



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

HOFFMANN DA SILVA QUEIFER

**RECONHECIMENTO DE GESTOS PARA ACIONAMENTO DE OBJETOS
USANDO CNN E O ALGORITMO HAAR CASCADE**

**PATOS
2019**

HOFFMANN DA SILVA QUEIFER

**RECONHECIMENTO DE GESTOS PARA ACIONAMENTO DE OBJETOS
USANDO CNN E O ALGORITMO HAAR CASCADE**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Área de concentração: Internet das Coisas.

Orientadora: Prof. Dra. Jannayna Domingues Barros Filgueira.

**PATOS
2019**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

Q3r Queifer, Hoffmann da Silva.

Reconhecimento de gestos para acionamento de objetos usando CNN e o algoritmo haar cascade [manuscrito] / Hoffmann da Silva Queifer. - 2019.

56 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas , 2019.

"Orientação : Prof. Dr. Jannayna Domingues Barros Filgueira , Coordenação do Curso de Computação - CCEA."

1. Visão computacional. 2. Internet das coisas. 3. Redes Neurais Convolucionais. 4. Reconhecimento de gestos. I. Título

21. ed. CDD 005.1

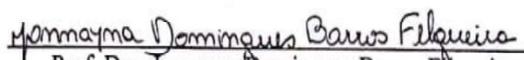
Hoffmann da Silva Queifer

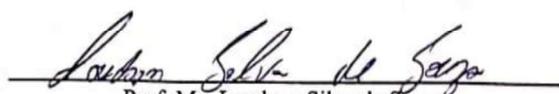
**RECONHECIMENTO DE GESTOS PARA ACIONAMENTO DE OBJETOS USANDO
CNN E O ALGORITMO DE HAAR CASCADE**

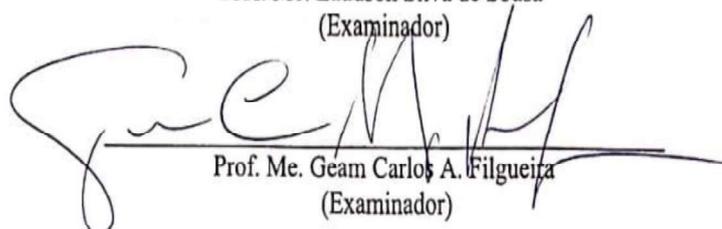
Trabalho de Conclusão de Curso apresentado ao
Curso de Bacharelado em Ciências da
Computação da Universidade Estadual da
Paraíba, em cumprimento à exigência para
obtenção do grau de Bacharel em Ciência da
Computação.

Aprovado em 25/11/2019

BANCA EXAMINADORA


Prof. Dra. Jannayna Domingues Barros Filgueira
(Orientadora)


Prof. Me. Laudson Silva de Sousa
(Examinador)


Prof. Me. Geam Carlos A. Filgueira
(Examinador)

Dedico este trabalho a minha mãe por sempre está me apoiando em tudo, seja nas alegrias ou nas dificuldades com muito companheirismo e amor.

AGRADECIMENTOS

Primeiramente agradeço a Deus por sempre me proporcionar bênçãos em minha vida e força para vencer cada batalha.

A minha mãe Rita pelo apoio e incentivo por toda vida seja acadêmica ou pessoal, me proporcionando alegrias, companheirismo, carinho e amor.

A minha família pelo apoio e incentivo em cada objetivo que pretendo conquistar seja ele qual for.

A minha amiga Caroline por sempre está ao meu lado demonstrando apoio, incentivo e conselhos seja elas expressadas em forma de palavras, carinho, companheirismo e amor.

Aos meus amigos Ricardo, Kaique, Frankly, Iuri, Ian, Gian, Giuan, Luis, Diego e faço devoto aos demais amigos por sempre estarem presentes, me ajudando seja qual for a situação, com opiniões, conselhos e compartilhando experiências de suas vidas seja elas de alegrias ou dificuldades.

A minha professora e orientadora Dra. Jannayna por sempre estar ao meu lado juntamente com meus colegas incentivando em cada projeto e trabalhos a se realizar, pelas opiniões, conselhos com um propósito de querer sempre meu melhor.

Por fim, aos meus colegas e professores que conheci na instituição, pelos momentos de amizade e apoio na minha vida acadêmica.

A todos o meu muito obrigado.

RESUMO

Diante do crescimento contínuo da tecnologia, o computador e os novos dispositivos estão cada vez mais interativos com o ser humano. Tarefas comuns do dia a dia estão sendo realizadas pelos *hardwares* de forma simples e dinâmica por meio de uma rede de computadores. Um exemplo disso são os *hardwares* Raspberry PI e a microcontroladora NodeMCU. A visão computacional busca interpretar os cenários em imagens capturadas por câmeras. Dentre as técnicas de visão computacional está o algoritmo *Haar Cascade*, e as *Redes Neurais Convolucionais* – CNN, ambos escolhidos para esta pesquisa. A Internet das Coisas refere-se à interconexão de objetos com a Internet, de modo que objetos conectados a uma rede possam transmitir dados. Com intuito de unir duas tecnologias, a visão computacional e a Internet das Coisas tem como característica principal a interação com os usuários. Dessa forma, a pesquisa apresenta um estudo sobre visão computacional e Internet das Coisas, com o propósito acionar objetos a partir da detecção de gestos manuais. Para isso foi utilizado uma Raspberry PI com uma câmera integrada, um notebook, uma NodeMCU e módulos relés para ação final. Os experimentos realizados acionaram uma lâmpada e um ventilador de forma satisfatória.

Palavras-Chave: Visão Computacional. Internet das Coisas. Reconhecimento de Gestos. Objetos.

ABSTRACT

As technology continues to grow, the computer and new devices are increasingly interactive with humans. Common everyday tasks are being performed by hardware simply and dynamically over a computer network. An example of this are the Raspberry PI hardware and the NodeMCU microcontroller. Computer vision seeks to interpret scenarios in images captured by cameras. Among the computer vision techniques is the Haar Cascade algorithm, and the Convolutional Neural Networks - CNN, both chosen for this research. The Internet of Things refers to the interconnection of objects with the Internet so that objects connected to a network can transmit data. In order to unite two technologies, the computer vision and the Internet of Things has as its main feature the interaction with users. Thus, the research presents a study on computer vision and Internet of Things, with the purpose of triggering objects from the detection of manual gestures. For this a Raspberry PI with an integrated camera, a notebook, a NodeMCU and relay modules for final action was used. The experiments performed activated a lamp and a fan satisfactorily.

Keywords: Computer vision. Internet of Things. Gesture Recognition. Objects.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplos de gestos de mãos	13
Figura 2 – Série de gesticulações	14
Figura 3 – Demonstração de gestos em forma mimica	14
Figura 4 – Alguns exemplos de máscaras de Haar Features	16
Figura 5 – Máscaras de Haar Features sendo aplicado nas imagens	16
Figura 6 – Resultado da aplicação do Haar Cascade na imagem	18
Figura 7 – Campos que ligam a visão computacional	19
Figura 8 – Processos de uma visão computacional	20
Figura 9 – Veículo autônomo	21
Figura 10 – Exemplo de uma performance de um vídeo	22
Figura 11 – Processos de digitalização	23
Figura 12 – Grupo de Neurônios	24
Figura 13 – Arquitetura de uma CNN	25
Figura 14 – Processo de camada de pooling	27
Figura 15 – Aplicações da IoT	29
Figura 16 – Módulos	32
Figura 17 – Raspberry PI 3B com sua câmera acoplada	36
Figura 18 – Módulo NodeMCU	37
Figura 19 – Módulo Relé de um canal	38
Figura 20 – Fluxograma da lógica dos testes realizados na pesquisa	40
Figura 21 – Detecção de gestos com acionamento de objetos em tempo real ...	41
Figura 22 – Imagens de detecção de gestos	43
Figura 23 – Imagens de detecção de gestos usando CNN	47
Figura 24 – Fluxograma passo a passo desde a conexão entre ambos os hardwares até o acionamento dos objetos	49
Figura 25 – Detecção de gesto em tempo real	50
Figura 26 – Hardwares utilizados na pesquisa	51

LISTA DE GRÁFICOS

Gráfico 1 – Resultados da detecção de gestos usando Haar Cascade com braço esquerdo	42
Gráfico 2 – Resultados da detecção de gestos usando Haar Cascade com braço direito	43
Gráfico 3 - Resultados da detecção com Haar Cascade usando braço esquerdo no notebook	44
Gráfico 4 - Resultados da detecção com Haar Cascade usando braço direito no notebook	44
Gráfico 5 - Resultados da comparação entre as detecções realizadas	45
Gráfico 6 - Resultados da detecção de gestos usando CNN	46

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Objetivo Geral	11
1.1.1 <i>Objetivos Específicos</i>	12
1.2 Justificativa	12
2 REFERENCIAL TEÓRICO	13
2.1 Gestos	13
2.2 Algoritmos Utilizados na Detecção de Gestos e Movimento	15
2.2.1 <i>Haar Cascade</i>	15
2.3 Visão Computacional	18
2.4 Processamento de Imagens	21
2.5 Redes Neurais Artificiais	23
2.5.1 <i>Redes Neurais Convolucionais</i>	24
2.6 Internet das coisas	27
2.7 Raspberry PI	31
2.8 Módulo Wi-Fi ESP8266 NodeMCU ESP-12	31
2.9 Módulos	32
2.10 Trabalhos Relacionados	32
3 METODOLOGIA	35
3.1 Materiais Utilizados	35
3.1.1 <i>Raspberry PI</i>	35
3.1.2 <i>Módulo Wi-Fi ESP8266 NodeMCU ESP-12</i>	36
3.1.3 <i>Módulos</i>	37
3.1.4 <i>MQ Telemetry Transport</i>	38
4 RESULTADOS E DISCUSSÕES	39
4.1 Detecção de gestos e resultados de sua classificação	39
4.2 Detecção de gestos utilizando o Haar Cascade	42
4.3 Comparativos das detecções	45
4.4 Detecção de gestos utilizando CNN por meio de webcam	46
4.5 Conexão entre Raspberry PI e a NodeMCU	47
5 CONCLUSÃO E TRABALHOS FUTUROS	52
5.1 Trabalhos Futuros	52
REFERÊNCIAS	53
ANEXO A - COMANDOS DE CONFIGURAÇÕES E INSTALAÇÃO DO BROKER 56	

1 INTRODUÇÃO

A tecnologia aproxima-se do homem de diferentes formas, uma delas é com o desenvolvimento de equipamentos que propagam informações importantes de forma mais rápida, assim resultando em um número crescente de dispositivos com uma maior desenvoltura, interação, comodidade e melhorias em diversas áreas.

O computador está cada vez mais presente na vida das pessoas para auxiliá-las em diversas tarefas. Contudo, o seu manuseio sempre esteve condicionado ao uso de algum dispositivo físico, restringindo a sua forma de interação. À medida que a tecnologia evoluiu, surgiu a necessidade de obter uma interação mais eficaz entre o homem e a máquina. Desde a criação do mouse como principal meio de interação gráfica, foram explorados novos paradigmas para tornar a comunicação entre o homem e a máquina tão natural quanto a interação entre pessoas (WIGDOR; WIXON, 2011 apud OGIELA; HACHAJ, 2014).

As diferentes formas de interação existentes na comunicação entre os seres humanos, como fala e gestos, foram utilizadas para aprimorar a área de Interação Homem-Computador (IHC), constituindo uma sub-área desta, chamada Interface Natural do Usuário (INU). A INU explora os diversos tipos de comportamento inerente ao corpo humano (reconhecimento de voz, de gestos, de ações, rastreamento do olhar, entre outros) que podem ser aproveitados para o controle direto de diferentes dispositivos (computador, consoles de jogos, celulares) em diferentes áreas, desde entretenimento até a área médica.

Algumas técnicas podem ser utilizadas para auxiliar nesta interação entre o homem e o computador, como por exemplo a visão computacional. Define-se visão computacional como a ciência responsável em dar visão a uma máquina, pela forma como o computador distingue à sua volta, extraindo dados significativos a partir de imagens capturadas por câmeras de vídeo, sensores, scanners, entre outros dispositivos. Estas informações permitem reconhecer, manipular e detectar os objetos que compõem uma imagem, fazendo com que a máquina interprete seu meio, de modo que os dispositivos consigam realizar tarefas que até os humanos não conseguem fazer (MILANO; HONORATO, 2011 apud ROCHA 2013).

Um conceito tecnológico que auxilia essa comunicação entre homem e máquina é o que chamamos de Internet das Coisas do inglês *Internet of Things* (IoT). Trata-se de como os objetos físicos estão conectados e como eles interagem

entre si e com os usuários. A Internet das Coisas surgiu dos avanços de várias áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. De fato, a IoT tem recebido bastante atenção, devido ao seu potencial de uso nas mais diversas áreas das atividades humanas. Essa tecnologia é capaz de impor funções a dispositivos através de comandos programados por uma determinada linguagem e qualquer ambiente de desenvolvimento integrado compatível, uma plataforma que executa esses comandos feitos por humanos com o hardware. Com isso a IoT tem o propósito de desenvolver dispositivos que se comunicam através da rede de computadores ou via *bluetooth* para impor práticas que são feitas por humanos e que hoje pode ser feita também pelos dispositivos, como por exemplo relógios inteligentes dizem os batimentos cardíacos, distância percorrida por trajeto realizado, dispositivos que se interage com os seres humanos de forma intuitiva e dinâmica lhe comprovando que tudo pode está na palma de sua mão ou próximo de você com poucos cliques (SANTOS, et al, 2016).

Segundo Guedes (2017) a utilização de gestos na comunicação entre indivíduos é uma forma de comunicação muito utilizada em qualquer cultura globalmente, mesmo quando esse tipo de comunicação não é expressamente direta, como no caso das linguagens de sinais. Por ser uma forma de transmitir informação naturalmente humana, com o aumento da interatividade dos dispositivos e a busca por interfaces para a interação humano-computador, os gestos têm sido cada vez mais explorados para cobrir essa necessidade. É utilizando essa comunicação que pode facilmente realizar tarefas para expressar mais comodidade evitando uma locomoção para um determinada atividade como ligar uma luz, um ar condicionado, uma tv e entre outras tarefas usando uma detecção de gesto a um dispositivo previamente treinado para seu objetivo final.

