

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
MINAS GERAIS - *CAMPUS* ITABIRITO
ENGENHARIA ELÉTRICA

Filipe Vieira Brison

**PROTÓTIPO *HUET*: AUTOMAÇÃO RESIDENCIAL UTILIZANDO
LIBRAS**

Itabirito - MG
2023

FILIPPE VIEIRA BRISON

**PROTÓTIPO *HUET*: AUTOMAÇÃO RESIDENCIAL UTILIZANDO
LIBRAS**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus Itabirito* para a obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Me. Elias José de Rezende Freitas

Coorientador: Prof. Carlos Dias de Silva Júnior

Itabirito - MG
2023

FICHA CATALOGRÁFICA

B861p Brison, Filipe Vieira
2023

Protótipo Huet: Automação residencial utilizando libras / Filipe Vieira Brison. – 2023.

53 f. : il.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Avançado Itabirito, 2023.

Orientador: Me. Elias José de Rezende Freitas.

Coorientador: Esp. Carlos Dias da Silva Júnior.

1. Internet das coisas. 2. Libras. 3. Visão computacional. 4. Reconhecimento de mão. I. Brison, Filipe Vieira. II. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Avançado Itabirito. III. Título.

CDD 419

Elaborada pela Biblioteca Jarbas Nazareth de Souza – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Avançado Itabirito

Bibliotecário Responsável: Veríssimo Amaral Matias – CRB-6/3266

PROTÓTIPO HUET: AUTOMAÇÃO RESIDENCIAL UTILIZANDO LIBRAS

Trabalho de conclusão de curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Educação Ciência e Tecnologia de Minas Gerais - Campus Avançado Itabirito para a obtenção do título de Engenheiro Eletricista.

Aprovado em 24/05/2023 pela banca examinadora:

Prof. Me. Elias José de Rezende Freitas (IFMG - Campus Ibirité)
Orientador

Prof. Esp. Carlos Dias da Silva Junior (IFMG - Campus Ibirité)
Coorientador (presidente da banca avaliadora)

Prof. Dr. Pedro Henrique Ferreira Machado (IFMG - Campus Ibirité)
Membro avaliador

Profa. Dra. Tamires Martins Rezende (FITec BH)
Membro avaliador



Documento assinado eletronicamente por **Carlos Dias da Silva Junior, Professor EBTT**, em 24/05/2023, às 22:34, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Pedro Henrique Ferreira Machado, Professor**, em 24/05/2023, às 22:34, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Tamires Martins Rezende, Usuário Externo**, em 24/05/2023, às 22:37, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **1563966** e o código CRC **D7209B03**.

Dedico esta monografia aos meus amados pais e aos meus irmãos, maiores incentivadores e fontes inesgotáveis de apoio, amor e compreensão.

AGRADECIMENTOS

Agradeço aos meus pais e aos meus irmãos agradeço por acreditarem em mim e pelo incentivo constante na realização deste trabalho.

Agradeço ao meu intérprete de Libras Paulo José Chaves Mendanha pela tradução e incentivo durante todo o tempo do curso. Agradeço às minhas intérpretes Estela Dias Figueiredo e Juanita Vitorino de Matos também.

Agradeço ao meu orientador Elias José de Rezende Freitas e ao coorientador Carlos Dias de Silva Júnior que contribuíram para a realização deste trabalho.

Agradeço aos professores e colegas do IFMG pelo aprendizado e apoio.

Agradeço à minha família e aos amigos surdos e ouvintes pelo apoio no caminho, do início até fim do curso.

“O que sabemos é uma gota; o que ignoramos é um oceano.”

Isaac Newton

RESUMO

Este trabalho apresenta um protótipo de automação residencial, no contexto de Internet das Coisas, que se diferencia por ser capaz de reconhecer alguns sinais de Libras para acionar equipamentos. O nome do protótipo é Huet, em homenagem ao francês Ernest Huet, que educava e incentivava os alunos surdos a aprenderem a língua de sinais no Brasil. O protótipo está na fase inicial, sendo composto de um nó central (um microcomputador de baixo custo, Raspberry Pi) que realiza o processamento das imagens capturadas de forma descentralizada por várias plataformas microcontroladas ESP32-CAM, que podem ser distribuídas pela casa. Ao identificar um determinado sinal de Libras pré-programado, o nó central envia uma mensagem via WiFi para outro microcontrolador que aciona/desaciona um relé, permitindo ligar ou desligar uma lâmpada ou um ventilador etc. Os resultados mostram o funcionamento geral do protótipo Huet com a utilização de um interruptor paralelo, realizando a análise do tempo e a quantidade dos acertos da identificação dos símbolos. Além disso, a esse trabalho contribui com a comunidade surda, por meio da tecnologia, permitindo romper as barreiras de comunicação dessa comunidade com a sociedade.

Palavras-chave: Internet das Coisas. Libras. Visão computacional. Reconhecimento de mão.

ABSTRACT

This work presents a home automation prototype, in the context of the Internet of Things, which stands out for being able to recognize some Signs of Libras to activate equipment. The name of the prototype is Huet, in honor of the Frenchman Ernest Huet, who educated and encouraged deaf students to learn sign language in Brazil. The prototype is in its initial phase, consisting of a central node (a low-cost microcomputer, Raspberry Pi) that performs the processing of images captured in a decentralized way by several ESP32-CAM microcontrolled platforms, which can be distributed throughout the house. When identifying a certain pre-programmed Libras sign, the central node sends a message via WiFi to another microcontroller that activates/deactivates a relay, allowing to turn on or off a lamp or a fan, etc. The results show the general functioning of the Huet prototype with the use of a parallel switch, performing the analysis of time and the number of correct identification of the symbols. In addition, this work contributes to the deaf community, through technology, allowing to break the communication barriers of this community with society.

Keywords: Internet of Things. Libras. Computer Vision. Hand Recognition.

LISTA DE ILUSTRAÇÕES

Figura 1 – Foto de <i>Ernest Huet</i>	18
Figura 2 – Representação do alfabeto manual em Libras.	19
Figura 3 – Representação dos números em Libras.	19
Figura 4 – Representação dos sinais em Libras.	19
Figura 5 – Constituintes da célula.	21
Figura 6 – Modelo de um neurônio artificial.	21
Figura 7 – Convenção dos eixos para representação de imagens digitais.	22
Figura 8 – Etapas do processamento de imagens.	23
Figura 9 – Detecção dos animais.	25
Figura 10 – Conjuntos da Ciência da Computação.	26
Figura 11 – Topologia da Rede Neural Convolutacional.	28
Figura 12 – Filtros das camadas convolucionais.	29
Figura 13 – Logo da biblioteca <i>OpenCV</i>	29
Figura 14 – <i>MediaPipe</i>	30
Figura 15 – <i>Hand land marks</i>	30
Figura 16 – Detecção de mão.	30
Figura 17 – <i>Raspberry Pi 4</i>	31
Figura 18 – Sistema operacional <i>Raspberry Pi</i>	32
Figura 19 – ESP32-CAM.	32
Figura 20 – ESP8266 - ESP-01.	33
Figura 21 – Componentes de um relé do tipo eletromecânico	33
Figura 22 – Módulo relé.	34
Figura 23 – Instalação em uma residência.	34
Figura 24 – Circuito de iluminação.	35
Figura 25 – Multiprotótipo em cada local residencial.	37
Figura 26 – Nós centralizados enviam HTTPs diferentes para nó central.	38
Figura 27 – Interruptor paralelo.	39
Figura 28 – Decisão da letra A	40
Figura 29 – Fluxograma do primeiro teste	40
Figura 30 – Fluxograma do segundo teste	41
Figura 31 – Sistema do protótipo <i>Huet</i>	42
Figura 32 – <i>Raspberry Pi 4</i>	43
Figura 33 – ESP32-CAM	43
Figura 34 – ESP8266, Relé e Lâmpada	44
Figura 35 – Esquema geral	44
Figura 36 – OpenCV e MediaPipe	45
Figura 37 – Identificação da letra A	45
Figura 38 – Coordenada de cada ponto na tela	46

Figura 39 – Indicações do pontos no gesto L	46
Figura 40 – Resultado do primeiro teste	47
Figura 41 – Resultado do segundo teste	47
Figura 42 – Pontos e linhas duplicados no gesto do número 3	47

LISTA DE TABELAS

Tabela 1 – Tabela da classificação de graus de surdez	17
Tabela 2 – Tabela comparativa de trabalhos relacionados.	36

LISTA DE ABREVIATURAS E SIGLAS

ASL	American Sign Language
CNN	Convolution Neural Network
CSL	Chinese Sign Language
DL	Deep Learning
FSL	Flemish Sign Language
IA	Inteligência Artificial
IDE	Integrated Development Environment
INES	Instituto Nacional de Educação de Surdos
IoT	Internet of Things
ISL	Indian Sign Language
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
k-NN	k-Nearest Neighbors
LED	Light Emitting Diode
Libras	Língua Brasileira de Sinais
MATLAB	Matrix Laboratory
MLP	Multilayer Perceptron
ML	Machine Learning
NA	Contato Normalmente Aberto
NF	Contato Normalmente Fechado
OpenCV	Open Source Computer Vision Library
RAM	Random Access Memory
RGB-D	Red, Green and Blue and Depth
RNA	Rede Neural Artificial
RPi	Raspberry Pi

SD	Secure Digital Card
SVM	Support Vector Machine
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	<i>Objetivo geral</i>	15
1.1.2	<i>Objetivos específicos</i>	15
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Surdez	17
2.2	Língua Brasileira de Sinais	17
2.3	Redes Neurais Artificiais (RNAs)	20
2.4	Processamento de Imagens e Visão Computacional	22
2.4.1	<i>Processamento de Imagens</i>	22
2.4.2	<i>Visão Computacional</i>	24
2.5	<i>Machine Learning (ML)</i>	24
2.5.1	<i>Deep Learning (DL)</i>	26
2.5.2	<i>Convolutional Neural Network (CNN)</i>	27
2.6	Materiais e métodos: <i>OpenCV</i> e <i>MediaPipe</i>	29
2.7	<i>Raspberry Pi 4</i>	31
2.8	Módulo ESP32-CAM	31
2.9	Módulo ESP8266 - ESP-01	32
2.10	Módulo Relé	33
2.11	Instalação Elétrica	34
2.12	Trabalhos Relacionados	35
3	METODOLOGIA	37
3.1	Arquitetura de comunicação e seus componentes	37
3.2	Processamento de imagens no Protótipo <i>Huet</i>	38
3.3	Python com <i>OpenCV</i> e <i>MediaPipe</i> e Fluxogramas	39
3.4	Sistema do protótipo <i>Huet</i>	41
4	RESULTADOS	43
4.1	Protótipo físico desenvolvido	43
4.2	Visão geral do esquema eletrônico	43

4.3	Testes da identificação das letras	44
4.4	Funcionamento geral do protótipo	48
5	CONCLUSÃO E TRABALHOS FUTUROS	49
5.1	Conclusão	49
5.2	Trabalhos Futuros	49
	REFERÊNCIAS	50

1 INTRODUÇÃO

Atualmente, estima-se no mundo que existam mais de 466 milhões de pessoas surdas ou com deficiência auditiva, de acordo com a Organização Mundial da Saúde (UNIDAS, 2020), sendo que cerca de 34 milhões dessas pessoas são crianças (UNICEF, 2019). No Brasil, cerca de 5% da população, o que equivale a aproximadamente 10 milhões de pessoas, possuem algum tipo de deficiência auditiva. No caso de pessoas que possuem perda auditiva total ou quase total, consideradas surdas, esse número no Brasil chega a 2,3 milhões, de acordo com o Instituto Locomotiva e a Semana da Acessibilidade Surda (BRASIL, 2021).

A audição é constituída por um sistema de canais que conduz o som até o ouvido interno, onde essas ondas são transformadas em estímulos elétricos, e, estes, por sua vez, são enviados ao cérebro, órgão responsável pelo reconhecimento e identificação daquilo que ouvem (MINISTÉRIO DA SAÚDE, 2023). Por sua vez, a surdez pode ser definida como impossibilidade ou dificuldade de ouvir, podendo ser classificada em ligeira, média, severa, profunda e cofose. Além disso, a surdez pode ser causada por diferentes fatores, como problemas genéticos, exposição a ruídos altos, infecções e lesões. Algumas pessoas nascem surdas, enquanto outras perdem a audição ao longo da vida.

O Brasil tem uma das maiores comunidades surdas do mundo, sendo que essa comunidade surda é formada não só por pessoas propriamente surdas, mas também por familiares, amigos, educadores, intérpretes e outros profissionais (SURDA, 2018).

Uma das barreiras dessa comunidade é a comunicação com a sociedade. Nesse sentido, a tecnologia tem um potencial de auxiliar e a promover inclusão nas várias áreas da sociedade. Atualmente, existem tecnologias como: aparelho auditivo, legenda automática, intérprete de Libras virtual pelo aplicativo, campainha e alarme para surdos, transcrição de voz em texto no celular e outros.

Por exemplo, o *Echo Dot*, criado pela Amazon, é um alto-falante inteligente, controlado por voz. É um dispositivo compacto e acessível, amplamente utilizado em casas inteligentes por meio da voz, cujos donos procuram uma maneira conveniente e prática de gerenciar suas tarefas diárias. O morador usa a assistente virtual Alexa para executar uma variedade de tarefas, como, por exemplo, tocar música, definir alarmes e lembretes, fazer chamadas, informar-se sobre notícias e clima, controlar dispositivos domésticos inteligentes, entre outras coisas. No entanto, as pessoas surdas sentem-se limitadas com esse tipo de tecnologia. Ao pretender controlar dispositivos domésticos inteligentes, como o *Echo Dot*, um usuário com este perfil pode se deparar com possíveis dificuldades impostas pelo dispositivo para que a sua voz – a voz de uma pessoa surda – seja compreendida. Se ele a fala for muito baixa ou pouco clara, por exemplo, o *Echo Dot* também pode ter dificuldade para interpretar e compreender os comandos de voz.

