

# Tugas 1 Pengolahan Citra Video

## Transformasi Intensitas, Kontras, dan Saturasi

Kenanya Keandra Adriel Prasetyo  
5024211004

September 10, 2023

### 1 Pendahuluan

Pada Tugas 1 mata kuliah Pengolahan Citra Video, penulis diminta untuk membuat aplikasi dengan menggunakan bahasa python untuk melakukan transformasi intensitas pada suatu citra dan melakukan filtrasi pergeseran kontras suatu citra.

### 2 Pembahasan Program

Kode keseluruhan pada tugas struktur data array dapat diakses pada [https://github.com/KenanyaKAP/pcv\\_1](https://github.com/KenanyaKAP/pcv_1)

```
1 import cv2
2 import numpy as np
3
4 import matplotlib
5 matplotlib.use('TkAgg')
6 import matplotlib.pyplot as plt
7
8 img = cv2.imread('image2.jpg')
9 brightness = 0
10 contrast = 0
11 saturation = 0
12
13 fig, axs = plt.subplots(2, 2)
14 fig.suptitle('Histogram')
15
16 def updateImage():
17     # Container
```

```

18     imgUpdate = img.astype(np.double)
19
20     # Contrast
21     imgUpdate = np.clip(((.006*contrast+1) if contrast <= 0 else (.04*
22     contrast+1))*(imgUpdate-128)+128, 0, 255)
23
24     # Brightness
25     imgUpdate = np.clip(imgUpdate+brightness, 0, 255)
26
27     # Saturation
28     imgSat = (imgUpdate[:, :, 0] + imgUpdate[:, :, 1] + imgUpdate[:, :, 2])/3
29     lerpT = (saturation+100)/100
30     imgUpdate[:, :, 0] = imgSat*(1-lerpT) + imgUpdate[:, :, 0]*lerpT
31     imgUpdate[:, :, 1] = imgSat*(1-lerpT) + imgUpdate[:, :, 1]*lerpT
32     imgUpdate[:, :, 2] = imgSat*(1-lerpT) + imgUpdate[:, :, 2]*lerpT
33     imgUpdate = np.clip(imgUpdate, 0, 255)
34
35     # Create Histogram
36     histrGs = cv2.calcHist([imgSat.astype(np.uint8)], [0], None
37     , [256], [0, 256])
38     histrB = cv2.calcHist([imgUpdate[:, :, 0].astype(np.uint8)], [0], None
39     , [256], [0, 256])
40     histrG = cv2.calcHist([imgUpdate[:, :, 1].astype(np.uint8)], [0], None
41     , [256], [0, 256])
42     histrR = cv2.calcHist([imgUpdate[:, :, 2].astype(np.uint8)], [0], None
43     , [256], [0, 256])
44
45     # Plot Grayscale
46     axs[0, 0].cla()
47     axs[0, 0].set_title('Grayscale')
48     axs[0, 0].fill_between(np.arange(histrGs.shape[0]), histrGs.reshape
49     (-1), color='gray')
50
51     # Plot Blue
52     axs[0, 1].cla()
53     axs[0, 1].set_title('Blue')
54     axs[0, 1].fill_between(np.arange(histrB.shape[0]), histrB.reshape(-1)
55     , color='blue')
56
57     # Plot Green
58     axs[1, 0].cla()
59     axs[1, 0].set_title('Green')
60     axs[1, 0].fill_between(np.arange(histrG.shape[0]), histrG.reshape(-1)
61     , color='green')
62
63     # Plot Red
64     axs[1, 1].cla()
65     axs[1, 1].set_title('Red')
66     axs[1, 1].fill_between(np.arange(histrR.shape[0]), histrR.reshape(-1)
67     , color='red')
68
69     # Show image
70     imgUpdate /= 255

```

```

62     cv2.imshow('Image', imgUpdate)
63
64     # Convert histogram canvas to image and show
65     fig.canvas.draw()
66     histoImg = np.fromstring(fig.canvas.tostring_rgb(), dtype=np.uint8,
67                             sep='')
67     histoImg = histoImg.reshape(fig.canvas.get_width_height()[::-1] +
68                                (3,))
68     histoImg = cv2.cvtColor(histoImg, cv2.COLOR_RGB2BGR)
69     cv2.imshow("Histogram", histoImg)
70
71 def changeBrightness(value):
72     global brightness
73     brightness = value
74     updateImage()
75
76 def changeContrast(value):
77     global contrast
78     contrast = value
79     updateImage()
80
81 def changeSaturation(value):
82     global saturation
83     saturation = value
84     updateImage()
85
86 updateImage()
87
88 cv2.createTrackbar('Brightness', 'Image', 0, 100, lambda x:
89                  changeBrightness(x))
89 cv2.setTrackbarMin('Brightness', 'Image', -100)
90
91 cv2.createTrackbar('Contrast', 'Image', 0, 100, lambda x: changeContrast(
92                  x))
92 cv2.setTrackbarMin('Contrast', 'Image', -100)
93
94 cv2.createTrackbar('Saturation', 'Image', 0, 100, lambda x:
95                  changeSaturation(x))
95 cv2.setTrackbarMin('Saturation', 'Image', -100)
96
97 while 1:
98     k = cv2.waitKey(33) & 0xFF
99     if k == 27:
100         break
101 cv2.destroyAllWindows()

