

USO DE INTELIGÊNCIA ARTIFICIAL PARA O DESENVOLVIMENTO DE UM SISTEMA ROBÓTICO COM CAPACIDADE DE RASTREAR INDIVÍDUO

Data de aceite: 02/05/2024

Lucas G. E. Vieira

Instituto Federal de Educação, Ciência e Tecnologia do Maranhão- IFMA

Almir S. S. Neto

Instituto Federal de Educação, Ciência e Tecnologia do Maranhão- IFMA

Francisco S. Viana

Instituto Federal de Educação, Ciência e Tecnologia do Maranhão- IFMA

RESUMO: A robótica tem sido de grande ajuda ao homem para completar inúmeras tarefas, ela é empregada em diversos lugares como hospitais, aeroportos, fábricas, entre outros. A visão computacional é utilizada em conjunto com a robótica a fim de obter informações do ambiente, entretanto, isso requer um grande poder de processamento, visto que grande parte dos trabalhos desenvolvidos recentes na área de detecção de objetos utilizam de redes neurais robustas, inviabilizando o uso de sistemas embarcados de baixo custo. Este artigo apresenta um sistema de baixo custo para detecção e rastreamento de uma pessoa usando uma câmera RGB em tempo real, para isso desenvolveu-se um sistema

de controle de movimento de robô utilizando apenas os parâmetros obtidos da detecção.

PALAVRAS-CHAVE: Robótica; Detecção de objetos; Rastreamento; MobileNet; Baixo custo

INTRODUÇÃO

A Robótica é um campo que reúne várias áreas do conhecimento, tais como mecânica, eletrônica, informática, inteligência artificial, engenharia de controle, entre outras.

A robótica é capaz de proporcionar maior flexibilidade, melhoria da qualidade de trabalho e aumento da segurança ocupacional (SIEGWART *et al.*, 2011). Por este motivo, a robótica tem sido utilizada em um grande número de processos industriais, tais como: manutenção de produtos, soldagem, aplicação de produtos (pintura, colagem...), carga e descarga de máquinas, mecanização, processos de corte, montagem de produtos, paletização, medição, inspeção e etc. (TURIEL *et al.*, 2002).

Embora na atualidade os robôs industriais executem múltiplas tarefas com grande precisão e rapidez, a principal desvantagem que apresentam é a sua falta de mobilidade, podendo atuar apenas dentro da sua área de trabalho, entendendo-a como o volume do espaço definido através os pontos acessíveis pelos elementos do robô (SIEGWART *et al.*, 2011).

Dentro da robótica está a robótica móvel, um ramo que se concentra no estudo da mobilidade de robôs. Nele, a maioria das aplicações se concentra tanto na teleoperação de robôs, quanto em seu comportamento autônomo (BRADSKI; KAEHLER, 2008).

Na robótica móvel, um dos aspectos fundamentais é a obtenção de informações do ambiente em que o robô opera. Para isso, robôs móveis contam com múltiplos sensores, como sensores de proximidade, pressão, sensores de luz... Apesar da quantidade de informações que podem ser extraídas desses sensores serem limitadas, eles têm sido utilizados em um grande número de trabalhos porque as informações desses sensores são muito fáceis de processar e interpretar. Uma das opções de sensoriamento mais importante hoje é a visão computacional, pois de uma imagem é possível extrair uma grande quantidade de informações como distância, localização, posição, formas e cores do objeto.

As principais desvantagens da visão computacional em robôs são o custo computacional necessário para extrair e processar as informações da imagem, bem como a dependência das condições ambientais, em termos de iluminação, presença de objetos, etc. Esse custo não seria um problema em outras áreas, mas na robótica móvel é, uma vez que a maioria das operações em robôs deve ser realizada em tempo real.

Neste trabalho, é apresentado um sistema robótico de baixo custo capaz de lidar com processamento de imagens, assim como utilizar desse recurso para seguir pessoas.

COMPONENTES DO SISTEMA

Raspberry Pi

O Raspberry Pi é um computador em placa única do tamanho de um cartão de crédito, mas que possui todos os componentes que um computador de custo mais elevado. Esta é uma placa de baixíssimo custo, tornando-a extremamente atrativa, versátil e viável para desenvolvimento de projetos (LIMA *et al.*, 2020).

O modelo utilizado nessa pesquisa trata-se do Raspberry Pi 3 B+, portando um processador Cortex-A53 (ARMv8) quad-core de 1.4GHz, 1GB de memória RAM, 40 pinos GPIO, 4 portas USB, entrada para cabo Ethernet, suporte a Wi-Fi, entrada para fonte DC de 5V/2.5A e saída HDMI.

OpenCV

OpenCV é uma biblioteca de funções de programação voltadas principalmente para a visão computacional em tempo real. Desenvolvida originalmente pela Intel, a biblioteca é multiplataforma e gratuita para usar sob a licença de código aberto BSD (BRADSKI; KAEHLER, 2008).

Ela possui mais de 2.500 algoritmos otimizados, que incorporam um conjunto abrangente de funções relacionadas tanto a visão computacional clássica, quanto a atual. Esses algoritmos podem ser usados para detectar e reconhecer rostos, classificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmera, etc. Todas as estruturas de *ar-rays* OpenCV são convertidas para *arrays* Numpy. Portanto é possível combinar operações realizadas no Numpy com o OpenCV, o que diminui imensamente o número de limitações (SEKHAR; RAO, 2019). Utilizou-se essa biblioteca visando ler, mostrar e fazer um pré-processamento da imagem obtida através da câmera.

MobileNet

Para a detecção de pessoas utilizou-se o modelo MobileNet. Este trata-se de uma classe de modelos eficientes, que se concentra principalmente em aplicativos móveis e sistemas embarcados de visão computacional. O motivo principal da utilização desse modelo é a razão entre a eficiência e o custo de processamento. Para fazer essa otimização ela usa um conceito novo para reduzir a sua complexidade computacional: Convolução Separável em Profundidade.

Para melhor compreensão, será feita uma revisão do modelo, baseado no artigo (HOWARD *et al.*, 2017).

Convolução separável em profundidade

é uma convolução em profundidade seguida por uma convolução pontual, seu modelo é representado na Figura 2.1.

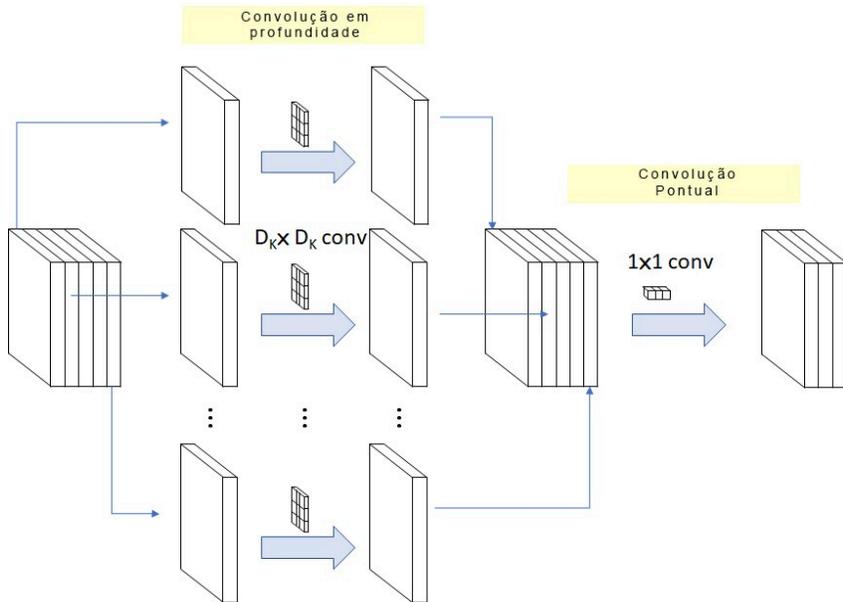


Figura 2.1: Convolução em profundidade seguida por uma convolução pontual. Imagem retirada e modificada de (SCIENCE, 2018)

Convolução pontual é a convolução espacial $D_K \times D_K$ em canais. Supondo que na Figura 2.1, tenhamos 5 canais, então tem-se 5 convolução espacial $D_K \times D_K$. Convolução pontual na verdade, é a convolução 1×1 para alterar a dimensão.

Com a operação acima, o custo de operação é:

$$D_K \cdot D_K \cdot M \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

onde M: Número de canais de entrada, N: Número de canais de saída, D_K : Tamanho do kernel e D_F : Tamanho do mapa de características.

Para convolução padrão, é:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2.1)$$

Assim, a redução do cálculo é:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (2.2)$$

Quando $D_K \times D_K$ é 3×3 , 8 a 9 vezes menos computação pode ser obtida, mas com apenas uma pequena redução na precisão.

Estrutura do modelo

Exceto a primeira camada, que se trata de uma convolução tradicional, as demais camadas da arquitetura MobileNet são baseadas em convoluções DSC. A última é completamente conectada, seguida por uma Softmax.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 512$
	Conv dw / s2	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 1024$
	Conv dw / s2	$3 \times 3 \times 1024$ dw
	Conv / s1	$1 \times 1 \times 1024 \times 1024$
	Avg Pool / s1	Pool 7×7
	FC / s1	1024×1000
	Softmax / s1	Classifier
		$1 \times 1 \times 1000$

Figura 2.2: Arquitetura do modelo Mobilenet. Imagem retirada de (SCIENCE, 2018)

Através da Figura 2.2, nota-se a utilização da estratégia de regularização *batchnormalization*, caracterizada pela normalização em lote (BN) e ReLU aplicadas após cada convolução.

Robô

Visando montar um sistema de baixo custo, utilizou-se um robô genérico, exibido na figura 2.3, com 2 motores de 3-6V. Para controle dos motores empregou-se um Driver Ponte H conectado ao Raspberry e 6 pilhas alcalinas em série para alimentação.

Além disso, uma webcam foi acoplada ao protótipo para obtenção das imagens.

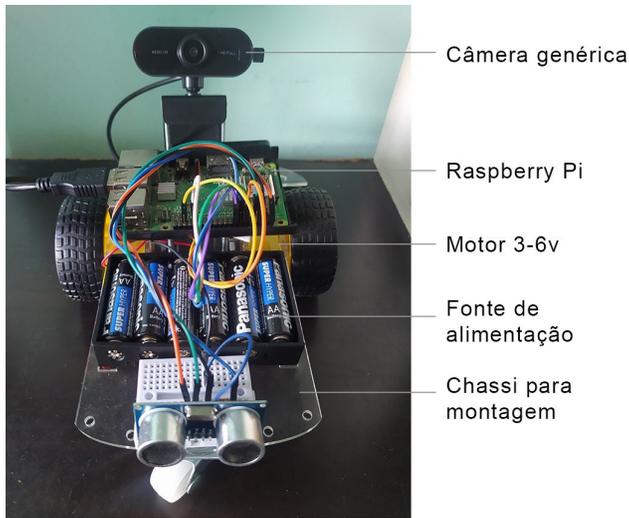


Figura 2.3: Protótipo do robô.

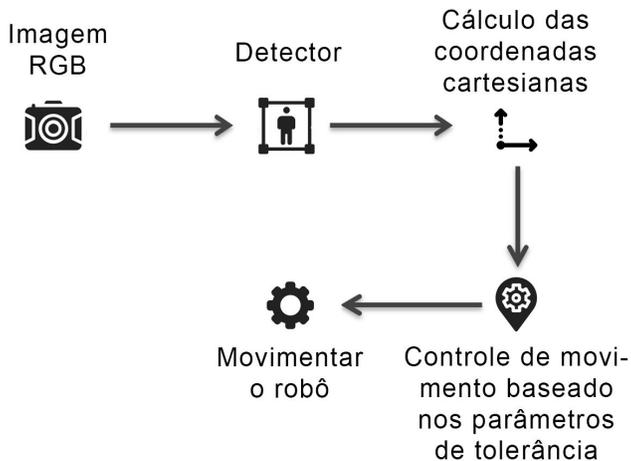


Figura 3.1: Fluxograma do código desenvolvido

TRABALHO DESENVOLVIDO

O fluxograma completo para o robô que segue pessoas pode ser dividido nas seguintes quatro partes (fig. 3.1).

A Rede Neural Convolutiva MobileNet foi utilizada para detectar pessoas em um determinado quadro RGB. Ao passar a imagem RGB no modelo e uma pessoa ser detectada, é retornado a coordenada cartesiana da caixa delimitadora. Para que sejam feitas as estimativas e correções, é calculada a centroide com os valores obtidos.

O sistema de coordenadas cartesianas do robô pode ser observado na figura 3.2.

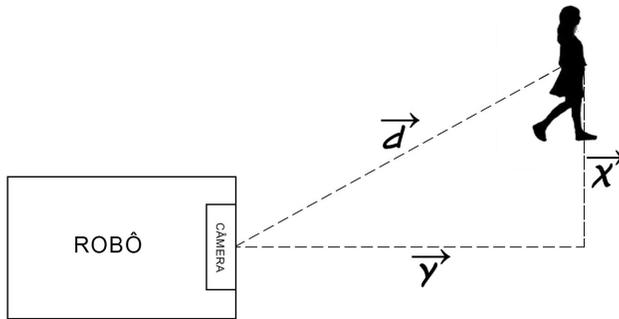


Figura 3.2: Sistema de coordenadas do robô

O controle de movimento é feito a partir de um mecanismo de acionamento diferencial. Esse sistema é baseado em parâmetros definidos previamente, e que podem ser ajustados de acordo com as necessidades do usuário, estes preceitos serão chamados de tolerância.

Quando o usuário estiver fora do limite definido como tolerável, serão feitos ajustes no movimento do robô para que os valores da centroide esteja dentro dos parâmetros estabelecidos, para facilitar o entendimento esse processo foi ilustrado na figura 3.3, as linhas verdes são os limites determinados anteriormente.

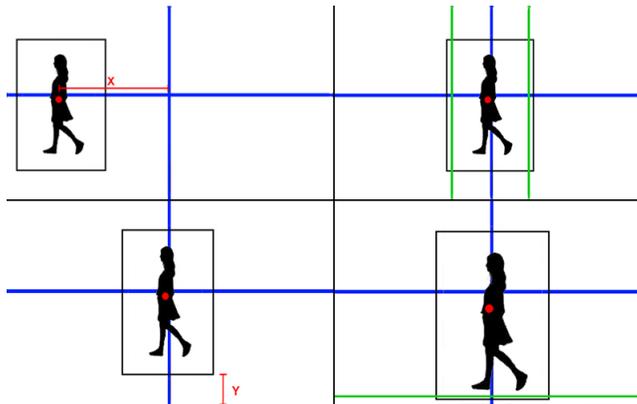


Figura 3.3: Mecanismo de acionamento diferencial

Para controle de quando seguir em frente ou parar, o Y final é utilizado como base. Quando esta coordenada estiver acima dos parâmetros estabelecidos, o robô andará até o Y final ficar a baixo da tolerância.

O manuseio de curvas é feito com base nos parâmetros relacionados ao eixo X. O posicionamento do usuário será estimado utilizando a média entre o X inicial e final, este valor será comparado com os parâmetros de tolerância, caso seja maior o robô virará à esquerda e caso seja menor o robô virará à direita, esses movimentos serão feitos até que a centroide esteja dentro da projeção estipulada.

CONCLUSÃO

O objetivo geral da pesquisa foi alcançado, como apresentado nesse artigo, um sistema de baixo custo baseado em visão computacional e aprendizado pro- fundo para detecção e rastreamento de pessoas em tempo real foi desenvolvido.

Esse sistema viabiliza que mais pesquisas e protó- tipos possam ser desenvolvidos, afinal toda a neces- sidade de equipamentos caros e robustos está sendo eliminada.

Na versão atual, o sistema robótico rastreia uma pessoa através do modelo MobileNet e mediante os resultados obtidos é calculado as coordenadas car- tesianas e o valor médio das mesmas. A partir des- ses dados é feito o controle de movimento do robô, um sistema de navegação autônomo baseado em vi- são computacional foi desenvolvido com o propósito de manter o baixo custo do protótipo, descartando a necessidade de sensores adicionais e/ou câmeras *depth sense*.

AGRADECIMENTOS

Os autores agradecem ao IFMA (Instituto Federal de Educação, Ciência e Tecnologia do Maranhão) pelo apoio ao desenvolvimento do projeto.

REFERÊNCIAS

BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer Vision with the OpenCV Library**. Estados Unidos: O'Reilly Media, 2008.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

LIMA, F.; DIAS, W.; MORENO, E. Implementação de um cluster embarcado usando a plataforma raspberry pi. **Anais da VI Escola Regional de Alto Desempenho do Rio de Janeiro**, p. 11–15, 2020.

SCIENCE, Towards Data. 2018.

SEKHAR, C.; RAO, K. V. Detection of vehicle intrusion using opencv. **arXiv preprint arXiv:1704.04861**, v. 8, p. 1596–1599, 2019.

SIEGWART, R.; NOURBAKSH, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. Londres: MIT press, 2011.

TURIEL, J.; FRAILE, J.; PERAN, J. Applications of robotics: Latest trends and new perspectives. **DYNA**, p. 61–68, 2002.