

ESTI019-17 - Codificação de Sinais Multimídia

Roteiro 03 - Codificação por Transformada

Prof. Celso Kurashima

Prof. Mario Minami

18 de outubro de 2019

Identificação do Aluno

Nome Completo

ESCREVA AQUI A SUA RESPOSTA

RA

ESCREVA AQUI A SUA RESPOSTA

Atividades em Aula

1. No seu diretório "**Documentos**", crie uma pasta com seu primeiro nome e uma sub-pasta "**lab3**".
2. Dentro desta pasta, copie todos arquivos fornecidos, incluindo os programas e as imagens,
3. Entre no **jupyter notebook** e abra o arquivo de programa fornecido em python:
lab3_codif_image_v1.ipynb.
4. Siga as instruções a seguir.

1: Separar os Canais nos Espaços de Cores

- O código abaixo realiza a leitura de um arquivo de **imagem** e mostra nos espaços RGB, YCBCR, e HSV. Estude passo a passo os comandos.

****Considerações Importantes sobre Conversão entre espaços de cores****

- Objetivo: entender os canais de uma imagem nos espaços de cores: **RGB**, **YCrCb**, **HSV**.
- As imagens coloridas são lidas em **BGR** (Azul-Verde-Vermelho), devido ao padrão do OpenCV.
- A função **imshow()** do OpenCV sempre assume que a Matriz mostrada está no espaço de cores BGR.
- O Matplotlib exibe no modo **RGB**. Portanto, as imagens não serão mostradas corretamente pelo Matplotlib se a imagem foi previamente lida pelo OpenCV. Procure no programa abaixo onde a mudança de BGR para RGB é realizada.

In []:

```
# Estude este código e Execute
# (clique em "Run", com esta célula selecionada)
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

bgr1 = cv.imread('messi5.jpg') # ATENÇÃO, está no formato BGR
#bgr1 = cv.imread('lena.bmp')
#bgr1 = cv.imread('peppers.png')

altura,largura,camadas= bgr1.shape
print("Resolução: ", largura, " x ", altura, " PIXELS. ", camadas, " camadas.")

# separa os canais e re-arranja para formar imagem RGB
b1,g1,r1 = cv.split(bgr1)
rgb2 = cv.merge([r1,g1,b1])

# Imprime cores distorcidas (BGR) cores reais (RGB) Lado a Lado
plt.figure(figsize=(9.7, 3.6))
plt.subplot(121);plt.imshow(bgr1); plt.title('BGR'); plt.xticks([], plt.yticks([]))
plt.subplot(122);plt.imshow(rgb2); plt.title('RGB'); plt.xticks([], plt.yticks([]))
plt.show()

# Converte para o formato YCrCb e o formato HSV
ycrcb = cv.cvtColor(bgr1, cv.COLOR_BGR2YCrCb)
hsv = cv.cvtColor(bgr1, cv.COLOR_BGR2HSV)

# Imprime o formatos YCrCb e HSV ao Lado das cores reais (RGB)
plt.figure(figsize=(15, 6))
plt.subplot(131);plt.imshow(rgb2); plt.title('RGB'); plt.xticks([], plt.yticks([]))
plt.subplot(132);plt.imshow(ycrcb); plt.title('YCrCb'); plt.xticks([], plt.yticks([]))
plt.subplot(133);plt.imshow(hsv); plt.title('HSV'); plt.xticks([], plt.yticks([]))
plt.show()
```

In []:

```
# Separação dos Canais RGB individualmente
# Estude este código e Execute

imageR = rgb2.copy()
imageR[:, :, 1:3] = 0

imageG = rgb2.copy()
imageG[:, :, [0, 2]] = 0

imageB = rgb2.copy()
imageB[:, :, 0:2] = 0

plt.figure(figsize=(15, 6))
plt.subplot(131);plt.imshow(imageR); plt.title('RGB_R'); plt.xticks([], plt.yticks([]))
plt.subplot(132);plt.imshow(imageG); plt.title('RGB_G'); plt.xticks([], plt.yticks([]))
plt.subplot(133);plt.imshow(imageB); plt.title('RGB_B'); plt.xticks([], plt.yticks([]))
plt.show()
```

