabhi *et al.*, 2018] desenvolveram uma Interface Homem-Máquina (HMI) para controlar uma cadeira de rodas elétrica através da detecção de expressões faciais. O sistema utiliza a câmera de um *smartphone* acoplada à cadeira e processa as imagens utilizando redes neurais e a biblioteca OpenCV em uma *Raspberry Pi*, alcançando uma acurácia de 98,6% nos testes realizados. Embora a solução demonstre alta precisão na interpretação de comandos faciais, os autores ressaltaram a necessidade de incluir uma câmera traseira para detecção de obstáculos, visando garantir a segurança do usuário durante a navegação.

Alguns outros artigos utilizam sensores vestíveis, a fim de detectar movimentos ou sinais mioelétricos. Na solução proposta por [Naves *et al.*, 2012], utilizam-se sensores que captam movimentos musculares no rosto, a fim de transcrever esses sinais em palavras e serem usados na comunicação. O trabalho necessita de um *host* para que o usuário consiga digitar usando o dispositivo, como pode ser visto em Fig. 2. O sistema teve um bom desempenho no escopo proposto, contudo, ele se utiliza de uma solução sutilmente invasiva, por conta da necessidade do contato dos detectores de sinais com o rosto do usuário.

Mais um artigo que segue uma linha semelhante é o de

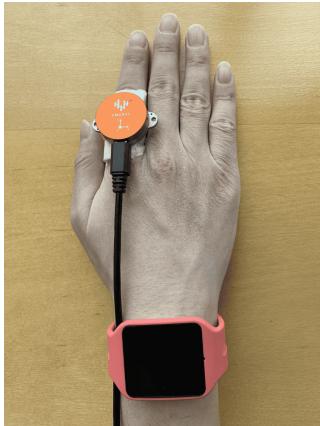


Figura 3. Dispositivos utilizados no trabalho de [Liu et al., 2020]

[Liu et al., 2020]. O objetivo do trabalho é, utilizando sensores vestíveis como anel inteligente e smartwatch (Fig. 3), rastrear gestos dos dedos e traduzir sinais da Língua de Sinais Americana. Para isso, o sistema utiliza principalmente os acelerômetros dos dispositivos para que, através de modelos probabilísticos, seja possível inferir gestos a partir dos dados coletados. De forma geral, o projeto proposto obteve um bom desempenho, conseguindo detectar bem as palavras expressas. Entretanto, assim como alguns outros trabalhos, esse também sofre pelo alto custo de produção e, como o trabalho apresentado no parágrafo acima, a utilização de sensores no corpo pode ser desconfortável e invasiva.

Focando na acessibilidade financeira e na filosofia de hardware livre, [González-Cely et al., 2022] propuseram um protótipo de cadeira de rodas controlada pelos movimentos da cabeça, com um custo total de produção em torno de US\$267. O sistema interpreta a posição, velocidade e orientação da cabeça do usuário para gerar comandos de movimentação, utilizando algoritmos desenvolvidos em Python e MATLAB, e disponibiliza todos os arquivos de projeto em repositórios abertos para replicação. Testado com 10 usuários e apresentando um tempo de reação de 100 ms, o trabalho destaca-se por reduzir significativamente as barreiras de custo, embora dependa de uma calibração precisa para diferentes perfis de uso.

Através desse levantamento, produziu-se a Tabela 1, a fim de comparar diretamente a tecnologia utilizada nos artigos, assim como o usuário a ser assistido e a aplicação do sistema desenvolvido.

### 3 Metodologia

#### 3.1 Descrição da Arquitetura

Os diagramas ilustrados pelas Fig. 4 e Fig. 5 apresentam a arquitetura e os casos de uso propostos para atingir os objetivos definidos neste trabalho, buscando definir o hardware necessário e também os modos de interação disponíveis aos usuários e aos cuidadores.

