Aplicação do Chroma Subsampling

Diogo do Nascimento Paza[1]

[1]Universidade Estadual de Maringa(UEM)
Processamento Digital de Imagens
Professor: Franklin César Flores

diogopazacvel@gmail.com

Chroma Subsampling

É um modelo usado na compressão de imagens e vídeo. Dentro de uma imagem o que mais interessa para a percepção humana é a luminância, e as cores não influenciam tanto a visão quanto a luminância.

O Chroma Subsampling é um esquema de compressão que mantém dentro de uma imagem os componentes de luminância com maior definição e os componentes de cores com menor definição.

Sistema YCbCr

O modelo YCbCr é largamente utilizado em vídeos digitais. A informação de luminância é representada por Y. A informação de cor é representada por Cb e Cr.

Y: componente de iluminação

Cb: componente de diferença-azul

Cr: componente de diferença-vermelho

YCbCr é usado para separar um sinal de luma (Y)que pode ser armazenado com alta resolução ou transmitido em alta largura de banda, e dois componentes de crominância (Cb e Cr)que podem ser reduzidos, subamostrados, compactados.

O Chroma Subsampling é um exemplo, onde se diminui a resolução de cor em comparação com o "preto e branco", já que os humanos são mais sensíveis à informação de luminância, para isto o o exemplo prático no final do artigo faz a conversão de uma imagem no sistema de cor RGB para o sistema YCbCr.

Constituição do Chroma Subsampling

Exemplo:

4:4:4 4= o primeiro 4 refere-se a 4 informações de luminância para 4 do vermelho(chamado de Cr) e o último 4 da crominância do azul(chamado de Cb)

No esquema 4:4:4 não há nenhuma degradação do esquema de cores no Chroma Subsampling.

O principal objetivo do Chroma Subsampling é diminuir a quantidade de informações a serem transmitidas e também informações a serem armazendas, resultando em arquivos menores.

Exemplo de compressões que usam Chroma Subsampling são as compressões JPG e a compressão H264.

OpenCV

OpenCV é a principal biblioteca de código aberto para a visão computacional, processamento de imagem e aprendizagem de máquina, e agora apresenta a aceleração de GPU para operação em tempo real.

OpenCV foi desenvolvida pela Intel e possui mais de 500 funções.

Aplicação prática

Demonstração de aplicação usando Python 3.6.5.

A figura 1 abaixo, mostra um arquivo de imagem gerado que usa o esquema 4:1:1 do Chroma Subsamplig no canal Cr. Gerando grande perca de qualidade na imagem de saída em comparação com a imagem de entrada.