```

## 2.1 Import Library

```

1 import cv2
2 import numpy as np
3
4 import matplotlib

```

```

5 matplotlib.use('TkAgg')
6 import matplotlib.pyplot as plt

```

Program di atas menggunakan library opencv untuk membuka dan mengolah citra. Digunakan juga matplotlib untuk memploting grafik. Kemudian digunakan juga library numpy untuk melakukan operasi matrix.

## 2.2 Global Variable Read Image

```

1 img = cv2.imread('image2.jpg')
2 brightness = 0
3 contrast = 0
4 saturation = 0
5
6 fig, axs = plt.subplots(2, 2)
7 fig.suptitle('Histogram')

```

Terdapat beberapa global variable yang ada pada program ini. Variable `img` untuk menampung matrix gambar yang diambil dengan menggunakan function opencv.

Variable `brightness` memuat nilai yang akan ditambahkan ke matrix gambar untuk menambahkan kecerahan gambar

Variable `contrast` memuat nilai kontras yang akan menentukan kontras gambar

Variable `saturation` memuat nilai saturasi yang menentukan saturasi gambar, yang dapat membuat gambar menjadi grayscale

## 2.3 Main Program

Program utama di program ini dibagi menjadi beberapa bagian

### 2.3.1 Control Trackbar

```

1 def changeBrightness(value):
2     global brightness
3     brightness = value
4     updateImage()
5
6 def changeContrast(value):
7     global contrast
8     contrast = value
9     updateImage()
10
11 def changeSaturation(value):
12     global saturation
13     saturation = value
14     updateImage()
15

```

```

16 cv2.createTrackbar('Brightness', 'Image', 0, 100, lambda x:
    changeBrightness(x))
17 cv2.setTrackbarMin('Brightness', 'Image', -100)
18
19 cv2.createTrackbar('Contrast', 'Image', 0, 100, lambda x: changeContrast(
    x))
20 cv2.setTrackbarMin('Contrast', 'Image', -100)
21
22 cv2.createTrackbar('Saturation', 'Image', 0, 100, lambda x:
    changeSaturation(x))
23 cv2.setTrackbarMin('Saturation', 'Image', -100)

```

Kode di atas digunakan untuk membuat window control panel yang dapat digunakan untuk mengontrol brightness, contrast, dan saturation. Digunakan trackbar dari library opencv. Dengan begitu pengguna dapat dengan mudah melihat perubahan secara real-time saat dilakukan translasi intensitas dan kontras.

### 2.3.2 Main Loop

```

1 while 1:
2     k = cv2.waitKey(33) & 0xFF
3     if k == 27:
4         break
5 cv2.destroyAllWindows()

```

Kode di atas adalah main loop dari program. Program akan melakukan perulangan terus menerus hingga pengguna menekan tombol Esc pada perangkat yang akan memicu break loop, dan program berhenti

## 2.4 Update Image Function

Program juga memiliki fungsi updateImage dimana gambar yang telah dimuat akan masuk ke fungsi ini untuk kemudian diolah dengan melakukan transformasi intensitas, kontras, dan saturasi. Fungsi updateImage terdiri dari beberapa bagian yang dijelaskan pada bab berikut ini.

### 2.4.1 Container Variable

```

1 # Container
2 imgUpdate = img.astype(np.double)

```

Membuat variable baru berasal dari variable global img yang kemudian diubah ke tipe data double.

### 2.4.2 Contrast

```
1 # Contrast
2 imgUpdate = np.clip(((.006*contrast+1) if contrast <= 0 else
3 (.04*contrast+1))*(imgUpdate-128)+128, 0, 255)
```

Untuk melakukan translasi kontras, digunakan fungsi linear yang mentranslasikan nilai input menjadi suatu nilai output. Fungsi linear memiliki kemiringan yang dapat diatur dengan trackbar dan akan mengubah nilai kontras gambar. Semakin kecil nilai kontras, semakin datar fungsi linearnya. Namun semakin besar nilai kontras, semakin curam fungsi linearnya. Juga digunakan fungsi np.clip untuk membatasi nilai gambar agar tidak kurang dan tidak lebih dari 0 dan 255.

