Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica e Informática

Projeto e Implementação de uma Câmera Inteligente para Smart-Homes

Pablo Michell Gurjão da Silva

Campina Grande, PB
Dezembro de 2018

Pablo Michell Gurjão da Silva

Projeto e Implementação de uma Câmera Inteligente para Smart-Homes

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Área de Concentração: Processamento de Imagem

Orientador: Danilo Freire de Souza Santos

Campina Grande, PB
Dezembro de 2018

Pablo Michell Gurjão da Silva

Projeto e Implementação de uma Câmera Inteligente para Smart-Homes

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovado em ___/___ /___

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Danilo Freire de Souza Santos, D.Sc.
Universidade Federal de Campina Grande
Orientador

Você vai ter momentos ruins, mas eles sempre vão te acordar para as coisas boas que não estavas prestando atenção. (Filme: Gênio Indomável, 1997)

Agradecimentos

Agradeço Primeiramente a Deus por me permitir trilhar meu caminho da forma com que foi feito até aqui, apesar de todos dias de estresse e noites sem dormir, também tive alegrias, realizações e amizades que levarei para vida toda.

Agradeço a minha família por sempre estar ao meu lado e me proporcionar todas condições que eu precisei para que eu continuar minha jornada. Agradeço a minha namorada e amigos por ter me dado suporte, me aguentado e me aconselhado em meus piores momentos.

Agradeço ao Professor Edmar Gurjão por ser meu primeiro mentor dentro do curso, ao Professor Péricles Barros por me proporcionar oportunidades de trabalho que me moldaram pessoal e profissionalmente, ao Professor Danilo Santos por ter me orientado, tendo grande disponibilidade e atenção para com minha pessoa e sempre passando conhecimento nas áreas necessárias.

Em especial, agradeço a uma das mulheres mais fortes que já conheci e que sempre esteve ao meu lado, moldando a pessoa que me tornei, Dona Tereza, minha avó, a qual me aconselhou ao saber que eu tinha sido aprovado para o curso de engenharia elétrica, dizendo: ”Meu filho, tenha cuidado nesta profissão e não vá subir em postes, pois é perigoso”, mas que infelizmente partiu antes de me ver terminando esta fase da minha vida.

Não viva para que sua presença seja notada, mas para que sua falta seja sentida.
(Bob Marley)

Resumo

Este trabalho apresenta uma aplicação do conceito de Inteligência Artificial, no cotidiano das pessoas, tendo o intuito de trazer uma maior comodidade no dia a dia. O ser humano sempre primou pela segurança e comodidade, fazendo com que a busca de novas tecnologias o auxiliassem em sua proteção e, se possível, proporcionasse comodidade. Partindo desta observação, esse trabalho apresenta uma solução pautada em uma das sub áreas da Inteligência Artificial, a Visão Computacional. Logo, um algorítimo de Visão Computacional foi aplicado utilizando um microprocessador e uma unidade de processamento gráfico. Estes componentes recebem os dados de uma câmera e, fazem a identificação de objetos e monitoram a formação de um cenário, para uma atuar nele. Para o transporte sem fio das informações, foi utilizado um protocolo padrão para a Internet das Coisas, o MQTT, o qual facilita a rapidez e eficiência entre a comunicação dos objetos. Com esse trabalho, portanto, espera-se fazer a identificação de dois objetos, a criação de um cenário que os envolva e que seja possível uma atuação nele, quando identificado.

Palavras chave: Inteligência Artificial, Visão Computacional, Aprendizado de Máquina; Reconhecimento de Objetos; Aprendizado Profundo.

Abstract

The work presents an application of the Artificial Intelligence concept, in every people day, In order to provide greater comfort in everyday life. Humans always looked for safety and comfort, hoping that new technologies could help in their protection and, if possible, to provide convenience. Based on this observation, this work presents a solution focused on Artificial Intelligence sub-area, the Computer Vision. Thus, a Computer Vision algorithm was applyied using a microprocessor and a graphics processing unit. These components receive frames from a camera and, with these data, they make an identication of objects and also monitoring the creation of a scenario, to act on it. For wireless transmission, a standard Internet of Thing protocol, the MQTT was used, which improves speed and efficiency in the objects communication. With this work, therefore, it is expected to identify two objects,the creation of a scenario that involves these objects and that it is possible to act on it when identified.

Keywords: Artificial Inteligence; Computer Vision; Machine Learning; Objects Recognition, Deep Learning.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AI Inteligência Artificial

ML Aprendizado de Máquina

MQTT Message Queuing Telemetry Transport

RNA Redes Neurais Artificiais

Sumário

1 Introdução

O ser humano vem desde os primórdios adaptando-se a situações adversas e aprendendo com as coisas que o rodeia, para que as possa transformar ao seu bel prazer, desta forma, fazendo com que seu esforço na realização de atividades seja reduzido, ou até mesmo inexistente. Logo, foi assim que surgiram as maiores invenções da humanidade, como a agricultura, controle do fogo, a roda, o computador, etc.

A partir do momento que o homem começou a utilizar o conceito de propriedade privada, ele passou a ter seu território. Não querendo que seus bens fossem usurpados ou tomados, surgiu, também, a necessidade de proteção. A necessidade da busca pela sobrevivência de forma direta foi parcialmente passada para um segundo plano, já que a hostilidade do ambiente vivido começava a diminuir. Assim, para que pudesse ser mantido com ele suas conquistas ao longo da vida e principalmente um local onde fosse possível relaxar e gozar da tranquilidade de não se preocupar com violência contra si próprio ou de sua família, era necessário começar a pensar em uma forma de assegurar suas regalias.

O monitoramento de determinado local é necessário para o auxílio a segurança e controle de uma situação. Com o avanço da tecnologia, foi possível realizá-lo através da implantação de câmeras nos ambientes, o que serviu de ajuda para investigações e acompanhamento de situações a distância. Mas apenas a capacidade de monitorar um ambiente e/ou armazenar em vídeo a observação não seria suficiente para garantir o conforto e proteção tão almejados. Logo, ainda era preciso a execução de inúmeras tarefas repetitivas como, por exemplo, se locomover a um interfone, perguntar quem estava chamando e depois abrir ou não o portão.

Assim, com a evolução do poder de processamento de dados aliado com o barateamento da tecnologia, foi possível optar por novas soluções, como o uso de algorítimos de aprendizado de máquina [1] ou visão computacional, que mais uma vez poderia garantir maior segurança e conforto ao ser humano.

O aprendizado de máquina [1], ou *Machine Learnig*, é uma área da computação baseada, na maioria das vezes, em redes neurais artificiais [7], que são construídas com o intuito de "imitar" o cérebro humano, sendo capaz de identificar padrões e aprender a tomar decisões. Dentro do estudo deste aprendizado existem vários outros subcampos, entre eles a visão computacional.

Pautado neste contexto, foi construído este Trabalho de Conclusão de Curso, tendo como objetivo construir um ambiente experimental que seja capaz de replicar a habilidade de visão do ser humano em prol da segurança e conforto das pessoas, fazendo jus mais uma vez a ânsia de comodidade e proteção.

1.1 Visão Computacional

"A visão computacional é o processo de modelagem e replicação da visão humana usando software e hardware."(EQUIPE DATA SCIENCE ACADEMY, 2017) [2]. É importante entender a diferença entre processar a imagem e a visão computacional, e assim saber que o processamento de imagens é uma das etapas necessárias para a replicação da habilidade humana de poder ver algo, entendê-lo e saber o que pode ser feito com o mesmo, sendo essa replicação, a visão computacional.

De acordo com a definição dada anteriormente, temos uma abertura para uma vasta gama de aplicações dado que a partir deste estudo é possível "ensinar" um computador a analisar determinado ambiente e com o processamento da imagem adquirido, saber o que se passa neste local. Com

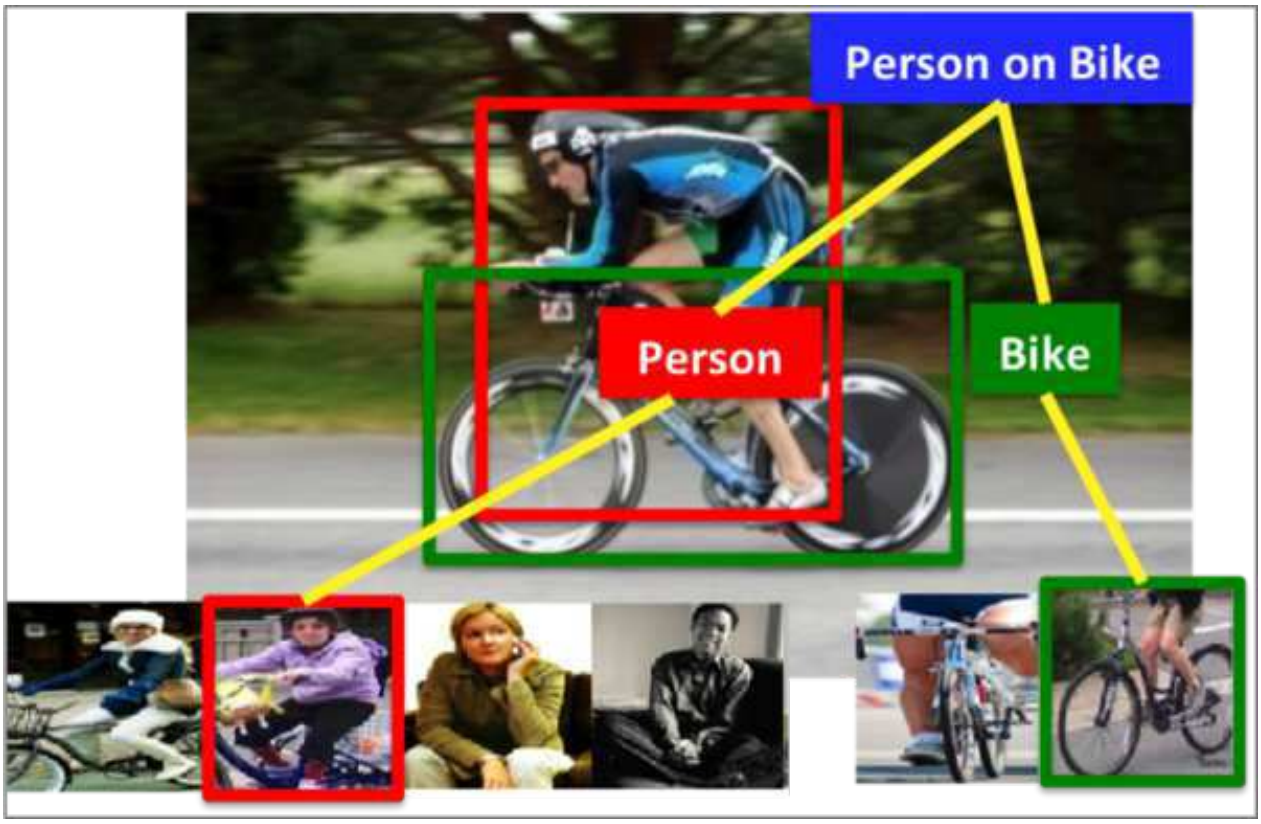

Figura 1.1: Exemplo da utilização de visão computacional
Fonte: cs.stanford.edu

a criação de cenários, se identificados, o mesmo poderá tomar uma decisões. Na Figura 1.1 temos uma imagem da identificação de diversos objetos, mostrando o poder deste conceito.