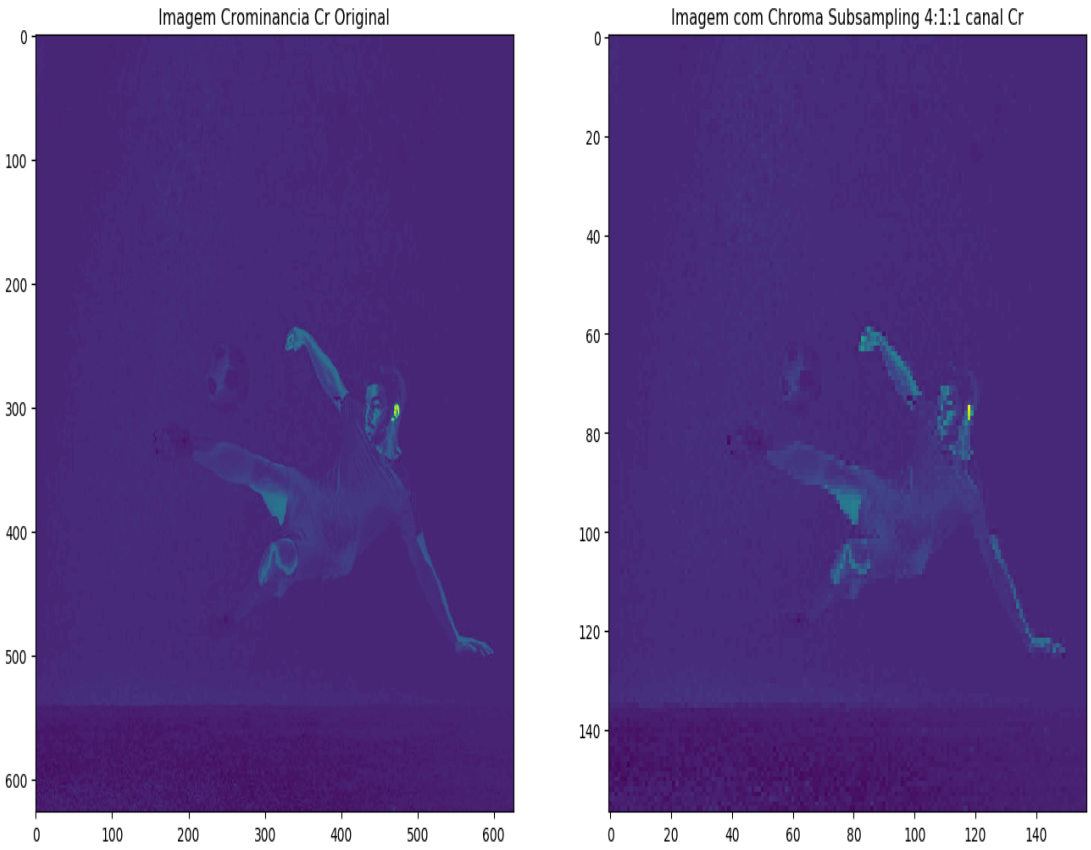

Figura 1

A figura 2 abaixo, mostra um arquivo de imagem gerado que usa o esquema 4:2:2 do Chroma Subsamplig no canal Cr. Onde a perda de qualidade já não é tão perceptível.

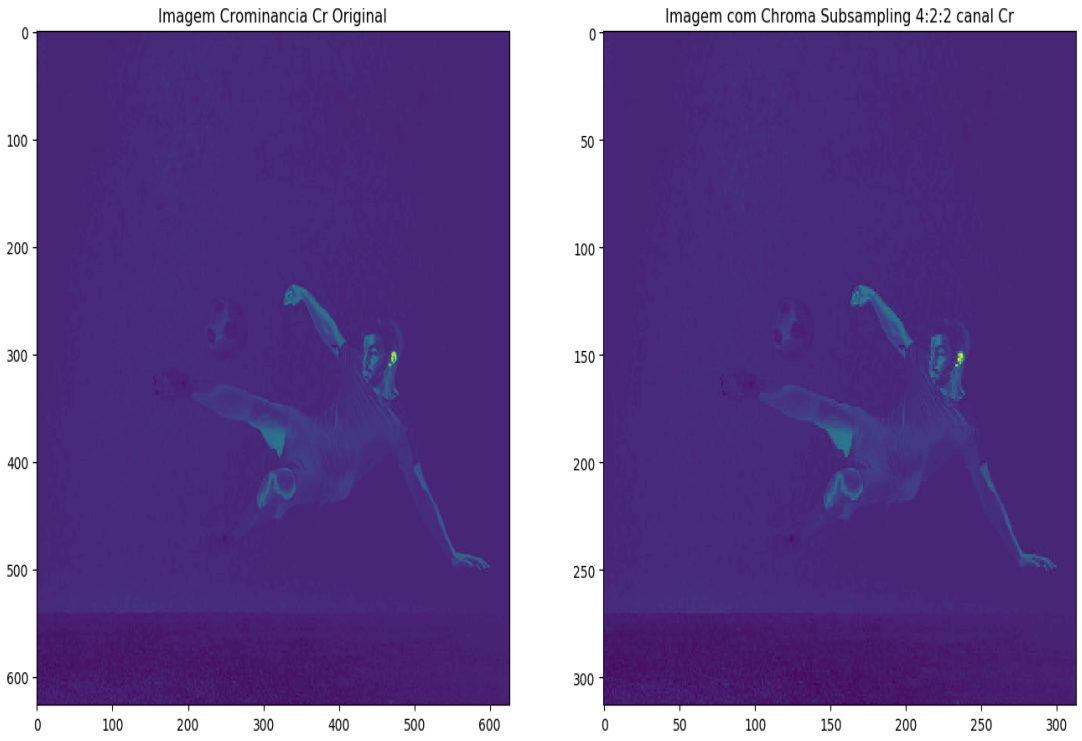

Figura 2

A figura 3 abaixo, mostra um arquivo de imagem gerado que usa o esquema 4:1:1 do Chroma Subsamplig no canal Cb. Gerando grande perca de qualidade na imagem de saída em comparação com a imagem de entrada.