1.1 Objetivo Geral

O objetivo geral deste trabalho acadêmico é desenvolver um protótipo para aquisição e classificação de sinais emitidos por gestos de mãos utilizando uma câmera conectada a um computador de baixo custo Raspberry PI, a fim de acionar objetos.

1.1.1 *Objetivos Específicos*

- Estudar a visão computacional, algoritmos de detecção, hardwares específicos e IoT;
- Aplicar técnicas de visão computacional para o reconhecimento de mãos e gestos;
- Adaptar algoritmos para detecção de gestos;
- Realizar comparações dos resultados dos algoritmos;
- Desenvolver um protótipo para detecção de gestos de mãos;
- Aplicar o algoritmo de melhor desempenho neste protótipo;
- Realizar conexão entre Raspberry PI e a NodeMCU;
- Realizar experimentos para acionar objetos;
- Realizar experimentos de validação de acionamento de objetos de acordo com gesto detectado.

1.2 Justificativa

O desenvolvimento de pesquisas na área da visão computacional e a IoT está em constante expansão. Essas tecnologias têm o intuito de ajudar a simplificar tarefas realizadas por humanos ou realizar tarefas que humanos não conseguem fazer. O intuito dessas tecnologias é aumentar a interação e comodidade, gerar soluções para diversos campos como na área de saúde, segurança, transporte e lazer. Com isso, esta pesquisa propõe desenvolver um protótipo para reconhecimento de gestos de mãos de modo que auxilie os usuário a realizar determinadas tarefas, como acionamento de objetos. Para isto utilizaremos técnicas de visão computacional e conceitos de Internet das Coisas.

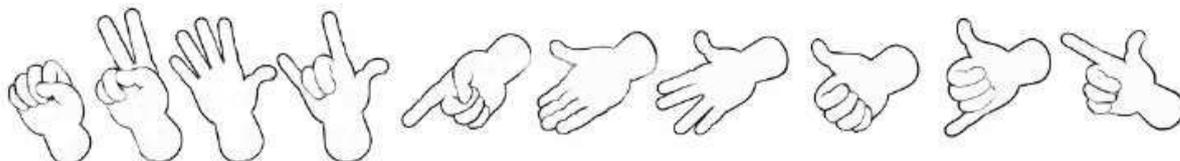
2 REFERENCIAL TEÓRICO

2.1 Gestos

Define-se gestos como sendo um movimento de um braço ou uma mão especificamente com a finalidade de mostrar ideias ou sinais. Eles podem variar de simples ações como apontar um objeto, até movimentos mais difíceis que permitem comunicações entre pessoas. Gestos simples pode expressar mensagens comuns e com significado amplamente conhecidos de forma independente da linguagem falada, como confirmações, negações e saudações.

Um gesto de mão pode ser classificado como estático ou dinâmico. Um gesto estático é um gesto de postura fixa e de possível significado sem precisar de movimentação, representado por uma simples imagem. Um gesto dinâmico é um gesto em movimento, precisa-se desse movimento para completar seu significado, representado por uma sequência de imagens (RIBEIRO, 2006). A Figura 1 ilustra alguns exemplos desses gestos.

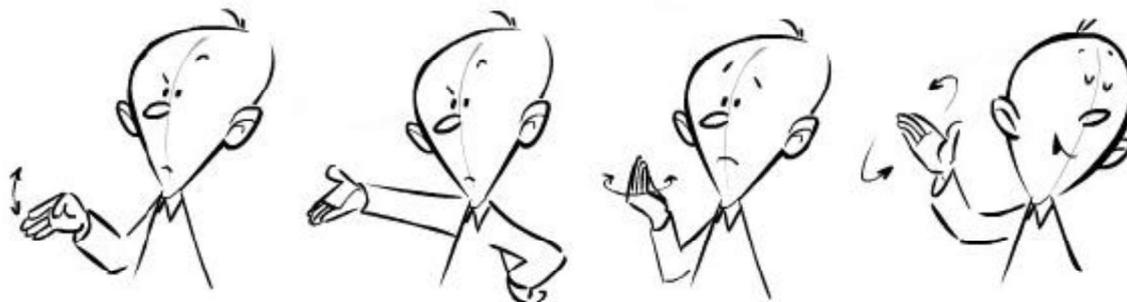
Figura 1 - Exemplos de gestos de mãos.



Fonte: (NETO, 2014).

Gesticulações consiste em um tipo de gesto mais comum entre os seres humanos, sendo expresso de forma espontânea ou inconsciente. Fazemos por algum tipo de hábito ou de forma quando expressamos algum tipo de nervosismo. Como por exemplo, em uma ligação por telefone ou em uma apresentação escolar ou a trabalho. A Figura 2 ilustra uma série de gesticulações que são realizadas de modo contínuo entre os seres humanos.

Figura 2 - Série de gesticulações.



Fonte: (NETO, 2014).

É por meio dos gestos que podemos falar sobre pantomimas, que são gestos feitos com intuito de outras pessoas tentar adivinhar, assim tornando uma atividade que exercita a sua mente de forma positiva e alegre conhecida popularmente como mímica. A Figura 3 ilustra uma série de gestos demonstrando que é um tipo de mímica.

Figura 3 - Demonstração de gestos em forma de mímica.



Fonte: (NETO,2014).

Libras ou Língua Brasileira de Sinais, é o nome dado ao conjunto de gestos providos de um treinamento que representa palavras ou coisas importantes do cotidiano para pessoas com deficiência auditiva que não consegue ouvir e nem falar. Com isso Libras são movimentos específicos realizados pelas palmas da mão, e dependem de um ponto ou espaço de localização em que esses sinais sejam realizados, pois, como toda língua, também precisa ser padronizada e isso acontece através de alguns parâmetros traçados para que todos realizem e possam compreender uns aos outros. Gestos e libras são sinais que estão no cotidiano do

nosso país para tornar mais uma língua ou forma de comunicação bastante intuitiva no dia a dia e também para quebrar o pouco de preconceito que existe entre pessoas com esse tipo de deficiência (DIZEU; CAPORALI, 2005).

Desenvolver mecanismos tecnológicos utilizando gestos seria algo considerado exemplar. Isso daria a função de gestos uma importância a mais e talvez até servisse de incentivo para pessoas praticassem libras e a desenvolvessem mais ideias sobre o mesmo com intuito de ajudar deficientes do modo em geral.

2.2 Algoritmos utilizados na detecção de Gestos e Movimento

2.2.1 Haar Cascade

Algoritmo desenvolvido por Paul Viola e Michel Jones em 2001, o Haar Cascade é considerado um eficiente algoritmo de busca de objetos em imagens. Denominado algoritmo de Viola-Jones ele tem o processo de aprendizagem em imagens positivas e imagens negativas. Classifica-se como imagens positivas aquelas que possuem o objeto a ser classificado e imagens negativas aquelas que não possuem o objeto que se deseja classificar. O mesmo realiza o aprendizado de máquina onde os classificadores são treinados a fim de obter e armazenar características em cascata dos objetos desejados.

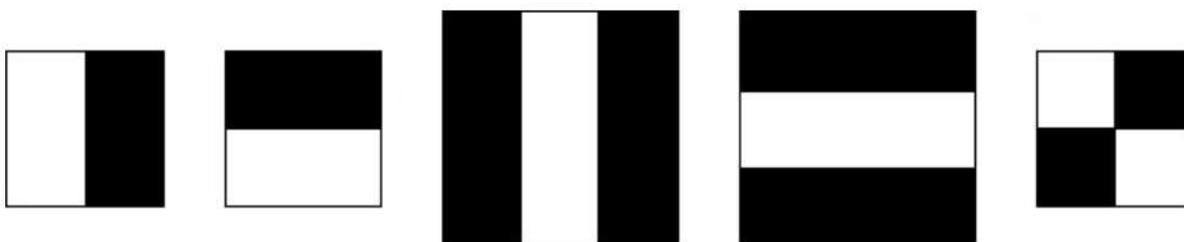
O Haar Cascade extrai as características do objeto desejado a partir do treinamento de aprendizagem de máquina chamado AdaBoost. Neste processo uma grande quantidade de imagens positivas e negativas são aplicadas ao algoritmo gerando uma cascata de características do objeto desejado no formato XML.

A técnica utilizada no classificador consiste em uma análise de imagens em tons de cinza, em que as diferenças de intensidade de regiões distintas são comparadas a características armazenadas em uma árvore de decisões, que é desenvolvida através de um algoritmo de treinamento e um conjunto de imagens das quais são extraídas características específicas do objeto a ser rastreado (OLIVEIRA, 2018, apud PAVAN; MODESTO, 2010).

Com o propósito de extrair características do objeto de interesse, o Haar Cascade utiliza máscaras denominadas *Haar Features*. Estas funcionam como filtros que percorrem um conjunto de imagens, que representam o objeto de interesse

segmentado, efetuando operações para extrair características do objeto (BARELLI, 2018). A Figura 4 ilustra algumas dessas máscaras.

Figura 4 - Alguns exemplos de máscaras de Haar Features.



Fonte: (BARELLI, 2018).

A Figura 5 ilustra a convolução a partir das máscaras apresentadas na figura anterior, do processo de extração de características em imagens segmentadas dos veículos.

Figura 5 - Máscaras de Haar Features sendo aplicado nas imagens.



Fonte: (BARELLI, 2018).

As máscaras de Haar Features sobrepõem o objeto de importante valor. Com isso, cada uma delas efetua a subtração da soma dos pixels sobrepostos pelo retângulo branco da soma dos pixels sobrepostos pelo retângulo preto. O resultado dessa operação é uma característica do objeto representado na imagem. Na prática, o que define as características do objeto é a variação de luminosidade e intensidade do tom de cinza dos pixels. As máscaras apresentadas na figura anterior capturam

esses resultados, permitindo que o objeto seja detectado de forma ágil e precisa (BARELLI, 2018).

Vale salientar que uma variação de luminosidade pode alterar no resultado final de uma possível detecção. Também é válido ressaltar que o Haar Cascade não é um classificador e sim um algoritmo para extração de características e detecção de objetos em imagens (BARELLI, 2018).

O Haar Cascade utiliza, junto com XML produzido, o método *detectMultiScale*. Este requer quatro parâmetros obrigatórios. O primeiro deles é a imagem, convertida em tons de cinza, em que o objeto de interesse será detectado. O segundo, denominado *scaleFactor*, indica o fator de escala, um número qualquer maior que 1.0. Esse fator possibilita que a imagem carregada seja reduzida, facilitando a detecção do objeto. O terceiro parâmetro, denominado *minNeighbors*, define a quantidade mínima de vizinhos que cada candidato, retângulo (objeto de interesse detectado) deve possuir para ser, de fato, considerado um objeto de interesse. O objetivo é evitar a ocorrência de falsos positivos, ou seja, impedir que objetos não desejados sejam detectados como objetos de interesse. Quanto maior o valor desse parâmetro, menor a chance de erro, porém, maior a chance de não serem detectados objetos de interesse reais. O quarto parâmetro, denominado *minSize*, indica o tamanho mínimo do retângulo para o possível objeto ser detectado (BARELLI, 2018).

Com isso a Figura 6 ilustra uma detecção utilizando o Haar Cascade, neste caso detectando faces.

Figura 6 - Resultado da aplicação do Haar Cascade na imagem.



Fonte: (OLIVEIRA, et al, 2018).

2.3 Visão Computacional

Os primeiros estudos com Visão Computacional aconteceram por volta de 1955, e na década de 70 foram aliadas às pesquisas de Inteligência Artificial, onde acreditavam que em pouco tempo conseguiram simular e representar o sentido da visão de forma completa em uma máquina. Com pesquisas subsequentes, verificaram que a complexidade era muito maior do que o imaginado, pois não tinham informações de como as imagens eram interpretadas dentro do cérebro humano, mas com o passar do tempo tudo foi ficando mais fácil com buscas mais aprofundadas e com as devidas soluções e ideias encontradas (ROCHA, 2013).

A visão computacional envolve diversas áreas como a matemática, física e entre outras. Hoje, existem diversas bibliotecas e linguagens de programação para desenvolvimento de softwares nessa área, todas elas abstraindo conceitos matemáticos e tornando seu uso mais fácil (BARELLI, 2018). Facilitando também assim o processo de captação de informações para aplicar na visão computacional. A Figura 7 ilustra campos de estudo que estão interligados com a Visão Computacional.

Figura 7 - Campos que ligam a visão computacional.



Fonte: (BARELLI, 2018).

Segundo Backes, Junior, (2016) a visão computacional pode ser constituída de várias fases. São elas:

- **Aquisição:** responsável pela captação das imagens, ou seja, tenta simular a função dos olhos. Os dispositivos que cumprem esse papel são os scanners, filmadoras, máquinas fotográficas etc.
- **Processamento de imagens:** responsável por “melhorar” a imagem, isto é, retirar ruídos, salientar bordas, suavizar a imagem. Essa etapa pode ser um fim em si mesma ou ter o propósito de fornecer uma imagem mais adequada para as próximas fases. É importante salientar que essa fase compreende tanto o que usualmente se denomina “pré-processamento”, como rotação da imagem, equalização de histograma. Quanto aos processamentos mais complexos, como, por exemplo, filtragens e aplicação de operadores morfológicos.
- **Segmentação:** responsável por particionar a imagem em regiões de interesse. Por exemplo, em uma imagem de paisagem, poderíamos estar interessados apenas na porção que representa o céu, ou a vegetação, ou apenas o lago, ou algum cisne nesse lago.
- **Extração de características/Análise de imagens:** responsável por obter um conjunto de características do objeto de interesse. Em outras palavras, essa fase é responsável por encontrar uma codificação numérica que represente determinada imagem, como uma espécie de “impressão digital” (analogia imperfeita) que permita identificá-lo.

- Reconhecimento de padrões: responsável por classificar ou agrupar as imagens com base em seus conjuntos de características. Por exemplo, se você ver a foto de uma única laranja, provavelmente saberá que aquele objeto pertence à classe “laranja” com base em atributos como cor, rugosidade da casca, formato, tamanho. É importante salientar que o objeto visto não é igual às laranjas vistas no passado, mas apenas similar (na verdade, segundo a filosofia, a igualdade é um conceito teórico que não existe na natureza). No entanto, mesmo com essa limitação conseguimos classificá-lo corretamente na maioria dos casos. A Figura 8 ilustra esses processos de uma visão computacional.

Figura 8 - Processos de uma visão computacional.



Fonte: (BACKES, JUNIOR, 2016).

Com os avanços tecnológicos inúmeras tarefas que realizamos manualmente podem ser automatizadas por sistemas de Visão Computacional. Exemplos disso são os sistemas de monitoramento por imagem, capazes de detectar pessoas em um ambiente, robôs industriais, preparados para montar automóveis, são exemplos comuns da aplicação da Visão Computacional em nosso cotidiano. Interessada no desenvolvimento de veículos autônomos, inteligentes para trafegar sem a intervenção humana, a indústria automotiva tem investido cada dia mais no desenvolvimento desse tipo de sistema por se tratar de uma nova experiência trará mais segurança e comodidade para as pessoas e essa realidade está bem próxima (BARELLI, 2018). A Figura 9 ilustra um desses veículos, equipado com um avançado sistema de Visão Computacional capaz de rastrear e detectar pessoas, placas, ruas e outros objetos.

Figura 9 - Veículo autônomo.



Fonte: (BARELLI, 2018).

Assim como na área industrial, os sistemas baseados em visão computacional também são bastante usados na área da saúde. Essa tecnologia unificada a técnicas de aprendizado de máquina, que permitem ao computador aprender e aperfeiçoar seu desempenho em alguma tarefa, tem sido empregada para detectar anomalias em exames por imagem, como: tomografia computadorizada, ressonância magnética, ultrassom e entre outras. Lazer é uma outra área que a visão computacional vem se expandindo. Com isso, detectar gestos de mãos é uma opção a mais de inúmeras soluções ou tarefas para se aplicar junto com visão computacional.

2.4 Processamento de Imagens

Uma imagem pode ser definida como sendo uma função bidimensional $f(x; y)$, em que x e y são coordenadas espaciais e a amplitude de f em qualquer par de coordenadas $(x; y)$ é chamada de intensidade ou nível de cinza da imagem naquele ponto. Quando $x; y$ e os valores de f são todos finitos e discretos, a imagem $f(x; y)$ é denominada de imagem digital (GONZALEZ; WOODS, 2008). Assim, uma imagem digital é composta por um número finito de elementos, cada um com localização e valor particulares, conhecidos como pixels. Portanto, uma imagem digital, pode-se aplicar operações aritméticas e lógicas, pois sua representação é totalmente numérica, uma matriz de pixels, ou seja, sua localização, intensidade de luz, tons e

cores são números armazenados em cada pixel e diante deste conceito, são criados filtros, melhoria na qualidade das imagens, correções, e criação de efeitos artísticos (ROCHA, 2013).

Um vídeo é uma função de intensidade espaço-temporal denotada por $F(x; y; t)$, em que x e y são coordenadas espaciais e t é a variável temporal. Da mesma forma que com as imagens, se $x; y; t$ e o valor de F são todos finitos e discretos, a função $F(x; y; t)$ corresponde a um vídeo digital. Pode-se assim visualizar o vídeo digital como sendo uma sequência de imagens digitais do tipo $f(x; y)$. Essas imagens são denominadas no contexto do vídeo digital de frames (TEKALP, 1995 apud FREITAS, 2011). A Figura 10 ilustra toda a performance formando de um vídeo.

Figura 10 - Exemplo de uma performance de um vídeo.



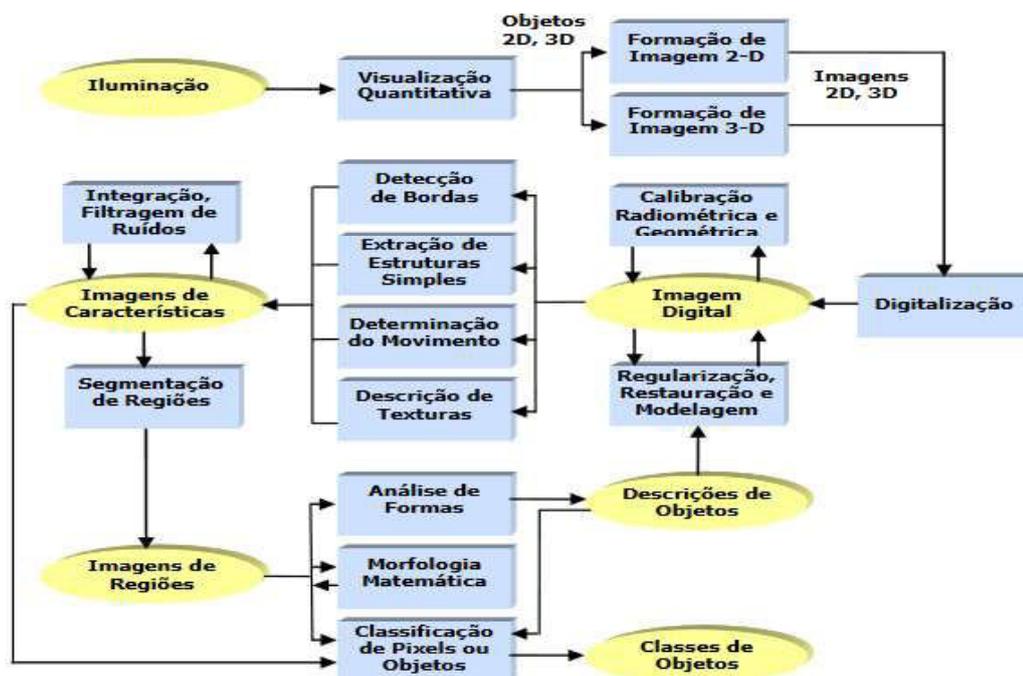
Fonte: (RAMOS, 2011).

Uma imagem colorida pode ser representada pelo modelo de cores RGB (Red, Green, Blue) é formado pela união de cores vermelho, verde e azul e que está relacionado com a forma que as cores são percebidas na retina do olho. Esse modelo é o utilizado nas televisões, computador e entre outros que projete cor com luz. Com isso uma imagem colorida pode ser representada por uma matriz tridimensional, contendo todas as matrizes de cores vermelha, verde e azul (Ramos, 2011).

Uma imagem ou vídeo de boa qualidade ela passa pelo Processamento Digital de Imagens (PDI) não é uma tarefa simples, na realidade envolve um conjunto de tarefas interligadas. Tudo se inicia com a captura de uma imagem, a qual, normalmente, corresponde à iluminação que é refletida na superfície dos objetos, realizada através e um sistema de aquisição. Após a captura por um processo de digitalização, uma imagem precisa ser representada de forma

apropriada para tratamento computacional. Imagens podem ser representadas em duas ou mais dimensões com isso fazendo com que haja filtros e efeitos que torne as imagens digitais, elegantes, dinâmicas e intuitivas com o mundo real. A Figura 11 ilustra as etapas para um processo de digitalização (QUEIROZ, GOMES, 2014).

Figura 11 - Processos de uma digitalização.

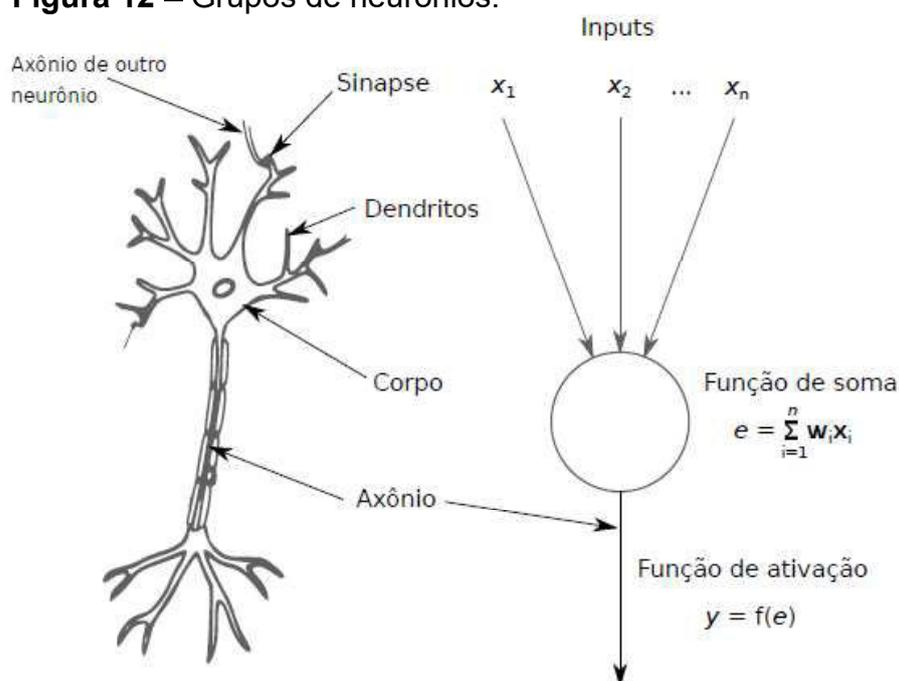


Fonte:(QUEIROZ, GOMES, 2014).

2.5 Redes Neurais Artificiais

Segundo Guedes (2017) Redes Neurais Artificiais são grupos de neurônios interligados que formam um sistema nervoso, tal como o cérebro humano. As redes neurais artificiais são abstrações do modelo biológico que permitem compor os neurônios como unidades simplistas de computação. Nessa analogia os neurônios artificiais replicam o comportamento dos biológicos ao agrupar os impulsos provenientes de entradas ou de axônios de outros neurônios até que um certo limite é atingido, esse limite é definido pela função de ativação. Quando esse limite é atingido o neurônio dispara um impulso por seu axônio que por sua vez pode estimular outros neurônios ou ser uma saída. A Figura 12 ilustra um pouco sobre isso.

Figura 12 – Grupos de neurônios.



Fonte: (GUEDES, 2017).

Esse conceito de redes neurais biológicas é explorado desde de 1940 quando *Warren McCulloch* e *Walter Pitts* modelaram uma simples rede neural usando circuitos elétricos. Isso gerou uma grande especulação gerando infinitas possibilidades que poderiam ser alcançadas, porém a falta de tecnologia e *hardwares* de alto processamento para suportar esses avanços e um sentimento de medo crescente devido aos efeitos que essa tecnologia poderia trazer para os seres humanos, com isso reduziram-se bastante o seu impulso (ROBERTS, 2000 apud GUEDES, 2017).

O avanço das tecnologias e dos *hardwares* possibilitou também a evolução das redes neurais fazendo-se mais presente no dia a dia, porém devido ao seu alto nível de processamento por se tratar de algoritmos enormes e de maior complexidade, as redes neurais podem ser substituída por técnicas como *Haar Cascade* que faça a mesma função de detecção e que requer menos processamento do *hardware*.

2.5.1 Redes Neurais Convolucionais

Convolutional Neural network – (CNN) é uma classe da Rede Neural aplicada ao processamento de imagens. Esta Rede Neural apresenta algumas vantagens tais

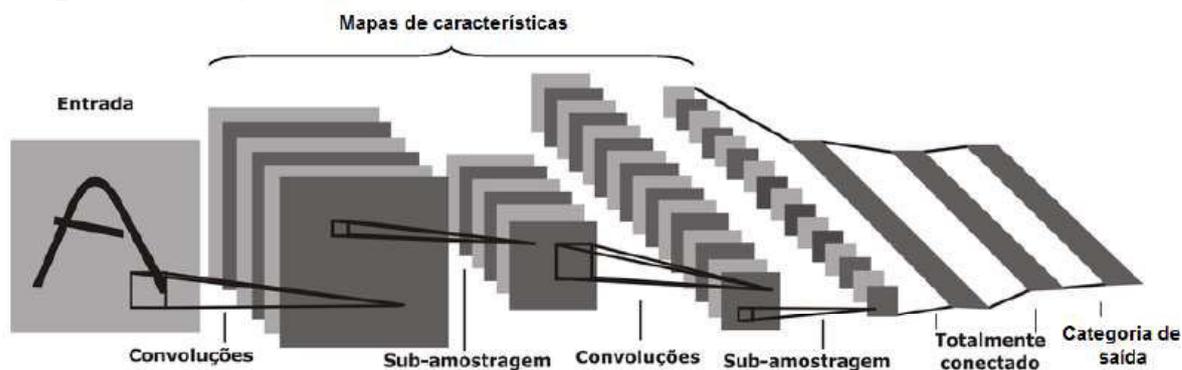
como a capacidade de extrair características relevantes através de aprendizado de transformações (*kernels*) e depender de menor número de parâmetros de ajustes do que redes totalmente conectadas com o mesmo número de camadas ocultas. Como cada unidade de uma camada não é conectada com todas as unidades da camada seguinte, há menos pesos para serem atualizados, facilitando assim o treinamento (ARAÚJO, et al, 2017).

Segundo Santos (2017) uma CNN tem os seguintes pontos arquiteturais:

- Mapeamento de características, ou campos receptivos locais, como um dos pontos de maior similaridade com as redes neurais biológicas que garante uma maior robustez a distorções locais.
- Pesos compartilhados que permitem uma redução dos parâmetros livres e invariância geométrica.
- Sub-amostragem temporal ou espacial, o que reduz o tamanho total dos mapas de características a cada camada, chegando assim na última camada apenas como valores unidimensionais, o que nesse ponto em diante as torna equivalentes a uma rede neural MLP.

A Figura 13 ilustra uma visão geral da arquitetura de uma Rede Neural Convolutiva.

A Figura 13 - Arquitetura de uma CNN.



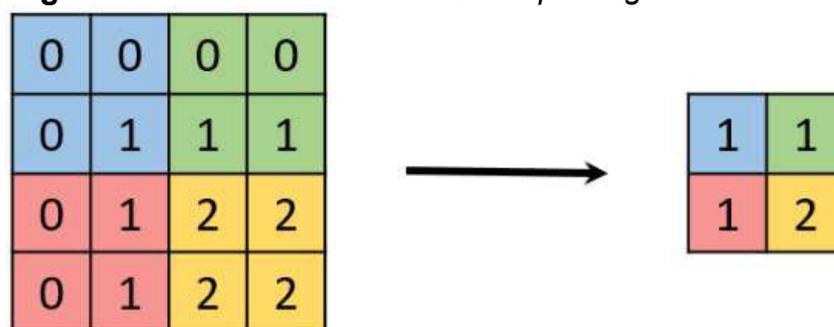
Fonte: (SANTOS, 2017).

A CNN possui três principais camadas, são elas, as convolucionais, de *pooling* e totalmente conectadas. As camadas convolucionais são responsáveis por extrair atributos dos volumes de entradas. As camadas de *pooling* são responsáveis por reduzir a dimensionalidade do volume resultante após as camadas convolucionais e ajudam a tornar a representação invariante a pequenas translações na entrada. As camadas totalmente conectadas são responsáveis pela propagação

do sinal por meio da multiplicação ponto a ponto e o uso de uma função de ativação. A saída da CNN é a probabilidade da imagem de entrada pertencer a uma das classes para qual a rede foi treinada (ARAÚJO et al, 2017).

As camadas convolucionais consistem de um conjunto de filtros que recebem como entrada um arranjo 3D, também chamado de volume. Cada filtro possui dimensão reduzida, porém ele se estende por toda a profundidade do volume de entrada. Por exemplo, se a imagem for colorida, então ela possui 3 canais e o filtro da primeira camada convolucional terá tamanho 5x5x3 (5 pixels de altura e largura, e profundidade igual a 3). Automaticamente, durante o processo de treinamento da rede, esses filtros são ajustados para que sejam ativados em presença de características relevantes identificadas no volume de entrada, como orientação de bordas ou manchas de cores. A relevância é avaliada de tal forma que os resultados sejam otimizados em função de um conjunto de amostras previamente classificadas (ARAÚJO, et al, 2017).

Após uma camada convolucional, geralmente existe uma camada de *pooling*. O objetivo dessa camada é reduzir progressivamente a dimensão espacial do volume de entrada, conseqüentemente a redução diminui o custo computacional da rede e evitar *overfitting* (KARPATHY, 2017 apud ARAÚJO, et al, 2017). Na operação de *pooling*, os valores pertencentes a uma determinada região do mapa de atributos, gerados pelas camadas convolucionais, são substituídos por alguma métrica dessa região. A forma mais comum de *pooling* consiste em substituir os valores de uma região pelo valor máximo (GOODFELLOW, et al, 2016 apud ARAÚJO, et al, 2017). Essa operação é conhecida como *max pooling* e é útil para eliminar valores desprezíveis, reduzindo a dimensão da representação dos dados e acelerando a computação necessária para as próximas camadas, além de criar uma invariância a pequenas mudanças e distorções locais. Outras funções de *pooling* comumente usadas são a média, a norma L2 e a média ponderada baseada na distância partindo do pixel central (AGGARWAL, et al, 2001 apud ARAÚJO, et al, 2017). A Figura 14 ilustra esse processo de *pooling*.

Figura 14 - Processo de camada de *pooling*.

Fonte: (ARAÚJO, et al, 2017).

Com a saída das camadas convolucionais e de pooling formam as características extraídas da imagem de entrada. O objetivo das camadas totalmente conectadas é utilizar essas características para classificar a imagem em uma classe pré-determinada, assim possibilitando um processo mais simples, mais leve e de melhor resultado que é detectar um objeto (ARAÚJO, et al, 2017).

2.6 Internet das Coisas

A Internet das Coisas (*Internet of Things* - IoT) se mostra como uma evolução da Internet e um novo paradigma tecnológico, social, cultural e digital. Com isso em 1990, *John Romkey* criou o primeiro dispositivo em Internet das Coisas. O autor criou uma torradeira que poderia ser ligada e desligada pela Internet e a apresentou na INTEROP '89 Conference. *Dan Lynch*, presidente da Interop na época, prometeu a *John Romkey* que, se a torradeira fosse ligada pela Internet, o aparelho seria apresentado em exposição durante a conferência. Diante desse desafio, *John Romkey* conectou a torradeira a um computador com rede TCP / IP, e foi um tremendo sucesso. Porém, durante esse teste, o pão foi incluído manualmente na torradeira. Após um ano, esse requisito foi corrigido, aprimorado de uma forma mais intuitiva e com isso foi apresentado na mesma conferência, foi acrescentado um pequeno guindaste robótico no sistema. Esse robô era controlado pela Internet, pegou a fatia de pão e colocou na torradeira, automatizando, dessa forma, o sistema de ponta a ponta (MANCINI, 2018).

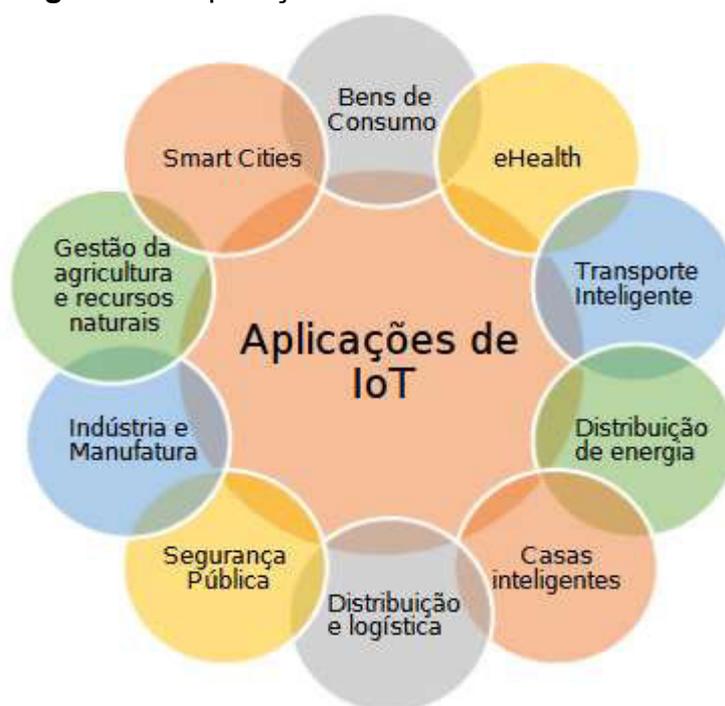
Essa tecnologia proporciona aos objetos do dia a dia, capacidade computacional e de comunicação, se conectarem à internet e também via *bluetooth* que é considerável uma rede sem fio do tipo Pan ou Wpan por se tratar de uma conexão de tráfego de dados em pequeno, curto espaço de tempo. Essa conexão viabiliza controlar remotamente os objetos, e acessá-los como objetos de serviços, e

se tornarão objetos inteligentes. Os objetos inteligentes possuem capacidade de comunicação e processamento aliados a sensores assim com essa junção de funcionalidades se tornará um objeto além de inteligente, também manipulador por se tratar de objetos que poderão fazer tarefas manuais feita por humanos e até mesmo tarefas que não podem ser feitas por humanos.

Atualmente não só computadores estão conectados à Internet, como também uma variedades de equipamentos, tais como TVs, laptops, eletrodomésticos, automóveis, smartphones, entre outros. Nesse novo cenário, a tendência é que aumente cada vez mais o número de dispositivos interligados a rede, com isso tudo indica que mais de 50 bilhões de dispositivos estarão conectados até 2020. Com o uso dos objetos inteligentes será possível detectar seu contexto, controlá-lo, viabilizar troca de informações uns com os outros, acessar serviços da internet e interagir com as pessoas. Com isso, uma gama de novas possibilidades de aplicações surge, como por exemplo, na saúde (*smart healthcare*); casas inteligentes (*smart home*); veículos inteligentes (carros autônomos) e dentre outros. Essas novas habilidades dos objetos inteligentes gerarão um grande número de oportunidades de pesquisas e projetos no âmbito acadêmico e empresarial (EVAN, 2011 apud MANCINI, 2018).

Com tanta inteligência e poder de aquisição envolvida nessa tecnologia, podemos ver que as aplicações de Internet das Coisas que são inúmeras e diversas, e permeiam praticamente a vida diária das pessoas, das empresas e sociedade como um todo, transformando o mundo em *smart world* (FREITAS DIAS, 2016; PATEL, PATEL, 2016 apud MANCINI, 2018). A Figura 15 ilustra algumas aplicações da IoT.

Figura 15 - Aplicações da IoT.



Fonte: (MANCINI, 2018).

Segundo Freitas (2016), explica como a IoT podem ser aplicadas:

- Bens de consumo. Bens adquiridos pelos consumidores, tais como *smartphones*, *smart house*, *smart car* e *smart TV*.
- eHealth. Fitness, bioeletrônica e cuidados com saúde. Por exemplo: monitoramento e controle da frequência cardíaca durante os exercícios; monitoramento das condições dos pacientes em hospitais e em casas de idosos.
- Transporte inteligente. Notificação das condições de tráfego, controle inteligente de rotas, monitoramento remoto do veículo, coordenação das rodovias e integração inteligente de plataformas de transporte.
- Distribuição de energia (*smart grid*). Acompanhamento de instalações de energia, subestações inteligentes, distribuição de energia automática e medições remotas de relógios residenciais.
- Casas inteligentes. Medições remotas de consumo, economia de energia, controle inteligente de equipamentos residenciais e segurança residencial.
- Distribuição e Logística. *Smart e-commerce*, rastreabilidade, gerenciamento na distribuição e inventário.

- Segurança Pública. Monitoramento no transporte de cargas perigosas e químicas, monitoramento da segurança pública, monitoramento das estruturas de construções de utilidade pública.
- Indústria e Manufatura. Economia de energia, controle da poluição, segurança na manufatura, monitoramento do ciclo de vida dos produtos, rastreamento de produtos manufaturados na cadeia de abastecimento, monitoramento de condições ambientais e controle de processos de produção.
- Gestão da agricultura e dos recursos naturais. Segurança e rastreabilidade de produtos agrícolas, gerenciamento de qualidade, monitoramento ambiental para produção e cultivo, gerenciamento no processo de produção, utilização de recursos para a agricultura.
- *Smart Cities*.
 - Monitoramento estrutural: monitoramento de vibrações e condições dos materiais em edifícios, pontes e monumentos históricos.
 - Energia elétrica: iluminação inteligente e adaptável conforme a rua.
 - Segurança: monitoramento por meio de vídeo digital, gerenciamento de controle de incêndio e sistemas de anúncio público.
 - Transporte: estradas inteligentes com avisos, mensagens e desvios de acordo com as condições climáticas e eventos inesperados como acidentes ou engarrafamentos.
 - Estacionamento: monitoramento em tempo real da disponibilidade de espaços de estacionamento, sendo possível identificar e reservar vagas disponíveis.
 - Gestão de resíduos: detecção de níveis de lixo em recipientes para otimizar a rota de coleta de lixo.

Diante de tantas aplicações feitas pela IoT, podemos ver o quanto ela é e vai ser importante para nosso crescimento tecnológico em diversas áreas e formas. Com a IoT e a visão computacional, elas duas juntas podem trazer melhorias de aplicações já existentes e novas aplicações. Reconhecer gestos de mãos para realizar uma tarefa doméstica seria uma opção de novas aplicações na área de lazer e comodidade.

2.7 Raspberry PI

Em 2011, a Fundação Raspberry PI desenvolveu seu primeiro computador em placa única. O Raspberry PI consiste em um computador com o tamanho reduzido, semelhante ao tamanho de um cartão de crédito, com baixo custo e baixo consumo de energia. O computador foi criado com o intuito de promover o ensino de programação para crianças e adultos. Hoje ele é utilizado de diversas formas, por exemplo, para programar, como vídeo game, mini servidor, automação residencial, robótica entre outros projetos (OLIVEIRA, et al. 2018).

Este dispositivo pode ser considerado um computador “completo”, podendo ser classificado como *Single-Board Computer* (SBC), por se tratar de uma única placa com processador, memória e interfaces de entrada e saída. O Raspberry PI se enquadra na categoria de equipamento denominada *Open-Source Hardware* (OSH), ou equipamento livre. Pois se trata de dispositivos cujas informações são disponibilizadas para todos, de forma que o público possa fazer uso, modificar, produzir ou distribuir este material.

Existe vários tipos de modelos de Raspberry PI, tornando-a a placa mais robusta com inúmeras melhorias, onde iniciou com a modelo do tipo A, e que a mais recente já está em Raspberry PI 4B+.

2.8 Módulo Wi-Fi ESP8266 NodeMCU ESP-12

Trata-se de uma placa para desenvolvimento de aplicações, produzida pela fabricante chinesa *Espressif*, desde seu lançamento no mercado revolucionou a utilização da comunicação wireless para projetos envolvendo a IoT, onde esse componente possui mais de 12 modelos. Com isso passou de apenas uma microcontroladora, módulo wi-fi ESP8266, para uma placa completa contendo pinos GPIO externos a mais para demais funcionalidades, recursos como entrada micro-usb, conversor de voltagem de 9 volts para 3.3 volts, com isso sendo renomada como NodeMCU que agrega com ela uma plataforma de desenvolvimento em Lua e também em C++ através da *Integrated Development Environment* (IDE) do Arduino, onde existe diversas bibliotecas compatíveis, com o intuito de poder se conectar ao wi-fi e assim interagir com demais dispositivos (DE OLIVEIRA,2017).

2.9 Módulos

Módulos são pequenas placas que contém sensores e outros componentes auxiliares como resistores, capacitores, leds e entre outros para garantir seu bom funcionamento, capazes de transformar uma grandeza física em um sinal elétrico, respondendo a um estímulo físico ou químico de maneira analogicamente (MATOS, 2015). Com esses módulos pode-se fazer inúmeras tarefas práticas e dinâmicas como, automatizar sua casa ligando lâmpadas, ventiladores, tvs, bombas de água e dentre outras tarefas. A Figura 16 ilustra alguns desses módulos existentes.

Figura 16 - Módulos.



Fonte: (DE FILIPEFLOP, 2018).

2.10 Trabalhos Relacionados

O trabalho de Guedes(2017) retrata o mesmo com o seguinte tema “Reconhecimento de Gestos Usando Redes Neurais Convolucionais”, demonstrou de forma detalhada passo a passo os processos realizados para detectar gestos de mãos. O mesmo teve ajuda e foco sobre o trabalho de Kolleret al.(KOLLER; NEY; BOWDEN, 2016) usando o dataset de 3359 imagens e também o modelo processamento do CNN utilizado pelos autores citados. Guedes(2017) obteve em seu trabalho os seguintes resultados das pastas expressadas com os seguintes nomes top-1 85% e top 5 94.8% de detecção.

O trabalho de Silva (2017) retrata o mesmo com seguinte tema “Reconhecimento De Gestos Customizados Da Mão Em Tempo Real usando Aprendizado De Métricas E Grafos De Ação”, o autor diz que a detecção de mãos juntamente com os sinais de Libras é uma opção de aproximar mais ainda homem e

máquina. Para a detecção em tempo real ele utilizou redes neurais juntamente com o sensor Leap Motion, criado pela empresa homônima e apresentado ao mercado em 2013, é uma tecnologia que consiste em um pequeno dispositivo dotado de sensores ópticos e luz infravermelha, que faz o reconhecimento e rastreamento de mãos e dedos do usuário em tempo real. Para os resultados foi desenvolvido um software na linguagem de programação C++ com a biblioteca OpenGL para visualização do esqueleto da mão rastreado pelo sensor Leap Motion. Foi utilizado ainda, a parte, o software Matlab para a etapa de aprendizagem das métricas, tanto no caso linear quanto no não linear. Foram criadas 4 aplicações distintas, uma para cada categoria de poses estáticas consideradas. Em cada uma delas foi feita a análise de acurácia nas tarefas: reconhecimento de poses estáticas, reconhecimento de gestos dinâmicos de forma offline e reconhecimento de gestos dinâmicos de forma online. Em relação aos resultados pode-se dizer que foram satisfatórios retornando a maioria dos resultados acima de 90% de detecção.

Segundo Bastos (2015) retrata o mesmo com o seguinte tema “Reconhecimento De Sinais Da Libras utilizando Descritores De Forma E Redes neurais Artificiais “Gestos são ações corporais não-verbais voltadas para a expressão de algum significado. Estes incluem movimentos de mãos, face, braços, dedos, entre outros, sendo abordados por trabalhos que visam reconhecê-los para promover interações humanas com sistemas computacionais. Para a extração de informações das imagens, optou-se pelo uso de um vetor de características resultante da aplicação dos descritores Histograma de Gradientes Orientados (HOG) e Momentos Invariantes de Zernike (MIZ), os quais voltam-se para as formas e contornos presentes nas imagens. Para o reconhecimento, foi utilizado o classificador Perceptron Multicamada, sendo este disposto em uma arquitetura onde o processo de classificação é dividido em 2 estágios. Devido à inexistência de datasets públicos da Libras, fez-se necessária, com o auxílio de especialistas da língua e alunos surdos, a criação de um dataset de 9600 imagens, as quais referem-se a 40 sinais da Libras. Isso fez com que a presente abordagem partisse desta criação do dataset até a etapa final de classificação dos sinais. Por fim, testes foram realizados e obteve-se 96,77% de taxa de acerto, evidenciando um alto índice de acerto.

Os trabalhos relacionados descritos anteriormente, apresentam característica em comum o reconhecimento de gestos de mãos, contudo, os trabalhos não

apresentam a conexão dos componentes em rede de computadores com o intuito de acionar objetos através do reconhecimento. Esta pesquisa utiliza o algoritmo de Haar Cascade e Redes Neurais Artificiais, bem como a conexão entre a Raspberry PI e a NodeMCU. No qual mostrou resultados satisfatórios e com sugestões de trabalhos futuros voltados para outras plataformas como Web, Android e IOS.

3 METODOLOGIA

Neste trabalho, o algoritmo de detecção de objetos em imagens *Haar Cascade* e a CNN foram estudadas com o propósito de encontrar e adaptar classificadores de gestos de mãos. Dessa forma foi realizado um estudo sobre detecção de gestos de mãos, em busca da técnica com melhor desempenho de processamento na detecção que foi realizado em imagens e vídeos em tempo real.

Tendo em vista a proposta de gestos de mãos, realizada as adaptações de melhor suavização, remoção de ruídos, parâmetros para melhor detecção, inclusões de funções caso detecte um gesto realize algum acionamento, ao algoritmo *Haar Cascade* disponível em seu repositório git de *Devashish Tanaja* e ao algoritmo de CNN de *Raj Shah* para uma melhor detecção de acordo com luminosidade e ambiente, dessa forma, o classificador poderá detectar a mão e o gesto realizado.

Após as adaptações dos classificadores de gestos, realiza-se a detecção e um comparativo entre as técnicas para identificar qual obteve o melhor desempenho de classificação. Pensando na comodidade e em auxiliar pessoas com dificuldade de locomoção, a classificação de gestos com as mãos pode ser utilizada para auxiliar em tarefas como ligar uma lâmpada, sem a necessidade de ir ao encontro do interruptor.

Posteriormente, realizou-se a configuração da comunicação entre Raspberry PI e a NodeMCU através do protocolo de transporte MQTT juntamente com a instalação de *broker* na raspberry PI. Dessa forma, no instante que a Raspberry PI detectar um gesto da mão, será enviado uma informação para o NodeMCU solicitando a ação do aparelho, como desligar ou ligar uma lâmpada, um ventilador e entre outros objetos.

3.1 Materiais utilizados

3.1.1 Raspberry PI

Neste trabalho usamos a Raspberry PI 3B que possui a seguinte configuração uma CPU ARMv8 quad-core de 64 bits com frequência de 1.2 GHz, núcleo de gráficos 3D Vídeo Core IV, 1GB de memória RAM, 4 portas USB, LAN sem fio 802.11n, Bluetooth 4.1. Associada ao computador Raspberry utilizamos uma câmera, Raspberry Câmera PI 3 V2 que possui um sensor Imx219 de 8 megapixels da Sony, que pode ser usado para capturar vídeo de alta definição, fotos estáticas e

captura contínua. A câmera acoplada a um micro servo 9G SG90 TowerPro fixado a um suporte de ferro para possíveis ajustes de posicionamento. A Figura 17 ilustra esse modelo juntamente com câmara acoplada ao suporte.

Figura 17 - Raspberry PI 3B com sua câmera acoplada.

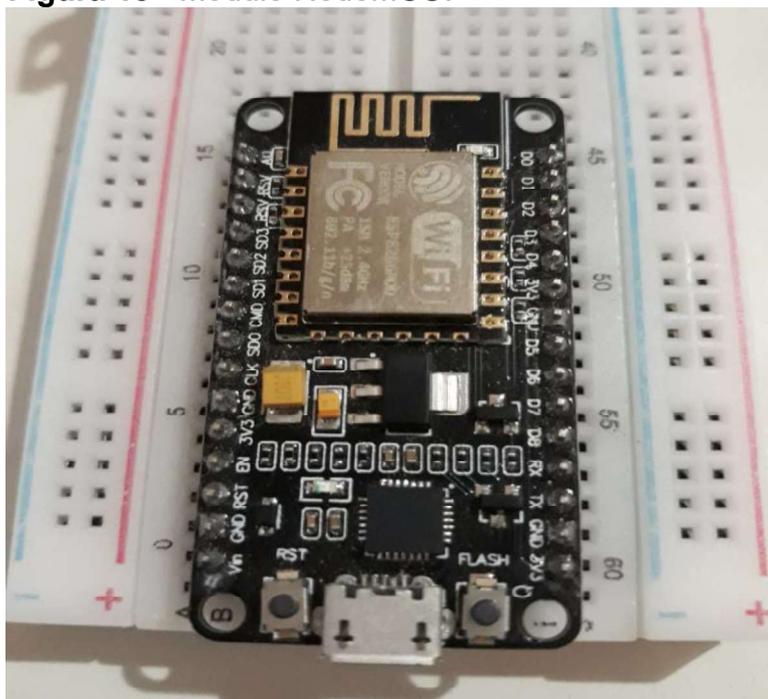


Fonte: (AUTOR, 2019).

O dispositivo tem a função de realizar o processamento das imagens em tempo real através dos algoritmos de Haar Cascade e CNN em uma linguagem *Python* juntamente com bibliotecas instaladas como OpenCV, Numpy, MQTT, broker *Eclipse Mosquitto*. No qual algoritmo em execução realiza o processo de conexão com a NodeMCU e detecção de gestos no qual irá detectar o gesto e enviará a mensagem de acionamento de um objeto para a microcontroladora.

3.1.2 Módulo Wi-Fi ESP8266 NodeMCU ESP-12

O modelo utilizado possui 30 pinos dentre eles: uma VIN para alimentação externa, quatro GND, onze GPIO, três de 3.3 volts e os demais onze pinos restantes são de reset, interface SPI e dentre outros. Caracterizando uma placa didática capaz de realizar diversos projetos, desde um simples liga led há algo mais complexo como uma automação residencial. A Figura 18 ilustra esse módulo NodeMCU acoplado em uma protoboard.

Figura 18 - Módulo NodeMCU.

Fonte: (AUTOR, 2019).

O módulo em questão tem como função acionar objetos através de comandos recebidos da Raspberry PI interligados a uma rede de computadores. No qual precisará de módulos como relé para o objetivo final, onde o mesmo trata todo o processo de controle de liberação de energia para ligar objetos especificados ligados a ele.

3.1.3 Módulo Relé

Nesta pesquisa utiliza-se um módulo relé de um canal que tem como função controlar o acionamento de algum objeto, uma lâmpada, um ventilador entre outros. O mesmo consiste em três pinos: o IN o pino responsável por receber o comando de uma placa microcontroladora, o VCC onde receberá sua alimentação de 5 volts e o GND o neutro. O relé consiste também em três bornes onde conecta-se o objeto que queremos acionar, o primeiro borne é o N.A ou normalmente aberto, o responsável de liberar ou não a energia que ali se passa de acordo com ação da NodeMCU. O segundo E ou entrada, direcionada pela entrada de energia e o terceiro borne N.F ou normalmente fechado, responsável por fechar o circuito, onde o relé desligado, sem ação alguma, não terá nenhum controle e com isso o objeto poderá ser acionado qualquer hora. A Figura 19 ilustra o módulo usado na pesquisa.

Figura 19 - Módulo Relé de um canal.



Fonte: (AUTOR, 2019).

3.1.4 MQ Telemetry Transport

MQ Telemetry Transport (MQTT) é um protocolo transporte de mensagens que tem como modelo publicação e assinatura do mesmo com simplicidade e leveza. Esse protocolo foi desenvolvido pelo Dr. *Andy Stanford-Clark*, da IBM, e *Arlen Nipper*, da *Arcom*, agora *Eurotech*, em 1999 (MQTT). O MQTT usa um conceito de *broker* que nada mais é que um *software* servidor que recebe solicitações e as repassa. Esse *broker* atuária da mesma forma que um servidor web com uma diferença, voltada para aplicações que repassam dados simples. O mesmo utilizado nesta pesquisa foi o *Eclipse Mosquitto* por se tratar de um intermediário de mensagens de código aberto e gratuito. Com esse protocolo junto com o *broker* torna-se adequado para projetos da IoT usando sensores, dispositivos móveis e microcontroladores (ECLIPSE MOSQUITTO).

O broker juntamente com o MQTT tem a função nesta pesquisa realizar a conexão da Raspberry PI e a NodeMCU de modo que ela fará a comparação de dados especificados entre elas e se existe equivalência, se sim, é realizado a conexão e por fim o acionamento do objeto, caso contrário, elas não se comunicam resultando no não acionamento do objeto.

4 RESULTADOS E DISCUSSÕES

Neste tópico, são descritos todos os testes realizados e resultados obtidos com a detecção de gestos juntamente com o acionamento de objetos com base na definições atribuídas na metodologia.

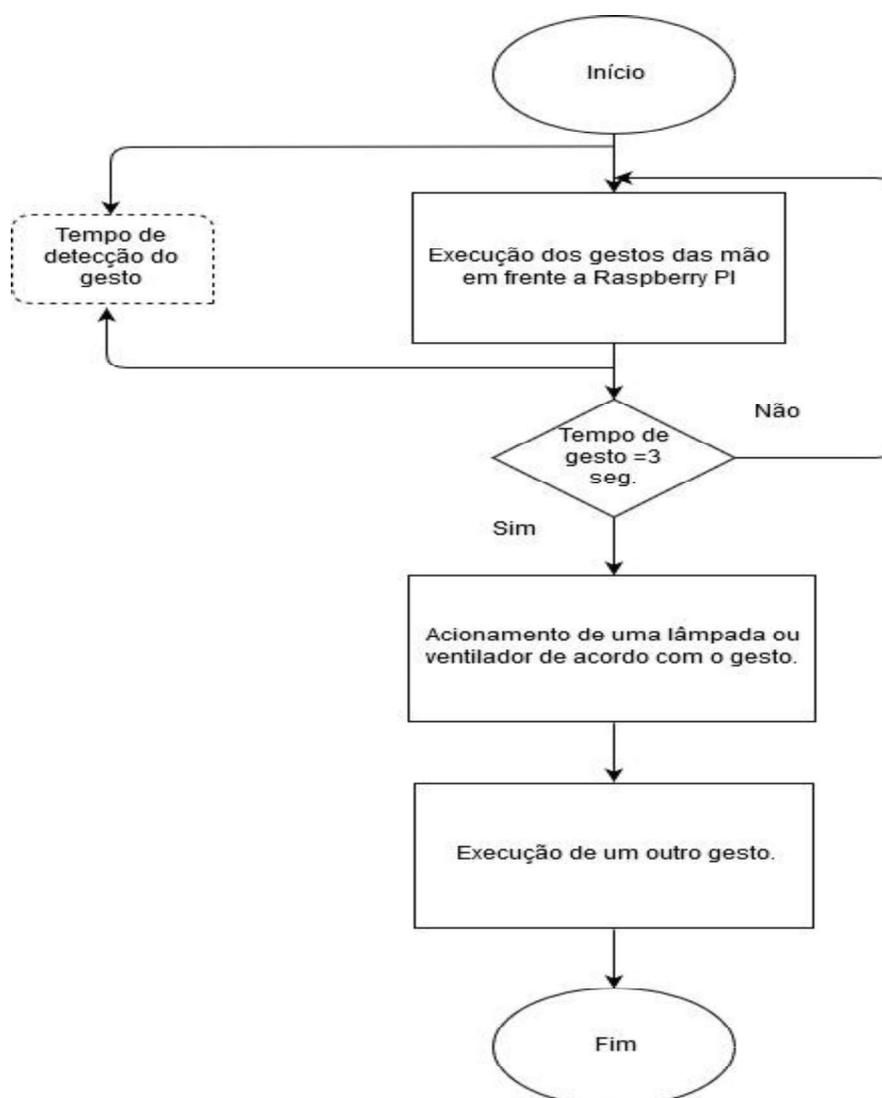
4.1 Detecção de gestos e resultados de sua classificação

Os primeiros testes realizados consistiram em realizar a detecção dos gestos em ambientes com luminosidade natural, porém esses testes não foram considerados satisfatórios devido a baixa luminosidade do ambiente, visto que este fator é de grande importância para que a imagem do gesto seja identificada corretamente. Assim, o ambiente de teste foi adaptado para que a luminosidade não interferisse nos resultados.

Após a adaptação do ambiente, realizou-se testes de confiabilidade seguindo o método disponível em Bonatto (2017). No primeiro teste realiza-se o gesto por três segundos, não havendo variações de interpretação do gesto, a Raspberry Pi se comunica com a NodeMCU através do wi-fi via protocolo MQTT acionando assim a execução de ligar ou desligar uma lâmpada ou um ventilador, objetos selecionados para essa pesquisa. O segundo teste foi o de aferição, tempo de detecção, como se trata de gestos simples e de pouco movimento entre um gesto e outro, tivemos uma base de 157 a 159 milissegundos em cada gesto detectado. A Figura 20 ilustra o fluxograma dos testes realizados desde a detecção até o acionamento do objeto.

O primeiro processo começa com “Tempo de detecção do gesto” que consiste apenas no tempo de detecção de cada gesto, onde resultou entre 157 a 159 milissegundos para cada gesto. No segundo processo ele passa detecção de gestos de 3 segundos, onde realiza uma comparação, se o gesto permanecer até 3 segundos sem variações, realiza-se uma ação, caso contrário, reinicia o ciclo. Com isso, segue até o fim da execução, realizando-se esses processos e em caso de positivo aciona um objeto.

Figura 20 - Fluxograma da lógica dos testes realizados na pesquisa.



Fonte: (AUTOR, 2019).

Posteriormente, realiza-se a execução do algoritmo de detecção de gestos na Raspberry PI, onde está sendo exposto na tela do notebook remotamente através do *Team Viewer*, detectado o gesto, envia-se uma informação de “liga” para a

NodeMCU através do wi-fi via MQTT, quando esta informação é recebida, a microcontroladora aciona os relés que consequentemente liga a lâmpada ou ventilador de acordo com gesto. A Figura 21 ilustra os objetos utilizados.

Figura 21 - Detecção de gestos com acionamento de objetos.



Fonte: (AUTOR, 2019).

O acionamento dos objetos segue os passos descritos a seguir:

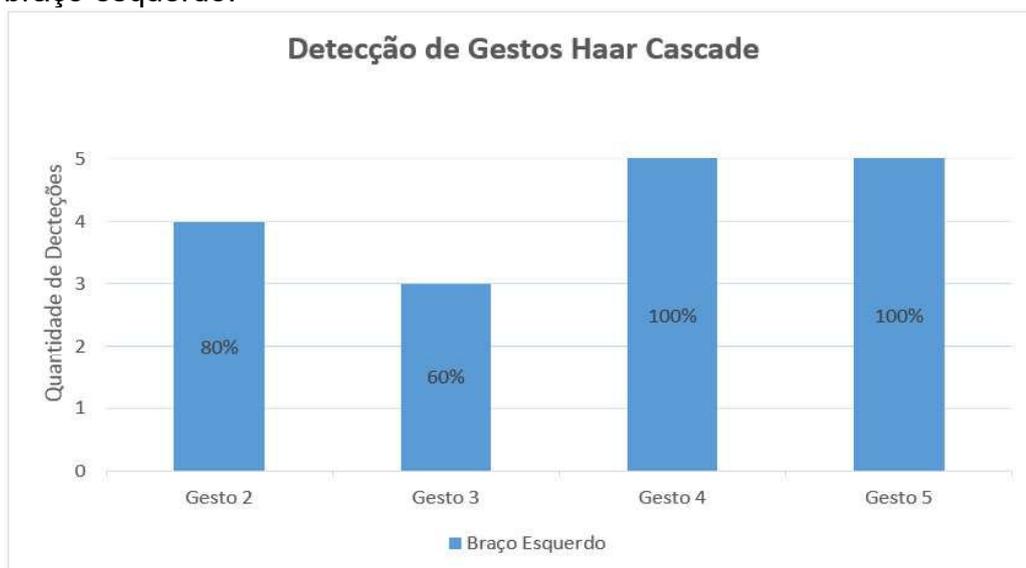
- Posicionamento do usuário com sua mão estendida expondo um determinado gesto exemplo quatro para frente da Raspberry Pi;
- Realizado a detecção do gesto, a Raspberry Pi envia uma mensagem de acionamento, exemplo “liga_Lampada” através do broker para NodeMCU;
- Broker verifica os dados ambos *hardwares* se são equivalentes, se sim envia a mensagem para microcontroladora, se não, realiza-se o novo processo de detecção e conexão entre ambos *hardwares*.
- Mensagem recebida na NodeMCU, ela verifica em seu algoritmo de execução, se existe a mensagem de acionamento, se sim, aciona o módulo que consequentemente aciona o objeto seja ele uma lâmpada ou um ventilador. Se não, realiza-se novo processo de detecção ou

verificação dos algoritmos se as mensagens estão sendo bem especificadas.

4.2 Detecção de gestos utilizando o Haar Cascade

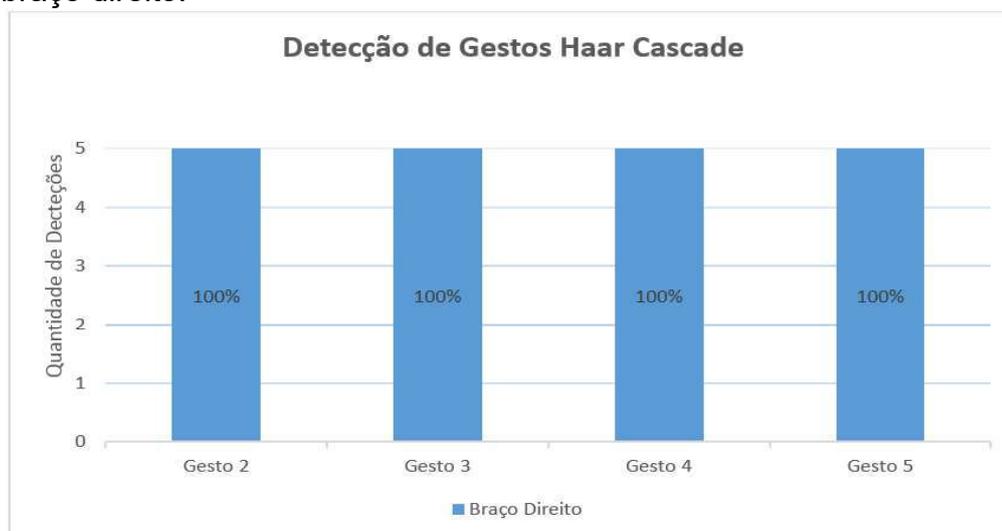
No algoritmo disponível em Devashish Tanaja, têm-se os gestos correspondente aos números 2, 3, 4 e 5. Os testes consistiram em realizar dez vezes os gestos 2, 3, 4 e 5, cada teste com 10 segundos de duração. Os testes foram realizados com o braço esquerdo posicionado a 60 cm da RaspCam e com o braço direito posicionado a 80 cm da RaspCam. Caso houvesse variação dos gestos duas vezes consecutivas, mudava-se o gesto. Os Gráficos 1 e 2 mostram esses resultados.

Gráfico 1 - Resultados da detecção de gestos usando Haar Cascade com o braço esquerdo.



Fonte: (AUTOR, 2019).

Gráfico 2 - Resultados da detecção de gestos usando Haar Cascade com o braço direito.



Fonte: (AUTOR, 2019).

Tendo em vista os resultados expostos anteriormente, realiza-se a demonstração dessas detecções. A Figura 22 ilustra essas detecções.

Figura 22 - Imagens de detecção de gestos.

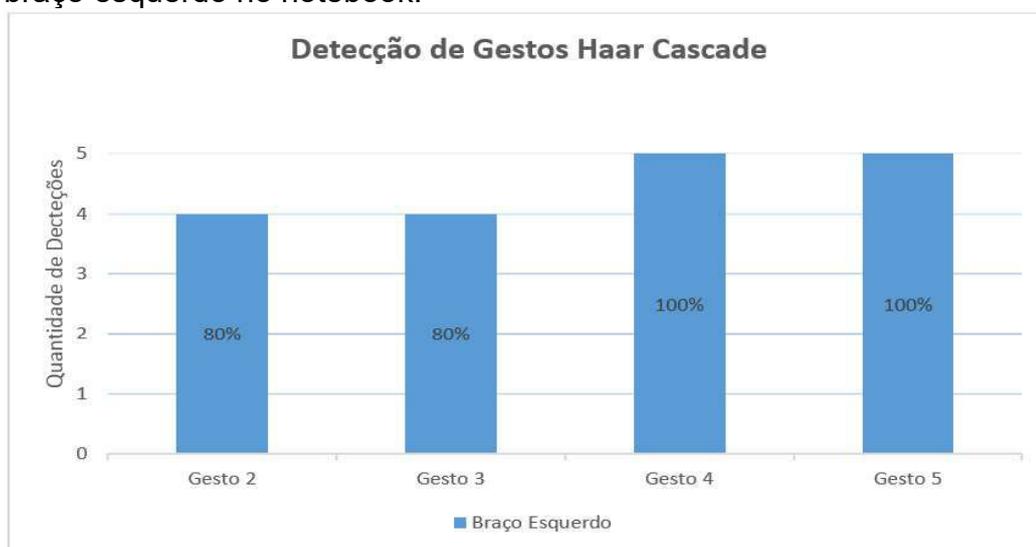


Fonte: (AUTOR, 2019).

O Algoritmo detectou os gestos de forma satisfatória. Porém, devido a carga de processamento na Raspberry PI, os resultados sofreram um *delay* de 4 a 7 segundos.

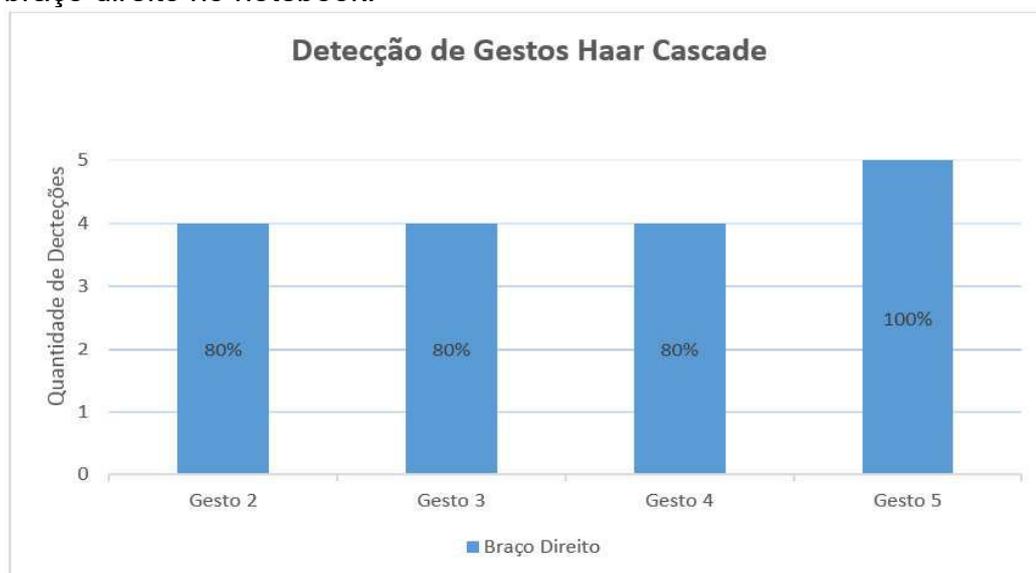
Como alternativa para solucionar o *delay* sobre as imagens, os testes foram realizados novamente utilizando a webcam de um notebook com um processamento superior ao da Raspberry PI. Os Gráficos 3 e 4 ilustram os resultados obtidos.

Gráfico 3 - Resultados da detecção de gestos usando Haar Cascade com o braço esquerdo no notebook.



Fonte: (AUTOR, 2019).

Gráfico 4 - Resultados da detecção de gestos usando Haar Cascade com o braço direito no notebook.



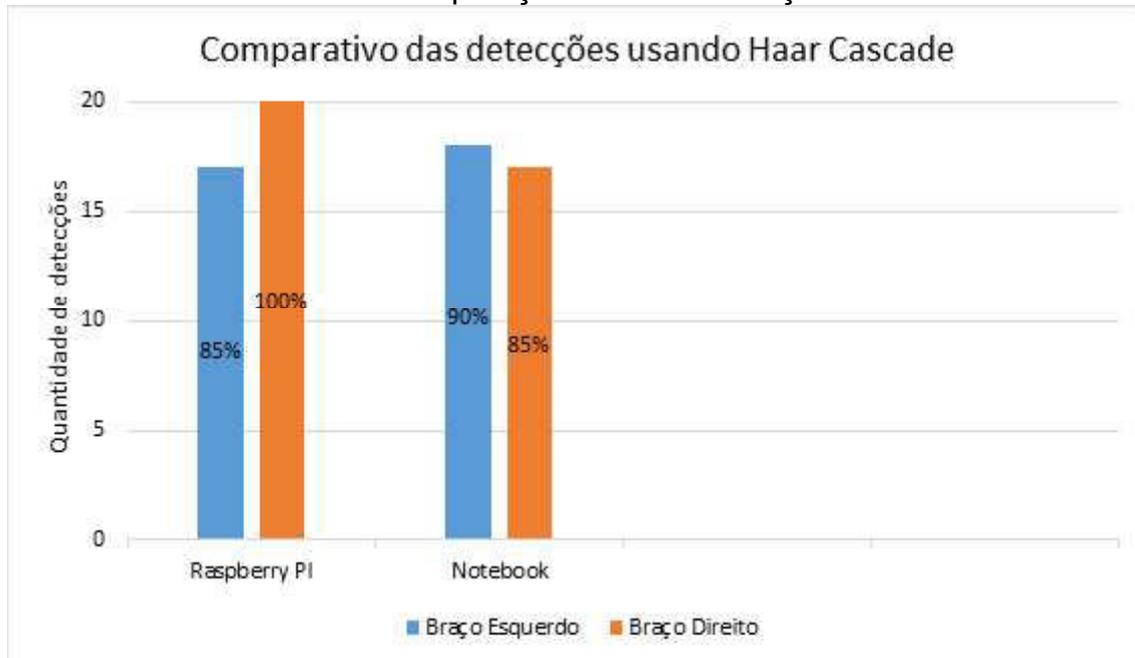
Fonte: (AUTOR, 2019).

Utilizando a webcam do notebook o algoritmo detectou os gestos de forma satisfatória, sem apresentar o *delay* descrito anteriormente. O fato da taxa de acerto no notebook ter sido inferior a taxa de acertos da Raspberry deve-se a baixa resolução da webcam.

4.3 Comparativos das detecções

Na Raspberry PI das 20 detecções ao todo foi detectado 17 acertos com uma probabilidade 85% do braço esquerdo. Onde o braço direito obteve uma marca de 20 acertos resultando em 100%. No notebook obteve-se no braço esquerdo 18 acertos resultando em 90% enquanto no braço direito 17 acertos chegando a 85%. Comparando-se os resultados da Raspberry PI com os resultados obtidos pela webcam do notebook, podemos concluir que o algoritmo com melhor desempenho foi o algoritmo executado na Raspberry PI, mesmo com suas limitações com uma marca de 2 acertos ou 5% a mais, obteve o melhor devido a câmera ser de melhor qualidade do que a webcam. Veja o Gráfico 5 os resultados desse comparativo.

Gráfico 5 - Resultados da comparação entre as detecções realizadas.

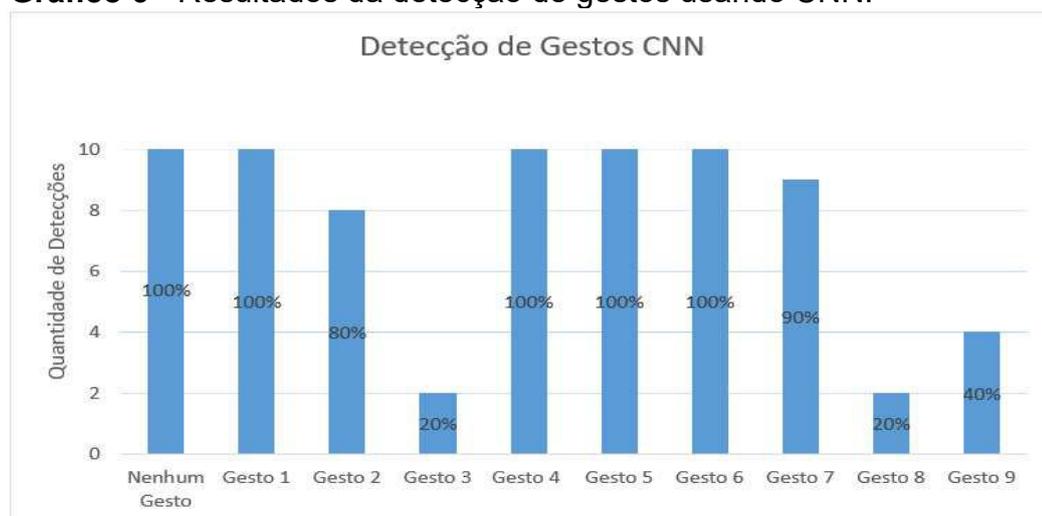


Fonte: (AUTOR, 2019).

4.4 Detecção de gestos utilizando CNN por meio de webcam

O algoritmo disponível em seu repositório git de Raj Shah contendo também uma base de dez mil imagens de gestos em tons de preto e branco para uma melhor detecção, sendo mil para cada gesto, onde poderia-se realizar um novo treinamento e atualizar o arquivo.meta da rede neural já treinada, tendo como gestos de mãos os números, nenhum gesto, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Observa-se nos primeiros testes que sua base disponibilizada foi realizada com a mão direita e não com as duas, devido a mal detecção com a mão esquerda. Após realizar adaptações para melhor detecção devido ao ambiente, realizamos dez tentativas de detecção, tendo como regra de dez segundos para possível detecção, e caso detectasse um outro gesto duas vezes sem ser o verdadeiro, mudava-se de gesto. Tendo em vista sua distância de aproximadamente de 60 centímetros. Veja no Gráfico 6 os resultados obtidos.

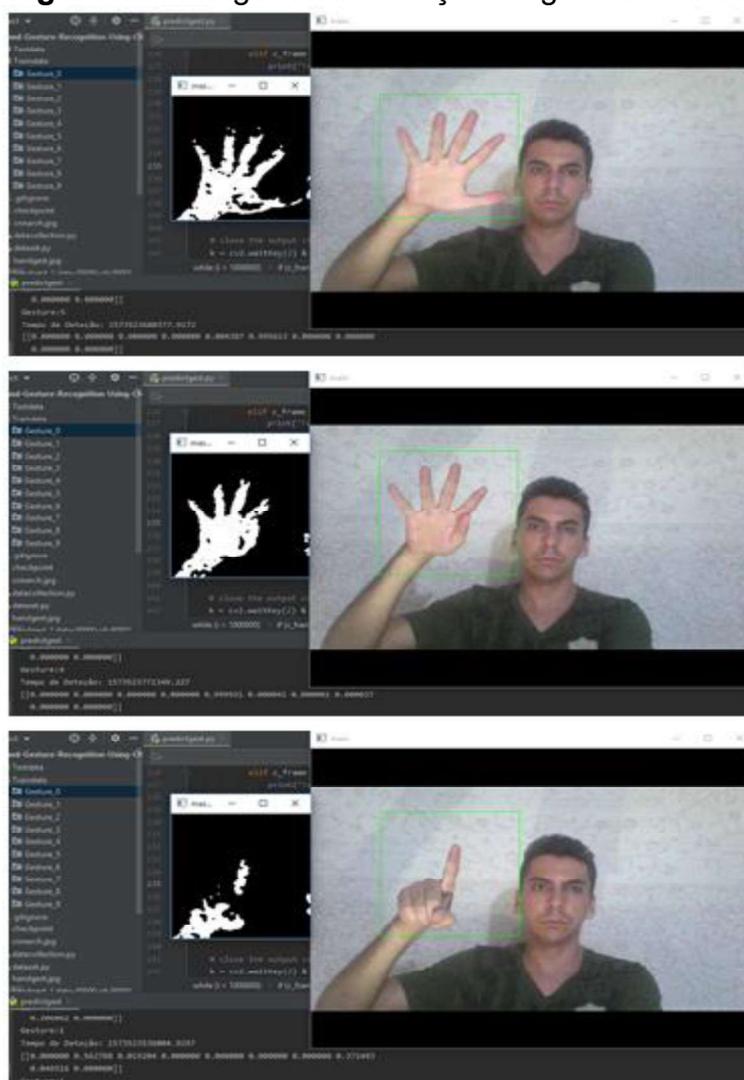
Gráfico 6 - Resultados da detecção de gestos usando CNN.



Fonte: (AUTOR, 2019).

Conclui-se que os resultados demonstrados no Gráfico 6 foram satisfatórios para a maioria dos gestos, mas acredita-se que os gestos de baixa detecção, tenha sido devido sua base de imagens de treinamento com má qualidade de captura desses gestos. A Figura 23 ilustra alguns detecções em tempo real utilizando CNN.

Figura 23 - Imagens da detecção de gestos usando CNN.



Fonte: (AUTOR, 2019).

Os resultados foram considerados satisfatórios devido a capacidade de processamento do notebook ser superior a Raspberry Pi. Realizou-se diversas tentativas de executar na Raspberry Pi, mas devido ao alto processamento e a baixa configuração do mesmo, onde resultou travamentos e más detecções.

4.5 Conexão entre Raspberry Pi e a NodeMCU

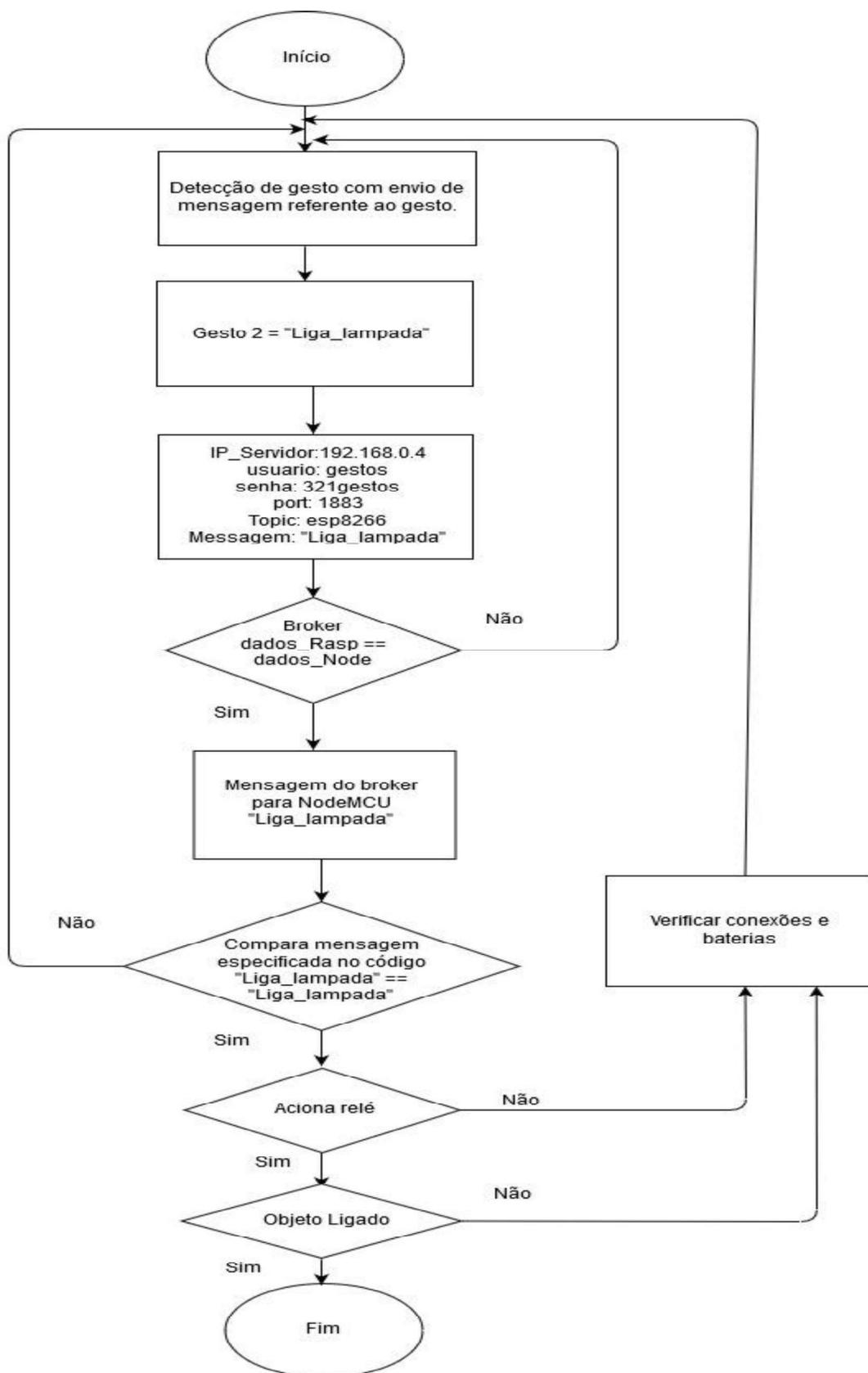
Definido os algoritmos para cada plataforma, de acordo com os resultados citados acima, realiza-se o acionamento de um objeto por meio desse gesto detectado, para isso na Raspberry Pi deixar-se um IP fixo para que sempre que conectar-se em uma rede wi-fi ela tenha o mesmo IP de conexão para não ter a necessidade de estar alterando o código na NodeMCU. Realizado o processo de ip fixo através de comandos no prompt de comando, instala-se o *broker Eclipse*

Mosquitto juntamente com o protocolo MQTT que atuará juntamente com servidor que receberá e enviará requisições para a NodeMCU que será o cliente assinante. Realizado a instalação do broker, cadastra-se um login, senha, nome do tópico referente a NodeMCU para cada mensagem recebida ou enviada e a porta de acesso, no caso 1883 que tem por padrão, para que a microcontroladora possa se conectar ao *broker*, assim possibilitando receber as mensagens de acionamento de acordo com cada gesto.

Tendo como observação que todas as configurações e instalações feitas, foram realizadas no prompt de comando da Raspberry PI, que se encontra no anexo A. Na NodeMCU especifica-se em seu código, login, senha, porta de acesso, o ip fixo do *broker* da Raspberry PI e o nome do tópico de cada publicação recebida ou enviada, cadastrada no *broker*. Tendo essas informações, ela tratará de especificar qual mensagem a receber ou enviar de acordo com o gesto especificado no código, onde por exemplo um gesto 2 seja uma mensagem “liga lâmpada” para acender uma lâmpada. E assim pode-se fazer para outros gestos determinando outras mensagens e conseqüentemente outros acionamentos.

Demonstrando-se em forma de fluxograma a conexão entre a Raspberry PI e a NodeMCU, inicia com o processo de detecção de gestos na Raspberry PI, detectado o gesto enviá-se uma mensagem referente aquele gesto juntamente com os dados cadastrados no *broker* para possível conexão com a microcontroladora, no processo seguinte *broker* realiza a comparação se os dados que ela tem cadastrados é igual a da NodeMCU, se sim, ela continua o processo de envio, caso contrário inicia um novo ciclo. Chegada a mensagem na microcontroladora, ela faz uma nova comparação se a mensagem recebida existe em sua base, se sim ela aciona o relé, em caso de não, reinicia um novo ciclo. AcionaCom isso chega ao fim de toda a execução. A Figura 24 ilustra a comunicação entre elas até o acionamento de um objeto.

Figura 24 - Passo a passo desde a conexão entre ambos os *hardwares* até o acionamento dos objetos.



Fonte: (AUTOR, 2019).

Diante dessa ação de conexão, o primeiro passo é realizar a detecção de forma mais precisa e satisfatória, com isso é realizado adaptações nos algoritmos com o intuito de melhor detectar e acionar um objeto. A Figura 25 ilustra uma detecção de gestos em tempo real.

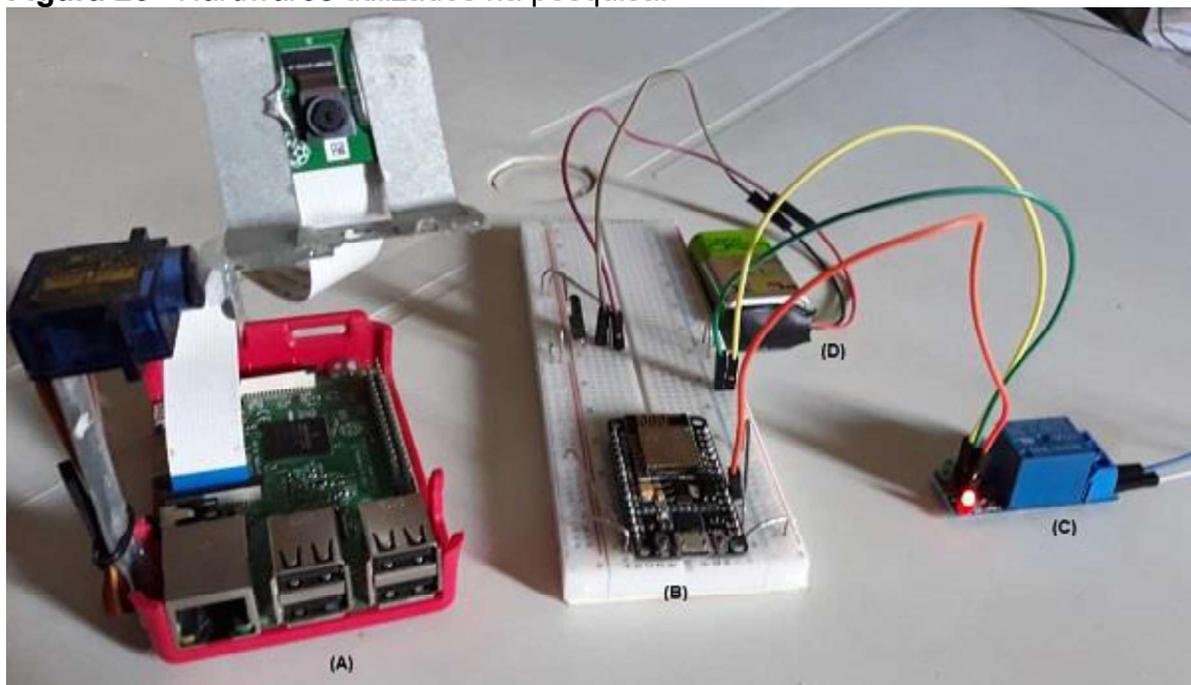
Figura 25 - Detecção de gestos em tempo real.



Fonte: (AUTOR, 2019).

A RaspCam acoplada em um suporte é responsável pela execução dos algoritmos de detecção sendo interpretadas em uma linguagem *Python*, onde consequentemente se instalou bibliotecas de processamento de imagem como OpenCV, Numpy, e MQTT para possível conexão ambos os *hardwares*. A NodeMCU é a responsável por se conectar ao wi-fi e ao *broker* executado na Raspberry PI com os devidos dados cadastrados. Ela tem como função receber os dados enviados do servidor para acionar o relé com a devida mensagem referente ao gesto. Consequentemente o módulo relé é o responsável por controlar o fluxo de energia de acordo com a ação da microcontroladora sendo ela de ligar ou desligar um objeto. A Figura 26 ilustra esses *hardwares* utilizados na pesquisa.

Figura 26 - *Hardwares* utilizados na pesquisa.



Fonte: (AUTOR, 2019).

Demonstra-se na Figura 26 os seguintes *Hardwares*:

- (A) Raspberry PI responsável pela detecção dos gestos;
- (B) NodeMCU responsável por impor ação ao módulo relé;
- (C) Módulo relé responsável pelo acionamento de um objeto;
- (D) Bateria de 9 volts responsável pela alimentação da NodeMCU;

5 CONCLUSÃO E TRABALHOS FUTUROS

Os estudos e testes aplicando visão computacional, IoT, o algoritmo *Haar Cascade* e CNN, comprovam que a detecção de gestos acionando objetos foram considerados satisfatórios de acordo com os resultados obtidos, tendo em vista esses resultados, pode-se dizer que, com um estudo mais detalhado, um treinamento melhor dos algoritmos voltada para uma plataforma mais robusta, como uma Raspberry PI 4 modelo B, com maior capacidade de processamento e memória, resultaria em uma detecção e acionamento mais eficaz, evitando possíveis limitações, observados no modelo utilizado.

5.1 Trabalhos Futuros

Sugere-se para trabalhos futuros um sistema de detecção de gestos e acionamento de objetos voltado para um ambiente de desenvolvimento como web, android e IOS. Esta aplicação teria uma contribuição significativa para pesquisas futuras, promovendo mais interação com usuários, comodidade e possíveis soluções de problemas de quem não pode se locomover para realizar alguma tarefa seja por deficiência ou distância.

REFERÊNCIAS

- ARAÚJO, Flávio H. D. et al. **Redes Neurais Convolucionais com Tensorflow: Teoria e Prática**. SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. III Escola Regional de Informática do Piauí. Livro Anais-Artigos e Minicursos, v. 1, p. 382-406, 2017.
- ARMTRONIX. **Install MQTT broker Mosquitto on Raspberry Pi**. 2016. Disponível em: < <https://www.instructables.com/id/Installing-MQTT-BrokerMosquitto-on-Raspberry-Pi/>>. Acesso em: 16 de Agosto 2019.
- BASTOS, Igor Leonardo Oliveira. **Reconhecimento de sinais da libras utilizando descritores de forma e redes neurais artificiais**. 2015. 105 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal da Bahia, Universidade Estadual de Feira de Santana, Salvador, 2015.
- BACKES, André Ricardo; SÁ JUNIOR, Jarbas Joaci de Mesquita. **Introdução à visão computacional usando Matlab**. Alta Books Editora, 2016.
- BARELLI, Felipe. **Introdução à visão computacional**. Casa do Código, 2018. 256 p.
- BONATTO, Alan Nunes. **Desenvolvimento de protótipo para reconhecimento de gestos de uma das mãos**. 2017. Monografia (Bacharelado em Engenharia Eletrônica) – Universidade de Caxias do Sul – UCS, 2017.
- CAMPOS, Roberto Augusto Freitas. **AUTOMAÇÃO RESIDENCIAL UTILIZANDO ARDUINO E APLICAÇÃO WEB**. 2014. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) - Universidade de Brasília, Brasília, 2014.
- COSMO, Daniel Luis. **Detecção de Pedestres Utilizando Descritores de Orientação do Gradiente e Auto Similaridade de Cor**. 2014. 83 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Espírito Santo, Brasil, 2014.
- CRUZ, Juliano EC; GUIMARÃES, Lamartine NF; SHIGUEMORI, Elcio H. Um estudo da detecção automática de campos de futebol de imagens aéreas e orbitais utilizando SVM e descritores HOG. In: **XII Workshop de Computação Aplicada. São José dos Campos:[sn]**. 2012.
- DIZEU, Liliane Correia Toscano de Brito; CAPORALI, Sueli Aparecida. **A Língua de Sinais Constituindo o Surdo como Sujeito**. Educ. Soc, v. 26, n. 91, p. 583-597, 2005.
- DE FILIPEFLOP. Guia Arduino Para Iniciantes. **Filipeflop**, 2018. Disponível em: < http://pedro.tuxcuiabano.com/wpcontent/uploads/2018/05/GUIA_ARDUINO_Filipeflop.pdf>. Acesso em: 5 de junho de 2019.
- DE MATTOS, Guilherme Caeiro. **PresentEye: Sistema de Controle de Presença por Reconhecimento Facial**. 2017. Trabalho de Conclusão de Curso (Bacharelado

em Sistemas de Informação) – Universidade Federal do Estado do Rio de Janeiro. Rio de Janeiro, 2017.

DE OLIVEIRA, Sérgio. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. Novatec Editora, 2017.

DE QUEIROZ, José Eustáquio Rangel; GOMES, Herman Martins. Introdução ao Processamento Digital de Imagens. **Revista RITA**, v. 8 , n. 1, 2001.

Eclipse Mosquitto. Disponível em: <<https://mosquitto.org/>>. Acesso em: 11 de Agosto de 2019.

FREITAS, Rodrigo Fernandes. **Sistema de Rastreamento da Mão Humana Utilizando Visão Artificial para Aplicações Embarcadas**. 2011. 88 f. Dissertação (Mestrado em Engenharia de Teleinformática.) - Universidade Federal do Ceará. Fortaleza, 2011.

GUEDES, André Bernardes Soares. **Reconhecimento de Gestos usando Redes Neurais Convolucionadas**. 2017. Monografia (Bacharelado em Engenharia de Software) - Universidade de Brasília, Brasília, 2017.

MANCINI, Mônica. Internet das Coisas: História, conceitos, aplicações e desafios. Disponível em: <https://pmisp.org.br/documents/acervo-arquivos/241-internet-das-coisashistoria-conceitos-aplicacoes-e-desafios/file>. Acesso em: 19 de maio de 2019, v. 27, n. 04, p. 2017, 2018.

MATOS, Saulo Neves. **Tutorial para criar um sistema de aquisição de dados remoto, alimentado por células fotovoltaicas**. 2015. Projeto de Pesquisa - Universidade Federal De Ouro Preto -UFOP.

MQTT. Disponível em: <<http://mqtt.org/>>. Acesso em: 5 de Agosto de 2019.

NETO, Elias Ximenes do Prado. **Deteção de gestos manuais utilizando câmeras de profundidade**. 2014. 153 f. Dissertação (Mestrado em Ciências) - Universidade de São Paulo, São Carlos, 2014.

PAVAN, Adilson Roberto; MODESTO, Fábio Alexandre Caravieri. **Reconhecimento de Gestos com Segmentação de Imagens Dinâmicas Aplicadas a Libras**. 2010

RAMOS. Luan Caius. Reconhecimento de Gestos usando Redes Neurais Artificiais. 2011. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) - Universidade de Brasília, Brasília, 2011.

RIBEIRO, Hebert Luchetti. **Reconhecimento de gestos usando segmentação de imagens dinâmicas de mãos baseada no modelo de mistura de gaussianas e cor de pele**. 2006. 144 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de São Paulo, São Carlos, 2006.

ROCHA, Patrick Francis Gomes. **Abordagem teórica e aplicabilidade de Detecção facial em Aplicativos Móveis com OpenCV e Google Android.** Trabalho de Conclusão de Curso. Fundação Educacional do Município de Assis, Assis, 2013.

SHAH, Raj. **Hand Gesture Recognition Using CNN, 2018.** Disponível em: <<https://github.com/raj-shah14/Hand-Gesture-Recognition-Using-CNN>> Acesso em: 12 de julho de 2019.

SANTOS, Carlos Alexandre Silva dos. **Reconhecimento de imagens de marcas de gado utilizando redes neurais convolucionais e máquinas de vetores de suporte.** 2017. 136 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Pampa, Alegrete, 2017.

SANTOS, Bruno P; et al. **Internet das Coisas: da Teoria à Prática.** 2016.

SILVA, Samuel Rocha. **Reconhecimento De Gestos Customizados Da Mão Em Tempo Real usando Aprendizado De Métricas E Grafos De Ação.** 2017. Programa De Pós Graduação Em Matemática. Universidade Federal De Alagoas. Maceió - AL

TANAJA, Devashish. **Hand Recognition, 2018.** Disponível em: <https://github.com/dtaneja123/Hand_Recognition> Acesso em: 20 de junho de 2019.

ANEXO A – COMANDOS DE CONFIGURAÇÕES E INSTALAÇÃO DO BROKER

Comandos para deixar IP fixo.

```
ifconfig
netstat -nr
sudo nano /etc/dhcpd.conf
```

No topo do arquivo dhcpd.conf, digite estas linhas:

```
interface {interface de rede que vc vai usar}
static ip_address={seu futuro ip fixo}/{máscara de rede}
static routers={ip do seu roteador}
static domain_name_servers={ip do seu roteador} 8.8.8.8 8.8.4.4
```

Salve o arquivo pressionando CTRL + X, depois S para salvar e Enter para sair do editor de arquivos Nano.

Instalação do Broker MQTT Mosquitto:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install mosquitto -y
sudo apt-get install mosquitto-clients -y
sudo nano /etc/mosquitto/mosquitto.conf
```

Dentro deste arquivo, localize e apague a seguinte linha:

```
include_dir /etc/mosquitto/conf.d
```

Após isto, inclua estas linhas:

```
allow_anonymous false
password_file /etc/mosquitto/pwfile
listener 1883
sudo mosquitto_passwd -c /etc/mosquitto/pwfile
```

{seu_nome_de_usuario}

Finalizado a instalação do MQTT

Realizando testes pós instalação

```
mosquitto_sub -d -u "nome dado ao subscribe, no caso usuário" -
P senha do usuário -t nome dado ao NodeMCU
mosquitto_pub -d -u login_usuario -P senha -t nomeNodeMCU -
m "mensagem que queira enviar ex: liga"
```

Comandos baseados em (ARMTRONIX, 2016).