



Anhanguera

# COMPUTAÇÃO GRÁFICA

Prof. Newton San Juan

# GRUPO

Breno Renan da Cunha    **RA:** 231910411919

Gustavo Da Silva Gouvêa    **RA:** 227486011919

Jailson Alves Dos Santos    **RA:** 230278611919

# COMPUTAÇÃO GRÁFICA?

- A **computação gráfica** é a área da computação destinada à geração de imagens em geral — em forma de representação de dados e informação, ou em forma de recriação do mundo real. Ela pode possuir uma infinidade de aplicações para diversas áreas, desde a própria informática, ao produzir interfaces gráficas para software, sistemas operacionais e sites na Internet, quanto para produzir animações e jogos.

# PROPOSTA DE PROJETO

- Localizar e recolorir objeto(s) simples em Imagens.
  - Criação de um software para detecção de objetos, por seleção geral de um tom de cor específico, Será utilizada a linguagem python juntamente com a biblioteca OpenCV para localizar a área onde a cor se encontra e mudá-la para uma cor pré-determinada.

# PROPOSTA DE PROJETO

## BIBLIOTECAS UTILIZADAS

- OpenCV: Biblioteca voltada ao desenvolvimento de aplicativos na área de Visão computacional.
- Numpy: Trabalha arrays e matrizes multidimensionais.



# FUNCIONAMENTO



**Exemplo de imagem a ser tratada**

# FUNCIONAMENTO

1. Leitura da imagem
2. Escolha da cor a ser detectada
3. Detecção da cor
4. Geração das máscaras necessária
5. Recoloração
6. Re-inserção na imagem original





# FUNIONAMENTO

- **Leitura da imagem**

Detecção da imagem de entrada independente de seu formato

- **Escolha/Detecção da cor**

No momento o programa trabalha somente com cores primarias, mas pode ser expandido, no exemplo detectamos a cor verde.

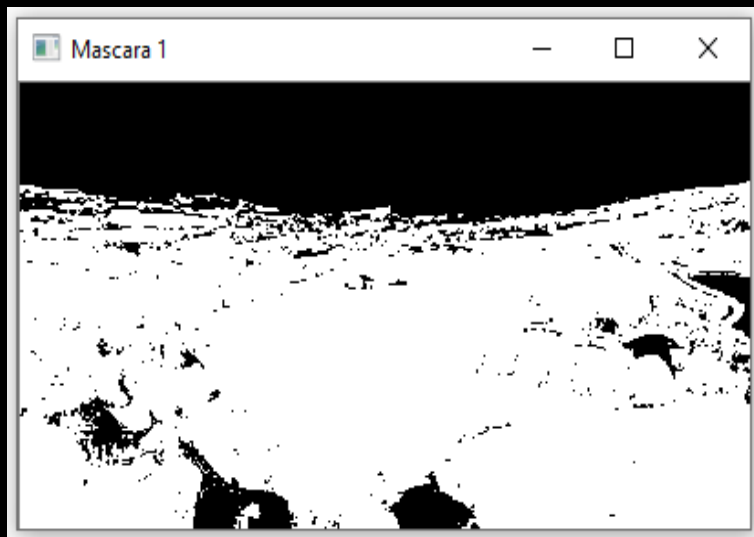
```
minimo_verde = np.array([20,20,20],np.uint8)
maximo_verde = np.array([80,255,255],np.uint8)
Mascara = cv2.inRange(hsv, minimo_verde, maximo_verde)
```

Neste trecho de código acima é mostrado a área de detecção de tons de verde que temos.

# FUNCIONAMENTO

- **Geração das máscaras necessárias**

Para tratamento é gerada duas máscaras a partir da imagem



1 - Máscara da área a ser recolorida

`mascara = cv2.inRange(hsv, minimo_verde, maximo_verde)`



2-Máscara invertida para ser usada na recolocação

`mascara_invertida = cv2.bitwise_not(mascara)`

# FUNCIONAMENTO

- **Recoloração da imagem**

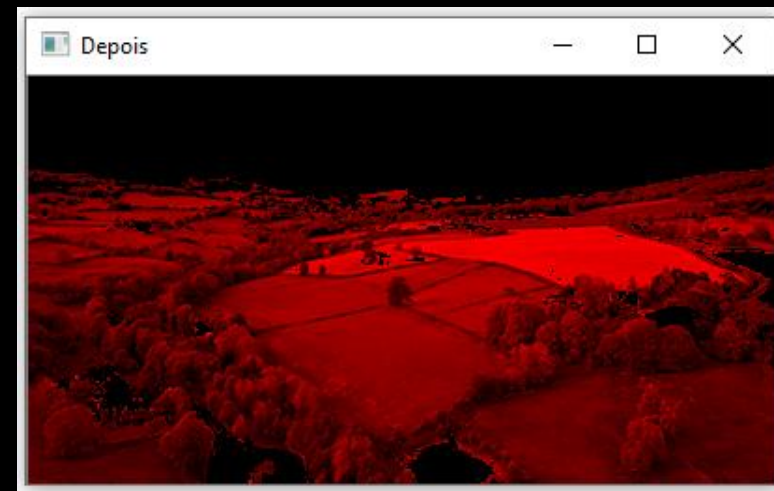
Máscara 1 é aplicada a imagem que em seguida é quebrada em canais RGB

```
resultado= cv2.bitwise_and(imagem,imagem, mask=mascara)
```

```
(canalAzul, canalVerde, canalVermelho) = cv2.split(resultado)
```

Os canais são tratados para aplicar a cor que quisermos (no caso, vermelho) e em seguida aplicados sobre a imagem

```
resultado = cv2.merge([canalAzul*0, canalVerde*0,  
canalVermelho*1])
```

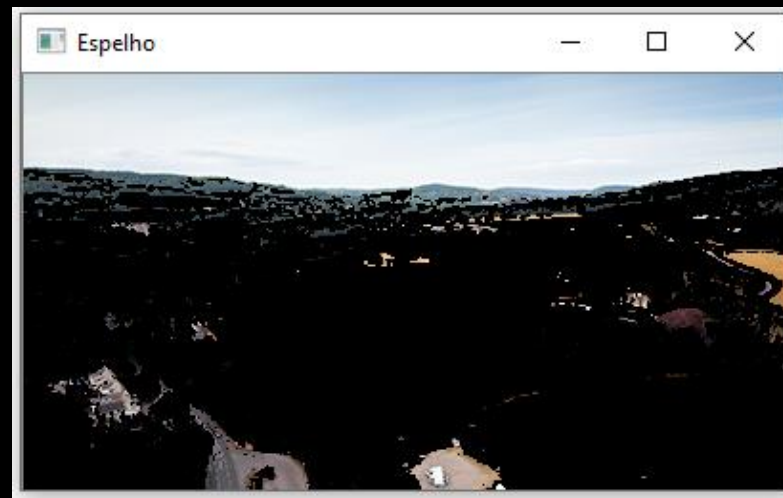


# FUNCIONAMENTO

- **Re-inserção na imagem original**

Após a recoloração precisamos re-inserir a área tratada na imagem original  
A máscara invertida então é aplicada na imagem original

```
espelho= cv2.bitwise_and(imagem,imagem, mask=mascara_invertida)
```

