Histogram

Contrast Stretching, Histogram Equalization

Nama : Muhamad Hilmy Haidar
NIM : G64170030
Tanggal : 13 Februari 2020
Nama Asisten : 1. Hilmi Farhan Ramadhani G64160048 2. Kautsar Ibrahim Hilmi G64160073

1. Membuat fungsi contrast stretching tanpa menggunakan OpenCV pada car.png

   Dengan menggunakan algoritma `Min Max Stretching` dapat diperoleh nilai renggangan untuk merubah nilai asli menjadi lebih tersebar (lebih contrast).

   ```
   def minMaxStretching(x, minimum, maximum):
      return (x - minimum) / (maximum - minimum) * MAX_PIXEL
   ```

   Operasi Contrast Stretching dimulai dengan loop 2D untuk mendapat nilai maksimum dan minimum dari nilai-nilai pixel yang ada di gambar.

   ```
   minimum = MAX_PIXEL
      maximum = MIN_PIXEL

      for i in range(row):
          for j in range(col):
              minimum = min(minimum, image[i, j])
              maximum = max(maximum, image[i, j])
   ```

   Setelah itu lakukan fungsi stretching tadi pada tiap-tiap pixel sambil dihitung frekuensi kemunculannya dari nilai gambar asli dan nilai yang telah di regangkan.

   ```
   frequency = [0] * (MAX_PIXEL + 1)
      real = [0] * (MAX_PIXEL + 1)
      value = 0
      for i in range(row):
          for j in range(col):
   ```

```
           value = int(minMaxStretching(image[i, j], minimum,
maximum).round())
           canvas[i, j] = value
           frequency[value] += 1
           real[image[i, j]] += 1
```

Berikut kode lengkap fungsi contrast stretching :

```
def minMaxStretching(x, minimum, maximum):
   return (x - minimum) / (maximum - minimum) * MAX_PIXEL


def plotHistogram(data, title=None, width=1):
   if title is None:
       title = ['Default']
   fig, ax = plt.subplots(len(data))
   freq = []
   for i in range(len(data)):
       freq.append({j: data[i][j] for j in range(0, MAX_PIXEL+1)})
       ax[i].set_title('{} Histogram'.format(title[i]))
       ax[i].bar(freq[i].keys(), freq[i].values(), width)

   fig.tight_layout()
   plt.show()


def contrastStretching(image):
   row, col = image.shape

   canvas = np.zeros((row, col, 1), np.uint8)

   minimum = MAX_PIXEL
   maximum = MIN_PIXEL

   for i in range(row):
       for j in range(col):
           minimum = min(minimum, image[i, j])
           maximum = max(maximum, image[i, j])

   frequency = [0] * (MAX_PIXEL + 1)
```

```
    real = [0] * (MAX_PIXEL + 1)
    value = 0
    for i in range(row):
        for j in range(col):
            value = int(minMaxStretching(image[i, j], minimum,
maximum).round())
            canvas[i, j] = value
            frequency[value] += 1
            real[image[i, j]] += 1

    plotHistogram(
        [frequency, real],
        title=['Contrast', 'Real Image']
    )

    cv2.imshow('Contrast', canvas)
    return frequency
```

2. Membuat fungsi histogram equalization tanpa menggunakan fungsi OpenCV dan melakukan proses histogram equalization pada car.png

   Hal yang pertama dilakukan yaitu hitung seluruh frekuensi kemunculan dari nilai-nilai pixel di gambar yang asli.

```
frequency = [0] * (MAX_PIXEL + 1)

    for i in range(row):
        for j in range(col):
            frequency[image[i, j]] += 1
```

   Lalu hitung peluang nya untuk mendapatkan frekuensi yang telah dinormalisasi.

```
normalize = []
    for value in frequency:
        normalize.append(value / sum(frequency))
```

   Setelah itu lakukan proses kumulatif dari nilai frekuensi tadi yang sudah dinormalisasi.

```
cumulative = []
   before = 0
   for value in normalize:
       cumulative.append(value + before)
       before += value
```

Yang terakhir yaitu mengalikan nilai maksimum dari range pixel (yaitu 255) dengan nilai kumulatif frekuensinya. Jangan lupa untuk membulatkan nilai nya ke bentuk integer.

```
equalize = []
   for value in cumulative:
       equalize.append(round(value * MAX_PIXEL))
```

Hasil pengalian dari sebelumnya dijadikan nilai pixel yang baru

```
for i in range(row):
       for j in range(col):
           canvas[i, j] = equalize[image[i, j]]
```

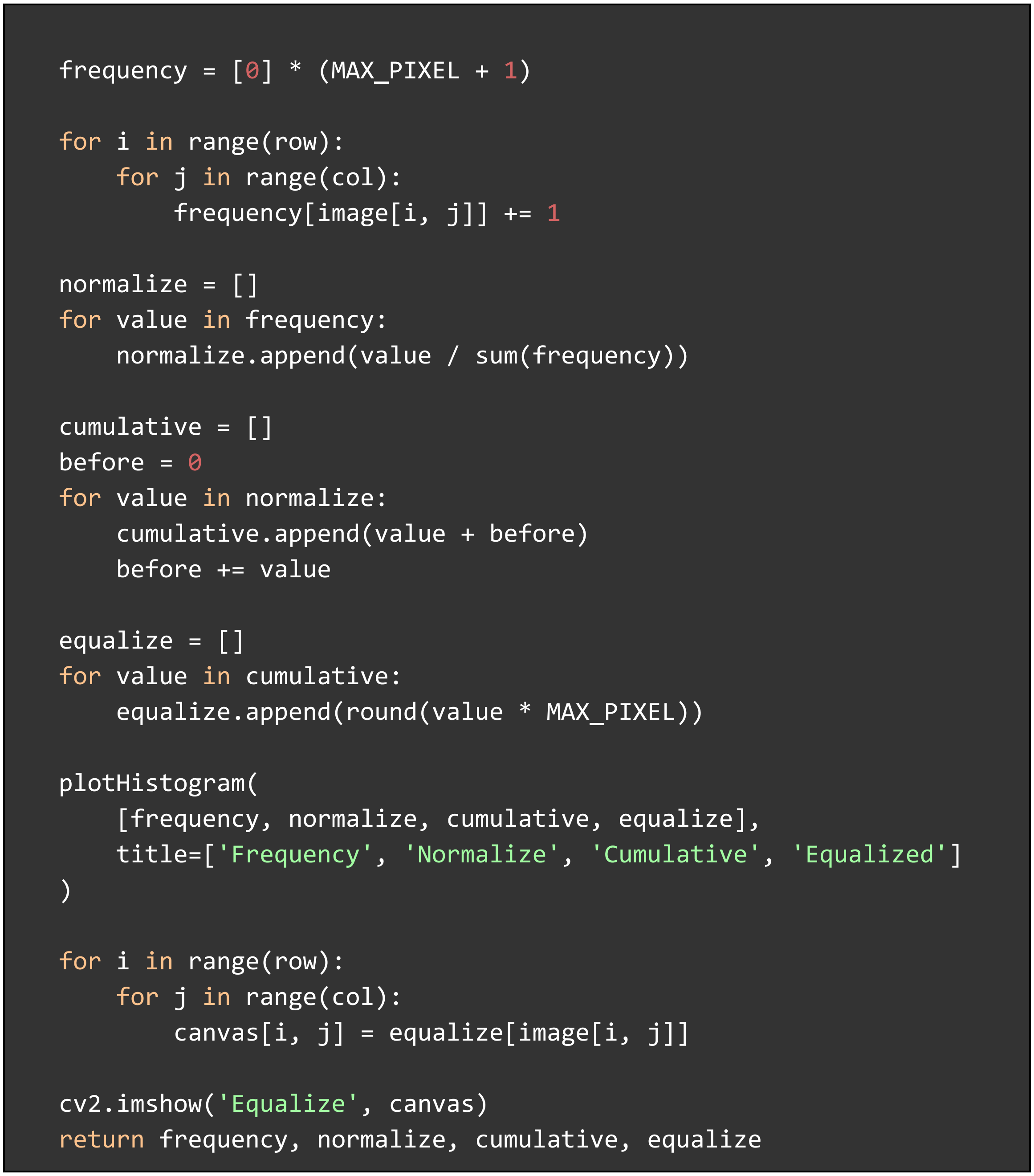
Kode lengkap nya :

```
def plotHistogram(data, title=None, width=1):
   if title is None:
       title = ['Default']
   fig, ax = plt.subplots(len(data))
   freq = []
   for i in range(len(data)):
       freq.append({j: data[i][j] for j in range(0, MAX_PIXEL+1)})
       ax[i].set_title('{} Histogram'.format(title[i]))
       ax[i].bar(freq[i].keys(), freq[i].values(), width)

   fig.tight_layout()
   plt.show()


def histogramEqualization(image):
   row, col = image.shape

   canvas = np.zeros((row, col, 1), np.uint8)
```



```
frequency = [0] * (MAX_PIXEL + 1)

for i in range(row):
    for j in range(col):
        frequency[image[i, j]] += 1

normalize = []
for value in frequency:
    normalize.append(value / sum(frequency))

cumulative = []
before = 0
for value in normalize:
    cumulative.append(value + before)
    before += value

equalize = []
for value in cumulative:
    equalize.append(round(value * MAX_PIXEL))

plotHistogram(
    [frequency, normalize, cumulative, equalize],
    title=['Frequency', 'Normalize', 'Cumulative', 'Equalized']
)

for i in range(row):
    for j in range(col):
        canvas[i, j] = equalize[image[i, j]]

cv2.imshow('Equalize', canvas)
return frequency, normalize, cumulative, equalize
```