Ahmad Maulvi Alfansuri
G64160081

1. Baca dua citra (citra Cameraman.jpg dan citra Equalized.jpg) yang telah di sediakan di LMS.
2. Ubah dua citra tersebut menjadi grayscale
3. Kemudian hitung nilai rata-rata intensitas setiap citra tersebut
4. Setelah itu lakukan pengubahan tiap nilai intensitas piksel citra dengan ketentuan sebagai berikut:
   Jika nilai intensitas piksel dibawah rata-rata, kalikan dengan angka 0.5
   Jika nilai intensitas piksel diatas rata-rata atau sama dengan rata-rata, kalikan dengan angka 2
5. Lakukan proses pengurangan antara citra Cameraman dan citra Equalized seperti yang diajarkan saat praktikum
6. Tampilkan hasilnya Setelah itu berikan penjelasan singkat terhadap citra yang dihasilkan.

Diberikan dua buah citra, Cameraman.jpg dan Equalized.jpg seperti dibawah ini.

Fungsi ImageDiff memiliki algoritma seperti yang sudah dijelaskan pada soal. Berikut adalah prosedur fungsi fungsi yang digunakan untuk menyelesaikan fungsi tersebut.

```python
import cv2
import numpy as np
```
Import library yang digunakan pada praktikum ini

```python
def calculate(b,g,r):
        grey = r * 0.299 + g * 0.587 + b * 0.114
        return grey

def greyscaling_image(image):
        greyscale = np.zeros((row,col,1), np.uint8) # create dataframe
        for y in range(row):
                for x in range(col):
                        # get pixel
                        b, g, r = image[y, x]
                        # calculate
                        grey =  calculate(b,g,r)
                        # assign
                        greyscale.itemset((y,x,0), grey) # assign
        return greyscale
```

Fungsi calculate adalah fungsi untuk menkonversi 3 channel pixel menjadi channel greyscale
Fungsi greyscaling_image melakukan grayscaling pada image warna dan akan mengembalikan image dengan satu channel

```python
def rata_rata_intensitas(image):
	jumlah = 0
	for y in range(row):
		for x in range(col):
			grey = image[y, x]
			# calculate sum
			jumlah += int(grey)
	rata = jumlah * 1.0 / (x * y)  # bagi sum dengan jumlah data (height * weight)
	return rata
```

Fungsi rata_rata_intensitas, akan menghitung nilai rata rata intensitas warna pada image grayscale yang diberikan pada parameter. Fungsi mengembalikan nilai float yang merupakan seluruh penjumlahan intensitas citra dibagi banyak pixel (height * width)

```python
def perkuat_citra(image):
	# buat dataframe baru
	new_image = np.zeros((row,col,1), np.uint8)
	# ambil rata rata intensitas
	rata_rata = rata_rata_intensitas(image)
	for y in range(row):
		for x in range(col):
			# jika intensitas diatas rata-rata * 2
			if(image[y, x] >= rata_rata):
				powered_pixel = (image[y, x]) * 2
				# Batasi pixel menjad 255 bila lebih dari 255
				if(powered_pixel > 255):
					powered_pixel = 255
				# Batasi pixel menjadi 0 bila kurang dari 0
				elif(powered_pixel < 0):
					powered_pixel = 0
				new_image.itemset((y, x, 0), powered_pixel)
			elif(image[y, x] < rata_rata):
				# perkalian 0.5 tidak akan membuat pixel lebih dari 255 dan 0
				powered_pixel = (image[y, x]) * 0.5
				# kodingan dibawah bisa dioptimasi untuk mempercepat
				if(powered_pixel > 255):
					powered_pixel = 255
```

```
                elif(powered_pixel < 0):
                        powered_pixel = 0
                # set dataset baru dengan pixel yang telah diperkuat
                new_image.itemset((y, x, 0), powered_pixel)

        return new_image
```

Pada algoritma yang ditugaskan. Kita harus mengalikan nilai pada pixel dengan suatu konstanta apabila memenuhi konstrain tertentu. Apabila intensitas pixel melebihi dan sama dengan rata rata maka nilai pixel akan dikalikan dengan dua (menaikkan kontras). Apabila dibawahnya maka akan dikalikan dengan 0.5 (mengurangi kontras). Terdapat konstrain bahwa pixel dapat melebihi nilai batas byte (0x00 - 0xff), sehingga dibatasi dengan konstanta maksimal 0xff dan minimal 0.
Fungsi mengembalikan image yang sudah diperkuat.

```python
def difference(image1, image2):
        # buat dataset baru
        new_image = np.zeros((row,col,1), np.uint8)
        for y in range(row):
                for x in range(col):
                        # mengurangi nilai pixel (integer) image1 dengan image2
                        diff = int(image1[y, x]) - int(image2[y, x])
                        # batasi nilai pixel antara 0 - 255
                        if(diff > 255):
                                diff = 255
                        elif(diff < 0):
                                diff = 0
                        # return image
                        new_image.itemset((y, x, 0), diff)

        return new_image

```

Fungsi difference memerlukan dua parameter image1 dan image2 yang keduanya merupakan image dengan channel grayscale. Fungsi difference akan mengurangi pixel image1 dengan pixel pada image2. Nilai pixel pada setiap image diambil menjadi integer terlebih dahulu agar function override dari object cv2.image tidak merusak operasi aritmatik pada code. Jika kita langsung menjumlahkan dengan format image[x, y] maka operasi penjumlahan akan dioverride dengan method yang disediakan dari opencv untuk aritmatik pada pixel. Yang diinginkan oleh LKP, adalah operasi aritmatik biasa. Ini terlihat dari intended picture yang disediakan pada LKP yang membutuhkan nilai integer.

```python
def imageDiff(image1, image2, saved=1, debug=0, hasil="saved.jpg"):
	# load kedua gambar dengan grayscale
	greyscale_1 = greyscaling_image(image1)
	greyscale_2 = greyscaling_image(image2)

	# perkuat image dengan konstrain yang ada pada soal
	greyscale_1_powered = perkuat_citra(greyscale_1)
	greyscale_2_powered = perkuat_citra(greyscale_2)

	# lakukan pengurangan integer pada grayscale
	image = difference(greyscale_1_powered, greyscale_2_powered)

	# untuk melakukan debug hasil hasil proses pada gambar
	if(debug == 1):
		cv2.imshow("greyscale 1", greyscale_1)
		cv2.imshow("greyscale 2", greyscale_2)
		cv2.imshow("greyscale 1 powered", greyscale_1_powered)
		cv2.imshow("greyscale 2 powered", greyscale_2_powered)
	if(saved == 1):
	# save gambar menjadi file
		cv2.write(hasil, image)
	return image
```
Fungsi imageDiff adalah fungsi utama yang harus dibuat berdasarkan LKP. Fungsi ini membutuhkan dua parameter yaitu image1 dan image2 yang akan diproses. Fungsi akan melakukan greyscaling pada kedua

Solusi akhir
```python
import cv2
import numpy as np

# bit.ly/LKP3-PCD

def calculate(b, g, r):
	grey = r * 0.299 + g * 0.587 + b * 0.114
	return grey

def greyscaling_image(image):
```

```
	greyscale = np.zeros((row,col,1), np.uint8)
	for y in range(row):
		for x in range(col):
			# get pixel
			b, g, r = image[y, x]
			# calculate
			grey =  calculate(b,g,r)
			# assign
			greyscale.itemset((y,x,0), grey) # assign
	return greyscale

def rata_rata_intensitas(image):
	jumlah = 0
	for y in range(row):
		for x in range(col):
			grey = image[y, x]
			jumlah += int(grey)
	rata = jumlah / (x * y)
	return rata

def perkuat_citra(image):
	new_image = np.zeros((row,col,1), np.uint8)
	rata_rata = rata_rata_intensitas(image)
	for y in range(row):
		for x in range(col):
			if(image[y, x] >= rata_rata):
				powered_pixel = (image[y, x]) * 2
				if(powered_pixel > 255):
					powered_pixel = 255
				elif(powered_pixel < 0):
					powered_pixel = 0
				new_image.itemset((y, x, 0), powered_pixel)
			elif(image[y, x] < rata_rata):
				powered_pixel = (image[y, x]) * 0.5
				if(powered_pixel > 255):
					powered_pixel = 255
				elif(powered_pixel < 0):
					powered_pixel = 0
				new_image.itemset((y, x, 0), powered_pixel)

	return new_image

def difference(image1, image2):
```

```
	new_image = np.zeros((row,col,1), np.uint8)
	for y in range(row):
		for x in range(col):
			diff = int(image1[y, x]) - int(image2[y, x])
			if(diff > 255):
				diff = 255
			elif(diff < 0):
				diff = 0
			new_image.itemset((y, x, 0), diff)

	return new_image

def imageDiff(image1, image2, saved=1, debug=0, hasil="saved.jpg"):
	# load kedua gambar menjadi
	greyscale_1 = greyscaling_image(image1)
	greyscale_2 = greyscaling_image(image2)

	greyscale_1_powered = perkuat_citra(greyscale_1)
	greyscale_2_powered = perkuat_citra(greyscale_2)
	image = difference(greyscale_1_powered, greyscale_2_powered)

	if(debug == 1):
		cv2.imshow("greyscale 1", greyscale_1)
		cv2.imshow("greyscale 2", greyscale_2)
		cv2.imshow("greyscale 1 powered", greyscale_1_powered)
		cv2.imshow("greyscale 2 powered", greyscale_2_powered)
	if(saved == 1):
		cv2.imwrite(hasil, image)
	return image


image1 = cv2.imread("k_cameraman.jpg")
image2 = cv2.imread("k_equalized.jpg")
(row, col, ch) = image1.shape
image_diff = imageDiff(image1, image2)
## code dibawah error karena error qt pada komputer saya. Karena itu saya langsung save
image menjadi .jpg dengan imwrite
cv2.imshow("image diff", image_diff)
cv2.waitKey(0)
```