O ser humano utiliza cerca de dois terços de sua capacidade cerebral apenas para o processamento da visão [13], sendo assim, é notório a necessidade de computadores robustos para que esta atividade seja realizada. A partir desta complexidade outros dois grandes campos foram criados, a *Big Data* e Processamento Paralelo em Unidades Gráficas de Processamento - GPU's que fazem com que processamento paralelo seja mais rápido, mais barato e mais poderoso.

Com este ideal a Intel criou acessório de computação neural, o Intel® Movidius™ [5], que tem o intuito de "Permitir prototipagem rápida, validação e implantação de aplicações de inferência de rede neural profunda (DNN) na borda."(INTEL®, 2018). Este acessório é bastante pequeno e provê um alto desempenho no processamento de dados, o que torna possível e viável a utilização da visão computacional em vários cenários.

Para fins de maiores entendimentos, é possível ilustrar alguns cenários a serem observados, nos casos apresentados em forma de cenário a seguir, temos algumas situações corriqueiras do dia a dia, descritas nas Tabelas 1, 2 e 3.

Ambientação	Uma pessoa dirige-se a entrada/saída de determinado cômodo e depare-se com uma porta trancada.
Solução Inteligente	Através de um algorítimo de visão computacional, ao ser identificado que a intenção da pessoa é entrar/sair de determinado local, a trava da porta seria aberta.

Tabela 1: Cenário 1

Ambientação	Uma pessoa vai para uma sala para descansar e quando senta no sofá, nota que está sem o controle da televisão ou do aparelho de som.
Solução Inteligente	Através de um algorítimo de visão computacional, ao ser identificado que a intenção da pessoa é ligar o aparelho televisor ou um som ambiente, ao ser detectado que a pessoa encontra-se no sofá, a televisão ou um som ambiente seriam ligados.

Tabela 2: Cenário 2

Ambientação	Uma pessoa estava assistindo televisão e sai do sofá para ir dormir, mas só depois que se deita que lembra da tv ligada.
Solução Inteligente	Através de um algorítimo de visão computacional, ao ser identificado que a pessoa não encontra-se mais na sala, neste caso a pessoa não está perto da tv ou do sofá, a televisão será desligada, juntamente com a lâmpada, se acesa.

Tabela 3: Cenário 3

1.2 Justificativa e Problemática

Tendo em vista que a necessidade de segurança inteligente e o perfil de busca pelo conforto do ser-humano, surge a problemática de desenvolver um ambiente com Inteligência Artificial [3] que tenha um arcabouço de decisão e seja capaz de realizar tarefas através da análise do contexto visual do ambiente.

É necessário elencar o que e quais artifícios poderiam ajudar a solucionar este aspecto. Pensando em um novo dispositivo que seja voltado para o processamento da imagem, que disponha da capacidade de analisar os fatores envolvidos e que possa atuar no ambiente ambiente através das constatações obtidas pela observação do ambiente, é possível desenvolve-lo com algumas tecnologias acessíveis, eficientes e que sejam bastante conhecidas pelo mercado.

1.2.1 Proposta de Solução

No tópico anterior foi mostrado alguns possíveis cenários, ilustrados por uma ambientação que caracteriza a problemática identificada e motivadora deste trabalho, juntamente com esta, também

foi apresentado uma possível solução a ser implementada de forma inteligente.

Ambientação	Uma pessoa vai para uma sala para descansar e quando senta no sofá, nota que está sem o controle da televisão ou do aparelho de som.
Solução Inteligente	Através de um algorítimo de visão computacional, ao ser identificado que a pessoa está próxima ao sofá dentro de um determinado período de tempo, o televisor é ligado e ao ser detectado que a pessoa não encontra-se mais próximo ao sofá, a televisão será desligada.

Tabela 4: Cenário Utilizado

Para o desenvolvimento deste trabalho escolheu-se como base de ambientação e aplicação de solução inteligente, uma ambientação baseada no problema ilustrado pela Tabela 2. O cenário pensado e montado é exposto e detalhado na Tabela 4.

1.3 Objetivos

Este Trabalho de Conclusão de Curso tem como objetivo projetar e implementar uma câmera inteligente com foco na análise de cenários, no qual esta será capaz de atuar em um ambiente a partir da identificação destes cenários. Logo, este trabalho pretende desenvolver uma estrutura que capta a imagem a partir de uma câmera, processa esta informação através do acessório de computação neural e com a identificação deste cenário poder atuar no mesmo. Isto permitirá um monitoramento inteligente que analisará o ambiente e atuará a partir de uma situação identificada.

1.4 Organização do Documento

Para relatar o conteúdo utilizado, criar uma base teórica sobre o que foi feito, fixar alguns conceitos aplicados no processo de resolução da problemática e expor como foi o processo de criação, desenvolvimento e implementação da solução proposta supracitada, foi preciso seguir uma ordem de execução, assim, será exposta neste documento uma organização de ideias para que a apresentação se torne didaticamente bem explicitada. Diante do exposto, a organização do documento é descrita a seguir.

O Capítulo 2 apresenta uma breve revisão bibliográfica, que visa explanar e explicar os conceitos e embasamento teórico da proposta. A inteligencia artificial, por ser o foco principal da solução, trás em sua composição mais conceitos que foram necessários para a implementação, como o aprendizado de máquina e a visão computacional. Por último mas não menos importante, neste capítulo também é explicitado as tecnologias utilizadas, assim como, algumas de suas especificações.

Já no Capítulo 3, será apresentado de forma a solução proposta supracitada na Tabela 4. Neste é explicado quais as métricas utilizadas para a escolha de cada tecnologia envolvida no processo, assim como, é detalhado o princípio de funcionamento da arquitetura montada e quais os procedimentos adotados para o desenvolvimento.

Para o Capítulo 4, foi reservado o processo de validação do ambiente criado, assim como, os testes utilizados para garantia de tal resultado.

Por último, temos a conclusão deste trabalho. Neste último tópico são abordados os resultados finais, são feitos alguns comentários de cunho reflexivo em relação as dificuldades, resolvidas e em aberto, deparadas ao decorrer do desenvolvimento assim como perspectivas futuras acerca desta tecnologia, no contexto de aplicação em casas inteligentes.

2 Revisão Bibliográfica

Para o entendimento deste trabalho é necessário aprofundar os conceitos e ferramentas utilizadas. Assim, a partir da proposta de um ambiente com inteligência artificial, AI [3], é necessário explicitar do que é composto uma estrutura que seja capaz de trabalhar de forma inteligente.

Além dos conceitos base para o desenvolvimento, também é necessário abordar os protocolos utilizados, como o MQTT [4] e aparatos tecnológicos como o Intel® Movidius™ [5], RaspBerry Pi 3B [6] e o NodeMCU [8].

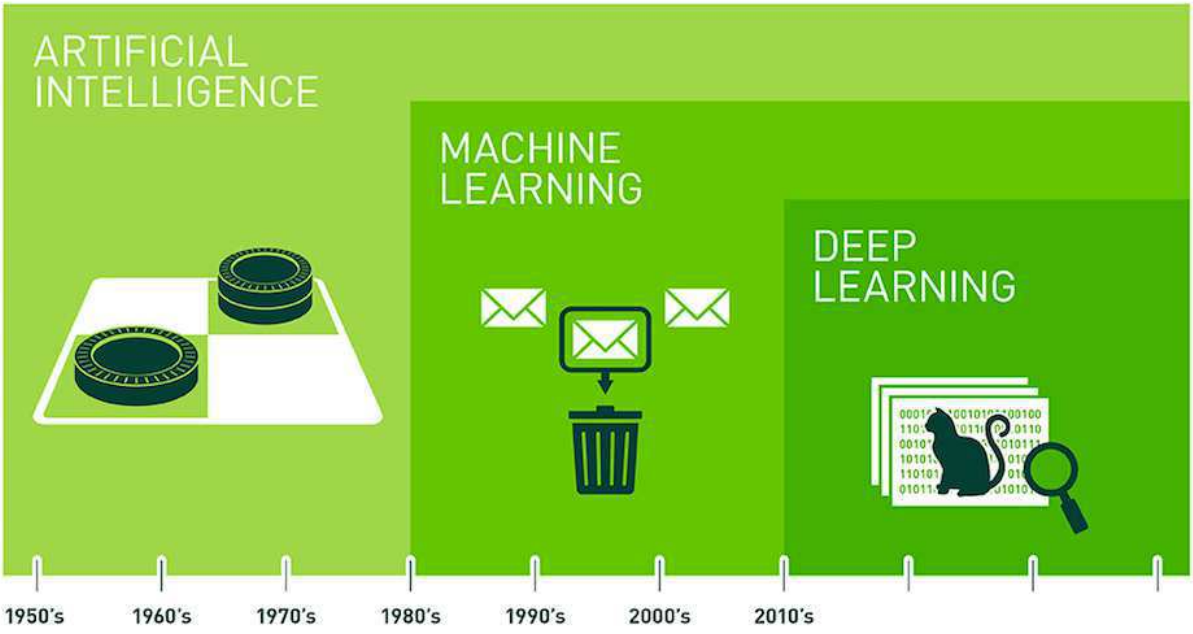

Figura 2.1: Representação da Inteligência Artificial e Algumas Sub-Áreas

2.1 Inteligência Artificial

Para o desenvolvimento deste trabalho, foi utilizado o conceito de AI limitada, ou seja, que é possível ensinar e/ou treinar uma máquina para que ela execute atividades que necessitam de "INTELIGÊNCIA" para serem realizadas, assim, com este artifício, tais atividades que eram essencialmente destinada aos serem humanos, podem ser executadas por máquinas.

Na Figura 2.1 [1], tem-se uma estrutura de AI [3] demonstrada em forma de diagrama. Está organização tem como principal característica executar atividades com inteligência e, em certos casos, tomar alguma decisão diante de algum cenário e/ou circunstância detectada. Para que esta incumbência seja realizada é necessário uma estrutura que utilize aprendizado de máquina.

Assim, um algorítimo de AI [3] é desenvolvido com base no conceito de aprendizado de máquina, sendo então a mais alta camada de abstração conceitual.

[1] Figura disponível em: https://medium.com/data-science-brigade/a-diferen%C3%A7a-entre-intelig%C3%AAncia-artificial-machine-learning-e-deep-learning-930b5cc2aa42

2.2 Aprendizado de Máquina

O aprendizado de máquina é uma técnica baseada em algorítimos como árvore de aprendizado, agrupamento, programação lógica indutiva, aprendizados forçados e principalmente redes neurais artificiais, sendo este ultimo, a principal base do aprendizado de máquina.

Redes Neurais Artificiais [7], são modelos computacionais que tem o intuito de simular o sistema nervoso central de um ser-humano. Obviamente estes modelos não são cem porcento fieis, devido a tamanha complexidade deste sistema.