Nesse contexto, este trabalho apresenta o protótipo *Huet* constituído pelos dispositivos eletrônicos, *Raspberry Pi 4* e módulos ESP32CAM e ESP8266, com as linguagens de progra-

mação *Python* e *C++*, para proporcionar acessibilidade às pessoas com deficiência auditiva. A proposta deste estudo é proporcionar às pessoas surdas ou com deficiência auditiva os benefícios semelhantes ao que o *Echo Dot* traz para os ouvintes, por meio da voz. A diferença é que, em vez de utilizar a voz, o protótipo desenvolvido permitirá que os usuários utilizem as mãos para os comandos, realizados por meio de gestos, assemelhando-se à Língua de Sinais Brasileira. Para superar as barreiras de comunicação, desenvolveu-se o protótipo baseado no *Raspberry Pi*, que roda um algoritmo de visão computacional para controlar as funções de automação residencial.

A metodologia utilizada foi feita com arquitetura de comunicação e seus componentes, interruptor paralelo, processamento de imagens no Protótipo Huet, Python com OpenCv e MediaPipe e fluxogramas. Os resultados encontrados apontam para a quantidade dos acertos, a diferença entre tempo médio do primeiro teste e do segundo, que é algoritmo adaptado. Mais detalhes poderão ser vistos nas seções seguintes.

O nome do projeto foi uma homenagem a uma figura muito importante, chamada Ernest Huet. Ele nasceu em 1820 e foi um professor e poliglota francês que ensinava a língua de sinais para as pessoas surdas. Escolher este nome para o projeto é uma forma de reconhecer alguém que fez muito pelos surdos.

Assim, este trabalho de conclusão de curso justifica-se pela atual e crescente discussão acerca da acessibilidade para pessoas com deficiência. De forma mais específica, esta proposta de estudo busca contribuir para a melhoria da qualidade de vida de pessoas com deficiência auditiva, frente às dificuldades enfrentadas no seu dia a dia, promovendo mais inclusão social por meio da tecnologia.

1.1 Objetivos

1.1.1 *Objetivo geral*

Este trabalho tem por objetivo desenvolver um dispositivo eletrônico de automação residencial, por meio de gestos.

1.1.2 *Objetivos específicos*

Como objetivos específicos, pode-se citar:

- Estudar e aplicar conceitos e técnicas de visão computacional;
- Construir um protótipo que permita visualizar as propostas do trabalho para acessibilidade às pessoas com deficiência auditiva. Este dispositivo deve ser de fácil ajuste e utilização;
- Levantar possíveis funcionalidades adicionais para o dispositivo proposto, como forma de garantir inovação ao projeto e contribuir ainda mais para a melhoria da qualidade de vida das pessoas com deficiência auditiva;

- Promover a garantia de acessibilidade para redução das barreiras na vida das pessoas com deficiência auditiva.

2 REVISÃO BIBLIOGRÁFICA

2.1 Surdez

A surdez consiste na perda maior ou menor da percepção normal dos sons. Verifica-se a existência de diferentes tipos de surdez, que variam de acordo com os diferentes graus de perda da audição (LIMA *et al.*, 2006). Há, por exemplo, pessoas totalmente surdas (com perda auditiva bilateral de grau profundo), pessoas parcialmente surdas e pessoas surdocegas. Além disso, a definição da pessoa surda passou a ser referente ao indivíduo que, apesar da perda auditiva, compreende o mundo e, ao mesmo tempo, interage com ele por meio de experiências visuais (ALVES; FRASSETTO, 2015).

A perda auditiva é caracterizada como um problema sensorial não visível, que acarreta dificuldades na detecção e percepção dos sons. Devido à natureza complexa do ser humano, traz sérias consequências para o desenvolvimento do indivíduo, já que padrões sociais, emocionais, linguísticos e intelectuais estão correlacionados (ARAÚJO; LACERDA, 2008). Na Tabela 1, é possível observar os cinco tipos de graus de surdez, com seus respectivos valores de referência em decibéis (dB): à medida que o nível da perda auditiva aumenta, o valor da perda em decibéis é maior.

Tabela 1 – Tabela da classificação de graus de surdez

Grau de Surdez	Perda dB
Audição Normal	Limiares auditivos abaixo de 25 dB
Perda Auditiva Leve	Limiares acima de 25 e abaixo de 40 dB
Perda Auditiva Moderada	Limiares acima de 40 e abaixo de 70 dB
Perda Auditiva Severa	Limiares acima de 70 e abaixo de 90 dB
Perda Auditiva Profunda	Limiares acima de 90 dB

Fonte: <https://portalotorrino.com.br>

2.2 Língua Brasileira de Sinais

A Língua Brasileira de Sinais (Libras) é utilizada pelas pessoas surdas como meio de comunicação e é o resultado da combinação de movimentos e configurações das mãos, de pontos de articulação, orientação e direcionalidade, de expressão facial e corporal.

A Libras foi desenvolvida desde a fundação do Instituto Nacional de Educação de Surdos (INES) em 1857, quando o francês *Ernest Huet*, representado na Figura 1, a seguir, foi convidado por Dom Pedro II, no século XIX, para ensinar os sinais aos surdos do novo país. Ao longo dos anos, a língua de sinais já foi proibida na educação de surdos, mas hoje é vista como um início de transformação na educação (MUNCINELLI, 2013).

Figura 1 – Foto de *Ernest Huet*.

Fonte: <https://www.timetoast.com>.

De acordo com o Art. 1º da Lei Nº 10.436, de 24 de abril de 2002 (BRASIL, 2002), percebe-se como a Libras é definida e como a comunicação nesta língua passa a ser um direito reconhecido legalmente aos surdos:

“Art. 1º É reconhecida como meio legal de comunicação e expressão a Língua Brasileira de Sinais - Libras e outros recursos de expressão a ela associados.

Parágrafo único. Entende-se como Língua Brasileira de Sinais - Libras a forma de comunicação e expressão, em que o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil.”(BRASIL, 2002).

O dicionário de Libras, intitulado *O Dicionário da Língua de Sinais do Brasil: A Libras em Suas Mãos*¹, documenta 14.500 sinais em entradas lexicais individuais, trazendo os verbetes correspondentes a cada um deles nas versões em português e em inglês. A publicação apresenta a definição do significado do sinal e dos verbetes, ilustrações e a descrição detalhada de como o sinal é produzido corporalmente (USP, 2018). Nas Figuras 2 a 4 são exibidas algumas dessas representações mais comuns na comunicação: o alfabeto manual, números e sinais.

¹ O dicionário mencionado no texto fonte é de autoria de Fernando Capovilla, Walkiria Duarte Raphael, Janice Gonçalves Temoteo e Antonielle Cantarelli Martins, e foi publicado pela Editoria da USP.

Figura 2 – Representação do alfabeto manual em Libras.



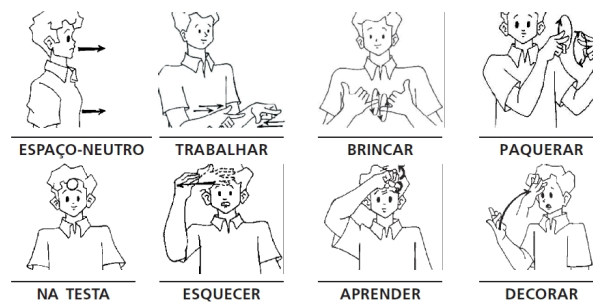
Fonte: <https://acessibilidadeemmaos.wordpress.com>.

Figura 3 – Representação dos números em Libras.



Fonte: <http://librasnamatematica.blogspot.com>.

Figura 4 – Representação dos sinais em Libras.



Fonte: <https://www.libras.com.br>.

As línguas de sinais não são universais, pois cada país possui a sua própria língua, que sofre inclusive influências da cultura local e, como qualquer outra língua, também possui expressões distintas de região para região, não existindo, assim, padronização em âmbito nacional.

Ao contrário do que muitos imaginam, as línguas de sinais não são simplesmente mímicas, ou apenas gestos individuais utilizados pelos surdos para facilitar a comunicação. As línguas de sinais possuem estruturas gramaticais próprias que podem ser observadas nos níveis linguísticos fonológico, morfológico, sintático e semântico. Logo, o que a diferencia das demais línguas (orais) é a sua modalidade visual-espacial (MUNCINELLI, 2013).

2.3 Redes Neurais Artificiais (RNAs)

As Redes Neurais Artificiais (RNAs) são baseadas em neurônios biológicos e são sistemas computacionais interconectados que trabalham o processamento dos dados como o cérebro humano. Elas são as partes de Inteligência Artificial, na área de Ciência da Computação, que desenvolvem sistemas capazes de agir de modo análogo aos humanos (XPEDUCAÇÃO, 2022). Nessa Seção serão apresentados alguns conceitos fundamentais de RNAs, em um resumo histórico baseado em (CARVALHO; COLLABORATORS, 2021) e (FREITAS; SANTOS; REZENDE, 2022).

O neurofisiologista McCulloch e o matemático Pitts desenvolveram e publicaram em 1943 o primeiro modelo de redes neurais, simulando “máquinas”; o psicólogo Hebb desenvolveu o modelo básico de rede de auto-organização, tornando-o público em 1949; e outro psicólogo, Rosenblatt, criou o modelo Perceptron de aprendizado supervisionado divulgando-o cientificamente em 1958.

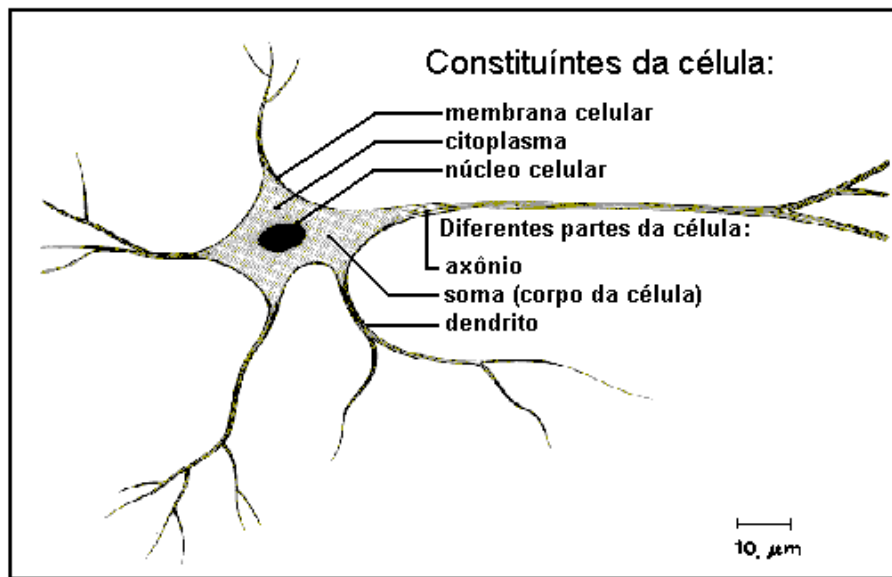
Alguns estudos históricos sobre a área costumam “ignorar” os anos 60 e 70 e sinalizarem o destaque da área com a publicação dos trabalhos de Hopfield - que relatam a utilização de redes simétricas para otimização - e de Rumelhart, Hinton e Williams (1982), que introduziram o poderoso método *Backpropagation*.

Entretanto, para se ter um panorama histórico completo, devem ser citados alguns pesquisadores que realizaram, nos anos 60 e 70, importantes trabalhos sobre modelos de redes neurais em visão, memória, controle e auto-organização como: Amari, Anderson, Cooper, Cowan, Fukushima, Grossberg, Kohonen, von der Malsburg, Werbos e Widrow.

As RNAs possuem uma unidade básica chamada neurônio, livremente inspirada no neurônio biológico. Essas estruturas recebem um conjunto de entradas em seus “axônios” e produzem uma saída em seu “dendrito”. Assim como os neurônios biológicos, mostrados na Figura 5, os neurônios artificiais também podem aprender a resolver um problema. No caso computacional, os neurônios que compõem uma RNA são adaptados em seu processo de treinamento para fazerem previsões a partir de dados. Em aplicações modernas, utiliza-se uma classe de treinamento para neurônios multi camadas chamada de *Deep Learning* (IMOBILIS, 2017).

Os neurônios artificiais são estruturados com uma arquitetura baseada nos biológicos, mas adaptada à realidade computacional. As entradas são ponderadas por pesos e associadas de acordo com uma função chamada “função de ativação”. Essa função é responsável por combinar

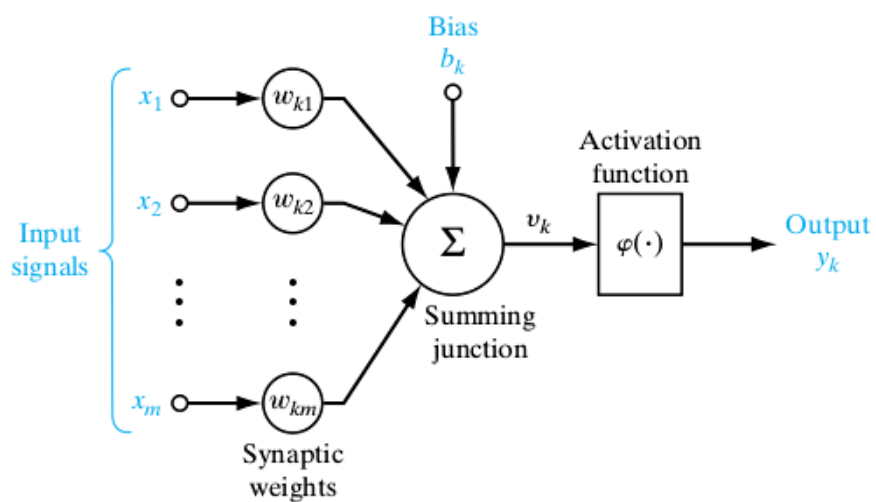
Figura 5 – Constituintes da célula.



