

Link para o repositório contendo o código completo:

<https://github.com/GustavoMatoi/Processo-seletivo-Tarken>

Passo 1: Análise da atividade

O principal objetivo dessa atividade, resumidamente, era contar objetos em uma imagem, com diferentes cores (branco e vermelho) e em diferentes situações (no céu ou em direção à água).

Passo 2: Escolha das tecnologias

Essa atividade foi feita em Python, uma vez que possui bibliotecas que facilitam trabalhar com imagens, tais como cv2, numpy e também a cvlib. A atividade foi desenvolvida no Google Collab, tendo em mente que permite o uso de recursos avançados que colaboram com o desenvolvimento de códigos e a falta de necessidade de baixar bibliotecas dentro da máquina local.

Passo 3: Análise dos Dados

Em projetos de análise de dados, é crucial o tratamento e refinamento da base de dados. No caso da atividade, a base de dados é a nossa imagem como um todo, então, separamos o nosso interesse nos seguintes passos:

1. A imagem toda importa?
2. Todas as cores da imagem importam?
3. Necessita de transparência ou filtros?

A partir disso, a base de dados foi refinada seguindo os seguintes passos:

Remoção do terreno

A parte do terreno não é relevante no tratamento do problema, contudo, a imagem foi cortada inferiormente em um editor de fotos para diminuir o tamanho da base de dados.



Imagem original



Recorte do terreno

Remoção do céu e das letras

A cor azul não era necessária na detecção das estrelas e dos meteoros, tal como as letras “Scipio labs”. Para isso, com auxílio de um editor de fotos para identificar o RGB do céu e das letras, foi proposto o seguinte trecho de código:

Link para o repositório contendo o código completo:

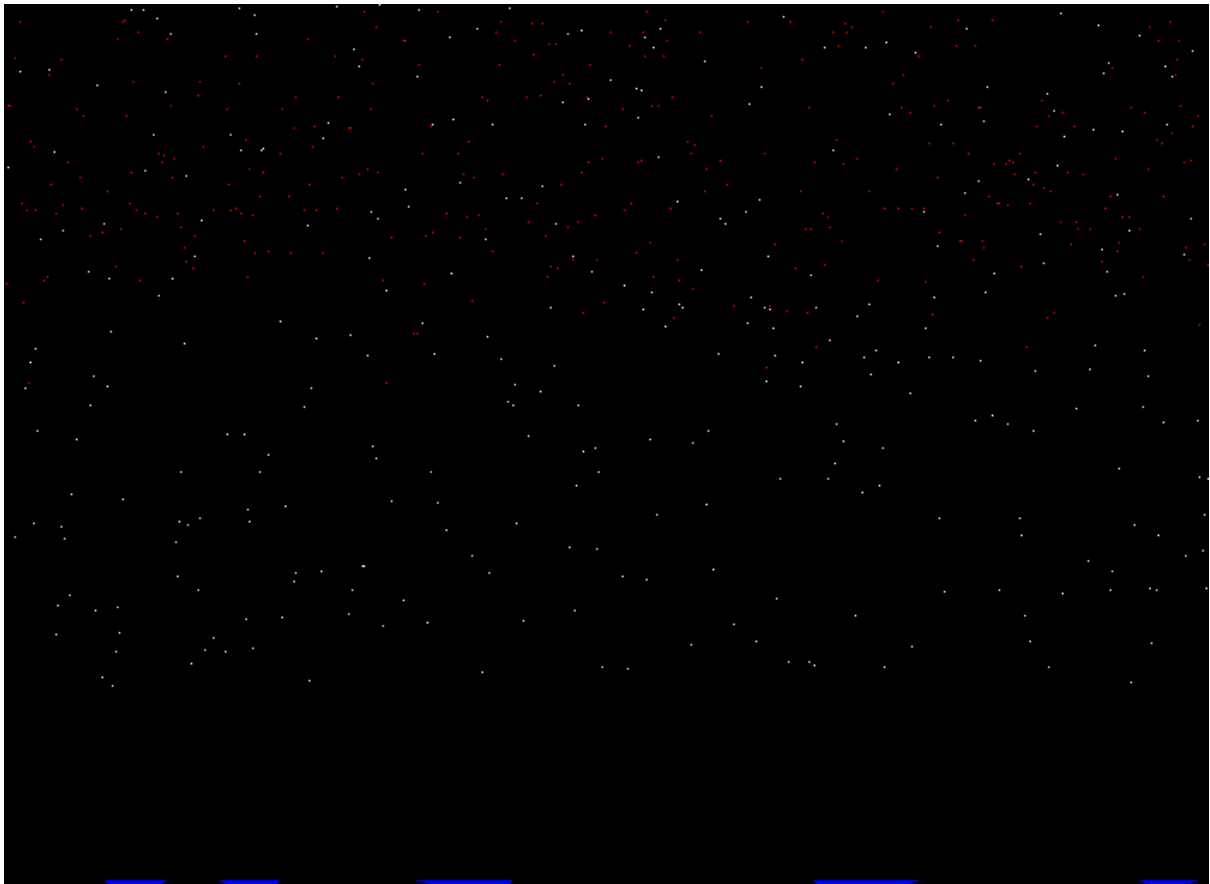
<https://github.com/GustavoMatoi/Processo-seletivo-Tarken>

```
cor_ceu = np.array([2, 119, 189]) #Cor em RGB do céu
cor_letras = np.array([52, 134, 183]) #Cor em RGB da letra
tolerancia = 30

# Criar máscaras para o céu e as letras
mascara_ceu = np.linalg.norm(imagem_rgb - cor_ceu, axis=-1) < tolerancia
mascara_letras = np.linalg.norm(imagem_rgb - cor_letras, axis=-1) < tolerancia

# Trocando os pixels equivalentes ao céu e as letras por preto
imagem_rgb[mascara_ceu] = [0, 0, 0]
imagem_rgb[mascara_letras] = [0, 0, 0]
```

No qual após a identificação das cores, criou-se uma máscara com 30 de tolerância, para a diferença entre as cores da imagem e as cores em RGB passadas por parâmetro. As máscaras foram criadas utilizando funções da biblioteca numpy, a qual dita se os pixels da foto estão próximos da cor delimitada (retornando um true ou false). E no fim, a cor dos pixels azuis das letras e do céu são trocados por preto, resultando na seguinte imagem:



Com a base de dados pré processada, partimos para a programação do projeto.

Passo 4: Programação

A função `contar_objetos` como o nome sugere, precisa de uma imagem e de uma cor para contá-los em uma imagem. A tolerância é implementada tanto superiormente quanto inferiormente pelas variáveis `cor_baixa` e `cor_alta`. Após isso, essas duas variáveis e a imagem são passadas para a função `cv2.inRange` cujo objetivo é criar uma máscara que está entre os intervalos de cores especificado e

Link para o repositório contendo o código completo:

<https://github.com/GustavoMatoi/Processo-seletivo-Tarken>

por fim, a função `num_objetos` é o retorno da função `countNonZero`, que conta quantos objetos estão na imagem a partir da máscara especificada. A função então retorna essa variável `num_objetos`, que será usada adiante.

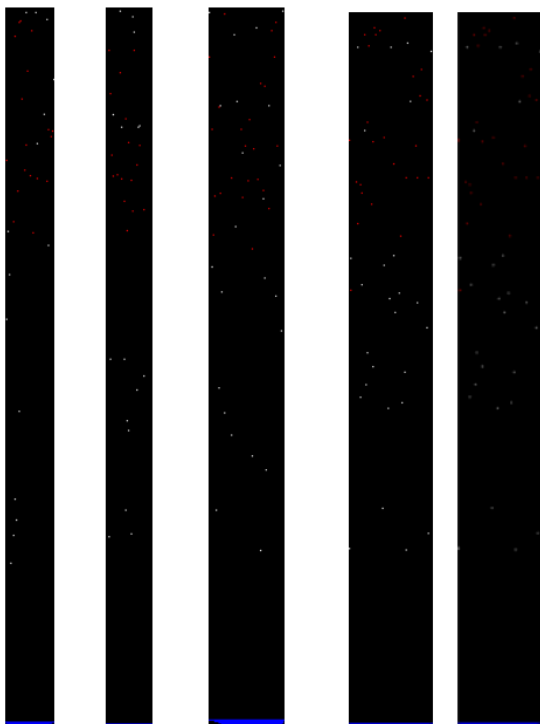
```
def contar_objetos(imagem, cor_rgb):
    cor_baixa = np.array(cor_rgb) - 30
    cor_alta = np.array(cor_rgb) + 30
    mascara = cv2.inRange(imagem, cor_baixa, cor_alta)
    num_objetos = cv2.countNonZero(mascara)
    return num_objetos

meteoros = contar_objetos(imagem_rgb, np.array([255, 255, 255]))
estrelas = contar_objetos(imagem_rgb, np.array([255, 0, 0]))
print(f'Número de meteoros: {meteoros}')
print(f'Número de estrelas: {estrelas}')
```

Os meteoros são contados utilizando essa função, passando a imagem pré-processada juntamente com a cor dos meteoros, que são representados pela cor branco (255, 255, 255 do RGB). Aqui, a função `contar_objetos` irá procurar o número de pixels brancos na imagem pré-processada. O mesmo vale para as estrelas, só que com a cor vermelha. O resultado dos dois `print` abaixo é:

```
Número de meteoros: 315
Número de estrelas: 328
```

A próxima atividade é contar os meteoros que estão caindo em direção ao oceano. Para isso, a imagem foi separada em diferentes imagens, cortando desde o limite superior até as margens de cada parte do oceano.



Link para o repositório contendo o código completo:

<https://github.com/GustavoMatoi/Processo-seletivo-Tarken>

Após isso, as imagens passaram por um laço de repetição transformando as imagens em imagens cv2 e convertendo para o formato RGB para então envia-las para a função contar_objetos, passando a cor dos meteoros (branco) e incrementando na variável contador.

```
contador = 0
for i in range(1,6):
    imagem = cv2.imread(f'/content/imgCortadaMeteoro{i}.png')
    imagem_cortada = cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)
    meteoros = contar_objetos(imagem_cortada, np.array([255, 255, 255]))
    contador += meteoros

print(f'O número de meteoros que cairá na água é igual a: {contador}')
```

O resultado desse print foi:

```
O número de meteoros que cairá na água é igual a: 194
```

Answers:

Number of Stars	328
Number of Meteors	315
Meteors falling on the Water	194
(optional) Hidden Phrase	Não entendi essa questão.

A questão número 4 não consegui entender o que era pra ser feito.