Ahmad Maulvi Alfansuri

G64160081

Soal :

1. Buatlah fungsi konversi citra RGB to HSV tanpa menggunakan fungsi OpenCV dan terapkan pada citra 'FACE DETECTION.png'. Rumus dapat dilihat dilampirkan
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.
    - Output 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()

Keterangan:

1. Buat fail .docx yang isinya kodingan + penjelasan maksud kodingan + screenshot hasil gambarnya, dengan format fail: LKP5 NIM Nama (Ex: LKP5 G64150000 Muhammad Al Fatih.docx)
2. Buat kodingan asli python nya, dengan format fail: LKP5 NIM Nama (Ex: LKP5 G64150000 Muhammad Al Fatih.py)
3. Kedua file diatas (.docx dan .py) disimpan dalam arsip .zip, dengan format fail: LKP5 NIM Nama (Ex: LKP5 G64150000 Muhammad Al Fatih.zip). **Hanya fail .zip ini yang dikumpulkan.**

Usahakan kerja mandiri ya, supaya biar bener-bener bisa dan nggak ketergantungan ke orang lain. Semoga sukses selalu :)

Jawaban

1. Import setiap library yang dibutuhkan

```
import cv2
import numpy as np
from pprint import pprint
from sys import *
import matplotlib.pyplot as plt

# bit.ly/LKP6-PCD
# Ahmad Maulvi Alfansuri
# G64160081

##### LKP6

# load image dengan argumen nama file dan return image object
def load_image(filename="FACE DETECTION.png"):
	image = cv2.imread(filename)
	return image
```

2. Fungsi untuk mengubah pixel bgr to hsv

```
# fungsi conversi pixel rgb menjadi hsv
def conv_pixel_bgr_to_hsv(pixel):
	# buat pixel
	new_pixel = np.zeros((3), np.uint32)
	# ambil nilai v
	v = max(pixel)
	# jadikan nilai v integer
	v = int(v)

	# inisiasi s dan h
	s = 0
	h = 0
	# jika v == 0. s = 0. agar tidak terjadi error pembagian terhadap 0
	if(v != 0):
		s = (v - min(pixel)) * 1.0 / v
```

```
	# ekstrak b g r
	b, g, r = pixel

	# jadikan b g r integer
	b = int(b)
	g = int(g)
	r = int(r)
	# jika v == min(pixel), agar v - min(pixel) tidak 0 dan menghindari
pembagian terhadap 0
	if(v == min(pixel)):
		h == 0
	else:
		# rumus mendapatkan nilai h
		if(v == r):
			h =  (60 * (g - b))/(v - min(pixel))
		elif(v == g):
			h = 120 + (60 * (b - r))/(v - min(pixel))
		elif(v == b):
			h = 240 + ((60 * (r - g))/(v - min(pixel)))
		if(h < 0):
			h += 360

	# set normalisasi nilai h, s, v
	v = int(v)
	# 0 < s < 1. Kalikan 255 agar menjadi nilai 0 < s < 255
	s = int(255 * s)
	# karena maks(h) = 360. Maka bagi h dengan 2 agar menjadi < 255
	h = int(h / 2)
	# set pixel baru
	new_pixel = np.array([h,s,v])
	# return nilai pixel baru
	return new_pixel
```

3. Convert

```
# fungsi conv_image_bgr_to_hsv akan melakukan convert image dengan format bgr
menjadi hsv
def conv_image_bgr_to_hsv(image):
	# ambil dimensi
	(row, col, chan) = image.shape
	# buat canvas
	new_image = np.zeros((row,col,3), np.uint8)
	# loop
	for y in range(row):
		for x in range(col):
```

```
                # set pixel pada canvas baru
                new_image[y, x] = conv_pixel_bgr_to_hsv(image[y, x])
    # kembalikan image baru
    return new_image
```

4. Fungsi untuk menampilkan gambar dan menulis ke disk

```
# fungsi show akan menampilkan gambar dan menulis pada disk gambar tersebut seusai
judul yang diberikan
def show(image, title):
    # tulis ke disk
    cv2.imwrite(title, image)
    # tampilkan gambar
    cv2.imshow(title, image)
```

5. Mask gambar berdasarkan bound

```
# Fungsi masking akan menselect gambar yang dipilih berdasarkan bound warna yang
diberikan.
# Fungsi akan mengembalikan image mask, dengan warna putih sebagai pixel yang
diseleksi
def masking(image, lower_color, upper_color):
    # ambil dimensi
    (row, col, chan) = image.shape
    # buat mask dengan satu channel
    masker = np.zeros((row,col,1), np.uint8)
    # loop
    for y in range(row):
        for x in range(col):
            # select pixel dengan bound yang diberikan
            if( np.all(lower_color <= image[y, x]) and \
                np.all(upper_color >= image[y, x]) ) :
                # warnai pixel dengan warna putih
                masker[y,x,0] = 255
    # kemblaikan image mask
    return masker
```

6. Select image dengan filter mask yang diberikan

```
# Fungsi select_image akan mengambil pixel image dimana pixel tersebut diselect
oleh image mask.
def select_image(image, masker):
    # ambil dimensi
```

```
	(row, col, chan) = image.shape
	# buat canvas untuk gambar baru
	new_image = np.zeros((row,col,3), np.uint8)
	for y in range(row):
		for x in range(col):
			# jika mask berwarna putih
			if(np.all(masker[y,x,0] == 255)):
				# ambil pixel image. Warnai image baru dengan pixel
tersebut
				new_image[y,x] = image[y,x]
	# kembalikan image
	return new_image
```

7. Fungsi utama. Dimasukkan fungsi main agar fungsi menjadi modular

```
def main():
	# load image pada soal
	image = load_image()
	# convert gambar rgb menjadi hsv manual
	image_hsv = conv_image_bgr_to_hsv(image)
	# convert gambar menjadi hsv dengan library opencv
	image_hsv_auto = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

	# skin tone bound selection.
	# tune this one to get best result
	# dapat nilai ini dari paper
	# https://arxiv.org/pdf/1708.02694/
	# 0.0 <= H <= 50.0 and 0.23 <= S <= 0.68 and

	# coba tuning dengan bruteforce nilai s dan nilai v dan dapat nilai terbaik
segini
	# dengan ketepatan tinggi dengan contoh-hasil.png
	lower_color = np.array([0, 5, 0])
	upper_color = np.array([20, 255, 255])

	# buat mask image, dengan bound yang diberikan
	masker = masking(image_hsv, lower_color, upper_color)
	# select image berdasarkan mask yang ada
	image_selected = select_image(image, masker)

	# tampilkan gambar
	# gambar asli
	show(image, "gambar asli.jpg")
	# gambar hsv
	show(image_hsv, "image hsv.jpg")
	# gambar hsv auto
```

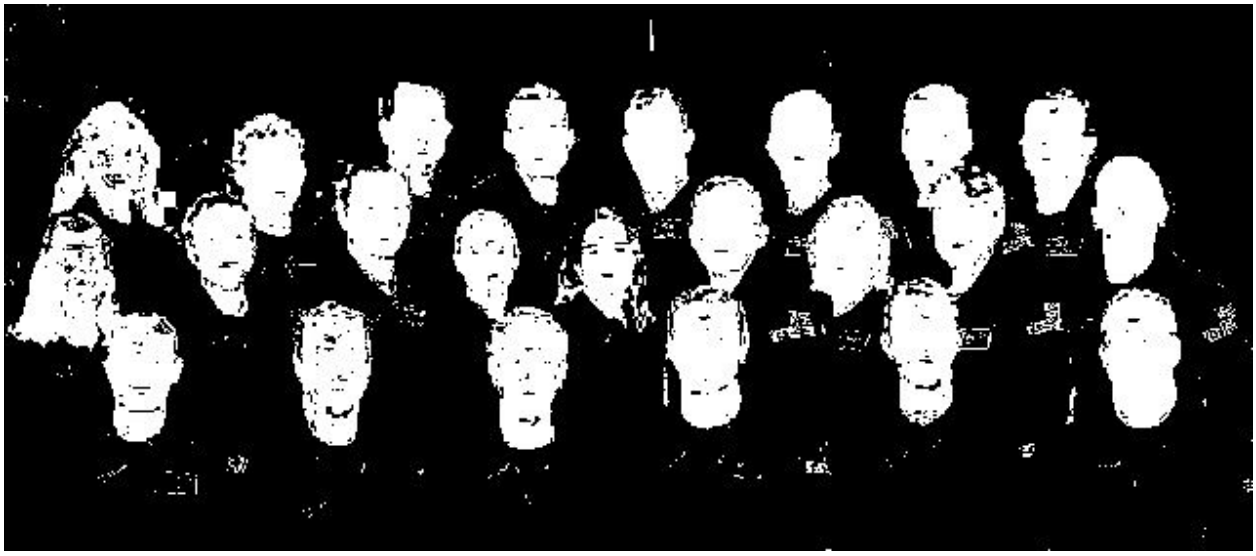
```
        show(image_hsv_auto, "image hsv auto.jpg")
        # mask image
        show(masker, "masker.jpg")
        # gambar hasil masking
        show(image_selected, "image selected.jpg")

        # finish
        cv2.waitKey(0)

# Fungsi main yang menjalankan program
main()
```

Screenshot dari skin formula hsv

er the values lie in a range of predefined threshold values for ea
The ranges for a skin pixel in different color spaces used by ou
$0.0 <= H <= 50.0$ and $0.23 <= S <= 0.68$ and
$R > 95$ and $G > 40$ and $B > 20$ and $R > G$ and $R > B$
and $|R - G| > 15$ and $A > 15$

Foto asli

Hasil HSV



Hasil masking

Hasil filter

Daftar isi:

Kolkur, S., et al. "Human skin detection using RGB, HSV and YCbCr color models." *arXiv preprint arXiv:1708.02694* (2017). https://arxiv.org/abs/1708.02694/