No diagrama de arquitetura do sistema, é possível notar o seguinte fluxo:

1. As imagens são capturadas pelo cuidador do usuário utilizando a câmera do dispositivo assistivo flexível, e já são automaticamente rotuladas enquanto são capturadas. No trabalho atual, são aceitas imagens de boca, mãos e olhos, com a pretensão de expansão para outras regiões

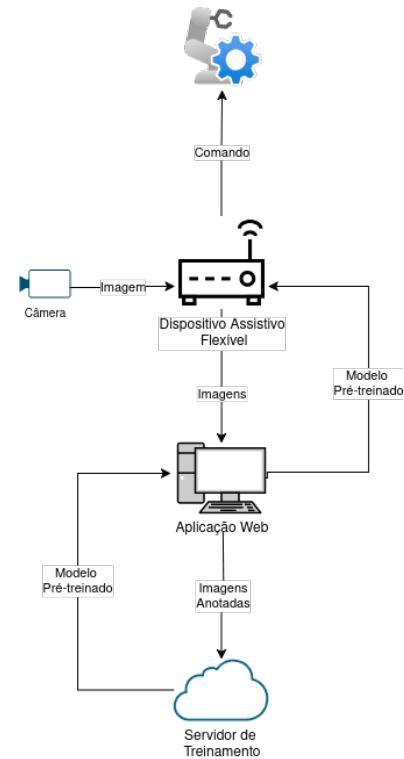


Figura 4. Diagrama de Arquitetura do Sistema

- em trabalhos futuros;
2. Essas imagens são enviadas para uma aplicação web (removendo o cartão SD do dispositivo, conectando ao computador e realizando o *upload* dessas imagens no site);
  3. Depois de enviadas, as imagens irão para um servidor de treinamento, para se realizar *fine-tuning* no modelo já treinado utilizando essas novas imagens;
  4. Após a conclusão desse processo, os novos pesos do modelo são devolvidos ao cuidador por e-mail;
  5. O cuidador baixa esses novos pesos e coloca-os no cartão SD do dispositivo;
  6. O dispositivo está pronto para ser usado. Neste estágio, atua o módulo de comando — a interface de hardware e software responsável por traduzir a inferência do modelo em uma ação física. Ao detectar o gesto ou região alvo, este módulo envia sinais elétricos (via GPIO, relés ou comunicação sem fio) para efetivar a ativação ou desativação do dispositivo externo controlado.

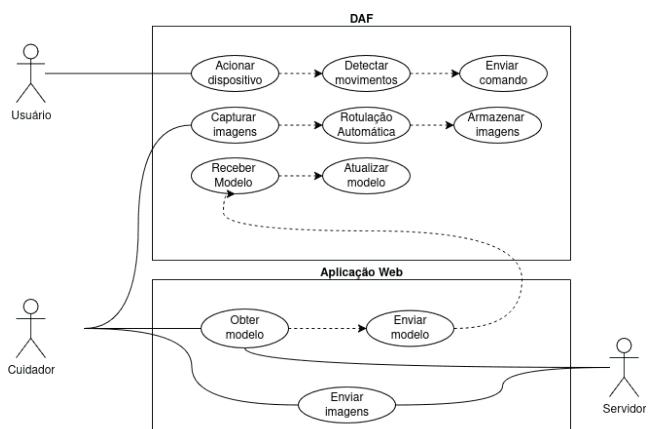


Figura 5. Diagrama de Casos de Uso

Título	Tecnologia Usada	Usuário Assistido	Aplicação
Free-View, 3D Gaze-Guided, Assistive Robotic System for Activities of Daily Living [Wang <i>et al.</i> , 2018]	Rastreamento ocular 3D, reconhecimento de objetos, câmera RGB-D	Pessoas com quadriplegia (ex.: ELA, paralisia cerebral)	Controle de braço robótico para atividades diárias
Alternative Communication System for People with Severe Motor Disabilities Using Myoelectric Signal Control [Naves <i>et al.</i> , 2012]	Eletromiografia (EMG), interfaces homem-computador	Pessoas com deficiência motora severa (ex.: ELA, tetraplegia)	Comunicação assistiva
A facial expression controlled wheelchair for people with disabilities [Rabhi <i>et al.</i> , 2018]	Câmera de Smartphone, Raspberry Pi, OpenCV, Redes Neurais	Pessoas com deficiências motoras severas (que necessitam de cadeira de rodas)	Controle de navegação de cadeira de rodas através do reconhecimento de expressões faciais/emoções
Assistance for Paralyzed Patient Using Eye Motion Detection [Pandey <i>et al.</i> , 2018]	Visão computacional, OpenCV, detecção de movimento ocular	Pacientes com paralisia severa (ex.: ELA)	Comunicação assistiva baseada em rastreamento ocular
Application Informed Motion Signal Processing for Finger Motion Tracking Using Wearable Sensors [Liu <i>et al.</i> , 2020]	Sensores vestíveis (IMU), modelo probabilístico (Bayesian Inference)	Pessoas surdas ou com deficiência na comunicação verbal	Tradução de gestos em linguagem de sinais para comunicação digital
Wheelchair prototype controlled by position, speed and orientation using head movement [González-Cely <i>et al.</i> , 2022]	Hardware Open-Source (custo reduzido), Python, MATLAB, Sensores de movimento	Pessoas com limitações motoras que preservam o controle da cabeça	Controle de cadeira de rodas baseado na orientação, velocidade e posição da cabeça do usuário

Tabela 1. Comparação entre os artigos analisados

Já nos casos de uso, percebe-se que existem três agentes (Usuário, Cuidador e Servidor) e dois ambientes (DAF e Aplicação Web), que são detalhados da seguinte forma:

- DAF (Dispositivo Assistivo Flexível): ambiente responsável pela parte prática de uso do dispositivo, isto é, toda a parte de captura de imagens, detecção de movimentos, acionamento de dispositivos, etc;
  - Usuário: usuário final que será beneficiado pela solução, aciona o dispositivo através dos seus movimentos e envia um comando para ativar/desativar outro dispositivo.
  - Cuidador: intermediador entre o usuário final e as questões técnicas do projeto, é responsável por, nesse ambiente DAF, capturar as imagens do usuário e atualizar os pesos do modelo.
- Aplicação Web: ambiente onde as imagens são enviadas para realização de *fine-tuning* e atualização dos pesos do modelo;
  - Cuidador: nesse ambiente, é responsável por enviar as imagens salvas e obter os pesos atualizados do modelo.
  - Servidor: encarregado de receber as imagens, realizar *fine-tuning* e retornar os pesos do modelo atualizados.

### 3.2 Comparação Entre Plataformas

É possível perceber que, de acordo com os artigos analisados nesse trabalho, existem divergências de plataformas entre as tecnologias existentes, ou seja, cada sistema utiliza uma

plataforma diferente. Alguns trabalhos utilizam dispositivos móveis vestíveis, outros utilizam *hosts* com auxílio de *webcams* ou sensores, dentre outras tecnologias. Nesse trabalho, o foco é a modularização da solução, para que o sistema possa ser instalado em qualquer lugar, além da redução de custos, não-invasão e simplicidade de utilização, como sugere [Ariza and Pearce, 2022]. Por esse motivo, optou-se pela utilização de algum dispositivo de borda — equipamento capaz de realizar o processamento computacional localmente, próximo à fonte de captura, sem depender de servidores externos — para que a solução tenha um baixo custo, seja modular, flexível e de simples utilização.

A Tabela 2 apresenta um comparativo técnico entre as plataformas Wio Lite AI, Sipeed MAix Bit/Maixduino, Dev Board Mini (Coral) e Intel Neural Compute Stick 2, considerando critérios essenciais para aplicações em tecnologia assistiva com visão computacional embarcada. A plataforma Wio Lite AI, baseada no microcontrolador STM32H725, destaca-se por sua boa conectividade (Wi-Fi e Bluetooth integrados), suporte a IDEs como STM32CubeIDE e disponibilidade de memória externa. No entanto, a ausência de um acelerador dedicado para IA limita seu desempenho em tarefas de inferência de redes neurais convolucionais.

Por outro lado, a Sipeed MAix Bit/Maixduino possui um co-processador KPU (Kendryte Processing Unit) para execução eficiente de CNNs, como Tiny-YOLO e Mobile-Net, diretamente no dispositivo. Essa característica tem sido explorada com sucesso em aplicações assistivas, como demonstrado por [Torres-Sánchez *et al.*, 2020], que utilizaram a MAix Go (com o mesmo chip Kendryte K210) para detecção



Figura 6. Placa utilizada na solução

de uso correto de máscaras em tempo real com desempenho médio de 13 FPS, evidenciando a viabilidade dessa plataforma para sistemas de visão embarcada com baixa latência e sem dependência da nuvem.

Expandindo a análise, a plataforma Dev Board Mini (Google Coral) incorpora o acelerador Edge TPU, com capacidade de até 4 TOPS, otimizando a execução de modelos TensorFlow Lite quantizados. Estudos como o de [Rosero-Montalvo *et al.*, 2024] evidenciam que, ao utilizar EfficientNet-B0 com quantização, o Coral foi capaz de manter a acurácia do modelo original com uma redução de até 4x no tempo de inferência e consumo energético, reforçando seu potencial para aplicações assistivas com restrições energéticas. De maneira similar, a Intel Neural Compute Stick 2, equipada com o VPU Myriad X, mostrou-se a plataforma mais versátil na comparação feita por [Rafal *et al.*, 2023], sendo capaz de executar uma ampla gama de modelos, incluindo MobileNet, EfficientNet e VGG16, mesmo com entradas de alta resolução. Apesar de demandar conexão com um host, seu desempenho robusto e compatibilidade com múltiplos frameworks (via OpenVINO) tornam-no um forte candidato para cenários em que há um computador auxiliar.

No entanto, ao considerar a arquitetura proposta para este projeto, onde o treinamento ocorre em um servidor externo e os modelos atualizados são posteriormente transferidos para execução local no dispositivo, a plataforma mais adequada deve conciliar autonomia computacional, suporte a IA embarcada e baixo custo. Neste contexto, a Sipeed MAix Bit/Maixduino (destacada em cinza na Tabela 2) oferece a melhor relação entre custo-benefício e desempenho em tempo real para tarefas de visão computacional assistiva. Sua arquitetura otimizada para inferência local, suporte a bibliotecas leves (como MaixPy) e integração com câmeras compatíveis a tornam ideal para o dispositivo de coleta embarcado que compõe o sistema proposto.

### 3.3 Captura de Imagens e Treinamento do Modelo

Durante essa etapa, foram capturadas imagens de três partes do corpo do autor (boca, mão e olho) e cada uma dessas partes teve duas posições diferentes de captura: mão aberta, mão fechada, boca para a esquerda, boca para a direita, olho para cima e olho em qualquer outra posição. A escolha dessas posições do corpo para captura ocorreu em virtude de essas regiões geralmente serem as últimas regiões em que o movimento é paralisado, em função de doenças degenerativas paralisantes, de acordo com [Domellöf, 2019].

As imagens foram capturadas em ambientes diferentes, com condições de iluminação distintas, a fim de gerar uma grande variabilidade nos dados, para que o modelo seja ca-



Figura 7. Exemplos de imagens capturadas de cada classe

paz de generalizar melhor suas classificações. Além disso, para facilitar o processo de captura de imagens, tanto no desenvolvimento do trabalho quanto na utilização por usuários, foi desenvolvido um sistema de rotulação automática, isto é, quando o processo de captura de imagem é inicializado, é apresentado um quadrado no centro do LCD mostrando que a região de interesse deve estar dentro desse delimitador. Em cada imagem capturada a legenda se altera de "Ativar" para "Desativar" para que o cuidador e o usuário saibam qual posição que estão capturando será responsável por ativar outro dispositivo e qual será responsável por desativá-lo. Na Fig. 7, são mostradas variações de imagens capturadas.

Ao final do processo de captura de imagens, obteve-se um total de 1451 imagens, sendo parte delas capturadas e parte delas geradas utilizando *data augmentation*, através da biblioteca Augmentor em Python. Após isso, as imagens foram separadas em treino e validação, numa proporção de 80-20.

Para o treinamento, utilizou-se um *framework* focado em modelos para IA de borda, chamado aXeRate, para facilitar o processo de treino e conversão dos pesos. Dentre os modelos disponíveis no *framework* que poderiam ser utilizados na placa (MobileNet 2.5, MobileNet 5.0 e SqueezeNet), o modelo que apresentou uma melhor precisão média (mAP) foi o MobileNet 2.5 e, por conta disso, foi o modelo escolhido para a versão final do sistema.

A supervisão do método ocorre durante a etapa de coleta de dados guiada. O sistema exibe na tela qual classe será capturada (ex: 'Mão Aberta'), e assume-se que todas as imagens coletadas nesse intervalo pertencem a essa classe (rotulação automática via inferência do estado da UI). A validação do acerto ou erro durante o uso real (inferência) é feita empiricamente pela observação da execução correta do comando pelo dispositivo externo.

### 3.4 Geração do Site e Servidor

Além do modelo de aprendizado de máquina, também foi gerado um site (Fig. 8), para que o usuário final possa enviar imagens, a fim de realizar *fine-tuning* no modelo treinado utilizando as novas imagens capturadas pelo usuário. O site conta com a funcionalidade de receber as imagens e o e-mail

Critério	Wio Lite AI	Sipeed MAix Bit / Maixduino	Dev Board Mini (Coral)	Intel Neural Compute Stick 2
<b>Arquitetura</b>	STM32H725 (Cortex-M7, 550 MHz)	Kendryte K210 (RISC-V dual-core, 400–600 MHz)	Edge TPU (Google) + Cortex-M4F (Microcontroller)	Myriad X VPU (Intel) via USB
<b>Acelerador de IA</b>	Não possui (execução via CPU)	Possui KPU (Kendryte Processing Unit) dedicado para CNNs	TPU dedicado (até 4 TOPS)	VPU dedicado (Myriad X, até 1 TOPS)
<b>Supporte a câmera</b>	Interface DCMI para câmeras OV2640, OV5640	Interface DVP, compatível com OV2640, OV7740, etc.	Suporte a câmera via MIPI CSI (ex: Raspberry Pi Camera)	Necessita câmera USB ou entrada via host
<b>Supporte a display</b>	Conector FPC para LCD RGB565 / RGB888	Display SPI TFT de 2.4" incluído (320x240 px)	Suporte via GPIO/MIPI (necessário adicionar display externo)	Não aplicável (acoplado a host com display)
<b>Memória RAM</b>	8 MB de PSRAM externa	8 MB de SRAM interna	8 MB SRAM + memória externa via eMMC ou SD	Depende do host (RAM interna mínima)
<b>Armazenamento Flash</b>	16 MB SPI NOR Flash	Armazenamento externo via cartão microSD	eMMC 8 GB embutido (modelo base)	Depende do host (sem armazenamento local)
<b>Conectividade</b>	Wi-Fi dual band + Bluetooth 5.1 integrados	Wi-Fi e Bluetooth via módulo ESP32 separado	Wi-Fi dual band + Bluetooth integrados (via módulo)	Sem conectividade nativa (depende do host)
<b>Facilidade de desenvolvimento</b>	STM32CubeIDE, TensorFlow Lite Micro	MaixPy (MicroPython), Maix-Tool, Arduino IDE	Python (PyCoral), C++ com suporte a TensorFlow Lite	Python + OpenVINO, suporte a TensorFlow/Caffe/ONNX
<b>Supporte a IA embarcada</b>	Limitado, sem unidade dedicada	Otimizado para modelos leves como Tiny-YOLO e MobileNet	Alto desempenho com modelos TFLite quantizados	Alta compatibilidade com modelos convertidos para OpenVINO
<b>Custo estimado</b>	Baixo (US\$30)	Baixo (US\$30)	Alto (US\$100)	Alto (US\$150 + host)

**Tabela 2.** Comparativo entre as plataformas Wio Lite AI, Sipeed MAix Bit/Maixduino, Dev Board Mini e Intel Neural Compute Stick 2 para aplicações em tecnologia assistiva com visão computacional



Figura 8. Página inicial do site

Classe	Maior precisão no treinamento
Aberta	0.79
Fechada	0.90
Cima	0.59
Outro	0.86
Direita	0.96
Esquerda	0.78

Tabela 3. Melhores valores de precisão para cada classe

da pessoa, armazenar essas informações e, quando o *fine-tuning* for concluído, o usuário receberá um e-mail contendo os pesos atualizados do modelo. O site serve apenas como representação da escalabilidade dessa solução e, por conta disso, está sendo hospedado localmente.

## 4 Discussões e Resultados

O desempenho do modelo foi heterogêneo: classes como "Direita" alcançaram precisão de até 96%, enquanto classes como "Cima" apresentaram resultados inferiores, limitando-se a 59%, conforme detalhado na Tabela 3.

Nas classes mais distinguíveis entre si, como as de mão (Aberta e Fechada) e boca (Direita e Esquerda), o modelo obteve ótimos resultados, identificando adequadamente as particularidades de cada gesto. Por outro lado, o desempenho nas classes de olho foi prejudicado pela sutileza das diferenças visuais. A baixa performance na classe "Olho Cima" (0,59) é atribuída principalmente à resolução espacial da entrada (224x224) e às limitações do sensor. A distinção entre "Olho Cima" e "Outro" depende da visualização de poucos pixels da esclera inferior; o contraste reduzido da câmera OV2640 tende a degradar as bordas da íris nessas condições, levando a arquitetura MobileNet a confundir a rotação vertical do olho.

Ressalta-se que este estudo avaliou a precisão frame-a-frame do classificador. Trabalhos futuros devem avaliar a "Taxa de Sucesso da Tarefa", medindo quantas tentativas o usuário necessita para efetivar um comando físico, métrica que captura melhor a experiência de uso real do que a acurácia isolada do modelo.

Além da precisão, a qualidade da solução foi avaliada considerando o tempo de resposta e a facilidade de uso. O

tempo médio obtido, de 1 a 2 segundos, mostrou-se satisfatório e coerente com a literatura de Tecnologia Assistiva. Segundo [Miller, 1968], respostas de até 2 segundos são aceitáveis para manter o fluxo de conversação homem-máquina. Adicionalmente, em interfaces controladas pelo olhar, tempos de fixação (*dwell time*) próximos a 1000ms são frequentemente deliberados para mitigar o "Problema do Toque de Midas" — a ativação acidental de dispositivos [Majaranta and Räihä, 2002] —, priorizando a segurança da ação sobre a velocidade instantânea. Por fim, o dispositivo demonstrou boa facilidade de uso, apresentando processos simples de instalação, captura de imagens, atualização de pesos e realização das detecções.

Comparando essa solução com outras citadas no trabalho, é possível perceber que seus grandes diferenciais são: uma tecnologia não-invasiva, isto é, nenhuma parte do dispositivo necessita ficar em contato com o usuário; possui um baixo custo de produção, podendo ser replicado por qualquer pessoa com uma placa do mesmo modelo; não precisa de um dispositivo *host*, facilitando sua modularização e transporte; atende diferentes tipos de usuários, sendo um dispositivo flexível que pode detectar diferentes partes do corpo (além de boca, mão e olho), bastando apenas capturar novas imagens e realizar o treinamento. O único ponto negativo da solução notado é a baixa precisão da classe de olho, que pode comprometer seu bom funcionamento, necessitando a captura de mais imagens e novas configurações de treino para melhorar o desempenho dessa classe.

## 5 Conclusão

De forma geral, o trabalho proposto cumpre seus objetivos iniciais, de gerar uma solução de baixo custo, acessível e reproduzível; utilizar Visão Computacional e IA embarcada para a detecção e interpretação de gestos ou expressões faciais; criar uma plataforma web para realização de *fine-tuning* no modelo; avaliar o desempenho da solução de acordo com métricas pré-definidas; e comparar com outras tecnologias assistivas existentes na literatura.

Para trabalhos futuros, deve-se focar no aumento da quantidade de imagens utilizadas, assim como em sua variabilidade, para remover vieses e aumentar a precisão da detecção de classes semelhantes. Também é recomendado hospedar o site em algum provedor, para que a solução possa efetivamente ser utilizada por diversos usuários, assim como automatizar o processo de recebimento de imagens, *fine-tuning* do modelo e devolução dos pesos atualizados, diminuindo o trabalho humano e melhorando o tempo de resposta para o usuário.