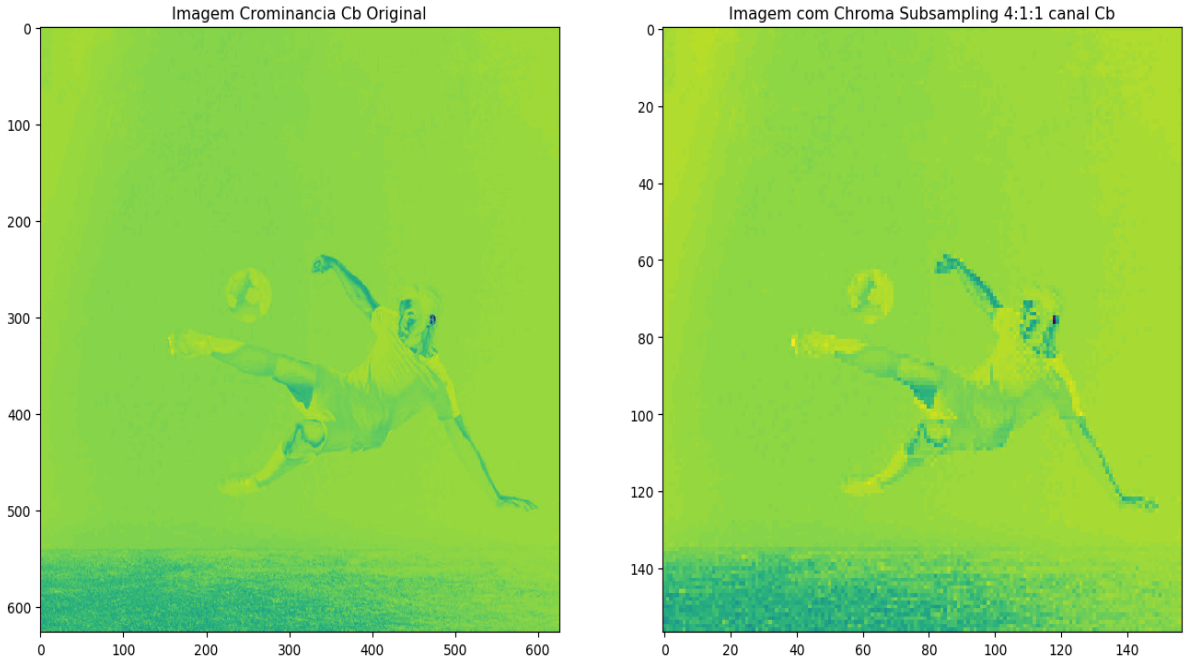

Figura 3

Código comentado

```
#importando bibliotecas
import cv2
import numpy as np
from matplotlib import pyplot as plt

#le a imagem do disco e cria o vetor
img = cv2.imread("jogador.jpg")

#converte em YCrCb
imgYCbCr=cv2.cvtColor( img, cv2.COLOR_BGR2YCR_CB)

#valor a dividir a crominância
v=2
```

```
h=2
#separa as crominâncias
cr= imgYCbCr[:,:,1]
cb= imgYCbCr[:,:,2]

cr_chroma=cr[::v,::h]
cb_chroma=cb[::v,::h]

#reconstroi a imagem
#Função nrepeat( imagem, vezes, eixo )
#Reproduzindo imagem com crominancia cr e cb valor igual a 2
m=2
img_aumentada_cr = np.repeat( cr_chroma, m, axis=0)
img_aumentada_cr = np.repeat(img_aumentada_cr, m, axis=1)

img_aumentada_cb = np.repeat( cb_chroma, m, axis=0)
img_aumentada_cb = np.repeat( img_aumentada_cb, m, axis=1)

pyplot python
imagemFinal = np.zeros( (img_aumentada_cr.shape[0],img_aumentada_cr.shape[1],3), np.uint8)
imagemFinal[:,:,0]= imgYCbCr[:,:,0]
imagemFinal[:,:,1]= img_aumentada_cr
imagemFinal[:,:,2]= img_aumentada_cb

imagem_final_rgb = cv2.cvtColor( imagemFinal, cv2.COLOR_YCrCb2RGB)

plt.subplot(121)
plt.imshow(imgYCbCr[:,:,1])
plt.title("Imagem Crominancia Cr Original")
plt.subplot(122)
plt.imshow(cr_chroma)
plt.title("Imagem com Chroma Subsampling 4:2:2 canal Cr")
```

```
plt.show()

cv2.waitKey(0)
cv2.destroyAllWindows()
```