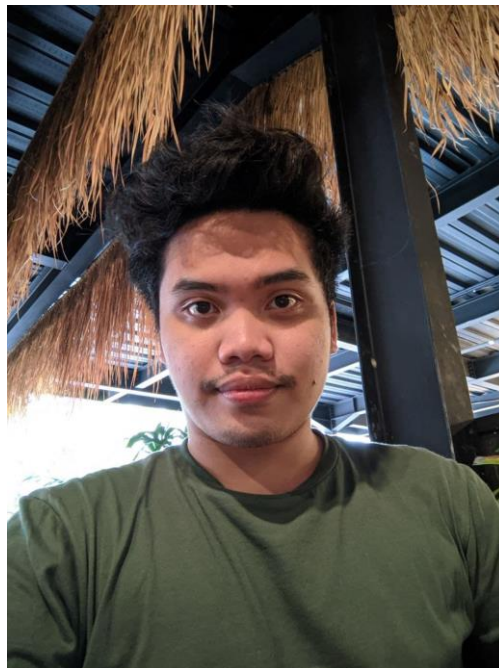


Ida Bagus Raditya A.M

18219117

Tugas Sismul: Discrete Cosine Transformation

Kali ini, saya akan melakukan sebuah kompresi citra/gambar dengan menggunakan metode Discrete Cosine Transformation (DCT). Gambar yang digunakan adalah foto selfie saya sendiri.



Gambar 1. Foto yang digunakan sebagai input

Ada beberapa tahapan untuk mengkompresi gambar tersebut menggunakan DCT. Tahapan-tahapan tersebut adalah

1. Mengkonversi warna RGB pada gambar tersebut menjadi YcbCr
2. Mengambil komponen gambar dan membaginya menjadi macroblock sebesar 8x8
3. Nilai warna dikurangi 128 untuk proses DCT
4. Melakukan Discrete Cosine Transformation (DCT)
5. Melakukan *quantization* dari hasil DCT
6. Hasil dari quantization tersebut akan dilakukan transform coding

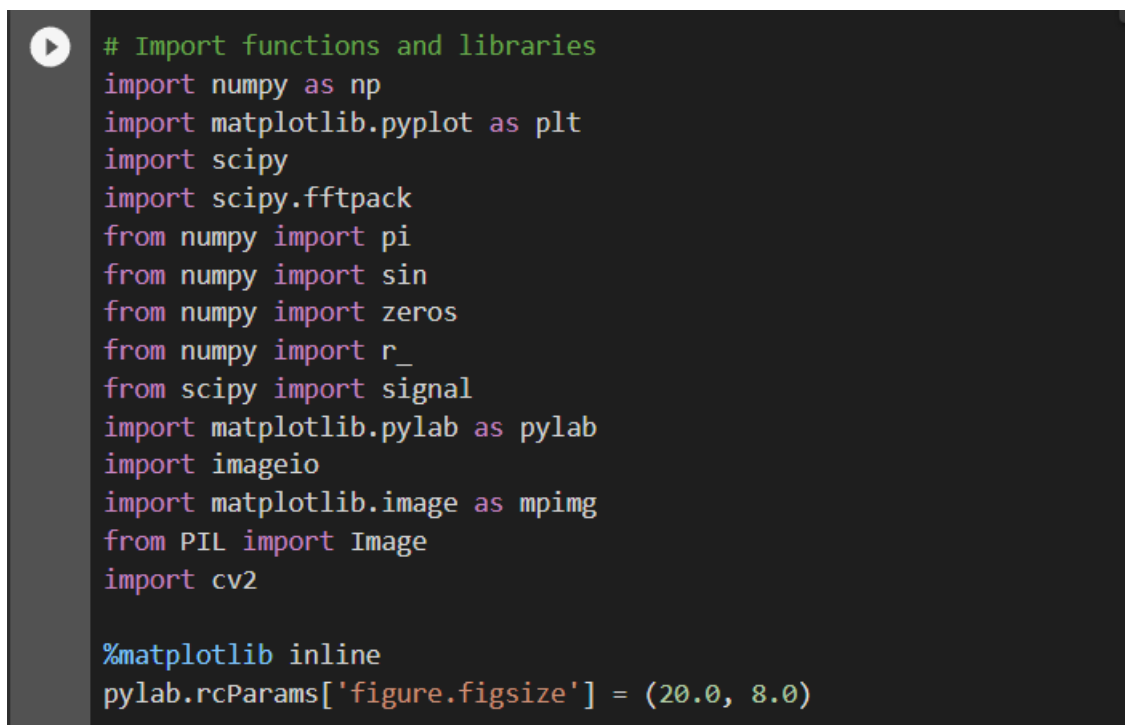
Kemudian, gambar yang telah dikompresi tersebut (secara lossy) akan di dekompresi dengan tahapan sebagai berikut