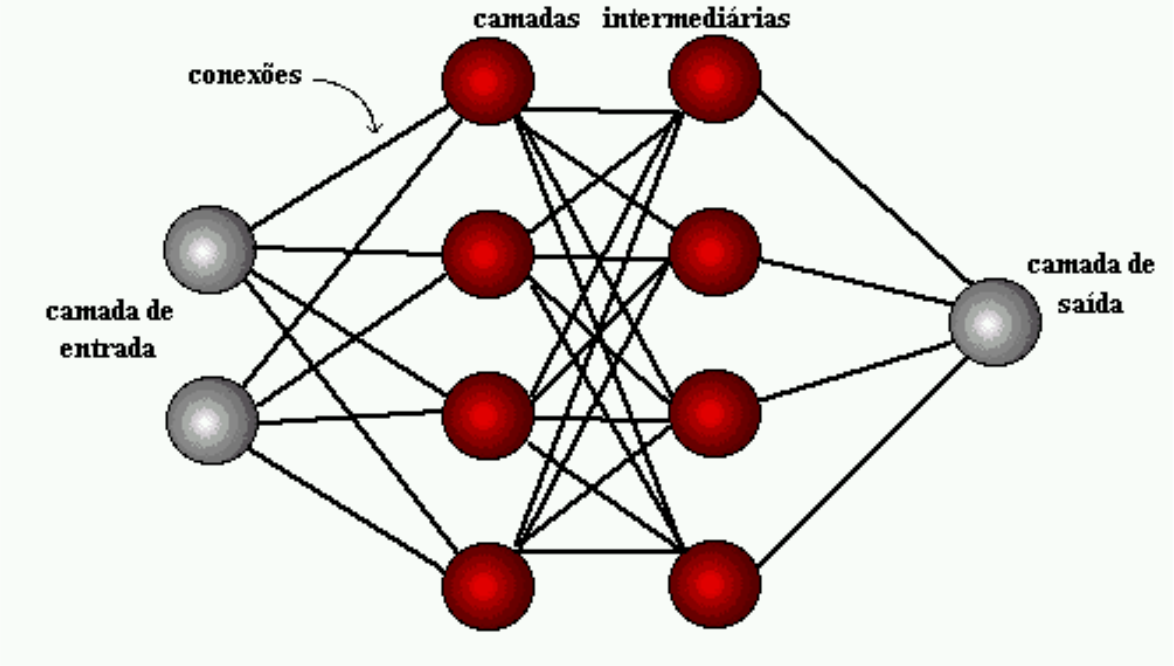

Figura 2.2: Representação das partes de uma Rede Neural

Como visto na Figura 2.2 [2], a estrutura de uma Rede Neural Artificial [7] é composta por basicamente três camadas:

- Camada de Entrada - CE: Local onde os dados e/ou potenciais padrões são expostos ao sistema;
- Camadas Intermediárias - CI: Onde o processamento é feito, como reconhecimento dos padrões e classificação por características;
- Camada de Saída - CS: Destino final dos dados pós-processados, ou seja, classificados por padrões e/ou características.

[2]Figura disponível em: http://conteudo.icmc.usp.br/pessoas/andre/research/neural/

Para maior entendimento desta estrutura precisamos saber que cada componente da camada de entrada está relacionado aos estados das camadas intermediárias através de um ”peso”ou influência atribuída a cada conexão. Assim,

$$CI(n) = PesodaConexão(n) * CE(n-1) \tag{1}$$

A utilização destes algorítimos também é algo importante de se notar, para o aprendizado de máquina existem duas principais formas de uso destas técnicas, assim, dependendo da problemática e/ou dos dados disponíveis, as principais formas para a obtenção da aprendizagem podem ser:

- Aprendizado Supervisionado[12]: Os problemas relacionados com este perfil de aprendizado, são classificados como problemas de ”regressão”ou ”classificação”. A Figura[3] 2.3 expõe a ilustração destes dois tipos de classificação. Assim, podemos entender este método como um aprendizado que dispõe de um determinado conjunto e é desejado classifica-lo de acordo com uma saída já conhecida.

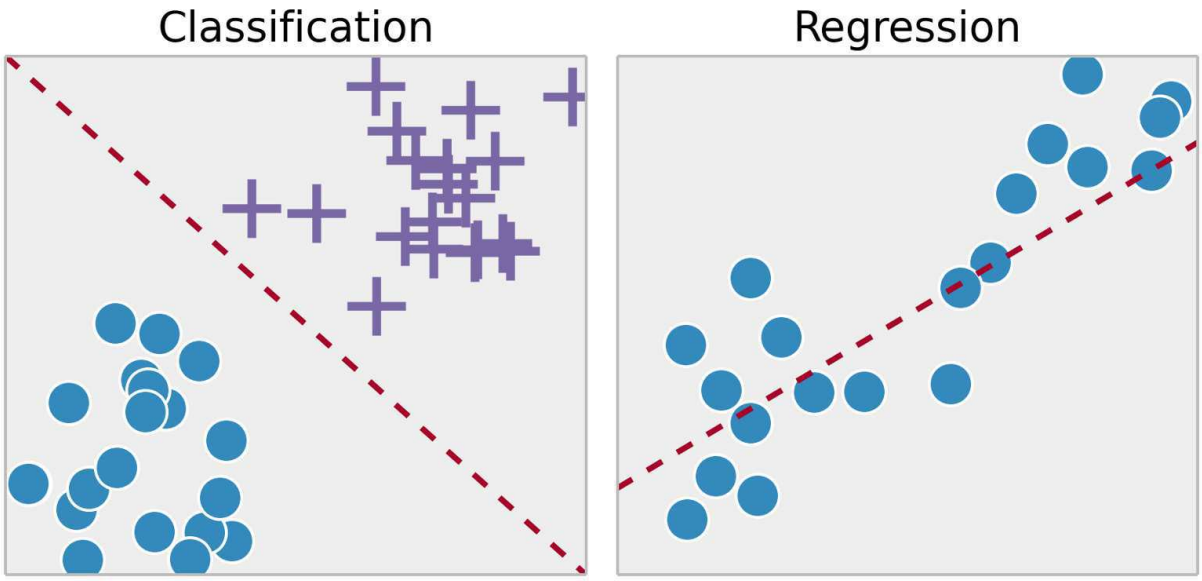

Figura 2.3: Classificação e Regressão

Exemplificando os tipos apresentados na Figura 2.3, temos por classificação um problema que nos apresenta, duas saídas possíveis, ou o símbolo de cruz ou de bola e então é necessário diferi-los dado sua posição e tamanho. Para um problema de regressão, podemos pensar na identificação de padrões, como determinada disposição de bolas que estão distribuídas em um caminho semelhante, por exemplo, a uma reta.

- Aprendizado Não-Supervisionado[12]: Para este tipo de aprendizado, tem-se por característica principal o agrupamento de dados com relação a uma determinada característica,

[3]Figura disponível em:https://medium.com/opensanca/aprendizagem-de-maquina-supervisionada-ou-n%C3%A3o-supervisionada-7d01f78cd80a

por exemplo, se é necessário diminuir o número de dimensões em um conjunto para que apenas as variáveis mais significativas entrem na análise, este tipo de aprendizado é utilizado, ou seja, esta técnica é usada quando não se tem, ou é muito pouca a ideia da saída. Na Figura

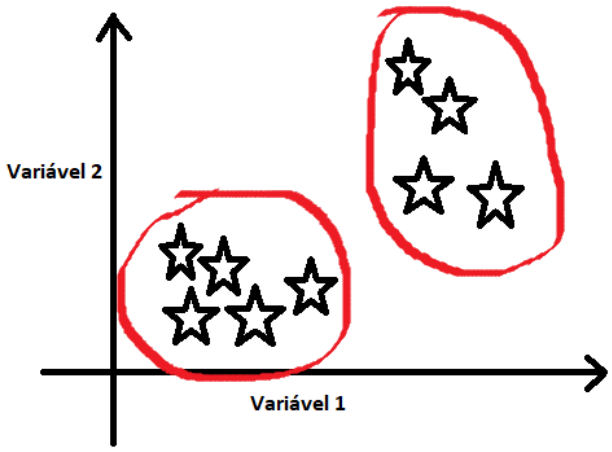

Figura 2.4: Aprendizado Não-Supervisionado

2.4 é mostrado o agrupamento de mesmos elementos em dois diferentes conjuntos, pois estes foram agrupados com relação a proximidade um com outro e/ou posição.

2.3 *Deep Learning*

”Falando de uma forma bem simples de entender, *Deep Learning* (aprendizagem profunda, em português) é um tema emergente dentro do campo da Inteligência Artificial. Uma subcategoria de aprendizado de máquina que diz respeito a oportunidades de aprendizagem profundas com o uso de redes neurais para melhorar as coisas, tais como reconhecimento de fala, visão computacional e processamento de linguagem natural.”(GAEA CONSULTING, 2018). A partir desta definição podemos ter uma ideia mais ampla do que este tema vem a ser e notar sua importância para a atribuição do termo ”Inteligência”para a chamada inteligência artificial.

A partir da evolução do processamento de dados, foi que a aprendizagem profunda pôde ser aplicada e, então, obter avanços como reconhecimento de imagens e tradução de voz em tempo real.

Com a ilustração apresentada na Figura 2.5 [4] podemos notar a razão que o Aprendizado Profundo vem sendo largamente utilizado. A análise é bastante simples, a performance continua crescente para uma grande quantidade de dados, pois com a disposição de muitas informações, o *BIG DATA*, é necessário uma técnica que seja eficiente não só em pequenas quantidades de dados, mas sim que seja suficientemente capaz de manter o desempenho quando o volume de dados também aumenta.

É notório que os algorítimos mais antigos de aprendizado de máquina não tem seu desempenho aumentado a partir de certa quantidade de dados, sendo assim, limitados de certa forma, essa ca-

[4]Figura disponível em: https://machinelearning-blog.com/2017/11/03/erster-blogbeitrag/

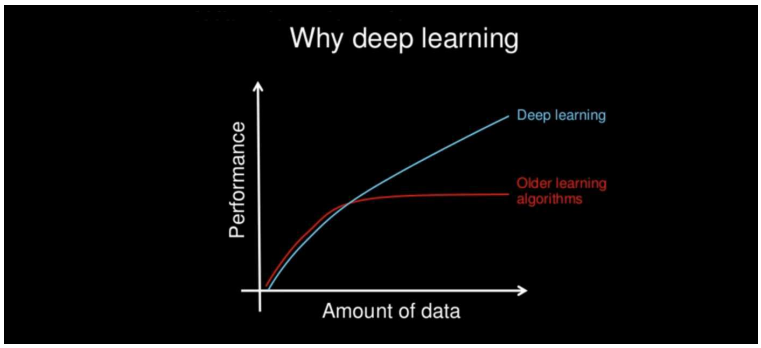

Figura 2.5: Vantagem da Aprendizagem Profunda

racterística faz com que o aprendizado profundo seja mais vantajoso quando dispomos de muitos dados.

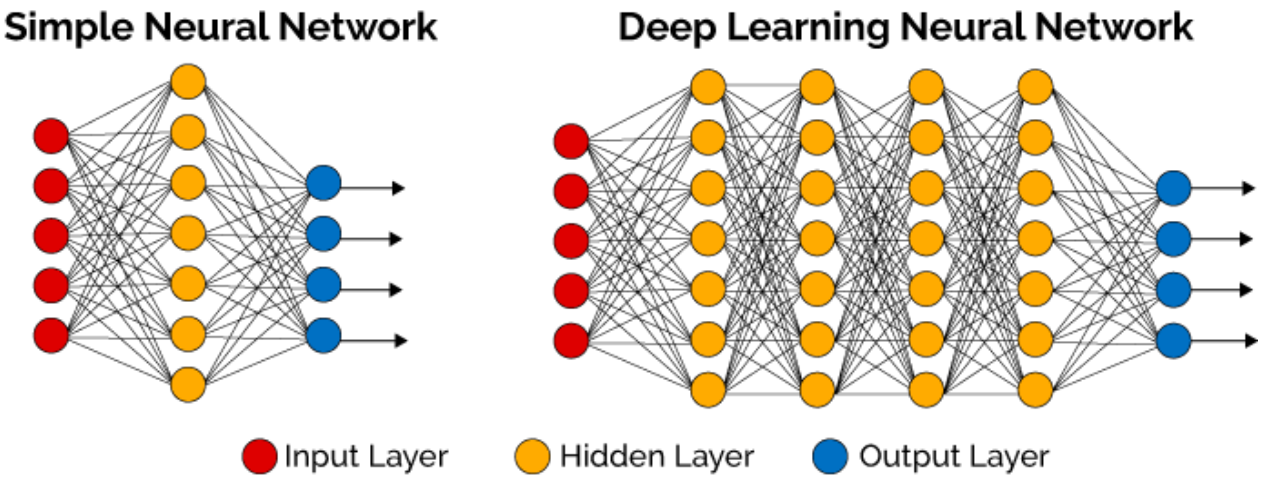

Figura 2.6: Diferença entre Aprendizado Profundo e outro Algorítimo de Rede Neural

Na figura 2.6 [5], é trazida uma imagem que torna possível aferir que em uma simples estrutura de redes neurais, são presentes poucas camadas intermediárias, já uma estrutura de Aprendizado Profundo de redes neurais, são envolvidas muito mais camadas intermediarias, justificando a alta eficiência, mostrada na Figura 2.5, de um método comparado ao outro.

