

Laporan Programming Assignment Pengolahan Citra dan Video 1

Pada program pertama `autocontrastwithcdf.py` menggunakan perhitungan dari rumus cdf, melalui beberapa cara yaitu mengload image dengan library pillow kemudian menjadikan data tersebut kedalam array melalui library numpy dan menjadikan array tersebut menjadi satu dimensi.

```
▶ M1

# fungsi untuk mencari cumulative histogram
def cumulative(a):
    a = iter(a)
    b = [next(a)]
    for i in a:
        b.append(b[-1] + i)
    return np.array(b)

cs = cumulative(hist)
plt.plot(cs)
```

```
▶ M1

# fungsi untuk mendapatkan histogram dari data image
def gethist(image, bins):
    histogram = np.zeros(bins)

    for pixel in image:
        histogram[pixel] += 1

    return histogram

hist = gethist(flat, 256)
```

Dengan menggunakan kedua fungsi tersebut, maka diperoleh data histogram yang kemudian digunakan untuk menghitung cdf.

```
▶ M1

# fungsi normalisasi agar data tidak terlalu besar
nj = (cs - cs.min()) * 255
N = cs.max() - cs.min()
cs = nj / N
# mengembalikan tipe data ke uint8 karena dalam image tidak berlaku float
cs = cs.astype('uint8')

plt.plot(cs)
```

Kemudian dilakukan normalisasi agar data yang dihasilkan tidak terlalu besar, dan dilakukan pengubahan tipe data kembali ke uint8 karena dalam image tidak berlaku float.

```
# memindahkan variabel cs untuk tiap indek ke variable img_new
img_new = cs[flat]
# mengembalikan ukuran image semula, yang sebelumnya diflatten
img_new = np.reshape(img_new, img1.shape)

# setup plot
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

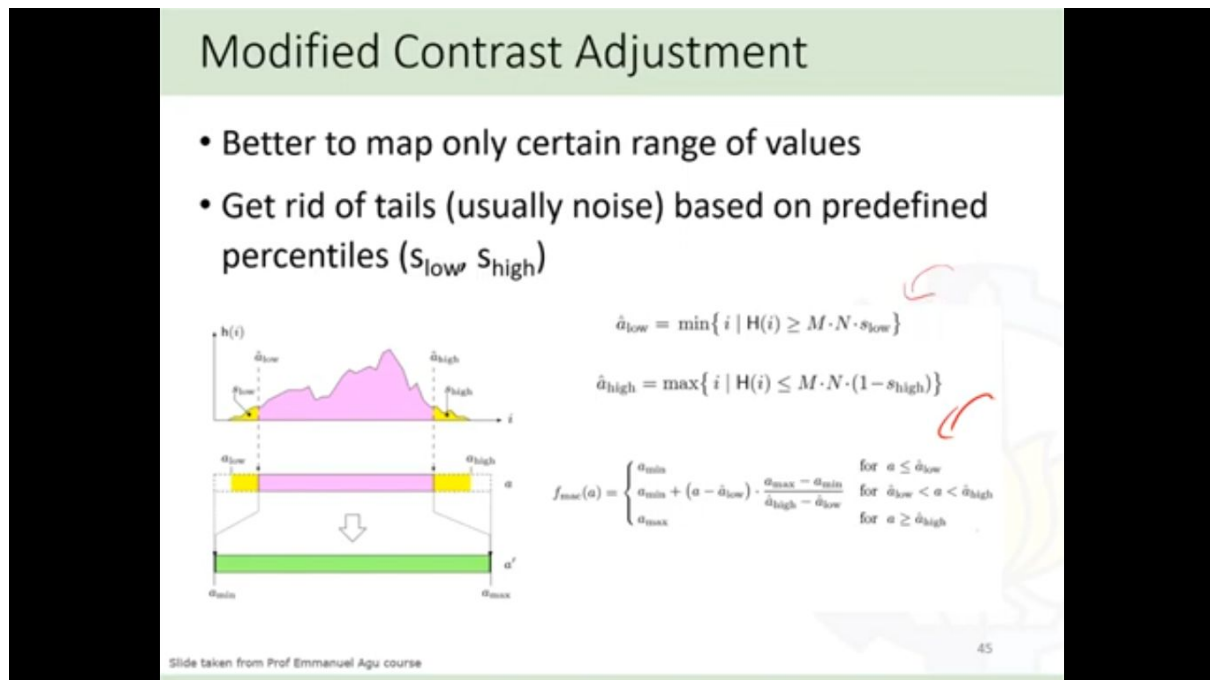
fig.add_subplot(1,2,1)
plt.imshow(img, cmap='gray')

fig.add_subplot(1,2,2)
plt.imshow(img_new, cmap='gray')

plt.show(block=True)
```

Setelah itu dilakukan pemindahan dan pengaturan ukuran kembali sesuai dengan image yang di load. dan hasil dapat di plot.

Pada program kedua yang bernama `autocontrastwithformula.py` menggunakan perhitungan dari rumus dari slide berikut.



Cara yang digunakan dalam program adalah menload image dengan library `opencv`.

```

M4
#Mencari nilai a_low dan a_high
a_low = 127
a_high = 127
for i in range(width):
    for j in range(height):
        if(img[j, i] < a_low):
            a_low = img[j, i]
        elif(img[j, i] > a_high):
            a_high = img[j, i]

print(a_low)
print(a_high)

```

kemudian mencari nilai `a_low` dan `a_high` dengan cara menloop semua pixel yang ada.

```

M4
# menentukan auto contrast
a_min = 0
a_max = 255
# set variable array kosong dengan ukuran sesuai image
img_new = np.zeros((height, width), np.uint8)
for i in range(width):
    for j in range(height):
        if(img[j, i] <= a_low):
            img_new[j, i] = a_min
        elif(img[j, i] > a_low and img[j, i] < a_high):
            img_new[j, i] = a_min + ((img[j, i] - a_low) * ((a_max - a_min) / (a_high - a_low)))

        elif(img[j, i] >= a_high):
            img_new[j, i] = a_max

# plot image sesudah dan sebelum
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

```

Lalu dilakukan perhitungan dengan rumus sesuai slide dan dapat diplot. (img_new digunakan sebagai variabel untuk menampung gambar yang baru)