RELATÓRIO – OPENCV

MISSÃO RAS

Instituição: IEE Robotics and Automation Society Chapter

Aluno: Leonardo Vieira Albuquerque

Nesta missão foi possível aprender alguns conhecimentos essenciais para o entendimento da visão computacional e suas implicações.

Foram executados 6 exemplos dos capítulos dos 1 e 2

Exemplo – Capítulo 1

Inicialmente foi importada a biblioteca cv2 que oferece as funcionalidades da biblioteca OpenCV e, assim, foi viável utilizar imagens e vídeos no código. Em seguida, foi-se atribuída a variável imagem a uma função que lê a o arquivo.jpg, a função em si é expressa por: cv2.imread('entrada.jpg'). Após isso, o código abaixo foi adicionado.

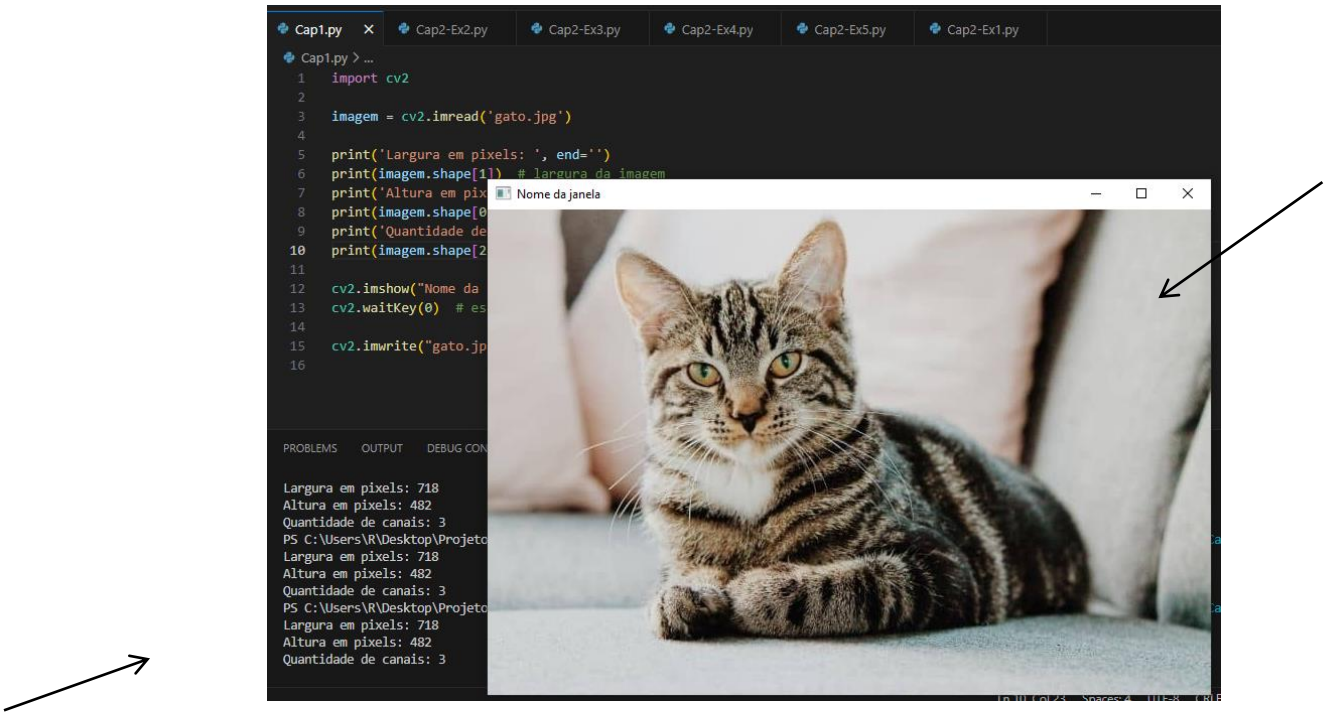
```
print('Largura em pixels: ', end='')
print(imagem.shape[1]) #largura da imagem
print('Altura em pixels: ', end='')
print(imagem.shape[0]) #altura da imagem
print('Qtde de canais: ', end='')
print(imagem.shape[2])
```

Lembre-se que quando usamos a função.imread(arquivo.jpg), a imagem é armazenaza como uma matriz. Com isso, podemos comentar sobre a função.shape que foi utilizado no código, essa função irá ser possível encontrar 3 valores, o primeiro desses valores irá representar a altura (número de linhas da imagem), o segundo irá representar a largura (número de colunas) e o terceiro irá representar o número de canais da imagem.