2.4 MQTT

O MQTT, ou *Message Queuing Telemetry Transport*, é um protocolo de comunicação baseado na estrutura de *Publish Subscribe*. É um protocolo leve e que pode ser implementado em dispositivos com altos níveis de restrição, principalmente com limitações de banda e latência.

Focado em transporte de mensagens simples, capaz de ser executado em redes TCP/IP, hoje é o protocolo mais usado nas aplicações *Machine To Machine*, ou de máquina para máquina, devido

[5]Figura disponível em: https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351

ao seu pouco requerimento de uso de banda e sua capacidade de ser usado mesmo em hardwares com menos capacidade

Dentro deste protocolo existem certos conceitos que necessitam serem fixados:

- *Broker*: É o responsável pela recepção e distribuição das mensagens publicadas pelos os clientes conectados a ele pela rede. Ele recebe mensagens publicadas pelos clientes e as publica para outros clientes que estão inscritos no tópico da mensagem publicada;
- Cliente: É um objeto que pode publicar alguma informação em determinado tópico e/ou receber mensagens, através do *broker*, pelos tópicos ao qual está inscrito.

O funcionamento deste protocolo pode ser explicado da seguinte forma, um Cliente publica em um tópico específico e esta publicação é recebida pelo *broker*, em algum momento outro cliente se inscreve nesse tópico e recebe do *broker* a mensagem publicada no tópico que o primeiro cliente publicou.

Assim, este padrão passa a se repetir, diferentemente de outros protocolos que usam comunicação P2P, ou *peer-to-peer*, em que há a comunicação direta entre emissor e receptor da mensagem, este protocolo tem o *broker* como peça central e essencial para o funcionamento do protocolo. Desta forma, pode-se ter controle de quais mensagens podem ser endereçadas para determinado Cliente.

2.5 Tecnologias Relacionadas

Neste tópico estão explicitados os *hardwares* utilizados para resolução da problemática proposta.

2.5.1 RaspBerry Pi 3B

O Raspberry Pi 3 Model B [6], mostrado na Figura [6] 2.7, é o mais novo modelo da terceira geração da linha. Este dispositivo é um computador do tamanho de um cartão que é capaz de reproduzir vídeos e conectar teclado, mouse e monitor em sua estrutura.

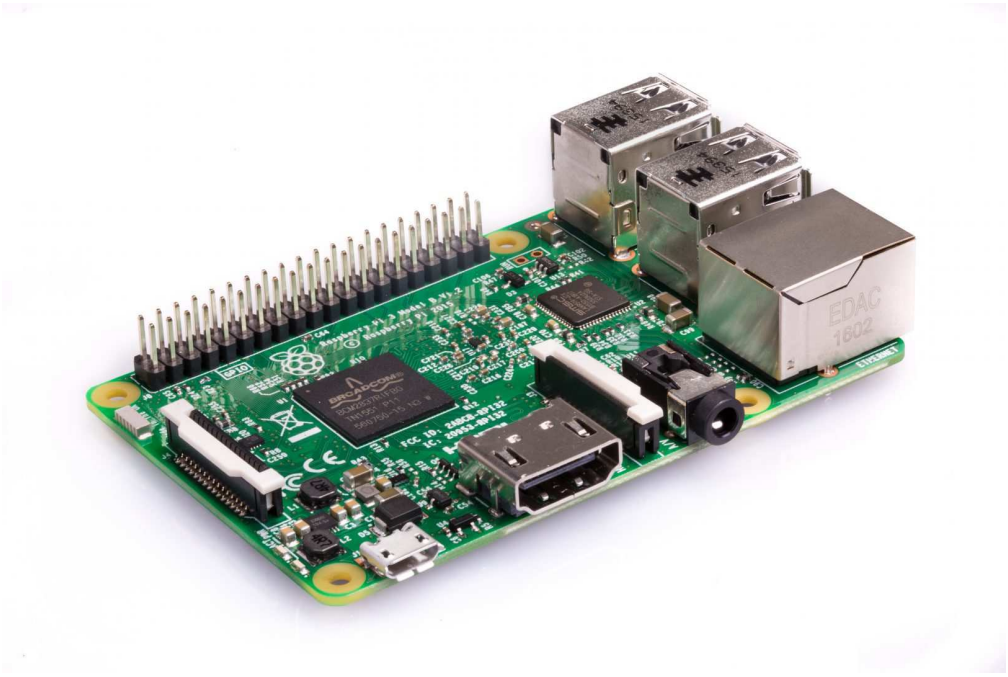

Figura 2.7: Raspberry Pi 3B

- *Quad Core 1.2GHz Broadcom BCM2837 64bit CPU*;
- 1GB RAM;
- BCM43438 *wireless LAN* e *Bluetooth Low Energy* (BLE) na própria placa;
- conector *Ethernet*;
- 40 pinos de extenção GPIO;
- 4 USB em 2 portas;
- Saída estéreo de 4 polos e porta de vídeo composto;

[6]Figura disponível em: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

- HDMI tamanho normal;
- Porta da câmera CSI para conectar uma câmera Raspberry Pi;
- Porta de exibição DSI para conectar uma tela sensível ao toque do Raspberry Pi;
- Porta Micro SD para carregar o sistema operacional e armazenar dados;
- Fonte de energia Micro USB comutada atualizada até 2,5A.

2.5.2 Intel® Movidius™

Este pequeno acessório de *Deep Learning* mostrado na Figura 2.8 permite a aprendizagem e programação da AI de forma facilitada. Ele é alimentado pela mesma unidade VPU (*Intel® Movidius ™ Vision Processing Unit*) de alto desempenho que pode ser encontrada em milhões de câmeras de segurança inteligentes, drones controlados por gestos e equipamentos de visão industrial.

Figura 2.8: Intel® Movidius™

As características deste acessório estão listadas abaixo.

- Processador: Unidade de processamento Myriad ™ 2 Vision (VPU) Intel® Movidius ™
- Estruturas suportadas: *TensorFlow , Caffe*
- Conectividade: USB 3.0 Type-A
- Dimensões do dispositivo USB: 2,85 pol. X 1,06 pol. X 0,55 pol. (72,5 mm x 27 mm x 14 mm)
- Temperatura de operação: 0 C a 40 C

Para que o Intel® Movidius™ possa ser utilizado é necessário que alguns critérios sejam atendidos, assim, deve-se dispor dos seguintes requisitos mínimos de sistema para utilização deste acessório:

- Computador x86-64 executando Ubuntu 16.04 ou Raspberry Pi 3 Modelo B
- Porta USB 2.0 tipo A (USB 3.0 recomendado)
- 1 GB de RAM
- 4 GB de espaço de armazenamento gratuito

2.5.3 NodeMCU

O NodeMCU é um microcontrolador que tem suporte para IDE arduino e também a linguagem de programação LUA, suas características seguem listadas abaixo. Na Figura 2.9 temos uma imagem deste microcontrolador [7] mostrando as funções de sua pinagem.

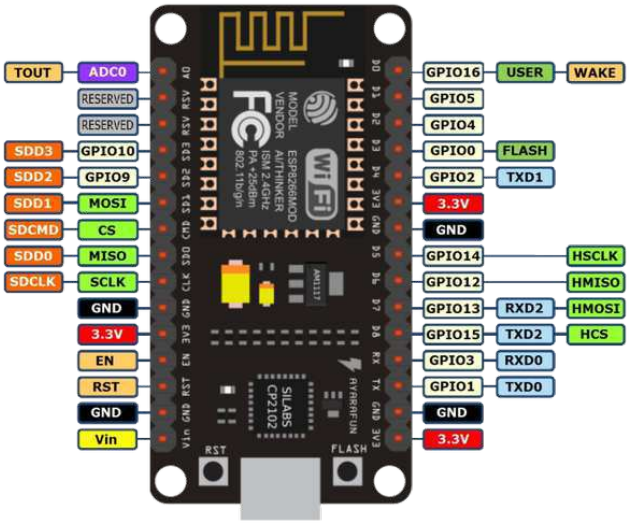

Figura 2.9: NodeMCU

- Wireless padrão 802.11 b/g/n
- Antena embutida
- Conector micro-usb
- Modos de operação: STA/AP/STA+AP
- Suporta 5 conexões TCP/IP
- Portas GPIO: 11
- GPIO com funções de PWM, I2C, SPI, etc

[7] Especificações disponíveis em:
https://www.electrodragon.com/w/ESP-12F_ESP8266_Wifi_Board
https://bennthomsen.wordpress.com/iot/iot-things/esp8266-wifi-soc/esp8266-getting-started-with-arduino-ide/

- Tensão de operação: 4,5 9V
- Taxa de transferência: 110-460800bps
- Suporta Upgrade remoto de firmware
- Conversor analógico digital (ADC)
- Distância entre pinos: 2,54mm
- Dimensões: 49 x 25,5 x 7 mm

3 Solução Proposta

A partir da problemática levantada, dos conceitos explanados no capítulo anterior e a disponibilidade das tecnologias supracitadas, foi possível construir uma solução factível, a qual será explicada com mais detalhes nos tópicos seguintes.

3.1 Raspberry PI 3B

Para a construção de um ecossistema que execute o que foi proposto, foi necessário a escolha de um microprocessador. Para esta escolha foi levado em conta critérios como:

- **Praticidade:** O microprocessador deve ter uma arquitetura para acesso prático as funcionalidades necessitadas nas aplicações;
- **Custo:** O pensamento que uma boa solução não é avaliada apenas pelo seu desempenho, mas sim pelo conjunto de custo X desempenho, assim, a solução deve ser a menos onerosa possível;
- **Disponibilidade e Suporte:** Para uma solução satisfatória é importante se ter uma estrutura que tenha uma base compatível com o máximo de tecnologias disponíveis e que disponha de um bom suporte para atualizações;

Analizando alguns microprocessadores disponíveis no mercado, o Raspberry Pi 3B [6] supriu os três critérios mostrados acima, por ter um baixo custo para o que se propõe, por ser *Open Source*, tem um bom suporte e arcabouço de aplicações e tem uma disposição física bastante intuitiva e acessível.

3.2 Intel® Movidius™

Neste ecossistema era necessário uma unidade que possa fazer o processamento da imagem adquirida de forma rápida e eficiente, ou seja, uma unidade especializada para uso em algorítimos de visão computacional.

Para suprir esta necessidade foi escolhido o Intel® Movidius™ [5], que dispõe desta capacidade de processamento e é uma ferramente que pode ser usada de forma gratuita após sua aquisição, além de poder ser usado *offline*.

