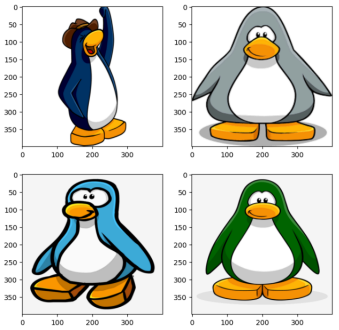
Projeto 1 - Victor Arruda Gorgal 524

1º Juntar todas as imagens em uma única

Primeiramente, convertemos as imagens em arrays, redimensionamos cada uma para 400x400 pixels e as convertemos para o formato RGB. Em seguida, mostramos as imagens separadamente

```
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image

p1_img = np.array(Image.open('P1.png').resize((400,400)))[:, :, :3]
p2_img = np.array(Image.open('P2.png').convert('RGB').resize((400,400)))[:, :, :3]
p3_img = np.array(Image.open('P3.jpg').resize((400,400)))[:, :, :3]
p4_img = np.array(Image.open('P4.png').resize((400,400)))[:, :, :3]

plt.figure(figsize=(8,8))
plt.subplot(2,2,1)
plt.imshow(p1_img)
plt.subplot(2,2,2)
plt.imshow(p2_img)
plt.subplot(2,2,3)
plt.imshow(p3_img)
plt.subplot(2,2,4)
plt.imshow(p4_img)
```



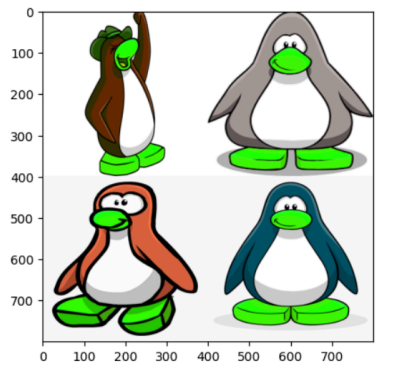
Em seguida fazemos a união de todas as imagens em uma só, através do método *hstack* e *vstack* da biblioteca NumPy e imprimimos a imagem final contendo todos os pinguins.

```
linha_topo = np.hstack((p1_img, p2_img))
linha_baixo = np.hstack((p3_img, p4_img))
final = np.vstack((linha_topo, linha_baixo))


plt.imshow(final)
plt.show()
```

2º Trocar as cores

Para trocar as cores da imagem, converti ela para HSV, troquei o Hue e voltei pra RGB. O processo foi feito através de duas funções: uma que converte uma imagem RGB para HSV (`conversaoHSV()`) e outra que faz o caminho inverso (`conversaoRGB()`). Após converter para HSV, alteramos o matiz (hue) da imagem em 30 graus, gerando uma troca nas cores. Após isso, convertemos a imagem de volta para RGB e imprimimos ela novamente.

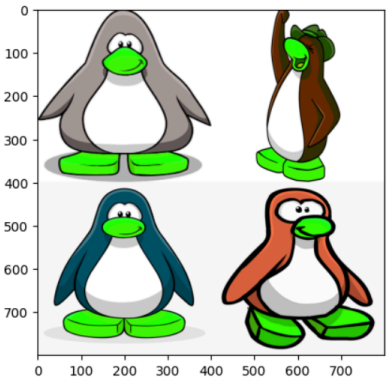
Resultado:



3º Aplicar o espelhamento

Aqui invertemos a coordenada horizontal (coluna j), o que resultou no espelhamento da imagem no eixo vertical.

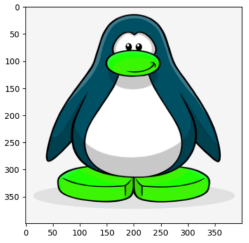
```
(l, c, p) = rgb_modificado.shape

reflexao = np.zeros(shape=rgb_modificado.shape, dtype=np.uint8)
for i in range(l):
    for j in range(c):
        new_x = -j
        new_y = i
        reflexao[new_y, new_x] = rgb_modificado[i, j]

plt.imshow(reflexao)
```



4º Recortar seu pinguim favorito (descarte os outros, deixando apenas um na imagem).

Extraímos o pinguim inferior à esquerda, para isso selecionamos as linhas de 400 a 800 e as colunas de 0 a 400 da imagem e armazenamos o recorte em baixo_esquerdo.

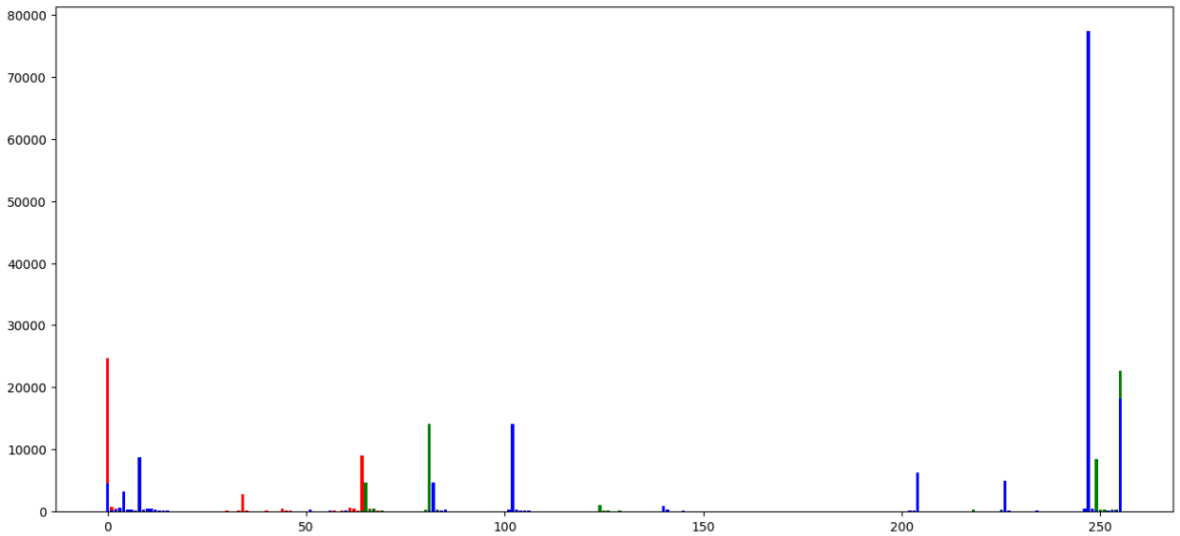
```
baixo_esquerdo = reflexao[400:800, 0:400]
plt.imshow(baixo_esquerdo)
```

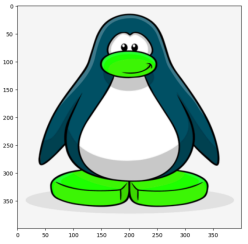


5º Faça a análise do histograma em seu pinguim favorito e escolha um valor de threshold. (Justifique a escolha do threshold)

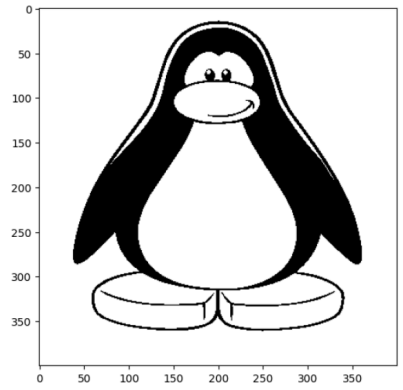
Primeiramente contamos quantos pixels tem cada intensidade de cor (de 0 a 255) e mostramos isso em gráficos de barras. Após imprimimos a imagem e o gráfico.

Resultado:





Após, convertemos a imagem para tons de cinza usando o método de lightness e, em seguida, aplicamos um threshold de 110. Pixels com valor maior ou igual a 110 viram brancos (255), e o restante virá pretos (0). Esse valor foi escolhido devido estar próximo ao meio do gráfico da análise de histograma, com esse valor é possível transformar as características importantes em preto e outras como reflexões branco.



6º Aplique uma conversão para que todos os pixels abaixo dele sejam pintados de sua cor favorita (por exemplo: Azul)

Aqui aplicamos o valor de threshold a imagem e colorimos apenas os pixels abaixo do valor definido, pixels com intensidade abaixo de 110 recebem a cor [68, 14, 232], enquanto os demais permanecem brancos.