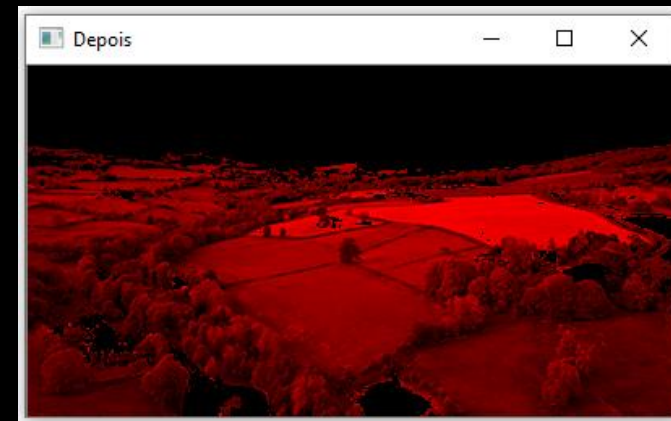
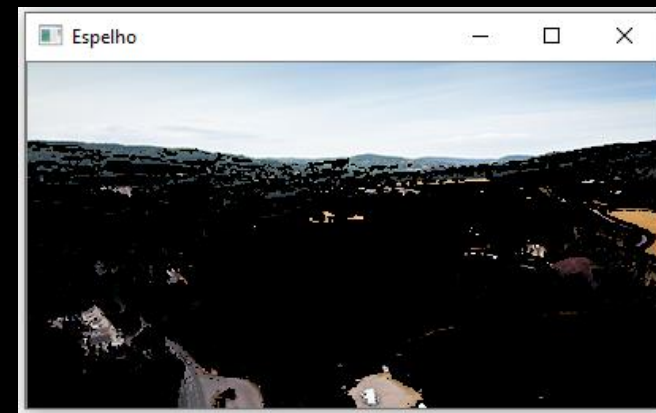
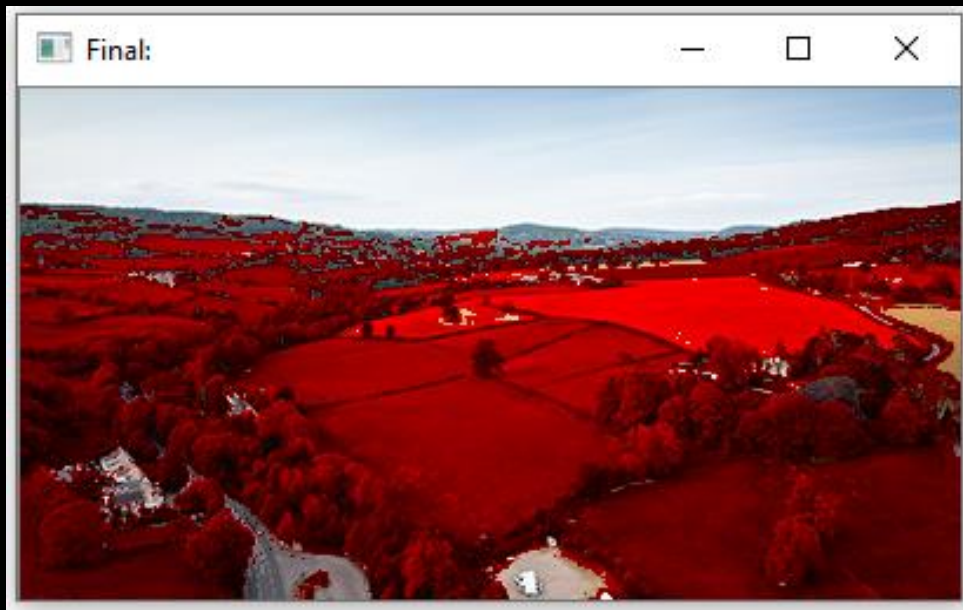


# FUNCIONAMENTO

- **Re-inserção na imagem original**

Então As duas imagens são juntadas novamente produzindo o resultado final.

```
final = cv2.add(resultado,espelho)
```





# APLICAÇÕES DO CÓDIGO

- Inicialmente a intenção de uso do código era para identificação de cores para auxiliar pessoas com algum nível de daltonismo, o auxílio seria feito através da seleção individual e identificação da cor de um objeto e então o display (mostrar) das informações da cor original na tela para o usuário, devido a restrições de tempo não atingimos este objetivo por inteiro porém o produto final pode ser trabalhado para esta intenção, e em seu estado atual também pode ser usado para outras intenções como.

# APLICAÇÕES DO CÓDIGO

- Detecção de áreas de queimadas em florestas
- Detecção de movimentações de nuvens (meteorologia)
- Trabalho de “greenscreen” para tratamento de vídeo
- Recoloração em imagens etc.



Anhanguera

OBRIGADO