Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

LUCAS DOS SANTOS MELLO

DESESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA CONTROLADA REMOTAMENTE UTILIZANDO RASPBERRY PI COM DETECÇÃO DE OBJETOS

Assis/SP
2017

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

LUCAS DOS SANTOS MELLO

DESESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA CONTROLADA REMOTAMENTE UTILIZANDO RASPBERRY PI COM DETECÇÃO DE OBJETOS

Projeto de pesquisa apresentado ao curso de do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito à obtenção do Certificado de Conclusão.

Orientando(a): Lucas dos Santos Mello
Orientador(a): Prof.Douglas Sanches da Cunha

Assis/SP
2017

FICHA CATALOGRÁFICA

M527d MELLO, Lucas dos Santos

Desenvolvimento de uma plataforma robótica controlada remotamente utilizando Raspberry Pi com detecção de objetos / Lucas dos Santos Mello. -- Assis, 2017.

60p.

Trabalho de conclusão do curso (Ciência da Computação). – Fundação Educacionaldo Município de Assis-FEMA

Orientador: Prof.Douglas Sanches da Cunha

1.Robótica 2.Programação

CDD 006.31

DESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA CONTROLADA REMOTAMENTE UTILIZANDO RASPBERRY PI COM DETECÇÃO DE OBJETOS

LUCAS DOS SANTOS MELLO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: Prof.Douglas Sanches da Cunha
Inserir aqui o nome do orientador

Examinador: Me.Fábio Eder Cardoso
Inserir aqui o nome do examinador

Assis/SP
2017

AGRADECIMENTOS

Agradeço primeiramente a Deus por me guiar e dar forças a continuar, aos meus pais Jose e Elias que sempre me apoiarão, e apoiam em todo momento, e também as minhas irmãs Gabriela e Isabella, sem eles nada disso seria possível, além também aos meus amigos e colegas de faculdade que me ajudaram em todo o processo de graduação, e sem deixar de mencionar a todos os professores que tive no curso ao longo desses quatro anos em que compartilharam seus conhecimentos acadêmicos, profissionais, e pessoais conosco.

É perigoso sair porta afora', ele costumava dizer.
'Você pisa na Estrada, e se não controlar seus pés,
não há como saber até onde você pode ser levado...'"

Bilbo Begins

RESUMO

Cada vez mais torna-se possivel a realização de projetos robóticos através de periféricos como arduino, raspberry pi, sensores, módulos entre outros periféricos, no presente trabalho foi desenvolvido uma plataforma robótica capaz de se mover por comandos via web, calcular distância de obstáculos, e realizar filmagem com detecção de objetos e rostos com conceitos de visão computacional, e é apresentado que com pesquisa, paciência e vontade, é totalmente possivel construir os mais variados tipos de projetos eletrônicos com um baixo custo.

Palavras-chave: Robótica, Programação;

ABSTRACT

Currently, it is possible to build robotic projects using peripherals such as arduino, raspberry pi, sensors, modules and other peripherals, in the present work, It was developed a robotic platform capable of moving by web commands, and filming with detection of objects and faces with concepts of computer vision, and it is presented that with research, patience and will, it is totally possible to build the most varied types of electronic projects with a low cost.

Keywords: Robotic, Programmation;

LISTA DE ILUSTRAÇÕES

Figura 1 - Pregador em difentes posições 18
Figura 2 - Primeira Imagem Digital 19
Figura 3- Representação de imagem em uma matriz 20
Figura 4- Representação de imagem digitalizada em BMP. 21
Figura 5- Representação de um globo ocular 23
Figura 6- Representação Funções globo ocular 23
Figura 7- Representação de um olho focalzando em uma árvore 24
Figura 8- Primeira Versão do Raspberry Pi B 29
Figura 9- Raspberry Pi B frente e verso 30
Figura 10 - LEDs de Status do Raspberry Pi ligadas. 31
Figura 11 - Diferentes Versões do Raspberry Pi já lançadas 34
Figura 12- RaspBerry Pi 3 B 38
Figura 13- RaspBerry Pi Camera. 38
Figura 14- Ponte H L298N 40
Figura 15- Sensor de Distância Ultrasônico HCR04 41
Figura 16- Chassi e peças menores 42
Figura 17- Desktop Raspbian 42
Figura 18- Estrutua Opencv 44
Figura 19 - Esboço Chassi Montado 45
Figura 20- Chassi Montado sem os componentes eletrônicos 45
Figura 21- Esquema do Projeto 46
Figura 22- Plataforma robótica montada 46
Figura 23 - Diagrama do gatilho de pulso do sensor 48
Figura 24 - Ilustração do pulso do sensor 49

Figura 25- Logo Raspberry 51

Figura 26 – Rosto detectado..... 52

Figura 27 - Imagem em diferentes fundos.....52

Figura 28 -Interface da aplicação WEB.....54

LISTA DE TABELAS

Tabela 1: Comparativo entre visão humana e visão arificial.....25

Tabela 2: Comparativo Raspberry Pi B e Raspberry Pi A.....29

Tabela 3: Especificações Raspberry Pi Camera.....39

SUMÁRIO

1. INTRODUÇÃO 13

1.1. OBJETIVOS 14

1.2. JUSTIFICATIVAS 14

1.3. MOTIVAÇÃO 14

1.4. ESTRUTURA DO TRABALHO 15

2. VISÃO COMPUTACIONAL 16

2.1. HISTÓRIA E ALGUNS CONCEITOS SOBRE A VISÃO COMPUTACIONAL 16

2.2. FORMAÇÃO DE UMA IMAGEM DIGITAL 19

2.3. O SISTEMA VISUAL HUMANO 22

2.4. PROCESSAMENTO DIGITAL DE IMAGEM 26

3. RASPBERRY PI 28

3.1. HISTÓRIA DO RASPBERRY PI 28

3.2. ANÁLISE DE COMPONENTES INTEGRADOS NO RASPBERRY PI 30

3.3. COMPONENTES EXTERNOS ESSENCIAIS PARA O RASPBERRY PI 33

3.4. DIFERENTES VERSÕES LANÇADAS DO RASPBERRY PI 34

3.5. SISTEMAS DISPONIVEIS PARA RASPBERRY 35

3.6. COMUNIDADE DO RASPBERRY PI 36

4. PROPOSTA DE TRABALHO 37

4.1. RECURSOS FISICOS UTILIZADOS 37

4.1.2. **Raspberry Pi 3 B** 37

4.1.3. **Raspberry Câmera Pi v1** 38

4.1.4. **Módulo Ponte H L298N** 39

4.1.5. **Sensor Ultrasônico HC-SR04** 40

4.1.6. **Chassi e Peças menores** 41
4.2. RECURSOS LÓGICOS UTILIZADOS 42
4.2.2. Raspbian 42
4.2.3. **Linguagem de Programação Python** 42
4.2.4. **Biblioteca Opencv** 43
4.3. CONSTRUÇÃO FISICA E DESENVOLVIMENTO DA PLATAFORMA ROBÓTICA 44
4.3.2. **Montagem** 44
4.4. DESENVOLVIMENTO 46
4.4.1. **Acesso Remoto ao Raspberry Pi** 47
4.4.2. **Programação para o controle dos motores** 47
4.4.3. **Programação para medir a distância com o sensor** 48
4.4.4. **Programação da câmera com detecção de objetos** 50
4.4.5. **Integração com a WEB** 54
5. CONCLUSÃO 55
6. REFERÊNCIAS 57

1. INTRODUÇÃO

O termo robô foi originalmente utilizado em 1921 pelo dramaturgo checo Karen Capek, na peça teatral "Os Robôs Universais de Russum (R.U.M)" como referência a um autômato que acaba rebelando-se contra o ser humano.Robô deriva da palavra "robota" de origem eslava, que significa "trabalho forçado".(ROMANO,DUTRA,2002).

O conceito moderno robô foi criadho por Joseph Engelberger, em 1962, junto com Devol, desenvolveu o primeiro protótipo de robô, chamado de Unimate, para ser utilizado em diversas aplicações industriais concretas.A primeira instalação registrada do robô Unimate aconteceu na Ford Motor Company para o descarregamento de uma máquina de fundição sobre pressão.Posteriormente, a grande maioria dos desenvolvedores em robótica basearam-se no desenvolvimento da tecnologia de computadores e microprocessadores em geral.Hoje e dia, a maioria dos robôs e quase todos na area industrial utilizam como controlador um computador pessoal ou algum tipo de controlador digital programável.(PAZOS;2002).

Atualmente o desenvolvimento robótico cresce cada vez mais, existem criações desde as mais simples como uma plataforma que só se move sem nenhuma funcionalidade extra, até robôs autonômos que possuem algoritmos de alta perfomance com aprendizado de máquina, onde aprendem com seus erros, e aprendem a identificar rostos, objetos, vozes, fotos, e são capazes de até mesmo de falar, além de outras diversas funcionalidades definidas pelos desenvolvedores.

Com o advento de circuitos integrados como o arduino e o Raspberry Pi, que possuem preços acessiveis, e possuem ferramentas para trabalhar opensource, foi gerado uma grande comunidade de desenvolvedores que buscam criar seus próprios pojetos, seja ele, educativo, cientifico, acadêmico, ou por simples hobby, isso afeta diretamente o crescimento da evolução da robótica, pois o que antes era somente desenvolvido por grandes empresas para ramos industriais, medicinais, ou agropecuários, se tornou barato e acessivel para o desenvolvimento em pequeno e médio porte, aumentando muito a geração de novas idéias e o crescimento da robótica.

1.1. OBJETIVOS

O presente trabalho tem como objetivo aprofundar conhecimentos na área da robótica e visão computacional, através do raspberry Pi, periféricos eletrônicos, e softwares open source, e a partir de pesquisas feitas será realizado o desenvolvimento de uma plataforma robótica capaz de utilizar câmera para realizar detecção de determinados objetos, e medir distância através do uso de um sensor de distância, a plataforma robótica será controlada remotamente, e capaz de realizar a detecção de determinados objetos através da câmera.

1.2. JUSTIFICATIVAS

Com a acessibilidade de desenvolvimento robótico cada vez maior, devido a ter uma grande variedade de softwares opensource, e hardwares com baixo custo para essa finalidade, e uma comunidade bastante ativa que buscam evoluir cada vez mais essa área, criar um protótipo utilizando essas ferramentas se torna a melhor maneira de aprende-las, por meio da prática de desenvolvimento.

1.3. MOTIVAÇÃO

A realização deste trabalho estabeleceu-se por uma enorme vontade, de realizar o desenvolvimento de uma plataforma robótica, e com o aumento nas pesquisas e desenvolvimento na área de aprendizado de máquina, e visão computacional, foi buscado uma maneira de tentar unir estes dois universos, a fim de unir duas áreas com um futuro muito promissor e que se complementam muito bem, pois o que podemos ter de grande evolução tecnologica no futuro, concerteza teram estes dois aspectos envolvidos, desta maneira trabalhar com temas tão atuais, e que evoluem tanto a cada dia se torna muito importante e divertido de aprender.

1.4. ESTRUTURA DO TRABALHO

O trabalho foi dividido em seis capitulos, sendo o primeiro a introdução já abordada anteriormente.O segundo capitulo busca abordar a parte teórica sobre a visão computacional, sua história, como funciona, além de também explanar sobre a visão humana e processamento de imagens.No terceiro capitulo é abordado elementos teóricos e técnicos sobre o raspberry pi como sua história, modelos lançados, ficha técnica, periféricos suportados, softwares suportados, etc.No quarto capitulo é apresentado a proposta do trabalho, apresentando os recursos fisicos e lógicos utilizados, e a montagem fisica da plataforma robótica, e o desenvovimento da parte lógica, além dos resultados obtidos.No quinto capitulo é feita as considerações finais por meio da conclusão do trabalho, onde é feita uma observação sobre os tópicos abordados no projeto, e como foi trabalhar com estes recursos.No sexto capitulo são feitas as referências que serviram de base para o desenvolvimento deste trabalho.

2. VISÃO COMPUTACIONAL

No presente capitulo é apresentado a parte teórica sobre a visão computacional tais como sua história, conceitos, além também de explanar sobre conceitos da visão humana, buscando fazer um comparativo com a visão artificial.Também é mostrado como é feito o processamento de imagens, mostrando cada etapa do processo, e os resultados esperados.

2.1. HISTÓRIA E ALGUNS CONCEITOS SOBRE A VISÃO COMPUTACIONAL

Visão computacional é a ciência de criar capacidade similar visual de humanos em computadores, e se possivel melhora-los.A definição mais técnica, no entanto, seria que a visão computacional é a ciência de ter máquinas para o processamento e análise de imagens digitais.(Demaagd et al,2012). Nós seres humanos quando vamos “ver” algo, pode parecer simples pois é algo natural no cotidiano, porém em uma abordagem mais profunda de como esse processo funciona, nosso corpo realiza várias análises e percepções do que esta em nosso campo de visão, análises tão complexas que exigem cerca de 2 terços do nosso cérebro, logo podemos perceber que tentar replicar algo desta magnitude em uma máquina não é algo tão simples.

Huang (1996), explica que foi com o desejo de tornar possivel que máquinas pudessem processar, analisar e identificar imagens que Larry Roberts em seu Ph.D, em 1963 no MIT discutiu com seu trabalho denominado “Machine Perception of Three-Dimensional Solids”, as possibilidades de extrair informações geométricas 3D, através da visão em perspectiva 2D de blocos, e com este trabalho foi estabelecido um alicerce para o campo da visão computacional. Um breve resumo dos 50 anos de visão computacional:

- 1960s: Processamento de imagem e reconhecimento de padrões aparecem na IA.
- 1970s: Horn, Koenderink, Longuet-Higgins contribuem para o processamento de imagens.
- 1980s: Matematica, a teoria da probabilidade e do controle é aplicada a visão computacional

- 1990s: A visão computacional é aumentada por computação gráfica e aprendizagem estastatistica.
- 2000s: Os avanços no reconhecimento visual e nas principais aplicações práticas têm impacto significativo na Visão computacinal.(Tadiou,2013).

Grande parte do que busca-se na visão computacional, é encontrar métodos que possam auxiliar problemas humanos relacionados a visão e percepção ao nosso redor, dessa maneira não subistituindo mais auxiliando a visão biológica naquilo que ela seja incapaz, e ou não totalmente eficaz de perceber, ou raciocinar, como encontrar doenças por fotos de exames através de determinados padrões, rastrear pessoas através de uma câmera de segurança em meio a um lugar populoso, entre outros exemplos em que as caracteristicas que se deseja buscar com a visão humana, são dificies de encontrar por problemas externos como longa distância, alto número de detalhes atrapalhando o campo de visão, ou caracteristicas muito pequenas e invisiveis ao olho nú.

Graças a todas pesquisas e projetos realizados no decorrer do tempo, para se aprimorar as teorias e métodos para extração de informações úteis, contidas em uma determinada imagem. A visão computacional avançou, e ainda avança muito, porém ainda tem muito o que melhorar, Szeliski (2010) destaca que atualmente temos técnicas confiáveis para calcular, com precisão um modelo 3D parcial de um ambiente de milhares de fotografias parcialmente sobrepostas, dado um grande conjunto de imagens para analisar, podemos também criar modelos de superficie 3D de objetos, podemos rastrear pessoas que se movem na rua através de câmeras inteligentes, e até mesmo encontrar e nomear pessoas em uma determinada foto, utilizando uma combinação de rostos, roupas e cabelos, com detecção e reconhecimento de imagem.

Porém apesar de todos avanços que possuimos, construir um computador com o mesmo nivel de detecção e reconhecimento de imagem de uma criança de 2 anos por exemplo, parece ainda estar distante.

Uma grande parte da filosofia de como a visão computacional funciona e como consegue ser trabalhada é resumida no livro de David Marr (1982).

Em particular, Marr (1982), introduziu sua noção dos três niveis de descrição de um sistema visual de processamento de informações, estes três niveis são:

- Teoria Computacional: qual é o objetivo da computação (tarefa) e quais são as restrições que são conhecidas ou podem ser trazidas para suportar um problema ?
- Representações e algoritmos: quais são as informações representados de entrada, saida e intermediárias, e quais algoritmos são usados para calcular o resultado desejado ?
- Implementação de hardware: Como as representações e algoritmos são mapeados em um hardware real, por exemplo, um sistema de visão biológica ? Como as restrições de hardware podem ser usadas para orientar a escolha da representação do algoritmo ? Por outro lado como as restrições de hardware podem ser usadas para orientar a escolha da representação e algoritmo ? (Szelisiki,2010);

Os computadores apesar de possuirem um alto nivel de processamento e execução de cálculos complexos, são fracos quando o assunto é ter um “raciocíonio”, como identificar determinado objeto em uma imagem, isso pois qualquer elemento externo pode prejudicar na análise, como a posição do objeto, sombras, brilho, iluminação, por exemplo na figura 1 onde é representada somente um tipo de objeto, porém em ângulos diferentes, que é facilmente identificado pelos olhos humanos como um pregador de roupas, se torna extremamente dificil para um computador identificar, ainda mais em um caso onde a imagem tiver pouca iluminação, ou tiver mais posições, como o pregador deitado, nas mãos de uma pessoa, ou estiver em fundos diferentes como em uma mesa, no varal, em uma prateleira etc, para o computador poder identificar um objeto nas diversas circunstâncias que ele pode estar representado é necessário fornecer uma larga quantidade de banco de imagens com o objeto, e levar em consideração todos estes problemas citados, ou seja é necessário ter imagens de todos tipos de situações possiveis com o objeto, para desta maneira realizar o processamento da imagem, e então tornar possivel a detecção do objeto requisitado, e mesmo depois de realizar todos estes procedimentos, a detecção pode não ficar tão precisa ao ponto de identificar o objeto em qualquer circunstância.Na figura a seguir (Figura 1) é apresentado um exemplo de um objeto em diferentes posições para exemplificar o exemplo transcrito anteriormente.

Figura 1 Pregador em difentes posições

Fonte: https://thumbs.dreamstime.com/b/o-peg-de-roupa.jpg

2.2. FORMAÇÃO DE UMA IMAGEM DIGITAL

Uma imagem pode ser definida como uma função bidimensional, f (x,y) em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x,y) é chamada de intensidade ou nivel de cinza da imagem nesse ponto, quando x, y e os valores de intensidades de f são quantidades finitas e discretas chamamos de imagem digital,(Gonzales et al, 1977).

Balan (2009) esclarece que imagem é uma representação de uma cena por meio da reflexão da luz na área enquadrada adquirida por meio de dispositivos sensiveis a luz, além de serem a representação da realidade que nos cerca, emolduradas pelo recorte natural do campo de visão, cujo sentido e percepção criam significados e conduz a reflexões ou ações.

A primeira imagem digital que se tem registro, foi realizada em 1975 por Russel Kirsh no NBS – National Bureau of Standards agora NIST – National Institute and Technology, é a imagem de um bebê (Figura 2) em uma foto 5x5 cm, o então NBS, desenvolveu o primeiro computador programável, o Standards Eastern Automatic Computer (SEAC), Kirsh e sua equipe criaram um escâner com tambor rotativo e com ele conseguiu escanear a foto tamanho 4x3 do bebê Waldem, que entrou para história como a primeira imagem adquirida e armazenada pela representação digital da imagem, a foto foi registrada com 176 pixels em preto e branco.(Balan,2009).

Figura 2 Primeira Imagem Digital

Fonte: http://www.mdig.com.br/imagens/curiosidade/primeira_imagem_digital.jpg

Para gerar uma imagem digital é necessário a imagem sofrer uma separação espacial (amostragem) e em aplitude (quantização), é feita uma amostragem normalmente uniforme de f(x, y) nas direções x e y, gerando uma matriz de M x N (Figura 3) pontos seguidos de uma quantização do valor de f(x, y) em niveis de cinza, a amostragem é a divisão do plano x, y em uma grade onde x e y serão números inteiros, os pontos da matriz são denominados pixels (PICTure Elements), cada pixel representa uma parte da cena real, desta forma a resolução espacial da imagem é proporcional aos valores M e N correspondentes na matriz, em geral a malha de amostragem, o formato dos pixels (x, y), é retangular, mas pode também ser triangular ou mais complexa, os valores de cada ponto da matriz, coluna x linha (x, y), que identifica um único pixel (M,N0), dever ser escolhidas de forma a respeitar a relação qualidade da imagem x espaço de armazenamento, em função da aplicação para a qual a imagem se destina, para uma imagem digital com 256 niveis de cinza o número de bytes ocupados para armazenar a imagem é o produto da linha vezes a coluna da matriz.(Balan,2009).

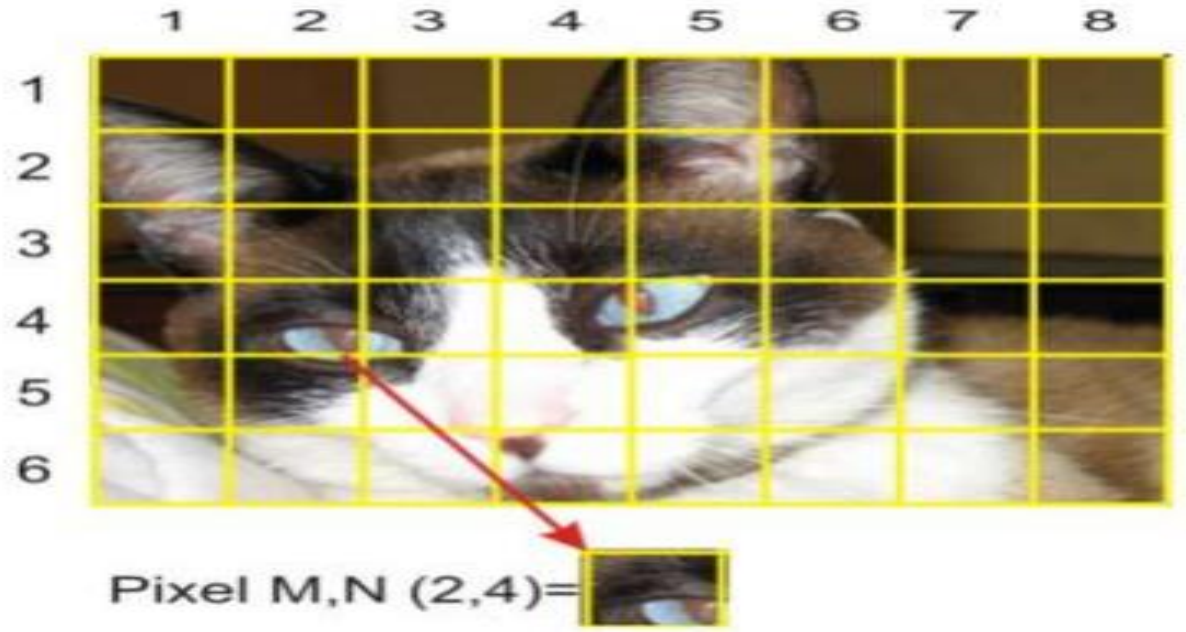

Figura 3 Representação de imagem em uma matriz

Fonte: http://www.willians.pro.br/textos/A%20Imagem%20Digital%20-%20Willians%20Cerozzi%20Balan.pdf

A imagem digital precisa reproduzir os mesmos elementos da imagem original. A representação digital se utiliza das técnicas de mapeamento por área. A imagem original é dividida no menor fragmento que a compõe, o pixel,(Balan,2009).

A figura 4 representa como é o processo de digitalização da imagem pelo formato BMP.

A estrutura do formato BMP é dividida em 4 partes, que levam as informações da imagem original para o formato digital:

- cabeçalho de arquivo: que contém a assinatura bitmap e informações sobre o tamanho e o lay-out, do arquivo, ou seja, a disposição dos dados dentro do arquivo;

- cabeçalho de mapa de bits: são as informações da imagem dentro do arquivo: dimensões, tipo de compressão caso haja, informações sobre as cores da imagem;

- paleta ou mapa de cores: é opcional utilizado em imagens que usa 16 ou 256 cores, correspondente a 4 e 8 bits/pixel;

- área de dados da imagem contida no arquivo: contém os dados dos pixels com as informações que permitem que a imagem seja exibida, ,(Balan,2009).

Figura ilustrando a explicação a seguir (Figura 4):

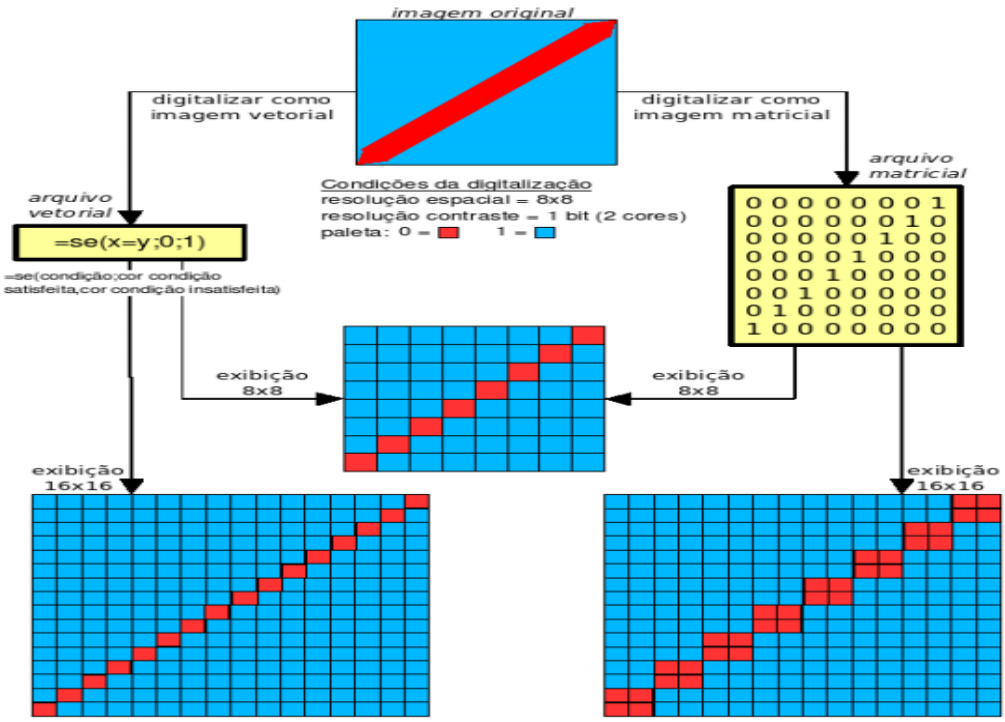

Figura 4 Representação de imagem digitalizada em BMP

Fonte: http://chasqueweb.ufrgs.br

2.3. O SISTEMA VISUAL HUMANO

Para realizar a detecção de objetos, e outras caracteristicas de um ambiente em uma imagem, ter um modelo que funcione para seguir e replicar, é essencial para se obter um resultado satisfatório, e em menor tempo, e o melhor exemplo para tomar como modelo é a visão humana, por este fator tornou-se necessário a criação de um subcapítulo explanando conceitos da visão humana, e como ela funciona.

Gonzales (1977), explica que o olho é praticamente uma esfera, com um diâmetro médio de aproximadamente 20 mm. Três membranas o revestem: a córnea e a cobertura externa da esclera; acoroide; e a retina.

A córnea é um tecido resistente e transparente que cobre a superfície anterior do olho.Como um prolongamento da córnea, temos a esclera,uma membrana opaca que reveste o restante do globo ocular. A coroide situa-se diretamente abaixo da esclera. Essa membrana contém uma rede de vasos sanguíneos que atua como a principal fonte de nutrição para o olho.O revestimento da coroide é substancialmente pig-mentado, ajudando a reduzir a quantidade de luz indese-jável que entra no olho e se espalha pelo globo ocular. Em seu extremo anterior, a coroide se divide em corpo ciliar e irís, a irís se contrai ou se expande para controlar a quantidade de luz que entra no olho, o cristalino é composto de camadas concêntricas de células fibrosas e é suspenso por fibras que se ligam ao corpo ciliar, e é o responsável por absorver a luz infravermelha e a ultravioleta, o membro mais interno do olho é a retina, quando o olho esta adequadamente focalizado, a luz de um objeto externo ao olho forma uma imagem na retina, a visão de padrões é obtida pela distribuição de receptores discretos de luz ao longo da superficie da retina, existem duas classes de receptores, os cones e bastonetes.Cada olho possui cerca de 6 a 7 milhões de cones, e são muito sensiveis a cor, e com estes cones é que os humanos podem distinguir pequenos detalhes, já os bastonetes estão em um número muito maior no olho, existem cerca de 75 a 150 milhões, e são distribuidos pela superficie da retina, os bastonetes servem para dar uma imagem geral do campo de visão, não envolvidos na questão de distingüir cores.A fóvea é uma depressão circular na retina de cercade 1,5 mm de diâmetro, podemos interpreta-lá como uma matriz sensora quadrada de tamanho 1,5mm x 1,5 mm.(Gonzales et al,1977).

A seguir na figura 5 é representado um globo ocular, e de que ele é formado:

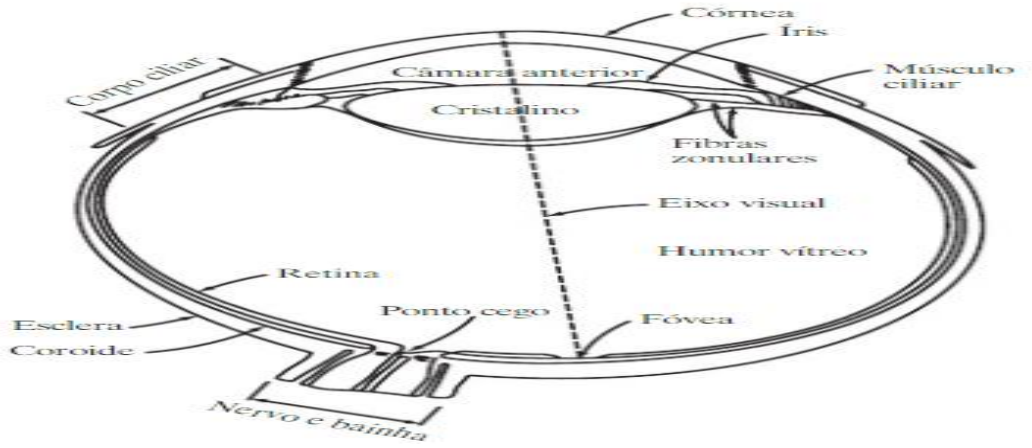

Figura 5 Representação de um globo ocular

Fonte: Processamento digital de images 3 ed

Abaixo outra figura de como a visão humana funciona, expondo atributos que formam a visão, e suas respectivas funções gerais:

Go

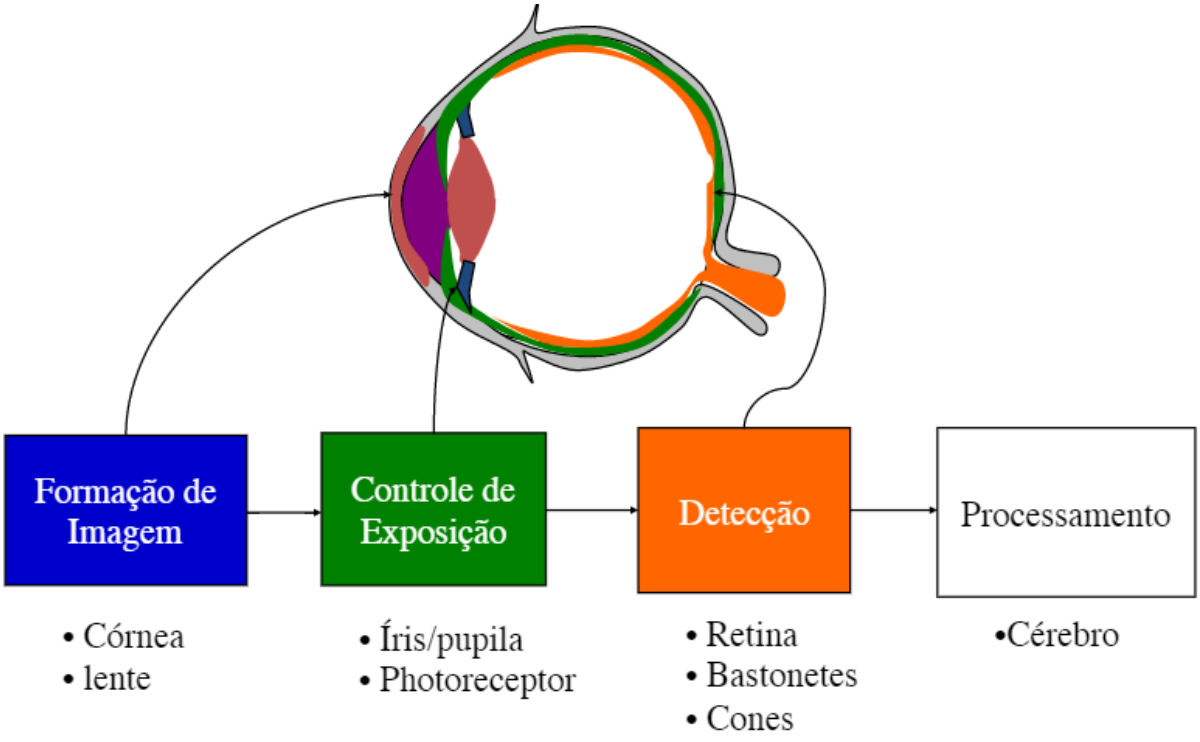

Figura 6 Representação Funções globo ocular

Fonte: https://image.slidesharecdn.com/camera-imaging-technology-basics-130705111021-phpapp02/95/camera-visual-imaging-technology-a-walkthrough-6-638.jpg?cb=1476511153

Gonzales (1977), explica que em uma câmera fotográfica comum, a lente tem uma distância focal fixa, e a focalização para diferentes distâncias é obtida variando a distância entre a lente e o plano-imagem, onde o filme (ou o chip de captura de imagem, no caso de uma câmera digital) se localiza. No olho humano, ocorre o oposto: a distância entre a lente e o plano-imagem (a retina) é fixa, e a distância focal necessária para atingir uma focalização obtida variando adequada é o formato do cristalino (que equivale a uma lente flexível). Isso é realizado pelas fibras zonulares, que achatam ou espessam o cristalino para focalização de objetos distantes ou próximos, respectivamente.

A figura 7 a seguir ilustra como calcular as dimensões de uma imagem formada na retina :

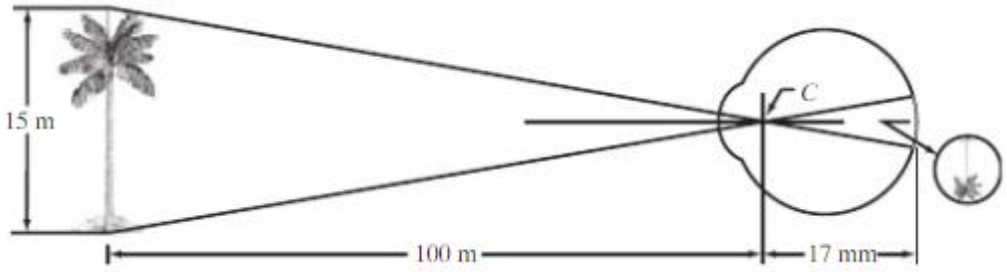

Figura 7 Representação de um olho focalzando em uma árvore

Fonte: Processamento digital de images 3 ed

A figura 7 representa o exemplo que uma pessoa esta olhando para uma árvore de 15 m de altura a uma distância de 100 m, se h for a altura do objeto na imagem formada na retina, a disposição geométrica da figura 7 nos leva a 15/100 = h/17 ou h = 2,55mm.

A seguir na tabela 1 um comparativo entre a visão humana e a visão artificial:

Tabela 1 – Comparativo entre visão humana e visão artificial

Fonte: Processamento digital de imagens 1 ed

	Sistema visual humano	Sistema de visão artificial
Espectro	Limitado à faixa de luz visível (300 nm a 700 nm) do espectro de ondas eletromagnéticas.	Pode operar em praticamente todo o espectro de radiações eletromagnéticas, dos raios X ao infravermelho.
Flexibilidade	Extremamente flexível, capaz de se adaptar a diferentes tarefas e condições de trabalho.	Normalmente inflexível, apresenta bom desempenho somente na tarefa para a qual foi projetado.
Habilidade	Pode estabelecer estimativas relativamente precisas em assuntos subjetivos.	Pode efetuar medições exatas, baseadas em contagem de pixels e, portanto, dependentes da resolução da imagem digitalizada.
Cor	Possui capacidade de interpretação subjetiva de cores.	Mede objetivamente os valores das componentes R, G e B para determinação de cor.
Sensibilidade	Capaz de se adaptar a diferentes condições de luminosidade, características físicas da superfície do objeto e distância ao objeto. Limitado na distinção de muitos níveis diferentes de cinza, simultaneamente.	Sensível ao nível e padrão de iluminação, bem como à distância em relação ao objeto e suas características físicas. Pode trabalhar com centenas de tons de cinza, conforme projeto do digitalizador.
Tempo de resposta	Elevado, da ordem de 0,1 s.	Dependente de aspectos de hardware, podendo ser tão baixo quanto 0,001 s.
2-D e 3-D	Pode executar tarefas 3-D e com múltiplos comprimentos de onda (dentro do espectro de luz visível) facilmente.	Executa tarefas 2-D com relativa facilidade, mas é lento e limitado em tarefas 3-D.
Percepção	Percebe variações de brilho em escala logarítmica. A interpretação subjetiva de brilho depende da área ao redor do objeto considerado.	Pode perceber brilho em escala linear ou logarítmica.

2.4. PROCESSAMENTO DIGITAL DE IMAGEM

Filho et al (1999) explica que o primeiro grande inpulso de uso no processamento de imagem veio em 1962, através dos primeiros computadores digitais de grande porte, no programa espacial norte-americano denominado “Jet Propulsion Laboratory”, o processamento foi utilizado para corrigir imperfeições e distorções das fotos, tiradas da lua que foram transmitidas para a terra através da sonda Ranger.

Atualmente a área de processamento de imagens vem sendo objeto de crescente interesse, por permitir viabilizar grande número de aplicações em duas categorias bem distintas, a primeira é o aprimoramento de informações pictóricas para interpretação humana, e segundo a análise automática por computador de informações extraidas de uma cena.(Filho et al,1999).

Segundo Jain (1989) o processamento de imagem é um campo envolvente com um rápido crescimento de aplicações na área de ciências e engenharia, o processamento de imagem possui a possibilidade de desenvolver uma máquina final, que poderia desempenhar as funções visuais de todos os seres vivos, mas infelizmente ainda é necessário muitas descobertas teóricas e tecnológicas para sermos capazes, de construir tal máquina, entretanto ainda existe uma abundância de aplicativos de processamento de imagem, que podem servir a humanidade, como tecnologias de detecção, reconhecimento, e de antecipar determinados eventos.

Gonzales (1977) explica que na área do processamento de imagens, existem três tipos de processos computacionais, os processos de baixo , médio e alto nivel.Os processos de nivel baixo envolvem operações primitivas, como o pré-processamento de imagens para reduzir o ruído, o realce de contraste e o aguçamento de imagens, um processo de nivel baixo é caracterizado pelo fato de tanto a entrada quanto a saida serem imagens.O processamento de imagens de nivel médio envolve tarefas como a segmentação (separação de uma imagem em regiões ou objetos), a descrição desses objetos para reduzi-los a uma forma adequada para o processamento computacional e a classificação (reconhecimento) de objetos individuais, um processo de nivel médio é caracterizado pelo fato de suas entradas, em geral, serem imagens, mas as saídas são atributos extraidos dessas imagens (isto é, bordas, contornos e a identidade de objetos individuais).E por fim o processamento de nivel alto envolve “dar sentido” a um conjunto de objetos reconhecidos,

como na análise de imagens e, no extremo dessa linha contínua, realizar as funções cognitivas normalmente associadas á visão.

Como explicado anterioremente existem três niveis de processos computacionais, no processamento de imagens, é importante notar que o que esses processos fazem nada mais é que pegar uma determinada imagem, e realizar uma bateria de processos e análises, no entanto se torna importante a explicação de cada parte destes processos para entender melhor o que cada uma faz, e qual a ordem desses processos.

Etapas fundamentais no Processo de imagem:

- Aquisição da Imagem: O primeiro passo é a aquisição da imagem que se deseja realizar a detecção, seja através de uma imagem estática, um video gravado, ou através de uma filmagem em tempo real.
- Pré-Processamento de Imagem: etapa necessária para melhorar a imagem, pois uma imagem sem pré-processamento pode apresentar diversas imperfeições como presença de pixels ruidosos, brilhos não adequados, distorções, entre outros problemas, logo torna-se necessária esta etapa para que aumente as chances de sucesso nos próximos processos
- Segmentação:Etapa necessária para segmentar a imagem, ou dividir a imagem em aspectos de interesse, ou seja se é requisitado a detecção de uma bola de futebol em uma imagem, é nesta etapa em que separamos a bola dos demais itens da imagem.
- Extração de Caracteristicas:Etapa necessária para extrair caracteristicas das imagens resultantes da segmentação, através de descritores,os descritores devem ser representados por uma estrutua correta para o algoritmo de reconhecimento, ou seja os descritores dependem do que irá ser extraido na imagem, como uma cor, uma determinada região,etc
- Reconhecimento e Interpretação: No reconhecimento é atribuido um rótulo a um determinado objeto, como base nas informações de suas caracteristicas que foram definidas nos descritores,na interpretação consiste atribuir um significado a um conjunto de objetos reconhecidos.

3. RASPBERRY PI

O presente capitulo busca ênfase teórica sobre o raspberry pi, mostrando sua história, os objetivos planejados para ele, componentes que o constituem, e outros componentes externos compativeis, como módulos e sensores, também é apresentado os sistemas e alguns programas compativeis, tais como raspbian, retropie, osmc, entre outros.

3.1. HISTÓRIA DO RASPBERRY PI

Segundo Sarthak Jain et al. (2014) , Raspberry é um computador com tamanho de um cartão de crédito, que foi desenvolvido no Reino Unido pela Raspberry Pi Fundation, com o objetivo de estimular o ensino de informática básica nas escolas.

O conceito por trás do raspberry Pi surgiu dos membros do laboratório de computação da universidade de Cambridge, Rob Mullins, Eben Upton, Jack Lang e Alan Mycroft, e em seu desenvolvimento o projeto sofreu várias alterações antes de sua versão final, em 2006 surgiu suas primeiras versões conceituais, que foram construidas com base do microcontrolador Atmel 8-BIT atmMega664, e após 6 anos de desenvolvimento, mais precisamente em fevereiro de 2012 a fundação raspberry pi começava a aceitar pedidos do seu primeiro modelo, o Raspberry Pi Model B, mostrado na (Figura 8).

Segundo Paiva et al.(2014) devido ao seu baixo custo e sua arquitetura o raspberry Pi não vem acompanhado de periféricos, e neste caso, é necessário adquirir separadamente um carregador comum com saida micro-USB, cartão SD pelo fato do raspBerry não possuir disco rigido, além de outros itens conforme for requisitado (mouse, teclado, joystick, etc).

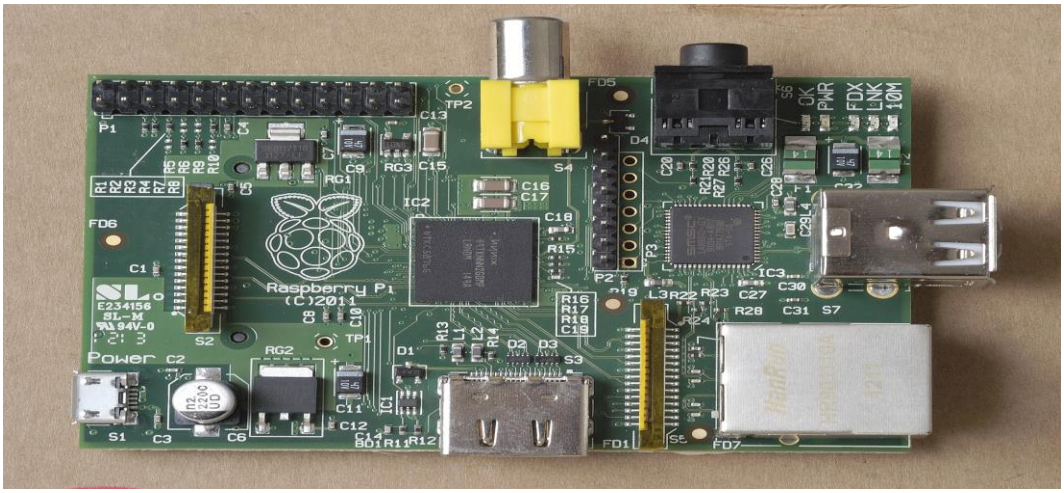

Figura 8 - Primeira Versão do Raspberry Pi B
Fonte: http://www.Raspberrypi.org

Além do modelo B que foi lançado primeiro e com o preço de US$35, também houve o lançamento do modelo A custando US$25, sendo assim o modelo B foi lançado mais caro, porém com algumas especificações que o modelo A não tinha, como podemos ver na tabela 2 a seguir, onde é mostrado as especificações de cada modelo.

Tabela 2 - Comparativo Raspberry Pi B e Raspberry Pi A

Fonte: https://www.element14.com/community/search.jspa?q=raspbeerry+pi

Technical Features	Modelo A	Modelo B
Chip	Broadcom BCM2835 SoC full HD multimedia applications processor	Broadcom BCM2835 SoC full HD multimedia applications processor
CPU	700 MHz Low Power ARM1176JZ-F Applications Processor	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	256MB SDRAM	512MB SDRAM
Ethernet	None	onboard 10/100 Ethernet RJ45 jack
USB 2.0	Single USB Connector	Dual USB Connector
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5mm jack, HDMI	3.5mm jack, HDMI
Onboard Storage	SD, MMC, SDIO card slot	SD, MMC, SDIO card slot
Operating System	Linux	Linux
Dimensions	8.6cm x 5.4cm x 1.5cm	8.6cm x 5.4cm x 1.7cm

3.2. ANÁLISE DE COMPONENTES INTEGRADOS NO RASPBERRY PI

Depois de visto a imagem de um RaspBerry Pi e suas especificações, é necessário um aprofundamento em cada componente que ele é composto para entender melhor sua arquitetura,os componentes serão detalhados conforme os numeros associados a cada um, como é mostrado na Figura 9 a seguir.

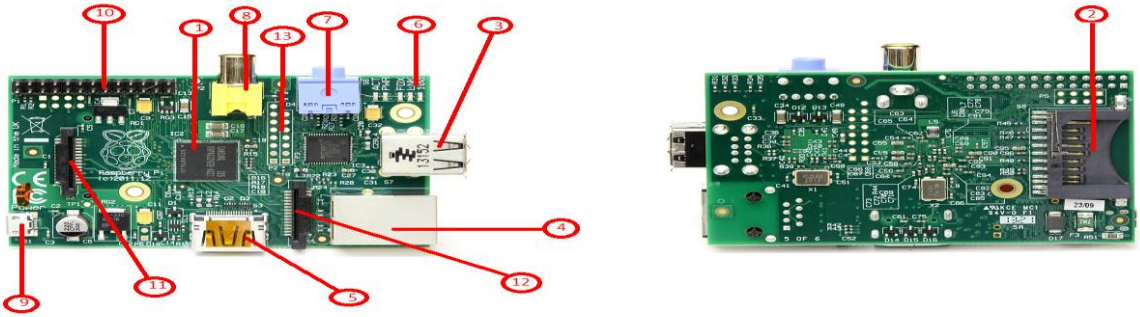

Figura 9- Raspberry Pi B frente e verso

- **Processador:** O processador do RaspBerry Pi é o mesmo processador que é encontrado em um Iphone 3G, ou no Kindle 2, mostrando que suas capacidades são comparáveis a esses dois dispositivos, o processador contêm um sistema-em-um-chip de 700 Mhz de 32 bits, contruido sobre a arquitetura ARM11.Chips ARM apresentam-se em uma variedade de arquiteturas com diferentes núcleos configurados para fornecer diferentes capacidades com preços diferentes.O modelo B possui 512 MB de memória RAM enquanto o modelo A possui apenas 256 MB.(Richardson;Wallace,2012,p.18).

- **Armazenamento :** Para diminuir o custo de produção e aumentar a portabilidade, o Raspberry Pi não contém um disco rigido, o armazenamento é feito através de um cartão de memória SD, que nos modelos mais recentes foi substituido por um microSD.(Richardson;Wallace,2012,p.19).

- **Porta USB :** O moedelo B possui duas portas USB 2.0, enquanto o modelo A possui apenas uma porta. Algumas das primeiras placas do RaspBerry Pi foram lançadas

com um limite de quantidade de corrente que elas poderiam fornecer.Alguns dispositivos USB podem chegar a 500mA.A placa original do Pi suportava 100 mA ou quase, porém com revisões mais recentes foi alcançado até as especificações completas do USB 2.0.(Richardson;Wallace,2012,p.19).

- **Porta Ethernet:** Enquanto o modelo B possui uma porta Ethernet padrão RJ45 (que é um adaptador Ethernet USB embutido). O modelo A não possui, mas é possivel que seja conectado a uma rede com fios por meio de um adaptador de rede Ethernet USB. A conectividade Wi-Fi por meio de um adapador USB externo (dongle) também é possivel.(Richardson;Wallace,2012,p.20).

- **Conector HDMI:** A porta HDMI fornece uma saida de áudio e video digital. São catorze resoluções de video diferentes suportadas, e o sinal HDMI, por meio de adaptadores externos, podem ser convertidos para DVI (que ainda é usado por muitos monitores), video composto por meio de um conector RCA, ou SCART para equipamentos audiovisuais.(Richardson;Wallace,2012,p.20).

- **LEDs de status:** O RaspBerry Pi possui cinco LEDs indicadores de status como é mostrado na Figura 10 a seguir.

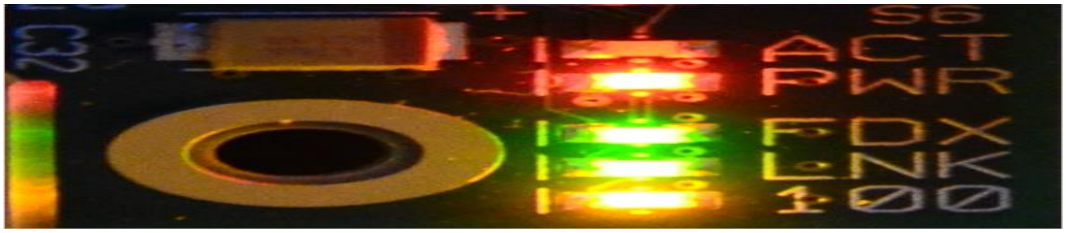

Figura 10- LEDs de Status do Raspberry Pi ligadas.
Fonte: https://www.safaribooksonline.com/library/view/raspberry-pi-hacks/9781449362737/images/rasp_0102.png.jpg

O led do ACT acende na cor verde quando o cartão SD é acessado, o led do PWR acende na cor Vermelha quando é conectado á alimentação de 33 V, o led do FDX, acende na cor verde se o adaptador de rede é full-deplex, o led LNK, acende na cor verde se houver indicação de atividade de rede, e o led do 100 acende na cor amarela se a conexão de rede for 100Mbps.(Richardson;Wallace,2012,p.20,21).

- **Saida de áudio analógico**: O raspbery Pi possui um conector de áudio analógico padrão de 3,5mm, que é destinado a conduzir cargas de alta impedância(como alto-falantes amplificados).Fones de ouvido e alto-falantes sem alimentação, não terão um som de qualidade, porém este problema pode ser considerado como sendo da parte de software pois o som com cabo HDMI possui uma qualidade muito superior.(Richardson;Wallace,2012,p.21).

- **Saida de video Composto**: O raspberry Pi possui um conector-padrão tipo RCA que fornece sinais de video composto NTSC ou PAL.Ambos formatos possuem uma resolução extremamente baixa se comparada com o sinal de HDMI.(Richardson;Wallace,2012,p.21).

- **Entrada de energia:** O Raspberry Pi não possui nenhum tipo de interruptor de alimentação, pois ele utiliza um conector micro USB exclusivo para o fornecimento de energia, a porta microUSB foi escolhida para ser utilizada pois seu conector é barato, e fontes de alimentação USB são fáceis de se encontrar por ser o principal meio de recarregamento, de energia para periféricos hoje em dia como nos smartphones, tablets, entre outros.(Richardson;Wallace,2012,p.21).

- **Pinos de Entrada de Uso geral (GPIO) e outros pinos**: Para realizar as conexões com outros periféricos como leds, camera, sensores entre outros acessórios, o Raspberry Pi possui um GPIO, que nada mais é que um conjunto de pinos responsável por realizar a comunicação de entrada e saida de sinais digitais, o número de pinos vária de acordo com a versão, a primeira versão lançada possuia 26 pinos.(Richardson;Wallace,2012,p.22).

- **Conector de interface serial do display (DSI)**: Este conector recebe um cabo em fita plana de 15 pinos que pose ser usada para se comunicar com uma tela de exibição LCD ou OLED.(Richardson;Wallace,2012,p.22).

- **Conector de interface serial da câmera (CSI)**: Esta porta permite que um módulo de camera seja conectado diretamente a placa.(Richardson;Wallace,2012,p.22).

- **Conector P2 e P3**: Estas duas linhas de conectores são os conectores JTAG de teste, para os chips BroaCom (P2) e de rede LAN9512 (P3).(Richardson;Wallace,2012,p.22).

3.3. COMPONENTES EXTERNOS ESSENCIAIS PARA O RASPBERRY PI

O Raspberry Pi é vendido separadamente de componentes externos, essenciais para seu funcionamento, sendo necessário adquiri-los antes de começar a usa-lo de uma forma plena, a seguir sera mostrado os principais componentes necessários.

- **Fonte de Alimentação**: Este que pode ser considerado o mais importante, para utilizar o Raspberry Pi é necessário utilizar um adaptador USB, que pode fornecer 5V e pelo menos 700mA de corrente (500mA para o modelo A), é necessário que a corrente não seja mais baixa que isso, como exemplo carregadores de celular que fornecem apenas 400 mA, se utilizar uma corrente menor o Raspberry Pi pode até funcionar, porém ele ficará estranho, e pode até travar.(Richardson;Wallace,2012,p23).

- **Cartão SD**: Para utilizar o Raspberry Pi é necessário um cartão de armazenamento de pelo menos 4GB para instalar, o sistema operacional, e deve ser um cartão de classe 4. Com estes cartões é possivel houver tranferência de pelo menos 4MB/seg. .(Richardson;Wallace,2012,p23).

- **Cabo HDMI**: Se tiver que conectar o Raspberry Pi em um monitor, vai ser necessário um cabo HDMI, ou um adaptador apropriado para um monitor DVI, é claro pode ser utilizado também um cabo RCA para saida de video composto, porém devido a baixa qualidade é recomendável o cabo HDMI. .(Richardson;Wallace,2012,p24).

- **Cabo Ethernet**: Se for utilizar a internet no Raspberry Pi é essencial o uso do cabo de ethernet, ou um adaptador para Wi-fi, pelo menos na primeira e segunda versão, ja que a terceira versão conta com uma placa de Wi-fi inbutido. .(Richardson;Wallace,2012,p24).

3.4. DIFERENTES VERSÕES LANÇADAS DO RASPBERRY PI

Atualmente o raspberry Pi possui várias versões (como é mostrado na Figura 11), sendo a sua terceira geração a mais nova até o momento, o Raspberry Pi 3 que sera aprofundada mais a frente

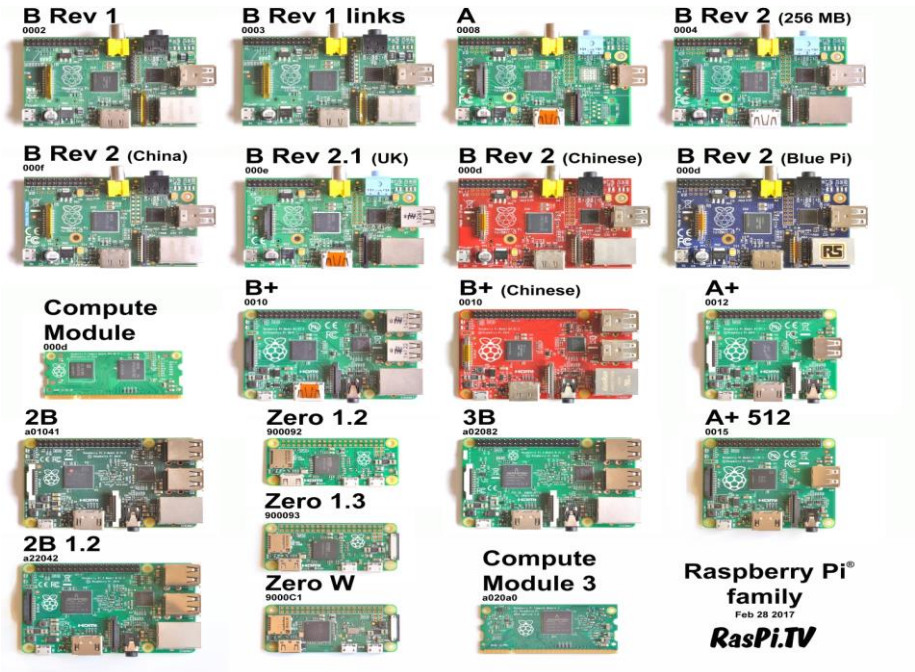

Figura 11- Diferentes Versões do Raspberry Pi já lançadas

Fonte: https://plus.google.com/+Raspberrypibra

3.5. SISTEMAS DISPONIVEIS PARA RASPBERRY

O raspberry pi possui diversos sistemas disponiveis para instalação, cada um com seu objetivo, podendo transformar o raspberry em diversas ferramentas diferentes como em uma central multimidia, um videogame-retro, um servidor dedicado, ou em um desktop comum para utilização, alguns desses softwares disponiveis serão mostrador a seguir :

- **RaspBian**: Sendo esse o sistema operacional principal e oficial do Raspberry, o raspbian foi criada de maneira independente por Mike Thompson e Peter Green e distribuida em 2012, ele é uma versão não oficial do Debian Wheezy armhf, e possui duas versões para instalação, o raspbian jessite lite que é mais leve e não possui um ambiente gráfico, possuindo apenas o terminal e softwares essenciais, e a versão desktop onde possui um ambiente gráfico e ja vem com diversos softwares para educação instalados como Python, Scratch , Mathematica, Sonic Pi, etc.
- **Ubuntu-Mate** : Desenvolvida por Martin Wimpress e Rohith Madhavan, seu foco foram o Raspberry Pi 2 e 3, e baseada no ubuntu armhf,e não o ubuntu “Snappy”, sendo assim o procedimento de instalação de aplicações é feita tradicionalmente pelo apt-get, apesar do ubuntu mate ter mais ferramentas e ser mais customizável e bonito que o rasbian, ele não possui o mesmo suporte, e não é compátivel com todas versões o raspberry como a primeira versão por ser um sistema que exige mais do hardware.
- **Windows 10 IoT Core**: Desenvolvida pela Microsoft o Windows 10 ioT Core, é uma versão otimizada do windows 10 para dispositivos menores com ARM e x84/x64 voltada para o uso na internet das coisas, ele utiliza a API de plataforma universal (UWP), que é extensivel para construçã de diversas soluções, e é compativel com linguagens opensource, e com o visual studio.
- **Kodi**: Kodi é um premiado player de midia e entreterimento gratuito e opensource que vem ficando cada vez mais famoso com seu uso no raspberry para transformar uma tv tradicional em uma smartTV, ele pode ser instalado no linux, android, OSX, e com ele é possivel reproduzir e vizualizar maioria de videos, músicas, podcasts, e arquivos de midia em geral, através da internet.
- **Retropie, Recalbox**: Ambos foram desenvolvidos com intuito de trasnformar o raspberry em um video game retro, sendo assim possuem vários emuladores como

de nintento, super nintendo, mega drive, playstation, atari, dreamcast, entre outros, e é muito utilizado para criação de fliperamas hoje em dia.

Além desses, o raspberry pi possui vários outros sistemas com caracteristicas semelhantes e que podem ser encontrados no site oficial do raspberry pi.

3.6. COMUNIDADE DO RASPBERRY PI

Assim como diversas outras comunidades de softwares, e hardwares, o raspberry possui sua própria comunidade que é muito ativa, e vem colaborando para o desenvolvimento e aprimoramento do raspberry pi através da disponibilização de softwares opensource, bibliotecas, compartilhando projetos realizados, respondendo dúvidas, e disponibilizando tutoriais. No livro “The Raspberry Pi Education Manual” realizado por colaboradores do Instituto autonomo de TI BCS, é citado que, se houver alguma dúvida, ou receio sobre o raspberry pi, o primeiro e melhor lugar para se buscar ajuda é no raspberrypi.org, onde o fórum é bastante movimentado, e possui vários tópicos importantes para quem esta começando neste meio, como dúvidas em relação a GPIO, programação, sistemas operacionais, e os membros são bastante amigáveis e possuem um alto grau de conhecimento para apontar qualquer um com dúvidas para a direção certa, é muito importante citar que para a realização deste projeto, a pesquisa, e consulta de tutoriais foi de extrema importância para atingir o resultado final esperado com êxito, desta maneira se tornou importante a criação de um subcapitulo para lembrar a importância da comunidade em uma determinada tecnologia.

4. PROPOSTA DE TRABALHO

Para a elaboração da montagem e desenvolvimento deste trabalho de conclusão de curso, serão consultados livros, e sites que forneçam informações referentes aos recursos utilizados para a elaboração da construção fisica, e lógica do projeto.

O projeto foi realizado primeiramente buscando os recursos necessários para a montagem da plataforma robótica, para que ele efetuasse suas funções, que eram medir a distância, e efetuar a filmagem.Depois de adquirir as peças necessárias e realizar a montagem, foi realizado uma pesquisa sobre visão computacional com foco em detecção de imagens para aplicar nas filmagens da câmera.

4.1. RECURSOS FISICOS UTILIZADOS

Seram utilizado peças de montagem, componentes eletrônicos e programas com diferentes propósitos,para o desenvolvimento do projeto, itens que serão abordados a seguir, mostrando seus aspectos gerais.

4.1.2. RaspBerry Pi 3 B

O raspberry Pi 3 (Figura 12), lançado em 2016 é o mais potente até o momento, e o escolhido para realizar este projeto pelo fato de possuir um desempenho melhor, e já possuir funcionalidades que os seus antecessores não possuiam, como wifi, e bluetooth inbutido, abaixo é mostrado suas especificações.

- 1GB RAM

- 4 Portas USB
- 40 GPIO pins
- Full Hdmi port
- Ethernet port
- Camera interface (CSI)
- Display interface (DSI)

- Micro sd CARD SLOT

- VideoCoreIV 3D graphics core
- A 1.2 GHZ 64-bit-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE).

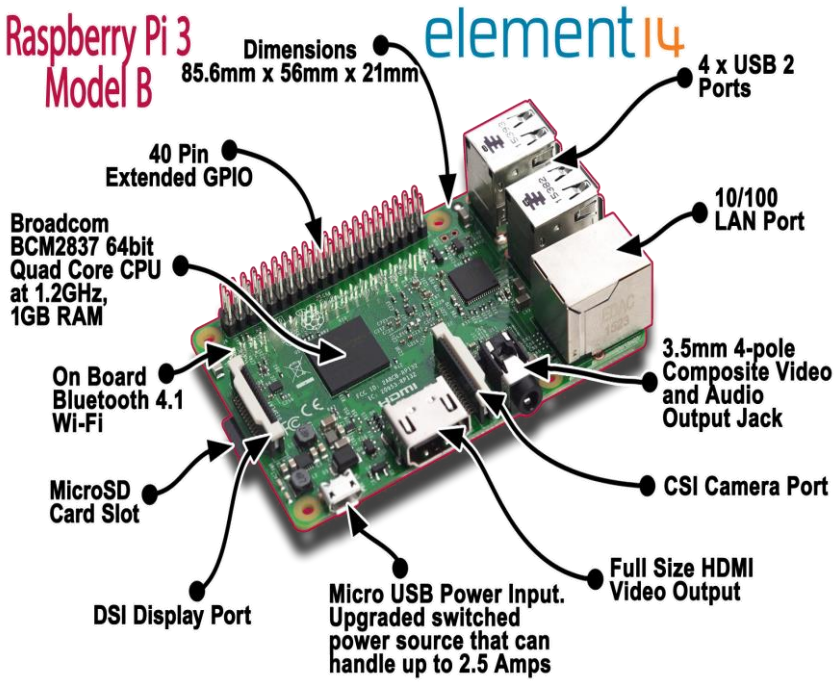

Figura 12-RaspBerry Pi 3 B

Fonte: https://www.element14.com/community/dtss-images/uploads/devtool/diagram/large/Raspberry_Pi_3_and_MathWorks_Learn_to_Program_Pack_Starter_Kit.png

4.1.3. Raspberry Câmera Pi v1

A câmera utilizada para realizar a captura de video sera a Pi câmera v1 (Figura 13), a câmera possui 5 megapixels, com suporte a 1080p30, 720p60 e VGA90, possuindo modo de fotografia e de gravação de video, ele se conecta por um cabo flat de 15 cm com a porta CSI do raspberry Pi, a tabela 3 ilustra as especificações da câmera.

Figura 13- RaspBerry Pi Camera

Fonte: https://www.raspberrypi.org/app/uploads/2017/05/Pi-Camera-front-1-462x322.jpg

Tabela 3 - Especificações RaspBerry Pi Camera

	Camera Module v1	Pixel size	1.4 µm × 1.4 µm
Net price	$25	Optical size	1/4"
Size	Around 25 × 24 × 9 mm	Full-frame SLR lens equivalent	35 mm
Weight	3g	S/N ratio	36 dB
Still resolution	5 Megapixels	Dynamic range	67 dB @ 8x gain
Video modes	1080p30, 720p60 and 640 × 480p60/90	Sensitivity	680 mV/lux-sec
Linux integration	V4L2 driver available	Dark current	16 mV/sec @ 60 C
C programming API	OpenMAX IL and others available	Well capacity	4.3 Ke-
Sensor	OmniVision OV5647	Fixed focus	1 m to infinity
Sensor resolution	2592 × 1944 pixels	Focal length	3.60 mm +/- 0.01
Sensor image area	3.76 × 2.74 mm	Horizontal field of view	53.50 +/- 0.13 degrees

4.1.4. Módulo Ponte H L298N

Para controlar o motor sera utilizado o circuito ponte h (Figura 14), que é um arranjo de 4 transistores, com ele é possivel acionar simultaneamente dois motores ao mesmo tempo, podendo controlar o sentido e velocidade do motor, além de ter um switch para ligar e desligar, dando a possibilidade de controlar a alimentação do motor, abaixo segue as especificações do Módulo Ponte H utilizado no projeto.

- Ci L298N
- Tensão para motores: 5-35V;
- Corrente máxima para os motores: 2A;
- Potencia máxima: 25W;
- Tensão lófica: 5V;
- Corrente lógica: 0-36mA;
- Dimensões: 43x43x27 mm

- Peso:30g.

Figura 14- Ponte H L298N
Fonte: http://blog.vidadesilicio.com

4.1.5. Sensor Ultrasônico HC-SR04

O sensor ultrasônico (Figura 15) será utilizado no projeto para calcular a distância de objetos e obstaculos que forem encontrados, mostrando ao usuário a distância que esta de colidir com algo.

O sensor emite pequenos pulsos sonoros de alta frequência que se propaga na velocidade do som, caso o pulso atinga um objeto, um sinal é refletido de volta para o sensor, então o cálculo da distância é feito pelo intervalo da emissão do sinal e do retorno do sinal ao sensor, abaixo segue as especificações do sensor ultrasônico.

- Alimentação: 5V
- Corrente em Espera:<2mA
- Corrente de Operação:15mA
- Ângulo de efeito: < 15
- Alcance: 2cm – 4m
- Precisão: 3mm
- Dimensão: 45x20x15mm

Figura 15- Sensor de Distância Ultrasônico HCR04
Fonte: https://www.robocore.net/upload/lojavirtual/620_1_H.png

4.1.6. Chassi e Peças menores

Para o esqueleto do da plataforma robótica foi escolhido um chassi pré-moldado em acrilico (figura 16), a escolha deste modelo de chassi foi pelo fato de economizar tempo na montagem, e de possuir vantegens na compra por já vir com os furos e cortes necessários, além de ja vir com algumas peças essenciais transcritas abaixo, junto com outras peças menores compradas separadamente

- 2 Motores DC (3~6V)
- 2 Rodas de borracha
- 1 Roda boba
- 2 Disco de encoder
- 1 suporte de 4 pilha, (além do suporte de pilha que veio foi necessário comprar mais 1 com 4 pilhas, totalizando 8 pilhas.
- Jogo de parafusos e espaçadores.
- 4 Jumper Femêa x Femêa
- 4 Jumper Macho x Femêa
- 1 Miniprotoboard
- 1 Resistor 1k
- 1 Resistor 1k8
- Cabo carregador mini usb cortado
- 8 Pilhas AA

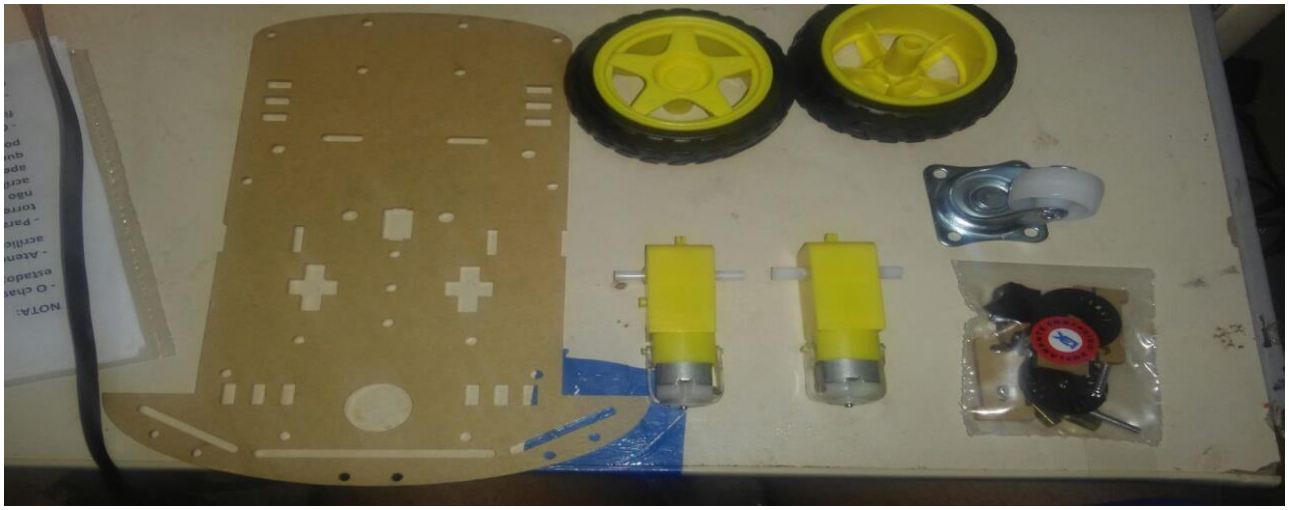

Figura 16- Chassi e peças menores

4.2. RECURSOS LÓGICOS UTILIZADOS

4.2.2. Raspbian

O sistema operacional raspbian (Figura 17) já explorado anteriormente, foi o escolhido para ser o ambiente desktop deste trabalho, pelo fato de ser o mais,estável, ter apoio o oficial da fundação raspberry, e possuir alguns dos softwares necessários já instalados para o desenvolvimento.

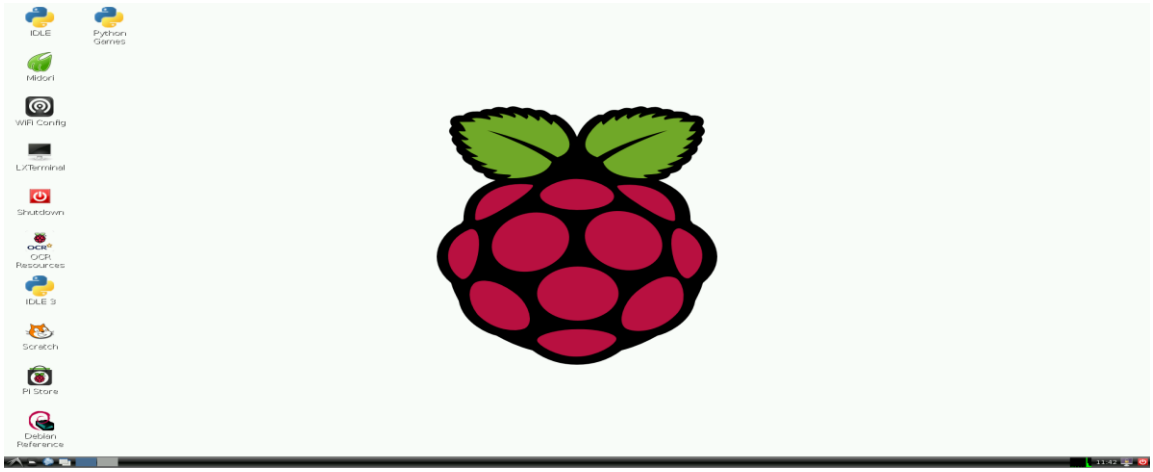

Figura 17-Desktop Raspbian

4.2.3. LINGUAGEM DE PROGRAMAÇÃO PYTHON

A linguagem python foi criada em 1990 por Guido van Rossum, no instituto Nacional de pesquisa para Matemática e Ciência da Computação a Holanda (CWI) e tinha originalmente

foco em usuários como fisicos e engenheiros, python foi concebido a partir de outra linguagem existente na época, chamada ABC (Borges,2010). Python é uma linguagem de programação opensource, de alto nivel, orientada a objetos e de tipagem dinâmica, interpretada e iterativa.

A linguagem python foi escolhida para realização deste projeto por ja vir instalada no raspbian e ter total compatibilidade para trabalhar com os GPIO, sendo assim tornando possivel a criação de scripts para o controle do robo através deles, além de também ser compátivel com a biblioteca opencv que irá ser utilizada nesse projeto para desenvolver a aplicação de detecção de objetos.

4.2.4. BIBLIOTECA OPENCV

OpenCV é uma biblioteca opensource escrita em C e C++,e disponivel para ser executada em Windows, Linux e Mac OS X, o opencv foi projetado para eficiência computacional e com foco forte no tempo real de aplicações, os objetivos que foram planejados com o desenvolvimento da biblioteca foram:

- Pesquisas avançadas fornecendo não só código aberto, mas também um código otimizado para infra-estrutura de visão básica, desta maneira não precisando mais reiventar a roda, pois o que antes era ter códigos dispersos de visão computacional sem um controle, foi reunido tudo somente em um lugar.
- Disseminar conhecimentos de visão fornecendo uma infra-estrutua comum que desenvolvedores poderiam utilizar, de modo que o código seria mais fácil de ler e compreender.
- Aplicações comerciais avançadas baseadas em visão computacional, tornando o desempenho melhor e optimizado, além do código não exigir uma licença comercial, e sim sendo gratuito. (Brafski et al,2008).

A biblioteca possui mais de 500 funções que abrange nas muitas áreas da visão, incluindo fábrica de inspeção de produtos, imagens médicas, segurança, interface do usuário, entre outros, o opencv também contém uma biblioteca de aprendizado de máquina (MLL) completa para o uso geral, que se concentra no reconhecimento estatistico de padrões e no agrupamento, o MLL é essencial para as tarefas de visão computacional. (Brafski et

al,2008).O openCV pode ser resumido em 5 componentes principais no qual é estruturado que é mostrado na figura 18 a seguir.

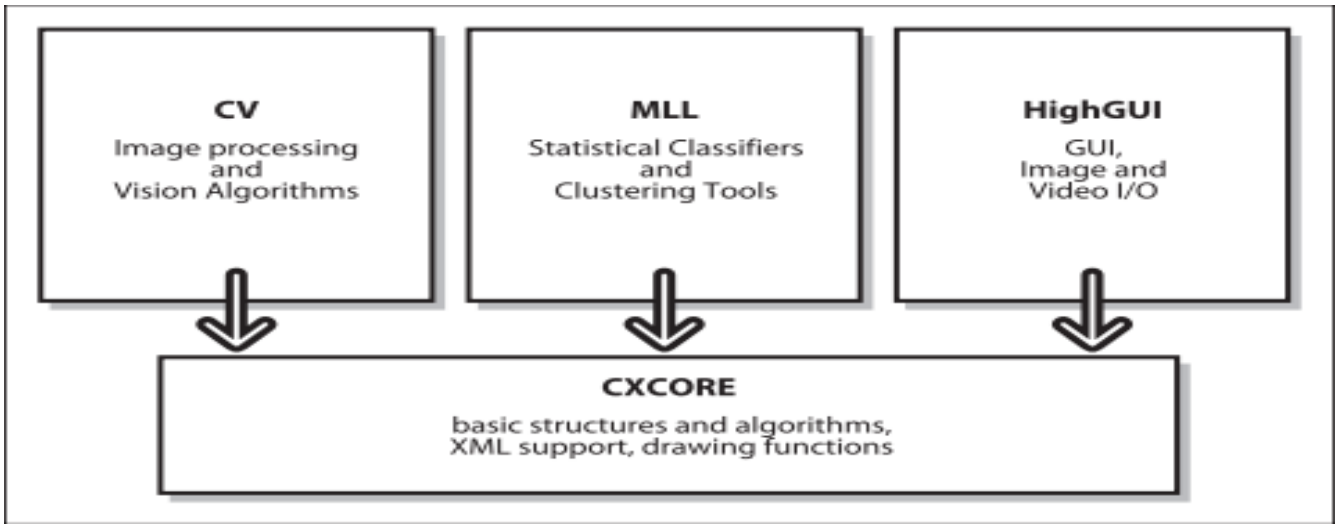

Figura 18-Estrutua Opencv
Fonte: Learning OpenCV 1ed

O primeiro componente (CV) é responsável pelo processamento básico de imagem e possui os algoritimos de nivel alto de visão computacional, o segundo componente (ML) é a biblioteca de aprendizado de máquina, que inclui diversos classificadores estatisticos e ferramentas de agrupamentos, o terceiro componente (HighGUI) contêm rotinas de entrada e saida, e funções para armazenar e carregar video e imagens, e o último componente (CXCORE) contêm as estruturas de dados básicas. (Brafski et al,2008).

4.3. CONSTRUÇÃO FISICA E DESENVOLVIMENTO DA PLATAFORMA ROBÓTICA

No seguinte subcapitulo será abordado a construção do projeto, mostrando o modelo no qual foi baseado, o esqueleto montado, e o projeto final montado

4.3.2. Montagem

Para a contrução do pataforma robótica foi realizada a montagem com base no esboço da figura 19, em que especifica onde é encaixado e posicionado cada peça de acrilico e os demais componentes como os motores, as rodas, e os parafusos, depois foi feito uma esquematização do projeto com o programa Fritzing (Figura 21), onde foi definido onde iria

cada componente eletrônico, e entao foi feita a monstagem com base nestas informações, com o resultado final demonstrado na figura 22.

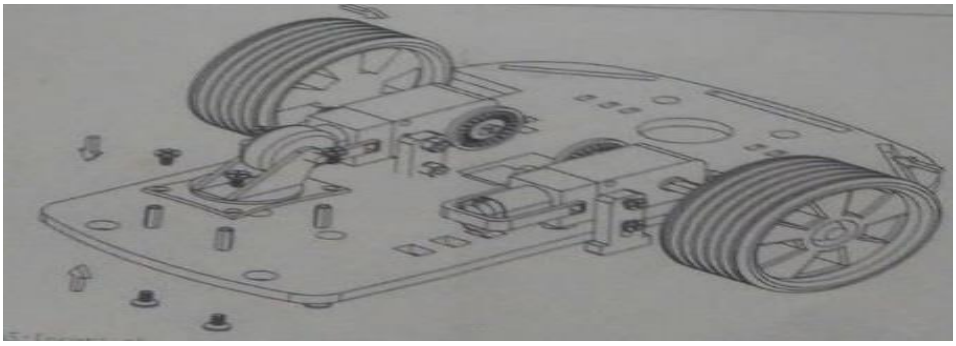

Figura 19- Esboço Chassi Montado

Figura 20- Chassi Montado sem os componentes eletrônicos

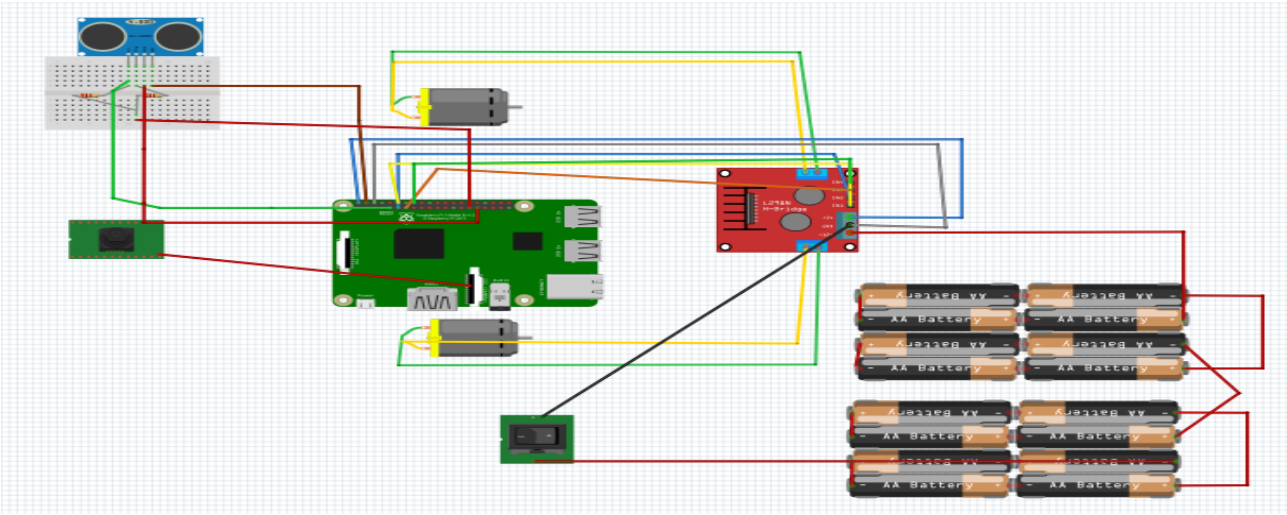

Figura 21 – Esquema do Projeto

Figura 22- Plataforma robótica montada

4.4. DESENVOLVIMENTO

A seguir é apresentado como foi feito o desenvolvimento da parte lógica do projeto, como a programação para o movimento dos motores, para o funcionamento do sensor, o funcionamento da câmera com detecção, e a junção destes elementos para aplicar na página web.

4.4.1. Acesso Remoto ao Raspberry Pi

Para realizar o desenvolvimento da parte lógica do trabalho, foi escolhido realizar o acesso remoto ao raspberry pi ao invés de conecta-lo a um monitor para facilitar a usabilidade e aumentar a produtividade, para realizar o acesso remoto foi utilizado o software VNC viewer, disponivel para Windows, Linux e Mac OX, para realizar a utilização do VNC no raspberry antes é preciso habilitar o modo VNC, para realizar este procedimento é necessário acessar as configurações do raspberry através do terminal, com o comando “sudo raspi-config”, então acessar a opção Interfacing Options > VNC > YES, depois é necessário descobrir o endereço de ip que o raspberry possui na rede e fornece-lo ao VNC.

4.4.2. Programação para o controle dos motores

Para a realização da programação dos motores foi utilizada a linguagem python, que é capaz de se comunicar com as portas GPIO através da biblioteca Rpi.GPIO, e as bibliotecas, getch para captura das teclas digitas, e time para a temporização dos movimentos dos motores, abaixo o código do controle dos motores :

```
#importando as bibliotecas
import RPi.GPIO as GPIO #Biblioteca para possibilitar o uso das portas GPIO
import time  #Biblioteca para controlar o tempo de resposta do motor
import getch  # biblioteca de leitura do teclado para realizar os comandos de movimento

GPIO.setmode (GPIO.BOARD) # Escolher qual modo de GPIO usar para representação dos fios GPIO ou Fisico
GPIO.setwarnings(False)  # desativa mensagens do sistema, como quais portas estão em uso

# Mapeamento dos Jumpers conectados no Raspberry e na ponte H para o controle dos motores
FRENTE_DIREITA = 16
FRENTE_ESQUERDA = 11
TRAS_DIREITA = 18
TRAS_ESQUERDA = 13

def definicao_motores (): #Definicao para definir os pinos da GPIO que estão em OUT
GPIO.setup(FRENTE_DIREITA, GPIO.OUT)
GPIO.setup(FRENTE_ESQUERDA, GPIO.OUT)
GPIO.setup(TRAS_DIREITA, GPIO.OUT)
GPIO.setup(TRAS_DIREITA, GPIO.OUT)

def move_frente ():        #Metodo para mover para frente
GPIO.output(FRENTE_DIREITA, GPIO.HIGH)
GPIO.output(FRENTE_ESQUERDA, GPIO.HIGH)
time.sleep (0.4)  #desliga os motores depois De um tempo para nao ficar ligado direto ao acionar uma tecla
GPIO.output(FRENTE_DIREITA, GPIO.LOW)
GPIO.output(FRENTE_ESQUERDA, GPIO.LOW)


def move_tras ():        #Metodo para mover para trás
GPIO.output(TRAS_DIREITA, GPIO.HIGH)
GPIO.output(TRAS_ESQUERDA, GPIO.HIGH)
time.sleep (0.4)  #desliga os motores depois e um tempo
GPIO.output(TRAS_DIREITA, GPIO.LOW)
GPIO.output(TRAS_ESQUERDA, GPIO.LOW)
```

```
def move_direita ():  # Metodo mover para direita
GPIO.output(FRENTE_ESQUERDA, GPIO.HIGH)
time.sleep(0.2)
GPIO.output(FRENTE_ESQUERDA, GPIO.LOW)

def move_esquerda ():   # Metodo mover para a esquerda
GPIO.output(FRENTE_DIREITA, GPIO.HIGH)
time.sleep(0.2)
GPIO.output(FRENTE_DIREITA, GPIO.LOW)

def le_tecla():   # Metodo para ler teclado
  while True:
  tecla_comando = getch.getch ()  # getch le o teclado que esta sendo pressionado
  if tecla_comando  == 'w':
      move_frente ()
       if tecla_comando = 's':
      move_tras()
     if tecla_comando = 'd':
      move_direita
       if tecla_comando = 'a':
      move_esquerda

definicao_motores () # inicializa os metodos da definicao de motores e da leitura de tecla
le_tecla
```

Como pode ser visto o código para o controle dos motores não é muito extenso,e nada muito complexo, se resumindo a determinar os estados de cada porta através de métodos para cada comportamento.

4.4.3. Programação para medir a distância com o sensor

Para a programação do sensor de distância também foi utilizada a linguagem python, com as mesmas bibliotecas que foi utilizada na programação dos motores, exceto a biblioteca de captura de teclas, antes de exibir o código do sensor é importante explicar algumas funções definidas no método “roda_medição”, que não são funções muito complexas, porém a explicação ficará muito extensa para deixar de comentario no código.O sensor possui duas portas para realizar a medição de distância, o sensor gera um pulso que dura 10 micro segundos como é ilustrada na figura 22, da documentação do sensor.

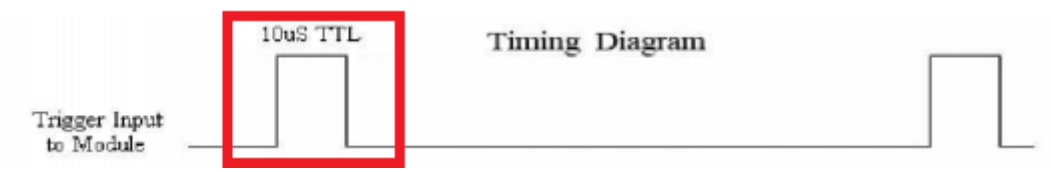

Figura 23- Diagrama do gatilho de pulso do sensor
Fonte: http://www.micropik.com/PDF/HCSR04.pdf

Ou seja, para efetuar a medição, depois de colocar a porta “1” do GPIO do sensor em alta é necessário aguardar 0.000010 microsegundos, que é o tempo que ele faz a emissão do pulso antes de colocar a porta em baixa novamente, depois de efetuar a emissão do pulso é necessário aguardar este sinal na porta 2, a figura 23 ilustra o processo em que a porta 1 (emissor) emite um pulso, este pulso colide com um objeto e volta para a porta 2 (receptor), para realizar este procedimento foi necessário a criação de dois laços onde o primeiro fica em um loop salvando o tempo enquanto a porta 2 esta em baixa, e logo após receber um sinal de alta (começar receber o inpulso de volta), ele salva este último tempo, sai do primeiro loop e vai para o segundo onde faz o mesmo procedimento, porém salvando o tempo em que a porta esta em alta, quando acaba a recepção do sinal ele salva o último tempo também, assim conseguindo o tempo exato do começo da recepção do sinal, e o tempo exato em que ele para de receber

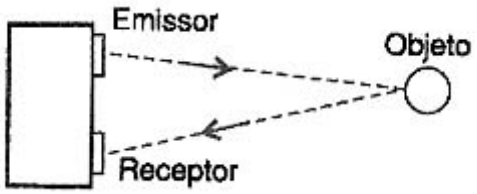

Figura 24- Ilustração do pulso do sensor
Fonte: http://www.micropik.com/PDF/HCSR04.pdf

Depois de descoberto o valor do começo da recepção de sinal e o tempo final, é necessário fazer a subtração um do outro (tempo_inicial _pulso – tempo_final_pulso) assim obtendo o tempo da duração do pulso, então depois de encontrar mais esse valor é feito o cálculo para encontrar a distância, que consiste na duração do pulso multiplicado pela sua velocidade, que no caso é a velocidade do som (343m/s), e então o resultado é dividido por 2 pois só é necessário o tempo a partir que o pulso bate no objeto e volta para o receptor e não do emissor também, exemplo da equação : (Duração_pulso * 34300/2), código a seguir.

```
#importando as bibliotecas
import RPi.GPIO as GPIO #Biblioteca para possibilitar o uso das portas GPIO
import time #Biblioteca para controlar o tempo de emissão de pulsos do sensor


GPIO.setmode(GPIO.BOARD) # Escolher qual modo de GPIO usar para representação dos fios GPI
GPIO.setwarnings(False) # desativa mensagens do sistema, como quais portas estão em uso

# Variaveis criadas para receber o posicionamento dos pinos
ECHO = 29
TRIG = 31

# configura os pinos de saida e de entrada (ECHO = ENTRADA, TRIG = SAIDA)
def setup_sensor();
GPIO.setup(ECHO, GPIO.IN)
GPIO.setup(TRIG,GPIO.OUT)

# Função para realizar a medição
def roda_medicao ():
global distancia_cm
distancia_cm = 0
while True: # inicialização do loop para efetuar a emissão de pulsos em 2 e 2 segundos
      time.sleep(2)
      GPIO.output(TRI,GPIO.HIGH)
      time.sleep(0.000010)
      GPIO.output(TRIG.GPIO.LOW)

      while (GPIO.input (ECHO)==0): # 1 loop (tempo que começa a receber)
              pulso_inicial=time.time()
      while (GPIO.input(ECHO) ==1): # 2 loop (tempo que termina de receber impulso)
      pulso_final=time.time()
      #Calculos para medir a distância
      duracao_pulso = pulso_final - pulso_incial
      distancia_cm - duracao_pulso/2 * 34300
      distancia_cm = round(distancia_cm, 0)
      print(distancia_cm, 'cm    ', end="\r")
  # Chamada das funções
      setup_sensor ()
      roda_medicao()
```

4.4.4. Programação da câmera com detecção de objetos

Para realizar a filmagem e detecção de objetos foi utilizada a biblioteca openCV já mencionada antes, na parte do script escrito em python as principais funções definidas foram as de ativar a câmera do raspberry pi com a biblioteca “picamera” disponivel para python, (a raspberry câmera também pode funcionar sem necessidade de executar script, através de linha de comando) e a partir disso configurar a sua execução como a resolução, os frames por segundo,etc.Depois das configurações básicas é definido a parte mais importante do código para conseguir analisar e detectar as fotos o haarCascade.

O modelo haarCascade foi desenvolvido por Paul Viola e Michael Jones em 2001, e consiste em um algoritmo de aprendizado de máquina em cascata no qual treina várias imagens, imagens positivas e negativas para assim detectar o objeto requisitado, este modelo é essencial para projetos de detecção de caracteristicas em imagens, ou em video pois economiza tempo de desenvolvimento, ao não precisar criar um algoritmo com aprendizado de máquina sofisticado, e ficando ao desenvolver somente usar esta

ferramenta para treinar as imagens.Treinar imagens com haarcascade não é algo muito dificil, porém trabalhoso pois, quanto mais imagens forem selecionadas para ser analisadas, melhor será o resultado,final na detecção.

O primeiro passo para se criar um haarcascade é conseguir imagens positivas do objeto que se deseja detectar, uma imagem positiva nada mais é que imagens que contenham o objeto que se deseja detectar, um axpecto importante de se ressaltar no treinamento de um haarcascade é que a quantidade de imagens necessárias para se treinar depende muito do que se vai detectar, por exemplo uma logo de uma marca como no caso da figura 24 precisaria de poucas imagens, pois uma logo é uma imagem estática e no máximo pode ficar de ponta cabeça ou outras posições similares, já em casos onde se procura treinar um haarcascde para detectar rostos humanos como exemplo necessitária de uma enorme quantidade de fotos (algo entre cinco mil, dez mil, ou mais imagens), pois diferentemente de um logo, o rosto humano possui diversas feições como estar sorridente, triste, nervoso, e também cada pessoa possui um rosto diferente, mais diferentes ainda quando são de raças diferentes como brancos, negros e japoneses, na figura 27 é mostrado um exemplo do haarcascade treinado por mim, onde foi utilizada 460 fotos positivas, e 651 fotos negativas, e mesmo com esse grande volume de imagens a detecção não funciona muito bem, sendo necessário muito mais fotos para não ter erros.O segundo passo é conseguir imagens negativas, images negativas nada mais é que imagens que não contenham o objeto que se deseja detectar, estas imagens servem para treinar o fundo que o objeto a ser detectado estará, a quantidade de imagens para treinar um haarcascade de maneira satisfatória tem que ser ainda maior que as positivas, pelo fato de um mesmo tipo de objeto a ser detectado poder possuir diversos fundos, como exemplo da figura 26.

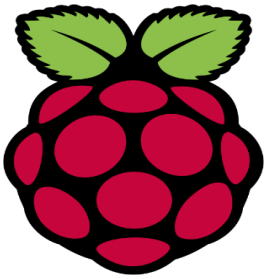

Figura 25- Logo Raspberry

Figura 26- Rosto Detectado

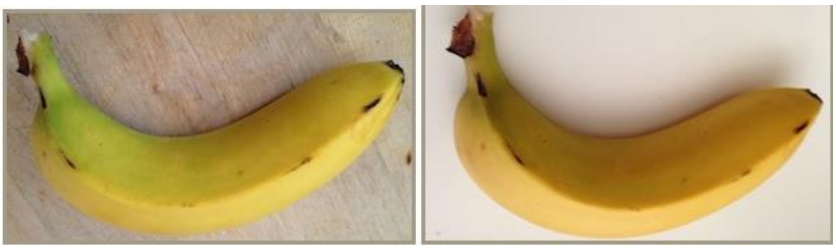

Figura 27- Imagem em diferentes fundos
Fonte: http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html

- No terceiro passo é necessário pegar amostras das imagens negativas, este passo consiste em escrever o caminho e nome de todas as imagens negativas em um arquivo .txt, também é necessário adquirir amostras das imagens positivas, porém diferente de como funciona com as imagens negativas com as positivas precisamos gerar os dados que a imagem tem em um formato binário onde é passado as dimensões em que o objeto para detecção esta, existem diversas formas de se realizar este procedimento, podendo ser realizado até mesmo manualmente, o openCV fornece uma ferramenta para realizar este processo chamada “opecv_createsamples” em que pode gerar estes dados automaticamente.Por último a etapa final é treinar o classificador, o openCV possui duas funções para realizar este processo o “opencv_haartraining” e o “opencv_traincascade” sendo este último capaz de rodar em multithread, e reduzindo o tempo de treinar o classificador, após selecionar qual usar basta passar os parâmetros de onde esta as amostras positivas e negativas e então começar a treinar o classificador, depois de terminado o processo é gerado um arquivo em XML que é necessário no scipt do programa de detecção. É importante ressaltar que este processo pode demorar muito tempo, em

casos onde se usa cinco,dez, vinte mil fotos pode demorar dias dependendo da capacidade do computador, por isso é bastante recomendável o contrato de um serviço de servidor para realizar este trabalho em determinados casos. No caso deste trabalho foi utilizada para a detecção 460 imagens positivas para detecção de rostos, 371 imagens positivas para detecção de relógio de pulso e 115 imagens positivas para detecção de moedas, e em todos os treinamentos foram utilizados 652 fotos negativas.

Após inserido o haarcascade, que fica em formato XML, foi efetuado o método para pegar aquilo que foi detectado dentro dos padrões do classificador e circular com um retângulo, código de detecção:

```
from picamera.array import PiRGBArray #importacao modulos camera
from picamera import PiCamera
import cv2 #Importação OpenCV
import time
#----------Definindo configurações basicas da camera como resolução e volocidade de frame rate
camera = PiCamera()
camera.resolution = (320, 240)
camera.framerate = 30
rawCapture = PiRGBArray(camera, size=(320, 240))

display_window = cv2.namedWindow("Detecção de Objetos")
#Passando o arquivo cascade
cascPath = "Classificadorxml"
ObjetoCascade = cv2.CascadeClassifier(cascPath)


time.sleep(1)

for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
```

```
    imagem = frame.array

    #Metodo de detecção e desenhando o retangulo a imagem quando encontra o objeto
    gray = cv2.cvtColor(imagem,cv2.COLOR_BGR2GRAY)
    objetos = ObjetoCascade.detectMultiScale(gray, 1.1, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(imagem,(x,y),(x+w,y+h),(255,0,0),2)

    #Mostrando na Tela
    cv2.imshow("Objeto", imagem)
    key = cv2.waitKey(1)

    rawCapture.truncate(0)

    if key == 27:
        camera.close()
        cv2.destroyAllWindows()
        break
```

4.4.5. Integração com a WEB

O acesso ao controle da plataforma robótica será via web como é mostrado na figura 27, o software para efetuar a comunicação da internet com o script em python é o servidor Flask, que por ter sido desenvolvido em python tem a vantagem de poder realizar a chamada dos script do controle, sensor e câmera, diretamente no código, desta maneira o controle da plataforma robótica poderá ser realizado através de smartphones e computadores através do endereço de ip que estará com o programa hospedado

Figura 28- Interface da aplicação na Web

5. CONCLUSÃO

Devido a grande evolução em que a área de inteligência artificial vem ganhando, seja em aprendizado de máquina, aprendizado profundo, visão computacional entre outros tópicos neste meio, cada vez mais é apresentado soluções e ferramentas para trabalhar neste meio, e neste contexto é importante o profissional de TI nunca estar aquém desta realidade, sempre tentando aprender ou estar por dentro de tecnologias com estes paradigmas.

A visão computacional é um campo muito importante para a evolução humana, cada vez será visto sistemas de segurança mais seguros através de detecções faciais mais sofisticadas, e no campo da medicina poderemos ter descobertas de doenças mais cedo através de exames, por meio de detecção de padrões de doenças cada vez mais sofisticadas também.Aprender como esta técnica funciona cada vez se torna mais importante, pois possivelmente a visão computacional é uma das áreas dentro do campo de inteligência artificial que mais cresce, e vai crescer no meio técnológico, e entender como a visão humana se assimila com a visão artificial ajuda a compreender como a visão artificial funciona e como é possivel melhora-lá com base em um comparativo entre as duas visões,e aplicar novos métodos para sofisticar a detecção e reconhecimento de imagens através do processamento de imagens

O raspberry Pi é uma ferramenta que vem cada vez mais se tornando popular, abrindo a mente das pessoas para gerar projetos engenhosos, e que se continuar assim é por meio destes projetos que poderemos ver tecnologias que poderam até mesmo ajudar a salvar o planeta com questões como poluição, controle de eventos que podem ser previstos, automação de atividades de pessoas com necessidades especiais, entre outros.Poderemos ver cada vez mais marcas buscando fazer seu próprio mini computador, e componentes para estes periféricos, cabendo aos desenvolvedores que iram utilizar estas ferramentas o que podem contruir para ajudar no mundo em alguma questão, ou simplesmente por simples prazer.

Desenvolver um trabalho que envolve construir algo fisico, pode ser um grande desafio pois tanto a parte fisica quando a lógica do projeto tem que estar em harmonia, e todos estes desafios foram encontrados no desenvolvimento deste projeto, e apesar das grandes dificuldades e problemas encontrados, ter terminado este projeto foi de grande importância para mim, e importante para demonstrar que no curso de ciência da computação, os

trabalhos finais não precisam ser necessariamente só desenvolvimento de software em área de trabalho/web, e pesquisa, mais sim construção e montagem de hardware também, trabalhar com o raspberry pi também me apresentou uma comunidade fantástica com muitas informações acerca deste meio, e mostrou como o raspberry pi pode ser versátil, com diversos tipos de sistemas disponiveis,e vários tipos de projetos inovadores que a comunide cria e compartilha, trabalhar com a visão computacional aliado a isso foi de grande valia para mim, pois conhecer um pouco da tecnologia por meio do desenvolvimento de detecção de objetos, e como a tecnologia foi desenvolvida me fez admirar todos os responsáveis pelo crescimento e o desenvolvimento desta área.

6. REFERÊNCIAS

PAZOS, FERNANDO. **Automação de sistemas e robóticas**. 1 ed. BRASIL: AXCEL BOOKS, 2002. 392 p.

BALLARD, BROWN. **Computer Vision**. 1 ed. New York: PRENTICE-HALL, 2010. 957 p.

BUILDBOT. **Como utilizar o sensor ultrasônico hc sr04**. Disponível em: <http://buildbot.com.br/blog/como-utilizar-o-sensor-ultrasonico-hc-sr04/>. Acesso em: 18 mar. 2017.

SZELISKI. **Computer Vision Algorithms and Applications**. 1 ed. Washington: SPRINGER, 2010. 957 p.

REAL PYTHON. **Face Recognition with Python, in Uner 25 Lines of Code**. Disponível em: < <http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>. Acesso em: 23 jul. 2017

RICHARDSON, Matt; , Shawn Wallace. **Getting started with raspberry pi**. 1 ed. [S.L.]: O'Reilly Media, 2012. 176 p.

VIDA DE SILICIO. **Módulo ponte h l298n – o primeiro passo para montar seu robô com arduino.** Disponível em: <http://blog.vidadesilicio.com.br/arduino/modulo-ponte-h-l298n-arduino/>. Acesso em: 18 mar. 2017.

CANAL TEACH**. O que é raspberry pi.** Disponível em: <https://canaltech.com.br/o-que-e/hardware/o-que-e-raspberry-pi/ 15/03/2017>. Acesso em: 17 mar. 2017.

RASPBERRY PI. **Products**. Disponível em: <https://www.raspberrypi.org/products/>. Acesso em: 17 mar. 2017.

GONZALES, WOODS. **Processamento Digial de Imagens**. 3 ed. BRASIL: PEARSON, 1977. 644 p.

DEMAAGD, OLIVER, OOSTENDORP, SCOTT. **Practical Computer Vision**. 1 ed. Cambridge: O'REILLY, 2012. 165 p.

FILHO, NETO. **Processamento digital de imagens**. 1 ed. New York: Brasport, 1999. 957 p.

JAIN, Sarthak; VAIBHAV, Anant; GOYAL, Lovely. **Raspberry Pi based interactive home automation system through E-mail**. IEEE, New Delhi, fev. 2015. Disponível em: <http://ieeexplore.ieee.org/document/6798330/>. Acesso em: 17 mar. 2017.

THE GUARDIAN. **Raspberry pi computer sale british**. Disponível em: <https://www.theguardian.com/technology/2012/feb/29/raspberry-pi-computer-sale-british 15/03/2017>. Acesso em: 17 mar. 2017.

TRUETEX**. Raspberry pi.** Disponível em: <http://www.truetex.com/raspberrypi>. Acesso em: 17 mar. 2017.

PAIVA, Omir; MOREIRA, Renata. Http://www.scielo.br/pdf/rb/v47n2/pt_0100-3984-rb-47-02-99.pdf. **Raspberry Pi: dispositivo de 35 dólares para visualização de imagens DICOM**,[S.L],p.2,mar./abr.2017.Disponível em:<http://www.scielo.br/pdf/rb/v47n2/pt_0100-3984-rb-47-02-99.pdf>.Acesso em: 18 mar. 2017.

RASPBERRY PI. **Raspbian**. Disponível em: <http://www.scielo.br/pdf/rb/v47n2/pt_0100-3984-rb-47-02-99.pdf>. Acesso em: 18 mar. 2017.

VIDA DE SILICIO. **Sensor distância ultrassônico hcsr04**. Disponível em: <http://www.vidadesilicio.com.br/modulos/sensores/proximidade/ultrassom-hc-sr04.html>. Acesso em: 18 mar. 2017.

GAGAN.**Training Haar-Cascade**. Disponível em: <https://singhgaganpreet.wordpress.com/2012/10/14/training-haar-cascade/>. Acesso em: 23 jul. 2017.

CODING ROBIN. **Training your own opencv Haar Classifier**. Disponível em: < https://realpython.com/blog/python/face-recognition-with-python/ >. Acesso em: 24 jul. 2017

MICROPIK. **Ultrasonic ranging module hc - sr04**. Disponível em: <http://www.micropik.com/pdf/hcsr04.pdf>. Acesso em: 25 jul. 2017.

FERNANDES, Leandro Augusto Frata. **Visão Computacional**. UFF, Niterói, p. 33, fev.2012. Disponível em: <http://www2.ic.uff.br/~julius/icc/vcomp.pdf>. Acesso em: 20 jul. 2017.