Fonte: <https://sites.icmc.usp.br/>.

linearmente os valores de entrada ponderados pelos pesos (e eventualmente um valor de BIAS), ativando o valor da saída. A Figura 6 mostra o funcionamento de um neurônio genérico. Estruturas como essas são usadas dentro de todas as redes neurais (IMOBILIS, 2017). Existem as funções de ativação, que são utilizadas dependendo de uma aplicação. Algumas funções de ativação: Linear, Degrau, Sigmoides, Tangente Hiperbólica, Unidade Linear Retificada e Softmax (IMOBILIS, 2017).

Figura 6 – Modelo de um neurônio artificial.



Fonte: <http://www2.decom.ufop.br>.

2.4 Processamento de Imagens e Visão Computacional

2.4.1 Processamento de Imagens

O processamento de imagens pode ser definido como técnicas que manipulam a qualidade das imagens por meio de procedimentos, como redução de ruídos, ajuste de brilho, contraste e resolução. O objetivo deste processo é a restauração da imagem, cuja entrada é capturada por meio do dispositivo de câmera.

A área de processamento digital de imagens vem evoluindo continuamente ao longo dos anos, com um aumento significativo de estudos envolvendo morfologia matemática, redes neurais, processamento de imagens coloridas, compressão e reconhecimento de imagens, e sistemas de análise de imagens baseados em conhecimento (GONZALEZ; WOODS, 2002).

A definição de uma imagem é uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando x, y e os valores de intensidade de f são quantidades finitas e discretas, chamamos de imagem digital (GONZALEZ; WOODS, 2002), Como ilustra a Figura 7.

Figura 7 – Convenção dos eixos para representação de imagens digitais.

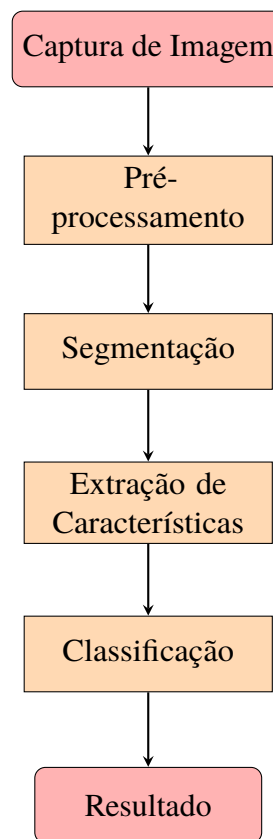


Fonte: GONZALEZ e WOODS (2000).

As etapas de processamento de imagens, representadas na Figura 8, são as mais comuns nas áreas de Engenharia Elétrica, Ciência da Computação e outras disciplinas.

- **Captura de Imagem:** Os autores Gonzalez e Woods (2002) mencionam mencionam que a

Figura 8 – Etapas do processamento de imagens.



Fonte:Elaborado pelo autor, 2023

aquisição consiste em receber uma imagem em formato digital. O pré-processamento de imagens, que abrange, por exemplo, o redimensionamento, é contemplado nesta etapa de aquisição de imagens.

- **Pré-processamento:** Correções dos defeitos gerados durante a aquisição na etapa de pré-processamento, como otimização da imagem (OLIVEIRA, 2004).
- **Segmentação:** As funções da etapa de segmentação são extração e identificação de áreas de interesse contidas na imagem. Esta etapa consiste na detecção de bordas ou de regiões similares na imagem (CAMPINAS, 2012).
- **Extração de Características:** A etapa de extração de características consiste em extrair as informações mais importantes contidas em cada imagem, depois de uma etapa de pré-processamento (NETTO; ALMEIDA; NETO, 2004).
- **Classificação:** A classificação de imagens determina uma classe como um grupo de pixel no processo de decisão. Dessa forma, um usuário tem o auxílio de sistemas computacionais na interpretação das imagens (VENTURIERI; SANTOS, 1998).

- **Resultado:** O resultado consiste em exibir uma imagem, em uma janela ou tela, como uma saída do sistema do processamento de imagens, depois da classificação.

2.4.2 *Visão Computacional*

Visão computacional é uma área da Ciência da Computação e da Inteligência Artificial (IA) que busca analisar, interpretar e extrair informações relevantes de imagens e/ou vídeos para que decisões possam ser tomadas ou, até mesmo, para gerar dados relevantes para uma aplicação futura (DIGITAL, 2021). A ideia é simular a visão humana, já que somos capazes de detectar objetos, entender distâncias e compreender o que há em um ambiente.

A visão computacional tem relações, por exemplo, com processamento de imagens, aprendizado de máquina, recuperação de informação, sistemas cognitivos e neurociência nas áreas de Engenharia, Ciência da Computação, Matemática, Biologia, Física e outras. Suas funções típicas são: reconhecimento, reconstrução de cena e restauração de imagens. Além disso, os sistemas da visão computacional possuem as seguintes tarefas: aquisição de imagem, pré-processamento, extração de características, detecção e segmentação e processamento de alto nível. Em suas aplicações, a visão computacional é utilizada na área da Medicina, para fazer diagnósticos sobre pacientes; na área industrial, para a detecção do tamanho ou defeito como inspeção; na área militar, para detecção dos soldados e dos mísseis da oposição; na área de automação veicular, para detectar pessoas, veículos, imóveis e materiais de trânsito para evitar acidentes. Portanto, o resultado da visão computacional permite a detecção de objetos - como a identificação dos animais mostrados na Figura 9, que são sinalizados de formas distintas por meio dos retângulos.

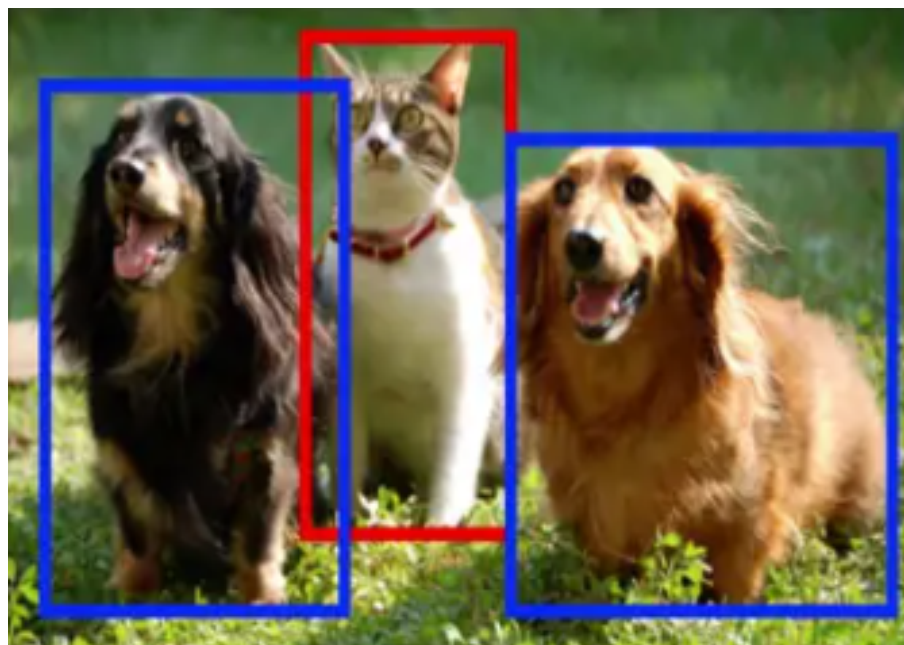
De acordo com FoxFly (2021), temos que:

“A diferença entre processamento de imagens e visão computacional é que enquanto o processamento de imagens foca em aperfeiçoar a imagem para facilitar a extração de informações, a visão computacional busca emular a visão humana, entender a imagem como um todo e utilizar as informações obtidas para a tomada de decisões.”(FOXFLY, 2021).

2.5 *Machine Learning (ML)*

Machine Learning ou Aprendizado de Máquina é uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseadas em experiências acumuladas, decorrentes de soluções bem sucedidas de problemas anteriores (MONARD; BARANAUSKAS, 2003). O começo do processo é a aprendizagem da máquina sobre os caminhos certos e errados. Depois diminui-se possibilidades dos caminhos errados, aumentando eficiência. As imagens e os

Figura 9 – Detecção dos animais.



Fonte: <https://didatica.tech>.

áudios são transformados em valores matemáticos que a máquina aprende para conseguir maior quantidade de acertos do que de erros. Portanto, o computador adquire capacidade de aprender e entender automaticamente, sem precisar ser programado várias vezes.

Há vários algoritmos de ML: Árvores de decisão, Floresta aleatória, SVM (*Support Vector Machine*), k-NN (K-vizinhos mais próximos), CNN (*Convolutional Neural Network*), *Gauss Mixture Model* etc.

Como a máquina vai treinar automaticamente? Inserem-se os dados com atributos ou recursos variados que o algoritmo, então, irá aprender e limitar a decisão com base nos dados que um programador fornece. Após isso, na fase de teste, colocam-se os dados de teste para ver alguns resultados como desejados. Os dados de treino consistem em informações que serão apresentadas ao algoritmo de ML para a criação do modelo. Esses representam cerca de 70% da totalidade dos dados (DIDÁTICA TECH, 2022). Os dados de teste consistem naqueles que são apresentados ao modelo depois da sua criação, simulando previsões de fatos que o modelo fará, permitindo assim que o desempenho real seja verificado. Estes dados representam aproximadamente 30% da totalidade dos dados (DIDÁTICA TECH, 2022).

Existem 3 formas de realizar o aprendizado, que são apresentadas abaixo:

Aprendizado supervisionado: é uma técnica em que se supervisiona todo o processo de aprendizado. Esse tipo de acompanhamento tem como objetivo prever o resultado, dado um conjunto de amostras de treinamento, junto com seus rótulos, ou, em outras palavras, classificação de um ponto de dados. Portanto, o algoritmo treina e prevê um dado.

Aprendizado não supervisionado: Diferente do tipo anterior, não se usa aqui os rótulos de treinamento para as amostras de treinamento. Formulam-se os algoritmos para poder encontrar estruturas e padrões adequados nos dados. Sua finalidade é permitir que a máquina aprenda sem a ajuda de um programador. Neste percurso, ela precisa aprender a ajustar os resultados e agrupamentos, quando tiver resultados mais adequados. Permite-se, desta forma, que a máquina entenda os dados, de forma autônoma, e processe-os como entender.

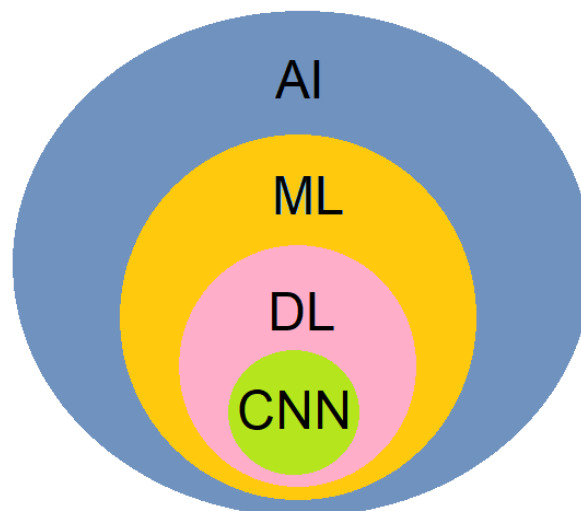
Aprendizado por reforço:

“O Aprendizado por Reforço (ou *Reinforcement Learning* – RL), conhecido como modelo de aprendizado semi-supervisionado em *Machine Learning*, é uma técnica para permitir que um agente tome ações e interaja com um ambiente, a fim de maximizar as recompensas totais.”(DEEP LEARNING BOOK, 2022).

Um robô, por exemplo, aprende como se comportar em um ambiente, toma ações e quantifica os resultados. Caso ele faça uma resposta certa, consegue um ponto de recompensa, o que aumenta a confiança do robô para fazer mais ações desse tipo.

Portanto, para se ter entendimento, apresenta-se geralmente os conjuntos dos níveis da área de Ciência da Computação, como se observa na Figura 10.

Figura 10 – Conjuntos da Ciência da Computação.



Fonte: Elaborado pelo autor, 2023

2.5.1 *Deep Learning* (DL)

O aprendizado profundo, ou *Deep Learning*, é um subconjunto do ML, em que redes neurais artificiais - algoritmos modelados para funcionar como o cérebro humano - aprendem com grandes quantidades de dados (ORACLE, 2023).

As camadas de redes neurais, que são algoritmos modelados livremente, do mesmo modo como os cérebros humanos funcionam, desenvolvem o aprendizado profundo. Com grandes quantidades de dados, o treinamento é o que configura os neurônios na rede neural. O resultado é um modelo de aprendizado profundo que, uma vez treinado, processa novos dados. Os modelos de aprendizado profundo absorvem informações de várias fontes de dados e analisam esses dados em tempo real, sem a necessidade de intervenção humana. No aprendizado profundo, as unidades de processamento gráfico (GPUs) são otimizadas para modelos de treinamento porque podem processar vários cálculos simultaneamente (ORACLE, 2023)

2.5.2 *Convolutional Neural Network (CNN)*

Uma Rede Neural Convolutiva (CNN) é um algoritmo de Aprendizado Profundo que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários aspectos / objetos da imagem e ser capaz de diferenciar um do outro. O pré-processamento exigido em uma CNN é muito menor em comparação com outros algoritmos de classificação. Enquanto, nos métodos primitivos, os filtros são feitos à mão, com treinamento suficiente, as CNNs têm a capacidade de aprender esses filtros / características (DEEP LEARNING BOOK., 2022).

A arquitetura de uma CNN é análoga àquela do padrão de conectividade de neurônios no cérebro humano e foi inspirada na organização do Visual Cortex. Os neurônios individuais respondem a estímulos apenas em uma região restrita do campo visual conhecida como Campo Receptivo. Uma coleção desses campos se sobrepõe para cobrir toda a área visual (DEEP LEARNING BOOK., 2022).

O módulo de convolução é responsável pela extração de elementos da imagem que descrevem o seu conteúdo. Já o módulo de classificação fica responsável pela tarefa de classificar os dados extraídos durante a etapa de convolução. O resultado da classificação é direcionado para a saída da rede (ATELIWARE, 2021).

