SISTEMA DE AUDIODESCRIÇÃO: Tecnologia Assistiva para descrição de objetos para deficientes visuais

AUDIO DESCRIPTION SYSTEM: Assistive Technology for describing objects for the visually impaired

Marcela Colodiano Gonçalves[1], Rogério Thomazella[2]

RESUMO: A inclusão de pessoas que apresentam alguma deficiência visual ainda é um grande desafio, isso ocorre pela falta de tecnologias para garantir a sua inclusão em eventos culturais, políticos e sociais. Com isso a Tecnologia Assistiva trabalha cada dia para a sua evolução de oferecer a esses indivíduos uma melhor inclusão na sociedade, fazendo assim que os mesmos se tornem cada dia mais independentes, ou seja, que essas pessoas não necessitam de alguém ao seu lado a todo momento do dia para fazer sua rotina. Assim, o presente trabalho desenvolve um dispositivo capaz de fazer a audiodescrição de objetos utilizando a ESP32-CAM, processamento de imagem e Text-to-Speech, assim possibilitando que pessoas com alguma deficiência visual possam se locomover e saber o que está ao seu redor. O protótipo desenvolvido busca ser uma maneira intuitiva e de fácil manuseio, com isso ele se baseia em um óculos, sendo assim um objeto usual e na altura dos olhos do usuário, oferecendo a ele uma maior comodidade e independência para realizar sua rotina, além buscar inseri-lo na sociedade ativamente. Por fim, é apresentado um protótipo funcionante, onde este consegue fazer o reconhecimento dos objetos colocados em sua frente e enviar para o smartphone, que por sua vez irá reproduzir o nome do objeto detectado.

Palavras chaves: Deficiência visual, Tecnologia Assistiva, ESP32-CAM,Text-to-speech.

ABSTRACT: The inclusion of people with visual impairments remains a significant challenge due to a lack of technology to ensure their inclusion in cultural, political, and social events. Assistive technology is constantly evolving to offer these individuals better inclusion in society, enabling them to become more independent and less dependent on someone to manage their daily routines. This work develops a device capable of providing audio description of objects using the ESP32-CAM, image processing, and Text-to-Speech, thus allowing visually impaired individuals to move around and understand their surroundings. The prototype aims to be intuitive and easy to use, based on a pair of glasses, making it a convenient object at eye level for the user, offering greater comfort and independence in their daily lives and actively integrating them into society. Finally, a working prototype is presented, which is able to recognize objects placed in front of it and send the information to a smartphone, which in turn will display the name of the detected object.

Keywords: Visual impairment, Assistive Technology, ESP32-CAM,Text-to-speech.

---

[1] Graduanda em Sistemas Biomédicos. E-mail: Marcela.goncalves@fatec.sp.gov.br

[2] Docente da Faculdade de Tecnologia de Bauru. E-mail: Rogerio.thomazella@fatec.sp.gov.br

1. INTRODUÇÃO

Em um cenário caracterizado pela falta de acessibilidade é um desafio para a Tecnologia Assistiva (TA), essa que assume um papel estratégico e fundamental para a inclusão. A TA é a tecnologia onde visa implementar objetos com alguma funcionalidade que venha ser inclusiva ao usuário dependente dela. Temos como tecnologia assistiva bengalas, tradução em libras, cadeiras de rodas, andadores; esses equipamentos e tecnologias tem apenas um fim, a acessibilidade da parcela da população a qual não tem acesso com facilidade às tecnologias oferecidas para a maioria da sociedade (Vencato,2025).

Com o avanço da Tecnologia Assistiva é possível que indivíduos com alguma deficiência possam estar cada dia mais independentes, ou seja, não necessitam de uma pessoa para ajudá-los em todas as tarefas, por exemplo, alguém que possui uma baixa mobilidade pode utilizar o andador para se mover de forma mais segura, sem que outra pessoa esteja acompanhado em todos os momentos do dia e da vida, outro exemplo a ser citado é para pessoas que necessitam da tradução em libras, atualmente muitos lugares já disponibilizam tradutores e se tratando dos celulares, televisões e outros meios de comunicação, esses implantaram a tradução de libras, garantindo assim que essas parcela da população possa ter acesso às informações e entretenimento (Bersch, 2017).

Há cerca de 2,2 bilhões de pessoas com alguma deficiência visual (OMS, 2021), sendo assim, a inclusão das mesmas é precária. Com a falta de inclusão para pessoas com deficiência visual, temos a dificuldade dessas pessoas no mercado de trabalho, em confraternizações, em lazeres como shows, cinema, teatros, esses que são indispensáveis para a cultura do indivíduo. Além da inserção do indivíduo na cultura não só do país, mas do mundo, é importante ressaltar o fato dele não estar socialmente ativo, pois em muitos lugares essa pessoa não conseguirá emprego e/ou condições para que possa exercer o mesmo, sendo assim, ela fica dependente de pessoas a sua volta, algo que interfere de forma negativa em sua vida, podendo desenvolver problemas como autoestima, depressão e outros fatores psicológicos.

O text-to-speech (TTS) é uma ferramenta essencial para o trabalho vigente, visto que o mesmo promove a conversão de textos em áudio. Essa tecnologia permite que os textos que são obtidos em diferentes plataformas possam ser reproduzidos em um áudio no celular ou em outros dispositivos que estejam disponíveis com o TTS em sua plataforma, para maior facilidade ao usuário foi utilizado o recurso oferecido pelo smartphone, como é evidenciado em metodologia (Google, 2025).

A possibilidade de acessibilidade através da áudio transcrição já existe, porém, não são todos os lugares que oferecem esse recurso essencial,com isso, o presente trabalho visa desenvolver um protótipo capaz de identificar objetos expostos a frente do deficiente visual, oferecendo assim uma maior comodidade e incentivando a independência.

2. REVISÃO BIBLIOGRÁFICA

2.1 Tecnologia Assistiva

É possível evidenciar a crescente discussão sobre a tecnologia assistiva (TA) no Brasil e no mundo, pois a promoção da inclusão das pessoas com deficiência junto com a autonomia das mesmas vem ganhando relevância em estudos de como aprimorar os métodos já existentes. No Brasil é evidenciado um progresso em pesquisas científicas tratando sobre a TA, principalmente em cenários educacionais. Para Utsch (2024), a TA “refere-se ao conjunto de recursos e serviços que visam proporcionar maior independência, autonomia e qualidade de vida para pessoas com deficiência”. Segundo Bersch(2017), a classificação da TA pode ser dividida segundo seus objetivos, sendo eles: auxílio para a vida, comunicação aumentativa e alternativa, acessibilidade ao computador, controle de ambiente, projetos arquitetônicos acessíveis, órteses e próteses, adequação postural e mobilidade, recursos para visão, recursos para audição, adaptações para transporte, esporte e lazer.

A TA é “todo o arsenal de recursos e serviços que contribuem para proporcionar ou ampliar habilidades funcionais de pessoas com deficiência e consequentemente promover vida independente e inclusão”, ou seja, a tecnologia assistiva não apenas a inclusão, mas são os meios que essa inclusão ocorre, seja por tecnologias, dispositivos computadorizados, bengalas, óculos, dentre outros aparelhos que auxiliam para pular barreiras existentes e ampliar a participação das pessoas com deficiência na sociedade, Bersch (2017).