1. Gambar hasil transform coding akan dilakukan *dequantization*
2. Kemudian hasil dequantization akan dilakukan Inverse DCT

3. Nilai warna kemudian ditambahkan kembali nilai 128 setelah proses inverse DCT
4. Gambar berhasil di dekompresi

Namun pada kali ini, saya akan menjelaskan bagaimana melakukan **kompresi** pada foto saya diatas. Pada kesempatan kali ini, karena keterbatasan resource, saya menggunakan Python untuk melakukan kompresi gambar.

1. Import libraries terkait

A screenshot of a code editor with a dark background. On the left, there is a play button icon. The code is written in a light green font. It starts with a comment '# Import functions and libraries' followed by several import statements: 'import numpy as np', 'import matplotlib.pyplot as plt', 'import scipy', 'import scipy.fftpack', 'from numpy import pi', 'from numpy import sin', 'from numpy import zeros', 'from numpy import r_', 'from scipy import signal', 'import matplotlib.pyplot as pylab', 'import imageio', 'import matplotlib.image as mpimg', 'from PIL import Image', and 'import cv2'. At the bottom, there is a magic command '%matplotlib inline' and a line 'pylab.rcParams['figure.figsize'] = (20.0, 8.0)'.

Gambar 2. Library

Pada kompresi gambar, kita akan menggunakan banyak library, terutama numpy untuk kebutuhan angka dan matriks, matplotlib untuk visualisasi gambar, scipy untuk kebutuhan fungsi DCT, imageio untuk melakukan importing gambar, dan OpenCV2 untuk kebutuhan konversi RGB ke YcbCr.

2. Kemudian, kita akan melakukan import gambar dan mengkonversi gambar ke color space YCbCr. Kemudian, untuk kemudahan, kita hanya akan menggunakan komponen Y saja dari gambar tersebut lalu menyimpannya ke sebuah gambar PNG.

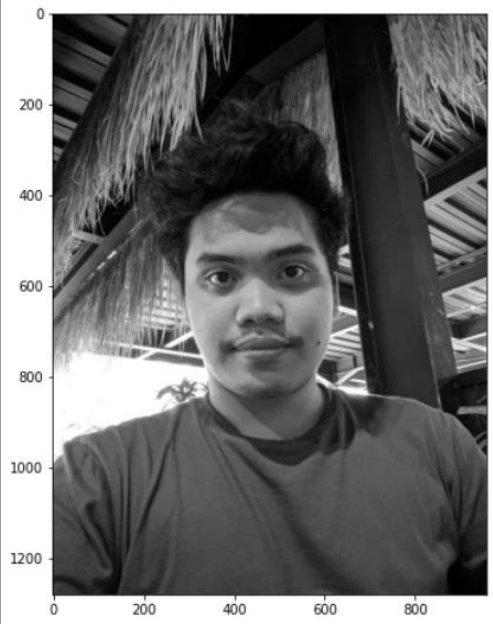
```
img = cv2.imread("tugus.jpg")
ycbcr = cv2.cvtColor(img, cv2.COLOR_BGR2YCR_CB)
cv2.imwrite('tugus-YCbCr.png', ycbcr[:, :, 0]) # get only the Luminance
```

Gambar 3. Import gambar

3. Kemudian kita akan lihat komponen Y dari gambar tersebut dengan kode berikut

```
im = imageio.imread('tugus-YCbCr.png')
im.shape # im is a numpy array
(256, 256)
f = plt.figure()
plt.imshow(im, cmap='gray')
```

<matplotlib.image.AxesImage at 0x7f234676b7d0>



Gambar 4. Gambar komponen Luminance

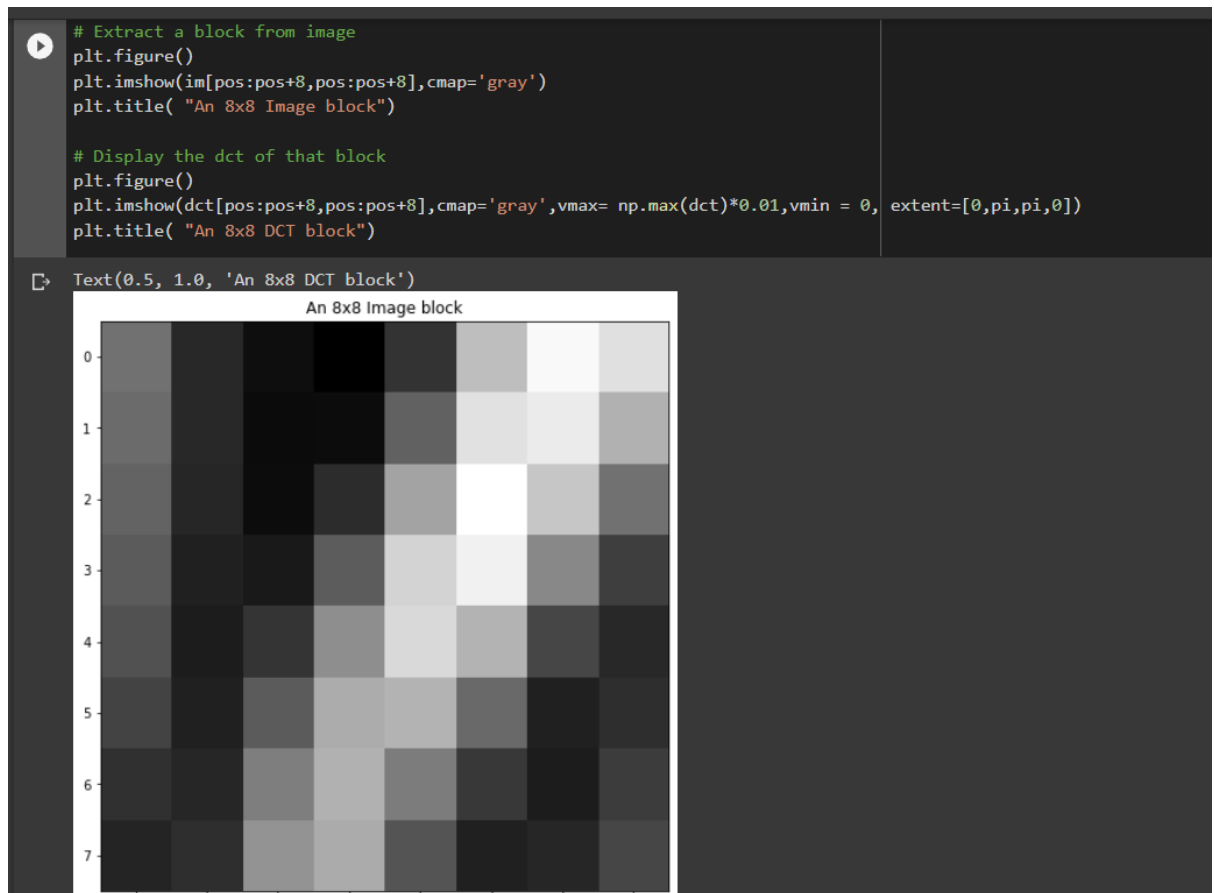
4. Lalu kita akan membuat rumus DCT dan Inverse DCT dengan Scipy. Scipy memiliki fungsi bawaan `fftpack.dct` yang dapat digunakan untuk proses kompresi DCT. Pada fungsi DCT ini, dilakukan juga quantization dan transform coding.

```
def dct2(a):
    return scipy.fftpack.dct( scipy.fftpack.dct( a, axis=0, norm='ortho' ), axis=1, norm='ortho' )

def idct2(a):
    return scipy.fftpack.idct( scipy.fftpack.idct( a, axis=0, norm='ortho' ), axis=1, norm='ortho' )
```

