APLICAÇÃO DE IMAGE CAPTIONING EM PROTÓTIPO DE ÓCULOS INTELIGENTES PARA DEFICIENTES VISUAIS

IMAGE CAPTIONING APPLIED ON SMARTGLASS PROTOTYPE FOR VISUALLY IMPAIRED

João Marcos Santos[1]
Renner Belo de Souza[1]
Fabio Andrijauskas[2]
joao.marcossantos@outlook.com | rennerbelo@gmail.com | fabio.andrijauskas@usf.edu.br

[1]Aluno do Curso de Engenharia de Computação
[2]Professor Orientador

RESUMO. Com cerca de 39 milhões de pessoas com visão nula no mundo, o avanço tecnológico voltado às práticas de acessibilidade é um expoente importante para possibilitar maior qualidade de vida e independência para pessoas com barreiras visuais. Desta forma, com o passar dos anos, cada vez mais novas tecnologias assistivas surgem para auxiliar pessoas em seu cotidiano. O uso de *machine learning* para identificação de objetos e caracteres juntamente com a conversão de texto para fala são exemplos de como aplicações de tecnologias computacionais podem ser aplicadas em dispositivos para a substituição visual, como os produtos denominados óculos inteligentes. Contudo, o custo e a disponibilidade de tais tecnologias podem ser um impasse para a disseminação para grande parte do público. Dessa forma, no presente artigo é apresentado o desenvolvimento de um dispositivo *wearable*, que se utiliza de artifícios de visão computacional, possibilitando que pessoas com ausência ou dificuldade de visão possam obter uma descrição do cenário em que se encontram, através de um retorno em som. Para isso, uma abordagem diferente com *machine learning* é utilizada, o *image captioning*, onde *features* das imagens são analisadas por meio de Redes Neurais de Convolução e submetidas a um processamento de linguagem natural com Redes Neurais Recursivas, em especial a *long short-term memory* (LSTM), para ocorrer a descrição da imagem. Utilizando-se de impressões 3D para materializar o protótipo e uma *Single Board Computer* Raspberry PI 4 para capturar as imagens através de uma pequena câmera, os óculos inteligentes buscam uma nova abordagem no quesito de dispositivos tangentes a acessibilidade, possibilitando uma maior contextualização do ambiente para o usuário.

PALAVRAS-CHAVE: aprendizado de máquina, visão computacional, *image captioning*, tecnologia assistiva, óculos inteligentes.

ABSTRACT: With about 39 million sight-impaired people with zero vision in the world, technological advances aimed at accessibility practices are an important exponent for enabling greater quality of life and independence for people with visual difficulty. Therefore, over the years, more and more new assistive technologies have emerged to help people in their daily lives. The use of machine learning for object and character identification along with text-to-speech conversion are examples of how applications of computational technologies can be applied in devices for visual replacement, such as products called smart glasses. However, the cost and availability of such technologies can be an impasse for dissemination to a large part of the public. Thus, this article presents the development of a wearable device, which uses computer vision, enabling people with visual impairment or difficulty to obtain a description of the scenario in which they find themselves, through sound feedback. For this, a different approach with machine learning is used, image captioning, where image features are analyzed

through Convolution Neural Networks and submitted to natural language processing with Recursive Neural Networks, especially long short-term memory (LSTM), to occur the description of the image. Using 3D prints to materialize the prototype and a Single Board Computer Raspberry PI 4 to capture the images through a small camera, the smart glasses seek a new approach in terms of accessibility-tangential devices, enabling greater contextualization of the environment for the user.

KEYWORDS: machine learning, computer vision, image captioning, assistive technology, smart glasses.

INTRODUÇÃO

Uma das questões sociais mais importantes proporcionadas pela evolução e implementação tecnológica é a acessibilidade. Com base em estimativas, existem no mundo cerca de 285 milhões de pessoas com deficiência visual, sendo 39 milhões cegas (PASCOLINI, 2012) e, portanto, sendo beneficiadas com a utilização de tecnologias assistivas (TA). Aparelhos e programas de leitura de texto (TTS), suportes de navegação e marcadores em braile foram desenvolvidos ao longo dos anos para auxiliar indivíduos com dificuldade visuais (ADVANI *et al.*, 2016), contudo, os chamados *smart glasses for visually impaired* ou óculos inteligentes para deficientes visuais ingressaram no mercado com a proposta de facilitar ainda mais o cotidiano dessas pessoas. Através da aplicação de visão computacional e processamento de imagem, os óculos inteligentes possibilitam a identificação de caracteres, objetos e seres. Existem diversos modelos de óculos ou dispositivos semelhantes que visam a acessibilidade para pessoas portadoras de alguma deficiência visual, porém, apesar de não necessariamente apresentarem a funcionalidade de descrição de cenários, a grande maioria se caracteriza no mercado por sua baixa distribuição e preço elevado.

Ao considerar o tamanho populacional de pessoas com visão limitada, verifica-se que o alto valor dos aparelhos presentes no mercado atual não contribui para a disseminação do produto em uma escala significativa. Desta forma, este trabalho objetiva no desenvolvimento de um dispositivo composto por uma câmera, unidade de processamento e saída em som que a partir de tecnologias como visão computacional e síntese de voz, possa apresentar ao usuário um contexto do ambiente em que se encontra visando promover uma maior interação com este ambiente. De forma distinta à trabalhos correlatos que utilizam unicamente da identificação de objetos singulares para o retorno aos usuários com deficiência visual, esse projeto propõem uma abordagem mais contextualizada, onde não apenas há o reconhecimento de objetos, mas também apresenta uma maior descrição em que se encontram na cena. Para isso será utilizado o *image captioning*, uma implementação de *Machine Learning* que se utiliza, por exemplo, de Redes Neurais de Convolução (CNN) para extração de *features* das imagens e LSTM – uma Rede Neural Recorrente, para o processamento de linguagem natural. Ainda no escopo, visa-se que o processo de desenvolvimento, assim como o dispositivo em si, apresente baixos custos de produção, possibilitando assim, maior alcance e disponibilidade da tecnologia.

Para sustentar o desenvolvimento do dispositivo idealizado, uma revisão das publicações mais recentes no tangente a *image captioning*, síntese de voz e visão computacional é realizada. Para a implementação do sistema, softwares de código aberto são utilizados para o processamento das imagens capturadas e transmitidas ao computador de placa única responsável pelo processamento. Adicionalmente, a armação do protótipo foi elaborada a partir de impressão 3D, constituindo um suporte aos óculos inteligentes para assim, integrar todo projeto em um único dispositivo.

Este trabalho é dividido em 5 seções. Na seção 2, discute-se sobre conhecimentos pilares e referenciais deste projeto, sendo constituído pela definição e histórico da tecnologia assistiva,

assim como exemplos voltados para deficientes visuais. Como assunto e tecnologia central do projeto, é dissertado sobre *image captioning* com suas características e aplicações. Em seguida, apresenta-se a metodologia utilizada para desenvolvimento e implantação da solução proposta, assim como explanações pertinentes. Na quarta seção, é apresentado os resultados obtidos e levantadas discussões. Na quinta e última seção, é realizada a conclusão do estudo.

REFERENCIAL TEÓRICO

O referencial teórico deste estudo apresenta 4 tópicos, a iniciar pela revisão de tecnologias assistivas já existentes e a lacuna presente na adoção de dispositivos similares ao proposto neste trabalho. Em seguida são revisadas tecnologias computacionais que podem ser aplicadas nesta temática, tal qual subáreas de estudo de visão computacional, como *image captioning* usadas na geração de descrições de imagens. Finalmente é discutido sobre *text-to-speech* quanto ao seu processo de conversão e geração de falas sintéticas.

Tecnologia assistiva

"Para as pessoas sem deficiência a tecnologia torna as coisas mais fáceis. Para as pessoas com deficiência, a tecnologia torna as coisas possíveis" (RADABAUGH, 1993 apud BERSCH, 2017). Pouco se nota a importância da evolução tecnológica presente na rotina das pessoas. Ferramentas cotidianas como talheres, canetas, computadores, controle remoto, automóveis, telefones celulares e relógio são, em um senso geral, instrumentos criados para facilitar o desempenho em funções pretendidas (BERSCH, 2017). Contudo, uma parcela da sociedade se encontra impedida ou apresenta certas dificuldades para execução de tarefas consideradas triviais para alguém sem dificuldades, dessa forma a implementação da tecnologia se destaca como método essencial para a vida dessas pessoas, tecnologia essa denominada de Tecnologia Assistiva (TA).

De acordo com o Comitê de Ajudas Técnicas (CAT) as tecnologias assistivas se caracterizam por práticas, produtos, recursos e serviços com o objetivo de promover maior autonomia, independência, qualidade de vida e inclusão social para pessoas portadoras de deficiência (CAT, 2007). Portanto, a mesma pode ser compreendida como um auxílio que promoverá a ampliação de uma habilidade funcional deficitária ou possibilitará a realização da função desejada, que se encontra impedida por circunstância de deficiência ou pelo envelhecimento. Pode-se dizer então, que o objetivo maior da tecnologia assistiva é proporcionar à pessoa com deficiência maior independência, qualidade de vida e inclusão social, através da ampliação de sua comunicação, mobilidade, controle de seu ambiente, habilidades de seu aprendizado e trabalho (BERSCH, 2017).

Entretanto, dentre todas as políticas adotadas com objetivo de desenvolver e disseminar as tecnologias assistivas pelo país, o acesso a produtos de alta qualidade tende a ser um problema para as pessoas portadoras de deficiência como exposto pela secretaria de direitos humanos da presidência da república (BRASIL, 2009). Apesar dos incentivos governamentais, o mercado para produtos de tecnologia assistiva são caracterizados por empresas pequenas, com o número de vendas insuficiente para reduzir o custo de produção e abaixar os preços. Como resultado, a maioria dos produtos se tornam extremamente caros, sendo acessíveis somente para as pessoas com condição financeira elevada ou com auxílio governamental, que acaba não suportando o volume excessivo de pessoas, às inserindo em uma fila de espera longínqua (WITTE et al, 2018).

Dentre as categorias abrangentes pela TA, os indivíduos que se enquadram como deficientes visuais dependem majoritariamente das informações táteis e auditivas, configurando assim, a necessidade de criação de uma gama de aparelhos e sistemas voltados para a substituição da função visual. Os recursos auxiliares mais antigos englobam desde métodos de

mobilidade, como a utilização de bengalas e implementação de suportes para navegação local, até a disponibilidade de livros e textos em braile (sistema de escrita tátil). Já em termos computacionais, mediante ao constante avanço tecnológico, existem uma variedade de *softwares* e *hardwares* voltados para usuários com visão baixa ou nula, desde teclados e mouses modificados, até programas de ampliação de telas. Neste contexto, uma área que vem cooperando no desenvolvimento de soluções é a visão computacional, visto aplicações como detecção de objetos e legendagem de cenas (WANG, 2019).

Visão Computacional

A Visão Computacional tem seu início por volta dos anos 60 quando Larry Roberts, em sua tese de Ph.D. no MIT discutiu a possibilidade de extrair informação geométrica em 3D através de uma imagem com perspectiva de blocos em 2D. A partir de então, pesquisadores do mundo todo em Inteligência Artificial continuaram os estudos tangente a Visão Computacional com base em seu trabalho, futuramente avançando para técnicas de detecção de bordas e segmentação (HUANG, 1996).

Portanto, Visão Computacional pode ser compreendida como a ciência responsável pela visão de uma máquina, com a capacidade de a partir de imagens capturadas por câmeras de vídeo, sensores, scanners, e outros dispositivos, extrair informações significativas a ponto de conseguir reconhecer, manipular e pensar sobre os objetos que compõem uma imagem (BALLARD, 1982).

Image Captioning

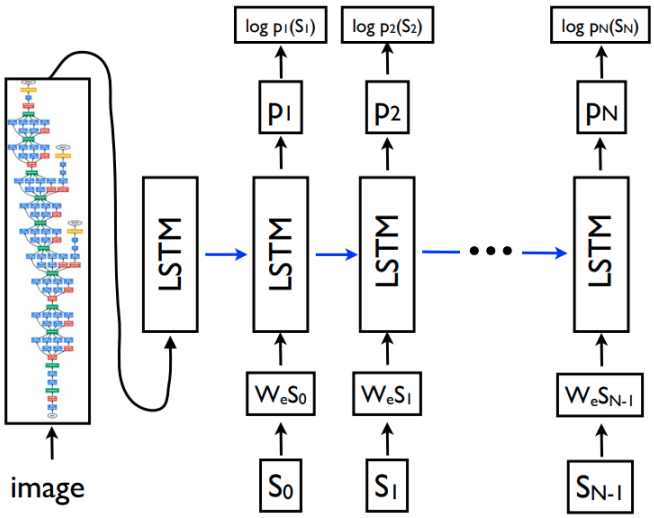
Com o surgimento de problemas mais complexos em análise de dados como no processamento de imagens, se torna inviável uma codificação rígida, onde se é programado visando valores em particular, e se faz necessário a implementação de métodos que possibilitem maior adaptabilidade ao processamento dos dados. Em campos diversos como classificação, reconhecimento de padrões, Internet das Coisas, biologia e aplicações médicas, algoritmos de *Machine Learning* (ML) são utilizados e permitem atingir sucesso em várias frentes de pesquisas (NAQA *et al.*, 2015). Algoritmos de *Machine Learning* visam possibilitar o aprendizado de relações de dados de entrada, e a partir deles formular modelos que são treinados e melhorados a partir da experiência adquirida (ZHOU *et al.*, 2017). Essas características são aplicadas em campos da Inteligência Artificial como Processamento de Linguagem Natural (PLN) e Visão Computacional, e juntas podem ser aplicadas em projetos que auxiliem pessoas com deficiências de visão com apoio de *image captioning* (YOU, 2016).

A geração de descrições de imagens de forma automática a partir de computadores, processo por vezes denominado *image captioning* ou *image annotation*, é uma área de pesquisa relevante com aplicações diversas que visa criar de forma sintética sentenças que retratem o assunto de uma imagem. A partir destas descrições, atividades como indexação de imagens (HOSSAIN *et al.*, 2018) e ajudar pessoas a verem (YOU *et al.*, 2016) podem ser executadas de uma melhor forma. Para a sua implementação, atualmente são consideradas como estado da arte o uso conjunto de Redes Neurais de Convolução (CNN na sigla em inglês) para a extração de *features* das imagens, tais como os objetos que possam compô-la, e *Long Short-Term Memory* (LSTM) pertencente ao grupo de Redes Neurais Recorrentes que possibilita propriamente a geração de sentenças (RAMPAL e MOHANTY, 2020).

Na figura 1, é apresentado o fluxo de operações para a geração das descrições. De início, a imagem capturada é ingerida em uma CNN pré-treinada que realiza a extração de *features* presentes na imagem. Com a compilação destas *features*, um processo de LSTM é iniciado para realizar a predição de palavras que possam compor a descrição (VINYALS *et al.*, 2015). Para

cada iteração, são escolhidas novas palavras a partir das *features* da imagem além das palavras já definidas em execuções anteriores.

Figura 1: Fluxo de processamento de image captioning usando CNN e LSTM.(Fonte: VINYALS *et al.* (2015)

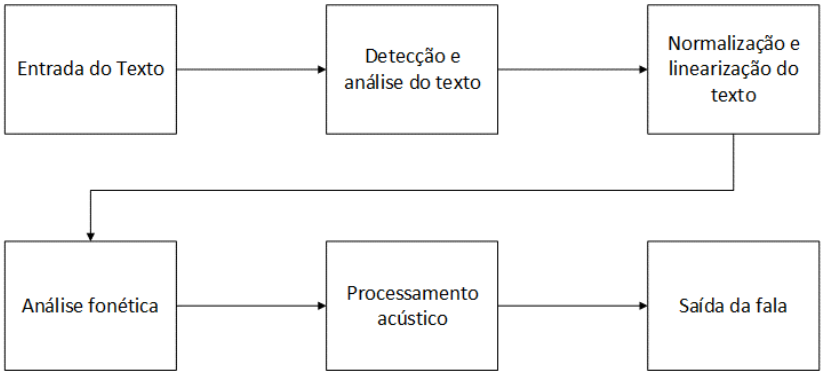


Text-to-Speech

A conversão de texto para fala, ou *text-to-speech* (TTS) é uma ferramenta amplamente utilizada que se destaca como essencial para os usuários portadores de deficiência visual, uma vez que amplia o entendimento dessas pessoas em diversas mídias (MOTTA et. al, 2010), dessa forma, não limitando seu uso apenas para utilização em computadores e celulares, mas podendo ser aplicados em conjunto de diversas tecnologias, tais como *image captioning* em dispositivos para acessibilidade. O *text-to-speech* é um sistema composto por um sintetizador de voz capaz de ler textos e caracteres para o usuário. Um sintetizador pode incorporar o modelo vocal assim como características da voz humana para assim, gerar um *output* em forma de áudio replicando a fala de uma pessoa (MITHE et al, 2013).

A primeira etapa no processo de TTS é a análise do texto, onde o sistema irá organizar os caracteres em uma lista de palavras a serem transformadas no texto por completo. Logo depois, será feito o processo de normalização e linearização, onde o texto será traduzido para sua forma pronunciável, com a identificação de pausas e pontuação entre as palavras. Depois é feito a análise fonética e o processamento acústico, onde a síntese de voz lerá o texto convertendo os símbolos ortográficos em símbolos fonéticos (MITHE et al, 2013). A figura 2 representa a conversão TTS básica.

Figura 2: Etapas da conversão TTS. (Fonte: Pelos Autores)



METODOLOGIA

Com o intuito de possibilitar o desenvolvimento do sistema proposto, inicialmente foram realizadas pesquisas sobre tecnologias assistivas e outras que poderiam ser utilizadas para esse fim, relevantes à aplicação que objetiva auxiliar indivíduos com deficiências de visão, sendo parte destas compiladas na seção anterior. A partir dos estudos realizados, uma série de tarefas em âmbitos de desenvolvimento e implementação de *software* e *hardware* são executadas visando disponibilizar ao seu usuário um dispositivo que fornece a descrição do local em que se encontra com uso de tecnologias como *image captioning* e *text-to-speech*, sendo estas executadas em um computador de placa única munido de uma câmera.

Hardware

Visando um protótipo capaz de executar o *software* idealizado, utiliza-se um computador de placa única (*Single Board Computer* - SBC) de baixo custo Raspberry Pi 4 Model B, que é ilustrado na figura 3. Este dispositivo é escolhido por apresentar pequenas dimensões, peso e consumo elétrico (RADZI, 2020) enquanto possui suporte a recursos de I/O relevantes como interfaces de câmera e entradas e saídas de som responsáveis por possibilitar a interação com o usuário.

Figura 3: SBC Raspberry Pi 4 Model B. (Fonte: Raspberry Pi ORG homepage)

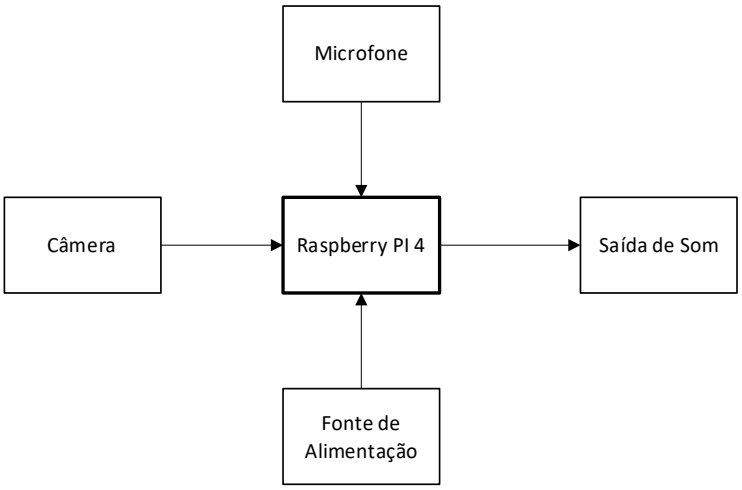
Compondo-se de um processador Broadcom BCM2711 conforme exposto na tabela 1 juntamente com outras especificações de *hardware*, utiliza-se este dispositivo para processamento das imagens capturadas a partir de uma câmera CSI atachada à placa. Para fins de interface entre o usuário e o sistema, um microfone USB e um fone de ouvido P2 sem cancelamento de ruídos como saída de som são incorporados na SBC. A figura 4 representa o diagrama do sistema proposto, contendo uma câmera, microfone e fonte de alimentação como itens de entrada no Raspberry Pi e a saída de som como elemento de saída.

Tabela 1: Especificações da Raspberry Pi 4 utilizada.

Unidade	Componente
Processador	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memória	8GB
Conexões e Portas de I/O	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless 2 USB 3.0 ports; 2 USB 2.0 ports. Raspberry Pi standard 40 pin GPIO header 2-lane MIPI DSI display port 2-lane MIPI CSI camera port

Fonte: Raspberry Pi Homepage

Figura 4: Diagrama do protótipo proposto. (Fonte: Pelos Autores)



Arquitetura do software

Para a execução do *software* projetado, o Raspberry Pi executa o Linux Raspberry Pi OS/Raspbian por ser uma opção gratuita, compatível com a arquitetura ARM utilizada do SBC e *open-source*, possibilitando uma plataforma menos custosa (MAKSIMOVIĆ, 2014). No tangente à escolha da linguagem de programação para desenvolvimento da aplicação, é definido o uso da linguagem Python, que de acordo com Raschka *et al* (2020), se caracteriza como a linguagem preferida para *Data Science* e *Machine Learning*, uma vez que ocorre um aumento de performance e produtividade ao se utilizar das bibliotecas *low-level* e APIs *high-level*. O *software* consiste em bibliotecas e códigos de visão computacional otimizados que possibilitam a plena interação com o usuário para a utilização do sistema de *image captioning* dos óculos.

Métodos de interface

Os meios de interface com o usuário definidos neste desenvolvimento são todos a partir de som, assim requerendo um reconhecimento de voz para identificação dos comandos solicitados pelo usuário e uma conversão *Text to Speech* para informar o usuário sem a necessidade de um *display* ou *feedback* visual, quanto aos resultados das solicitações realizadas. Para a primeira etapa foi utilizada a biblioteca Python SpeechRecognition, sob a Licença BSD, devido a sua facilidade de uso conforme apontado por AMOS *et al* (2018), com suporte a *engines* e APIs de funcionamento *offline*. Para o segundo item, o sintetizador de voz escolhido foi o Google *Text-to-Speech*, ou gTTS, sob a licença MIT que de acordo com KURLEKAR *et al* (2020) possibilita a realização de customizações em relação a fala enquanto mantem a entonação, leitura de abreviaturas entre outras características.

Modelo de Image Captioning

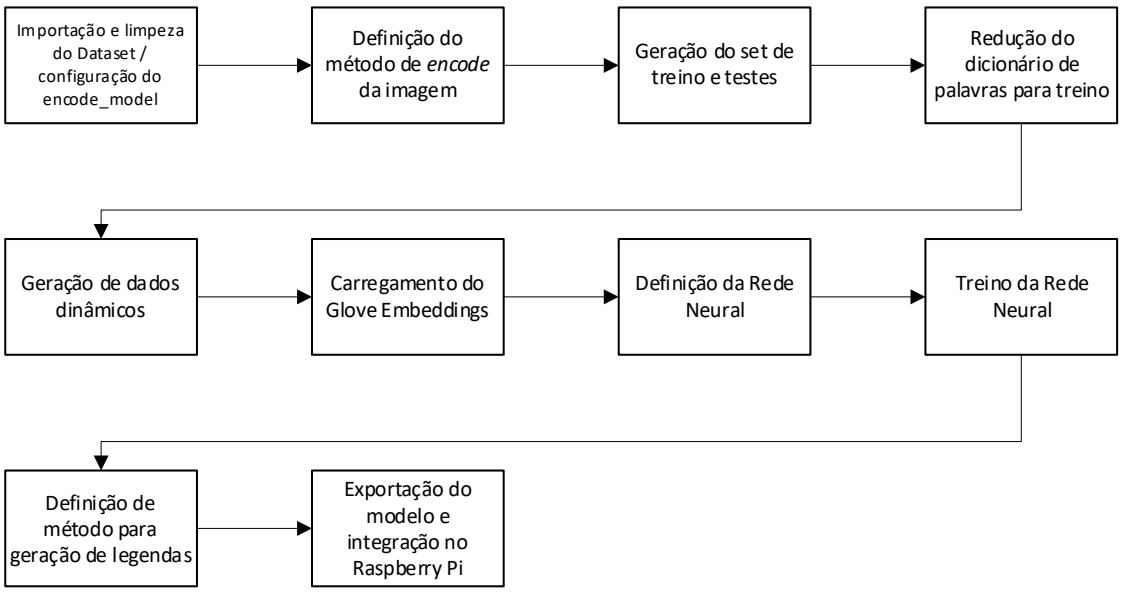
Com o módulo de descrição de imagens, o usuário recebe em detalhes uma transcrição referente aos cenários, objetos e ações presentes na imagem capturada, dessa forma, a pessoa com deficiência visual terá a capacidade de compreender as informações do local sem a necessidade da visão. O modelo é dividido em dois componentes de Redes Neurais que agem em sequência, a iniciar com o extrator de *features* das imagens a partir de uma CNN e posteriormente a geração das descrições usando as *features* adquiridas na etapa anterior com uma RNN. Para a elaboração do código foram implementados dois projetos por meio de *transfer learning,* o *InceptionV3* e o *Glove.* *InceptionV3* consiste em uma rede neural pré-

treinada com a finalidade de extrair características de imagens, visando acelerar e aprimorar a geração do modelo, é escolhido devido a sua ênfase na importância na gestão de recursos de memória e processamento, sendo adequado em aplicações em dispositivos móveis (RIDELL, 2017), enquanto o *Glove* se classifica como um *embedding* de processamento de linguagem natural, composto por conjuntos de vetores originário de palavras relacionadas do vocabulário.

No que se refere à primeira etapa, são implementadas funções de *encode* para processamento das imagens com o modelo de CNN Inception-v3 que em trabalhos relacionados é apresentado como um melhor modelo em comparações recentes (SESHADRI, 2020), em relação ao segundo estágio, é implementado uma LSTM em uma função de *decode* visando desenvolver um processamento sequencial inerente a escolha de palavras mais adequadas que melhor fazem descrição a imagem processada anteriormente. Adicionalmente, são preparados métodos auxiliares para a conexão entre as duas redes neurais, importação das imagens para treinamentos e validações. No tangente a estes dois últimos tópicos, utiliza-se o *dataset* Flickr30k composto de 31783 imagens com 5 descrições cada para o treinamento supervisionado da rede neural que quando concluída, é avaliada pela pontuação BLEU do mesmo.

Desta forma, para a implementação dos algoritmos de *machine learning* necessários para o *image captioning*, neste estudo é utilizado a biblioteca Tensorflow desenvolvida pelo Google e disponibilizada a partir da licença Apache License 2.0 em conjunto com o Keras, que de acordo com SHAH *et al* (2017), proporciona uma "plataforma heterogênea de execução, ou seja, pode ser executado em dispositivos com menor poder de processamento como dispositivos móveis", o que se configura adequado ao protótipo proposto. No que se refere as etapas de processamento e geração do modelo de *image captioning*, neste estudo, segue-se as etapas ilustradas na figura 5 nas quais partem em conformidade com Heaton (2020).

Figura 5: Fluxo de processos para image captioning. (Fonte: Pelos Autores)



A primeira etapa para implementação do código de *Image Captioning* é a importação e limpeza do *dataset*. Depois de importado é necessário um tratamento dos dados das descrições, a fim de eliminar espaços em branco, pontuações e palavras com componentes não alfabéticos. Na segunda etapa é estabelecido a configuração do modelo de *encode* utilizando o *InceptionV3*. O modelo de *encode* tem como finalidade converter as imagens para que possam ser processadas futuramente. Desse modo, deve ocorrer o redimensionamento das imagens para um tamanho padronizado, logo após, as imagens devem ser convertidas para arrays. Por fim, o

modelo de CNN *InceptionV3* é invocado para realizar as extrações das características e a imagem é remodelada para ser aceita pelo modelo de *captioning* LSTM. A terceira etapa consiste basicamente na separação das imagens em conjuntos de treinamento e de testes. Os dois processos necessitam percorrer por todos os arquivos de imagem JPG do conjunto de dados, por isso, estes são salvos em um arquivo pickle, com intuito de serializar os arquivos para facilitar processos futuros com as imagens, em seguida também é separado as descrições das imagens para o treinamento. O quarto processo objetiva a redução do dicionário de palavras para o conjunto de treinamento, uma vez que palavras com baixa frequência podem prejudicar o treinamento da rede neural. A etapa de geração de dados dinâmicos é realizada com o propósito de gerar os dados do conjunto de treinamento conforme a rede neural necessita, dessa forma há um sistema mais eficiente no quesito de gerenciamento de memória, uma vez que existe uma grande quantidade de imagens contendo cinco descrições cada.

A sexta etapa, carregamento do *Glove embeddings*, visa por realizar a leitura e extração dos vetores de palavras do *Glove* para uma matriz que é utilizada em etapas seguintes tangentes a construção da rede neural para geração das sentenças com o objetivo de aprimorar os resultados utilizando os princípios de transferência de aprendizado. Em seguida, para a definição da rede neural é estabelecido dois parâmetros de entrada, as imagens e as descrições. Somado a isso, a matriz contendo as palavras do *Glove* é utilizada para a definição das entradas de todas as palavras das descrições, gerando uma nova matriz com o tamanho do vocabulário e o número de *features* geradas pelo *Glove* para cada palavra, tal matriz é utilizada como parâmetro de aprendizado para Rede Neural. Por fim o modelo criado e compilado. A oitava etapa remete ao treinamento da Rede Neural. Primeiramente é definido a quantidade de imagens para cada iteração do treinamento, denominada *epochs*. Cada *epoch* tem como objetivo o processamento das imagens e descrições para o treinamento da rede, desse modo, a cada iteração ocorre a diminuição da *loss function*, indicando maior acuraria em relação aos parâmetros de aprendizado. Após o treinamento da rede neural, a definição do método de geração das legendas é executada visando obter uma sentença a partir de uma imagem recebida como parâmetro. Para isso, o método requisitara predição da imagem para o modelo treinado, onde as palavras com maior probabilidade serão adicionadas em uma cadeia de string, tal processo se repetirá até a rede neural predizer o token final ou se a descrição atingir o tamanho máximo.

Dessa forma, com os modelos e métodos uma vez já executados será feito uma portabilidade do código para um sistema a ser processado no computador de placa única Raspberry PI 4. Esse sistema recebera as imagens a partir de fotos capturadas pela câmera, no quais estas serão submetidas ao método de geração de legendas, retornando a descrição para o usuário em forma de áudio.

RESULTADOS E DISCUSSÃO

Nesta seção são abordados os proventos desta pesquisa iniciando-se quanto as implementações e desenvolvimento de *software* visando o sistema composto por tecnologias como *text-to-speech* e reconhecimento de voz para interface com o usuário e *image captioning* como conjunto de algoritmos para a geração de descrições oportunas para seu usuário final. Posteriormente apresenta-se a modelagem e implementação do protótipo físico, encerrando com uma discussão quanto aos resultados obtidos.

Interface com usuário

Ao ligar os óculos, o dispositivo comunica ao usuário a inicialização do sistema através de uma mensagem de boas-vindas. Para isso, foi implementado uma função *text-to-speech*, sendo esta utilizada para o retorno em som de todo sistema. A figura 6 apresenta a

implementação do gTTS, o sintetizador de voz escolhido. A função joroglass_speak definida na linha 1 tem como objetivo transformar uma cadeia de texto em um arquivo mp3, para logo em seguida ser reproduzido a partir do omxplayer, reprodutor de áudio nativo do RaspberryPi OS. Na linha 2, é definido a geração da fala onde se configura o texto recebido pelo método que será convertido e em qual idioma este texto será falado. Na terceira linha é realizada a exportação da fala para um arquivo mp3 temporário enquanto na linha 4 é definido o comando de execução deste arquivo com o omxplayer. Na quinta linha é realizada a execução do comando criado na linha anterior e após a reprodução do áudio, na linha 6, o arquivo temporário é excluído.

Figura 6: Método para reprodução do som gerado pelo TTS. (Fonte: Pelos Autores)

```
1       def joroglass_speak(audio_string):
2           tts = gTTS(text = audio_string, lang = 'en')
3           tts.save('audio.mp3’)
4           audio_play_cmd = 'omxplayer audio.mp3'
5           os.system(audio_play_cmd)
6           os.remove(‘audio.mp3’)
```

Uma vez definido o TTS, é necessário a identificação dos comandos por parte do usuário. Na figura 7 é exposto o método desenvolvido para o reconhecimento de voz do sistema. Para isso, é realizado a importação de speech_recognition e instanciado um Recognizer nas linhas 1 e 2, respectivamente. Dentro do método record_audio definido a partir da linha 4, é iniciado a gravação do áudio e atribuído em uma variável chamada “audio", a partir do método listen na linha 6 utilizando as configurações padrões de sr.Microphone a partir de source declarado no *with* na linha 5. Na linha 7 é inicializada uma cadeia de texto vazia em uma variável chamada voice_data na qual armazenará o texto reconhecido. Uma vez que adversidades podem ocorrer no processo de requisição à API, um escopo try é posto na linha 8 e em seu escopo na linha 9 está posto o método que consome o áudio anteriormente salvo em “audio" e retorna sob voice_data o texto esperado.

Figura 7: Método de gravação de som que realiza o reconhecimento das palavras ditas. (Fonte: Pelos Autores)

```
 1   import speech_recognition as sr
 2   r = sr.Recognizer()
 3
 4   def record_audio():
 5       with sr.Microphone() as source:
 6           audio = r.listen(source)
 7           voice_data = ""
 8           try:
 9               voice_data = r.recognize_google(audio)
10           except sr.UnknownValueError:
11               joroglass_speak("Sorry, I did not get that")
12           return voice_data
```

Módulos do sistema

A figura 8 remete ao método de controle de funcionamento principal dos óculos. Com a inicialização do sistema o dispositivo saudará o usuário via síntese de voz, assim como

exposto na linha 18, e logo após entrará na função menu, linha 1, aguardando a seleção do módulo desejado. Para cada interação por parte do usuário, no estágio final de cada execução do seu comando é utilizado o método joroglass_speak na qual possibilita a geração da fala necessária para a comunicação. Nessa seção foram definidas as seguintes opções: na linha 2, o usuário poderá perguntar o nome dos óculos, também o sistema retornara as horas, utilizando-se da função ctime() se requisitada assim como na linha 5, a terceira possibilidade é o módulo de *image captioning*, que quando perguntado sobre o que está em sua frente, linha 8, o sistema retornará a confirmação de entrada do módulo, linha 9, logo em seguida inicializará a captura por meio da câmera, para depois ser gerado a descrição da imagem capturada, linha 11 e retornada ao usuário em forma de áudio na linha 13.

Figura 8: Método de controle principal. (Fonte: Pelos Autores)

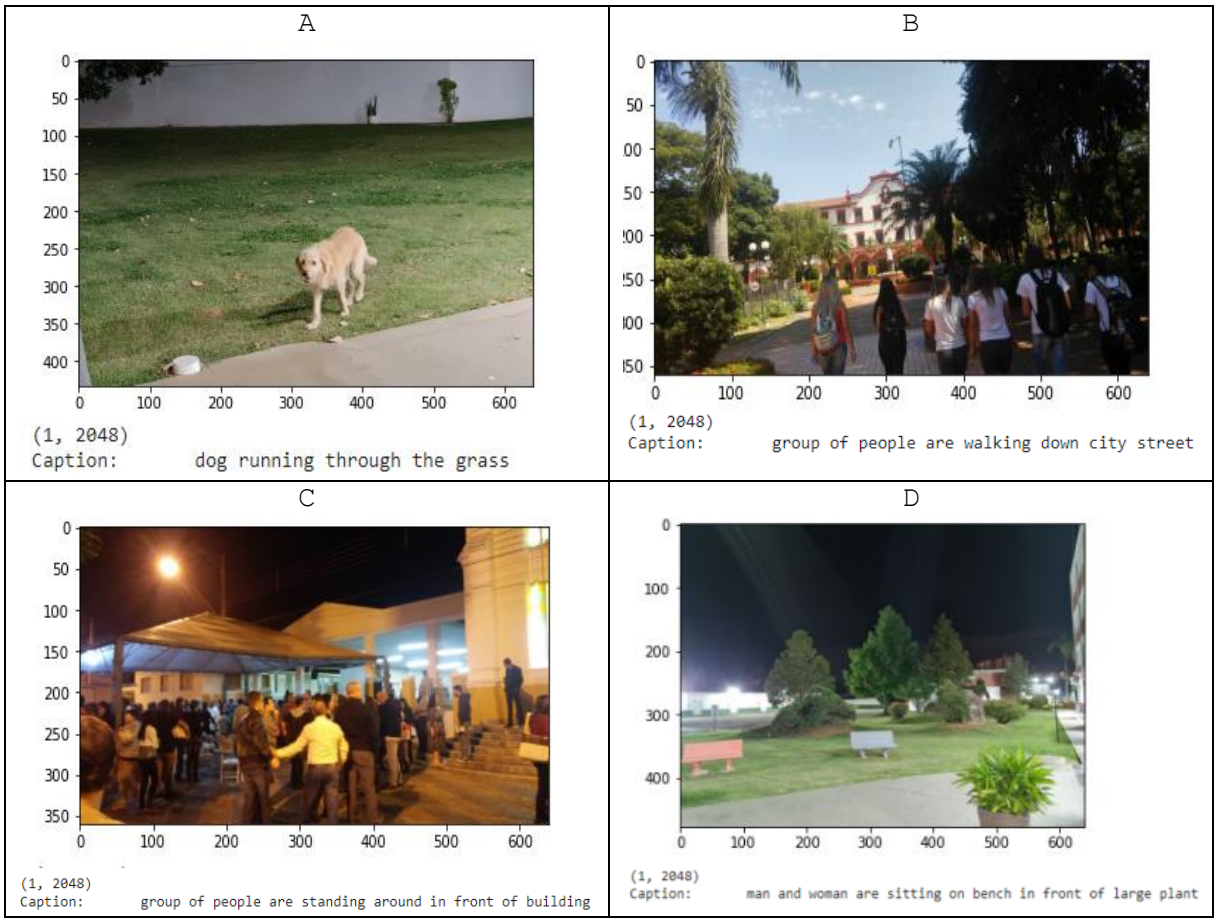
```
1  def menu(voice_data):
2      if "what is your name" in voice_data:
3          joroglass_speak("My name is joroglass")
4
5      if "what time is it" in voice_data:
6          joroglass_speak(ctime())
7
8      if "what is in front of me" in voice_data:
9          joroglass_speak("opening image caption module")
10         camera.capture('./image.jpg')
11         caption = imageCapture()
12         joroglass_speak("Done. I found the following:")
13         joroglass_speak(caption)
14
15     if "exit" in voice_data:
16         exit()
17
18 joroglass_speak("Welcome to JoroGlass, how can I help you?")
19 while 1:
20     voice_data = record_audio()
21     menu(voice_data)
```

Image Captioning

Com o treinamento do modelo de *image captioning* no Keras e Tensorflow seguindo o método apresentado anteriormente, foram obtidos resultados relevantes para o protótipo tendo em vista as dimensões do banco de imagens e legendas. O modelo treinado foi capaz de realizar o reconhecimento de cenas gerais com legendas adequadas em situações similares aquelas encontradas no *dataset* e apresentando pequenos erros ou minimamente contendo artefatos relacionados a imagem consumida no caso de situações muito divergentes, como esperado. A figura 9 expõe uma grade onde em cada quadrante há uma amostra de imagens capturadas pelos autores acompanhadas da legenda extraída pelo modelo de *image captioning*, na figura é possível avaliar que o resultado do quadrante “A” está condizente com o que é exposto. Em geral, foram obtidos resultados positivos na detecção de animais, tais como cães e gatos, e atividades com pessoas, como postas nas imagens dos quadrantes “B” e “C”. No caso destas duas últimas, nota-se que houve o reconhecimento do grupo de pessoas, além do apontamento

de suas ações e estados como andar ou ficar em pé. Entretanto, foi encontrado situações em que as legendas geradas não obtiveram um nível de acurácia muito elevado como pode-se notar na figura do quadrante “D” como exemplo, onde foi detectado arbustos e assentos de bancos, mas também é indicado erroneamente a presença de um homem e uma mulher na cena.