### 2.4.3 Brightness

```
1 # Brightness
2 imgUpdate = np.clip(imgUpdate+brightness, 0, 255)
```

Untuk melakukan translasi intensitas, hanya tinggal menambahkan matrix gambar dengan sebuah nilai, dan menggunakan fungsi np.clip untuk membatasi nilai gambar agar tidak kurang dan tidak lebih dari 0 dan 255.

### 2.4.4 Saturation

```
1 # Saturation
2 imgSat = (imgUpdate[:, :, 0] + imgUpdate[:, :, 1] + imgUpdate[:, :, 2])/3
3 lerpT = (saturation+100)/100
4 imgUpdate[:, :, 0] = imgSat*(1-lerpT) + imgUpdate[:, :, 0]*lerpT
5 imgUpdate[:, :, 1] = imgSat*(1-lerpT) + imgUpdate[:, :, 1]*lerpT
6 imgUpdate[:, :, 2] = imgSat*(1-lerpT) + imgUpdate[:, :, 2]*lerpT
7 imgUpdate = np.clip(imgUpdate, 0, 255)
```

Untuk melakukan translasi saturasi, perlu dilakukan kalkulasi saturasi grayscale terlebih dahulu. Untuk mendapatkan grayscale, bisa dilakukan dengan memuat gambar secara grayscale ataupun tinggal menambahkan ketiga elemen Red Green Blue, kemudian dibagi dengan 3.

Selanjutnya nilai Red Green Blue pada gambar asli diubah dengan menggunakan Lerp-Function. Fungsi Lerp digunakan untuk mencari sebuah angka di antara dua angka, yang diatur dengan suatu nilai. Dalam hal ini adalah lerpT. Bila lerpT bernilai 0, fungsi akan mengeluarkan nilai grayscale. Namun bila lerpT bernilai 1, fungsi akan mengeluarkan nilai asli Red Green Blue gambar. Dan bila lerpT bernilai lebih dari 1, nilai akan dikalikan sehingga saturasi gambar bertambah

### 2.4.5 Histogram

```
1 # Create Histogram
2 histrGs = cv2.calcHist([imgSat.astype(np.uint8)], [0], None, [256], [0, 256])
3 histrB = cv2.calcHist([imgUpdate[:, :, 0].astype(np.uint8)], [0], None
4     , [256], [0, 256])
5 histrG = cv2.calcHist([imgUpdate[:, :, 1].astype(np.uint8)], [0], None
6     , [256], [0, 256])
7 histrR = cv2.calcHist([imgUpdate[:, :, 2].astype(np.uint8)], [0], None
8     , [256], [0, 256])
```

Kode diatas digunakan untuk menghitung masing-masing histogram pada nilai warna gambar. Masing-masing untuk grayscale, warna biru, warna hijau, dan warna merah.

### 2.4.6 Plot the Histogram

```
1 # Plot Grayscale
2 axs[0, 0].cla()
3 axs[0, 0].set_title('Grayscale')
4 axs[0, 0].fill_between(np.arange(histrGs.shape[0]), histrGs.reshape(-1),
5     color='gray')
6
7 # Plot Blue
8 axs[0, 1].cla()
9 axs[0, 1].set_title('Blue')
10 axs[0, 1].fill_between(np.arange(histrB.shape[0]), histrB.reshape(-1),
11     color='blue')
12
13 # Plot Green
14 axs[1, 0].cla()
15 axs[1, 0].set_title('Green')
16 axs[1, 0].fill_between(np.arange(histrG.shape[0]), histrG.reshape(-1),
17     color='green')
18
19 # Plot Red
20 axs[1, 1].cla()
21 axs[1, 1].set_title('Red')
22 axs[1, 1].fill_between(np.arange(histrR.shape[0]), histrR.reshape(-1),
23     color='red')
```

Kemudian histogram yang sudah dikalkulasi ditampilkan dengan menggunakan bantuan library matplotlib pyplot. Masing-masing untuk grayscale, warna biru, warna hijau, dan warna merah.

### 2.4.7 Show Image

```
1 # Show image
2 imgUpdate /= 255
3 cv2.imshow('Image', imgUpdate)
```

```

4
5 # Convert histogram canvas to image and show
6 fig.canvas.draw()
7 histoImg = np.fromstring(fig.canvas.tostring_rgb(), dtype=np.uint8, sep=
    ')
8 histoImg = histoImg.reshape(fig.canvas.get_width_height()[::-1] + (3,))
9 histoImg = cv2.cvtColor(histoImg, cv2.COLOR_RGB2BGR)
10 cv2.imshow("Histogram", histoImg)

```

Kemudian langkah terakhir adalah menampilkan hasil gambar yang telah ditransformasi. Masing-masing untuk gambar yang dimuat dan gambar histogram.