3.3 NodeMCU

Para ser possível a atuação no cenário de uma forma prática e bastante acessível, foi necessário um microcontrolador que seria capaz de receber o gatilho do microprocessador, de preferência via *Wifi*, assim, necessitaria disponibilizar uma antena e compactuar dos critérios explicitados para a escolha do microprocessador.

Assim, um dispositivo que vem a um certo tempo tendo bastante notoriedade por seu baixo custo, robustez e aplicabilidade, foi escolhido o NodeMCU [8].

3.4 Arquitetura do Sistema

Para a construção da solução, utilizou-se como centro do trabalho e desenvolvimento a configuração experimental mostrada na Figura 3.1, está é composta pelo microprocessador Raspberry PI 3B [6], um acessório Intel® Movidius™ [5] e uma câmera para a captura de imagem, neste caso uma câmera própria do microprocessador.

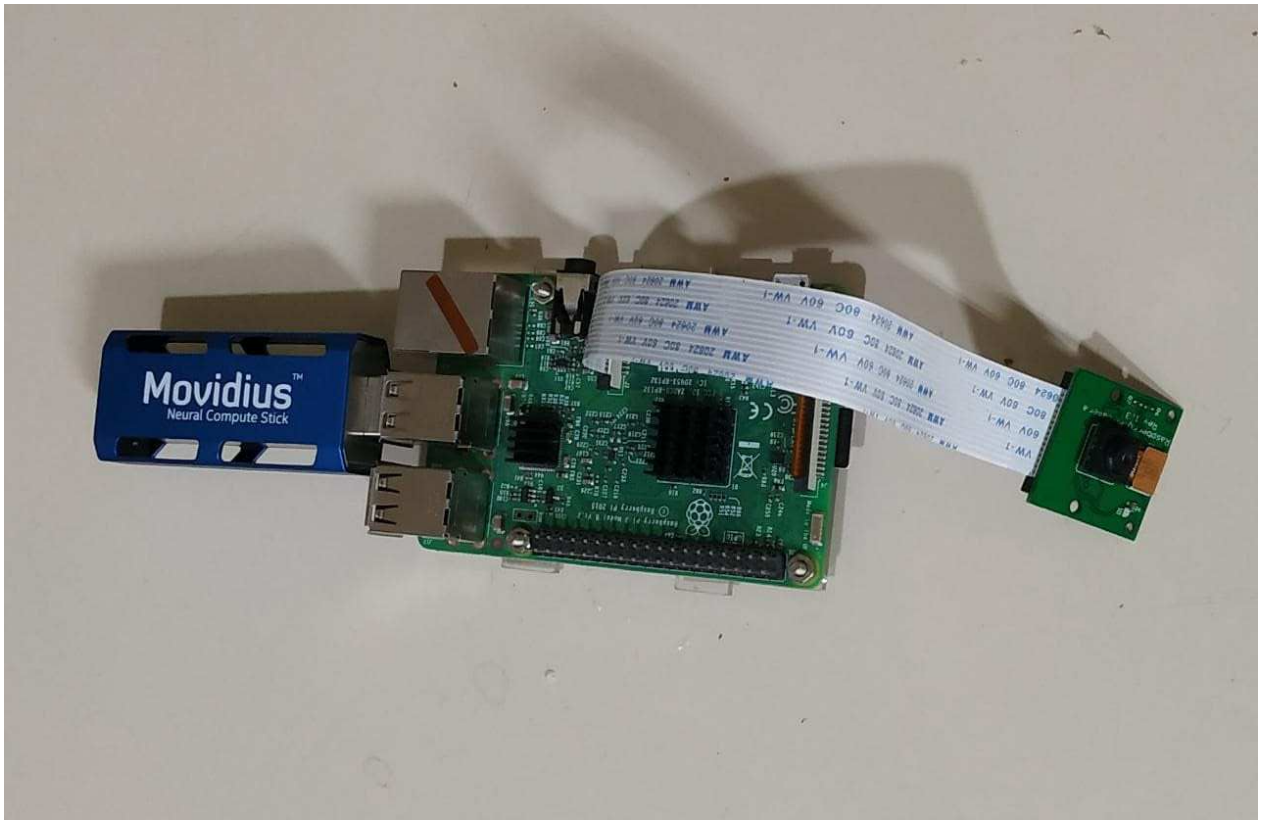

Figura 3.1: Foto do Sistema Físico da Arquitetura do Sistema

Além dos componentes expostos, ainda contamos com um NodeMCU [8] e alguns periféricos que permitiram a utilização do Raspberry PI 3B [6], como monitor, mouse, teclado e um cartão de memória 32Gb onde foi gravado o sistema operacional carregado no dispositivo.

Para uma melhor ilustração do ecossistema utilizado como um todo, temos a Figura 3.2, a qual contém um diagrama de como a estrutura funciona. Analisando-o , é possível notar o Raspberry

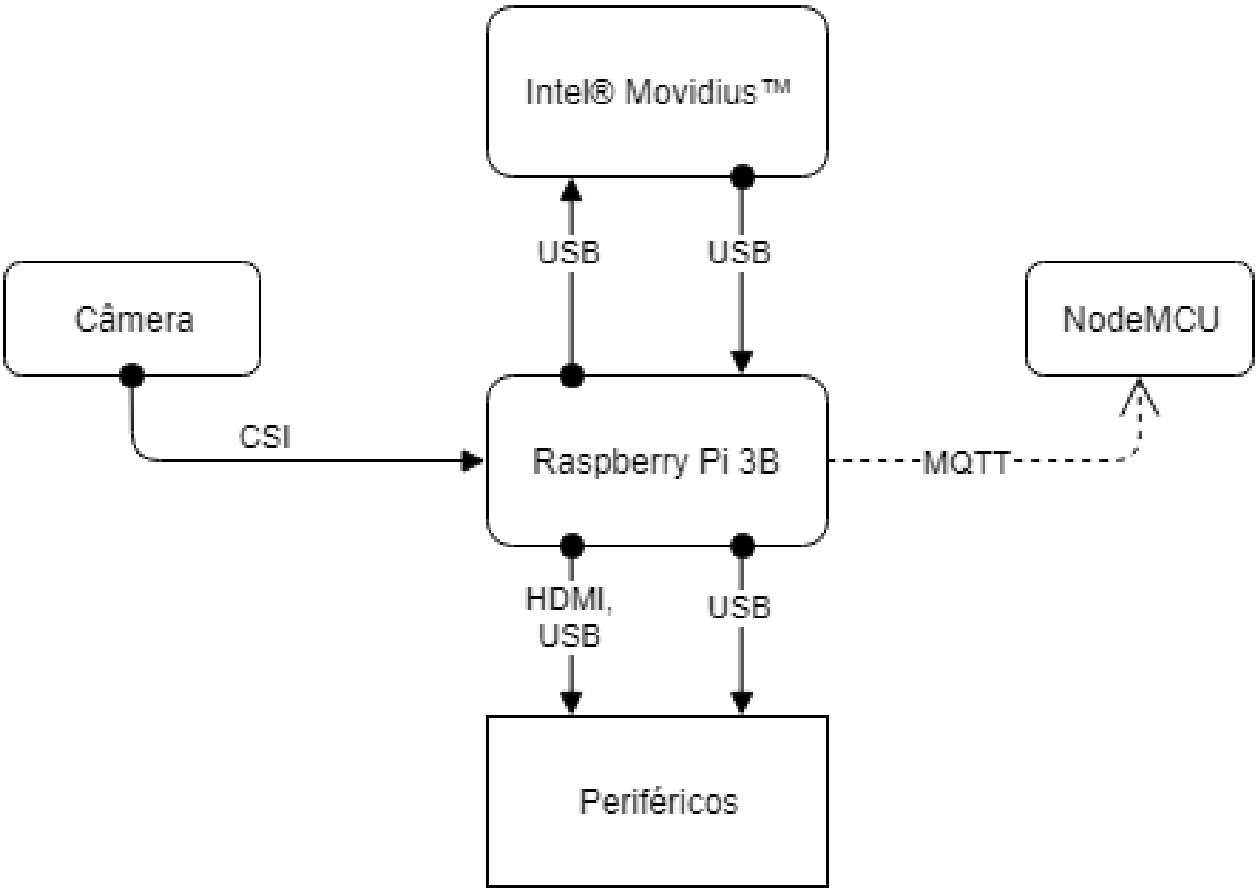

Figura 3.2: Diagrama Físico da Arquitetura do Sistema

PI 3B [6] como o centro físico de toda arquitetura montada e de fato tendo um papel essencial na captação e interpretação dos dados obtidos pela câmera, assim como, na gerência e processamento das atividades que o sistema esta proposto a executar.

Nesse contexto, é necessário descrever como é o processo da aplicação da arquitetura, mostrada no diagrama da Figura 3.2, no viés da problemática identificada em capítulos anteriores e então resultando nesta solução proposta.

A câmera é a principal responsável pela aquisição de dados para processamento, a partir do momento em que o sistema começa a executar, ela começa a capturar quadros de imagens do ambiente monitorado. Ao obter os quadros, estes são enviados ao Raspberry PI 3B [6] através de um cabo de fita, ou *flat*, utilizando o protocolo de comunicação CSI, o qual já tem um *slot* dedicado a este protocolo no microprocessador, facilitando seu uso e deixando suas portas USB disponíveis para outras aplicações.

Com os quadros de imagem, o microprocessador interage com o Intel® Movidius™ [5] e recebe os dados processados, onde, a partir destes dados, se o Raspberry PI 3B [6] identificar um cenário predeterminado, um gatilho é gerado e então é enviado um *"Publish"* no tópico referente ao cenário, da aplicação para o *broker*, que está rodando no próprio dispositivo, o qual já identifica o NodeMCU [8] que está inscrito no tópico do cenário identificado.

Ao receber o gatilho pelo tópico que está inscrito, o NodeMCU [8] atua no cenário.

3.5 Princípios de Funcionamento

Para que o algorítimo de reconhecimento de imagens fosse implementado no Raspberry PI 3B [6], tendo este também o auxílio dos periféricos e dispositivos auxiliares, a fim de solucionar a problemática, foi necessário idealizar possíveis formas de construção do código para melhor utilização do algorítimo e tal idealização trouxe novos problemas a nível de implementação como também surgiram as soluções de viés criacional para com os impasses afrontados.

Nos tópicos que seguem são apresentados os problemas encontrados, seguidos de sua respectiva solução, proposta pelo autor do projeto.

3.5.1 Configuração do Ambiente Virtual de Trabalho

Tomando por base a necessidade do uso de um microprocessador, viu-se que seria preciso a instalação de um sistema operacional para o Raspberry PI 3B [6], assim, o sistema escolhido foi o *Raspbian* [6]. Para sua instalação utilizou-se uma imagem chamada *Noobs* a qual serviria de facilitadora da instalação.

Com o *Raspbian* instalado no cartão de memória do Raspberry PI 3B, partiu-se para a configuração do ambiente de trabalho e instalação do módulos do Intel® Movidius™, para isto foi seguido o tutorial de inicialização fornecido em sua página da *Web*[8], o qual é dividido em três passos:

- **Passo 1**: Ter um Raspberry PI 3B, um Intel® Movidius™ e fazer o *download* do *Software Development Kit - SDK*;
- **Passo 2**: Foi criado um diretório de trabalho, feito download do SDK e instalado.

```
mkdir -p ~/workspace cd ~/workspace
git clone https://github.com/movidius/ncsdk.git
cd ncsdk
make install
```

- **Passo 3**: O dispositivo foi conectado e executado um exemplo para confirmar a instalação.

```
cd ~/workspace/ncsdk
make examples
```

Logo em seguida foi necessário a conexão da Pi Câmera, para esta conexão o Raspberry Pi 3B possui uma porta CSI que serve de fácil encaixe para a câmera, mostrada na Figura 3.3 [9].

