

# РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image\_09\_003

## Детектори кутів. FAST алгоритм

### [FAST Algorithm for Corner Detection](#)

```
# ВСЯКИЕ РАЗНЫЕ ПАКЕТЫ
import numpy as np
import matplotlib.pyplot as plt
import skimage.io as io
```

### Тестовго зображення

```
%%script false
## Зарузка файла
test_img = np.zeros ((50, 50, 3), dtype=np.uint8)

## Визначення стрктури та розміру зображення
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)

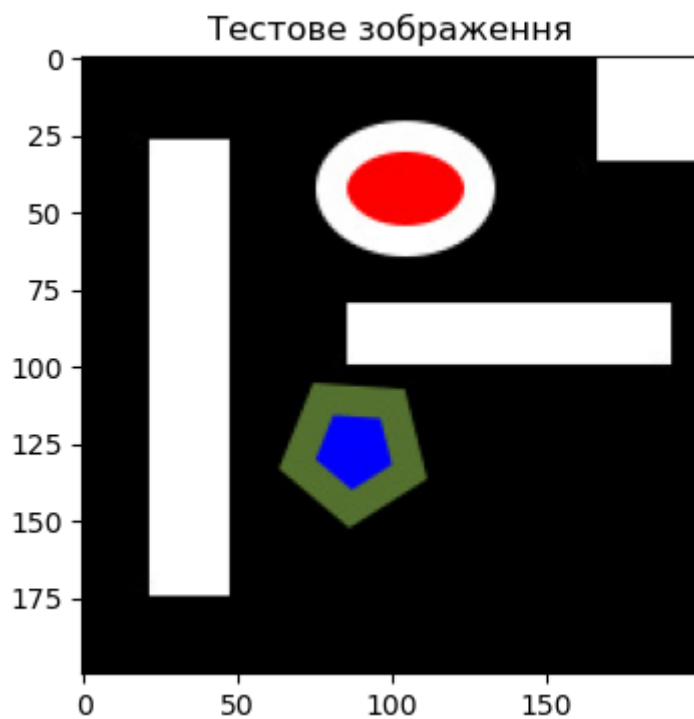
for i in range (20,30,1):
    for j in range (20,30,1):
        test_img[i, j, :] = [200,200,200]
```

Couldn't find program: 'false'

```
## %%script false
# Завантаження зображення
path = './IMAGES/'
# filename = 'Im_02_01.jpg'
filename = 'Morfo_C1_1.jpg'
# filename = 'Lenna.png'
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 (200, 200, 3) IMAGE SIZE 120000
ROWS NUMBER 200 CLMS NUMBER 200
```



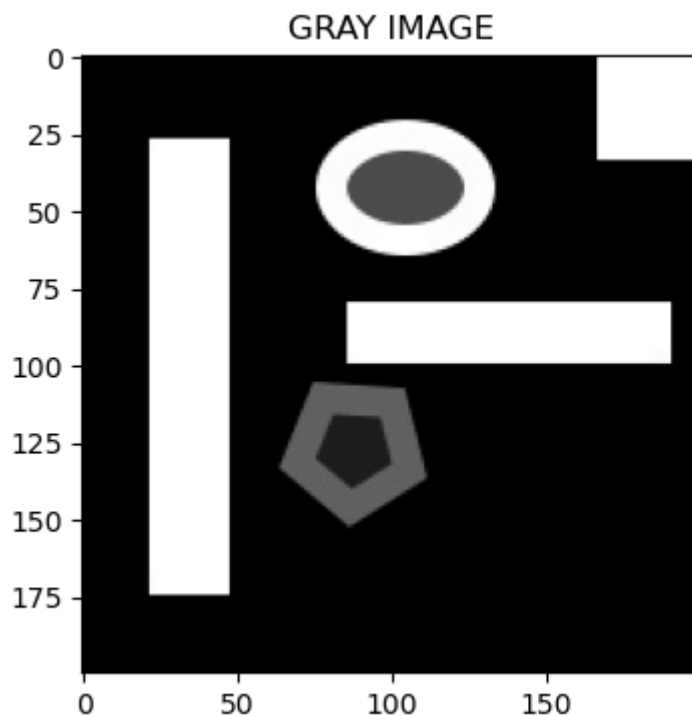
```
test_image_gray = np.zeros ( (irows_num, iclms_num, 3), dtype=np.uint8)
print ('Gray_Im SAPE', test_image_gray.shape, 'Gray_Im SIZE', irows_num *
iclms_num)

for i in range (irows_num):
    for j in range (iclms_num):
        # Gray image
        test_image_gray [i, j, :] = 0.299*test_img [i, j, 0]+0.587*test_img [i, j,
1]+0.114*test_img [ i, j, 2]

## L_Min, L_Max
L_min = np.amin(test_image_gray[:, :,0])
L_max = np.amax(test_image_gray[:, :,0])
print ('Intensity MIN = ', L_min , 'Intensity MAX = ', L_max)
```

```
## Серое на экран
fig, ax = plt.subplots(figsize=(4, 4))
plt.title('GRAY IMAGE')
plt.imshow(test_image_gray)
plt.show()
```

```
Gray_Im SAPE (200, 200, 3) Gray_Im SIZE 40000
Intensity MIN = 0 Intensity MAX = 255
```



