Luiz Felipe Nagatani Rabello

Relatório LAMIA - Módulo 19

O card 19 se trata de um projeto de reconhecimento e filtragem de cores utilizando da biblioteca opencv para utilizar a webcam do computador como método de obter dados e do sistema de cores HSV.

HSV

É um sistema de cores formado por Hue(matiz), saturation e value.

Hue: uma variável que comporta todas as cores do espectro, mudar a hue significa mudar a tonalidade da cor, variando de vermelho para magenta.

Saturação: define o quanto a cor será proeminente no resultado final, quanto menor a saturação mais perto de uma cor neutra estará, quando a saturação estiver em 50% a cor resultante será um misto da cor escolhida no hue com cinza.

Value: Determina o brilho da cor resultante, caso variando de 0 a 100, com 0 sendo preto e 100 sendo branco.

Projeto

No começo do vídeo é inicializada a câmera através do comando cv2.videocapture, sendo escolhido a webcam do computador, com cada imagem sendo lida frame por frame utilizando um loop while infinito até que seja pressionada a tecla esc para a interrupção da execução.

```
cap = cv2.VideoCapture(0)

print(cap)

while True:
    _, frame = cap.read()

    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

Para o reconhecimento de cores foi selecionado o sistema de cores HSV através do comando cv2.cvtColor(cv2.COLOR_BGR2HSV), foi necessário mapear as cores no formato HSV para servirem como máscaras a fim de isolá-las.

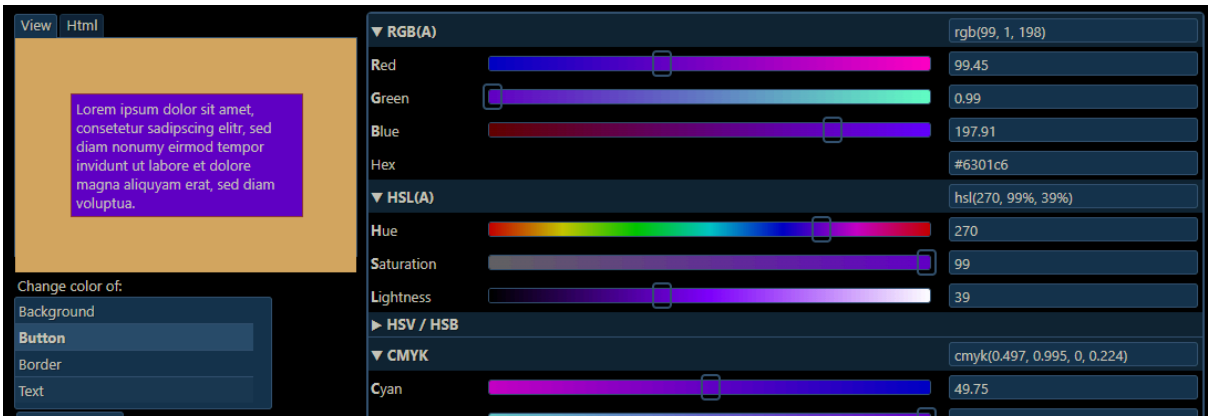
```
# Red color
low_red = np.array([0, 48, 51])
height_red = np.array([0, 255, 255])
red_mask = cv2.inRange(hsv_frame, low_red, height_red)
red = cv2.bitwise_and(frame, frame, mask=red_mask)

# Blue color
low_blue = np.array([94, 82, 2])
height_blue = np.array([126, 255, 255])
blue_mask = cv2.inRange(hsv_frame, low_blue, height_blue)
blue = cv2.bitwise_and(frame, frame, mask=blue_mask)
```

A aplicação das máscaras é realizada através do cv2.bitwise_and(x, y, z) onde se recebe duas imagens x e y, mesclando-as ao mesmo tempo que se aplica uma máscara z.

O breve vídeo foi útil para conhecer as funções do opencv e integrá-las a interpretação de imagens ao vivo, ao invés de processar apenas imagens fornecidas em arquivo.

Para fins de experimentação criei uma máscara para a cor roxa, para encontrar um valor razoável para ser utilizado em HSV, utilizei um visualizador encontrado no link "https://colorizer.org/"




```
# Roxo
low_roxo = np.array([70, 68, 35])
high_roxo = np.array([70, 255, 255])
roxo_mask = cv2.inRange(hsv_frame, low_roxo, high_roxo)
roxo = cv2.bitwise_and(frame, frame, mask = roxo_mask)
```