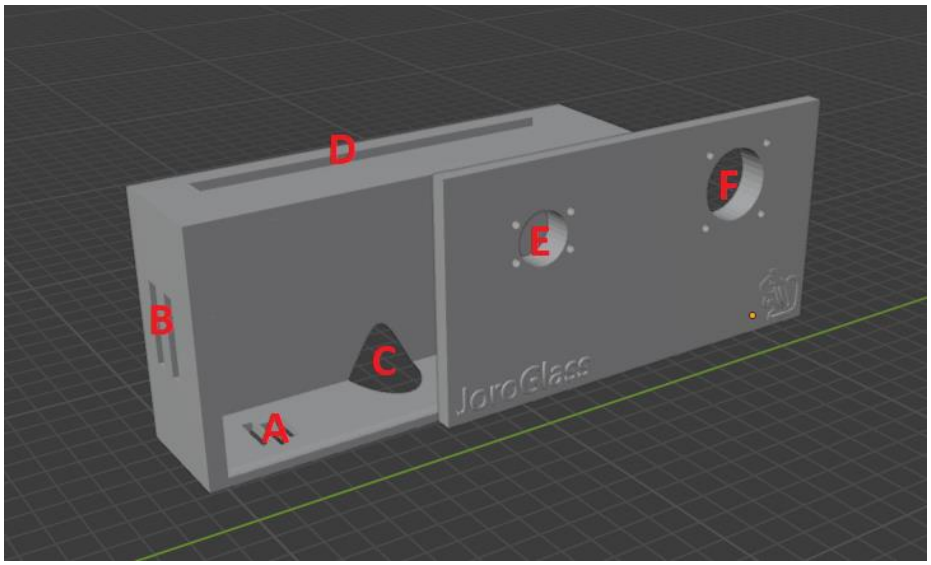
Figura 9: Mosaico com figuras e suas respectivas legendas geradas pelo modelo de image captioning. (Fonte: Pelos Autores)



Modelagem dos Óculos

Para sustentar os componentes de *hardware* mencionados em seções anteriores, foi desenvolvido o modelo do protótipo proposto. Para a modelagem utilizou-se do software gratuito 3D Blender, distribuído pela licença GNU GPLv2+. A figura 10 representa a modelagem desenvolvida, assim como a função de cada elemento destacado. A abertura “A” representa a saída de ar e passagem de som a ser captada pelo microfone, “B” são as presilhas a serem acopladas a corda de fixação, “C” é a abertura e acomodação do nariz do usuário, “D” permite a passagem de entradas e saídas da Raspberry, “E” representa a abertura para a câmera e “F” para o *cooler* de refrigeração do sistema.

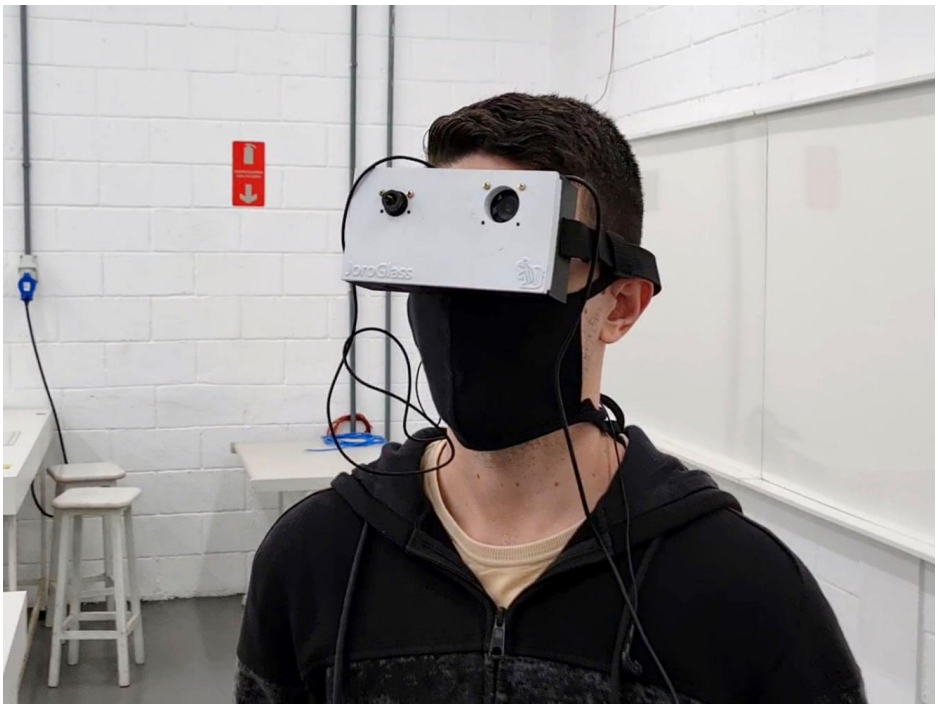
Figura 10: Modelagem do protótipo proposto. (Fonte: Pelos Autores)



A partir da modelagem, o protótipo foi constituído por meio de impressões em 3D. Possuindo dimensões de 15cm x 8cm x 4cm, os óculos, exposto em sua versão final pela figura 11 consegue condensar em um único sistema todos os componentes definidos. Já na figura 12 é visto sua utilização, onde as alças são ajustáveis para acomodar cada usuário, nota-se também o uso do fone em apenas um ouvido, para que o usuário não se abstenha dos sons exteriores. Fones de condução óssea também poderiam ser utilizados, para assim minimizar totalmente a ofuscação de sons do ambiente.

Figura 11: Imagem do protótipo com ênfase no componente da câmera. (Fonte: Pelos Autores)

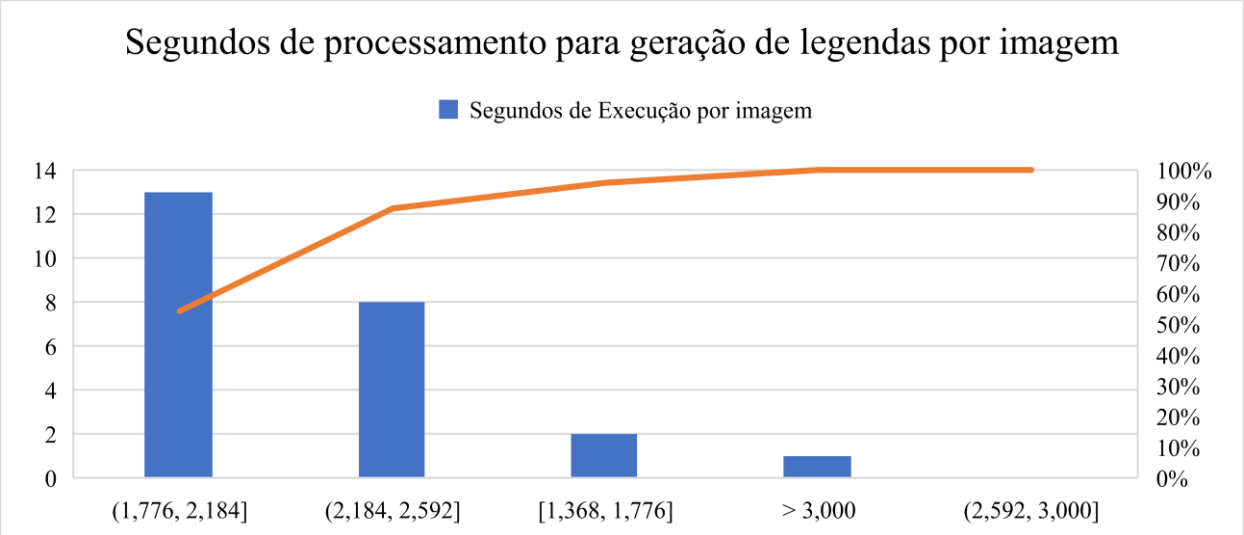
Figura 12: Pessoa utilizando o protótipo desenvolvido. (Fonte: Pelos Autores)



Desempenho do sistema

Um estudo visando medir a responsividade do protótipo foi realizado e o tempo de processamento para geração de legendas das fotos capturadas, em segundos, são representados pela figura 13. Devido a inicialização do modulo de *image captioning* a primeira execução levara cerca de 7 a 9 segundos para ser processada, retornando a descrição gerada para a imagem, contudo, para os próximos processamentos, acompanhando o histórico observado no gráfico, são esperadas que cerca de 95% das respostas ocorram entre 1 a 2,5 segundos, configurando assim, um sistema relativamente responsivo referente ao *feedback* para o usuário.

Figura 13: Gráfico de Pareto compilando tempos de processamento de geração de legenda. (Fonte: Pelos Autores)



No tangente ao desempenho da Raspberry Pi 4, a figura 14 e figura 15 relatam algumas métricas obtidas a partir do PSensor, uma aplicação de código aberto sob a licença GPLv2,

referentes a 20 minutos de execução dos óculos. Em termos de uso de memória, durante a execução do software foi encontrado uma média em torno de 30% da memória disponível com um consumo aproximado de 2,5GB. No que se refere à carga sob o processador do Raspberry Pi 4, em seus picos de processamento foram averiguados valores em torno de 66% da capacidade total do dispositivo. Já em questão da temperatura, a figura 14 representa o sistema em funcionamento sem nenhum sistema de resfriamento, já a figura 15 se refere ao uso com a utilização do cooler, notando-se uma diminuição de 20 graus célsius. Desta forma, pode-se afirmar que o *hardware* definido atendeu as expectativas no que se refere ao desempenho e funcionalidade, enquanto devido ao seu relativo baixo custo por unidade – por volta de R$ 1000,00 que compreende os custos com o SBC, microfone e fone de ouvido para interface com o usuário, câmera e insumos para impressão da armação do protótipo – cooperou nas questões orçamentárias, possibilitando um projeto sustentável e com facilidades para a reprodução da pesquisa.

Figura 14: Monitoramento de temperatura do sistema sem cooler (Fonte: Pelos Autores)

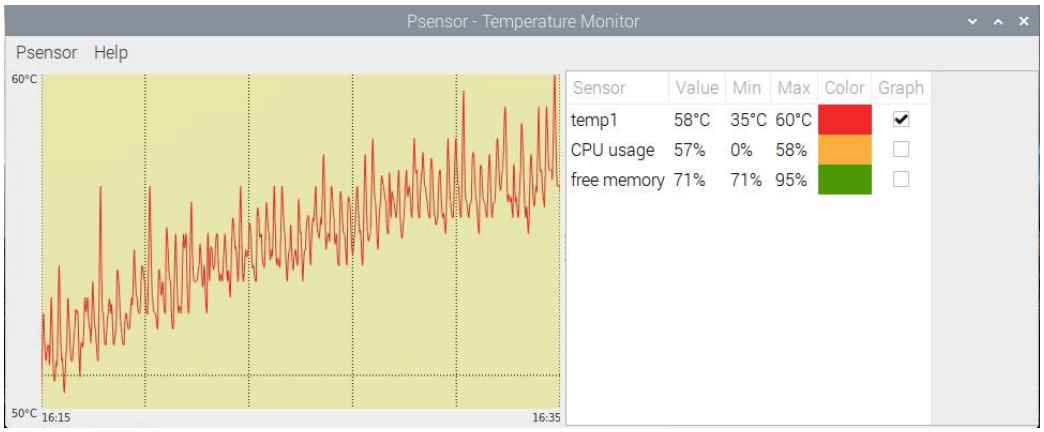
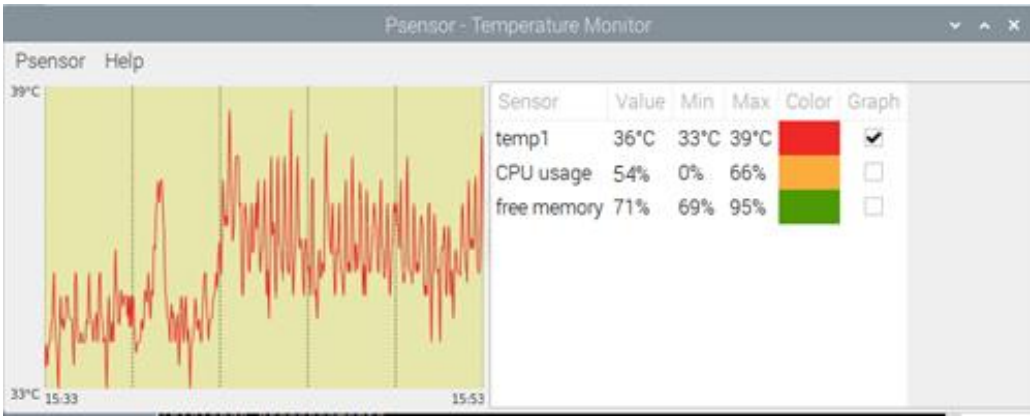


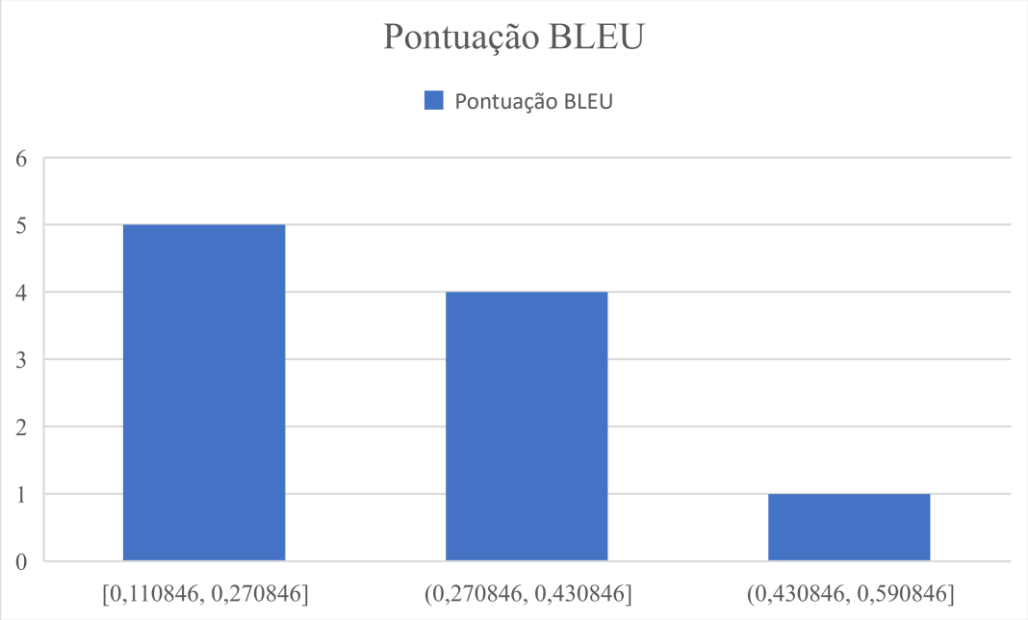
Figura 15: Monitoramento de temperatura do sistema com cooler (Fonte: Pelos Autores)



Como forma de mensurar a qualidade do modelo de *image captioning* treinado, a partir de uma amostra aleatória de validação composta de 10 imagens foram geradas pontuações BLEU a partir das legendas geradas pelo modelo e compiladas no histograma da figura 16. Verifica-se com os dados que vão de 0 a 1, sendo 1 uma legenda idêntica àquela apontada por um humano e zero uma legenda sem nenhuma similaridade, que o modelo gerado obteve pontuações regulares com legendas que conseguiram apontar o reconhecimento de objetos e determinadas situações, mas que apresentaram artefatos ou não necessariamente descreveu a cena no mesmo plano. Como apontado em seções anteriores, para este protótipo foi selecionado como *dataset* o Flickr30k. Tal *dataset* foi submetido a um treinamento com 20 *epochs* em uma

máquina contendo uma placa de vídeo Nvidia Geforce GTX 1650, com uma duração de cerca de 28 horas para finalização do treinamento, acredita-se que para a finalidade desse projeto o resultado do modelo se mostra suficiente, porém é notável que o modelo treinado possui margens para melhora, sendo que este ponto poderia ser aprimorado ao se utilizar de *datasets* maiores para o treinamento, assim como o *hypertuning* para determinar os melhores parâmetros a serem implementados para a rede neural.

Figura 16: Resultados da pontuação BLEU em uma amostra de legendas geradas pelo modelo de *image captioning* treinado. (Fonte: Pelos Autores)



CONCLUSÕES

No presente artigo, foi apresentado como o *image captioning* em conjunto com outras tecnologias, possibilitam o desenvolvimento de aplicações tangentes a acessibilidade podendo ser implementadas em um dispositivo inteligente de baixo custo para auxílio de deficientes visuais. Diante dos materiais utilizados e sistemas desenvolvidos, constatou-se a total capacidade da construção de um protótipo com habilidade de transcrição de cenários e situações para a forma de áudio, possibilitando assim, que usuários com baixa visão possam identificar o que está presente em sua frente.

O sistema demonstrou bom desempenho ao ser executado em um computador de placa única Raspberry Pi 4, apresentando boa disponibilidade e velocidade de processamento. Com isso, o dispositivo permite total mobilidade ao usuário, configurando uma locomoção independente de uma conexão externa a um servidor, uma vez que todo processamento é realizado dentro dos óculos.

No que se refere as funções implementadas, que consagram o adjetivo inteligente ao dispositivo, existe ainda a possibilidade de expansão de funcionalidades, porém, o foco se deu no módulo de descrições de imagens, o *image captioning*, que apresentou resultados satisfatórios, contudo, o modelo utilizado para geração das legendas se beneficiaria de aprimoramentos a fim de elevar sua performance, como a utilização de *datasets* maiores para a fase de treinamento, o que acarretaria maior tempo e poder de processamento.

No tangente a acessibilidade em dispositivos denominados *wearables*, apesar de estarem presentes no mercado, poucos conseguem atender a demanda necessária, pois apresentam

preços elevados e baixa distribuição, somado a isso, a presença de óculos inteligentes para deficientes visuais, no que toca a transcrição visual, geralmente se limitam a funções de leitura de texto, utilizando-se do OCR, e a identificação de objetos singulares. O protótipo realizado, porém, argumenta sobre a implementação de uma tecnologia ainda pouco disseminada, no que se referente a descrição de fotos retiradas pelo usuário, contextualizando os cenários e situações em que este se encontra.

Com isso, utilizando-se de bibliotecas de código aberto para a elaboração do *software* e *hardwares* de baixo custo somado a impressões 3D para construção do protótipo, o sistema apresenta baixo custo de desenvolvimento. Dessa forma, a realização do sistema proposto remonta o potencial do protótipo elaborado, configurando assim, um dispositivo de *image captioning* funcional e acessível para pessoas com algum tipo de deficiência visual, permitindo a elas maior nível de independência e autonomia em suas vidas.

REFERÊNCIAS BIBLIOGRÁFICAS

ADVANI, Siddharth et al. **A multitask grocery assist system for the visually impaired: Smart glasses, gloves, and shopping carts provide auditory and tactile feedback**. IEEE Consumer Electronics Magazine, v. 6, n. 1, p. 73-81, 2016.

AMOS, David et al. **The Ultimate Guide To Speech Recognition With Python**. Real Python, 2018.

BALLARD, Dana H.; BROWN, Christopher M. **Computer Vision**. Englewood Cliffs, N.J: Prentice Hall, 1982

BERSCH, Rita. **Introdução à tecnologia assistiva**. Porto Alegre: CEDI, v. 21, 2008.

BRASIL. Presidência da República, Secretaria Especial dos Direitos Humanos, Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência**, Tecnologia assistiva**. 2009.

CAT, 2007. **Ata da Reunião VII**, abril de 2007, Comitê de Ajudas Técnicas, Secretaria Especial dos Direitos Humanos da Presidência da República (CORDE/SEDH/PR).

HEATON, Jeff. **Applications of deep neural networks**. arXiv preprint arXiv:2009.05673, 2020.

HOSSAIN, MD Zakir et al. **A comprehensive survey of deep learning for image captioning**. ACM Computing Surveys (CsUR), v. 51, n. 6, p. 1-36, 2019.

HUANG, T. **Computer Vision: Evolution and Promise**. Geneva: CERN. pp. 21–25, 1996

KURLEKAR, Supriya et al. Reading Device for Blind People using Python OCR and GTTS. **International journal of Science and Engineering Applications**, v. 9, n. 4, p. 049-052, 2020.

MAKSIMOVIĆ, Mirjana et al. **Raspberry Pi as Internet of things hardware: performances and constraints**. **design issues**, v. 3, n. 8, p. 1-6, 2014.

MITHE, Ravina; INDALKAR, Supriya; DIVEKAR, Nilam. **Optical character recognition**. International journal of recent technology and engineering (IJRTE), v. 2, n. 1, p. 72-75, 2013.

MOTTA, L.M.V. A Audiodescrição vai à Ópera. In MOTTA, L.M.V. e ROMEU FILHO, P. (orgs): **Audiodescrição: Transformando Imagens em Palavras**. Secretaria dos Direitos da Pessoa com Deficiência do Estado de São Paulo, 2010.

NAQA, Issam; MURPHY, Martin J. **What is machine learning?** In: machine learning in radiation oncology. Springer, Cham, 2015. p. 3-11.

PASCOLINI, Donatella; MARIOTTI, Silvio Paolo. **Global estimates of visual impairment: 2010**. British Journal of Ophthalmology, v. 96, n. 5, p. 614-618, 2012.

RADZI, Syafeeza Ahmad et al. **IoT based facial recognition door access control home security system using raspberry pi**. International Journal of Power Electronics and Drive Systems, v. 11, n. 1, p. 417, 2020.

RAMPAL, Harshit; MOHANTY, Aman. **Efficient CNN-LSTM based Image Captioning using Neural Network Compression**. arXiv preprint arXiv:2012.09708, 2020.

RASCHKA, Sebastian; PATTERSON, Joshua; NOLET, Corey. **Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence**. Information, v. 11, n. 4, p. 193, 2020.

Raspberry Pi Foundation. **Raspberry pi 4**: model b. Disponível em: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. Acesso em: 21 de agosto de 2021.

RIDELL, Patric; SPETT, Henning. **Training set size for skin cancer classification using Google's inception v3**. 2017.

SESHADRI, Madhavan; SRIKANTH, Malavika; BELOV, Mikhail. **Image to Language Understanding: Captioning approach**. arXiv preprint arXiv:2002.09536, 2020.

SHAH, Parth; BAKROLA, Vishvajit; PATI, Supriya. **Image captioning using deep neural architectures**. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). IEEE, 2017. p. 1-4.

VINYALS, Oriol et al. **Show and tell: A neural image caption generator**. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 3156-3164

WANG, Linda; WONG, Alexander. **Implications of computer vision driven assistive technologies towards individuals with visual impairment**. arXiv preprint arXiv:1905.07844, 2019.

WITTE, Luc et al. **Assistive technology provision: towards an international framework for assuring availability and accessibility of affordable high-quality assistive technology**. Disability and Rehabilitation: Assistive Technology, v. 13, n. 5, p. 467-472, 2018.

YOU, Quanzeng et al. **Image captioning with semantic attention**. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 4651-4659.

ZHOU, Lina et al. **Machine learning on big data: Opportunities and challenges**. Neurocomputing, v. 237, p. 350-361, 2017.