In []:

```
# Separação dos Canais YCbCr individualmente
# Estude este código e Execute

y1,cr1,cb1 = cv.split(ycrcb)

imageCR = ycrcb.copy()
y2 = imageCR[:, :, 0]
imageCR[:, :, 0] = 0
imageCR[:, :, 2] = 0
Cr = cv.cvtColor(imageCR, cv.COLOR_YCrCb2RGB)

imageCB = ycrcb.copy()
imageCB[:, :, 0] = 0
imageCB[:, :, 1] = 0
Cb = cv.cvtColor(imageCB, cv.COLOR_YCrCb2RGB)

plt.figure(figsize=(15, 5))
plt.subplot(141);plt.imshow(rgb2); plt.title('RGB Original'); plt.xticks([], plt.yticks([]))
plt.subplot(142);plt.imshow(y1, cmap='gray'); plt.title('YCrCb_Y'); plt.xticks([], plt.yticks([]))
plt.subplot(143);plt.imshow(Cr); plt.title('YCrCb_CR'); plt.xticks([], plt.yticks([]))
plt.subplot(144);plt.imshow(Cb); plt.title('YCrCb_CB'); plt.xticks([], plt.yticks([]))
plt.show()
```

****Atividade Requerida:****

- Neste mesmo programa, teste também as imagens peppers e lena, trocando o comentário no programa.
- Observe a distorção visual quando as imagens YCrCb e HSV são plotadas, pois o comando imshow não faz re-conversão para RGB.

2. Compressão de Imagens com perdas:

- O formato JPEG permite compressão da imagem ao salva-la em arquivo com o comando **imwrite()**.
- A compressão afeta a qualidade da imagem. O parametro que a controla é **IMWRITE_JPEG_QUALITY**, na faixa de 0 - 100 (quanto maior, melhor qualidade). Default é 95.
- Estude o código abaixo, e execute.

In []:

```
# Estude bem este código e Execute
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

bgr = cv.imread('lena.bmp') # formato BGR
#bgr = cv.imread('peppers.png') # formato BGR

# Salva o arquivo com menor qualidade, neste caso fator 25 e fator 5
cv.imwrite('jpeg25.jpg', bgr, [cv.IMWRITE_JPEG_QUALITY, 25])
cv.imwrite('jpeg05.jpg', bgr, [cv.IMWRITE_JPEG_QUALITY, 5])

# Leitura dos arquivos salvos
bgr25 = cv.imread('jpeg25.jpg')
bgr05 = cv.imread('jpeg05.jpg')

# Apresentação do resultado na tela
cv.imshow('RGB_ori', bgr)
cv.imshow('JPEG_25', bgr25)
cv.imshow('JPEG_05', bgr05)

k = cv.waitKey(0)

if k == 27:
    cv.destroyAllWindows()

# Converte o resultado para RGB para plotar
rgb1 = cv.cvtColor(bgr, cv.COLOR_BGR2RGB)
rgb25 = cv.cvtColor(bgr25, cv.COLOR_BGR2RGB)
rgb05 = cv.cvtColor(bgr05, cv.COLOR_BGR2RGB)

plt.figure(figsize=(15, 6))
plt.subplot(131);plt.imshow(rgb1); plt.title('RGB_ori'); plt.xticks([], plt.yticks([]))
plt.subplot(132);plt.imshow(rgb25); plt.title('jpeg25'); plt.xticks([], plt.yticks([]))
plt.subplot(133);plt.imshow(rgb05); plt.title('jpeg05'); plt.xticks([], plt.yticks([]))
plt.show()
```

****Atividade Requerida:****

- ANOTE os tamanhos dos arquivos, calcule o **FATOR DE COMPRESSÃO**, e analise sua correlação com a **qualidade Visual** da imagem resultante.
 - (OBS.: Para o tamanho correto de arquivos no diretório do windows, selecione o arquivo com o botão direito do mouse e clique em "propriedades".)
- ...
- Repita a execução com outras imagens fornecidas, e também para as imagens com as fotos de todos integrantes da equipe, modificando o próprio programa acima, para leituras dos arquivos utilizados. E realize a mesma análise do fator de compressão e a qualidade visual
- Crie uma tabela para inserir todos dados deste experimento.

3. Cálculo da qualidade objetiva da compressão de imagem:

- Esse código abaixo é uma continuidade do programa anterior, pois utiliza as variáveis executadas pela última vez.
- Objetivo é utilizar o método **PSNR** para mensurar a qualidade da imagem reconstruída após a compressão.
- Anote o valor obtido na tabela.

In []:

```
# esta é continuação da célula anterior

psnr1 = cv.PSNR(bgr, bgr)
psnr25 = cv.PSNR(bgr, bgr25)
psnr05 = cv.PSNR(bgr, bgr05)

print("psnr_ori= %.2f (dB)" % psnr1)
print("psnr 25 = %.2f (dB)" % psnr25)
print("psnr 05 = %.2f (dB)" % psnr05)
```

Função PSNR() Definição no OpenCV:

https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html#ga3119e3ea73010a6f810bb05aa36ac8d6
(https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html#ga3119e3ea73010a6f810bb05aa36ac8d6)

- Para verificar o quão imperceptível foi a operação de compressão de imagem, utiliza-se algum sistema para verificar a similaridade ou as diferenças. O algoritmo mais comum usado para isso é o PSNR (também conhecido como relação sinal / ruído de pico). A definição mais simples começa com a média erro de esquadro. Seja duas imagens: **I1** e **I2**; com um tamanho bidimensional *iej*, composto por número *c* de canais.

$$MSE = \frac{1}{c * i * j} \sum (I_1 - I_2)^2$$

Portanto, o valor de **PSNR** é dado por:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Aqui **MAX 1** é o valor máximo válido para um pixel. No caso da imagem simples de byte único por pixel por canal, é 255. Quando duas imagens são as mesmas, o MSE dará zero, resultando em uma operação inválida de divisão por zero na fórmula PSNR. Nesse caso, o PSNR é indefinido, e devemos lidar com este caso separadamente. A transição para uma escala logarítmica é feita porque os valores de pixel têm uma faixa dinâmica muito ampla.

****Atividade Requerida:****

- Elabore um programa que calcula o valor de PSNR, para todas imagens utilizadas no item anterior.
- ANOTE os valores obtidos na sua tabela do fator de compressão e qualidade visual.

4. Obtenção da DCT num bloco 8x8 para componente Y de uma imagem:

- Esse código apresenta um exemplo simples para o cálculo da Transformada Discreta de Cosseno (DCT), num bloco de 8x8 pixels de uma imagem.
- Objetivo deste experimento é comprovar a característica da DCT de permitir eliminação de redundância da informação nas imagens.

In []:

```
# Estude bem este código e Execute
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('messi5.jpg')
alt,larg,cam= img.shape

ycrCb = cv.cvtColor(img, cv.COLOR_BGR2YCrCb)
y, cr,Cb = cv.split(ycrCb)

ball = y[280:340, 330:390] #
h, w = ball.shape

cx = round(w/2)
cy = round(h/2)

bloco8x8 = ball[cx-4:cx+4,cy-4:cy+4]

print("(1)")
print("Matriz 8x8: componente Y original")
print(bloco8x8)

bloco8x8f = np.float32(bloco8x8)/255.0 # conversão formato float
dct8x8f = cv.dct(bloco8x8f) # cálculo da DCT
dct8x8 = np.int64( (dct8x8f*255.0) ) # conversão formato inteiro

print("(2)")
print("Imagem Y 8x8 (formato ponto flutuante)")
print( np.around(bloco8x8f, decimals = 2) )

print("(3)")
print("DCT de Y (ponto flutuante)")
print( np.around(dct8x8f, decimals = 2) )

print("(4)")
print("DCT de Y (formato inteiro)")
print(dct8x8)
```

****Atividade Requerida:****

- Responda às questões.
 - Na matriz DCT obtida neste exemplo onde se localizam os coeficientes com os maiores valor absoluto? E os coeficientes com os menores valor absoluto? Complemente a resposta, analisando em relação às frequências espaciais.
 - Numa amostragem **zig-zag** do JPEG, quais seriam os primeiros cinco termos?

(-- Obs. não é necessário fazer programa para este item.)

****Relatório da Atividade****

Elaborar o relatório das atividades práticas, no formato HTML5:

- (a) Descreva todos os procedimentos realizados.
- (b) Disponibilize os programas elaborados na forma de arquivo **.ipynb**, e coloque o "link" para o mesmo no relatório html5.
- (c) Insira as imagens utilizadas, na forma multimídia do relatório html5.
- (d) Insira a explicação falada por voz.
- (e) Envie os arquivos na pasta disponibilizada.

ESCREVA AQUI A SEU PROGRAMA

In []:

```
### INSIRA AQUI O CÓDIGO ###
```

Referências

MINICHINO, J., HOWSE, J.. Learning OpenCV 3 Computer Vision with Python, 2nd Edition, Packt Publishing, 2015.

Tutorial: OpenCV-Python Tutorials https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html
(https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html)

Tutorial: Getting Started with Images https://docs.opencv.org/3.4.1/dc/d2e/tutorial_py_image_display.html
(https://docs.opencv.org/3.4.1/dc/d2e/tutorial_py_image_display.html)

Color conversions: https://docs.opencv.org/4.1.2/de/d25/imgproc_color_conversions.html
(https://docs.opencv.org/4.1.2/de/d25/imgproc_color_conversions.html)

YCbCr: <https://en.wikipedia.org/wiki/YCbCr> (<https://en.wikipedia.org/wiki/YCbCr>)

Função PSNR():

https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html#ga3119e3ea73010a6f810bb05aa36ac8d6
(https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html#ga3119e3ea73010a6f810bb05aa36ac8d6)

DCT e outras operações:

https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html#ga85aad4d668c01fbd64825f589e3696d4
(https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html#ga85aad4d668c01fbd64825f589e3696d4)

- Insira novas referencia, caso tenha utilizado.