

# OpenCV. РОБОТА із ЗОБРАЖЕННЯМИ

Файл: CV\_Image\_12\_001

Дескриптор, детектор: HAAR. Viola Jones Classifier

[OpenCV Docs](#)

[HOG cv::HOGDescriptor Struct Reference](#)

Також дивись [Learn OpenCV](#)

```
# Завантаження бібліотек
import numpy as np
import sys
import skimage.io as io
import matplotlib.pyplot as plt
plt.rcParams['font.size'] = 5
```

```
# Імпорт CV
import cv2 as cv
```

## Завантаження зображення

```
## %%script false
# Завантаження зображення
path = './IMAGES/'
# filename = 'Lena_Gray.jpg'
# filename = 'Lenna.png'
# filename = 'per00057.png'

#filename = 'Lenna.png'
filename = 'Faces_two_.jpg'
#filename = 'Faces_many_.jpg'
#filename = 'Run_image_2.jpg'
test_img_ = io.imread(path+filename)
# Визначення структури та розміру зображення
print('---- TEST IMAGE ----')
print('IMAGE SHAPE', test_img_.shape, 'IMAGE SIZE', test_img_.size)
irows_num = test_img_.shape[0] # кількість рядків
iclms_num = test_img_.shape[1] # кількість колонок
print('ROWS NUMBER', irows_num, 'CLMS NUMBER', iclms_num)
fig, ax = plt.subplots(figsize=(4, 4))
plt.imshow(test_img_)
plt.title("Тестове зображення")
plt.show()
```

```
---- TEST IMAGE ----
IMAGE SHAPE (499, 750, 3) IMAGE SIZE 1122750
ROWS NUMBER 499 CLMS NUMBER 750
```



```
### Завантаження за допомогою CV
test_img = cv.imread(path+filename)
if test_img is None:
    sys.exit("Could not read the image.")

test_img_gray = cv.cvtColor(test_img , cv.COLOR_BGR2GRAY)
#test_img_gray = cv.resize(test_img_gray, dsize = (0, 0), fx = 0.5, fy = 0.5) #
якщо є потреба ресайзу
#test_img_gray = cv.equalizeHist(test_img_gray) # якщо є потреба гістограмного
покращення
test_img_.shape[1] # кількість колонок
print('ROWS NUMBER', test_img_gray.shape[0], 'CLMS NUMBER',
test_img_gray.shape[1])

# wait for a key press and close the window
while True:
    cv.imshow('Display window', test_img_gray)
    if cv.waitKey(1) == ord('q'):
        break
cv.destroyAllWindows()
```

ROWS NUMBER 499 CLMS NUMBER 750



## Ознаки HAAR та Viola Jones класифікатори в OpenCV

[CV CascadeClassifier](#)

[PreTrain haarcascades](#)

```
# Завантаження пре_трейн класифікаторів
# ! Файли повинні бути в тій самій директорії
# Завантаження попередньо навченого класифікатора для облич
#face_cascade_name = 'haarcascade_frontalcatface.xml'
face_cascade_name = 'haarcascade_frontalcatface_extended.xml'
# Завантаження попередньо навченого класифікатора для очей
eyes_cascade_name = 'haarcascade_eye.xml'

face_cascade = cv.CascadeClassifier()
eyes_cascade = cv.CascadeClassifier()

#-- перевірка присутності каскадів
if not face_cascade.load(cv.samples.findFile(face_cascade_name)):
    print('--(!)Error loading face cascade')
    exit(0)

if not eyes_cascade.load(cv.samples.findFile(eyes_cascade_name)):
    print('--(!)Error loading eyes cascade')
    exit(0)
```

## Детектування облич та очей

```
#-- Detect faces
faces = face_cascade.detectMultiScale(test_img_gray, scaleFactor=1.01,
minNeighbors = 5, minSize= (50, 50))
print('FACES', faces)
```

```

for (x,y,w,h) in faces:
    test_img = cv.rectangle(test_img,(x,y),(x+w,y+h),(255,0,0),2)
    faceROI = test_img_gray[y:y+h,x:x+w]
    #-- In each face, detect eyes
    eyes = eyes_cascade.detectMultiScale(faceROI)
    print('EYES', eyes)
    for (x2,y2,w2,h2) in eyes:
        eye_center = (x + x2 + w2//2, y + y2 + h2//2)
        radius = int(round((w2 + h2)*0.25))
        test_img = cv.circle(test_img, eye_center, radius, (0, 0, 255), 2)

```

```

FACES [[524 111 54 54]
       [527 117 67 67]
       [592 333 106 106]
       [144 117 175 175]]
EYES ()
EYES ()
EYES ()
EYES [[ 68 85 24 24]
      [104 52 37 37]
      [ 53 103 60 60]]

```

```

while True:
    cv.imshow('Capture - Face detection', test_img)
    if cv.waitKey(1) == ord('q'):
        break

cv.destroyAllWindows()

```