Após a conexão física da câmera, foi necessário habilitar sua utilização no sistema operacional do Raspberry Pi 3B, o *Raspbian*, para isto foi seguido os passos mostrados na Figura 3.4.

Foi aberto o menu principal, e navegado para a aba de "Preferências"e dentro dela selecionado a opção "Raspberry Pi Configuration", após esta seleção é aberta uma janela referente a opção, nela escolheu-se a opção "Interfaces"e nesta foi marcado "*Enabled*"na alternativa da Câmera e apertado o botão "OK".

[8]Tutorial disponível em: https://software.intel.com/en-us/movidius-ncs-get-started

[9]Figura disponível em: https://www.filipeflop.com/produto/camera-raspberry-pi-v2-8mp/

Figura 3.3: Pi Câmera

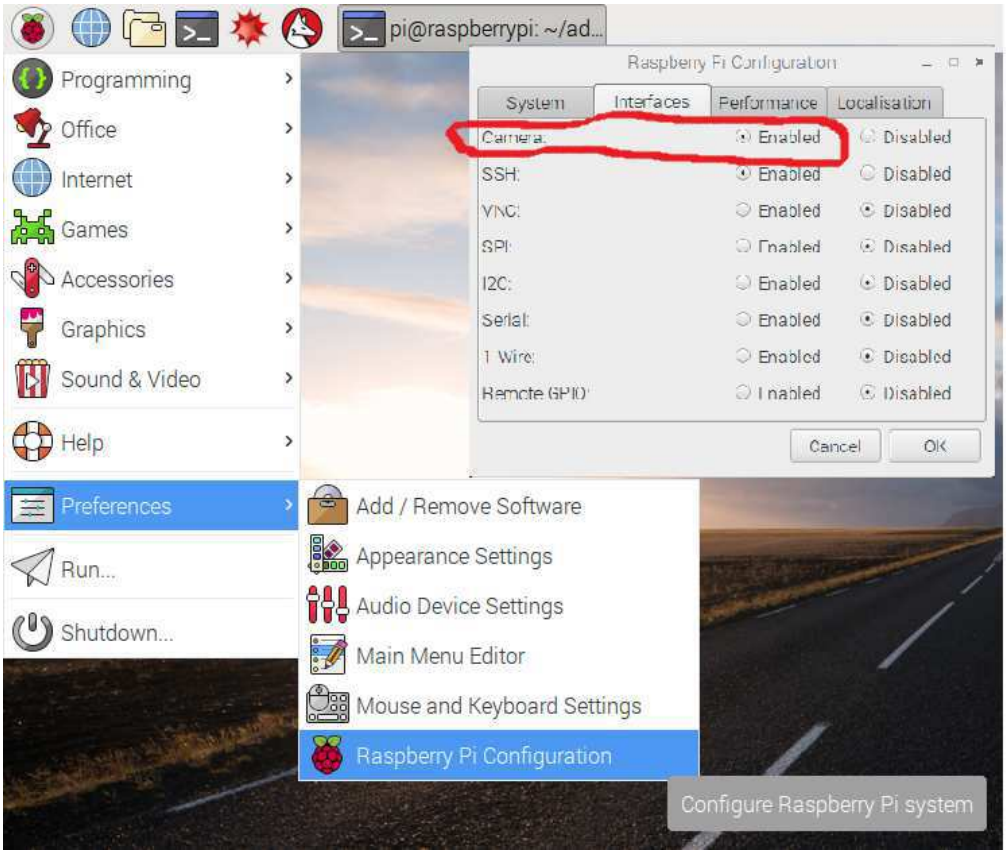

Figura 3.4: Configuração da Pi câmera

Por último, em um computador com o sistema operacional Windows 10, foi feito o download do ambiente de desenvolvimento do Arduíno com auxílio de Tutorial [10]. Neste ambiente de desenvolvimento foi necessário baixar os pacotes referentes ao NodeMCU ou ESP8266, que são instalados

[10]Tutorial disponível em: https://www.filipeflop.com/blog/programar-nodemcu-com-ide-arduino/

automaticamente, já sendo disponibilizado instantaneamente para uso.

3.5.2 Montagem do Ambiente Físico de Trabalho

O Raspberry PI 3B foi instalado em uma plataforma de acrílico para evitar o contato da parte inferior da placa com qualquer superfície, encaixado o Intel® Movidius™ na porta USB superior direita, que por ter um *design* espaçoso, bloqueou outras duas portas. Então foi plugado um extensor USB para que houvesse a possibilidade de conectar mouse e teclado. Por fim, utilizou-se um monitor conectado a um conversor HDMI-VGA, para que a conexão com a porta HDMI do microprocessador.

3.5.3 Código em Python

Para a codificação utilizou-se uma ferramenta de edição de texto, o *Geany*, ao qual é disposto na Figura 3.5 como a ferramenta é apresentada. Esta pôde ser encontrada no Sistema Operacional instalado no Raspberry Pi 3B.

A codificação foi feita na linguagem de programação Python.

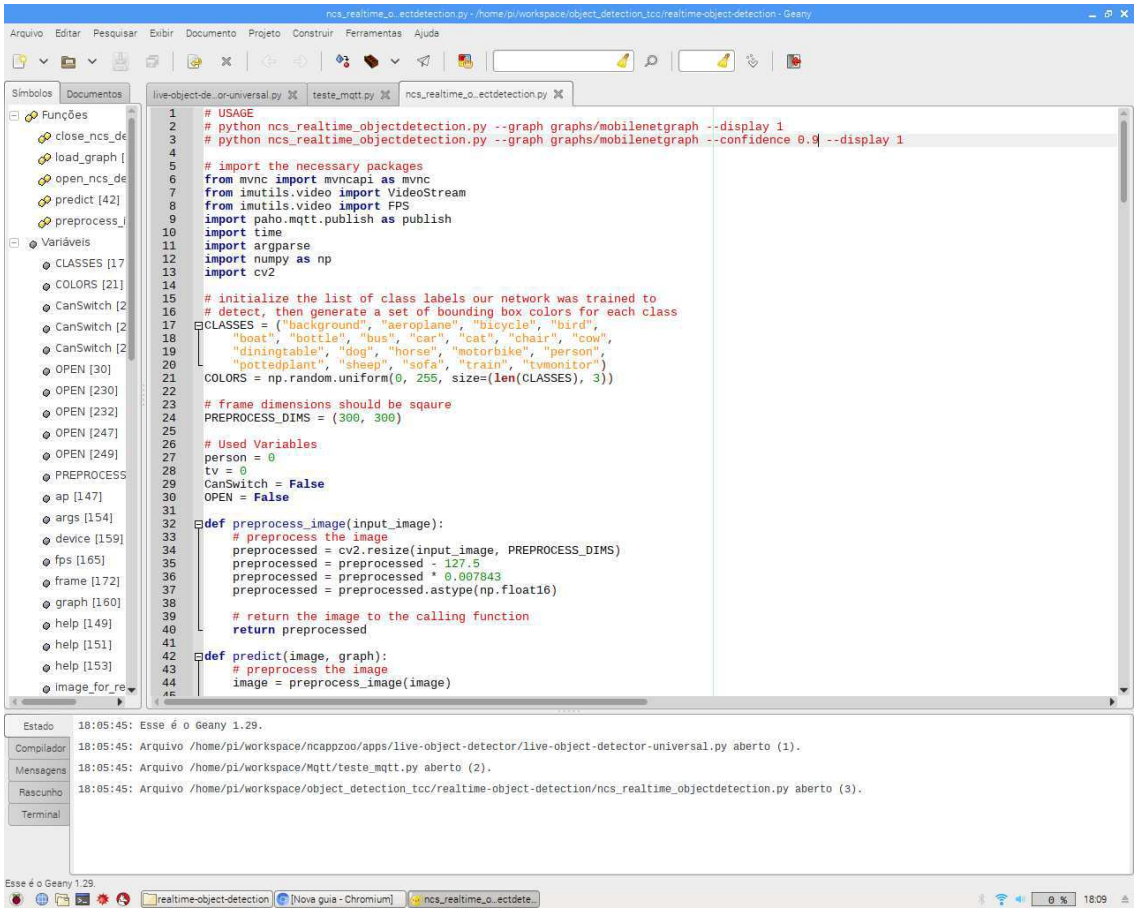

Figura 3.5: Editor de Textos Geany

Para criação do ambiente foi criado um ambiente de desenvolvimento com controle de versão em um repositório no BitBucket.

No início do código foram importadas as bibliotecas e pacotes necessários para o funcionamento do código, entre elas as bibliotecas da intel e utilizado os pacotes do paho que é um serviço para publicação e inscrição em tópicos Mqtt.

Neste código utilizou-se de uma função, que tem por principal objetivo, receber uma imagem e a redimensionamos para o tamanho definido anteriormente e fazer alguns ajustes para sua exibição.

A utilização desta função de redimensionamento é dada em uma das principais funções desta aplicação, descrita no Código 3.1, a função ”Predict”. Na linha 1 deste trecho é recebida uma imagem e um arquivo *”graph”*.

Este arquivo *”graph”* é o resultado de um treinamento feito para determinado banco de dados. O arquivo utilizado neste trabalho está disponível nos exemplos fornecidos pelo repositório da intel [11].

A imagem redimensionada é passada para uma variável na linha 2, variável esta que é carregada no arquivo *”graph”* na linha 6 e seu resultado guardado na variável interna chamada ”output”na linha 8.

```
1  def predict(image, graph):
2      # preprocess the image
3      image = preprocess_image(image)
4
5      # send the image to the NCS and run a forward pass to grab the
           network predictions
6      graph.LoadTensor(image, 'My Prediction')
7
8      output, userobj = graph.GetResult()
9
10     # grab the number of valid object predictions from the output,
11     # then initialize the list of predictions
12     num_valid_boxes = output[0]
13     predictions = []
```

Código 3.1

No código 3.2, três variáveis de grande importância são preenchidas nas linhas 4,5 e 6. Estas variáveis guardam a informação da classe prevista, da confiabilidade daquela predição e as coordenadas no qual se encontra o objeto detectado.

```
1
2          # grab the prediction class label, confidence (i.e.,
               probability),
3          # and bounding box (x, y)-coordinates
4          pred_class = int(output[base_index + 1])
5          pred_conf = output[base_index + 2]
6          pred_boxpts = ((x1, y1), (x2, y2))
7          # create prediciton tuple and append the prediction to the
```

[11] Exemplos disponíveis em: https://github.com/movidius/ncappzoo

```
8           # predictions list
9           prediction = (pred_class, pred_conf, pred_boxpts)
10          predictions.append(prediction)
11
12      # return the list of predictions to the calling function
13      return predictions
```

Código 3.2

Então estas três variáveis, na linha 9, são postas como uma predição formada na variável ”prediction”que por sua vez é adicionada a lista principal de predições, na linha 10.

No Código 3.3 é mostrado um trecho de código que esta em um ciclo de repetições e que ocorrem para todas as predições detectadas. Na linha 4, são recuperadas as características individuais de cada predição.

```
1           # loop over our predictions
2           for (i, pred) in enumerate(predictions):
3               # extract prediction data for readability
4               (pred_class, pred_conf, pred_boxpts) = pred
```

Código 3.3

No Código 3.4 é analisado se a predição tem um grau de confiabilidade maior que 90% de certeza, se não a predição é descartada. Dentro deste trecho encontra-se os códigos 3.5 e 3.6, os quais tem os mesmos níveis hierárquicos.

```
1               # filter out weak detections by ensuring the `confidence`
2               # is greater than the minimum confidence
3               if pred_conf > args["confidence"]:
4                   # print prediction to terminal
5                   print("[INFO] Prediction #{}: class={}, confidence={},
                        "
6                    "boxpoints={}".format(i, CLASSES[pred_class],
                        pred_conf,
7                    pred_boxpts))
8
9                   print( "I found these objects in "
10                  + " ( %.2f ms ):" % ( np.sum( inference_time ) ) )
11
12                  # check if we should show the prediction data
13                  # on the frame
```

Código 3.4

Notamos no Código 3.5 é analisado, na linha 1, se a imagem do video foi ativada. Se a condição for verdadeira, é extraído o nome referente a classe identificada na previsão na linha 4.

Já da linha 8 a linha 12, são extraídos as coordenadas das caixas que contém o objeto identificado pela previsão. Só na linha 15 é construído o quadro e na linha 17 é posto o nome identificado na linha 4.

```
1                   if args["display"] > 0:
2                       # build a label consisting of the predicted class
                           and
3                       # associated probability
4                       label = "{}: {:.2f}%".format(CLASSES[pred_class],
5                           pred_conf * 100)
6
7                       # extract information from the prediction boxpoints
8                       (ptA, ptB) = (pred_boxpts[0], pred_boxpts[1])
9                       ptA = (ptA[0] , ptA[1]) #1st coordenate
10                      ptB = (ptB[0] , ptB[1]) #2nd coordenate
11                      (startX, startY) = (ptA[0], ptA[1])
12                      y = startY - 15 if startY - 15 > 15 else startY +
                           15
13
14                      # display the rectangle and label text
15                      cv2.rectangle(image_for_result, ptA, ptB,
16                          COLORS[pred_class], 2)
17                      cv2.putText(image_for_result, label, (startX, y),
18                          cv2.FONT_HERSHEY_SIMPLEX, 1, COLORS[pred_class
                               ], 3)
```

Código 3.5

No Código 3.6 está o trecho responsável pela criação do cenário, os trechos definidos da linha 1 a linha 16 é semelhante ao trecho definido entre as linhas 18 e 33.

Na linha 1 é checado qual status da classe prevista. Se a classe identificada for uma pessoa, então a condição é satisfeita e então são extraídos, nas linhas de 2 a 5, o ponto médio, os limites direito e esquerdo e o valor da metade de um lado. Na linha 6 é guardado um valor de tempo.

Na linha 8 é checado se no ciclo anterior a classe sofá já havia sido identificada.

Se sim, é checado se a diferença do tempo de previsão da classe pessoa para a previsão da classe sofá é menor que 0.5 segundos, se sim, é checado se a diferença do meio da previsão da classe pessoa e o meio da classe sofa é menor que a soma da metade do lado da previsão da pessoa e da previsão do sofa.

Na Figura 3.6 esta exposto de forma gráfica a lógica utilizada, assim se:

$$|DistanciaP1P2| < HalfP1 + HalfP2 \tag{2}$$

Então a flag OPEN é modificada para verdadeiro na linha 13. Se não esta é mantida em falso na linha 15 e na linha 16 a indicação que a classe sofa foi identificada é zerada.

Como explicado o trecho entre as linhas 18 e 33 é semelhante ao processo descrito.

Com a flag OPEN sendo verdadeira, na linha 35 é permitido satisfazer a condição, então na linha 37 é publicado no tópico tvStatus o valor ”1”para o endereço do broker, ao qual tem a intenção de ligar o aparelho televisor.

Mas se a flag OPEN for ou se manter em falso, na linha 40 é satisfeita a condição e então é publicado no tópico tvStatus o valor ”0”para o endereço do broker, ao qual tem a intenção de desligar o aparelho televisor.

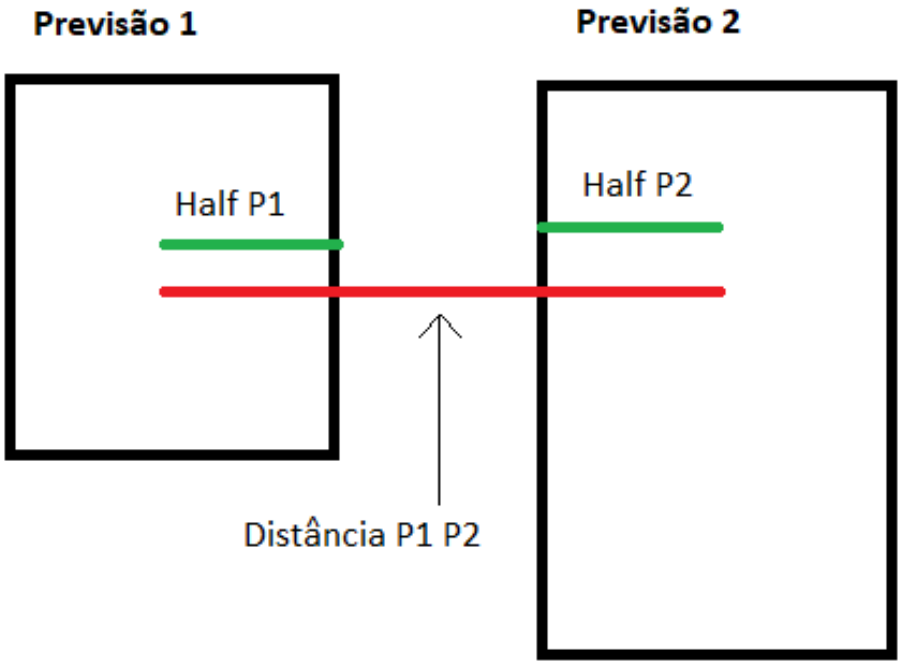

Figura 3.6: Algorítimo utilizado para detecção de proximidade das predições

```
                if CLASSES[pred_class] == "person":
                    personMiddlePoint = (((ptA[0]+ptB[0])/2),((ptA[1]+
                       ptB[1])/2))
                    personXrightlimit = ptB[0]
                    personXleftlimit = ptA[0]
                    personXhalfSide = abs(personMiddlePoint[0] -
                       personXrightlimit)
                    tperson = time.perf_counter()
                    person = 1
                    if sofa == 1:
                        if abs(tsofa-tperson) < 0.5:
                            if abs(personMiddlePoint[0] -
                               sofaMiddlePoint[0]) < (personXhalfSide +
                                sofaXhalfSide):
                                print(abs(personMiddlePoint[0] -
                                   sofaMiddlePoint[0]))
                                print((personXhalfSide + sofaXhalfSide)
                                   )
                                OPEN = True
                        else:
                            OPEN = False
                            sofa = 0
```

```
17
18                  if CLASSES[pred_class] == "sofa":
19                      sofaMiddlePoint = (((ptA[0]+ptB[0])/2),((ptA[1]+ptB
                            [1])/2))
20                      sofaXrightlimit = ptB[0]
21                      sofaXleftlimit = ptA[0]
22                      sofaXhalfSide = abs(sofaMiddlePoint[0] -
                            sofaXrightlimit)
23                      tsofa = time.perf_counter()
24                      sofa = 1
25                      if person == 1:
26                          if abs(tsofa-tperson) < 0.5:
27                              if abs(personMiddlePoint[0] -
                                    sofaMiddlePoint[0]) < (personXhalfSide +
                                     sofaXhalfSide):
28                                  print(abs(personMiddlePoint[0] -
                                        sofaMiddlePoint[0]))
29                                  print((personXhalfSide + sofaXhalfSide)
                                        )
30                                  OPEN = True
31                          else:
32                              OPEN = False
33                              person = 0
34
35                  if OPEN == True and CanSwitch == False:
36                      print("Turnig TV on")
37                      publish.single("tvStatus", "1", hostname="
                            192.168.0.21")
38                      CanSwitch = True
39
40                  if OPEN == False and CanSwitch == True:
41                      print("Turnig TV off")
42                      publish.single("tvStatus", "0", hostname="
                            192.168.0.21")
43                      CanSwitch = False
```

Código 3.6

No final deste código é onde se o usuário decidir parar o *software*, o mesmo pode fazê-lo precionando a tecla ”q”ou as teclas ”Ctrl + C”.

4 Validação

Para validar todo o processo desenvolvido, foi criada uma rotina de validação para o sistema, o que permitiu investigar e apurar o completo funcionamento do ecossistema montado.

4.1 Ferramentas utilizadas

Para fazer o processo de validação, foram necessárias as seguintes ferramentas:

- O sistema funcionando;
- Monitor e teclado para saber o momento certo da captura;
- Um cabo de rede para conectar no Raspberry, pois o mesmo necessita de um IP fixo e conhecido;
- Um ambiente com boa iluminação e um sofá;

4.2 Processo de Validação

Com *software* criado no Raspberry Pi 3B pronto, posicionou-se o dispositivo em uma sala de estar. Com o microprocessador funcionando e estando conectado a um monitor e teclado e mouse sem fio, assim como a um cabo de rede, posicionou-se a câmera para que a mesma possa captar o sofá e algum outro objeto presente no cenário.

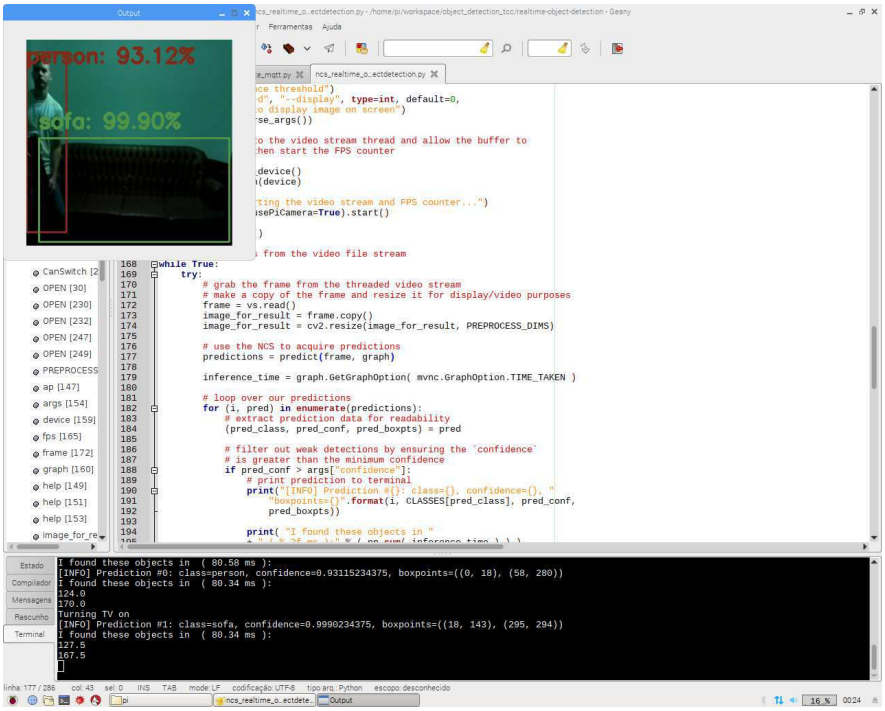

Figura 4.1: Momento ao qual é identificado uma pessoa próxima ao sofá

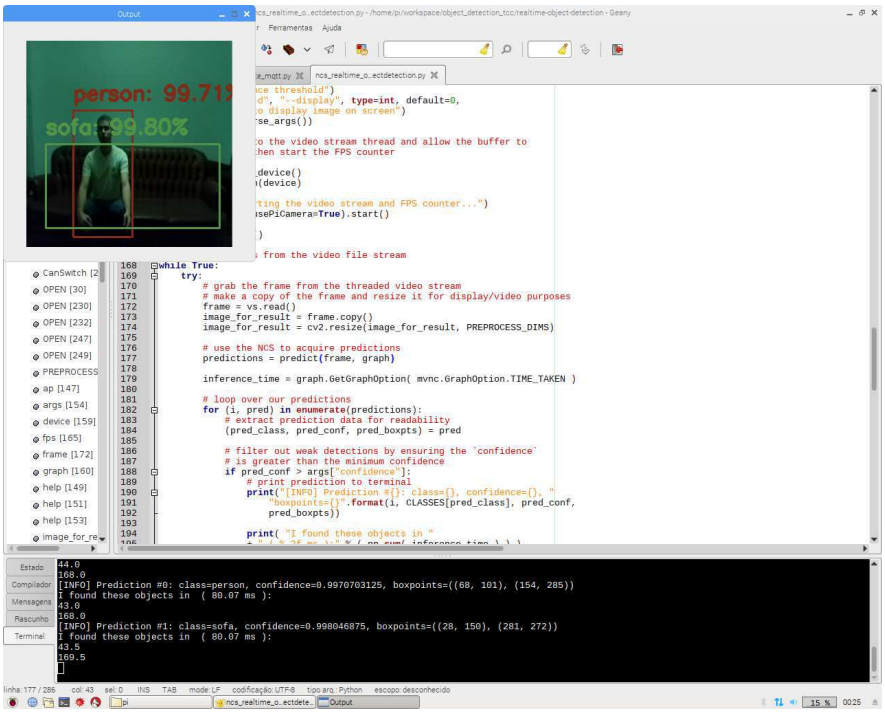

Figura 4.2: Enquanto a pessoa está no ambiente

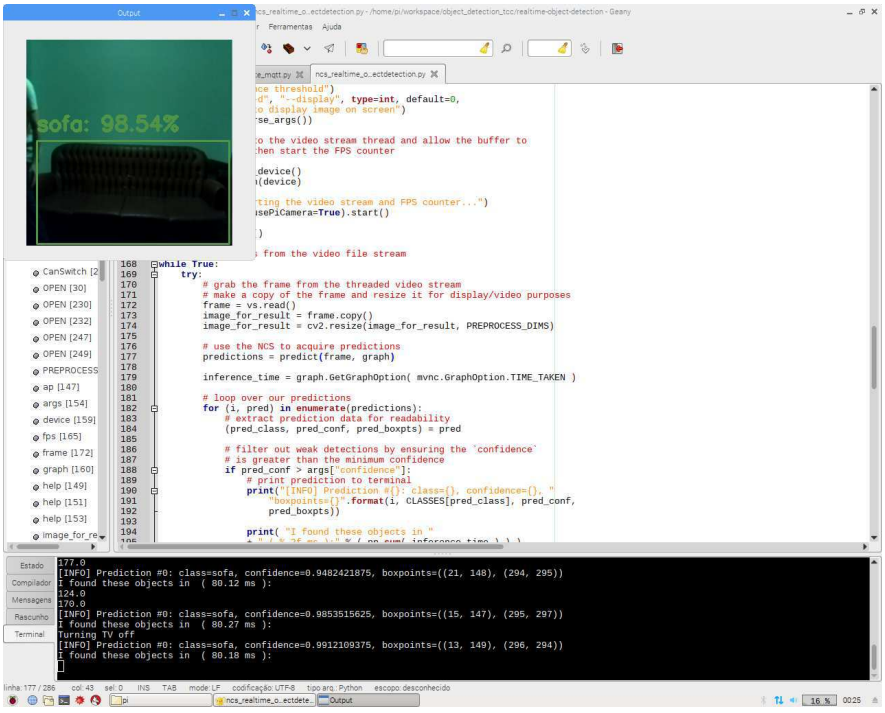

Figura 4.3: Momento ao qual é Identificado que uma Pessoa se retira do ambiente

Estando o sofá posicionado em frente a câmera, utilizou-se o teclado sem fio para tirar fotos da tela do Raspberry Pi 3B, enquanto o *software* estava sendo executado.

As Figuras 4.1, 4.3, 4.2 mostram imagens que foram capturadas durante o processo de validação. Na Figura 4.1 é o momento em que uma pessoa chega no ambiente e se aproxima do sofá, neste caso, ao ser identificado esta situação um sinal para o microcontralador é enviado via internet, utilizando o protocolo Mqtt. Esta ação pode ser identificada pela mensagem *"Turnig TV on"*.

Na Figura 4.2 temos um cenário a qual a pessoa permanece no ambiente próximo ao sofá, a situação pode ser identificada pela alternância de identificação de objetos, neste caso uma pessoa e um sofá.

Na Figura 4.3 é capturado o momento em que uma pessoa se retira do ambiente e se afasta do sofá, neste momento um sinal é enviado para o microcontralador, via internet, utilizando também o protocolo Mqtt, esta ação pode ser identificada pela mensagem *"Turnig TV off"*.

5 Conclusão

A Inteligência Artificial é um conceito, uma ideia ou uma filosofia que vem sendo sonhada há bastante tempo, tanto por satisfazer as necessidades dos seres humanos como por tornar certos diagnósticos ou execuções mais precisas. A expansão deste conceito vem tendo um crescimento notório nos últimos anos devido o surgimento de tecnologias robustas o suficiente para tornar possível a execução de atividades, que exigem uma grande capacidade de processamento, em tempo real.

A utilização deste conceito é muito mais abrangente que a aplicação feita neste trabalho, mas a partir desta aplicação é possível notar o poder das tecnologias que podem surgir ou serem aprimoradas a partir de conceitos embutidos dentro da inteligência artificial, como a visão computacional.

Com este trabalho foi possível criar uma solução para a problemática apresentada e que este se propôs a executar, além da validação do funcionamento correto. O cumprimento da proposta consiste em uma solução que aplique Inteligência Artificial para a identificação de presença de uma pessoa em um ambiente, para que com isso possa haver uma atuação no cenário montado. Para chegar neste ponto foram utilizados os conceitos de visão computacional.

Durante a execução deste projeto, algumas dificuldades foram observadas. O uso do Intel® Movidius™ se tornou um desafio, por ser uma tecnologia nova e haver pouco material de referência para seu uso, mas que pôde ser contornada com pesquisas e testes de execuções.

Um grande desafio enfrentado foi a criação de cenário, pois a identificação esta sendo feita uma a uma, logo, não se tem uma previsão na mesma exata janela de tempo, o que dificultou a identificação de dois objetos na mesma janela. Este obstaculo também foi contornado, mas com algumas restrições. Por exemplo, se algum objeto for identificado durante a execução do código, a rotina nunca permite que o televisor seja ligado, ou se o ambiente estiver mal iluminado os objetos não são identificados corretamente ou oscilam bastante. Neste caso de oscilação na identificação dos objetos deve-se parte ao arquivo resultante de treinamento utilizado e parte a qualidade da câmera utilizada.

Com as dificuldades enfrentadas neste projeto, foi possível identificar quais caminhos seriam os corretos a se seguir, sendo também os erros dessa forma um degrau para o conhecimento.

5.1 Trabalhos Futuros

A expectativa de desenvolvimento de novas tecnologias, como dispositivos que tem cada vez mais poder de processamento, e/ou aprimoramento da atuais, faz com que uma espessa gama de aplicações possam surgir ou serem aprimoradas.

A perspectiva é que em um futuro próximo, sejam criados verdadeiros *ecossistemas tecnológicos*, capazes de realizarem tarefas de forma completamente autônoma, em sua totalidade. Para que isto venha acontecer, é necessário a utilização de algorítimos robustos o bastante para suportarem o processamento em tempo real, aliado com dispositivos igualmente capazes deste processamento. Neste futuro, a existência de um ambiente totalmente inteligente é rodeado e embasado principalmente no conceito de inteligência artificial.

Com estes avanços e o uso do conceito de visão computacional não se resumirá a ambientes residenciais, mas também será expandido para ambientes corporativos e quem sabe utilizados para monitoramento de toda uma cidade.

Para tal propósito é possível melhorar o algorítimo de identificação proposto neste trabalho, assim como, expandir para a identificação de outros objetos e outros cenários mais complexos com três ou mais objetos. Além disso, com os resultados obtidos neste trabalho, é possível idealizar sistemas mais robustos e eficientes, mantendo a utilização das tecnologias escolhidas, podendo ser aplicado nos mais diversos campos da ciência, e para trabalhos futuros, assim, evidenciando a importância desta pesquisa e da realização deste Trabalho de Conclusão de Curso.

6 Referências

[1] Aprendizado de Máquina - *Website com descrição do Aprendizado de máquina*. Disponível em: https://news.sap.com/brazil/2017/10/o-que-e-machine-learning-ou-aprendizagem-de-maquina/
Acesso: 11 de Dezembro 2018

[2] Visão Computacional - *O QUE É VISÃO COMPUTACIONAL?*. Disponível em: http://datascienceacademy.com.br/blog/o-que-e-visao-computacional/
Acesso: 11 de Dezembro 2018

[3] Inteligência Artificial - *Website com imagem e descrição de Inteligência Artificial*. Disponível em: https://medium.com/data-science-brigade/a-diferen\%C3\%A7a-entre-intelig\%C3\%AAncia-artificial-machine-learning-e-deep-learning-930b5cc2aa42
Acesso: 11 de Dezembro 2018

[4] MQTT - *Message Queuing Telemetry Transport*. Disponível em: http://mqtt.org/
Acesso: 11 de Dezembro 2018

[5] Intel® Movidius™ - *Acessório voltado para processamento das imagens*. Disponível em: https://www.movidius.com/
Acesso: 11 de Dezembro 2018

[6] RaspBerry PI 3B - *Computador de placa única com conectividade LAN sem fio e Bluetooth*. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
Acesso: 11 de Dezembro 2018

[7] RNA - *Redes Neurais Artificiais*. Disponível em: http://conteudo.icmc.usp.br/pessoas/andre/research/neural/
Acesso: 11 de Dezembro 2018

[8] NodeMCU - *Esp 8266*. Disponível em: http://www.nodemcu.com/index_en.html
Acesso: 11 de Dezembro 2018

[9] Aprendizagem de Máquina - *Aprendizado Supervisionado e Não-Supervisionado*. Disponível em: https://medium.com/opensanca/aprendizagem-de-maquina-supervisionada-ou-n\%C3\%A3o-supervisionada-7d01f78cd80a
Acesso: 14 de Dezembro 2018

[10] Deep Learning - *Afinal, o que é Deep Learning?*. Disponível em: https://gaea.com.br/afinal-o-que-e-deep-learning/
Acesso: 14 de Dezembro 2018

[11] Deep Learning - *Deep Learning made easy with Deep Cognition*. Disponível em: https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351
Acesso: 14 de Dezembro 2018

[12] Deep Learning - *Why is Deep Learning taking off ?*. Disponível em: https://machinelearning-blog.com/2017/11/03/erster-blogbeitrag/
Acesso: 14 de Dezembro 2018

[13] Córtex Cerebral . Disponível em: https://www.portalsaofrancisco.com.br/corpo-humano/cortex-cerebral
Acesso: 14 de Dezembro 2018