Continuando, foi utilizada a função imshow("String com nome da janela da imagem, imagem(jpg)"), está função apresenta a imagem lida anteriormente. Também foi utilizada a função waitKey(0) que faz com que o código só prossiga se alguma tecla for clicada.

Por fim, foi utilizada a função.imwrite("arquivo.jpg", imagem), essa função é utilizada para salvar a imagem.

O resutaldo deste código está apresentado abaixo:



Exemplo 1 – Capítulo 2

Neste exemplo, a biblioteca cv2 e a função cv2.imread foram utilizadas. Esses conceitos já foram discutidos no Exemplo – Capítulo 1. Dessa forma, eu irei comentar acerca das condições implementadas neste Exemplo 1 (capítulo 2) em específico.

Abaixo está o código:

```python
Cap2-Ex1.py > ...
import cv2
imagem = cv2.imread('gato.jpg')
(b, g, r) = imagem[0, 0]  # veja que a ordem BGR e não RGB

print('O pixel (0, 0) tem as seguintes cores: ')
print('Vermelho: ', r, 'Verde: ', g, 'Azul: ', b)
```

É possível perceber o código encontra os valores dos canais vermelho, verde e azul quando o ponto na imagem é x = 0 e y = 0.

O resultado desse código está representado abaixo:

```
O pixel (0, 0) tem as seguintes cores:
Vermelho:  205 Verde:  200 Azul:  197
PS C:\Users\R\Desktop\Projeto RAS>
```

Exemplo 2 – Capítulo 2

Nesse exemplo, a biblioteca cv2 e algumas funções como cv2.imread, cv2.imshow, imagem.shape[] e cv2.waitKey foram utilizadas. Esses conceitos já foram discutidos no Exemplo – Capítulo 1. Dessa forma, eu irei comentar acerca das condições implementadas neste Exemplo 2 (capítulo 2) em específico.

Abaixo está o código:

```
1   import cv2
2
3   imagem = cv2.imread('gato.jpg')
4   for y in range(0, imagem.shape[0]):
5       for x in range(0, imagem.shape[1]):
6
7           imagem[y, x] = (255, 0, 0)
8
9   cv2.imshow("Imagem modificada", imagem)
10
11  cv2.waitKey(0)
12
```

É possível observar que está sendo formado um loop, o primeiro cria um loop que percorre as coordenadas (0, altura da imagem) e o segundo loop percorre as coordenadas (largura da imagem, 0). Além disso, vale destacar que dentro dos loops, na linha 7, é possível perceber que para cada valor percorrido de x e y está sendo transformado na cor (255, 0 , 0). Dessa forma, cabe mencionar que esta transformação está diretamente relacionada ao BGR que é um modelo de cor que representa uma combinação das cores vermelho, verde e azul. Com isso, quando a imagem nas coordenadas x e y é transformada na cor azul pura, representada por (255, 0, 0).

O resultado desse código está representado abaixo:

Exemplo 3 – Capítulo 2

Nesse exemplo, a biblioteca cv2 e algumas funções como cv2.imread, cv2.imshow, imagem.shape[] e cv2.waitKey foram utilizadas. Esses conceitos já foram discutidos no

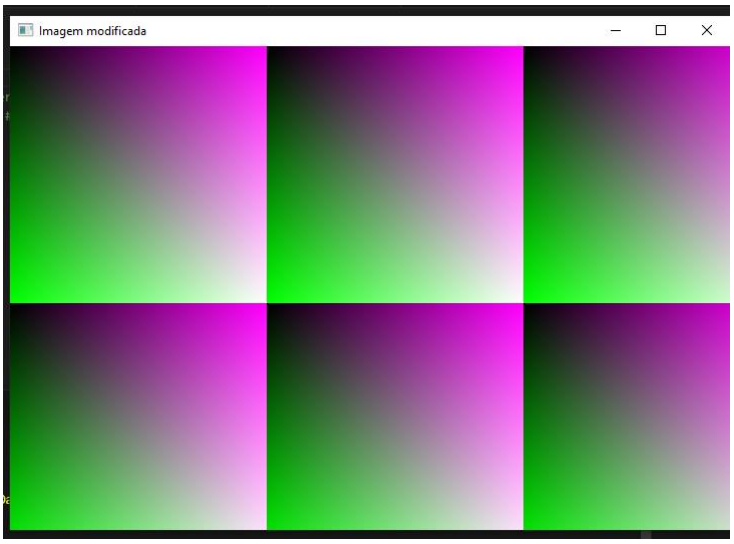
Exemplo – Capítulo 1. Dessa forma, eu irei comentar acerca das condições implementadas neste Exemplo 3 (capítulo 2) em específico.

Abaixo está o código:

```python
import cv2

imagem = cv2.imread('gato.jpg')
for y in range(0, imagem.shape[0]):  # percorre as linhas
    for x in range(0, imagem.shape[1]):  # percorre as colunas

        imagem[y, x] = (x % 256, y % 256, x % 256)

cv2.imshow("Imagem modificada", imagem)
cv2.waitKey(0)
```

É possível perceber a formação de loops já explicados anteriormente, no entanto, na linha 7, ao invés de definir os valores dos canais de cor quando estiver passando por um ponto específico [x, y], os valores desses canais serão o resto da divisão do valor de x/256 e y/256.

O resultado desse código está representado abaixo:



Exemplo 4 – Capítulo 2

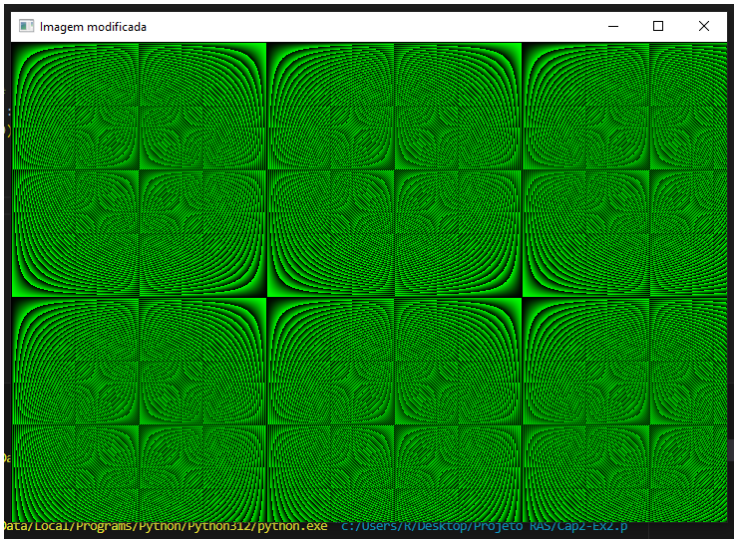
Nesse exemplo, a biblioteca cv2 e algumas funções como cv2.imread, cv2.imshow, imagem.shape[] e cv2.waitKey foram utilizadas. Esses conceitos já foram discutidos no Exemplo – Capítulo 1. Dessa forma, eu irei comentar acerca das condições implementadas neste Exemplo 4 (capítulo 2) em específico.

Abaixo está o código:

```python
import cv2
imagem = cv2.imread('gato.jpg')

for y in range(0, imagem.shape[0], 1):  # percorre as linhas
    for x in range(0, imagem.shape[1], 1):  # percorre as colunas
        imagem[y, x] = (0, (x*y) % 256, 0)

cv2.imshow("Imagem modificada", imagem)
cv2.waitKey(0)
```

É possível perceber a formação de loops já explicados anteriormente, no entanto, na linha 6, os valores dos canais azul e vermelhos estão definidos (não haverá espectro azul nem vermelho) e o canal verde o resto será a multiplicação dos valores de x e y dividido por 256. Note também que o valor 1 no final da linha 4 representa que será a coluna será percorrida uma por uma, esta mesma lógica serve para a coordenada x.

O resultado desse código está representado abaixo:



Exemplo 5 – Capítulo 2

Nesse exemplo, a biblioteca cv2 e algumas funções como cv2.imread, cv2.imshow, imagem.shape[] e cv2.waitKey foram utilizadas. Esses conceitos já foram discutidos no Exemplo – Capítulo 1. Dessa forma, eu irei comentar acerca das condições implementadas neste Exemplo 5 (capítulo 2) em específico.

Abaixo está o código:

Cap2-Ex5.py > ...

```python
import cv2
imagem = cv2.imread('gato.jpg')
for y in range(0, imagem.shape[0], 10):  # percorre as linhas
    for x in range(0, imagem.shape[1], 10):  # percorre colunas
        imagem[y:y+5, x:x+5] = (0, 255, 255)

cv2.imshow("Imagem modificada", imagem)
cv2.waitKey(0)
```

É possível perceber a formação de loops já explicados anteriormente, no entanto, na linha 5, foi definidor que não haverá canal azul e haverá canal de cor verde puro e vermelho puro, formando a cor amarela. Vale ressaltar que o código percorrerá de 10 em 10 pixels, tanto para a coordenada x quanto para coordenada y e, além disso, para cada coordenada [x, y] será definido um bloco de 5 de tamanho na coordenada x e um bloco de 5 de tamanho na coordenada y.

O resultado desse código está representado abaixo: