

Dados do relatório Lista 0

Matéria: Processamento Digital de Imagens (PDI) **código:** CMP1084 A01

Aluno: Vitor de Almeida Silva

Matrícula: 20161003305497

ENUNCIADO

1. Mostre a imagem escolhida e em seguida os componentes vermelho, verde e azul separadamente da sua imagem escolhida.
2. Troque os componentes R e G da imagem.
3. Primeiro preencha de zero a matriz correspondente ao componente verde e azul e mostre a imagem. Em seguida preencha de zero a matriz correspondente ao componente vermelho e azul e mostre a imagem. E por fim preencha de zero a matriz correspondente ao componente vermelho e verde e mostre a imagem.
4. Conceitue o que é o sistema de cores HSV e Cielab. Em seguida converta sua imagem para o sistema de cores retocitados e para escala de cinza. Utilize as funções *rgb2hsv*, *rgb2lab* e *rgb2gray* respectivamente.
5. Torne a imagem de entrada mais clara ou mais escura.
6. Reduza o tamanho da imagem pela metade. Rotacione a imagem original em 180°. Utilize os comandos *imresize* e *imrotate*.

DESENVOLVIMENTO

1. Mostre a imagem escolhida e em seguida os componentes vermelho, verde e azul separadamente da sua imagem escolhida.

Foram utilizadas as seguintes funções:

- **Imread('caminho da img'):** lê a imagem de algum local especificado do disco;
- **imgShow(img):** mostra a imagem;

Para mostrar os elementos R, G e B separados, deve-se especificar isso utilizando a matriz da imagem utilizando a seguinte forma:

- **img (linha, coluna, (1, 2 ou 3)):** 1, 2 ou 3 se referem aos elementos R, G e B respectivamente;

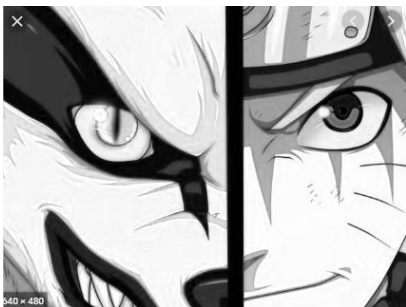
As Figura 1, 2, 3 e 4 mostram a imagem original e seus elementos R, G e B respectivamente.

Figura 1: imagem original



Fonte: Autoral

Figura 2: Componente R



Fonte: Autoral

Figura 3: Componente G



Fonte: Autoral

Figura 4: Componente B



Fonte: Autoral

Código questão 1:

```
%0) lendo e mostrando uma imagem do disco
img=imread('C:\Users\Vitor\Desktop\faculdade\9 nono periodo\PDI\Lista
0\Naruto.jpg');
imshow(img(:,:,:));
figure();

%1) mostrando elemento R,G e B da imagem
imshow(img(:,:,1)); %R
figure();
imshow(img(:,:,2)); %G
figure();
imshow(img(:,:,3)); %B
figure();
```

2. Troque os componentes R e G da imagem.

Para trocar estes elementos basta concatenar os vetores 1 e 3 da imagem utilizando a função `cat`.

Cat(dim, vetor1, vetor2): A função `cat` permite se escolher uma das dimensões para concatenar em linha(1), coluna(2) ou ambas(3);

A Figura 5 mostra o resultado da imagem ao se trocar os componentes R e G.

Figura 5: troca dos componentes R e G



Fonte: Autoral

Código questão 2:

```
%0) lendo e mostrando uma imagem do disco
img=imread('C:\Users\Vitor\Desktop\faculdade\9 nono periodo\PDI\Lista
0\Naruto.jpg');
imshow(img(:,:,,:));
figure();
imgGR= cat(3,img(:,:,2),img(:,:,1),img(:,:,3));
imshow(imgGR);
figure();
```

3. Primeiro preencha de zero a matriz correspondente ao componente verde e azul e mostre a imagem. Em seguida preencha de zero a matriz correspondente ao componente vermelho e azul e mostre a imagem. E por fim preencha de zero a matriz correspondente ao componente vermelho e verde e mostre a imagem.

Para realizar essa operação basta selecionar o elemento correto (R, G ou B) da matriz e atribuir a ele 0.

- 1) Preencha de zero a matriz correspondente ao componente verde e azul e mostre a imagem. A Figura 6 mostra o resultado da operação.

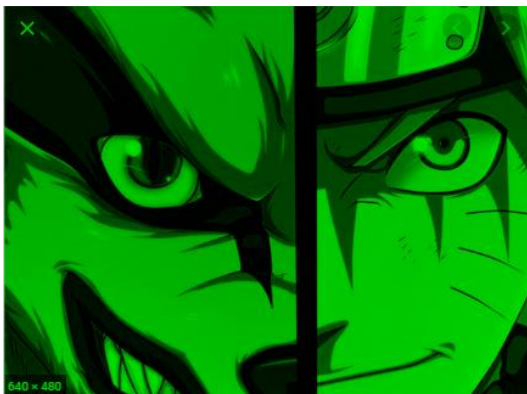
Figura 6: preenchimento dos elementos G e B com 0



Fonte: Autoral

- 2) preencha de zero a matriz correspondente ao componente vermelho e azul e mostre a imagem.

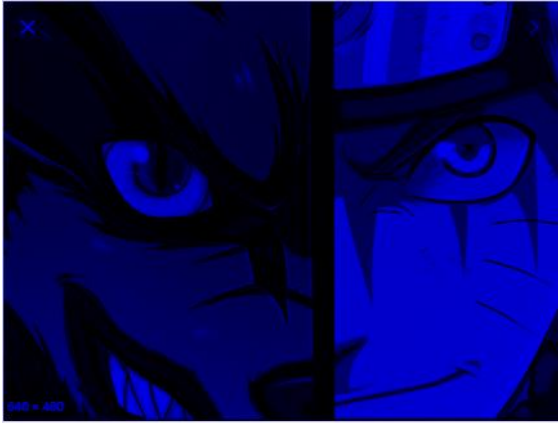
Figura 7: preenchimento dos elementos R e B com 0



Fonte: Autoral

- 3) preencha de zero a matriz correspondente ao componente vermelho e verde e mostre a imagem.

Figura 8: preenchimento dos elementos R e G com 0



Fonte: Autoral

Código questão 3:

```
%0) lendo e mostrando uma imagem do disco
img=imread('C:\Users\Vitor\Desktop\faculdade\9 nono periodo\PDI\Lista
0\Naruto.jpg');
imshow(img(:,:,:));
figure();

%-- preencha de zero a matriz correspondente
%   ao componente verde e azul e mostre a imagem.
imgGB= img;
imgGB(:,:,2)=0;
imgGB(:,:,3)=0;
imshow(imgGB);
figure();

%-- elementos R e B
imgRB= img;
imgRB(:,:,1)=0;
imgRB(:,:,3)=0;
imshow(imgRB);
figure();

%-- elementos R e G
imgRG= img;
imgRG(:,:,1)=0;
imgRG(:,:,2)=0;
imshow(imgRG);
figure();
```

4. Conceitue o que é o sistema de cores HSV e Cielab. Em seguida converta sua imagem para o sistema de cores supracitados e para escala de cinza. Utilize as funções *rgb2hsv*, *rgb2lab* e *rgb2gray* respectivamente.

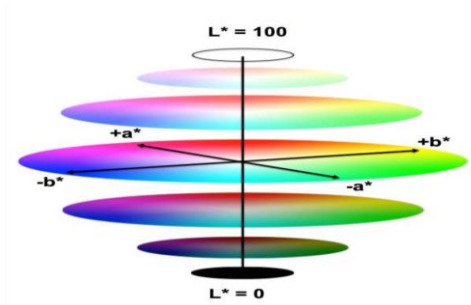
- **HSV (*Hue, Saturation, Value*):** Este espaço de cores, assim como o RGB, também utiliza cores primária. O HSV é próximo de como os humanos percebem as cores. Este, por sua vez, descreve as cores em termos de seu *shade* (tonalidade) (saturação ou quantidade de cinza) e de seu valor de brilho [1]. O HSV é composto por três componentes, que são os seguintes:
 - **Hue (matiz):** Hue é a porção de cor do modelo, expressado com um numero de 0 a 360 graus:
 - **Vermelho:** fica entre 0 e 60 graus;
 - **Amarelo:** fica entre 61 e 120 graus;
 - **Verde:** fica entre 121 e 180 graus;
 - **Ciano:** fica entre 181 e 240 graus;
 - **Azul:** Fica entre 241 e 300 graus;
 - **Magenta:** Fica entre 301 e 360 graus;
 - **Saturação:** Saturação descreve a quantidade de cinza em uma cor particular, de 0 a 100 por cento. Reduzindo esse componente tendendo a zero introduz mais cinza e produz um efeito de desbotado [1]. As vezes a saturação é definida em um intervalo entre 0 e 1, onde 0 é cinza e 1 é alguma cor primária [1].
 - **Valor (brilho):** Valor funciona em conjunto com a saturação e descreve o brilho ou intensidade de cor, de 0 a 100 por cento, onde 0 é completamente preto e 100 é o mais brilhante e revela a maior cor [1].
- **Cielab (*lab color space*):** É um sistema de cores com 3 eixos com dimensão L para brilho, a e b para a dimensão das cores. Esse espaço de cores inclui todas as cores no espectro, como também as cores fora do espectro de percepção humana. Cielab é o meio mais exato de representar cores e dispositivos independentes. Esse espaço de cor é fornece a mais exata representação de cor, porém não é a mais utilizada,

normalmente este tipo é convertido para outro espaço como o RGB [3]. A seguir uma explicação de cada dimensão:

- L^* = brilho (também referido como luminância), a escuridão e claridade de uma cor;
- a^* = vermelho para verde (+a = tons vermelhos, -a=tons verdes);
- b^* = amarelo para azul (+b = tons amarelos, -b = tons azuis);
- Onde os dos eixos de cores interceptados = cinza neutro;

A Figura 9 mostra os eixos de cores do CIELAB.

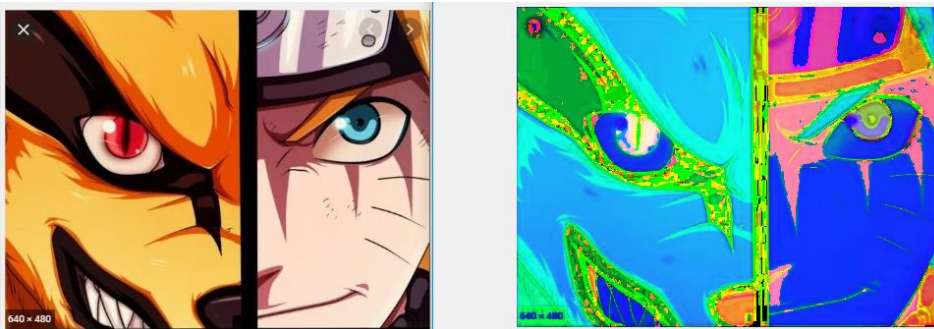
Figura 9: CIELAB espectro



Fonte: (SAPPI, 2013) [2]

Conversão da imagem de RGB para HSV mostrada na Figura 10 (esquerda RGB e direita HSV):

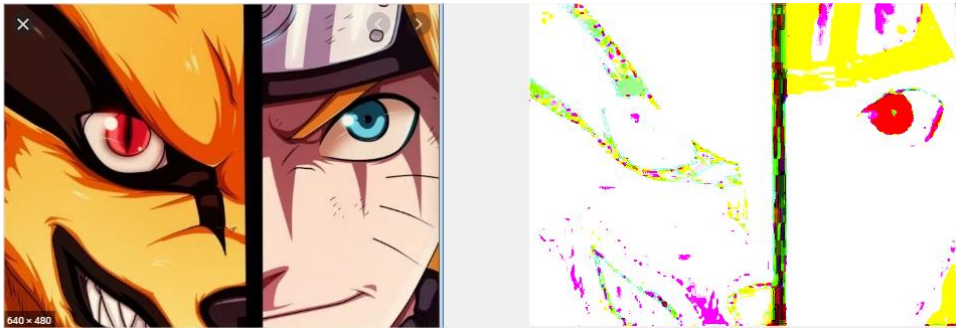
Figura 10: RGB para HSV



Fonte: Autoral

Conversão da imagem de RGB para Cielab mostrada na Figura 11 (esquerda RGB e direita Cielab):

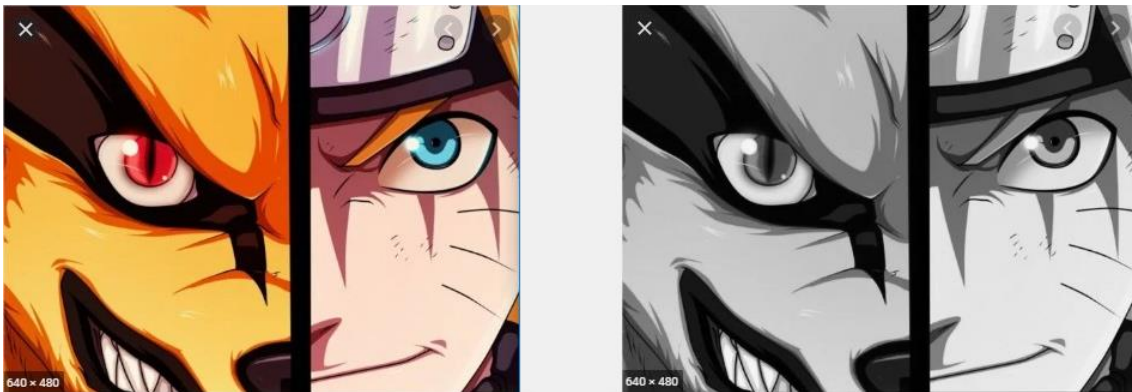
Figura 11: RGB para Cielab



Fonte: Autoral

Conversão da imagem de RGB para escala de cinza mostrada na Figura 13 (esquerda RGB e direita escala de cinza):

Figura 13: RGB para escala de cinza



Fonte: Autoral

Código Questão 4:

```
%0) lendo e mostrando uma imagem do disco
img=imread('C:\Users\Vitor\Desktop\faculdade\9 nono
período\PDI\Lista 0\Naruto.jpg');
imshow(img(:,:,:));
figure();

%RGB para HSV
imgHSV= rgb2hsv(img);
imshow(imgHSV);
figure();

%RGB para Cielab
imgLab= rgb2lab(img);
imshow(imgLab);
figure();

%escala de cinza
imgGray= rgb2gray(img);
imshow(imgGray);
figure();
```

5. Torne a imagem de entrada mais clara ou mais escura.

Aumentar ou reduzir o brilho da imagem é feito através da soma de uma constante em cada pixel da imagem. A Figura 14 mostra a imagem mais original a direita, mais clara ao centro e mais escura a esquerda.

Figura 14: escurecendo e clareando a imagem somando 150 como constante



Fonte: Autoral

Código questão 5:

```
%0) lendo e mostrando uma imagem do disco
img=imread('C:\Users\Vitor\Desktop\faculdade\9 nono
período\PDI\Lista 0\Naruto.jpg');
imshow(img(:,:,,:));
figure();

%imagem mais clara
imgClara=img;
for i=1: 326
    for j=1 : 433
        imgClara(i,j,:)= imgClara(i,j,:) + 150;
    end
end
imshow(imgClara);
figure();

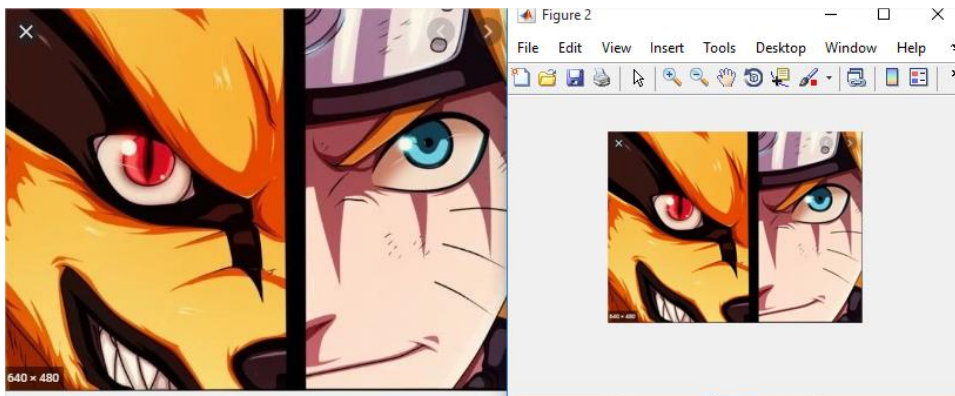
%imagem mais escura
imgEscura = img;
for i=1: 326
    for j=1 : 433
        imgEscura(i,j,:)= imgEscura(i,j,:) - 150;
    end
end
imshow(imgEscura);
figure();
```

6. Reduza o tamanho da imagem pela metade. Rotacione a imagem original em 180°. Utilize os comandos *imresize* e *imrotate*.

- **Imresize(A, scale):** retorna uma imagem com tamanho alterado de acordo com a escala;
- **Imrotate(A, angle):** Rotaciona a imagem na quantidade do ângulo especificado. A imagem é rotacionada por padrão no sentido ant-horário, porém, é possível rotacionar no sentido horário especificando um valor negativo para o ângulo. *Imrotate* usa interpolação dos vizinhos mais próximos, atribuindo 0 aos valores dos pixels de B que estão fora da rotação da imagem.

A Figura 15 mostra a imagem reduzida à metade. A figura original está a esquerda e a redimensionada está à direita.

Figura 15: Imagem reduzida a metade `imresize(img,1/2)`



Fonte: Autoral

Imagem rotacionada em 180 graus mostrada na Figura 16.

Figura 16: Imagem rotacionada em 180 graus



Fonte: Autoral

Código questão 6:

```
%0) lendo e mostrando uma imagem do disco
img=imread('C:\Users\Vitor\Desktop\faculdade\9 nono
período\PDI\Lista 0\Naruto.jpg');
imshow(img(:,:, :));
figure();

%reduzindo a imagem pela metade
imgMetade=imresize(img,1/2);
imshow(imgMetade);
figure();

%rotacionando a imagem em 180 degrees
img180=imrotate(img,180);
imshow(img180);
figure();
```

REFERÊNCIAS

- [1] BEAR, Jacci Howard. **The HSV Color Model in Graphic Design**. 2019. Disponível em: <https://www.lifewire.com/what-is-hsv-in-design-1078068>. Acesso em: 05 mar. 2020.
- [2] SAPPI. **Defining and Communicating Color: The CIELAB System**. North America: Sappi, 2013.
- [3] MATHWORKS. **Representing color with the Lab color space**. 2019. Disponível em: <https://www.mathworks.com/discovery/lab-color.html>. Acesso em: 05 mar. 2020.