Anais do Congresso Brasileiro de Iniciação Científica
(https://periodicos.unisanta.br/COB)

ISSN 2317-1316

DISPOSITIVO INTELIGENTE PARA PESSOAS COM BAIXA VISÃO

SMART DEVICE FOR PEOPLE WITH LOW VISION

Gabriele Cristina dos Santos Silva, Pedro Paulo Nunes da Silva, Luís Fernando Pompeo Ferrara, Raquel Galhardo de Carvalho Lopes Araújo

Universidade Santa Cecília, Faculdade de Engenharia, Curso de Engenharia da Computação
E-mail para contato: gs220060@alunos.unisanta.br

RESUMO - Este trabalho apresenta um dispositivo inteligente acoplado a óculos para auxiliar pessoas com baixa visão, integrando modelos de redes neurais e visão computacional. O objetivo foi criar um protótipo capaz de reconhecer textos e convertê-los para áudio, identificar gestos previamente definidos e realizar reconhecimento facial com processo de cadastro de usuário. Para isso, foram utilizadas bibliotecas open source e redes neurais como MobileNetV3small e MobileFaceNet. Os testes mostraram que o dispositivo operou de maneira satisfatória sob diferentes condições de iluminação. O protótipo físico foi construído com sucesso via impressão 3D. Conclui-se que o sistema atinge os objetivos propostos, mostrando-se uma ferramenta eficiente para a inclusão social.

Palavras-chave: Baixa visão; Visão computacional; Redes neurais.

ABSTRACT – This work presents a smart device coupled to glasses to assist people with low vision, integrating neural network models and computer vision. The goal was to create a prototype capable of recognizing text and converting it to audio, identifying previously defined gestures, and performing facial recognition, with a user registration process. To this end, open-source libraries and neural networks, such as MobileNetV3small and MobileFaceNe, were used. The tests showed that the device operated satisfactorily under different lighting conditions. The physical prototype was successfully built using 3D printing. It is concluded that the system achieves the proposed goals, proving to be an efficient tool for social inclusion.

Keywords: Low vision; Computer vision; Neural networks.

ISSN 2317-1316

1. INTRODUÇÃO

Conforme estabelece o Estatuto da Pessoa com Deficiência, uma pessoa com deficiência visual possui limitações de caráter físico, que somadas a fatores sociais, impedem o seu pleno envolvimento na sociedade e de estar em condições iguais às da população em geral. Entre os tipos de deficiências visuais, está a baixa visão, ou visão subnormal, caracterizada pela grande dificuldade de enxergar (OTTAIANO *et al.*, 2019).

A baixa visão pode ser classificada em leve, moderada, grave e profunda, de acordo com a acuidade e o campo visual da pessoa. Essa condição não pode ser amenizada ou reparada através de cirurgias refrativas ou utilizando lentes de contato ou óculos (IORJ, 2025).

O Instituto Brasileiro de Geografia e Estatística (IBGE), por meio do Censo Demográfico realizado no ano de 2010, concluiu que cerca de 18,6% da população brasileira possuía alguma forma de deficiência visual, o equivalente a cerca de 6,5 milhões de brasileiros. Desses 6,5 milhões, cerca de 6 milhões apresentavam grande dificuldade para enxergar, como a baixa visão e visão subnormal, mesmo utilizando lentes ou óculos, afetando, dessa maneira, a vida cotidiana dessa parcela da população (IBGE, 2010).

O reconhecimento de caracteres ópticos, OCR – Optical Character Recognition, é um processo que extrai caracteres de textos impressos, manuscritos, documentos e placas de veículos de imagens. Atualmente, existem diversas bibliotecas e modelos pré-treinados que realizam a função de reconhecimento; os mais conhecidos são o Tesseract, que possui suporte para diversos idiomas, o EasyOCR e também o Keras-OCR. O reconhecimento pode possuir diversas etapas, dependendo do modelo a ser usado, mas basicamente foca em pré-processar as imagens, eliminando possíveis ruídos que possam comprometer a identificação dos caracteres, extrair atributos, que consiste no reconhecimento do caractere, e por último classificar de acordo com uma técnica de classificação que pode ser bayesiana, rede neural, etc. (MIRANDA, 2013).

O reconhecimento de gestos permite a interação entre máquinas e seres humanos, sendo empregado em diversas áreas, como o reconhecimento de linguagem de sinais e interação com ambientes virtuais. Os gestos podem ser divididos em estáticos ou dinâmicos; os gestos estáticos podem ser identificados apenas com uma imagem, diferente dos dinâmicos, que

ISSN 2317-1316

dependem de uma sequência para ser interpretados. Os gestos estáticos das mãos são os mais usados na área de reconhecimento de gestos, visto a possibilidade de diferentes formas de representar uma ação ou pedido e também pela facilidade de identificação, já que a imagem é estática (PEREIRA, 2022). A técnica baseada em Visão Computacional utiliza uma câmera para capturar a imagem, que posteriormente é processada por algum modelo de IA, que mapeia e classifica o gesto, analisando diversos pontos da imagem; assim, após a classificação, a ação desejada é executada.

O processo de reconhecimento facial envolve algoritmos de *Machine Learning* e funciona de forma semelhante ao reconhecimento de gestos. Os modelos de IA geralmente são treinados com um grande banco de dados contendo diversas imagens de pessoas ao redor do mundo. Cada pessoa possui características únicas que formam o seu rosto; o *software* tem a função de mapear cada rosto e analisar as características de cada pessoa, como a distância entre os olhos, marcas e cicatrizes no rosto, o formato do rosto, entre outros. As informações são transformadas em vetores e armazenadas em um banco de dados para serem posteriormente comparadas. Quando é introduzida uma face para teste de reconhecimento facial, o algoritmo busca as informações no banco de dados e as compara com as características da face fornecida; se forem semelhantes, o rosto é reconhecido. Para treinar os modelos de reconhecimento facial, são utilizadas Redes Neurais Convolucionais - CNNs e arquiteturas reduzidas, como o MobileNet ou FaceNet.

Com o avanço da tecnologia e estudos sobre a IA, alguns dispositivos embarcados foram especialmente desenvolvidos para pessoas com deficiências visuais. Equipamentos como óculos inteligentes com sensores de proximidade e dispositivos capazes de reconhecer textos, pessoas, códigos de produtos, gestos e até moedas de diversos países.

Dessa forma, este trabalho visa desenvolver um protótipo de um dispositivo embarcado, baseado no treinamento de IA com imagens, capaz de auxiliar pessoas com baixa visão.

2. MATERIAIS E MÉTODOS

Diante do objetivo proposto de desenvolver um dispositivo inteligente para pessoas com baixa visão, foi necessário combinar elementos de *software* e *hardware*.
A arquitetura geral do dispositivo permitiu a integração entre os componentes eletrônicos e os

ISSN 2317-1316

algoritmos desenvolvidos para reconhecimento facial, identificação de gestos, reconhecimento de textos e funções para melhorar a experiência do usuário, como informar a carga da bateria e botões para captura de imagens e cadastramento de usuário. Os componentes eletrônicos e os métodos de treinamento de IA estão listados na tabela 1.

Tabela 1: Componentes Eletrônicos

Componente	Especificações / Características	Função Principal
Raspberry Pi 4 – 8 GB	8 GB de memória SDRAM, Wi-Fi, Bluetooth, 40 pinos GPIO, interface CSI-2 e compatibilidade com Python.	Processamento central das imagens, reconhecimento de gestos, reconhecimento facial e leitura de textos.
Câmera	5 MP, resolução de até 2592 × 1944 pixels e conexão USB	Captura de imagens em tempo real para reconhecimento facial, reconhecimento de gestos e leitura de textos.
Fonte de alimentação / UPS HAT	Alimentação de 5 V, mínimo de 2,5 A e comunicação via protocolo I²C	Fornecimento de energia ao sistema e monitoramento do nível de carga da bateria.
Fone de ouvido	Conector Jack 3,5 mm	Fornecimento de saída de áudio com maior conforto e privacidade ao usuário.
Alto-falante	Micro alto-falante	Alternativa de saída de áudio para utilização sem fones de ouvido.
Botões	Botões sem trava conectados aos pinos digitais GPIO	Seleção das funções do dispositivo, como verificação da bateria, controle do modo de áudio e ativação do reconhecimento facial, gestos e texto.
Microfone	Microfone integrado ao fone de ouvido	Captação de comandos de voz do usuário, como comandos para cadastro de rostos.

Fonte: Elaborado pelo autor (2026).

Para o reconhecimento de gestos, optou-se por usar o banco de dados Hand Gesture Recognition Image Dataset, pelo fato de possuir 33 diferentes tipos de gestos, representados por diversas pessoas em ambientes diferentes. Apesar desse número elevado de gestos, no protótipo apenas três gestos são utilizados para acessar funções do dispositivo.

Na figura 1a, observa-se a palma da mão aberta, este gesto ativa o reconhecimento de textos. A figura 1b o gesto do punho fechado para acionar o reconhecimento facial e a figura 1c, a representação de dois dedos estendidos em sinal de “L” para efetuar a leitura da carga da bateria.

ISSN 2317-1316

Figura 1: (a) Gesto palma da mão aberta; (b) Gesto punho fechado; (c) Gesto em sinal de "L" (KAPITANOV, 2025).



(a) (b) (c)

Primeiramente, para poder usar a rede neural MobileNetV3Small, foi necessário organizar e dividir a base de dados utilizada. Para isso, utilizaram-se métodos de bibliotecas em Python chamadas os, zipfile e shutil, que tinham como função separar dos conjuntos de dados somente as pastas dos gestos escolhidos e dividi-los em duas pastas: treino e validação, com a porcentagem de 80% e 20%, respectivamente. Para que essas imagens pudessem ser utilizadas, também foi feito um pré-processamento através da biblioteca OpenCV e MediaPipe, para recortar apenas a região que indicasse o gesto, eliminando possíveis ruídos no momento do treino e validação.

Para processar todas as imagens no total de 2000, cada classe, foi necessário implementar um código que pudesse carregar e processar as imagens em lotes para não consumir toda a memória RAM do ambiente, o número escolhido foi 50.

Após toda a parte de pré-processamento, para treinar o modelo, o treinamento foi realizado utilizando a técnica de transferência de aprendizagem. Dessa forma, a base convolucional da rede neural MobileNetV3Small foi congelada para ser utilizada como extratora de características, aproveitando, dessa forma, todo o conhecimento que a rede possuía quando foi treinada na sua criação.

Desse modo, apenas a camada final foi customizada e adaptada de acordo com a saída esperada. Nesse caso, a distinção entre três classes: *fist*, *stop* e *thumb_index*, ou seja, três neurônios na camada final com ativação da função *softmax*, retornando uma probabilidade para cada classe.

ISSN 2317-1316

Após o treinamento do modelo, ele foi convertido para o formato Tensor Flow Lite, visando uma execução mais otimizada no momento do reconhecimento do gesto.

O fluxo de funcionamento do dispositivo inicia-se com a captura automática da imagem, disparada após a escolha da funcionalidade por meio de um botão. A imagem passa por um pré-processamento de recorte e identificação das *landmarks* da mão do usuário. Após a identificação, a imagem é recortada e redimensionada para o tamanho de 128 x 128 pixels, servindo como entrada para o modelo treinado.

A função *softmax*, presente na camada final, tem o papel de retornar um vetor de probabilidades para as classes treinadas (*fist*, *stop* e *thumb_index*). Para evitar a ocorrência de falsos positivos, utilizou-se um limiar de confiança de 70%; dessa forma, o gesto aciona a função correspondente apenas para resultados acima desse índice, evitando erros que interfiram no funcionamento do sistema.

O reconhecimento facial foi desenvolvido com o auxílio da biblioteca NumPy, utilizada para manipular e transformar as imagens em vetores processáveis pela rede MobileFaceNet. A biblioteca OpenCV foi empregada no pré-processamento da imagem, sendo responsável pela captura da imagem. A detecção da face é feita pela MediaPipe Face Detection, um *framework* de código aberto pertencente a Google, dentro do fluxo existe um *fallback* para o Haar Cascade do OpenCV, caso a detecção realizada pelo MediaPipe apresente falhas durante o processo de detecção da face. Após a captura e detecção do rosto na imagem, o mesmo é recortado e convertido para o espaço de cores RGB, sendo redimensionado para o padrão de 112 x 112 pixels, para ser usado como entrada na rede neural escolhida.

A extração de características é realizada pela rede neural MobileFaceNet, convertida para o formato TensorFlow Lite para garantir uma execução otimizada no *hardware* embarcado de modo que não afete o desempenho e processamento do Raspberry Pi. A rede extrai as características faciais do usuário e retorna um vetor de *embeddings* de 192 dimensões. Estes dados, juntamente com o nome do usuário e um ID de identificação, são armazenados em um banco de dados SQlite.

A etapa de identificação utiliza a biblioteca *scikit-learn* com o método de Similaridade do Cosseno. Este método compara o vetor capturado na imagem com o *embedding* presente no banco de dados; caso o valor de similaridade atinja o valor base de 0,75, o rosto é reconhecido,

ISSN 2317-1316

retornando o nome do usuário armazenado no banco de dados relacionado àquele vetor. A interface de interação é mediada por voz, utilizando o Vosk STT para o reconhecimento de fala em português e o Piper TTS para a síntese de voz, ambos otimizados para dispositivos embarcados.

a. Leitura da Carga da Bateria

Com o objetivo de aumentar a autonomia e a usabilidade do dispositivo, foi implementado um sistema de monitoramento da carga da bateria. Ele é responsável por informar ao usuário o nível de energia disponível. Essa funcionalidade é especialmente importante para garantir que o equipamento permaneça operacional durante o uso e para alertar o usuário quando houver necessidade de recarregar o dispositivo.

O monitoramento é realizado por meio de um módulo UPS HAT conectado ao Raspberry Pi através do protocolo de comunicação I2C. A comunicação é feita utilizando a biblioteca Python SMBus2, que permite acessar os registradores internos do controlador da bateria.

Durante o processo de leitura, o *software* consulta dois registradores principais do módulo. O primeiro fornece a tensão atual da bateria, enquanto o segundo retorna o percentual estimado de carga.

Após a obtenção dos dados, o sistema gera uma mensagem informativa contendo a porcentagem de carga disponível e a tensão atual da bateria. Essas informações são convertidas em áudio através do sistema de síntese de voz Piper, permitindo que o usuário receba o retorno de forma auditiva, sem a necessidade de consultar uma interface visual.

Além da informação básica sobre a bateria, foram implementados mecanismos de alerta para situações de carga baixa. Por exemplo, quando o nível da bateria fica abaixo de 50%, o sistema informa ao usuário que é recomendável conectar o carregador. Caso a carga seja inferior a 20%, é emitido um alerta crítico solicitando que o carregador seja conectado imediatamente para evitar o desligamento inesperado do dispositivo.

O fluxo de funcionamento inicia-se quando o usuário solicita a leitura da bateria através do gesto previamente definido ou por um botão físico do dispositivo.

ISSN 2317-1316

b. Reconhecimento de Texto (OCR)

Para a construção de um processo de reconhecimento do texto (OCR – Optical Character Recognition) desenvolveu-se um fluxo com foco em execução eficiente em dispositivos embarcados. As etapas consideram as limitações de processamento e de arquitetura do projeto proposto. Para isso, optou-se por uma abordagem que consiste em pré-processamento de imagens utilizando visão computacional, particionamento de locais com texto e reconhecimento de caracteres com um modelo otimizado para sistemas embarcados.

c. Pré-processamento da Imagem

O pré-processamento da imagem foi implementado utilizando a biblioteca OpenCV, ela é responsável por melhorar a qualidade da imagem antes da leitura e reconhecimento do texto.

Primeiro, a imagem é capturada e convertida para uma escala de cinza, reduzindo a complexidade computacional necessária para reconhecer o texto. Em seguida, um filtro Gaussiano é aplicado para suavizar os ruídos causados pela captura (tanto baixa iluminação, como alta iluminação).

Após isso, utiliza-se o algoritmo CLAHE (Constrast Limited Adaptive Histogram Equalization), que melhora o contraste da imagem e permite uma maior visibilidade dos caracteres em relação ao fundo, isso ajuda o algoritmo final de reconhecimento, diferenciar caracteres reais de caracteres falsos (ruídos que podem passar no reconhecimento de caracteres).

A etapa de binarização é realizada utilizando uma margem que se adapta às circunstâncias locais de cada parte da imagem, o que permite uma melhor segmentação do texto mesmo em condições em que a iluminação possa ser irregular.

Por último, para lidar com imagens inclinadas, distorcidas ou rotacionadas, foi implementado um processo de correção de rotação que detecta os contornos e calcula o ângulo

ISSN 2317-1316

dominante do texto na imagem, assim mesmo que a imagem esteja fora do ângulo de reconhecimento, o algoritmo faz o tratamento e coloca na posição desejada.

i. Detecção de Texto

Para melhorar a precisão do OCR e reduzir o processamento desnecessário, foi implementada uma etapa de detecção de regiões contendo texto. Nesta etapa, considera-se a utilização de modelos leves, permitindo identificar áreas que sejam mais relevantes na imagem antes da leitura.

Dessa forma, reduz-se muito o poder computacional necessário para executar o reconhecimento, uma vez que o OCR será apenas aplicado nas regiões detectadas, e não na imagem completa.

ii. Reconhecimento de Caracteres

Para a etapa de reconhecimento, foi utilizada a biblioteca Tesseract OCR, devido à sua compatibilidade e otimização em dispositivos embarcados, além do suporte ao português. Foi considerado antes a criação de um modelo próprio para a detecção, mas devido às limitações do Raspberry essa possibilidade foi descartada por ser incompatível com a arquitetura e poder de processamento disponíveis.

O *Tesseract* foi configurado para operar no modo LSTM (Long Short-Term Memory), que apresenta um melhor desempenho na leitura de textos impressos, como rótulos, livros e embalagens.

Outros parâmetros também foram ajustados como: PSM (Page Segmentation Mode), para ajudar na leitura de blocos de texto em livros ou revistas; Whitelist de caracteres, para restringir o reconhecimento quando necessário.

iii. Conversão do Texto em Áudio

Após a extração do texto, ele é convertido em áudio utilizando o sistema de síntese de voz Piper, permitindo que o usuário receba as informações de forma auditiva. Essa etapa é essencial para garantir a acessibilidade do sistema para pessoas com baixa visão.

ISSN 2317-1316

d. Placa de Circuito Impresso

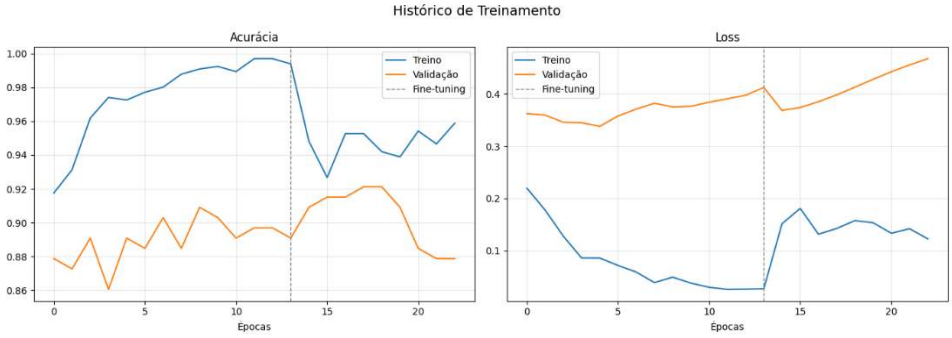
A placa de circuito impresso, que conecta todos os componentes físicos à Raspberry Pi, foi elaborada no *software* de código livre KiCad. Primeiramente, foi feito um desenho esquemático do circuito, selecionando e conectando cada componente ao seu respectivo GPIO ou trilha de alimentação. No projeto todo foram utilizados 6 botões com as respectivas funções: reconhecimento facial, reconhecimento de gestos, conversão de texto para áudio, botão para confirmar os dados recebidos pelo microfone, leitura da carga da bateria e alternância de áudio, onde o usuário pode selecionar a forma de saída de áudio entre o alto-falante e o fone de ouvido.

3. RESULTADOS E DISCUSSÃO

a. Reconhecimento de Gestos

O modelo treinado utilizando MobileNetV3Small atingiu cerca de 92,1% de acurácia após 15 épocas de treinamento. A figura 2 mostra a evolução do desempenho do modelo treinado, de acordo com os conjuntos de teste e validação. Após o início da etapa de *fine-tuning*, o modelo teve uma pequena queda na acurácia em relação aos dados de treinamento sendo depois estabilizado e ficando por volta de aproximadamente 96%. A acurácia de validação manteve cerca de 88% durante todo o treinamento, mantendo-se constante.

Figura 2: Histórico de treinamento.



Fonte: Elaborado pelo autor (2026).

ISSN 2317-1316

Com as imagens da pasta de teste, a acurácia ficou em torno de 92%. A matriz de confusão registrou 152 classificações corretas de um total de 165 amostras.

Foi realizado um teste com imagens capturadas por um *smartphone* e por uma *webcam*, em que o modelo conseguiu classificar corretamente as três imagens de teste para cada gesto, retornando à probabilidade de cada classe. Não houve diferença expressiva em relação aos resultados das classes, embora a *webcam* não possuísse grande resolução em comparação ao *smartphone*. Os testes feitos com a câmera da Raspberry Pi foram satisfatórios, o modelo conseguiu reconhecer os gestos e retornar a classe correta.

b. Reconhecimento Facial

Para o reconhecimento facial, também foi utilizada uma imagem capturada pelo smartphone. Após o processamento e recorte da imagem, o modelo retornou, em forma de vetor, as características faciais do rosto presente na imagem e, posteriormente, as armazenou no banco de dados, contendo número de identificação, nome e vetor facial. O modelo de reconhecimento facial demonstrou um bom desempenho para a tarefa, porém ainda deve ser otimizado para ser processado dentro da Raspberry Pi 4. Os testes realizados com o modelo MobileFaceNet e a câmera da Raspberry Pi atingiram o objetivo de reconhecer rostos cadastrados.

c. Reconhecimento de Texto

Para a identificação de texto está sendo utilizado um fluxograma de pré-processamento ligado ao reconhecimento de áreas de interesse e reconhecimento de caracteres que é mais eficiente para o processamento de imagens. A biblioteca de pré-processamento OpenCV é utilizada para deixar as imagens mais claras e corrigir quaisquer imperfeições que possam prejudicar o resultado.

O reconhecimento do texto apresentou resultados satisfatórios para os três tipos de cenários previstos (baixa luz, luz regular, e luz alta), além de para objetos específicos como rótulos, placas e livros.

ISSN 2317-1316

d. Protótipo

Para construir o protótipo, foi utilizado o *software* Autodesk Fusion. Ele permitiu a criação e a modelagem das peças virtualmente antes da fabricação. Isso facilitou o planejamento das dimensões, dos encaixes e demais características do projeto.

Após essa etapa, as peças foram produzidas por meio de impressão 3D utilizando o material PLA (Ácido Polilático). A escolha desse material ocorreu devido à sua facilidade de impressão, baixo custo, boa resistência para aplicações em protótipos. Além disso, é um material biodegradável, o que facilita um processo de desenvolvimento e testes sustentável.

Dessa forma, foi possível realizar ajustes durante o processo de desenvolvimento sem a necessidade de métodos de fabricação mais complexos ou com custos elevados.

4. CONCLUSÃO

Este trabalho atingiu seus objetivos ao desenvolver um protótipo inicial de dispositivo inteligente embarcado para auxílio a pessoas com baixa visão, utilizando técnicas de visão computacional e inteligência artificial. Em ambiente controlado e dentro da área para que foi treinado, nota-se que o desempenho é satisfatório, e há uma boa eficiência de processamento, mesmo com um *hardware* limitado. As funcionalidades de reconhecimento de gestos, identificação facial e leitura de texto com conversão para áudio foram otimizadas, e funcionam bem. No momento, há limitações no que diz respeito a variações de iluminação, ângulos (no reconhecimento de caracteres), e de ruído (transcrição de fala). Diante disso, trabalhos futuros devem focar no aumento da base de dados de treinamento, na otimização da precisão dentro dos sistemas de transcrição e fala, no tempo de resposta dos algoritmos, e no aprimoramento da integração entre *hardware*, *software* e *design*, para torná-lo mais acessível, intuitivo, e prático no cotidiano de seus usuários.

ISSN 2317-1316

5. DECLARAÇÃO DE USO DE IA

Durante a elaboração deste trabalho, utilizou-se a ferramenta Gemini (Flash 3.6 | Pro 3.1) para correção ortográfica. O conteúdo gerado foi revisado, editado e validado pelos autores, que assumem total responsabilidade pelo texto final.

6. REFERÊNCIAS

IBGE. Censo demográfico: Características da população e dos domicílios. 1. Rio de Janeiro: [s.n.], 2010. p. 1-270. Disponível em: https://educa.ibge.gov.br/jovens/conheca-o-brasil/populacao/20551-pessoas-com-deficiencia.html. Acesso em: 07 de março de 2025.

IORJ – INSTITUIÇÃO DE OFTALMOLOGIA DO RIO DE JANEIRO. Baixa visão. 2025. Disponível em: https://iorj.med.br/baixa-visao/. Acesso em: 03 de março de 2025.

MIRANDA, R. A. R. et al. Handwritten character recognition based on frequency, character-edge distances and densities. In: WORKSHOP DE VISÃO COMPUTACIONAL (WVC), 9., 2013, Rio de Janeiro. Anais... Rio de Janeiro: FGV, 2013.

KAPITANOV, Alexander. Hagrid – Hand Gesture Recognition Image Dataset. Disponível em: https://github.com/hukenovs/hagrid?tab=readme-ov-file. Acesso em: 26 de novembro de 2025.

OTTAIANO, José Augusto Alves; ÁVILA, Marcos Pereira de; UMBELINO, Cristiano Caixeta; TALEB, Alexandre Chater. Condições de saúde ocular no Brasil – 2019. São Paulo: Conselho Brasileiro de Oftalmologia, 2019. Disponível em: https://www.cbo.com.br/novo/publicacoes/condicoes_saude_ocular_brasil2019.pdf. Acesso em: 03 de março de 2025.

PEREIRA, Matheus Silva. Desenvolvimento de óculos inteligentes com tecnologia assistiva para pessoas com deficiência visual. 2022. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Universidade de Brasília, Brasília, 2022. Disponível em: https://bdm.unb.br/bitstream/10483/34512/1/2022_MatheusSilvaPereira_tcc.pdf. Acesso em: 10 jun. 2025.