



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CAMPUS SEDE  
CAPÍTULO ESTUDANTIL IEEE RAS UFCG  
PROCESSO SELETIVO DE INTEGRANTES DO CAPÍTULO ESTUDANTIL  
IEEE RAS UFCG**

**SEGUNDA ETAPA  
MISSÃO DE ESPECIALIZAÇÃO  
MISSÃO 4.0**

**DETECTAR OBJETOS POR COR**

**CANDIDATA: Mércia Regina da Silva**

**CAMPINA GRANDE, 2024.**

## SUMÁRIO

1. INTRODUÇÃO .....	3
2. OBJETIVOS .....	4
3. METODOLOGIA EXPERIMENTAL.....	5
4. REFERÊNCIAS .....	7

## **1. INTRODUÇÃO**

Este diretório possui dois arquivos: `detectar-cor` e `maskGreenYellow`

Na visão computacional a influência da luz interfere diretamente na variação da cor.

Desse modo, o código "`detectar-cor`" permite você identifique o limiar máximo e mínimo de cada cor que pretende rastrear.

Esses valores (max e min) serão necessários para você ajustar as cores no `maskGreenYellow`, caso queira rastrear outras cores.

Originalmente, o código "`maskGreenYellow`" realiza o processo de visão computacional, rastreando as cores: verde e amarelo.

Após identificar as cores, é criado uma máscara retangular sobre as respectivas cores.

## 2. OBJETIVOS

Criar uma primeira aplicação com OpenCV. O Object Tracking by color é uma aplicação em que objetos são reconhecidos por meio das suas cores. Além disso, deve exibir a uma parte da trajetória do objeto detectado.

Materiais necessários:

- ✓ Python3;
- ✓ Bibliotecas;
- ✓ Download OpenCV: Linux;
- ✓ Download Numpy.

Atividade:

- ✓ Detectar objetos por cor;
- ✓ Desenhar um retângulo nos objetos detectados;
- ✓ Criação de uma interface gráfica para criação de máscaras;
- ✓ Exibir uma parte da trajetória do objeto.
- ✓ Enviar os códigos comentados:
- ✓ Explicação do que o código faz;
- ✓ Explicação do que cada linha do código faz.

### 3. METODOLOGIA EXPERIMENTAL

Para que um robô visualize o ambiente, junto com a detecção do objeto, a detecção de sua cor em tempo real também é muito importante.

Por que isso é importante?

Em carro com direção automática, para detectar os sinais de trânsito.

A detecção de várias cores é usada em alguns robôs industriais, para realizar tarefas de pick-and-place na separação de objetos de cores diferentes.

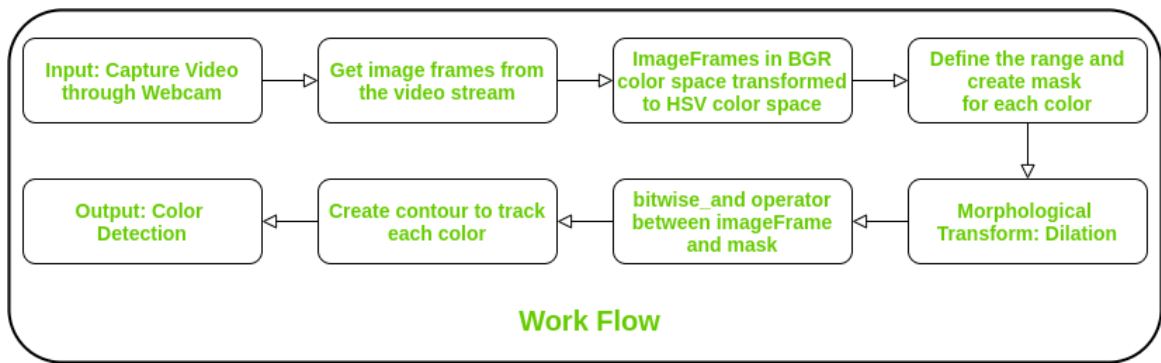
Esta é uma implementação de detecção de várias cores (aqui, apenas as cores Green (Verde) e Yellow (Amarelo) foram consideradas) em tempo real usando a linguagem de programação Python.

Bibliotecas Python usadas:

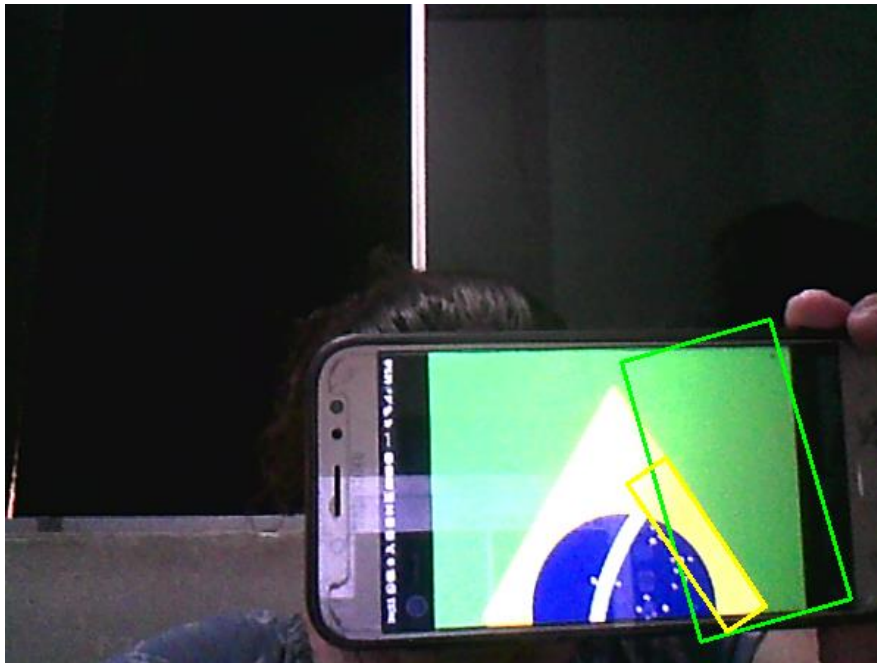
- ✓ NumPy;
- ✓ OpenCV-Python.

#### **Descrição do Fluxo de Trabalho:**

- Etapa 1: Entrada: Capture vídeo pela webcam;
- Etapa 2: Leia o fluxo de vídeo em quadros de imagem;
- Etapa 3: Converta o imageFrame em BGR (espaço de cores RGB representado como três matrizes de vermelho, verde e azul com valores inteiros de 0 a 255) para espaço de cores HSV (hue-saturation-value). Matriz descreve uma cor em termos de saturação, representa a quantidade de cor cinza nessa cor e o valor descreve o brilho ou intensidade da cor. Isso pode ser representado como três matrizes no intervalo de 0-179, 0-255 e 0-255, respectivamente.
- Etapa 4: Defina o intervalo de cada cor e crie a máscara correspondente.
- Etapa 5: Transformação Morfológica: Dilatação, para remover ruídos das imagens.
- Etapa 6: Bitwise\_and entre o quadro da imagem e a máscara é realizada para detectar especificamente aquela cor particular e separar outras.
- Etapa 7: Cria um contorno para as cores individuais para exibir a região colorida detectada de maneira distinta;
- Etapa 8: Saída: Detecção das cores em tempo real.



**Resultado:**



## **4 REFERÊNCIA**

Consultas realizadas para realização da Missão 4.0:

Acesso: 24.03.2024 à 27.03.2024

<https://acervolima.com/deteccao-de-varias-cores-em-tempo-real-usando-python-opencv/>

<https://pyimagesearch.com/2014/08/04/opencv-python-color-detection/>

Consultas GitHub:

<https://github.com/MariaEduardaDeAzevedo/detector-de-cores/blob/master/README.md>

<https://github.com/riosmarcel>