O que diferencia as Redes Neurais Convolutivas das Redes Neurais Artificiais clássicas é o número de camadas intermediárias (ou, camadas escondidas). Em resumo, cada camada da Rede Convolutiva é responsável por uma sub-tarefa específica. As principais camadas que compõem uma rede neural convolutiva são (ATELIWARE, 2021):

- **Camada de convolução:** responsável por extrair, bem como mapear o conteúdo da imagem, transformando-o em dados. Esse processo ocorre por meio da aplicação de pequenos blocos, conhecidos como filtros, que permitem a obtenção das informações de sub-blocos da imagem;
- **Pooling:** a camada de pooling recebe os blocos que contêm as informações extraídas na etapa de convolução. Ela simplifica a informação, resumindo os dados do sub-bloco da

imagem em um único valor e os repassa para uma camada totalmente conectada;

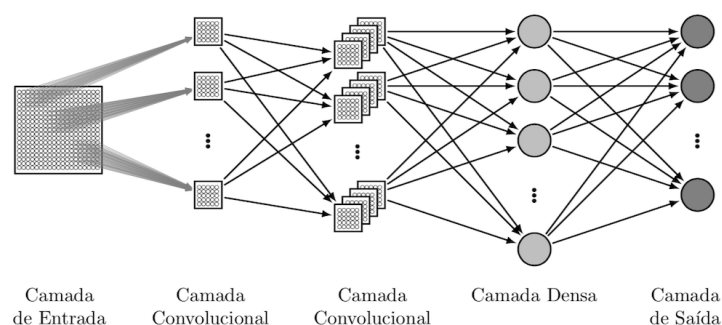
- **Camada totalmente conectada:** onde é iniciado o processo para classificar as informações extraídas pelas camadas anteriores. A camada totalmente conectada achata o sub-bloco contendo os dados extraídos, ou seja, o bloco é transformado em uma única linha que contém todas as informações extraídas.

Ao final, como classificação, a camada de *dropout* possui a responsabilidade de reduzir o overfitting - sobreajuste - da rede. Ele ocorre quando a rede decora os dados utilizados no aprendizado e não é capaz de aplicar as relações aprendidas em dados novos. É comum dizermos que, neste cenário, o modelo gerado, a partir da rede treinada, não tem capacidade de generalização, portanto, não apresenta bom desempenho quando colocado em produção (ATELIWARE, 2021).

A função de ativação, por sua vez, é responsável pelo aprendizado da rede, bem como pela relação entre as variáveis, decidindo quais neurônios serão ativados, por exemplo, com o uso de Softmax (ATELIWARE, 2021).

Observando-se a Figura 11, o diferencial das CNNs está nas diversas camadas convolucionais, que aplicam uma função matemática de Convolução nos dados de entrada e depois realizam o Agrupamento (*pooling*). A saída da convolução é passada para a próxima camada convolucional até chegar na última camada conhecida como Camada Densa, normalmente representada por uma rede *Perceptron* de múltiplas camadas (do inglês *Multilayer Perceptron* - MLP) (SAKURAI, 2017).

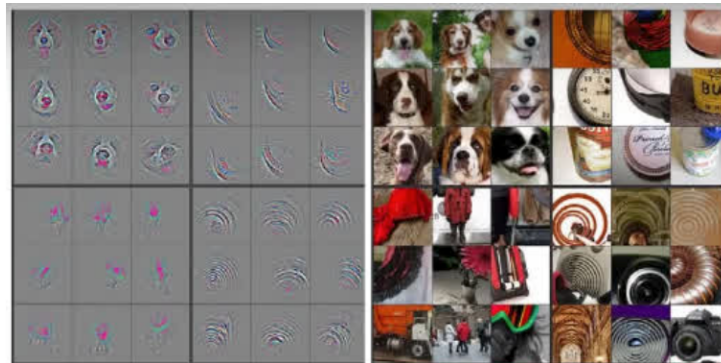
Figura 11 – Topologia da Rede Neural Convolucional.



Fonte: <https://www.sakurai.dev.br>.

Na Figura 12, a metade à esquerda representa os filtros das camadas convolucionais, que extraem os valores em pixel das imagens mais relevantes, enquanto a metade direita, da mesma figura, representa as imagens originais.

Figura 12 – Filtros das camadas convolucionais.

Fonte: <https://ateliware.com>

2.6 Materiais e métodos: *OpenCV* e *MediaPipe*

Essas bibliotecas *OpenCV* e *MediaPipe* são projetadas para resolver problemas de visão computacional. A biblioteca *OpenCV*, cuja logo está representada na Figura 13, foi desenvolvida pela *Intel* e possui mais de 500 funções (BRADSKI, 2000). Ela foi idealizada com o objetivo de tornar a visão computacional acessível a usuários e a programadores que atuam em áreas como a interação humano-computador em tempo real e a robótica (MARENGONI; STRINGHINI, 2009). Essa biblioteca possui módulos de Processamento de Imagens e Vídeo I/O, estrutura de dados, controle de mouse e teclado, álgebra linear e interface gráfica do usuário. Existem os algoritmos de Visão Computacional como filtros de imagem, calibração de câmera, reconhecimento de objetos e outros. O processamento dela se dá em tempo real.

Figura 13 – Logo da biblioteca *OpenCV*.Fonte: <https://opencv.org>.

A *MediaPipe*, cuja logo é ilustrada na Figura 14 oferece soluções de *Machine Learning* personalizáveis. Com esse *framework* é possível implementar rapidamente uma ML para identificação de rostos, segmentação de objetos, rastreamento de mãos e poses, identificação de íris, segmentação de cabelos, entre outras infinitudes de aplicações (FERNANDES; RIBEIRO, 2021).

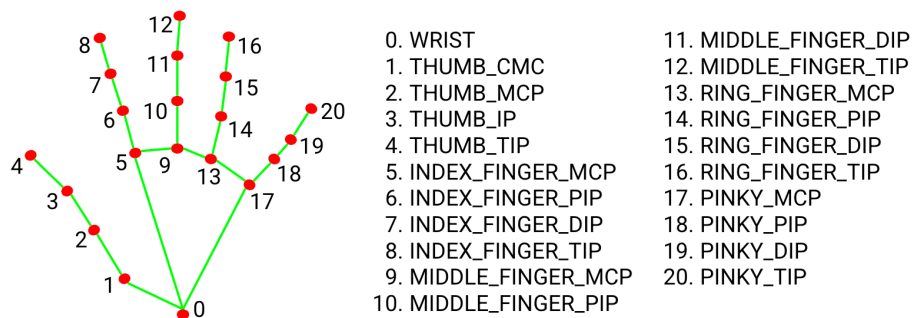
Figura 14 – *MediaPipe*.



Fonte: <https://mediapipe.dev/>.

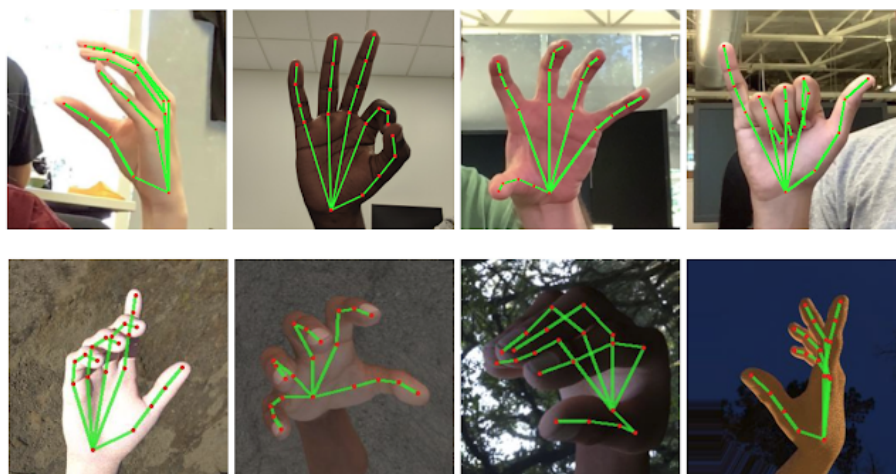
A biblioteca *MediaPipe* da mão possui 21 pontos de referência e 22 linhas, como ilustra a Figura 15. Esses pontos e linhas acompanham a mão em tempo real, mostrando gestos livres como exemplifica a Figura 16. Detectando-se uma mão, cada ponto é posicionado conforme as coordenadas da imagem. A CNN é utilizada nessa biblioteca para funcionar a detecção da mão.

Figura 15 – *Hand land marks*.



Fonte: <https://google.github.io>.

Figura 16 – Detecção de mão.



Fonte: <https://www.analyticsvidhya.com>.

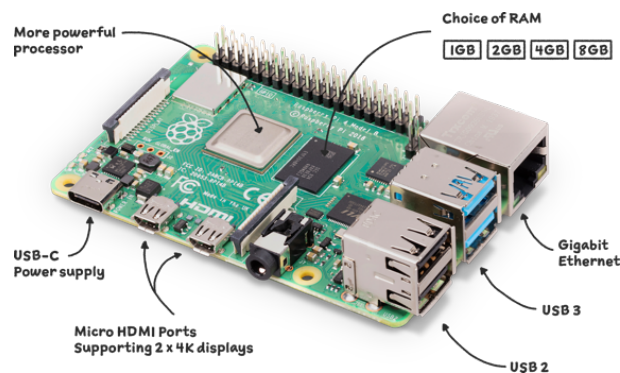
Na utilização da biblioteca da visão computacional é possível construir-se esse pipeline de percepção como um gráfico de componentes modulares. A *MediaPipe* foi criada para equipes

de ML, desenvolvedores de *software* que implementam aplicativos de ML prontos para produção, e/ou estudantes/pesquisadores que publicam códigos e protótipos como, por exemplo, braços robóticos, braços robóticos, reconhecimento de sono no carro e detecção de mão. Neste trabalho, utiliza-se, por meio do protótipo, a detecção de mãos, por causa da língua das pessoas surdas.

2.7 Raspberry Pi 4

O *Raspberry Pi 4* modelo B, como ilustrado na Figura 17, é um dispositivo eletrônico, que executa o algoritmo pronto por uma pessoa, faz atuação na saída desse dispositivo e é ligada a um dispositivo externo (mouse, monitor, câmera) por meio de uma entrada USB.

Figura 17 – *Raspberry Pi 4*.



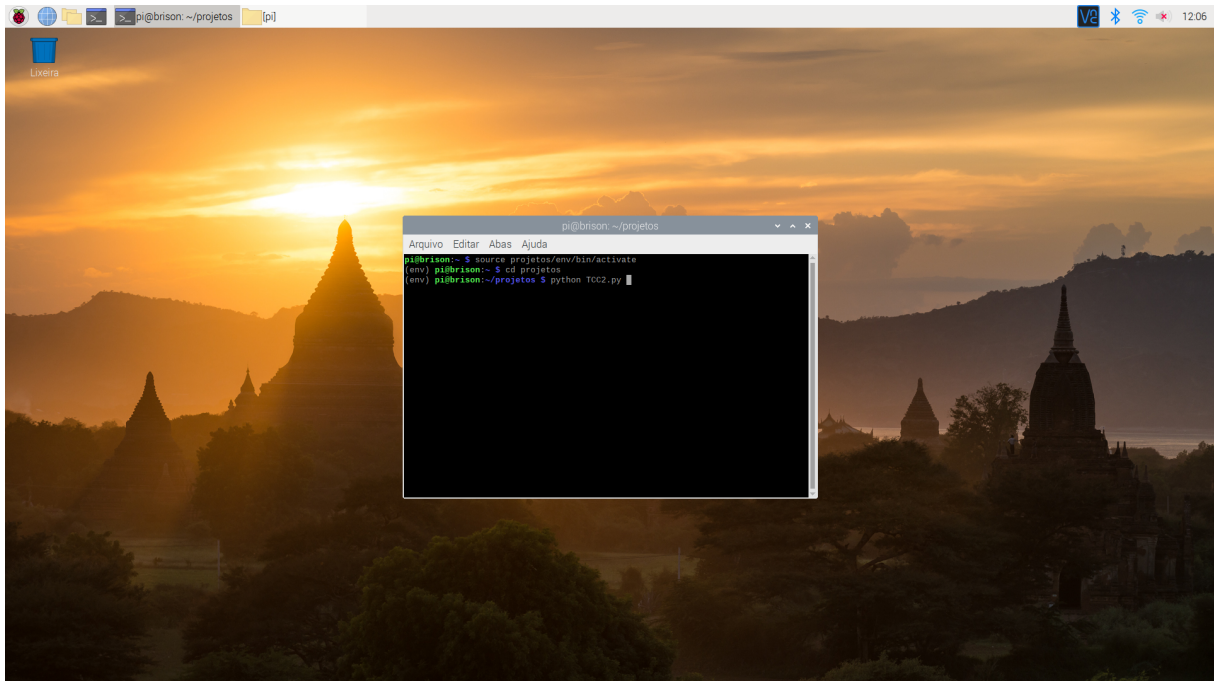
Fonte: <https://www.raspberrypi.com>.

O *Raspberry Pi* não é um microcontrolador, mas sim um microprocessador. É maior que o microcontrolador médio, usa mais energia, tem uma velocidade de *clock* entre 700MHz e 1,5GHz e uma RAM entre 1 e 8GB (TIPS, 2023). Para mais informações das especificações, acesse o link <https://static.raspberrypi.org>.

O *Raspberry Pi 4* possui o próprio sistema operacional, como pode se observar na Figura 18, baseado em *Debian Linux* para criar os códigos dentro desse dispositivo e executá-lo. A versão desse sistema operacional é chamada de *Buster*.

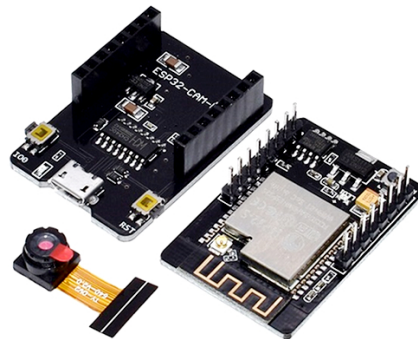
2.8 Módulo ESP32-CAM

O módulo ESP32-CAM, ilustrado na Figura 19 é uma placa de desenvolvimento ESP32 com uma câmera OV2640, suportando cartão micro SD para gravação de dados, além de contar com *flash* em LED integrado, além de vários pinos GPIOs para conectar periféricos (USINAINFO, 2022).

Figura 18 – Sistema operacional *Raspberry Pi*

Fonte: Autor, 2023

Figura 19 – ESP32-CAM.

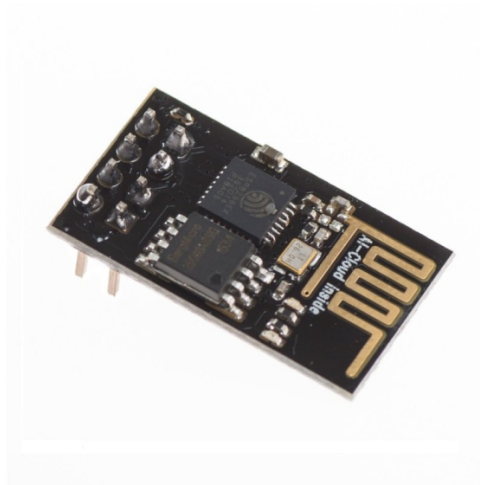


Fonte: <https://www.usinainfo.com.br>.

2.9 Módulo ESP8266 - ESP-01

O módulo WiFi ESP8266 ESP01, como apresenta a Figura 19, é um microcontrolador cuja função é a conectar projetos e sistemas na internet via Wi-Fi para que acessar, controlar o projeto que possa adquirir informações de forma distante através do chip ESP8266 (CURTOCIRCUITO, 2023). É possível usar esse módulo para ter comunicação com Arduino, PIC, Raspberry Pi ou outros microcontroladores (CURTOCIRCUITO, 2023).

Figura 20 – ESP8266 - ESP-01.

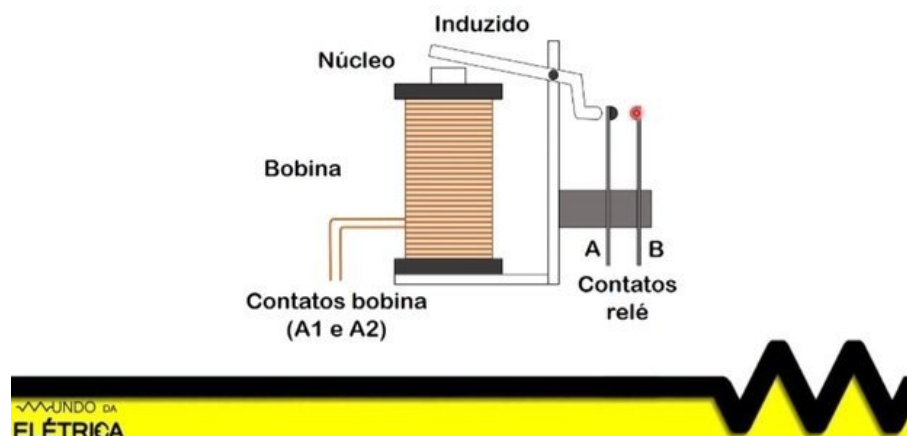


Fonte: <https://curtocircuito.com.br>.

2.10 Módulo Relé

O relé é um componente eletromecânico, ou seja, ele conta com uma parte mecânica de contato e o seu acionamento ocorre através da corrente elétrica em perpassa uma bobina. Na Figura 21, é possível visualizar todos os componentes de um relé eletromecânico, que são (ELÉTRICA, 2021): Bobina, Induzido, Núcleo, Contatos da bobina, Contatos do relé.

Figura 21 – Componentes de um relé do tipo eletromecânico

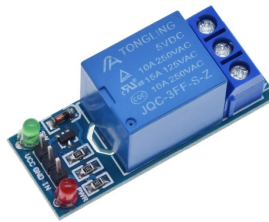


Fonte: <https://www.mundodaeletrica.com.br>.

A parte principal é a bobina de cobre que está envolvendo um núcleo de ferrite, uma espécie de eletroímã. No momento em que se acionar a bobina, o núcleo irá atrair o induzido, e ao mesmo tempo em que se desloca em direção à bobina, também empurra o contato A do relé em direção ao contato B, fechando-os entre si (ELÉTRICA, 2021).

O Módulo Relé de 1 Canal, como ilustrado na Figura 20, é um módulo que realiza o acionamento de relés, e que é conectado às placas microcontroladores, como, por exemplo, PIC ou Arduino. A vantagem é que os projetos tornam-se mais organizados e bonitos, permitindo também mais espaço (CURTOCIRCUITO., 2023). Neste trabalho, utiliza-se acionamento mecânico cujo módulo relé liga à lâmpada. Esse módulo é acoplado ao módulo ESP8266.

Figura 22 – Módulo relé.



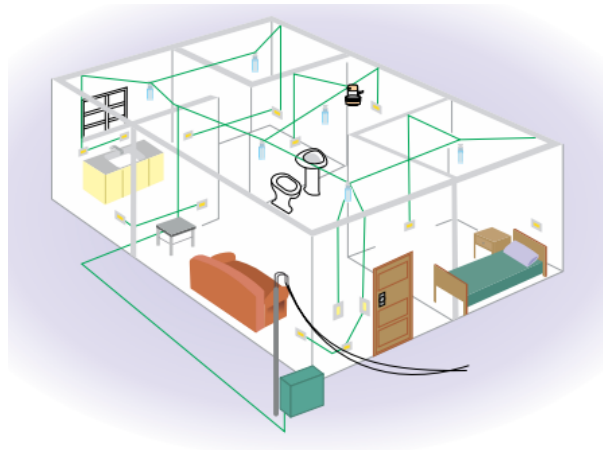
Fonte: <https://curtocircuito.com.br>.

2.11 Instalação Elétrica

A instalação elétrica compreende a implementação física dos componentes das ligações elétricas, a conexão entre a fonte geradora e as cargas elétricas. Nas instalações elétricas em baixa tensão, a fonte geradora vem da concessionária e as cargas são os eletrodomésticos e eletroeletrônicos que são conectados às tomadas (ELÉTRICA, 2021).

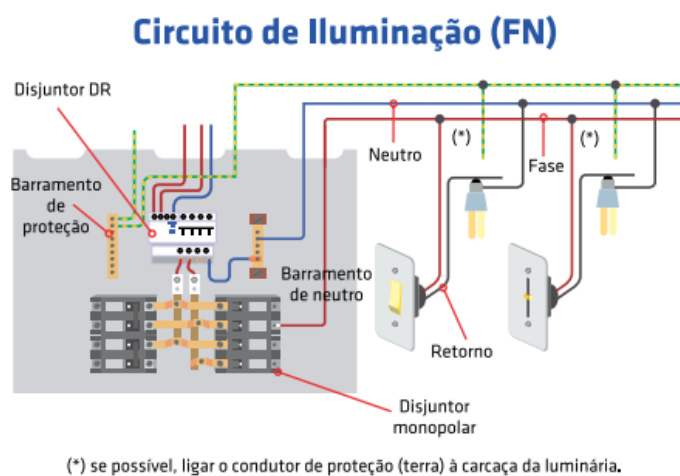
Nas Figuras 23 e 24, estão representados uma instalação em uma residência e um circuito de iluminação. Seria possível instalar um protótipo para automação residencial nos circuitos de iluminação.

Figura 23 – Instalação em uma residência.



Fonte: <https://br.prysmiangroup.com>.

Figura 24 – Circuito de iluminação.



Fonte: <https://br.prysmiangroup.com>.

2.12 Trabalhos Relacionados

De acordo com Bastos, Angelo e Loula (2015), as línguas de sinais representam o principal meio de comunicação entre as pessoas com deficiência auditiva. Elas não são universais, cada país possui a sua própria língua de sinais. E, dentre as várias línguas de sinais, este trabalho apresenta a Libras, que é utilizada pelos surdos brasileiros.

Em (AMARAL *et al.*, 2017) é fornecido uma visão geral de funções matemáticas de técnicas do processamento de imagens, para segmentação e binarização, Transformada de distância e Rede neural convolucional. Por sua vez, Costa *et al.* (2017) apresentaram trabalho com etapas semelhantes do reconhecimento de padrões utilizando a seguinte metodologia: aquisição de imagens, segmentação de configuração manual, extração de características e classificação.

De maneira similar, em (PANSARE; GAWANDE; INGLE, 2012) é apresentado o resultado do reconhecimento dos 26 gestos da mão, que foram testados para obter a quantidade dos acertos, em porcentagem, e a taxa geral de sucesso foi de 90,19% com o tempo de reconhecimento de cerca de 0,5 segundo, utilizando uma câmera da web de 10 megapixels e MATLAB. O resultado do TCC mostra a quantidade dos acertos, em porcentagem, no teste do gesto de cada símbolo mostrado à câmera.

Outro trabalho relacionado é apresentado em (SHENOY *et al.*, 2018), o qual utiliza técnicas de processamento de imagens e o algoritmo de k-NN, para obter resultado de 33 poses de mão e alguns gestos da Língua de Sinais Indiana. Com isso, os gestos foram capturados de uma câmera de celular e suas imagens foram transmitidas para um servidor remoto para processamento. Nesse TCC, as imagens são transmitidas do módulo para o Raspberry, através do Wi-Fi. Além disso, os testes com câmera são feitos em 7 símbolos.

Por exemplo, em Abed e Rahman (2017), as especificações do minicomputador Raspberry

Pi são apresentadas em detalhes, usando Python, OpenCV e câmera do RPi. Apesar do trabalho não estar voltado a uma língua de sinais, eles projetaram o robô com Raspberry Pi. Neste TCC, os Python e OpenCV no Raspberry são utilizados para se ter visão computacional e o reconhecimento da mão no protótipo Huet.

Um método para tornar o reconhecimento de sinais mais fácil para as pessoas durante a comunicação, que é a conversão dos sinais em texto é apresentado em Golekar *et al.* (2022). Nesse trabalho, os autores capturaram gestos da mão através de uma *webcam* e converteram esta imagem em uma outra em tons de cinza. A segmentação da imagem de um gesto de mão em tons de cinza foi executada usando o algoritmo do método *Otsu*.

Uma arquitetura de redes CNN que utiliza o método *pooling*, *max-pooling* em que o valor máximo está em um local vizinho ao mapa de recursos é apresentado em (PIGOU *et al.*, 2014). Para guardar os dados de vídeo, o *max-pooling* é executado em três dimensões. A arquitetura do modelo consiste em duas CNNs, uma para extração de características manual e outra para extrair as características da parte superior do corpo. Cada CNN tem três camadas profundas. Uma rede neural artificial clássica com uma camada oculta produz uma classificação após concatenar os resultados dessas CNNs.

Como indicado em (REZENDE; CASTRO; ALMEIDA, 2016), não é possível diferenciar os sinais utilizando apenas a expressão facial. O método do reconhecimento na pesquisa envolveu a avaliação das expressões faciais em 10 sinais (*acalmar, acusar, aniquilar, amar, engordar, felicidade, magro, sorte, surpresa e raiva*). Cada sinal foi capturado 10 vezes por um sensor RGB-D. As quatro etapas desse método são: detecção e recorte da região de interesse, resumo do vídeo utilizando o conceito de diversidade maximizada, criação do vetor de características e classificação de sinais através de k-NN. Uma precisão média de mais de 80% foi alcançada no modelo proposto.

Por fim, na Tabela 2, é apresentado o resumo dos trabalhos apresentados nessa seção, destacando a língua de sinais, o método de aquisição de dados, a aplicação, classificação e se o sinal é ou não dinâmico.

Tabela 2 – Tabela comparativa de trabalhos relacionados.

Trabalho	Língua dos sinais	Aquisição de dados	Aplicação	Classificação	Dinâmico
(BASTOS; ANGELO; LOULA, 2015)	Libras	Microsoft HD Lifecam 3000	Estudo do reconhecimento	RNA MLP	Não
(AMARAL <i>et al.</i> , 2017)	Libras	sensor RealSense	Operação de Transformada de Distância	CNN	Não
(COSTA <i>et al.</i> , 2017)	Libras	sensor kinect	Reconhecimento da configurações de mão de Libras	kNN	Sim
(PANSARE; GAWANDE; INGLE, 2012)	ASL	FronTech E-cam	Alfabeto manual	Distância Euclidiana	Sim
(SHENOY <i>et al.</i> , 2018)	ISL	Câmera do celular de Android	Reconhecimento de Língua dos sinais	kNN	Sim
(ABED; RAHMAN, 2017)	-	modulo de camera de RPI	Robótica	Region of Interest	Sim
(GOLEKAR <i>et al.</i> , 2022)	ASL/ ISL	Kinect Microsoft	Reconhecimento de Língua dos sinais	CNN	Sim
(PIGOU <i>et al.</i> , 2014)	ASL/CSL/FSL	Kinect Microsoft	Reconhecimento de Língua dos sinais	CNN	Sim
(REZENDE; CASTRO; ALMEIDA, 2016)	Libras	Kinect	Reconhecimento da expressão facial	kNN	Sim

Fonte: Elaborado pelo autor, 2023.

3 METODOLOGIA

A metodologia proposta tem como base a implementação de algoritmos e utilização dos dispositivos *Raspberry*, ESP32CAM e ESP 01 acoplado ao relé. Os algoritmos em Python e C++ foram criados para ter um reconhecimento da mão e ter duas comunicações sem fio entre RPi e ESP32-CAM e entre RPi e ESP8266 - ESP 01. Ao detectar um letra manual por meio da câmera, o RPi comunica-se com ESP 01 que liga ou desliga o relé conectado à lâmpada. Uma vez que o Protótipo *Huet* emula a visão humana e entendimento dos gestos da Libras e o uso da comunicação sem fio desse protótipo é útil no que se refere à utilização de espaço.

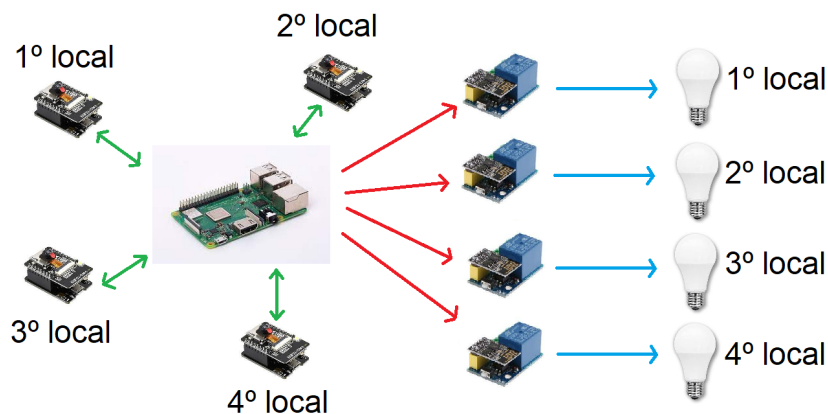
No entanto, esse reconhecimento não se origina de uma base já pronta de informações em Libras, sendo preciso construir as estruturas *if/else* para criar a identificação das letras. Isso permite as comparações entre as variáveis dos pontos na mão. O que possibilita o sistema ser embarcado em uma plataforma microcontrolada.

O protótipo foi chamado de *Huet*, que é homenageado nesse trabalho por incentivar as pessoas a aprenderem sinais, que compõem a principal forma de comunicação entre os surdos no Brasil.

3.1 Arquitetura de comunicação e seus componentes

Uma arquitetura de comunicação sem fio foi criada entre RPi e ESP32-CAM, e uma entre RPi e ESP8266 foi criada conforme apresenta o esquema geral da Figura 25.

Figura 25 – Multiprotótipo em cada local residencial.

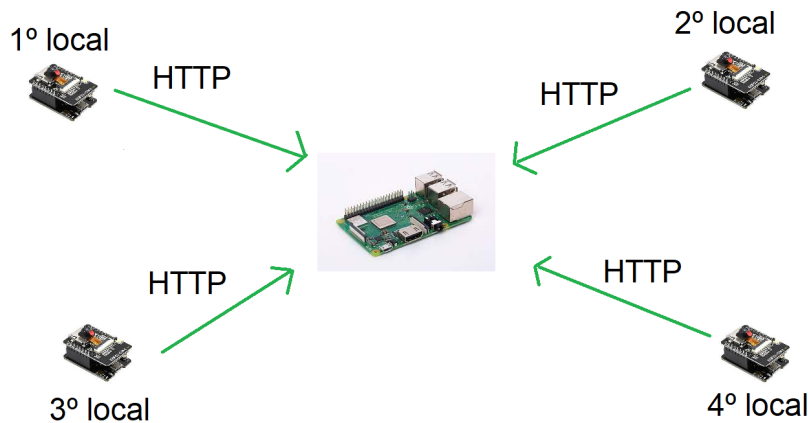


Fonte: Elaborada pelo autor, 2023.

No ponto central, o RPi é escolhido, uma vez que esse microcomputador tem processador 4GB de RAM, Wi-Fi, o nó central conecta-se aos nós descentralizados dos ESP32-CAM, que é capaz de produzir uma imagem para ser processada no ambiente, sem que o usuário precise ir ao ponto central. Esses módulos funcionam como pontos remotos de câmeras espalhadas pela casa

que transmitem as imagens para o nó central de processamento. Além disso, o nó central recebe HTTP de cada nó centralizado para manter a conexão entre eles, como ilustra a Figura 26.

Figura 26 – Nós centralizados enviam HTTPs diferentes para nó central.



Fonte: Elaborada pelo autor, 2023.

Finalmente, o nó central com processamento do RPi recebe HTTP de cada nó descentralizado também, (mesmo sentido da seta da Figura 26). O envio do controle de dados para nós descentralizados dos ESP8266 ocorre quando houver uma identificação de um gesto que está no ESP32-CAM, e, em seguida, um dos módulos ESP8266 atua no relé para acionar uma lâmpada.

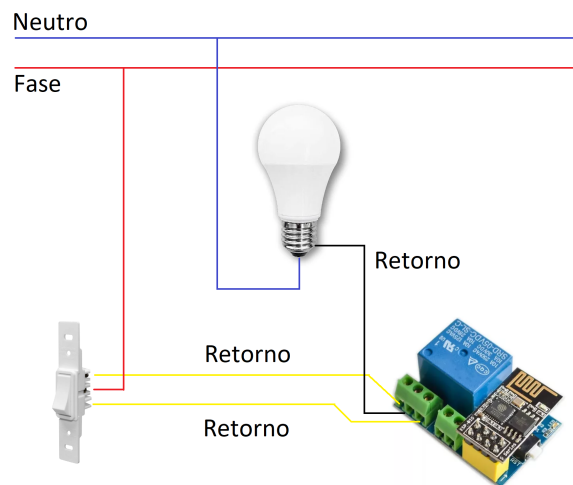
Na recomendação para o uso do protótipo, é necessário instalar um interruptor paralelo (*three way*), como ilustra a Figura 27, para se ter duas opções: acionamento automático ou manual, e também para evitar travamento da lâmpada acesa como *loop* caso o ESP8266 não receba um sinal mais ou o ESP32-CAM não funcione. A fase passa até o interruptor paralelo; as duas ligações entre interruptor e módulo são retornos que entram nos contatos NA e NF do relé e o terceiro retorno passa do comum do relé a lâmpada que é ligada ao neutro.

3.2 Processamento de imagens no Protótipo *Huet*

As imagens capturadas pela câmera OV2640 de 2 megapixels são enviadas do ESP32-CAM para Raspberry Pi via Wi-fi, sendo manipuladas pelas técnicas do processamento de imagens. Essas técnicas implementadas no algoritmo são utilizadas para as imagens ficarem claras, de forma a se ter um objeto detectado através da visão computacional, como *MediaPipe*.

Para capturar um vídeo, é preciso usar a linguagem de programação C++ no módulo ESP32-CAM, pois o *PlatformIO* ou *Arduino IDE* possuem bibliotecas que são colocadas nesse dispositivo, que roda um código para gerar as imagens e o algoritmo com HTTP do ESP32-CAM é rodado pelo RPi para obter um vídeo recebido.

Figura 27 – Interruptor paralelo.



Fonte: Elaborada pelo autor, 2023.

3.3 Python com OpenCV e MediaPipe e Fluxogramas

Tendo em vista a implementação em um sistema com recursos computacionais reduzidos, o código desenvolvido em *Python* possui muitas estruturas condicionais *if/else*. Para se obter uma identificação, utiliza-se *OpenCV*, cujas técnicas de processamento de imagens melhoram a qualidade do vídeo, e o *MediaPipe* permite o processamento do reconhecimento de mão com linhas e pontos.

Com a referência dos pontos na mão, obtido, por exemplo, por meio da Figura 15, a expressão da estrutura *if* da identificação da letra A é representada, como ilustra da Figura 28, $lm_list[PontaIndicador].y$ é o vetor como a variável em *y* da coordenada da imagem, o valor de *PontaIndicador* é igual a 8 como a posição da mão. O outro vetor $lm_list[PontaIndicador - 1].y$ é outra variável em *y* e o de *PontaIndicador - 1* é igual 7, outra posição, vizinha da posição de *PontaIndicador*. Essas variáveis são comparadas pelo algoritmo, com seguir o padrão da coordenada em que *x* é horizontal e seu sentido é de origem para direito e *y*, vertical e seu sentido é de origem para baixo, se $lm_list[PontaIndicador].y$ for maior que $lm_list[PontaIndicador - 1].y$, é verdadeiro. Com isso, outros vetores dos pontos também são comparados para a letra A ser identificada ou não. Além disso, as estruturas *if* das outras letras também possuem os vetores $lm_list[...].x$ e $lm_list[...].y$.

No trecho de código do algoritmo apresentado na Figura 29, por exemplo, quando uma pessoa sinalizar uma letra B para câmera, se o gesto for igual a A, a resposta é não, em seguinte, se gesto for igual a B na outra decisão, a resposta é sim, logo, o algoritmo mostra letra B. No entanto, é a execução rápida e pode ter uma resposta errada quando a mão sinalizada ainda não está pronta e o algoritmo identifica outra letra, pois a mão do usuário surdo parece alguma letra de Libras que é mostrada imediatamente quando o usuário começa com mão livre ou altera a letra manual à câmera. É preciso recriar o algoritmo do segundo teste para imitar uma pessoa

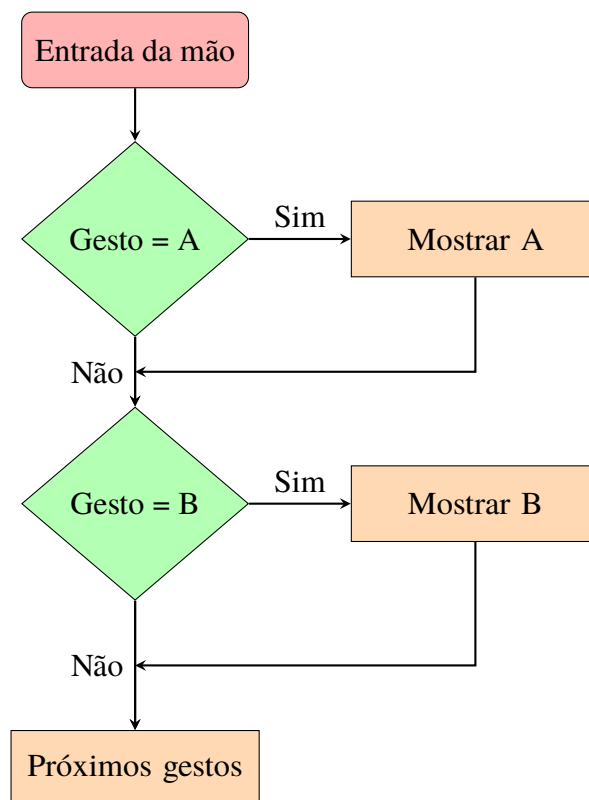
Figura 28 – Decisão da letra A

```
# Símbolo A
inicio = time.time()
if (lm_list[PontaIndicador-2].y < lm_list[PontaIndicador-1].y < lm_list[PontaIndicador].y <
    lm_list[Palma].y) and (lm_list[PontaMedio-2].y < lm_list[PontaMedio-1].y <
    lm_list[PontaMedio].y < lm_list[Palma].y) and (lm_list[PontaAnelar-2].y <
    lm_list[PontaAnelar-1].y < lm_list[PontaAnelar].y < lm_list[Palma].y) and (
    lm_list[PontaMinimo-2].y < lm_list[PontaMinimo-1].y < lm_list[PontaMinimo].y <
    lm_list[Palma].y) and (lm_list[PontaPolegar].y < lm_list[PontaPolegar-1].y <
    lm_list[PontaPolegar-2].y < lm_list[PontaPolegar-3].y < lm_list[Palma].y) and (
    lm_list[PontaMinimo-2].x > lm_list[PontaAnelar-2].x > lm_list[PontaMedio-2].x >
    lm_list[PontaIndicador-2].x > lm_list[PontaPolegar].x) and (lm_list[PontaPolegar].y <
    lm_list[PontaIndicador-2].y) and (lm_list[Palma].x > lm_list[PontaIndicador].x) and (
    lm_list[PontaPolegar-2].y > lm_list[PontaMedio-2].y) and deteccao == 0:
```

Fonte: Elaborada pelo autor, 2023

que entende e responde normalmente na comunicação, como pode ser observado no fluxograma da Figura 30.

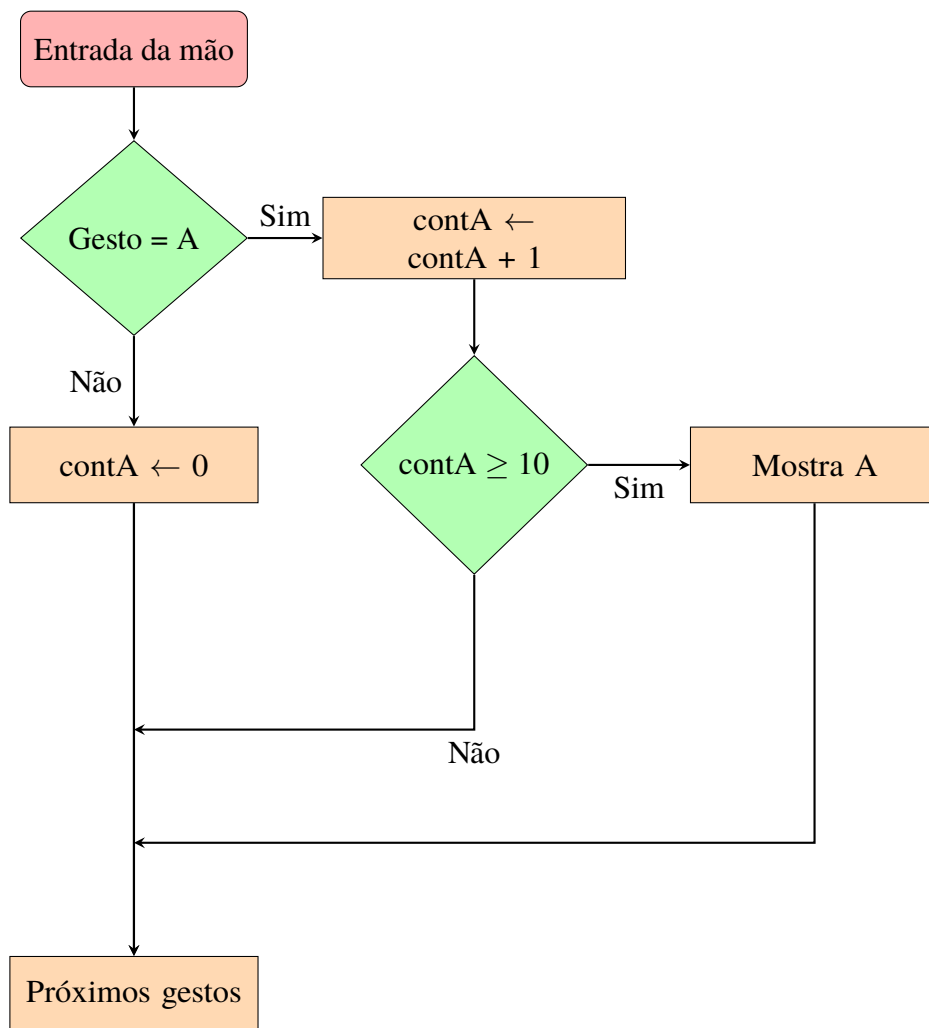
Figura 29 – Fluxograma do primeiro teste



Fonte: Elaborada pelo autor, 2023.

A partir da detecção de uma mão próxima à câmera do ESP-CAM: na primeira decisão, se o gesto for igual à A, a variável *contA* receberá mais um, caso contrário, será zero para *contA*. Na segunda decisão, se *contA* for maior ou igual a 10, o algoritmo mostra A, caso contrário, ainda não confirma que mostra A. Para se ter a confirmação, contando até 10, uma letra surge, como se uma pessoa entendesse um gesto da mão em Libras e respondesse a letra correta.

Figura 30 – Fluxograma do segundo teste



Fonte: Elaborada pelo autor, 2023.

3.4 Sistema do protótipo *Huet*

O sistema do protótipo *Huet* permite o acionamento dos dispositivos eletrônicos por meio de sinais em Libras de um usuário. Resumidamente, a partir da detecção de uma mão, entrada do sistema, capturada pelo módulo ESP32-CAM que transmite esses dados para RPi. Após receber essa informação, o algoritmo identifica o gesto da mão e envia um sinal do RPi para o módulo ESP-01. Assim, o módulo correto ESP-01 liga ou desliga um relé, que conseqüentemente aciona um equipamento, como apresenta a sequência da Figura 31.

Figura 31 – Sistema do protótipo *Huet*



Fonte: Elaborada pelo autor, 2023.

4 RESULTADOS

4.1 Protótipo físico desenvolvido

O Raspberry, ilustrado na Figura 32, executa o código para ter comunicações com dois módulos, ter detecção de mão na automação residencial.

Figura 32 – *Raspberry Pi 4*



Fonte: Autor, 2023

Para capturar imagens, utiliza-se o módulo ESP32-CAM, ilustrado na Figura 33, conectado ao minicomputador via Wi-Fi.

Figura 33 – ESP32-CAM



Fonte: Autor, 2023

Ao detectar o gesto, o módulo ESP8266 recebe um sinal de *Raspberry* e liga o relé para acionar o circuito da iluminação, como demonstrado pela Figura 34. Utiliza-se extensão com soquete para lâmpada ser ligada ao módulo relé e ter o seu acionamento iniciado automaticamente.

4.2 Visão geral do esquema eletrônico

Na Figura 35, os componentes foram montados juntos para implementar um protótipo para automação residencial, utilizando gestos, via Wi-Fi do módulo ESP32-CAM e do módulo

Figura 34 – ESP8266, Relé e Lâmpada



Fonte: Autor, 2023

ESP8266, com relé ligado à lâmpada. Precisou-se utilizar a fonte de alimentação de 5V para cada dispositivo.

Figura 35 – Esquema geral



Fonte: Autor, 2023

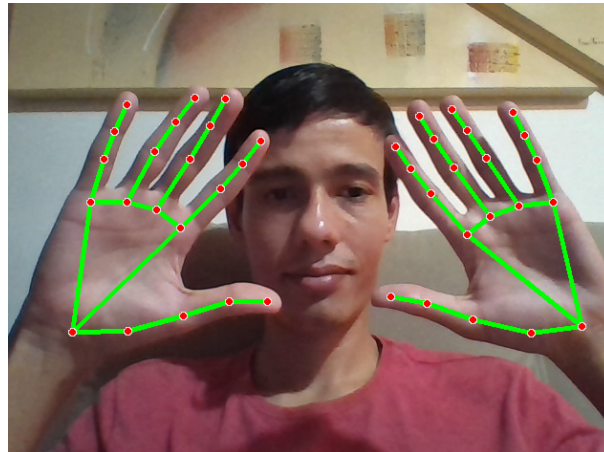
Seria interessante, posicionar cada protótipo *Huet*, por exemplo, na sala, na cozinha, nos quartos. Entretanto, manteve-se *Raspberry* como ponto central na sala. Além disso, pode-se ter um ESP8266 com dois relés em um local com duas lâmpadas.

4.3 Testes da identificação das letras

Como demonstrado na Figura 36, a *OpenCV* gera a imagem e a *MediaPipe* gera os pontos e as linhas nas mãos. O algoritmo usa estruturas de comparação no retorno da Biblioteca, com base na *MediaPipe* para identificar uma letra, de acordo com os valores desejados de posições relativas.

Os vários pontos de referência de uma mão são reconhecidos pela *MediaPipe* em tempo real. A Figura 37 apresenta o resultado desse reconhecimento de mão. A visão computacional mostra os pontos e as linhas e a imagem da letra correspondente, e a imagem da letra A é

Figura 36 – OpenCV e MediaPipe



Fonte: Autor, 2023

mostrada pelo algoritmo no canto superior esquerdo do vídeo, após ser identificada. No entanto, a *MediaPipe* não conhece o alfabeto manual em Libras.

Figura 37 – Identificação da letra A



Fonte: Autor, 2023

Essa tela possui 307200 pixels, 640 na horizontal e 480 na vertical. Os valores das coordenadas (Figura 38), correspondentes aos pontos posicionados na tela, mostram a identificação do gesto A. Conforme ilustra a Figura 28, alguns pontos são colocados em uma estrutura condicional para a identificação da letra A pelo algoritmo.

Mais um resultado da identificação da letra L, como ilustrado na Figura 39, em y , se o valor do 8 for menor que o valor do 7, o valor do 7 for menor que 6, e em x , o valor do 4 for menor que o do 3, mais outras comparações, resultará em uma verdade lógica e o algoritmo mostra letra L.

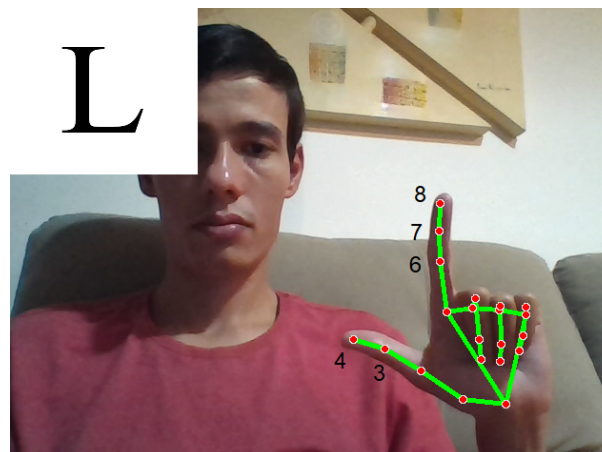
Para fazer o teste, mostra-se o gesto de cada símbolo para a câmera, 10 vezes, para se obter os resultados que informam tempo médio de identificação da letra em Libras (T_m), desvio

Figura 38 – Coordenada de cada ponto na tela

A	10 : (475 , 190)
0 : (518 , 313)	11 : (476 , 236)
1 : (472 , 305)	12 : (480 , 259)
2 : (437 , 256)	13 : (512 , 225)
3 : (429 , 209)	14 : (504 , 202)
4 : (430 , 175)	15 : (501 , 246)
5 : (454 , 222)	16 : (502 , 269)
6 : (450 , 184)	17 : (540 , 231)
7 : (454 , 218)	18 : (530 , 215)
8 : (458 , 245)	19 : (525 , 246)
9 : (484 , 221)	20 : (525 , 264)

Fonte: Autor, 2023

Figura 39 – Indicações do pontos no gesto L



Fonte: Autor, 2023

padrão de tempo (Dp), acertos em porcentagem e símbolo em 10 posições. As letras A, L, M e V significam Alexa, lâmpada, motor e ventilador, respectivamente, e os números 1, 2 e 3, identificação de cada lâmpada.

Após obter o resultado do primeiro teste, como mostra a Figura 40, percebem-se erros em alguns símbolos pela rapidez com que o algoritmo "lê" os valores. O menor acerto, em porcentagem, é do símbolo 2; os tempos médios e desvios resultam em zero, pois rodar o código foi muito rápido.

Após modificação do algoritmo, seria possível ter muitos acertos ao longo do tempo, o que modifica o algoritmo é acrescentar as estruturas condicionais, como no caso do fluxograma do segundo teste representado, Figura 30, no capítulo anterior. Veja o resultado do segundo teste, na Figura 41, não há erros, uma vez que o algoritmo confere uma letra até 10 vezes.

Mas pode haver um defeito instantâneo durante a realização do teste, pois há um tempo médio maior do teste, é do símbolo 3. Isso significa que não foi exatamente resposta em uma vez

Figura 40 – Resultado do primeiro teste

Resultados de cada letra:				
Símbolo	Tm	Dp	Acerto (%)	Sequência
A	0.0	0.0	70	['A', 'A', 'M', 'A', '3', 'M', 'A', 'A', 'A', 'A']
L	0.0	0.0	80	['L', 'L', 'L', 'L', 'L', 'L', 'L', '3', 'L', 'L']
M	0.0	0.0	100	['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M']
V	0.0	0.0	100	['V', 'V', 'V', 'V', 'V', 'V', 'V', 'V', 'V', 'V']
1	0.0	0.0	70	['M', '1', '1', 'A', 'M', '1', '1', '1', '1', '1']
2	0.0	0.0	50	['1', '2', '2', 'L', 'L', 'L', '2', '2', '2', 'L']
3	0.0	0.0	90	['2', '3', '3', '3', '3', '3', '3', '3', '3', '3']

Fonte: Autor, 2023

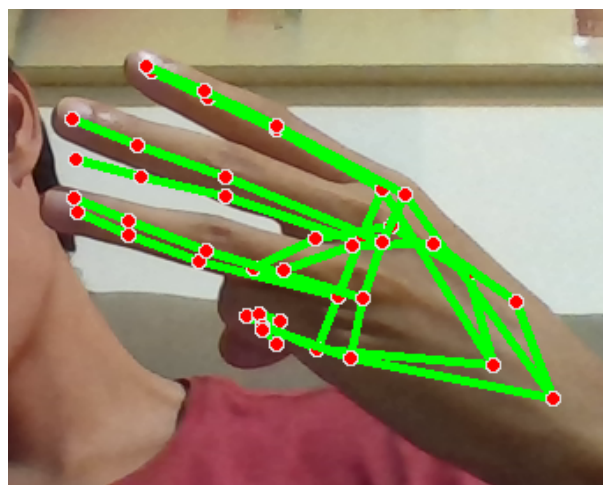
Figura 41 – Resultado do segundo teste

Resultados de cada letra:				
Símbolo	Tm	Dp	Acerto (%)	Sequência
A	1.25	0.08	100	['A', 'A', 'A', 'A', 'A', 'A', 'A', 'A', 'A', 'A']
L	1.32	0.15	100	['L', 'L', 'L', 'L', 'L', 'L', 'L', 'L', 'L', 'L']
M	1.61	0.23	100	['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M']
V	1.30	0.22	100	['V', 'V', 'V', 'V', 'V', 'V', 'V', 'V', 'V', 'V']
1	1.41	0.43	100	['1', '1', '1', '1', '1', '1', '1', '1', '1', '1']
2	1.39	0.22	100	['2', '2', '2', '2', '2', '2', '2', '2', '2', '2']
3	2.56	2.16	100	['3', '3', '3', '3', '3', '3', '3', '3', '3', '3']

Fonte: Autor, 2023

da rodada do algoritmo, visto que os pontos e as linhas foram duplicados na hora do teste, como aponta a Figura 42.

Figura 42 – Pontos e linhas duplicados no gesto do número 3



Fonte: Autor, 2023

Caso a mão sinalizada saia da câmera e a variável *cont* não chegue a 10 vezes, a imagem de uma letra não aparece no canto superior esquerdo do vídeo, não havendo, então, resultado de

letra nenhuma.

Ao contar até 10, o algoritmo confirma a resposta da letra que é desejada pelo usuário. Então, o objetivo disso é evitar os erros no protótipo *Huet*, pois sem modelo do segundo teste no protótipo, lâmpada L2 será ligada quando um usuário sinalizar L1 na sala com duas lâmpadas, isso causa insatisfação para mesmo usuário.

4.4 Funcionamento geral do protótipo

Foram realizados diversos testes utilizando uma lâmpada como equipamento conectado ao relé, podendo ser alterado para o equipamento desejado, apenas modificando o relé, de maneira a suportar a corrente do equipamento. Para testar o protótipo, basta realizar o gesto da mão, em Libras, próximo à câmera do ESP32-CAM, que, por sua vez, transmite a imagem para o *Raspberry*, possibilitando o processamento do algoritmo que "entende" os valores dos pontos da mão que a visão computacional informa. Se todas comparações forem verdadeiras, o sinal é enviado para ESP8266 atuar no relé, que, por fim, ligará ou desligará o equipamento em uma casa de um usuário surdo.

As letras H e L foram escolhidas para significarem *Huet* e lâmpada, respectivamente. Ao sinalizar a letra H, aciona-se o protótipo *Huet*, e, ao sinalizar a letra L, faz-se o comando para que *Huet* ligue ou desligue uma lâmpada. Utiliza-se o código do primeiro teste sem problema, uma vez que ele possui uma lâmpada. A visualização de alguns desses testes de funcionamento geral do protótipo *Huet*, com a utilização de um interruptor paralelo, pode ser acessada no seguinte vídeo: <https://youtu.be/2Dlko_wzaCg>.

Por fim, o código utilizado no protótipo *Huet* está disponível no GitHub<<https://github.com/filipeBrison/TCC-prototipo-huet>>.