## Declarações complementares

### Agradecimentos

Gostaria de agradecer minha participação no Programa Institucional de Iniciação Científica e Tecnológica da Universidade Federal de São Carlos.

### Contribuições dos autores

LL e LN contribuíram para a concepção deste estudo. LL realizou os experimentos. LL é o principal contribuidor e escritor deste manuscrito. Todos os autores leram e aprovaram o manuscrito final.

## Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

## Disponibilidade de dados e materiais

Os conjuntos de dados (e/ou softwares) gerados e/ou analisados durante o estudo atual serão feitos mediante solicitação.

## Referências

- Ariza, J. and Pearce, J. M. (2022). Low-cost assistive technologies for disabled people using open-source hardware and software: A systematic literature review. *IEEE Access*, 10:124894–124927. DOI: 10.1109/ACCESS.2022.3221449.
- Domellöf, F. P. (2019). The extraocular muscles are selectively spared in als. In Hegde, M. L., editor, *Amyotrophic Lateral Sclerosis - Recent Advances and Therapeutic Challenges*, chapter 6. IntechOpen, London. DOI: 10.5772/intechopen.89504.
- González-Cely, A. X., Callejas-Cuervo, M., and Bastos-Filho, T. (2022). Wheelchair prototype controlled by position, speed and orientation using head movement. *HardwareX*, 11:e00306. DOI: <https://doi.org/10.1016/j.ohx.2022.e00306>.
- Instituto Brasileiro de Geografia e Estatística, . (2022). *Censo Demográfico 2022*. IBGE, Rio de Janeiro.
- Liu, Y., Jiang, F., and Gowda, M. (2020). Application informed motion signal processing for finger motion tracking using wearable sensors. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8334–8338. DOI: 10.1109/ICASSP40776.2020.9053466.
- Majaranta, P. and Räihä, K.-J. (2002). Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications, ETRA '02*, page 15–22, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/507072.507076.
- Miller, R. B. (1968). Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I, AFIPS '68 (Fall, part I)*, page 267–277, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1476589.1476628.
- Naves, E., Rocha, L., and Pino, P. (2012). Alternative communication system for people with severe motor disabilities using myoelectric signal control. In *2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, pages 1–4. DOI: 10.1109/BRC.2012.6222176.
- Pandey, M., Chaudhari, K., Kumar, R., Shinde, A., Totla, D., and Mali, N. (2018). Assistance for paralyzed patient using eye motion detection. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBE A)*, pages 1–5. DOI: 10.1109/ICCU-BEA.2018.8697455.
- Rabhi, Y., Mrabet, M., and Fnaiech, F. (2018). A facial expression controlled wheelchair for people with disabilities. *Computer Methods and Programs in Biomedicine*, 165:89–105. DOI: <https://doi.org/10.1016/j.cmpb.2018.08.013>.
- Rafal, T., Grzegorz, W., Piotr, G., Nikodem, C., and Sebas- tian, L. (2023). Edge devices inference performance comparison. *Journal of Computing Science and Engineering*, 17(2):51–59. DOI: 10.5626/jcse.2023.17.2.51.
- Rosero-Montalvo, P. D., Tözün, P., and Hernandez, W. (2024). Optimized cnn architectures benchmarking in hardware-constrained edge devices in iot environments. *IEEE Internet of Things Journal*, 11(11):20357–20366. DOI: 10.1109/JIOT.2024.3369607.
- Torres-Sánchez, E., Alastruey-Benedé, J., and Torres-Moreno, E. (2020). Developing an ai iot application with open software on a risc-v soc. In *2020 XXXV Conference on Design of Circuits and Integrated Systems (DCIS)*, pages 1–6. DOI: 10.1109/DCIS51330.2020.9268645.
- Wang, M.-Y., Kogkas, A. A., Darzi, A., and Mylonas, G. P. (2018). Free-view, 3d gaze-guided, assistive robotic system for activities of daily living.