

# РОБОТА із ЗОБРАЖЕННЯМИ

## Файл: Image\_8\_001

### Частина 1. Завдання порівняння зображень

Примітка. В наступних прикладах зображення, що порівнюються, мають однаковий розмір

```
%matplotlib inline
```

```
import numpy as np
import matplotlib.pyplot as plt
import skimage.io as io
from skimage.color import rgb2gray
from scipy import ndimage
plt.rcParams['font.size'] = 10
```

## ВИЗНАЧЕННЯ МЕТРИК

Функції метрик приймають два параметри: два ахроматичних зображення однакового шейпу

- im1 : TYPE numpy array.  
DESCRIPTION: gray image
- im2 : TYPE numpy array.  
DESCRIPTION: gray image

Функції повертають

- res: TYPE float. DESCRIPTION: обчислена метрика

```
# RSSD Сума квадратів різниць
def ssd(im1, im2):
    r = im1.shape[0]
    c = im1.shape[1]

    s = 0.0
    for i in range(r):
        for j in range(c):
            s += (im1[i, j]-im2[i, j])*(im1[i, j]-im2[i, j])
    s = 1.0 - s/(r*c)
    return s
```

```
# Середньоквадратична похибка
def sqerror(im1, im2):
    r = im1.shape[0]
    c = im1.shape[1]
    s = 0.0
    for i in range(r):
        for j in range(c):
            s += (im1[i, j]-im2[i, j])*(im1[i, j]-im2[i, j])
    s = 1.0 - np.sqrt(s/(r*c))
    return s
```

```
# Нормована кореляція
def cor(im1, im2):
    r = im1.shape[0]
    c = im1.shape[1]

    res = 0.000001
    im1_sqr = 0.000001
    im2_sqr = 0.000001
    for i in range(r):
        for j in range(c):
            res += im1[i, j]*im2[i, j]
            im1_sqr += im1[i, j]*im1[i, j]
            im2_sqr += im2[i, j]*im2[i, j]
    #print('--->', res, im1_sqr, im2_sqr)
    res = res / (np.sqrt(im1_sqr) * np.sqrt(im2_sqr))
    return res
```

```
# Метрика Мінковського
def mink(im1, im2):
    r = im1.shape[0]
    c = im1.shape[1]

    res = 0.0

    for i in range(r):
        for j in range(c):
            res += np.abs(im1[i, j]-im2[i, j])
    res = 1.0 - res / (r * c)
    return res
```

```
# Метрика Хаусдорфа
def hausdorf(im1, im2):
    r = im1.shape[0]
    c = im1.shape[1]

    res = 0.0

    for i in range(r):
        for j in range(c):
            dif = np.abs(im1[i, j]-im2[i, j])
```

```
        if dif > res:
            res = dif
    res = 1.0 - res
    return res
```

```
# Завантаження 1-го зображення
path = './IMAGES/'
#filename1 = 'Face_0_300_X_400_B.jpg'
filename1 = 'Face_1_300_X_400.jpg'
img_1 = io.imread(path+filename1)
# Визначення структури та розміру зображення
print('---- IMAGE 1 ----')
print('IMAGE SHAPE', img_1.shape, 'IMAGE SIZE', img_1.size)
irows_num_1 = img_1.shape[0] # кількість рядків
iclms_num_1 = img_1.shape[1] # кількість колонок
print('ROWS NUMBER', irows_num_1, 'CLMS NUMBER', iclms_num_1)
```

```
---- IMAGE 1 ----
IMAGE SHAPE (400, 300, 3) IMAGE SIZE 360000
ROWS NUMBER 400 CLMS NUMBER 300
```

```
# Завантаження 2-го зображення
path = './IMAGES/'
# filename2 = 'Face_0_300_X_400_B.jpg'
# filename2 = 'Face_0_300_X_400_W.jpg'
# filename2 = 'Face_1_300_X_400.jpg'
# filename2 = 'Face_1_300_X_400_005.jpg'
# filename2 = 'Face_1_300_X_400_180.jpg'
filename2 = 'Face_2_300_X_400.jpg'
# filename2 = 'Face_3_300_X_400.jpg'
img_2 = io.imread(path+filename2)

img_1 = io.imread(path_1+filename1)
# Визначення структури та розміру зображення
print('---- IMAGE 2 ----')
print('IMAGE SHAPE', img_2.shape, 'IMAGE SIZE', img_2.size)
irows_num_2 = img_2.shape[0] # кількість рядків
iclms_num_2 = img_2.shape[1] # кількість колонок
print('ROWS NUMBER', irows_num_2, 'CLMS NUMBER', iclms_num_2)
```

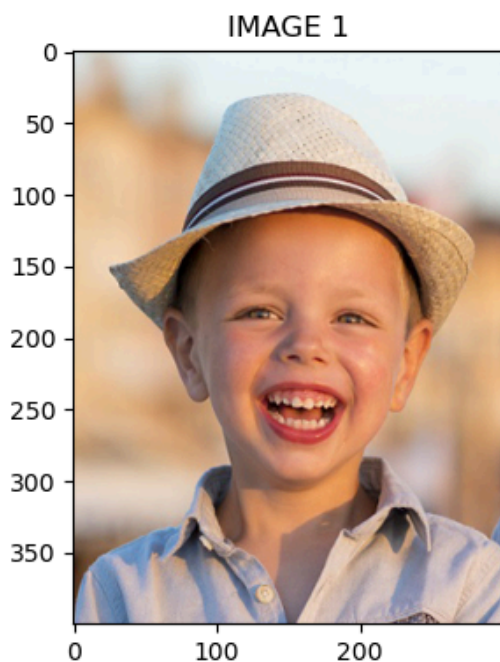
```
---- IMAGE 2 ----
IMAGE SHAPE (400, 300, 3) IMAGE SIZE 360000
ROWS NUMBER 400 CLMS NUMBER 300
```

```

fig, axes = plt.subplots(1, 2, figsize=(8, 4))
ax = axes.ravel()
ax[0].imshow(img_1)
ax[0].set_title("IMAGE 1")
ax[1].imshow(img_2)
ax[1].set_title("IMAGE 2")

fig.tight_layout()
plt.show()

```



```

# Перетворення до сірого
img_1_gray = rgb2gray(img_1)
img_2_gray = rgb2gray(img_2)
print('IMAGE 1 GRAY SHAPE', img_1_gray.shape)
print('IMAGE 1 GRAY PIXEL 0', img_1_gray[0, 0])
print('IMAGE 2 GRAY SHAPE', img_2_gray.shape)
print('IMAGE 2 GRAY PIXEL 0', img_2_gray[0, 0])

```

```

IMAGE 1 GRAY SHAPE (400, 300)
IMAGE 1 GRAY PIXEL 0 0.9020211764705882
IMAGE 2 GRAY SHAPE (400, 300)
IMAGE 2 GRAY PIXEL 0 0.9566552941176472

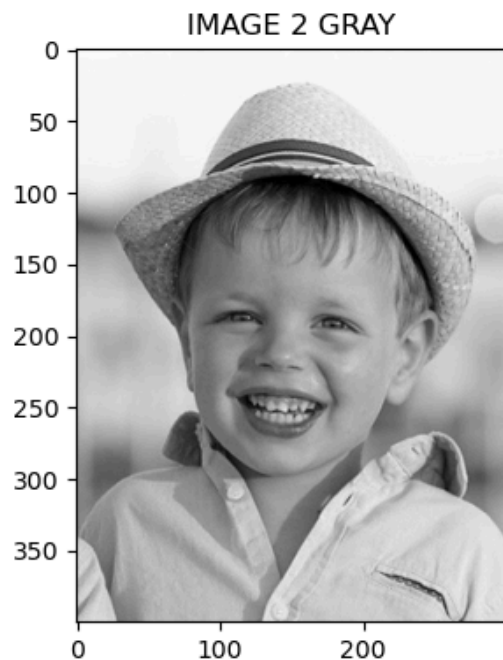
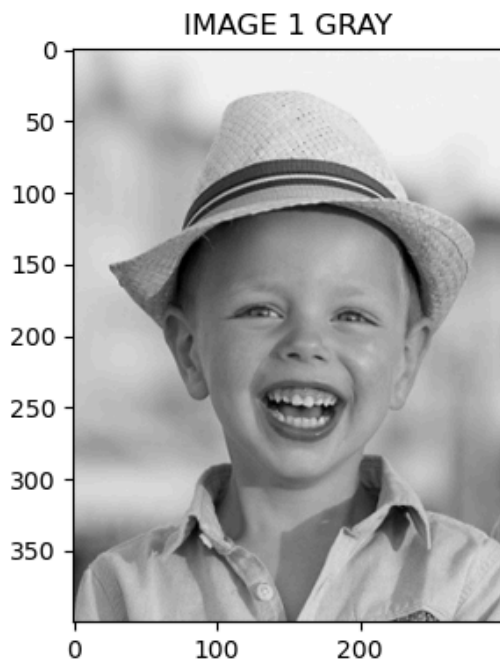
```

```

fig, axes = plt.subplots(1, 2, figsize=(8, 4))
ax = axes.ravel()
ax[0].imshow(img_1_gray, cmap=plt.cm.gray)
ax[0].set_title("IMAGE 1 GRAY")
ax[1].imshow(img_2_gray, cmap=plt.cm.gray)
ax[1].set_title("IMAGE 2 GRAY")

fig.tight_layout()
plt.show()

```



## Обчислення метрик

```

cor12 = cor(img_1_gray, img_2_gray)
ssd12 = ssd(img_1_gray, img_2_gray)
sqer12 = sqerror(img_1_gray, img_2_gray)
mink12 = mink(img_1_gray, img_2_gray)
haus12 = hausdorf(img_1_gray, img_2_gray)

print('COR   =', cor12)
print('SSD   =', ssd12)
print('ERROR =', sqer12)
print('MINK   =', mink12)
print('Hausd  =', haus12)

```

```

COR   = 0.9600559477382871
SSD   = 0.9614783351651769
ERROR = 0.8037306319498045
MINK   = 0.8612503971209139
Hausd  = 0.022494901960784297

```