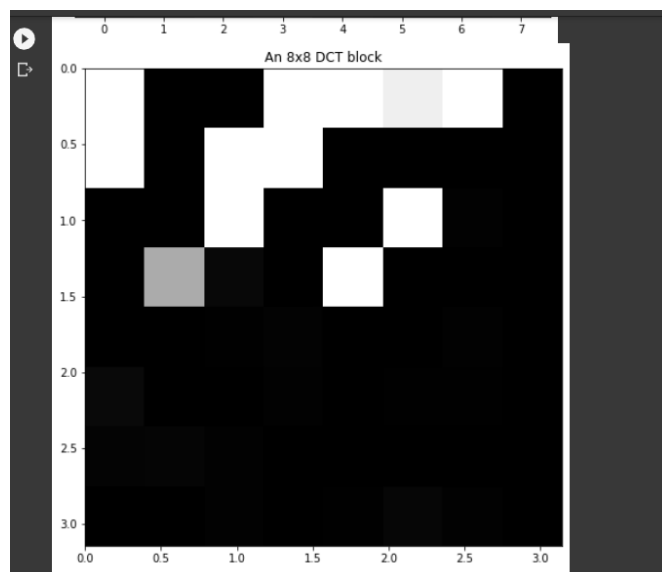
Gambar 5. Rumus DCT

5. Kemudian, kita akan mencuplik gambar menjadi banyak macroblock berukuran 8x8. Macroblock 8x8 yang dijadikan contoh adalah macroblock dari pixel ke 256.




Gambar 5. Macroblock

6. Kemudian, akan dilakukan DCT terhadap seluruh Macroblock berukuran 8x8. Sebagai contoh, ini hasil dari proses DCT pada makroblok contoh



Gambar 7. DCT pada sebuah Macroblock

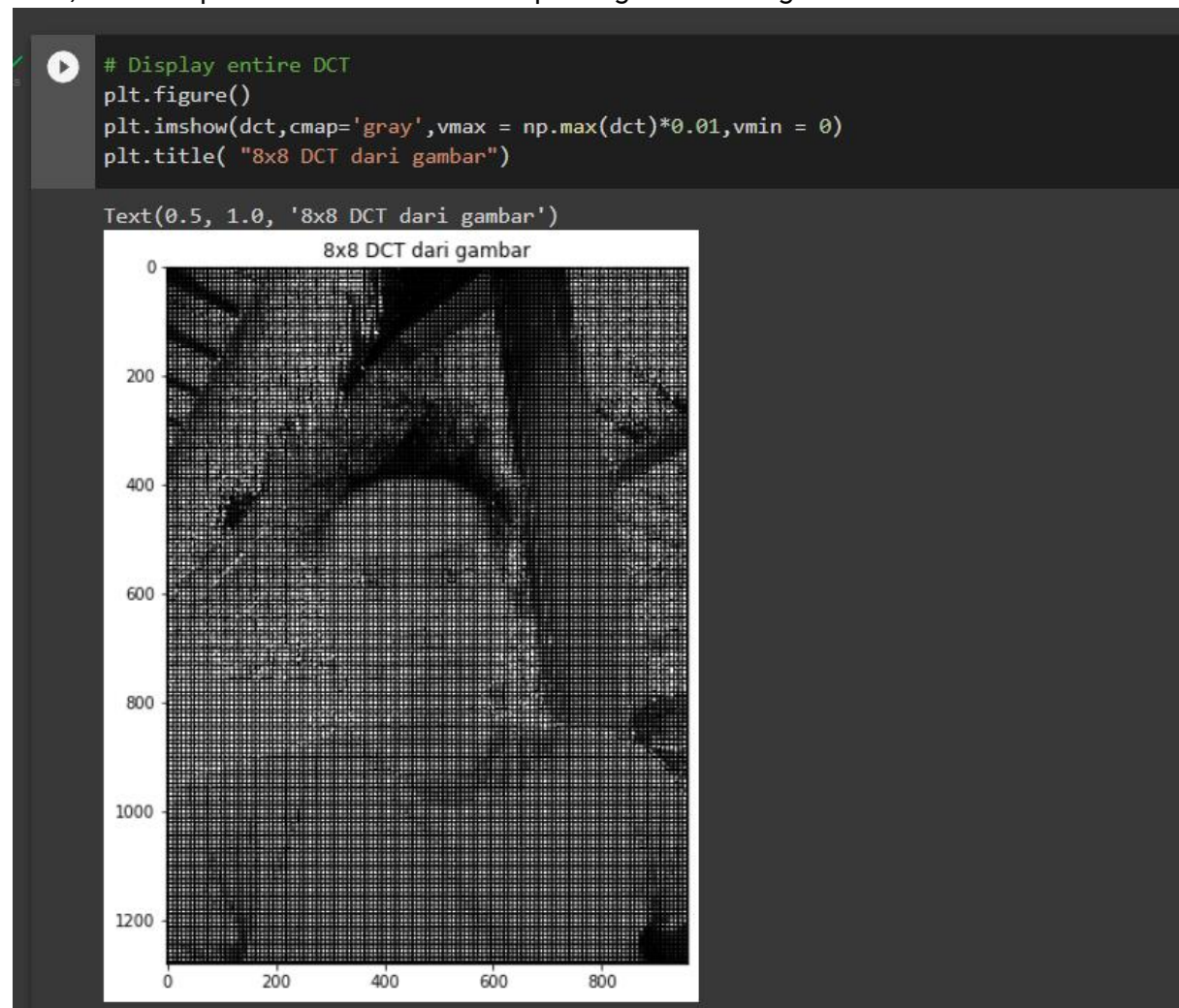
Untuk melakukan DCT pada seluruh macroblock, digunakan kode berikut. Pada saat dilakukan DCT dengan fungsi *dct2*, langsung dilakukan quantization dan transform coding

```
✓ Us  imsize = im.shape
dct = np.zeros(imsize)

# Do 8x8 DCT on image (in-place)
for i in r_[:imsize[0]:8]:
    for j in r_[:imsize[1]:8]:
        dct[i:(i+8),j:(j+8)] = dct2(im[i:(i+8),j:(j+8)])
```

Gambar 8. DCT pada seluruh Macroblock

7. Lalu, kita tempilkan seluruh hasil DCT pada gambar dengan kode berikut



Gambar 9. DCT dari seluruh macroblock

8. Kemudian, dilakukan dekompresi dari gambar dengan rumus Inverse DCT untuk melihat hasil kompresi yang dilakukan. Perlu diingat, saat melakukan rumus *idct2*, dilakukan pula *dequantization*



Gambar 10. Perbandingan gambar hasil kompresi dan dekompresi

Berikut lebih jelasnya



Gambar 11. Perbandingan gambar

Gambar di kiri adalah gambar asli, dan gambar yang kanan adalah gambar hasil kompresi DCT.



Gambar 12. Perbandingan zoomed