## Памка

```
def fast_corner_detector(image, threshold=50):
    # Define a circular pattern of 16 pixels
    circle_pattern = np.array([
        [0, 3], [1, 3], [2, 2], [3, 1], [3, 0], [3, -1], [2, -2], [1, -3],
        [0, -3], [-1, -3], [-2, -2], [-3, -1], [-3, 0], [-3, 1], [-2, 2], [-1, 3]
    ])

    corners = []
    for i in range(3, image.shape[0] - 3):
        for j in range(3, image.shape[1] - 3):
            intensity = image[i, j]
            darker = 0
            brighter = 0
            for di, dj in circle_pattern:
                if image[i + di, j + dj] < intensity - threshold:
                    darker += 1
                if image[i + di, j + dj] > intensity + threshold:
                    brighter += 1
            if darker >= 9 or brighter >= 9:
```

```
        corners.append((i, j))
        break
    return corners
```

```
corners = fast_corner_detector(test_image_gray[:, :, 0], 50)
```

```
print("Detected corners:", len(corners))
for corner in corners:
    print (corner)
```

Detected corners: 98

```
(27, 22)
(27, 23)
(27, 24)
(27, 45)
(27, 46)
(27, 47)
(28, 22)
(28, 23)
(28, 46)
(28, 47)
(29, 22)
(29, 47)
(31, 167)
(32, 167)
(32, 168)
(33, 167)
(33, 168)
(33, 169)
(37, 132)
(40, 86)
(40, 123)
(41, 86)
(41, 123)
(42, 86)
(42, 123)
(42, 133)
(43, 86)
(43, 123)
(44, 86)
(44, 123)
(45, 86)
(45, 123)
(80, 86)
(80, 87)
(80, 88)
(80, 188)
(80, 189)
(80, 190)
(81, 86)
(81, 87)
(81, 189)
(81, 190)
(82, 86)
```

(82, 190)  
(97, 86)  
(97, 190)  
(98, 86)  
(98, 87)  
(98, 189)  
(98, 190)  
(99, 86)  
(99, 87)  
(99, 88)  
(99, 188)  
(99, 189)  
(99, 190)  
(106, 75)  
(106, 76)  
(107, 75)  
(108, 102)  
(108, 103)  
(108, 104)  
(109, 104)  
(117, 81)  
(117, 83)  
(118, 96)  
(130, 76)  
(130, 77)  
(131, 99)  
(132, 65)  
(132, 99)  
(133, 64)  
(133, 65)  
(134, 65)  
(135, 111)  
(136, 110)  
(136, 111)  
(137, 110)  
(139, 87)  
(139, 88)  
(151, 85)  
(151, 86)  
(151, 87)  
(151, 88)  
(152, 86)  
(152, 87)  
(172, 22)  
(172, 47)  
(173, 22)  
(173, 23)  
(173, 46)  
(173, 47)  
(174, 22)  
(174, 23)  
(174, 24)  
(174, 45)  
(174, 46)  
(174, 47)

## Обираємо BOX

```
test_im_points = test_image_gray.copy()
# print (test_im_points.shape)

for corner in corners:
    # print (corner)
    test_im_points [corner[0], corner[1], :] = [255,0,0]

fig, ax = plt.subplots(figsize=(4, 4))
plt.imshow(test_im_points)
plt.title("Points of interes")
plt.show()
```