Gomes (2023), diz que a TA deve ser compreendida não apenas como instrumentos ou dispositivos, mas como uma estratégia global de inclusão e cidadania, isto é, uma participação maior das pessoas que possuem alguma deficiência no cotidiano como todos são assegurados. No conceito da educação, Gomes pontua que ela é “essencial para viabilizar o acesso ao currículo escolar, a comunicação e a socialização” , sendo assim indispensável na sociedade.

No Brasil, a Lei Brasileira de Inclusão (Lei nº 13.146/2015), assegura que a inclusão de pessoas com deficiência seja feita, ou seja, garante que todas elas possuem o acesso a produtos, estabelecimentos e serviços inclusivos para melhorar sua autonomia e qualidade de vida, além da Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva (2008), onde ratifica o tópico da Convenção sobre os Direitos das Pessoas com Deficiência (ONU, 2006), impondo na educação brasileira o direito de inclusão de crianças com deficiência.

> A educação escolar se faz na convivência entre todas as pessoas, em salas de aulas comuns, reconhecendo e respeitando nossas diferentes formas de comunicar, mover, perceber, relacionar-se, sentir, pensar. Isso implica revisitar constantemente sistemas de ensino, políticas, conceitos e práticas, a fim de transformar nossas escolas para serem mais e mais acessíveis a todas as pessoas (Ministério da Educação, 2025).

2.2 Visão

Segundo a Organização Mundial da Saúde (OMS, 2021) "A visão desempenha um papel crítico em todas as facetas e fases da vida" , para cada passo que nós seres humanos damos é indiscutível a necessidade da visão, desde que saímos do ventre de nossa mãe, o primeiro contato com o ar, com o mundo vem através da visão, quando começa a reconhecer os objetos, as pessoas, os lugares, os animais.

"Num mundo construído sobre a capacidade de ver, a visão, o mais dominante dos nossos sentidos, é vital em todo o momento das nossas vidas" (OMS, 2021), conforme os anos vão se passando a visão se torna algo cada vez mais importante para o desenvolvimento, seja ainda na infância e até a maior idade, é notório a relevância da visão para o desenvolvimento das crianças nas escolas, principalmente pela maioria das vezes se tratar de um primeiro contato não apenas com a alfabetização, mas também o lado psicossocial, onde este se não bem desenvolvido terá marcas para o resto de sua vida. A partir desse primeiro contato com a sociedade nota-se o quão importante será a visão para a sua próspera vida no mundo, onde este é construído com base na capacidade de ver, ou seja, a visão é um fator social, onde quem tem alguma deficiência visual abstém-se das experiências e fundamentos da globalização (OMS, 2021).

> Globalmente, pelo menos 2,2 bilhões de pessoas têm uma deficiência visual ou cegueira, das quais pelo menos 1 milhar de milhões tem uma deficiência visual que poderia ter sido evitada ou ainda não foi tratada (OMS, 2021).

Neste caso, 2,2 bilhões de pessoas no mundo que estão sem acesso a inúmeras oportunidades e vivências sociais.

No Brasil segundo o Censo (2022), 7,9 milhões de pessoas no país sofrem com uma dificuldade ao enxergar, essas que mesmo utilizando óculos ou lentes de contato ainda apresentavam uma certa dificuldade em visualizar bem o ambiente em que estava inserida.

O Censo Demográfico de 2010, foi apresentado que 23,9% da população possuíam alguma deficiência e dentre elas 18,6% era afetada pela deficiência visual e desse valor, 3,46% apresentava de forma severa a deficiência visual. Cerca de 1,6% são completamente cegos desses 3,46% apresentados anteriormente (Censo,2010).

2.3 TEXT-TO-SPEECH

O Text-to-speech é a tecnologia que transforma textos em áudios com uma voz humana, ou o mais próximo de uma. O processo de conversão da entrada de texto em dados de áudio é chamado de síntese, e a saída da síntese é chamada de fala sintética (Google,2025), com isso é possível que se obtenha áudio com uma qualidade superior através apenas de sínteses, ou seja, pode ser dados brutos ou dados SSML (Linguagem de marcação de síntese de fala), caso utilizado este modelo é o que permite ao usuário inserir pausas, pronúncias de siglas, e outros detalhes enviados para o TTS (Google,2025).

Segundo Charlotte Hu (2015), o TSS funciona a partir de uma análise linguística, essa que recebe um conjunto de dados de áudio e transcrições correspondentes, assim ajudando o sistema a combinar o texto com a voz. Para transformar o texto em voz Charlotte Hu (2015) complementa que é necessário duas etapas, a primeira transforma o texto em características adequadas, como um espectrograma, é isso que captura as características das vozes e leva em consideração os detalhes como pronúncia e tempo das palavras.

A etapa seguinte do processo é quando “uma rede de codificação de voz (vocoder) pode transformar os recursos adequados ao tempo em formas de onda de áudio que os computadores podem converter em voz com som natural” (Charlotte Hu, 2015). Isso ajuda na acessibilidade na área da saúde, pois com o Text-To-Speech os profissionais podem se comunicar com os pacientes de maneira acessível, deixando assim de modo inclusivo o atendimento (Charlotte Hu, 2015).

3. METODOLOGIA

Para o desenvolvimento do presente trabalho acadêmico, foi utilizado diferentes metodologias.

Na imersão sobre o tema proposto foi realizado uma revisão bibliográfica, sendo ela caracterizada por analisar obras que estão relacionadas com o tema do trabalho, assim garantindo uma contextualização, além de promover um maior conhecimento sobre a deficiência e como as tecnologias estão buscando melhorar a acessibilidade em um mundo globalizado.

Em um outro momento foi indicado os materiais e softwares a serem utilizados no protótipo, esses foram separados conforme o quadro abaixo:

Quadro 1

Materiais	Software
Esp32-cam	Arduino IDE
FTDI[1]	Edge Impulse
Óculos de proteção	

Autora,2025

[1] FTDI: Future Technology Devices International, fabricante de chips conversores USB-Serial

.

A figura 1, mostra a Esp32-cam e o FTDI, um programador usado para importar o código feito através do arduino IDE (Figura 2).

Figura 1

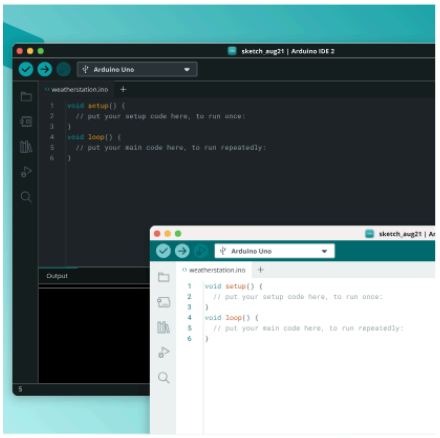
Esp32-Cam

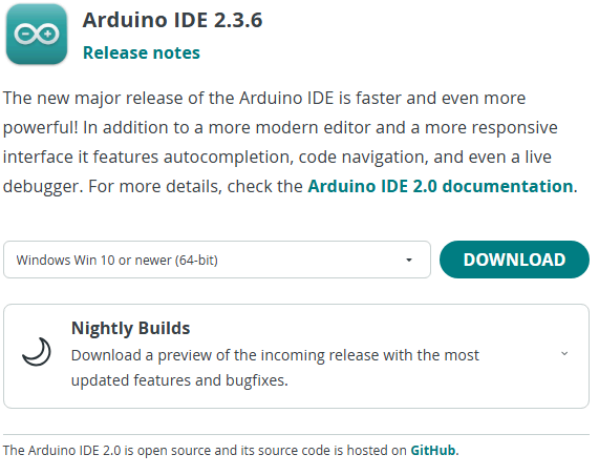
Autora,2025

FTDI

Autora,205

Figura 2

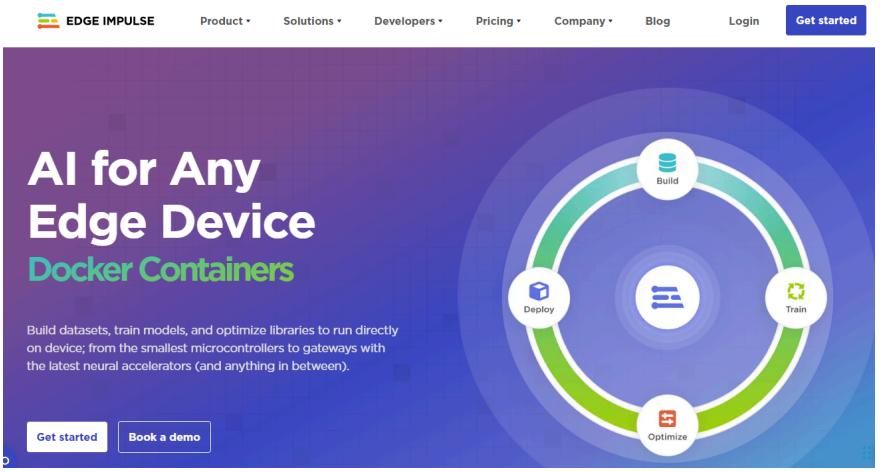




Autora,2025

Na figura 3 é possível observar a interface do Edge Impulse, site utilizado para o treinamento da I.A.

Figura 3



Autora,2025

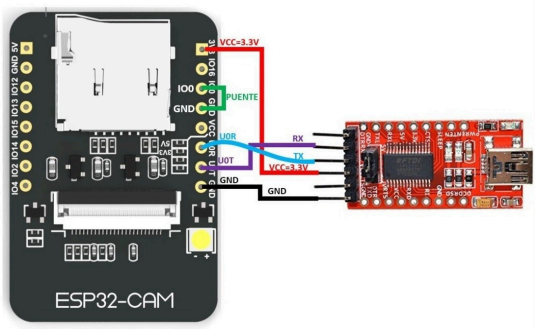
Para o funcionamento em conjunto, foi utilizado um óculos de proteção (figura 4), esse pensado em um custo benefício menor com contraste em sua utilidade ao protótipo.

Figura 4

Autora,2025

Após a definição dos materiais e software a serem utilizados começou a montagem do protótipo, a primeiro momento foi realizada a ligação do Esp32-cam com o FDTI no computador (figura 5), fazendo assim uma conexão para a captura das imagens.

Figura 5



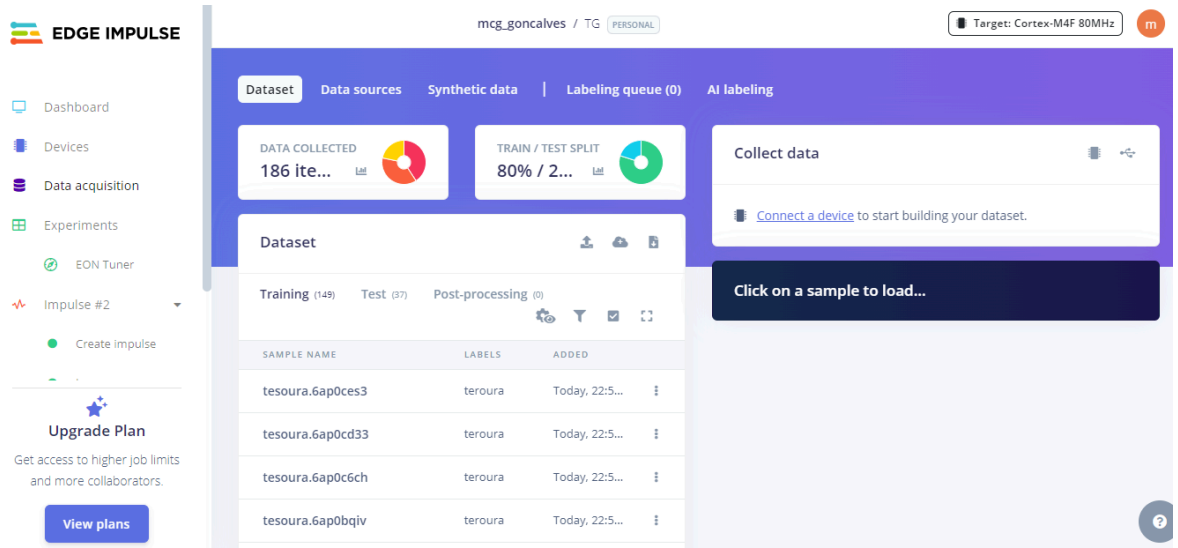
Aseef, 2020

Conseguindo ter acesso a câmera da Esp32-cam através do WebCameraServer, foi coletado imagens dos objetos, como essas imagens seriam passadas para uma inteligência artificial na plataforma Edge Impulse, o total de imagens de cada objeto resultou em 50, dessas é importante ressaltar que foram imagens que diferenciavam ângulos e distância. Os objetos utilizados para obter as imagens foram, uma tesoura, uma caneta e uma calculadora. Com essas imagens obtidas, foi possível realizar o upload delas na plataforma Edge Impulse e assim treinar a I.A que faz a classificação das imagens de acordo com as fotos enviadas.

Para o treinamento da inteligência artificial vinculada ao Edge Impulse foi seguido os seguintes passos:

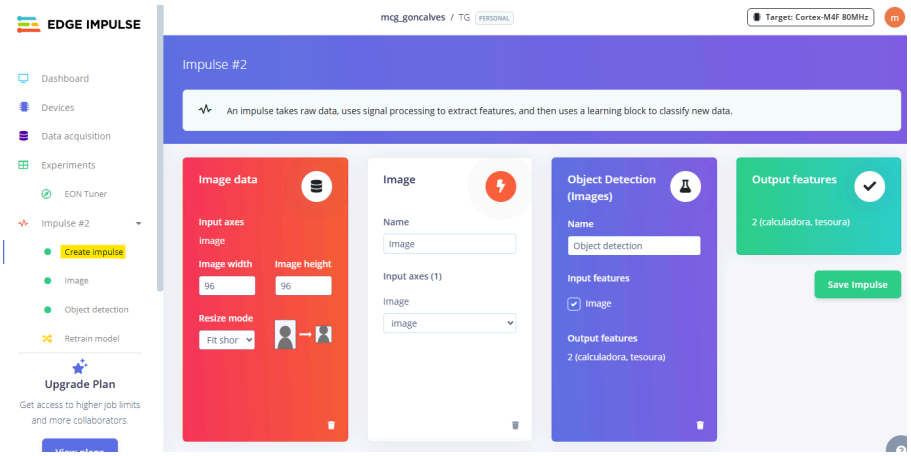
1. Aquisição das imagens (50 imagens para cada objeto) em “Data acquisition” e feito a classificação das imagens obtidas;
2. Em “create impulse” foi definido o tamanho da imagem e nome das imagens, seguindo as classificações feitas anteriormente;
3. Para verificar como a I.A estava interpretando as imagens, foram coletadas utilizando diferentes ângulos e distâncias. Com as classes foi feito em “Image” um gráfico que mostra como está a separação da detecção dos objetos;
4. O treinamento da inteligência artificial foi realizado no campo de “Object detection”, nesse espaço foi possível visualizar a performance dos resultados que foram entregues ao usuário;
5. Posteriormente foi utilizado a guia “Deployment”, essa permite baixar toda a biblioteca com a classificação dos objetos feita pelo site

Figura 6



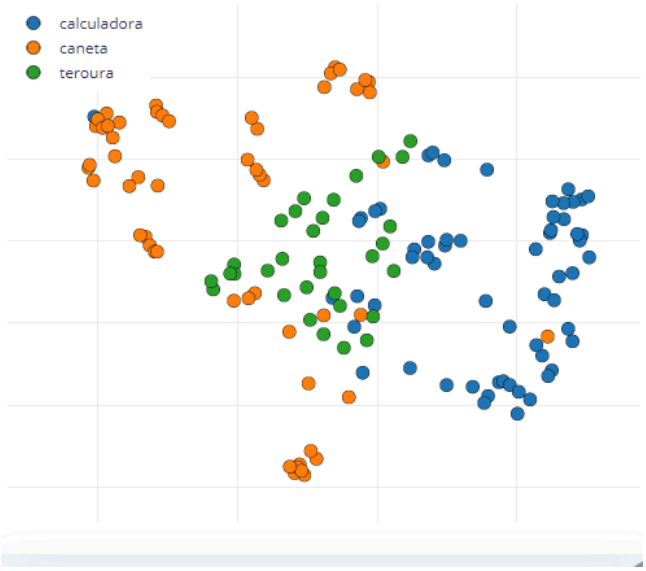
Autora,2025

Figura 7



Autora,2025

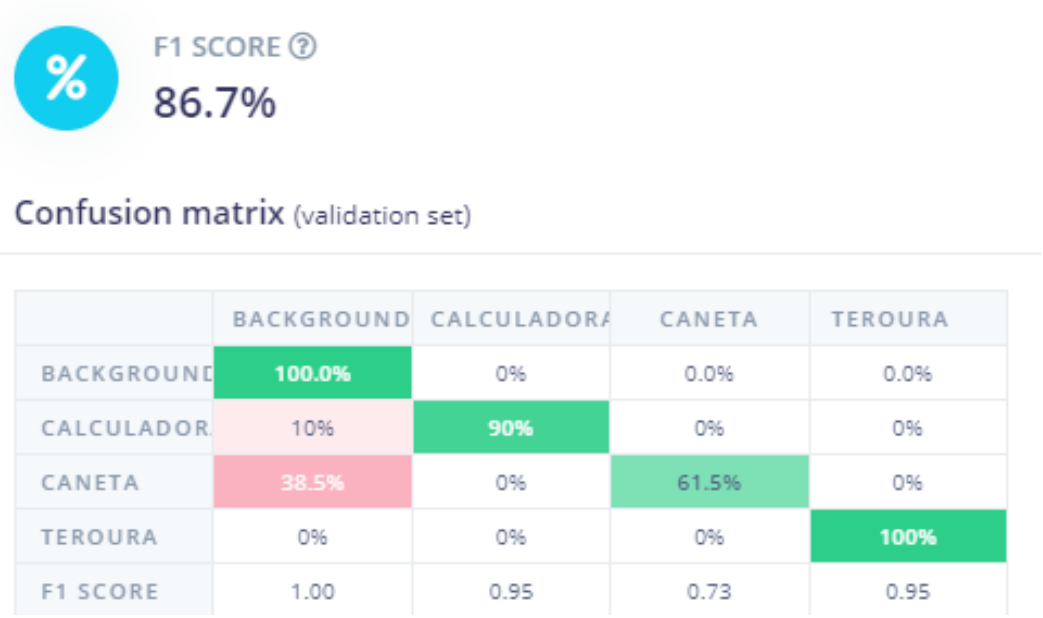
Figura 8



Autora,2025

A figura 8 mostra a distribuição de como a I.A está reconhecendo os objetos, a calculadora que está em azul é perceptível seu deslocamento em relação a caneta, assim como essa está em relação a tesoura representada pela cor verde.

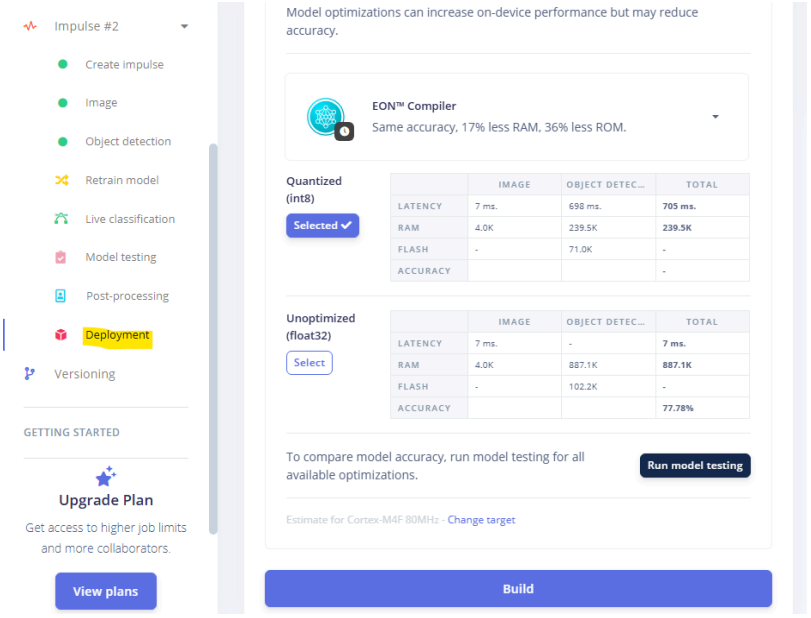
Figura 9



Autora,2025

A matriz de confusão apresentada avalia o desempenho da I.A ao ler os objetos. Nota-se que as classes background, calculadora e tesoura apresentaram altos índices de acerto, com F1-Score de 1.00, 0.95 e 0.95, isso indica que possui uma excelente capacidade de distinção pelo modelo. Em outro lado, a classe caneta apresentou maior taxa de erro, sendo assim, pode ser confundida com o background, o que resultou em um F1-Score de 0.73. Esses resultados mostram que a classe caneta é facilmente confundida, enquanto as demais classes demonstram desempenho satisfatório no processo de classificação.

Figura 10



Autora,2025

Após a biblioteca baixada, foi inserida ao Arduino IDE, o código inicial que a biblioteca fornece apenas para verificar se a Esp32-cam está fazendo o reconhecimento dos objetos, porém como projeto conta com a audiodescrição deles, foi necessário fazer alguns ajustes para que quando fosse detectado o objeto fosse enviado automaticamente ao usuário. Para uma melhor visualização do que se foi realizado, abaixo esta partes do código principal, já o seu todo está em Anexo 1.

Figura 11

```
//Biliotecas incluidas
#include <teste_inferencing.h> // biblioteca gerada
#include "edge-impulse-sdk/dsp/image/image.hpp"
```

Autora,2025

A figura 11 mostra a biblioteca que foi incluída, essa gerada pelo site Edge Impulse e contém as informações do treinamento realizado na inteligência artificial para o reconhecimento dos objetos pré determinados.

Figura 12

```
//Determina variavel que envia ao celular
String objetoDetectado;
WebServer server(80);
```

Autora,2025

Figura 13

```
//Envia sinal de Wi-Fi da Esp  para o celular

WiFi.softAP("ESP32CAM_Voz", "12345678"); // cria a rede AP da esp
IPAddress IP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(IP);
server.on("/", handleRoot);
server.begin();
Serial.println("Servidor iniciado");
server.on("/data", [](){
server.send(200, "text/plain", objetoDetectado);
});
```

Autora,2025

Na figura 13 mostra o código utilizado para ativar o AP do Esp32-cam, esse que funciona como um roteador, liberando assim o sinal para que o celular possa se comunicar através de seu número de IP, esse é enviado para o monitor serial, sendo assim, quando colocado na barra de pesquisa do chrome é possível ter acesso as demais informações do código e ao corpo da página web criada, como mostra a figura 14.

Figura 14

```
//Pagina web para audiodescrição
   void handleRoot() {
    String html = R"=====(
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name='viewport' content='width=device-width, initial-scale=1.0'>
<style>
body { font-family: Arial; text-align: center; margin-top: 50px; font-size: 24px; }
#objeto { font-weight: bold; color: #007bff; }
button {
  padding: 12px 20px;
  font-size: 20px;
  margin: 20px;
  border-radius: 10px;
}
</style>
</head>
<body>
<h2>Objeto detectado:</h2>
<div id="objeto">---</div>
```

Autora,2025

Figura 15

```
<!--  Botão obrigatório para liberar TTS -->
<button onclick="ativarTTS()">Clique aqui para ativar áudio</button>
<script>
let ultimo = "";
let ttsAtivo = false;
function ativarTTS() {
    ttsAtivo = true;
    let u = new SpeechSynthesisUtterance("Áudio ativado");
    u.lang = 'pt-BR';
    speechSynthesis.speak(u);
```

Autora,2025

Figura 16

```
#if EI_CLASSIFIER_OBJECT_DETECTION == 1
    ei_printf("Object detection bounding boxes:\r\n");
    for (uint32_t i = 0; i < result.bounding_boxes_count; i++) {
        ei_impulse_result_bounding_box_t bb = result.bounding_boxes[i];
        if (bb.value == 0) {
            continue;
        }
        ei_printf("  %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\r\n",
                bb.label,
                bb.value,
                bb.x,
                bb.y,
                bb.width,
                bb.height);

                //Qual nome do objeto deve ser enviado
                objetoDetectado = String(bb.label);
```

Autora,2025

Para a atribuição das informações da resposta da Esp32-cam para o celular, foi preferível utilizar o Wi-Fi já existente pela placa, sendo assim ela funciona como um roteador liberando um sinal para que o celular possa se comunicar virtualmente e caso fosse utilizado o Bluetooth a placa não suportaria pelo nível de processamento e envio de informações.

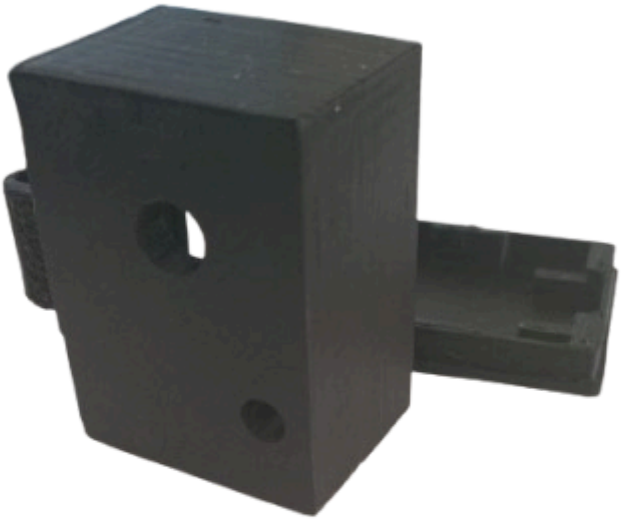
A página web é possível ser acessada através do IP da Esp32-cam, esse que foi coletado primeiramente pelo monitor serial e usado como padrão para os próximos acessos. Nesta página a primeiro modo é necessário conceder permissão para que saia aúdio dela, esse princípio é dos próprios navegadores, sendo assim não foi possível deixar de adicionar o botão para que o usuário conceda esse ponto no próprio celular.

No design foi analisado as opções e a que mais aderiu ao projeto proposto foi a utilização de um óculos (figura 17) com um suporte para a Esp32-cam (figura 18), assim ela é capaz de capturar a imagem a altura dos olhos do usuário.

figura 17

Autora,2025

figura 18



Autora,2025

Ambas escolhas foram consideradas pela facilidade de mercado e pelo custo benefício oferecido comparado a sua qualidade e empregabilidade ao protótipo.

4. RESULTADOS E DISCUSSÃO

4.1 TESTES

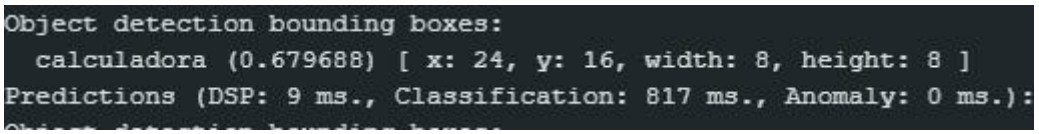
Foram realizados diversos testes ao longo do trabalho desenvolvido, os principais teste foram:

1. Teste para reconhecimento da imagem: nesse teste foi possível visualizar pelo monitor serial do Arduino IDE que a Esp32-cam estava detectando corretamente os objetos que eram expostos à câmera.
2. Teste da saída de áudio no celular: com esse teste foi capaz de verificar se a conexão entre o celular e o Esp32-cam estava de fato acontecendo e se a saída de áudio estava devidamente funcionando.
3. Teste de acurácia: neste teste foi analisado com amostras o quão preciso está o dispositivo.
4. Teste com o Esp32-cam no óculos: com esse teste final, foi verificado a funcionalidade do projeto final, sendo a junção da Esp32-cam com o óculos, assim sendo possível analisar a operação do sistema como um todo.

4.2 RESULTADOS

4.2.1 Teste de reconhecimento de imagem: foi satisfatório e apresentou bom resultado, pois reconhece o objeto e mostra o resultado no monitor serial do Arduino IDE.

Figura 19



Autora, 2025

4.2.2 Teste da saída de áudio no celular: o teste apresentou resultado satisfatório, ressaltando que para que o som seja emitido pelo smartphone é necessário aceitar o pedido para a liberação do TTS.

4.2.3 Teste de acurácia, sensibilidade e especificidade: segue em tabelas os resultados dos testes realizados:

Resultado do teste de acurácia da tesoura

Verdadeiro positivo= 6	Falso positivo= 3
Falso negativo= 2	Verdadeiro negativo= 3

$$\frac{Vp+Vn}{Vp+Vn+Fn+Fp}=\frac{9}{14} = 0,6428 = 64,3\%$$

Resultado do teste de sensibilidade da tesoura

$$\frac{Vp}{Vp+Fn} =\frac{6}{8}= 0,75 = 75\%$$

Resultado do teste de especificidade da tesoura

$$\frac{Vn}{Fp+Vn} = \frac{3}{6} = 0,50 = 50\%$$

Resultado do teste de acurácia da calculadora: o resultado se baseou na mesma fórmula apresentada anteriormente, sendo assim:

Verdadeiro positivo= 5	Falso positivo= 1
Falso negativo= 1	Verdadeiro negativo= 3

$$\frac{8}{10} = 0,8 = 80\%$$

Resultado do teste de sensibilidade da calculadora

$$\frac{5}{6} = 0,8333 = 83,3\%$$

Resultado do teste de especificidade da tesoura

$$\frac{3}{4} = 0,75 = 75\%$$

Resultado do teste de acurácia da caneta: foi efetuado os mesmos procedimentos relatados, obtendo assim os seguintes resultados:

Verdadeiro positivo= 3	Falso positivo= 1
Falso negativo= 5	Verdadeiro negativo= 2

$$\frac{5}{11} = 0,4545 = 45,5\%$$

Resultado do teste de sensibilidade da caneta

$$\frac{3}{8} = 0,375 = 37,5\%$$

Resultado do teste de especificidade da tesoura

$$\frac{2}{3} = 0,6666 = 66,67\%$$

Posteriormente a estes testes o resultado conclusivo é que a calculadora apresentou o melhor desempenho geral, com 80% de acurácia, aliando alta sensibilidade (83,3%) e boa especificidade (75%).A tesoura apresentou desempenho intermediário, comparada aos testes da calculadora, com 64,3% de acurácia, sensibilidade de 75%, mas especificidade moderada de apenas 50%.

Onde se obteve um número menor foi nos testes com a caneta, ou seja, o dispositivo apresentou uma maior dificuldade ao reconhecer esse objeto, apresentando assim uma sensibilidade baixa de 37,5%.

4.2.4 Teste com o Esp32-cam no óculos: Nesse teste foi utilizado o sistema completo, ou seja, a Esp32-cam no suporte e acoplado ao óculos, com isso foi possível observar se o protótipo funcionaria mesmo com um ambiente mais restrito, foi realizado o total de 20 testes para reconhecimento e esses mostrou 80% de resultados correspondentes ao objeto detectado, sendo assim, mostrou um resultado satisfatório aos testes submetido.

5. CONCLUSÃO

O projeto de audiodescrição de objetos para pessoas com deficiência visual foi bem executado, trazendo assim uma maneira de inclusão para os usuários, com a ajuda da tecnologia assistiva e com o avanço da inteligência artificial para reconhecimentos de imagens, pode-se notar que haverá grandes mudanças no cenário da acessibilidade, dando assim oportunidades para pessoas que anteriormente eram reclusas da sociedade.

Para uma sugestão futura desse projeto, seria de valia a projeção de um aplicativo, assim o processamento não dependeria apenas da Esp32-cam, trazendo uma maior eficácia no processamento e na capacidade de armazenamento das imagens, além de deixar de depender do Wi-fi da Esp32-cam, ou seja, o usuário não fica sem internet quando conectado ao óculos.

6. REFERÊNCIAS

ASEEF. **Serial Communication between an ESP32-CAM and Arduino Mega2560**. 2020. Disponível em: https://forum.arduino.cc/t/help-serial-communication-between-an-esp32-cam-and-arduino-mega2560/684313/9. Acesso em: 25 out. 2025.

BERSCH, R. D. C. R. (2017). Introdução à tecnologia assistiva. Porto Alegre: CEDI. Disponível em: https://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf. Acesso em: 18 ago 2025.

BRASIL. **Cartilha do Censo 2010**. Brasília: Secretaria Nacional de Promoção dos Direitos da Pessoa Com Deficiência, 2012. Disponível em: https://siac.fpabramo.org.br/uploads/acaoinstitucional/SNPDPCD_censo_2012.pdf. Acesso em: 24 out. 2025.

BRASIL. **Censo 2022: Brasil tem 14,4 milhões de pessoas com deficiência**. 2022. IBGE. Disponível em: https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/notici

as/43463-censo-2022-brasil-tem-14-4-milhoes-de-pessoas-com-deficiencia. Acesso em: 24 out. 2025.

BRASIL, Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. **Manual de Adaptações de Acessibilidade**. Brasília: Ministério da Mulher, da Família e dos Direitos Humanos, 2020. Disponível em: https://www.gov.br/mdh/pt-br/navegue-por-temas/pessoa-com-deficiencia/publicacoes/manual_mdhc_laudos.pdf. Acesso em: 07 nov. 2025.

CHARLOTTE HU. **O que é Text to Speech?** Disponível em: https://www.ibm.com/br-pt/think/topics/text-to-speech. Acesso em: 18 out. 2025.

GOOGLE (org.). **Cloud Text-to-Speech basics**. Disponível em: https://docs.cloud.google.com/text-to-speech/docs/basics. Acesso em: 18 out. 2025.

HENRIQUE, Eduardo. **Introdução ao ESP32-CAM**. Disponível em: https://blog.eletrogate.com/introducao-ao-esp32-cam/. Acesso em: 25 ago. 2025.

Osama, M., Yehia, A., Mohamed, S., Sherief, R., Elmasry, N., Adel, V., and Hamdy, A. (2021). Design and implementation of visually impaired assistant system. In Int. Mobile, Intelligent, and Ubiquitous Comp. Conf., pages 303–310. Acesso em: 20 ago 2025.

SONI, Gaurav; SAINI, Satnam Singh; MALHI, Simarjit Singh; SRAO, Bhupinder Kaur; SHARMA, Ashim; PURI, Digvijay. Design and Implementation of Object Motion Detection Using Telegram. **2021 International Conference On Technological Advancements And Innovations (Ictai)**, [S.L.], p. 203-206, 10 nov. 2021. IEEE. http://dx.doi.org/10.1109/ictai53825.2021.9673226. Disponível em: https://ieeexplore.ieee.org/abstract/document/9673226. Acesso em: 25 ago. 2025.

Supekar, A. and Patil, S. (2022). Design and development of portable navigation system for disabled person using image, text and audio. In IEEE Delhi Section Conference (DELCON), pages 1–4. Acesso em: 20 ago 2025

VENCATO, Anna Paula. **POR UMA POLÍTICA DO AFETO E DO CUIDADO: ACESSO EPERMANÊNCIA DE PESSOAS COM DEFICIÊNCIA NASUNIVERSIDADES**. Disponível em: https://doi.org/10.1590/SciELOPreprints.11916. Acesso em: 07 nov. 2025.

World Health Organization (2019). World Report on Vision. World Health Organization, Geneva, Switzerland. Acesso em: 22 ago 2025

ANEXO 1

```
//Biliotecas incluidas
#include <teste_inferencing.h> //gerada pelo edge impulse
#include "edge-impulse-sdk/dsp/image/image.hpp"

#include "esp_camera.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>

//Define qual modelo de camera esta usando
#define CAMERA_MODEL_AI_THINKER

//Define o pino do flash
#define FLASH_PIN 4

#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#else
#error "Camera model not selected"
#endif

#define EI_CAMERA_RAW_FRAME_BUFFER_COLS           320
#define EI_CAMERA_RAW_FRAME_BUFFER_ROWS           240
#define EI_CAMERA_FRAME_BYTE_SIZE                 3

static bool debug_nn = false;
static bool is_initialised = false;
uint8_t *snapshot_buf;

static camera_config_t camera_config = {
    .pin_pwdn = PWDN_GPIO_NUM,
    .pin_reset = RESET_GPIO_NUM,
```

```
    .pin_xclk = XCLK_GPIO_NUM,
    .pin_sscb_sda = SIOD_GPIO_NUM,
    .pin_sscb_scl = SIOC_GPIO_NUM,

    .pin_d7 = Y9_GPIO_NUM,
    .pin_d6 = Y8_GPIO_NUM,
    .pin_d5 = Y7_GPIO_NUM,
    .pin_d4 = Y6_GPIO_NUM,
    .pin_d3 = Y5_GPIO_NUM,
    .pin_d2 = Y4_GPIO_NUM,
    .pin_d1 = Y3_GPIO_NUM,
    .pin_d0 = Y2_GPIO_NUM,
    .pin_vsync = VSYNC_GPIO_NUM,
    .pin_href = HREF_GPIO_NUM,
    .pin_pclk = PCLK_GPIO_NUM,

    //XCLK 20MHz or 10MHz for OV2640 double FPS (Experimental)
    .xclk_freq_hz = 20000000,
    .ledc_timer = LEDC_TIMER_0,
    .ledc_channel = LEDC_CHANNEL_0,

    .pixel_format = PIXFORMAT_JPEG, //YUV422,GRAYSCALE,RGB565,JPEG
      .frame_size =  FRAMESIZE_QVGA,      //QQVGA-UXGA Do not use sizes
above QVGA when not JPEG

    .jpeg_quality = 12, //0-63 lower number means higher quality
     .fb_count = 1,         //if more than one, i2s runs in continuous
mode. Use only with JPEG
    .fb_location = CAMERA_FB_IN_PSRAM,
    .grab_mode = CAMERA_GRAB_WHEN_EMPTY,
};

bool ei_camera_init(void);
void ei_camera_deinit(void);
bool  ei_camera_capture(uint32_t  img_width,  uint32_t  img_height,  uint8_t
*out_buf) ;

    //Determina o que deve enviar ao celular
    String objetoDetectado;
    WebServer server(80);
/**
* @brief      Arduino setup function
*/
void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
     //comment out the below line to start inference immediately after
upload
    while (!Serial);
    Serial.println("Edge Impulse Inferencing Demo");
```

```
    if (ei_camera_init() == false) {
        ei_printf("Failed to initialize Camera!\r\n");
    }
    else {
        ei_printf("Camera initialized\r\n");

        //Liga o flash
        pinMode(FLASH_PIN, OUTPUT);
        digitalWrite(FLASH_PIN, LOW);
    }

    ei_printf("\nStarting continious inference in 2 seconds...\n");
    ei_sleep(2000);

    //Envia sinal de Wi-Fi da Esp  para o celular

    WiFi.softAP("ESP32CAM_Voz", "12345678"); // cria a rede AP da esp
    IPAddress IP = WiFi.softAPIP();
    Serial.print("AP IP address: ");
    Serial.println(IP);
    server.on("/", handleRoot);
    server.begin();
    Serial.println("Servidor iniciado");
    server.on("/data", [](){
    server.send(200, "text/plain", objetoDetectado);
    });
}

//Pagina web para audiodescrição
   void handleRoot() {
    String html = R"=====(
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name='viewport' content='width=device-width, initial-scale=1.0'>
<style>
body { font-family: Arial; text-align: center; margin-top: 50px;
font-size: 24px; }
#objeto { font-weight: bold; color: #007bff; }
button {
  padding: 12px 20px;
  font-size: 20px;
  margin: 20px;
  border-radius: 10px;
}
</style>
</head>
<body>
<h2>Objeto detectado:</h2>
<div id="objeto">---</div>
```

```
<!-- ✅ Botão obrigatório para liberar TTS -->
<button onclick="ativarTTS()">Clique aqui para ativar áudio</button>
<script>
let ultimo = "";
let ttsAtivo = false;
function ativarTTS() {
    ttsAtivo = true;
    let u = new SpeechSynthesisUtterance("Áudio ativado");
    u.lang = 'pt-BR';
    speechSynthesis.speak(u);
}
function atualizar() {
    fetch('/data')
      .then(r => r.text())
      .then(txt => {

        document.getElementById('objeto').innerText = txt;

         if (ttsAtivo && txt !== ultimo && txt !== "" && txt !== "null"
&& txt !== "---") {
            ultimo = txt;
            let u = new SpeechSynthesisUtterance(txt);
            u.lang = 'pt-BR';
            speechSynthesis.speak(u);
        }
      });
}
setInterval(atualizar, 1200);
</script>
</body>
</html>
    )=====";

    server.send(200, "text/html", html);
}
/**
* @brief      Get data and run inferencing
*
* @param[in]  debug  Get debug info if true
*/
void loop()
{
    //Ligar flash da esp
    digitalWrite(FLASH_PIN, HIGH);

    //Conexão do Wi-fi
    server.handleClient();

      // instead of wait_ms, we'll wait on the signal, this allows
threads to cancel us...
    if (ei_sleep(5) != EI_IMPULSE_OK) {
```

```
        return;
    }

      snapshot_buf = (uint8_t*)malloc(EI_CAMERA_RAW_FRAME_BUFFER_COLS *
EI_CAMERA_RAW_FRAME_BUFFER_ROWS * EI_CAMERA_FRAME_BYTE_SIZE);

    // check if allocation was successful
    if(snapshot_buf == nullptr) {
        ei_printf("ERR: Failed to allocate snapshot buffer!\n");
        return;
    }

    ei::signal_t signal;
               signal.total_length    =    EI_CLASSIFIER_INPUT_WIDTH    *
EI_CLASSIFIER_INPUT_HEIGHT;
    signal.get_data = &ei_camera_get_data;

             if    (ei_camera_capture((size_t)EI_CLASSIFIER_INPUT_WIDTH,
(size_t)EI_CLASSIFIER_INPUT_HEIGHT, snapshot_buf) == false) {
        ei_printf("Failed to capture image\r\n");
        free(snapshot_buf);
        return;
    }

    // Run the classifier
    ei_impulse_result_t result = { 0 };

    EI_IMPULSE_ERROR err = run_classifier(&signal, &result, debug_nn);
    if (err != EI_IMPULSE_OK) {
        ei_printf("ERR: Failed to run classifier (%d)\n", err);
        return;
    }

    // print the predictions
       ei_printf("Predictions  (DSP:  %d  ms.,  Classification:  %d  ms.,
Anomaly: %d ms.): \n",
                        result.timing.dsp, result.timing.classification,
result.timing.anomaly);

#if EI_CLASSIFIER_OBJECT_DETECTION == 1
    ei_printf("Object detection bounding boxes:\r\n");
    for (uint32_t i = 0; i < result.bounding_boxes_count; i++) {
        ei_impulse_result_bounding_box_t bb = result.bounding_boxes[i];
        if (bb.value == 0) {
            continue;
        }
           ei_printf("   %s  (%f)  [ x: %u, y: %u, width: %u, height: %u
]\r\n",
                bb.label,
                bb.value,
                bb.x,
```

```
                bb.y,
                bb.width,
                bb.height);

                //Qual nome do objeto deve ser enviado
                objetoDetectado = String(bb.label);

    }

    // Print the prediction results (classification)
#else
    ei_printf("Predictions:\r\n");
    for (uint16_t i = 0; i < EI_CLASSIFIER_LABEL_COUNT; i++) {
        ei_printf("  %s: ", ei_classifier_inferencing_categories[i]);
        ei_printf("%.5f\r\n", result.classification[i].value);
    }
#endif

    // Print anomaly result (if it exists)
#if EI_CLASSIFIER_HAS_ANOMALY
    ei_printf("Anomaly prediction: %.3f\r\n", result.anomaly);
#endif

#if EI_CLASSIFIER_HAS_VISUAL_ANOMALY
    ei_printf("Visual anomalies:\r\n");
    for (uint32_t i = 0; i < result.visual_ad_count; i++) {
                            ei_impulse_result_bounding_box_t    bb    =
result.visual_ad_grid_cells[i];
        if (bb.value == 0) {
            continue;
        }
           ei_printf("   %s (%f) [ x: %u, y: %u, width: %u, height: %u
]\r\n",
                bb.label,
                bb.value,
                bb.x,
                bb.y,
                bb.width,
                bb.height);

                //Qual nome do objeto deve ser enviado

                objetoDetectado = String(bb.label);


    }
#endif


    free(snapshot_buf);
```

```
}

/**
 * @brief   Setup image sensor & start streaming
 *
 * @retval  false if initialisation failed
 */
bool ei_camera_init(void) {

    if (is_initialised) return true;

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

    //initialize the camera
    esp_err_t err = esp_camera_init(&camera_config);
    if (err != ESP_OK) {
      Serial.printf("Camera init failed with error 0x%x\n", err);
      return false;
    }

    sensor_t * s = esp_camera_sensor_get();
      // initial sensors are flipped vertically and colors are a bit
saturated
    if (s->id.PID == OV3660_PID) {
      s->set_vflip(s, 1); // flip it back
      s->set_brightness(s, 1); // up the brightness just a bit
      s->set_saturation(s, 0); // lower the saturation
    }

#if defined(CAMERA_MODEL_M5STACK_WIDE)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
#elif defined(CAMERA_MODEL_ESP_EYE)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
    s->set_awb_gain(s, 1);
#endif

    is_initialised = true;
    return true;
}

/**
 * @brief      Stop streaming of sensor data
 */
void ei_camera_deinit(void) {

    //deinitialize the camera
```

```
    esp_err_t err = esp_camera_deinit();

    if (err != ESP_OK)
    {
        ei_printf("Camera deinit failed\n");
        return;
    }

    is_initialised = false;
    return;
}


/**
 * @brief      Capture, rescale and crop image
 *
 * @param[in]  img_width     width of output image
 * @param[in]  img_height    height of output image
 * @param[in]  out_buf       pointer to store output image, NULL may be
used
 *                           if ei_camera_frame_buffer is to be used
for capture and resize/cropping.
 *
 * @retval     false if not initialised, image captured, rescaled or
cropped failed
 *
 */
bool ei_camera_capture(uint32_t img_width, uint32_t img_height, uint8_t
*out_buf) {
    bool do_resize = false;

    if (!is_initialised) {
        ei_printf("ERR: Camera is not initialized\r\n");
        return false;
    }

    camera_fb_t *fb = esp_camera_fb_get();

    if (!fb) {
        ei_printf("Camera capture failed\n");
        return false;
    }

      bool  converted  =  fmt2rgb888(fb->buf,  fb->len,  PIXFORMAT_JPEG,
snapshot_buf);

   esp_camera_fb_return(fb);

   if(!converted){
       ei_printf("Conversion failed\n");
       return false;
```

```
    }

    if ((img_width != EI_CAMERA_RAW_FRAME_BUFFER_COLS)
        || (img_height != EI_CAMERA_RAW_FRAME_BUFFER_ROWS)) {
        do_resize = true;
    }

    if (do_resize) {
        ei::image::processing::crop_and_interpolate_rgb888(
        out_buf,
        EI_CAMERA_RAW_FRAME_BUFFER_COLS,
        EI_CAMERA_RAW_FRAME_BUFFER_ROWS,
        out_buf,
        img_width,
        img_height);
    }


    return true;
}

static int ei_camera_get_data(size_t offset, size_t length, float
*out_ptr)
{
      // we already have a RGB888 buffer, so recalculate offset into
pixel index
    size_t pixel_ix = offset * 3;
    size_t pixels_left = length;
    size_t out_ptr_ix = 0;

    while (pixels_left != 0) {
        // Swap BGR to RGB here
        // due to https://github.com/espressif/esp32-camera/issues/379
           out_ptr[out_ptr_ix] = (snapshot_buf[pixel_ix + 2] << 16) +
(snapshot_buf[pixel_ix + 1] << 8) + snapshot_buf[pixel_ix];

        // go to the next pixel
        out_ptr_ix++;
        pixel_ix+=3;
        pixels_left--;
    }
    // and done!
    return 0;
}
#if    !defined(EI_CLASSIFIER_SENSOR)    ||    EI_CLASSIFIER_SENSOR    !=
EI_CLASSIFIER_SENSOR_CAMERA
#error "Invalid model for current sensor"
#endif
```