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 Conclusão

Este trabalho propôs uma alternativa à barreira de comunicação existente entre um usuário surdo e um dispositivo assistente como a Alexa, por exemplo. O protótipo *Huet* é semelhante ao dispositivo *Echo Dot, Alexa*; no entanto, a diferença entre eles é comunicação. O primeiro possui um microfone para a captura da voz humana e um alto-falante para transmitir as informações aos ouvintes. O segundo, de acordo com a proposta deste trabalho, captura, por meio de uma câmera, os comandos executados pelos gestos de uma pessoa surda.

O protótipo *Huet* é constituído de componentes como a câmera, que atua de modo similar à visão humana, o minicomputador, *Raspberry*, que executa as funções de modo semelhante ao cérebro, e o relé, que funciona por módulo sem fio, ligando ou desligando a lâmpada. Na metodologia utilizada, o *Raspberry*, os módulos ESP32-CAM e ESP8266 – ESP 01, acoplado ao relé, as linguagens de programação *Python* e *C++* e as lâmpadas compõem o dispositivo assistente, compõe um sistema de comunicação eficiente. Com isso, um usuário sinaliza o gesto da mão à uma câmera, realizando os acionamentos automaticamente.

Desta forma, foi feito análises do tempo de resposta da identificação da letra para ver as quantidades de acertos de cada símbolo nos resultados, sendo adequados para a execução real.

Além disso, este trabalho é uma contribuição para fomentar a inclusão social dos surdos por meio da tecnologia, possibilitando a acessibilidade através de um dispositivo assistente semelhante ao *Echo Dot, Google Home* e outros.

5.2 Trabalhos Futuros

Como continuidade do projeto, pretende-se avaliar a realização do reconhecimento da mão por meio da biblioteca *MediaPipe*, de maneira a obter diretamente a identificação de uma letra ou coordenadas de cada ponto. Esses dados poderão ser treinados com a utilização de uma rede neural, por exemplo, a SVM para obter um modelo para as letras, números e sinais de Libras, com a criação de uma base de dados robusta de sinais de Libras.

Considerando o *hardware* proposto, pretende-se adicionar um *display* LCD ou um *tablet* em ponto central para receber a informação, por exemplo, das lâmpadas, ventiladores e cortinas em uma casa grande.

REFERÊNCIAS

- ABED, A. A.; RAHMAN, S. A. Python-based raspberry pi for hand gesture recognition. **International Journal of Computer Applications**, v. 173, n. 4, p. 18–24, 2017. Citado 2 vezes nas páginas 35 e 36.
- ALVES, E. G.; FRASSETTO, S. S. Libras e o desenvolvimento de pessoas surdas. **Aletheia**, n. 46, 2015. Citado na página 17.
- AMARAL, L.; LIMA, G.; VIEIRA, T.; VIEIRA, T. Reconhecimento de gestos estáticos da mão usando a transformada de distância e aplicações em libras. **Universidade Federal de Alagoas**, 2017. Citado 2 vezes nas páginas 35 e 36.
- ARAÚJO, C. C. M.; LACERDA, C. B. F. d. Examinando o desenho infantil como recurso terapêutico para o desenvolvimento de linguagem de crianças surdas. **Revista da Sociedade Brasileira de Fonoaudiologia**, SciELO Brasil, v. 13, p. 186–192, 2008. Citado na página 17.
- ATELIWARE. **Redes Neurais Convolucionais**. 2021. Disponível em: <<https://ateliware.com/blog/redes-neurais-convolucionais>>. Citado 2 vezes nas páginas 27 e 28.
- BASTOS, I. L.; ANGELO, M. F.; LOULA, A. C. Recognition of static gestures applied to brazilian sign language (libras). In: IEEE. **2015 28th SIBGRAPI Conference on Graphics, Patterns and Images**. [S.l.], 2015. p. 305–312. Citado 2 vezes nas páginas 35 e 36.
- BRADSKI, G. **The OpenCV Library**. 2000. Acesso em 16 abr 2023. Disponível em: <<https://www.drdobbs.com/open-source/the-opencv-library/184404319?pgno=1>>. Citado na página 29.
- BRASIL. **Lei nº 10.436 de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais – Libras e dá outras providências**. 2002. <https://www.planalto.gov.br/ccivil_03/leis/2002/110436.htm>. Acesso em 8 abr. 2023. Citado na página 18.
- BRASIL, A. Oms estima 2,5 bilhões de pessoas com problemas auditivos em 2050. **Agência Brasil**, 3 2021. Disponível em: <<https://agenciabrasil.ebc.com.br/saude/noticia/2021-03/oms-estima-25-bilhoes-de-pessoas-com-problemas-auditivos-em-2050>>. Citado na página 14.
- CAMPINAS, U. de. **Processamento e Análise de Imagens**. 2012. <<https://docplayer.com.br/6324057-Processamento-e-analise-de-imagens-mc940-analise-de-imagens-mo445.html>>. Citado na página 23.
- CARVALHO, A. de; COLLABORATORS. **Neural Networks**. 2021. Acesso em 15 abr 2023. Disponível em: <<https://sites.icmc.usp.br/andre/research/neural/>>. Citado na página 20.
- COSTA, C. F. F.; SOUZA, R. S. d.; SANTOS, J. R. d.; SANTOS, B. L. d.; COSTA, M. G. F. A fully automatic method for recognizing hand configurations of brazilian sign language. **Research on Biomedical Engineering**, SciELO Brasil, v. 33, p. 78–89, 2017. Citado 2 vezes nas páginas 35 e 36.
- CURTOCIRCUITO. **Módulo Rele 1 Canal Com Optoacoplador - 5v/10a**. 2023. Disponível em: <<https://curtocircuito.com.br/modulo-rele-1-canal-com-optoacoplador-5v-10a.html>>. Citado na página 34.
- CURTOCIRCUITO. **Módulo WiFi ESP8266 ESP-01**. 2023. Disponível em: <<https://curtocircuito.com.br/modulo-wifi-esp8266-esp-01.html>>. Citado na página 32.

DEEP LEARNING BOOK. **Capítulo 41 - Campos Receptivos Locais em Redes Neurais Convolucionais**. 2022. <<https://www.deeplearningbook.com.br/campos-receptivos-locais-em-redes-neurais-convolucionais/>>. Acesso em 8 abr 2023. Citado na página 27.

DEEP LEARNING BOOK. **Capítulo 63 - Aplicações da Aprendizagem por Reforço no Mundo Real**. 2022. <<https://www.deeplearningbook.com.br/aplicacoes-da-aprendizagem-por-reforco-no-mundo-real/>>. Acesso em 10 jun 2023. Citado na página 26.

DIDÁTICA TECH. **Dados de Treino e Teste**. 2022. <<https://didatica.tech/dados-de-treino-e-teste/>>. Acesso em 10 jun 2023. Citado na página 25.

DIGITAL, S. **O que é Visão Computacional e para que serve?** 2021. Disponível em: <<https://santodigital.com.br/o-que-e-visao-computacional-e-para-que-serve/>>. Citado na página 24.

ELÉTRICA, M. da. **Instalações Elétricas: O que são?** 2021. Disponível em: <<https://www.mundodaeletrica.com.br/instalacoes-eletricas-o-que-sao/>>. Citado 2 vezes nas páginas 33 e 34.

FERNANDES, L.; RIBEIRO, L. **Interface Mediapipe Hands e Arduino**. 2021. Acesso em 6 mai 2023. Disponível em: <<https://embarcados.com.br/interface-mediapipe-hands-e-arduino/>>. Citado na página 29.

FOXFLY. **Processamento de imagens e visão computacional: saiba a diferença**. 2021. Acesso em 15 abr 2023. Disponível em: <<https://foxfly.com.br/computer-vision/processamento-de-imagens-e-visao-computacional-saiba-a-diferenca/>>. Citado na página 24.

FREITAS, E. J. R.; SANTOS, S. S.; REZENDE, T. M. Redes neurais artificiais: Uma visão histórica. In: GUIMARAES, O. S. (Ed.). **Engenharia, Gestão e Inovação**. Belo Horizonte: Editora Poison, 2022. v. 1, cap. 15, p. 123–138. Citado na página 20.

GOLEKAR, D.; BULA, R.; HOLE, R.; KATARE, S.; PARAB, S. Sign language recognition using python and opencv. 2022. Citado na página 36.

GONZALEZ, R.; WOODS, R. E. Processing. **New Jersey: Upper saddle river**, p. 7458, 2002. Citado na página 22.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000. Citado na página 22.

IMOBILIS. **Fundamentos de Redes Neurais**. 2017. Acesso em 8 abr 2023. Disponível em: <<http://www2.decom.ufop.br/imobilis/fundamentos-de-redes-neurais/>>. Citado 2 vezes nas páginas 20 e 21.

LIMA, D. *et al.* Educação infantil: saberes e práticas da inclusão: dificuldades de comunicação e sinalização: surdez. **Natal-RN2006**, 2006. Citado na página 17.

MARENGONI, M.; STRINGHINI, S. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, p. 125–160, 2009. Citado na página 29.

MINISTÉRIO DA SAÚDE. **Surdez - Causas**. 2023. Acesso em 18 mar 2023. Disponível em: <<https://bvsmis.saude.gov.br/surdez-3/#:~:text=%E2%80%9320a%20surdez%20de%20c%C3%B3clea%20ou,problemas%20metab%C3%B3licos%2C%20tumores.>> Citado na página 14.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003. Citado na página 24.

MUNCINELLI, S. E. Libras: língua brasileira de sinais. **Extensão em Foco (ISSN: 2317-9791)**, v. 1, n. 1, p. 27–33, 2013. Citado 2 vezes nas páginas 17 e 20.

NETTO, A. V.; ALMEIDA, O. C. P. D.; NETO, J. D. E. S. B. **Extração de características de imagens do olho humano**. [S.l.]: São Carlos: Universidade de São Paulo. Instituto de Ciências Matemática e de ..., 2004. Citado na página 23.

OLIVEIRA, C. d. Q. Digital processing methodology applied to exploring of radiological images; metodologia de processamento digital aplicada a exploracao de imagens radiologicas. 2004. Citado na página 23.

ORACLE. **What is Deep Learning?** 2023. Acesso em 01 abr 2023. Disponível em: <<https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-deep-learning/>>. Citado 2 vezes nas páginas 26 e 27.

PANSARE, J. R.; GAWANDE, S. H.; INGLE, M. Real-time static hand gesture recognition for american sign language (asl) in complex background. Scientific Research Publishing, 2012. Citado 2 vezes nas páginas 35 e 36.

PIGOU, L.; DIELEMAN, S.; KINDERMANS, P.-J.; SCHRAUWEN, B. Sign language recognition using convolutional neural networks. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 572–578. Citado na página 36.

REZENDE, T. M.; CASTRO, C. L. de; ALMEIDA, S. G. M. An approach for brazilian sign language (bsl) recognition based on facial expression and k-nn classifier. In: **29th SIBGRAP, Workshop on Face Processing Applications, on Proceedings**. [S.l.: s.n.], 2016. p. 1–2. Citado na página 36.

SAKURAI, R. **Redes Neurais Convolucionais com MapReduce**. 2017. Acesso em 01 mai 2023. Disponível em: <<https://www.sakurai.dev.br/cnn-mapreduce/>>. Citado na página 28.

SHENOY, K.; DASTANE, T.; RAO, V.; VYAVAHARKAR, D. Real-time indian sign language (isl) recognition. In: IEEE. **2018 9th international conference on computing, communication and networking technologies (ICCCNT)**. [S.l.], 2018. p. 1–9. Citado 2 vezes nas páginas 35 e 36.

SURDA, C. **Comunidades Surdas**. 2018. Acesso em 06 mai 2023. Disponível em: <<https://culturasurda.net/comunidades-surdas/>>. Citado na página 14.

TIPS, R. **Is Raspberry Pi a Microcontroller?** 2023. <<https://raspberrytips.com/is-raspberry-pi-a-microcontroller/>>. Acesso em: 19 de junho de 2023. Citado na página 31.

UNICEF. **17,5 milhões de crianças não estão matriculadas na educação infantil**. 2019. Acesso em 18 mar 2023. Disponível em: <<https://www.unicef.org/brazil/comunicados-de-imprensa/175-milhoes-de-criancas-nao-estao-matriculadas-na-educacao-infantil>>. Citado na página 14.

UNIDAS, N. **OMS estima 2,5 bilhões de pessoas com problemas auditivos em 2050**. 2020. Acesso em 18 mar 2023. Disponível em: <[https://news.un.org/pt/story/2020/03/1705931#:~: text=Em%20todo%20o%20mundo%2C%20466,de%20audi%C3%A7%C3%A3o%20nesse%20mesmo%20n%C3%ADvel.](https://news.un.org/pt/story/2020/03/1705931#:~:text=Em%20todo%20o%20mundo%2C%20466,de%20audi%C3%A7%C3%A3o%20nesse%20mesmo%20n%C3%ADvel.)> Citado na página 14.

USINAINFO. Programando esp32-cam wifi com esp32-cam-mb. **UsinaInfo Blog**, 2022. Disponível em: <<https://www.usinainfo.com.br/blog/programando-esp32-cam-wifi-com-esp32-cam-mb/>>. Citado na página 31.

USP, J. da. **Dicionário da Língua de Sinais exigiu 25 anos de pesquisas**. 2018. Disponível em: <<https://jornal.usp.br/cultura/dicionario-da-lingua-de-sinais-exigiu-25-anos-de-pesquisas/>>. Citado na página 18.

VENTURIERI, A.; SANTOS, J. R. dos. Técnicas de classificação de imagens para análise de cobertura vegetal. In: ASSAD, ED; SANO, EE (Ed.). Sistema de informações geográficas ..., 1998. Citado na página 23.

XPEDUCAÇÃO. **Redes Neurais Artificiais**. 2022. <<https://blog.xpeducacao.com.br/redes-neurais-artificiais/>>. Acesso em 19 jun 2023. Citado na página 20.