

BRAÇO ROBÓTICO COM VISÃO COMPUTACIONAL COMANDADO POR VOZ PARA APLICAÇÕES URBANAS, RURAIS E INDÚSTRIA 4.0

Thalita Zanetti Menechelli¹;
Profa. Ana Paula Zaniboni Barreto²;
Prof. Dr. Edio Roberto Manfio³

¹Aluna da Etec Astor de Mattos Carvalho

²Professora da Etec Astor de Mattos Carvalho

³Professor da Faculdade de Tecnologia de Garça

Área do Conhecimento: Linguagens de Programação

Palavras-chave: Visão Computacional; OpenCV; Robô de conversação.

INTRODUÇÃO

Este estudo pretende mostrar como a visão computacional aliada ao Python e à biblioteca OpenCV (ANTONELLO, 2020; BARELLI, 2018) tem grande importância e muita utilidade para apresentar soluções inovadoras tecnológicas desde soluções mais simples às mais complexas na 4ª Revolução Industrial, conhecida como Indústria 4.0 e que através do aprendizado de máquinas, é possível realizar o reconhecimento de formas e cores através de comandos de voz aliados a um braço robótico. Esta linha de estudo está de acordo com as necessidades do Projeto Fapesp 2017/19136-0 outorgado em 2018, desenvolvido e concluído na própria IES e no IPMet de Bauru, e que teve por finalidade coletar dados de chuvas de granizo em áreas do Estado de São Paulo e Paraná, com ênfase na região de Garça-SP.

OBJETIVOS

O objetivo deste projeto é a implementação de um sistema de Visão Computacional a um braço robótico que faz parte ao projeto Solar. Como objetivos específicos tem como meta detectar formas geométricas e as cores verde e amarela através de uma câmera integrada a partir de um microcomputador de baixo custo, um Raspberry pi que trabalhará juntamente com a plataforma Arduino pelo método de aprendizagem semi-supervisionada, ou seja, às vezes terá auxílio de interação humana, no caso o comando por voz, para que o braço possa operar.

METODOLOGIA

Para a realização do presente projeto serão utilizados uma câmera, um computador compacto, um microcontrolador e um braço robótico. A câmera será o modelo Pi, compatível com o Raspberry Pi e como microcontrolador será aplicado o Arduino Mega. O protótipo de atuador será um braço robótico com 7 articulações cuja estrutura básica encontra-se finalizada. O protótipo para a implementação da Visão Computacional é um braço robótico inspirado na anatomia humana. O controle por PLN permitirá que movimentos ou ações de mãos e braços - que parecem triviais aos seres humanos - sejam estudados em partes e aprimorados paulatinamente. Atualmente, o operador envia os comandos ao braço por voz ou texto escrito a partir do robô Solar que está instalado no servidor. O robô, por sua vez, solicita os dados aos protótipos que retornam imediatamente, seja via USB em ambiente controlado, seja via rádio quando à distância.

O Solar é um robô de conversação desenvolvido em C# que utiliza o reconhecedor o Coruja (FALA BRASIL, 2019). A busca de palavras-chave é feita por meio de tabelas *hash* (MORENO, 2017, ZIVIANI, 1999) e há um recurso de aprendizado de máquina - *Machine Learning* - em implementação usando Python (SILVA; MANFIO; GUERRA, 2018) por alunos da Fatec. Os recursos de PLN utilizados no Solar permitem que possa atender a comandos nas linguagens escrita e

falada e responder por texto escrito ou por síntese de voz. Ele constitui-se em uma IHC bastante natural ao operador (MORENO et al., 2017) e estará presente em versões mais avançadas.

RESULTADOS OBTIDOS E DISCUSSÃO

O principal de objetivo deste projeto que era a implementação de um sistema de Visão Computacional a um braço robótico foi atingido e os resultados foram satisfatórios. Todas as etapas de desenvolvimento e programação das plataformas Raspberry e Arduino foram completadas.

O código responsável por ligar a webcam com a biblioteca OpenCV. Ele foi escrito no programa PyCharm em um computador arquitetura X86 e enviado para o Raspberry.

Especificamente o comando `#!/usr/bin/env python3` permitiu a funcionalidade dessa etapa.

A câmera do Raspberry em operação depende de um processo de compilação cruzada. Após esse procedimento ele foi enviado para o Raspberry através de uma conexão `src`. Para que o arquivo executável funcionasse diretamente no Raspberry, foi necessária uma autorização desse mesmo arquivo através do comando `chmod +x <detectaCorCirculo.py>`, comando responsável por permitir a execução do arquivo através de linhas de comando em um outro sistema operacional.

Assim como na visão humana, a disponibilidade de luz interfere no reconhecimento de cores. Houve testes de falso positivo em que duas cores foram detectadas no mesmo ambiente, mas em horários diferentes. Com o mesmo programa executado na parte da noite, notou-se que apenas a cor verde foi detectada. Houve também a identificação individual da cor amarela e das cores azul, amarela e vermelha em conjunto.

Finalizado o treinamento com as cores, procedeu-se com a configuração do braço robótico. Com a câmera e braço robótico previamente posicionados em ambiente com iluminação adequada, objetos coloridos (azul, amarelo e vermelho) foram colocados no campo de detecção (visão) da câmera. Uma vez detectada a cor, o Raspberry envia um sinal ao Arduino que o processa e movimenta o braço para a coleta do objeto.

Em outro teste, restringiu-se a coleta do objeto a apenas uma cor. Programou-se o sistema para proceder com a coleta apenas quando o objeto fosse amarelo. Depois o mesmo teste foi feito com as cores vermelha, azul e verde. O sistema respondeu bem nos 4 testes, embora com tempos de processamento diferenciados.

A cor que apresentou mais falhas na identificação é a verde, que resulta em vários falsos positivos com o amarelo. Duas cores de fácil identificação pelo sistema foram a vermelho e a azul. O amarelo, embora seja também facilmente identificável, apresentou algumas poucas falhas.

CONSIDERAÇÕES FINAIS

Pode-se dizer que o uso desse projeto é absolutamente viável para muitas aplicações educacionais, domésticas e até industriais, uma vez que atuadores de diversos formatos pode manipular ou movimentar objetos por cor de modo significativamente eficiente. Os testes com formato, embora também tenham sido realizados, necessitam mais pesquisas pois requerem maior capacidade de processamento e tempo de execução.

REFERÊNCIAS BIBLIOGRÁFICAS

ANTONELLO, Ricardo. Introdução à Visão Computacional com Python e OpenCV. Instituto Federal Catarinense – Luzerna. Acesso em mar.2020.

ARDUINO. Arduino Mega 2560 Rev3. Disponível em: <<https://store.arduino.cc/usa/mega2560-r3>>. Acesso em 24 mar. 2019.

BARELLI, Felipe. Introdução à Visão Computacional: Uma abordagem prática com Python e OpenCV. São Paulo: Casa do Código, 2018.

FALA BRASIL. Disponível em: <<http://labvis.ufpa.br/falabrasil/>>. Acesso em: 02 jan. 2019.

MORENO, Fabio Carlos; GUERRA, Marcos Paulo Guimarães; MANFIO, Edio Roberto; MORAIS, Marcos Vinícius Bueno de. IHCs dedicadas a energias sustentáveis. In Revista ef@tec.. ISSN: 2317-451X, vol. 7. n. 1, 2017. Garça, 2017.

MORENO, F. C. Visual Tahs: ferramenta para analisar a eficácia de buscas das funções hash em um Léxico para Língua Natural. Dissertação (Mestrado em Ciência da Computação). Universidade Estadual de Londrina, 2017.

RASPBERRY. Camera module. Disponível em:
<<https://www.Raspberrypi.org/documentation/hardware/camera/>>. Acesso em 24 mar. 2019a.

RASPBERRY. Raspberry Pi B+. Disponível em:
<<https://static.Raspberrypi.org/files/productbriefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>>. Acesso em 24 mar. 2019b.

SZELISKI, Richard. Computer Vision: Algorithms and Applications. Springer: August 18, 2010.

ZIVIANI, Nivio. Projeto de algoritmos: com implementação em Pascal e C. 4. ed. São Paulo: Pioneira, 1999.