

3. Гістограми

1. Вирівнювання гістограми

КОД:

```
import cv2

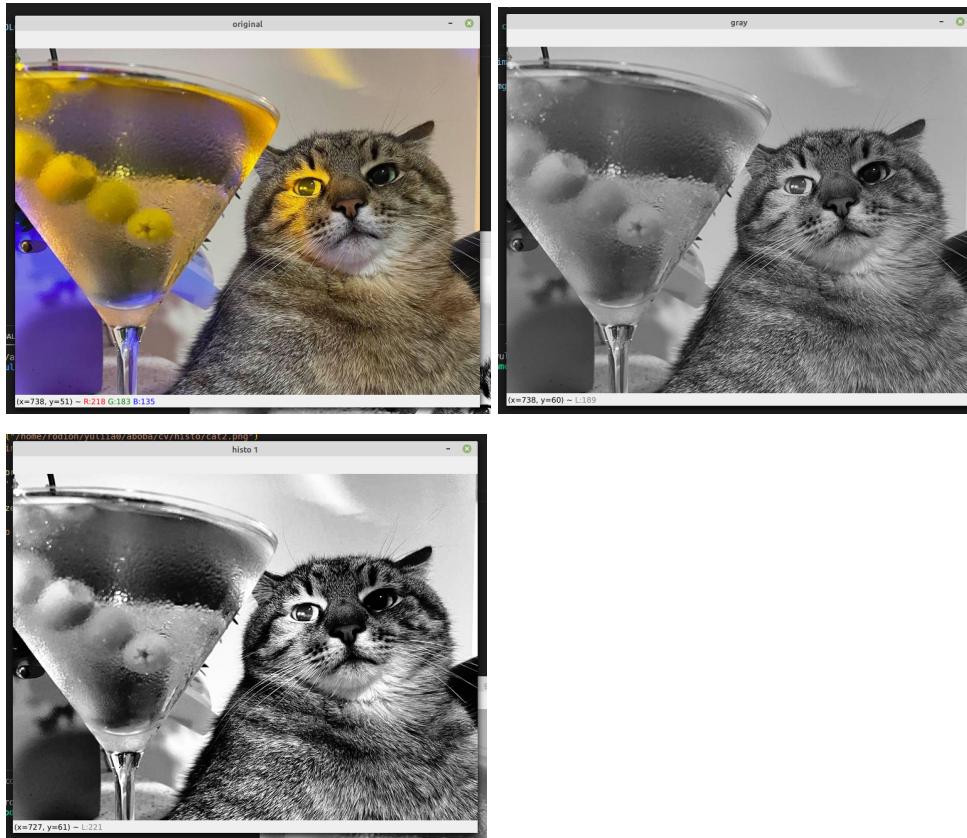
img = cv2.imread("/home/rodion/yuliia0/aboba/cv/histo/cat2.png")
cv2.imshow('original', img)

img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('gray', img)

img = cv2.equalizeHist(img)

cv2.imshow('histo 1', img)
cv2.waitKey(0)
```

результат:



2. Розрахунок гістограми

КОД:

```
import cv2
import numpy as np

img = cv2.imread("/home/rodion/yuliia0/aboba/cv/histo/cat2.png")

bgr= cv2.split(img)

b = cv2.calcHist(bgr, [0], None, [256], (0, 256), accumulate=False)
g = cv2.calcHist(bgr, [1], None, [256], (0, 256), accumulate=False)
r = cv2.calcHist(bgr, [2], None, [256], (0, 256), accumulate=False)

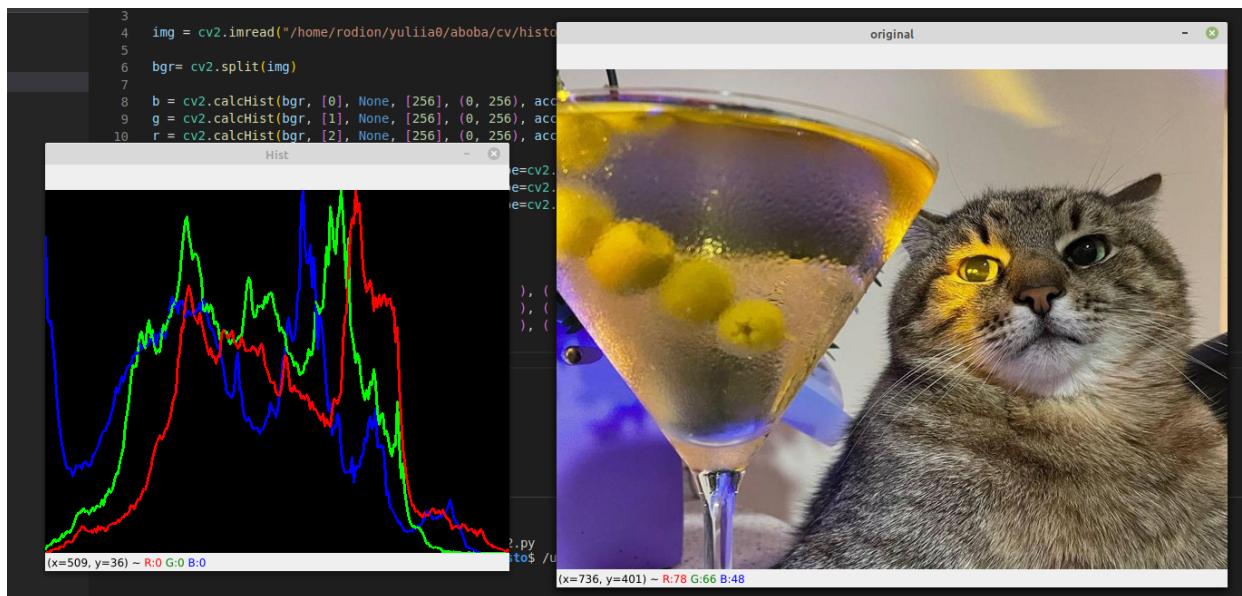
cv2.normalize(b, b, alpha=0, beta=400, norm_type=cv2.NORM_MINMAX)
cv2.normalize(g, g, alpha=0, beta=400, norm_type=cv2.NORM_MINMAX)
cv2.normalize(r, r, alpha=0, beta=400, norm_type=cv2.NORM_MINMAX)

hist = np.zeros((400, 512, 3), dtype=np.uint8)

for i in range(1, 256):
    cv2.line(hist, ( 2*(i-1), 400 - int(b[i-1]) ), ( 2*(i), 400 - int(b[i]) ), ( 255, 0, 0), thickness=2)
    cv2.line(hist, ( 2*(i-1), 400 - int(g[i-1]) ), ( 2*(i), 400 - int(g[i]) ), ( 0, 255, 0), thickness=2)
    cv2.line(hist, ( 2*(i-1), 400 - int(r[i-1]) ), ( 2*(i), 400 - int(r[i]) ), ( 0, 0, 255), thickness=2)

cv2.imshow('original', img)
cv2.imshow('Hist', hist)
cv2.waitKey(0)
```

результат:



3. Порівняння гістограм

КОД:

```
import cv2
import numpy as np

base = cv2.imread('/home/rodion/yuliia0/aboba/cv/histo/cat2.png')
test1 = cv2.imread('/home/rodion/yuliia0/aboba/cv/histo/cat3.jpg')
test2 = cv2.imread('/home/rodion/yuliia0/aboba/cv/histo/cat4.jpg')

HSV_base = cv2.cvtColor(base, cv2.COLOR_BGR2HSV)
HSV_test1 = cv2.cvtColor(test1, cv2.COLOR_BGR2HSV)
HSV_test2 = cv2.cvtColor(test2, cv2.COLOR_BGR2HSV)

HSV_half_down = HSV_base[HSV_base.shape[0]//2:,:,:]

histSize = [50, 60]
ranges = [0, 180] + [0, 256]
channels = [0, 1]

hist_base = cv2.calcHist([HSV_base], channels, None, histSize, ranges,
accumulate=False)
```

```

cv2.normalize(hist_base, hist_base, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX)

hist_half_down = cv2.calcHist([hsv_half_down], channels, None,
histSize, ranges, accumulate=False)
cv2.normalize(hist_half_down, hist_half_down, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX)

hist_test1 = cv2.calcHist([hsv_test1], channels, None, histSize,
ranges, accumulate=False)
cv2.normalize(hist_test1, hist_test1, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX)

hist_test2 = cv2.calcHist([hsv_test2], channels, None, histSize,
ranges, accumulate=False)
cv2.normalize(hist_test2, hist_test2, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX)

for compare_method in range(4):
    base_base = cv2.compareHist(hist_base, hist_base, compare_method)
    base_half = cv2.compareHist(hist_base, hist_half_down, compare_method)
    base_test1 = cv2.compareHist(hist_base, hist_test1, compare_method)
    base_test2 = cv2.compareHist(hist_base, hist_test2, compare_method)

    print('Method:', compare_method, 'Perfect, Base-Half, Base-Test(1),',
          'Base-Test(2) :', \
          base_base, '/', base_half, '/', base_test1, '/', base_test2)

```

результат:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
/usr/bin/python3 /home/rodion/yulilia0/aboba/cv/histo/hi3.py
• rodion@rodion-System-Product-Name:~/yulilia0/aboba/cv/histo$ /usr/bin/python3 /home/rodion/yulilia0/aboba/cv/histo/hi3.py
Method: 0 Perfect, Base-Half, Base-Test(1), Base-Test(2) : 1.0 / 0.7091052319213833 / 0.5294127517522248 / 0.4141867379560196
Method: 1 Perfect, Base-Half, Base-Test(1), Base-Test(2) : 0.0 / 80.63692024093037 / 233.87211738927692 / 339.63983185859377
Method: 2 Perfect, Base-Half, Base-Test(1), Base-Test(2) : 23.882153554997785 / 19.861272195932543 / 12.516323589268723 / 8.818261532083852
Method: 3 Perfect, Base-Half, Base-Test(1), Base-Test(2) : 0.0 / 0.2671657269813732 / 0.5246719370941044 / 0.537582940145528
○ rodion@rodion-System-Product-Name:~/yulilia0/aboba/cv/histo$
```

4. Зворотна проекція

КОД:

```

import cv2
import numpy as np

img = cv2.imread("/home/rodion/yulilia0/aboba/cv/histo/cat2.png")

```

```

cv2.imshow("original", img)
img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

hue = np.empty(img.shape, img.dtype)
cv2.mixChannels([img], [hue], (0, 0))

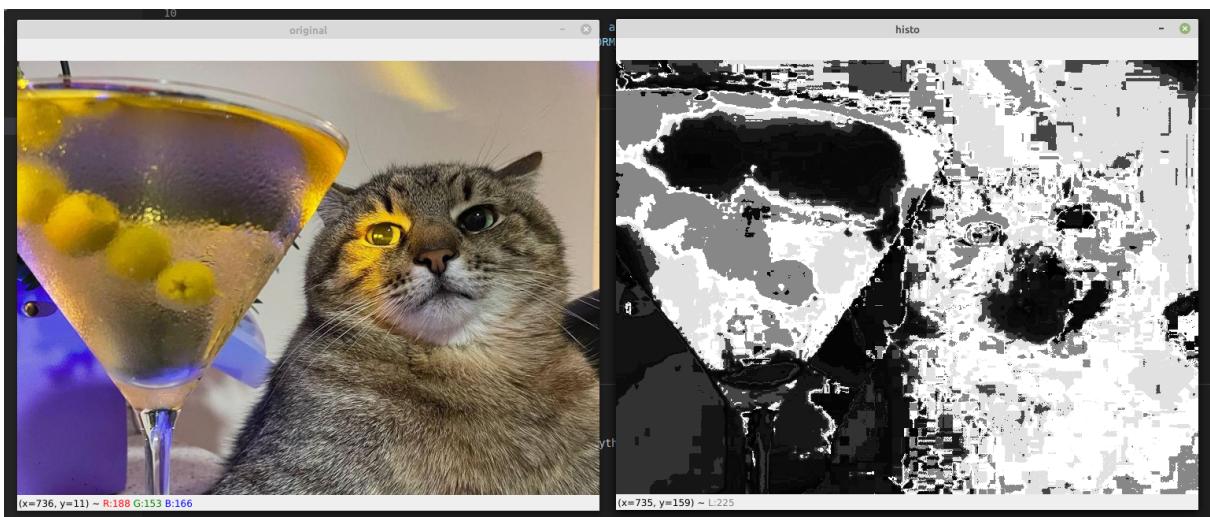
hist = cv2.calcHist([hue], [0], None, [max(50, 2)], [0, 180],
accumulate=False)
cv2.normalize(hist, hist, alpha=0, beta=255, norm_type=cv2.NORM_MINMAX)

b= cv2.calcBackProject([hue], [0], hist, [0, 180], scale=1)

cv2.imshow('histo', b)
cv2.waitKey(0)

```

результат:



5. Відповідність шаблону

КОД:

```

import cv2
import numpy as np

img = cv2.imread('/home/rodion/yuliia0/aboba/cv/histo/cat2.png')
templ = cv2.imread('/home/rodion/yuliia0/aboba/cv/histo/templ.jpg')
img_display = img.copy()
cv2.imshow("biba", img)

```

```

cv2.imshow('templ', templ)

mask=cv2.cvtColor(templ, cv2.COLOR_BGR2GRAY)
ker=np.array([[-1,-1,-1],[-1,9,-1],[-1,-1,-1]])
mask=cv2.filter2D(mask,-1,ker)
cv2.imshow('mask', mask)

result=cv2.matchTemplate(img,templ, cv2.TM_SQDIFF)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, minLoc, (minLoc[0] + templ.shape[0],
minLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.rectangle(result, minLoc, (minLoc[0] + templ.shape[0], minLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('1', result)

result=cv2.matchTemplate(img,templ, cv2.TM_SQDIFF_NORMED)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, minLoc, (minLoc[0] + templ.shape[0],
minLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.rectangle(result, minLoc, (minLoc[0] + templ.shape[0], minLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('2', result)

result=cv2.matchTemplate(img,templ, cv2.TM_SQDIFF_NORMED, mask)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, minLoc, (minLoc[0] + templ.shape[0],
minLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.rectangle(result, minLoc, (minLoc[0] + templ.shape[0], minLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('2+mask', result)

result=cv2.matchTemplate(img,templ, cv2.TM_CCORR)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, minLoc, (minLoc[0] + templ.shape[0],
minLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.rectangle(result, minLoc, (minLoc[0] + templ.shape[0], minLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('3', result)

result=cv2.matchTemplate(img,templ, cv2.TM_CCORR_NORMED)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, maxLoc, (maxLoc[0] + templ.shape[0],
maxLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )

```

```

cv2.rectangle(result, maxLoc, (maxLoc[0] + templ.shape[0], maxLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('4', result)

result=cv2.matchTemplate(img,templ, cv2.TM_CCOEFF)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, maxLoc, (maxLoc[0] + templ.shape[0],
maxLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.rectangle(result, maxLoc, (maxLoc[0] + templ.shape[0], maxLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('5', result)

result=cv2.matchTemplate(img,templ, cv2.TM_CCOEFF_NORMED)
_minVal, _maxVal, minLoc, maxLoc = cv2.minMaxLoc(result, None)
cv2.rectangle(img_display, maxLoc, (maxLoc[0] + templ.shape[0],
maxLoc[1] + templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.rectangle(result, maxLoc, (maxLoc[0] + templ.shape[0], maxLoc[1] +
templ.shape[1]), (255,255,255), 2, 8, 0 )
cv2.imshow('6', result)

cv2.waitKey(0)

```

результат:

