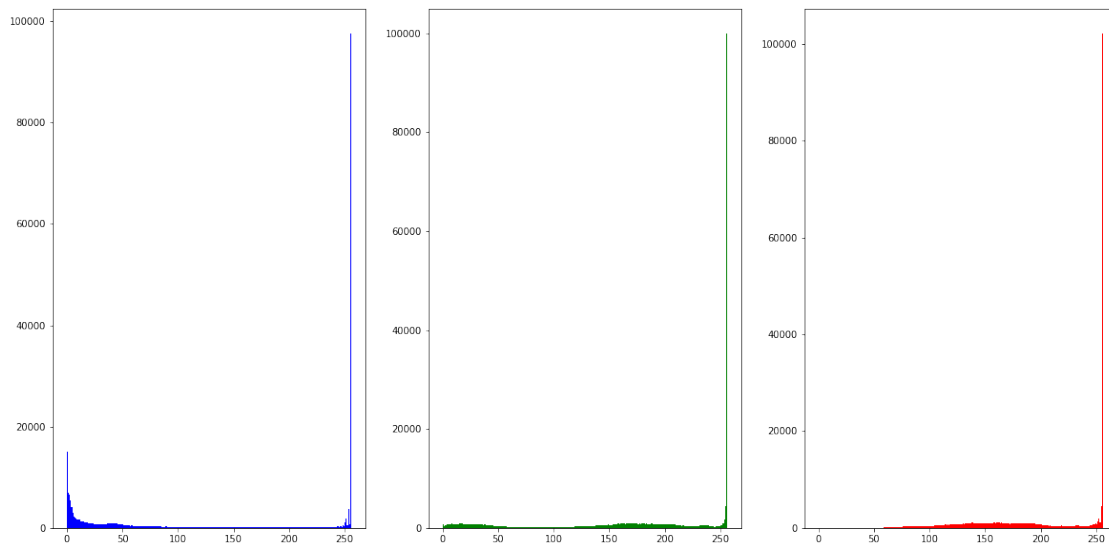


## Plot Image Histogram warna RGB

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

# membaca image
img = cv.imread('apel.jpg',cv.IMREAD_COLOR)

#set ukuran plot
plt.figure(figsize=(20, 10))
#membuat histogram dari warna biru
plt.subplot(1, 3, 1),plt.hist(img[:, :,0].ravel(),256,[0,256], color =
'b');
#membuat histogram dari warna hijau
plt.subplot(1, 3, 2),plt.hist(img[:, :,1].ravel(),256,[0,256], color =
'g');
#membuat histogram dari warna merah
plt.subplot(1, 3, 3),plt.hist(img[:, :,2].ravel(),256,[0,256], color =
'r');
#Menampilkan hasilnya
plt.show()
```



## Image Histogram Equalization

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
```

```
#membaca image
```

```
img = cv.imread('low.jpg',0)
```

```
#membuat hist dan bins
```

```
hist,bins = np.histogram(img.flatten(),256,[0,255])
```

```
# hist adalah nilai dari gray levelnya / banyaknya nilai tiap intensity
```

```
hist
```

```
array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  2,  1,  0,  4,  2,  3, 10,  9, 16,
       20, 27, 28, 39, 83, 84, 128, 174, 243, 267, 357,
       461, 484, 612, 682, 803, 834, 1030, 906, 813, 736, 656,
       545, 533, 565, 495, 533, 559, 596, 529, 484, 609, 525,
       567, 621, 758, 617, 674, 971, 1389, 766, 726, 587, 782,
       634, 680, 988, 775, 748, 798, 879, 787, 1142, 1825, 818,
       1675, 871, 910, 917, 1433, 1950, 1114, 1616, 1136, 1220, 997,
       1632, 1037, 693, 714, 756, 781, 532, 752, 718, 780, 520,
       606, 483, 370, 357, 325, 350, 380, 434, 578, 342, 353,
       350, 320, 460, 377, 656, 397, 413, 395, 206, 139, 111,
       87,
```

```

3,      99,   44,   37,   34,   21,   11,    6,    7,    4,    5,
0,      1,    1,    0,    0,    0,    0,    0,    0,    1,    0,
0,      0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,      0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,      0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,      0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,      0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,      0,    0,    0], dtype=int64)

```

*#bins sebagai gray level / intensity*  
bins

```

array([ 0.          ,  0.99609375,  1.9921875 ,  2.98828125,
        3.984375 ,  4.98046875,  5.9765625 ,  6.97265625,
        7.96875  ,  8.96484375,  9.9609375 , 10.95703125,
       11.953125 , 12.94921875, 13.9453125 , 14.94140625,
       15.9375   , 16.93359375, 17.9296875 , 18.92578125,
       19.921875 , 20.91796875, 21.9140625 , 22.91015625,
       23.90625  , 24.90234375, 25.8984375 , 26.89453125,
       27.890625 , 28.88671875, 29.8828125 , 30.87890625,
       31.875    , 32.87109375, 33.8671875 , 34.86328125,
       35.859375 , 36.85546875, 37.8515625 , 38.84765625,
       39.84375  , 40.83984375, 41.8359375 , 42.83203125,
       43.828125 , 44.82421875, 45.8203125 , 46.81640625,
       47.8125   , 48.80859375, 49.8046875 , 50.80078125,
       51.796875 , 52.79296875, 53.7890625 , 54.78515625,
       55.78125  , 56.77734375, 57.7734375 , 58.76953125,
       59.765625 , 60.76171875, 61.7578125 , 62.75390625,
       63.75     , 64.74609375, 65.7421875 , 66.73828125,
       67.734375 , 68.73046875, 69.7265625 , 70.72265625,
       71.71875  , 72.71484375, 73.7109375 , 74.70703125,
       75.703125 , 76.69921875, 77.6953125 , 78.69140625,
       79.6875   , 80.68359375, 81.6796875 , 82.67578125,
       83.671875 , 84.66796875, 85.6640625 , 86.66015625,
       87.65625  , 88.65234375, 89.6484375 , 90.64453125,
       91.640625 , 92.63671875, 93.6328125 , 94.62890625,
       95.625    , 96.62109375, 97.6171875 , 98.61328125,
       99.609375 , 100.60546875, 101.6015625 , 102.59765625,
      103.59375  , 104.58984375, 105.5859375 , 106.58203125,
      107.578125 , 108.57421875, 109.5703125 , 110.56640625,
      111.5625   , 112.55859375, 113.5546875 , 114.55078125,
      115.546875 , 116.54296875, 117.5390625 , 118.53515625,
      119.53125  , 120.52734375, 121.5234375 , 122.51953125,
      123.515625 , 124.51171875, 125.5078125 , 126.50390625,
      127.5     , 128.49609375, 129.4921875 , 130.48828125,
      131.484375 , 132.48046875, 133.4765625 , 134.47265625,

```

```

135.46875 , 136.46484375, 137.4609375 , 138.45703125,
139.453125 , 140.44921875, 141.4453125 , 142.44140625,
143.4375 , 144.43359375, 145.4296875 , 146.42578125,
147.421875 , 148.41796875, 149.4140625 , 150.41015625,
151.40625 , 152.40234375, 153.3984375 , 154.39453125,
155.390625 , 156.38671875, 157.3828125 , 158.37890625,
159.375 , 160.37109375, 161.3671875 , 162.36328125,
163.359375 , 164.35546875, 165.3515625 , 166.34765625,
167.34375 , 168.33984375, 169.3359375 , 170.33203125,
171.328125 , 172.32421875, 173.3203125 , 174.31640625,
175.3125 , 176.30859375, 177.3046875 , 178.30078125,
179.296875 , 180.29296875, 181.2890625 , 182.28515625,
183.28125 , 184.27734375, 185.2734375 , 186.26953125,
187.265625 , 188.26171875, 189.2578125 , 190.25390625,
191.25 , 192.24609375, 193.2421875 , 194.23828125,
195.234375 , 196.23046875, 197.2265625 , 198.22265625,
199.21875 , 200.21484375, 201.2109375 , 202.20703125,
203.203125 , 204.19921875, 205.1953125 , 206.19140625,
207.1875 , 208.18359375, 209.1796875 , 210.17578125,
211.171875 , 212.16796875, 213.1640625 , 214.16015625,
215.15625 , 216.15234375, 217.1484375 , 218.14453125,
219.140625 , 220.13671875, 221.1328125 , 222.12890625,
223.125 , 224.12109375, 225.1171875 , 226.11328125,
227.109375 , 228.10546875, 229.1015625 , 230.09765625,
231.09375 , 232.08984375, 233.0859375 , 234.08203125,
235.078125 , 236.07421875, 237.0703125 , 238.06640625,
239.0625 , 240.05859375, 241.0546875 , 242.05078125,
243.046875 , 244.04296875, 245.0390625 , 246.03515625,
247.03125 , 248.02734375, 249.0234375 , 250.01953125,
251.015625 , 252.01171875, 253.0078125 , 254.00390625,
255. ] )

```

*#cdf adalah sum dari nilai hist*

```
cdf = hist.cumsum()
```

```
cdf
```

```

array([ 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 2, 3, 3,
       7, 9, 12, 22, 31, 47, 67, 94, 122,
      161, 244, 328, 456, 630, 873, 1140, 1497, 1958,
      2442, 3054, 3736, 4539, 5373, 6403, 7309, 8122, 8858,
      9514, 10059, 10592, 11157, 11652, 12185, 12744, 13340, 13869,
      14353, 14962, 15487, 16054, 16675, 17433, 18050, 18724, 19695,
      21084, 21850, 22576, 23163, 23945, 24579, 25259, 26247, 27022,

```

```

27770, 28568, 29447, 30234, 31376, 33201, 34019, 35694, 36565,
37475, 38392, 39825, 41775, 42889, 44505, 45641, 46861, 47858,
49490, 50527, 51220, 51934, 52690, 53471, 54003, 54755, 55473,
56253, 56773, 57379, 57862, 58232, 58589, 58914, 59264, 59644,
60078, 60656, 60998, 61351, 61701, 62021, 62481, 62858, 63514,
63911, 64324, 64719, 64925, 65064, 65175, 65262, 65361, 65405,
65442, 65476, 65497, 65508, 65514, 65521, 65525, 65530, 65533,
65534, 65535, 65535, 65535, 65535, 65535, 65535, 65535, 65536,
65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536,
65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536,
65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536,
65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536,
65536, 65536, 65536, 65536], dtype=int64)

```

*#cdf\_m adalah masking nilai 0/ menghilangkan nilai 0*

```

cdf_m = np.ma.masked_equal(cdf,0)

```

```

cdf_m

```

```

masked_array(data=[--, --, --, --, --, --, --, --, --, --, --, --, --, --,
--,
                --, --, --, --, --, --, --, --, --, --, --, --, --, --,
--,
                --, --, --, --, --, --, --, --, --, --, --, --, --, --,
--,
                --, --, --, --, --, --, --, --, --, --, --, --, --, --,
--,
                --, --, --, --, --, --, --, --, --, --, --, --, --, --,
22,
                31, 47, 67, 94, 122, 161, 244, 328, 456, 630, 873,
6403,
                1140, 1497, 1958, 2442, 3054, 3736, 4539, 5373,
                7309, 8122, 8858, 9514, 10059, 10592, 11157, 11652,
16054,
                12185, 12744, 13340, 13869, 14353, 14962, 15487,
                16675, 17433, 18050, 18724, 19695, 21084, 21850,
22576,
                23163, 23945, 24579, 25259, 26247, 27022, 27770,
28568,
                29447, 30234, 31376, 33201, 34019, 35694, 36565,
37475,
                38392, 39825, 41775, 42889, 44505, 45641, 46861,
47858,
                49490, 50527, 51220, 51934, 52690, 53471, 54003,
54755,
                55473, 56253, 56773, 57379, 57862, 58232, 58589,
58914,
                59264, 59644, 60078, 60656, 60998, 61351, 61701,
62021,

```

[illegible]



0.003891109958189642, 0.003891109958189642,  
0.01945554979094821, 0.027237769707327494,  
0.03891109958189642, 0.07782219916379284,  
0.11284218878749962, 0.17509994811853388,  
0.2529221472823267, 0.35798211615344705,  
0.46693319498275704, 0.618686483352153,  
0.9416486098818934, 1.2685018463698232,  
1.7665639210180974, 2.443617053743095,  
3.3891567735831782, 4.428083132419813,  
5.817209387493515, 7.61101107821894,  
9.494308297982727,  
11.875667592394787, 14.529404583880122,  
17.653965880306405, 20.899151585436567,  
24.906994842371898, 28.432340464491713,  
31.595812860499894, 34.45966978972747,  
37.012237922299875, 39.13289284951323,  
41.20685445722831, 43.40533158360546,  
45.33143101290933, 47.40539262062441,  
49.580523087252416, 51.89962462233345,  
53.95802179021577, 55.84131900997955,  
58.211004974517046, 60.25383770256661,  
62.460097048860135, 64.8764763328959,  
67.82593768120365, 70.22675252540665,  
72.84936063722648, 76.62762840662862,  
82.03238013855403, 85.0129703665273,  
87.83791619617298,  
90.1219977416303, 93.1648457289346,  
95.63180944242683,  
98.2777642139958, 102.12218085268715,  
105.13779107028412, 108.04834131900998,  
111.15344706564531, 114.57373271889401,  
117.63603625598925, 122.07968382824183,  
129.18095950193793, 132.36388744773706,  
138.8814966277047, 142.2706534012879,  
145.81156346324045, 149.37971129490035,  
154.9556718649861, 162.54333628345591,  
166.87803277687917, 173.16606646931365,  
177.58636738181707, 182.33352153080844,  
186.2129581591235, 192.56324961088902,  
196.59833063753166, 199.29486983855708,  
202.0731223487045, 205.01480147709586,  
208.05375835444198, 210.12382885219887,  
213.04994354075748, 215.84376049073762,  
218.87882625812554, 220.90220343638416,  
223.2602160710471, 225.13962218085268,  
226.57933286538287, 227.96845912045654,  
229.23306985686818, 230.59495834223458,  
232.07358012634663, 233.76232184820094,  
236.01138340403455, 237.3421430097354,  
238.71570482497634, 240.07759331034273,



```

241.3227484969634, 243.11265907773065,
244.57960753196815, 247.13217566454054,
248.67694631794183, 250.28397473067415,
251.82096316415905, 252.62253181554613,
253.16339609973448, 253.59530930509354,
253.93383587145604, 254.31905575731682,
254.49026459547716, 254.63423566393018,
254.76653340250863, 254.8482467116306,
254.8910489211707, 254.91439558091983,
254.94163335062714, 254.9571977904599,
254.97665334025086, 254.98832667012542,
254.99221778008362, 254.9961088900418,
254.9961088900418, 254.9961088900418,
254.9961088900418, 254.9961088900418,
254.9961088900418, 254.9961088900418, 255.0, 255.0,
255.0, 255.0, 255.0, 255.0, 255.0, 255.0, 255.0,
255.0,
255.0, 255.0, 255.0, 255.0, 255.0, 255.0, 255.0,
255.0, 255.0, 255.0, 255.0, 255.0, 255.0, 255.0,
255.0, 255.0, 255.0, 255.0, 255.0, 255.0, 255.0,
255.0, 255.0, 255.0, 255.0, 255.0, 255.0, 255.0,
255.0],
mask=[ True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  True,
      True,  True,  True,  True,  True,  True,  False,
      False, False, False, False, False, False, False,
      False, False, False, False, False, False, False,

```

```
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False,
False], fill_value=0)

cdf_m.min()
0.0
cdf_m.max()
255.0
```

*#round atau pembulatan nilai terdekat menghasilkan nilai histogram yang baru*

```
cdf = np.ma.filled(cdf_m,0).astype('uint8')
```

```
cdf
```

```
array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
 0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
 0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
 0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
 0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
 0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
 0,
        0,  1,  1,  2,  3,  4,  5,  7,  9, 11, 14, 17,
20,
       24, 28, 31, 34, 37, 39, 41, 43, 45, 47, 49, 51,
53,
       55, 58, 60, 62, 64, 67, 70, 72, 76, 82, 85, 87,
90,
       93, 95, 98, 102, 105, 108, 111, 114, 117, 122, 129, 132,
138,
      142, 145, 149, 154, 162, 166, 173, 177, 182, 186, 192, 196,
199,
      202, 205, 208, 210, 213, 215, 218, 220, 223, 225, 226, 227,
229,
      230, 232, 233, 236, 237, 238, 240, 241, 243, 244, 247, 248,
250,
      251, 252, 253, 253, 253, 254, 254, 254, 254, 254, 254, 254,
254,
      254, 254, 254, 254, 254, 254, 254, 254, 254, 254, 254, 255,
255,
      255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255,
      255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255,
      255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255,
      255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255,
      255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255], dtype=uint8)
```

*# nilai histogram yang baru implementasi ke img*

```
img2 = cdf[img]
```

*#melakukan normalisasi nilai img histogram (menggantikan nilai image histogram yang lama ke yang baru)*

[illegible]

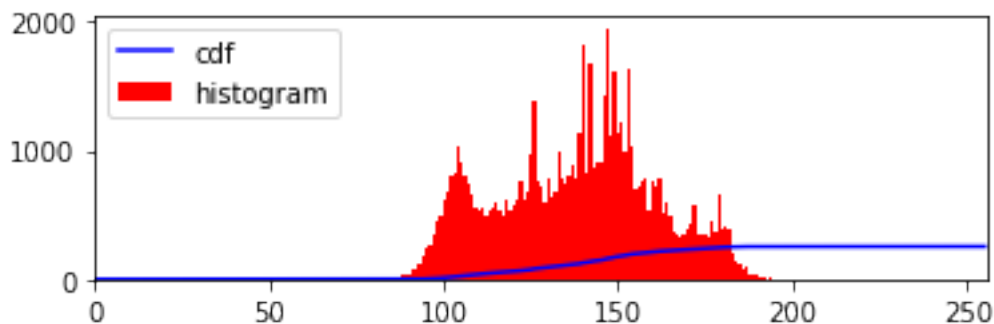
```

1942.35294118, 1942.35294118, 1942.35294118, 1942.35294118,
1942.35294118, 1942.35294118, 1942.35294118, 1942.35294118,
1942.35294118, 1942.35294118, 1942.35294118, 1942.35294118,
1942.35294118, 1942.35294118, 1942.35294118, 1942.35294118,
1942.35294118, 1942.35294118, 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.,
1950., 1950., 1950., 1950.
]
```

*#menampilkan hasil plot img hist yang belum histogram equlizer*

```

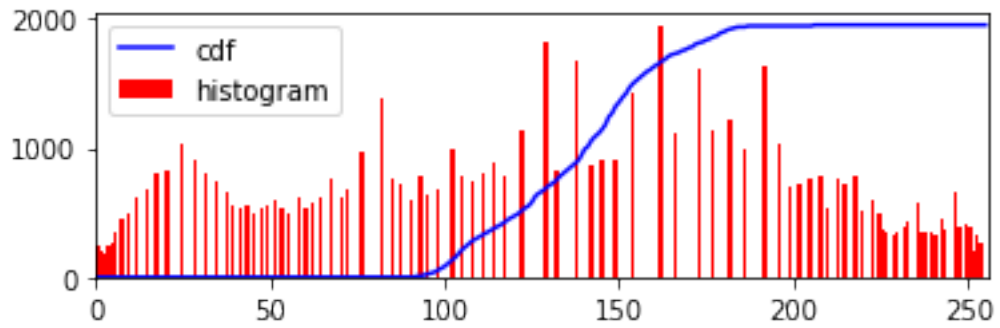
plt.subplot(2, 1, 1)
plt.plot(cdf, color = 'b')
plt.hist(img.flatten(),256,[0,255], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()
```



*#menampilkan hasil plot img hist yang sudah dihistogram equlizer*

```

plt.subplot(2, 1, 2)
plt.plot(cdf_normalized, color = 'b')
plt.hist(img2.flatten(),256,[0,255], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()
```



*#menampilkan hasil img sebelum dan sesudah di histogram equlizer*

```
cv.imshow('img1',img)  
cv.imshow('img2',img2)  
cv.waitKey(0)  
cv.destroyAllWindows()
```

## Menggunakan Function Image histogram Equalization openCV

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

#membaca image
img = cv.imread('low.jpg',0)
#melakukan image histogram equlizer dengan Function OpenCV
equalizeHist
equ = cv.equalizeHist(img)
#Menggabungkan 2 gambar sebelum dan sesudah dari image histogram
equlizer
res = np.hstack((img,equ)) #stacking images side-by-side

#save output proses diatas dengan nama res.jpg
cv.imwrite('res.jpg',res)
```

True