Jawaban LKP 6
Mata Kuliah Pengolahan Citra Digital

Nama : Muhammad Fakhri Alauddin Hidayat
NIM : G64170015
Hari : Kamis
Tanggal : 27 Februari 2020
Waktu : 08.00 - 10.00 WIB

Soal

1. Buatlah fungsi konversi citra RGB to HSV tanpa menggunakan fungsi OpenCV dan terapkan pada citra 'FACE DETECTION.png'. Rumus dapat dilihat dibawah

RGB ↔ HSV

In case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range.

$$V \leftarrow max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - min(R, G, B)) & \text{if } V = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

The values are then converted to the destination data type:

- 8-bit images: $V \leftarrow 255V, S \leftarrow 255S, H \leftarrow H/2$(to fit to 0 to 255)
- 16-bit images: (currently not supported) $V < -65535V, S < -65535S, H < -H$
- 32-bit images: H, S, and V are left as is

2. Lakukan face detection pada citra 'FACE DETECTION.png'
    - Pakai ilmu image subtraction, konvolusi, thresholding dan ilmu-ilmu lain dari pertemuan sebelumnya pada color space selain RGB sehingga menghasilkan segmentasi yang lebih bagus. Berilah alasan kenapa memilih color space tersebut, jika perlu sertakan juga paper rujukan.
    - Outpun dari proses ini adalah citra baru yang isinya gambar wajah saja.

***) Seluruh operasi dilakukan secara manual tanpa library**, kecuali operasi dasar seperti imwrite(), imshow(), imread()

Jawaban

Nomor 1

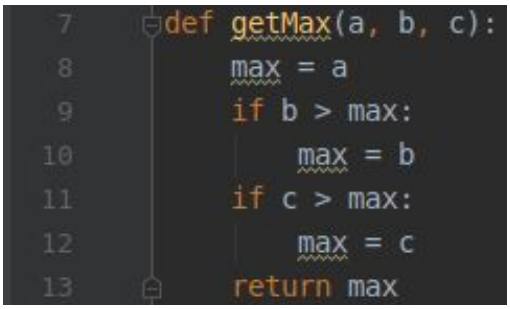
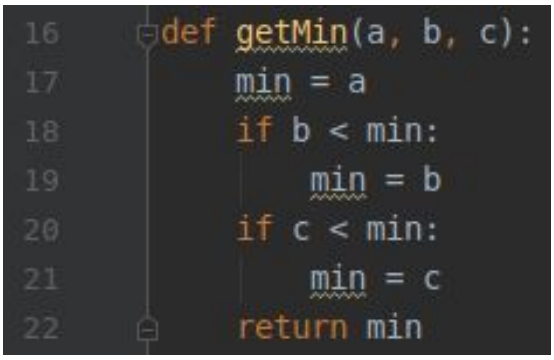
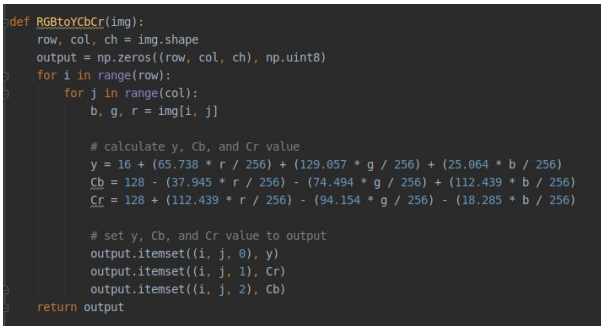
```python
def RGBtoHSV(img):
    row, col, ch = img.shape
    output = np.zeros((row, col, ch), np.uint8)
    for i in range(row):
        for j in range(col):
            b, g, r = img[i, j]

            # Change R G B to [0,1] range value
            r, g, b = r / 255.0, g / 255.0, b / 255.0
            cmax = getMax(r, g, b)
            cmin = getMin(r, g, b)
            diff = cmax - cmin

            # determine H value
            if cmax == cmin:
                h = 0
            elif cmax == r:
                h = (60 * ((g - b) / diff) + 360) % 360
            elif cmax == g:
                h = (60 * ((b - r) / diff) + 120) % 360
            elif cmax == b:
                h = (60 * ((r - g) / diff) + 240) % 360

            # determine S value
            if cmax == 0:
                s = 0
            else:
                s = (diff / cmax) * 100

            # determine V value
            v = cmax * 100
            output.itemset((i, j, 0), h)
            output.itemset((i, j, 1), s)
            output.itemset((i, j, 2), v)
    return output
```

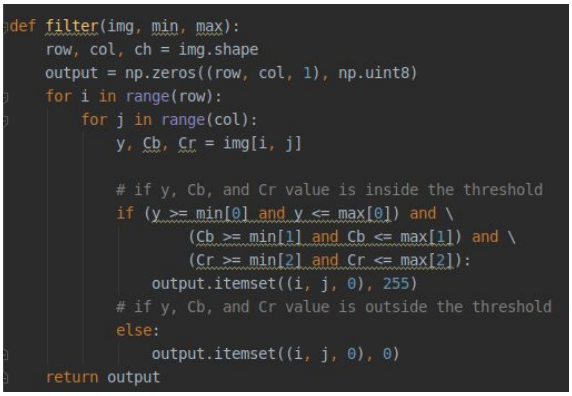
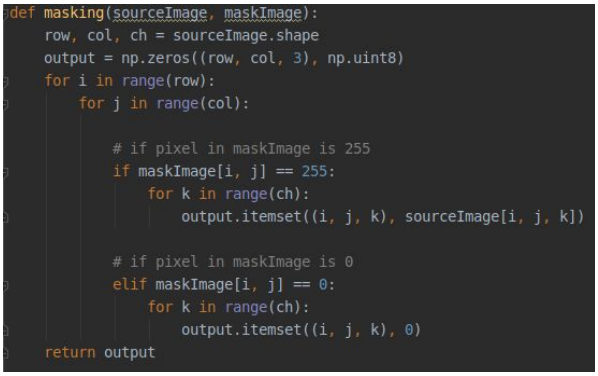
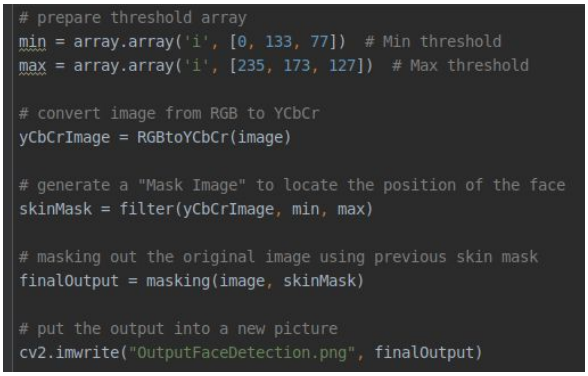
Fungsi RGBtoHSV():
Fungsi ini berfungsi untuk mengubah gambar dari RGB ke dalam HSV.
Ada iterasi yang akan dilakukan terhadap seluruh pixel dari gambar sumber. Lalu nilai *channel* RGB dari pixel di gambar sumber diekstrak kemudian digunakan untuk menentukan nilai channel HSV sesuai rumus yang ada di soal. Nilai channel HSV yang baru dipetakan ke dalam range nilai 0-255
Terakhir, nilai channel HSV yang didapatkan akan dimasukan ke dalam canvas output pada posisi yang bersesuaian dengan posisi pixel di gambar sumber yang sedang di proses.

Gambar sumber

Gambar hasil konversi RGB ke HSV

Nomor 2

	**getMax()** Fungsi ini berfungsi untuk mendapatkan nilai maksimum dari tiga buah nilai.
	**getMin()** Fungsi ini berfungsi untuk mendapatkan nilai minimum dari tiga buah nilai.
	**RGBtoYCbCr()** Fungsi ini berfungsi untuk mengkonversi RGB ke dalam bentuk YCbCr. Proses konversi menggunakan nilai channel RGB dari gambar sumber dan formula berikut: **Y= 16 + 65.738*R/256 + 129.057*G/256 + 25.064*B/256** **Cb = 128-37.945*R/256 - 74.494*G/256 + 112.439*B/256** **Cr = 128+112.439*R - 94.154*G/256 - 18.285*B/256** Setiap pixel dari gambar sumber akan diambil nilai channel RGB nya lalu nilai channel Y, Cb dan Cr akan dihitung menggunakan formula dan nilai channel RGB dari gambar sumber. Lalu nilai channel Y, Cb, dan Cr yang sudah dihitung akan dimasukan ke dalam canvas *output* pada posisi yang bersesuaian dengan pixel yang sedang di proses

	**filter()** Fungsi ini berfungsi untuk memfilter pixel mana yang memenuhi suatu threshold tertentu. Threshold yang dipakai kali ini adalah: **Channel Y : min = 0, max = 235** **Channel Cb : min =133, max = 173** **Channel Cr : min = 77, max = 127** Jika pixel yang sedang diproses memenuhi threshold diatas, maka pixel di canvas output pada posisi yang bersesuaian dengan pixel yang sedang di proses, akan diset nilainya menjadi 255. Selainnya akan diset nilainya menjadi 0.
	**masking()** Fungsi ini berfungsi untuk menghilangkan warna dari semua pixel (membuat nilai pixel menjadi 0) yang tidak *memenuhi threshold*. Fungsi ini menggunakan sebuah maskImage, yaitu gambar satu channel yang hanya berisi pixel dengan nilai 255 atau nilai 0. Jika pixel di maskImage bernilai 255 maka pixel di canvas output pada posisi yang bersesuain dengan pixel di maskImage akan diisi dengan nilai pixel dari gambar sumber di posisi yang sama. Selainnya, pixel akan diisi dengan nilai 0.
	Disamping adalah kode utama untuk melakukan *face detection.* Pertama, threshold disiapkan dalam bentuk array min dan array max. Lalu gambar dikonversi dari RGB ke YCbCr. Selanjutnya masking image dibuat dengan menggunakan gambar yang telah dikonversi dan threshold yang sudah disiapkan sebelumnya. Terakhir, dilakukan masking terhadap gambar sumber menggunakan masking image yang sudah dibuat sebelumnya dan outputnya berupa gambar baru yang bernama “OutputFaceDetection” dengan format PNG.

	
Gambar Sumber	**Gambar OutputFaceDetection.PNG**

Pada kasus kali ini saya memilih format YCbCr. Hal ini saya lakukan karena berdasarkan paper yang saya baca, bahwa format YCbCr sangat baik untuk melakukan *skin detection* yaitu mendeteksi kulit manusia dan juga karena pada gambar sumber tidak ada kulit lain yang terlihat selain kulih wajah. Selain itu, threshold yang saya pilih juga sangat membantu dalam mendeteksi kulit wajah, akan tetapi masih ada beberapa bagian yang bukan merupakan bagian dari wajah tapi lolos seperti bagian bendera dan *badge* nama.

Paper referensi yang saya sebutkan diatas, akan saya sertakan di dalam sebuah file ZIP bersama dengan laporan LKP ini.