РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image_02_003

Геометричні перетворення. Масштабування. Обертання. Код Родштейна

```
## Завантаження пакетів
import numpy as np
import matplotlib.pyplot as plt
import skimage.io as io
from skimage.io import imread
```

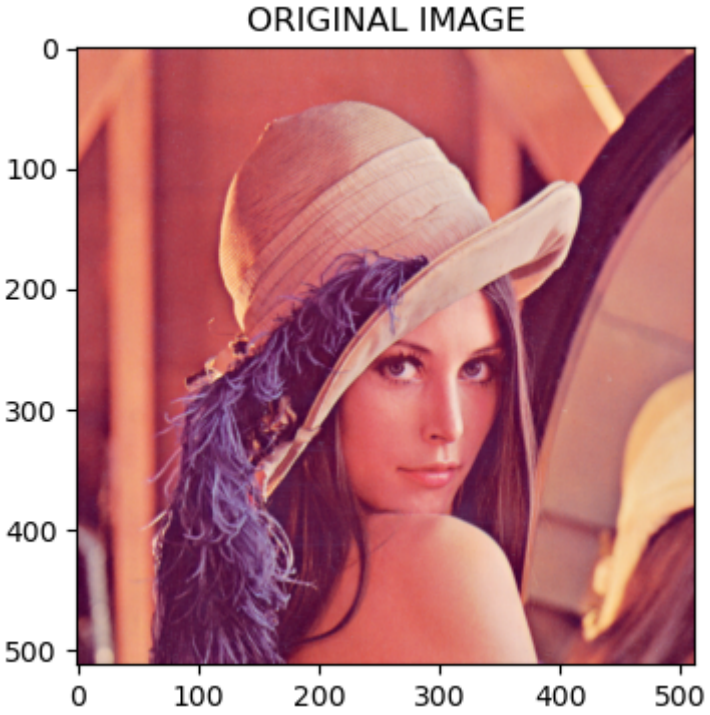
Використовуємо алгоритм БРЕЗЕНХЕМА

```
## ------------------------------------------
## Алгоритм Брезенхема
## Перший октант. !!! Кут менш 45 градусів
## ------------------------------------------

def brez(p1,p2):
    a = p2[0] - p1[0]
    b = p2[1] - p1[1]
    e = 2*b -a
    delta_es = 2*b
    delta_ed = 2*b-2*a
    #print (a,b, e,delta_es,delta_ed)

    rotsht_cod = np.zeros (a , dtype=np.uint8)
    #print (rotsht_cod)
    x = 0 ; y = 0
    while (x < a):
        #print(x,y)
        if e > 0 :
            rotsht_cod[x] = 1
            x += 1; y += 1;
            e += delta_ed
        else :
            x += 1
            e += delta_es
    #print (rotsht_cod)
    return rotsht_cod

#Point_1 = np.array ([0,  0])
#Point_2 = np.array ([10, 5])
# print(brez(Point_1,Point_2))
```

Завантаження зображення

```
## Завантаження файлу зображення
path = './IMAGES/'
filename = 'Lenna.png'
test_im = io.imread(path + filename)
## Визначення стркутури та розміру зображення
print ('IMAGE SHAPE', test_im.shape, 'IMAGE SIZE', test_im.size)
rows_num = test_im.shape[0] ## кількість рядків
clms_num = test_im.shape[1] ## кількість колонок
pix_num = rows_num*clms_num ## кількість пікселів
bins = 256 ## кількість рівнів яскравості
print ('ROWS NUMBER', rows_num, 'CLMS NUMBER', clms_num, 'PIX NUMBER', pix_num,
'Bins',bins)


fig, ax = plt.subplots(figsize=(4, 4))
plt.title('ORIGINAL IMAGE')
plt.imshow(test_im)
plt.show()
```

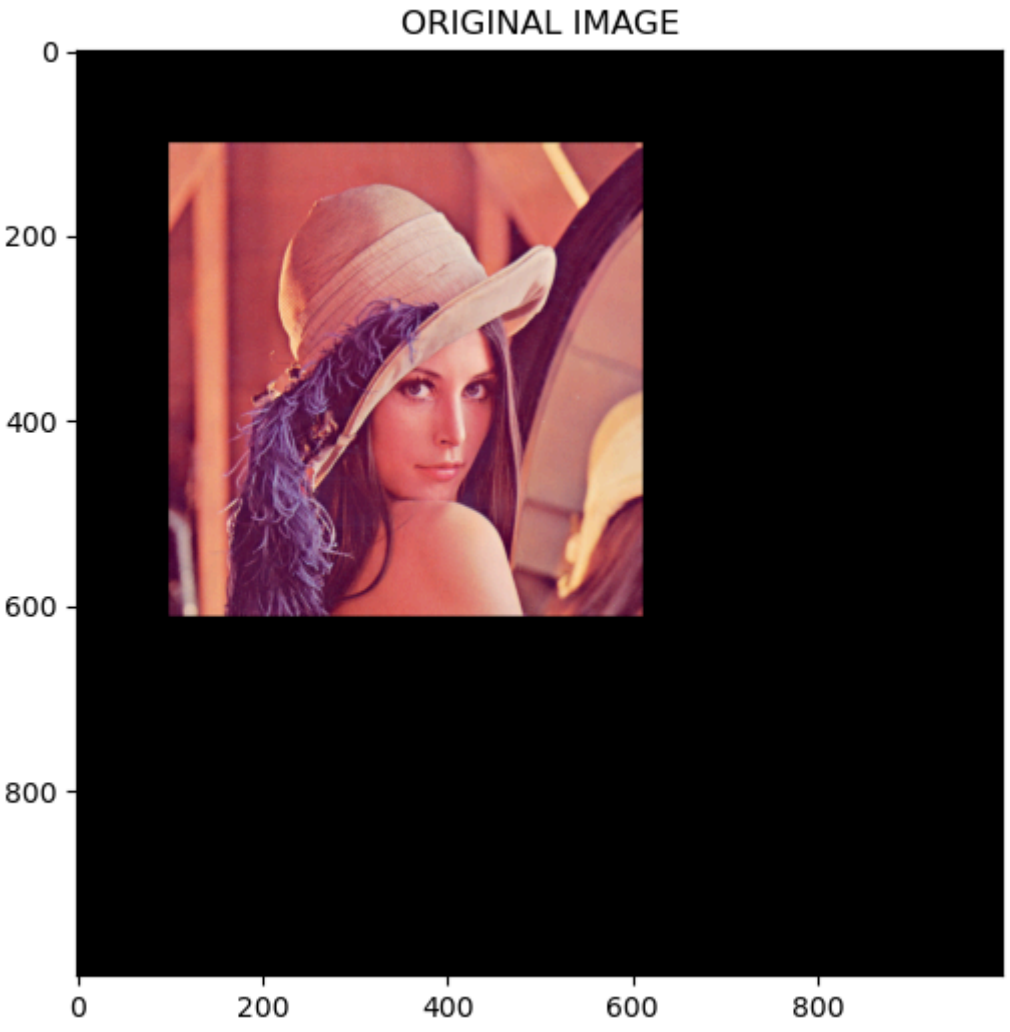
```
IMAGE SHAPE (512, 512, 3) IMAGE SIZE 786432
ROWS NUMBER 512 CLMS NUMBER 512 PIX NUMBER 262144 Bins 256
```



Додаємо зоббраження в фонове (чорне)

```
rws_num = 1000 ## кількість рядків
cms_num = 1000 ## кількість колонок
main_im = np.zeros ((rws_num, cms_num, 3), dtype=np.uint8)
```

```
Start_X = 100
Start_Y = 100
for i in  range (rows_num):
    for j in  range (clms_num):
        main_im [Start_X+i, Start_Y+j, : ] = test_im [i,j,:]

fig = plt.figure(figsize=(6, 6))
plt.title('ORIGINAL IMAGE')
plt.imshow(main_im)
plt.show()
```



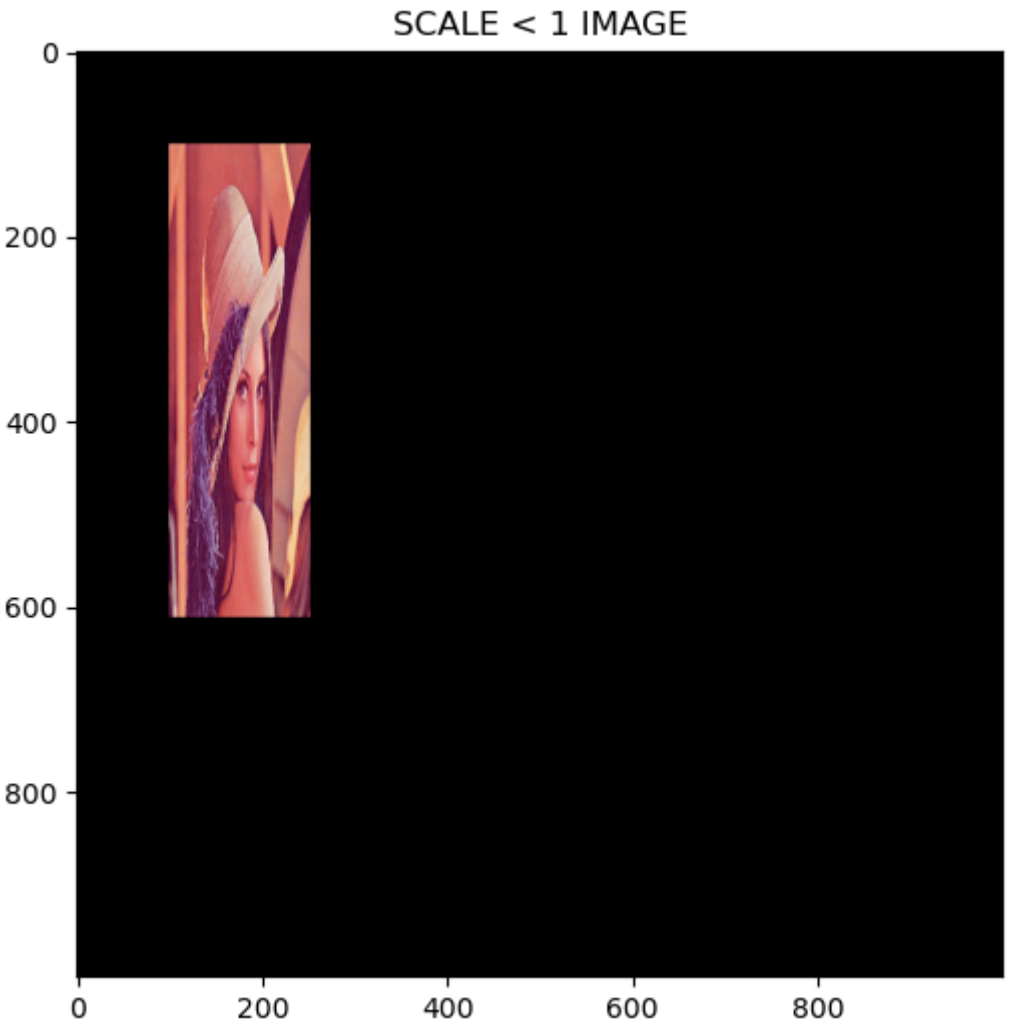
Масштабування по X (S < 1)

```
# Формуємо код Родштейна
scale = .3
Point_1 = np.array ([0, 0])
Point_2 = np.array ([clms_num, np.uint(clms_num*scale)])
print (Point_2)
rod_cod = brez(Point_1,Point_2)
rod_cod_len = len(rod_cod)
print ('Длина кода Родштейна', rod_cod_len )
print (rod_cod)
```

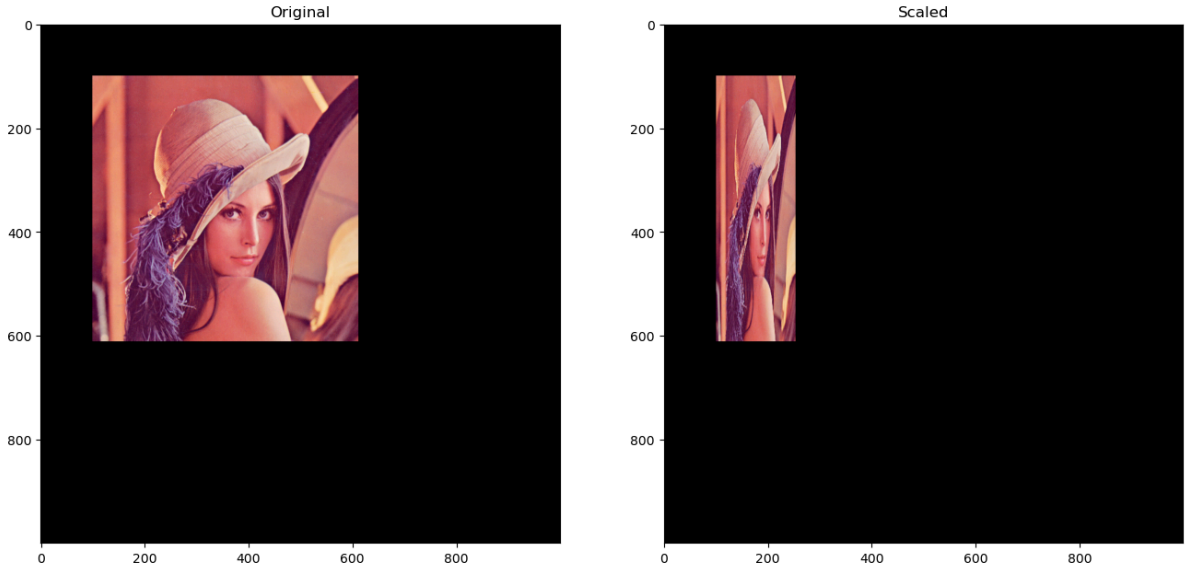
```
[512 153]
Длина кода Родштейна 512
[0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0
 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0
 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0
 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0
 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0
 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0
 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0
 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0
 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1
 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1
 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1
 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1
 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1
 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0]
```

Перерахунок

```
rws_num = 1000 ## кількість рядків
cms_num = 1000 ## кількість колонок
main_im_ = np.zeros ((rws_num, cms_num, 3), dtype=np.uint8)

##
Start_X = 100
Start_Y = 100

colm_index = 0 # индекс столбца в результирующем изображении

for j in  range (clms_num):

    if rod_cod[j] == 1 : # код Родштейна == 1 берем столбец
        for i in  range (rows_num):
            main_im_ [Start_X + i, Start_Y + colm_index, : ] = test_im [i,j,:]
        colm_index += 1


fig = plt.figure(figsize=(6, 6))
plt.title('SCALE < 1 IMAGE')
plt.imshow(main_im_)
plt.show()
```




```
ig, axes = plt.subplots(1, 2, figsize=(16, 8))
ax = axes.ravel()
ax[0].imshow(main_im)
ax[0].set_title("Original")
ax[1].imshow(main_im_)
ax[1].set_title("Scaled")
plt.show()
```

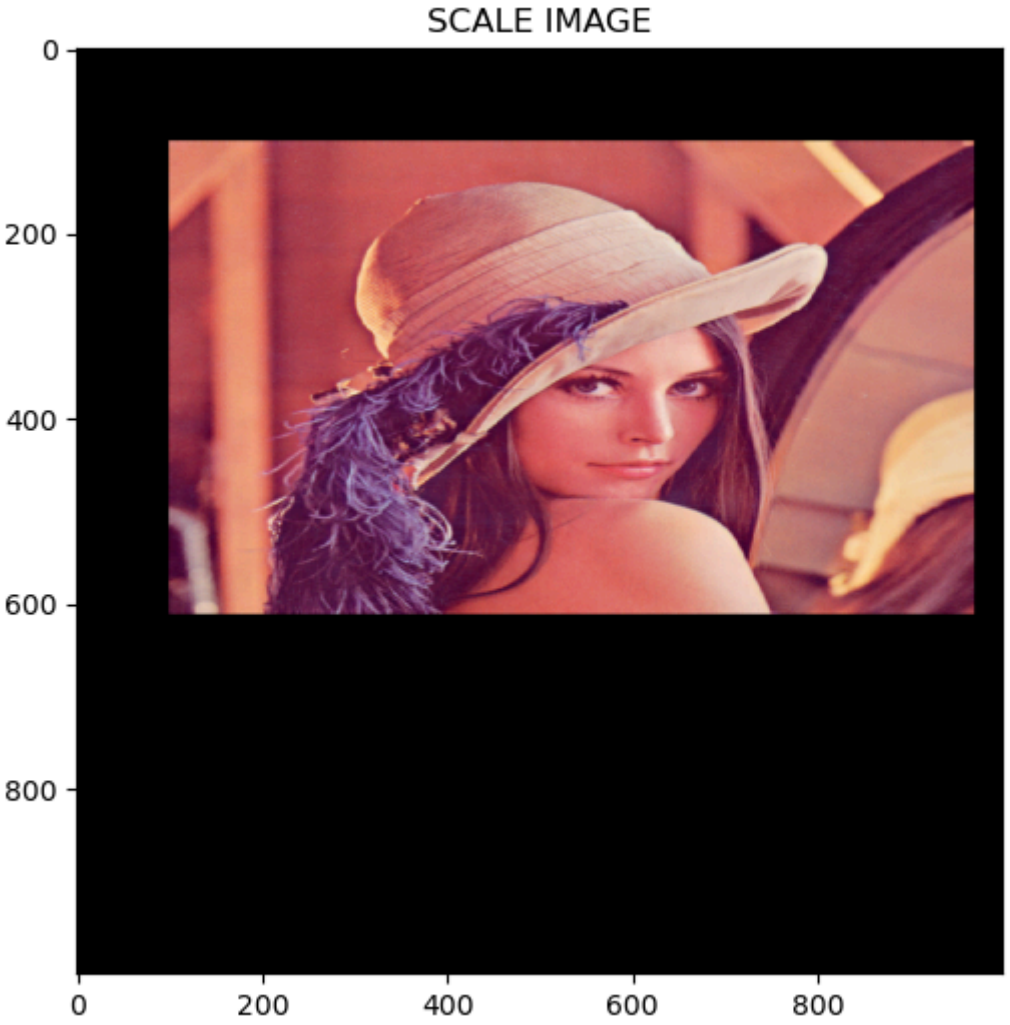


Масштабування по X (S > 1)

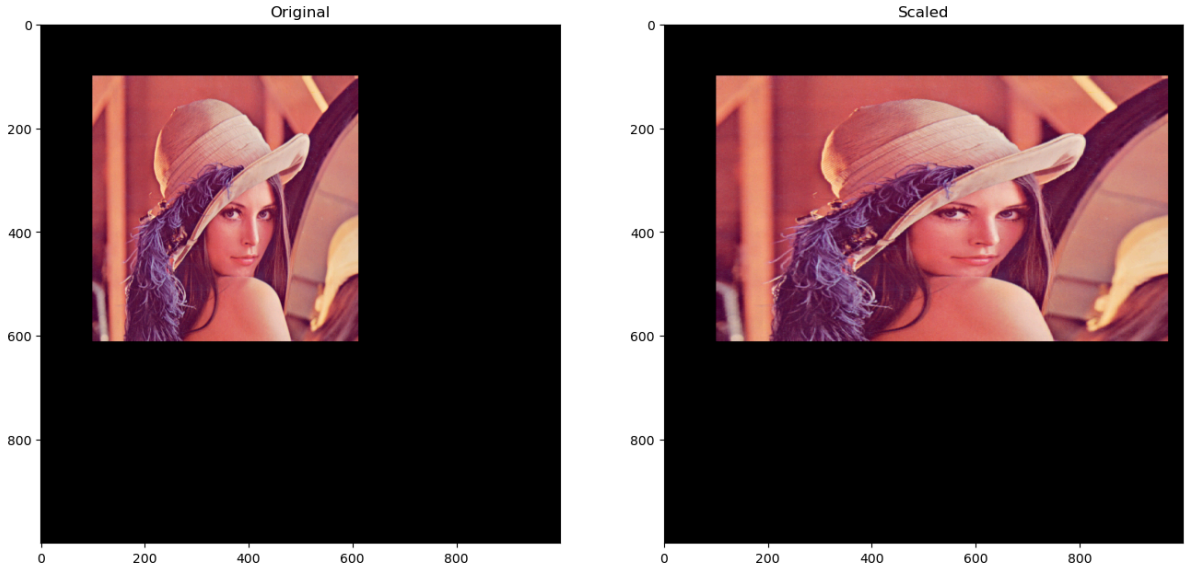
```
# Формируемо код Родштейна
scale = 1.7
Point_1 = np.array ([0, 0])
Point_2 = np.array ([np.uint(clms_num*scale), clms_num])
rod_cod = brez(Point_1,Point_2)
rod_cod_len = len(rod_cod)
print ('Длина кода Родштейна', rod_cod_len )
print (rod_cod)
```

```
Длина кода Родштейна 870
[1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1
 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1
 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1
 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1
 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1
 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0
 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1
 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1]
```

```
rws_num = 1000 ## кількість рядків
cms_num = 1000 ## кількість колонок
main_im_ = np.zeros ((rws_num, cms_num, 3), dtype=np.uint8)

##
Start_X = 100
Start_Y = 100

for i in  range (rows_num):
#for i in  range (3):
    rod_cod_index = 0
    for j in  range (clms_num):
        main_im_ [Start_X+i, Start_Y+rod_cod_index, : ] = test_im [i,j,:]
        rod_cod_index += 1
        if rod_cod_index < rod_cod_len-1:
            if rod_cod[rod_cod_index] == 0 : # код ротштейна вставляет
ДОПОЛНИТЕЛЬНЫЙ столбец
                main_im_ [Start_X+i, Start_Y+rod_cod_index, : ] = test_im [i,j,:]
                rod_cod_index += 1


fig = plt.figure(figsize=(6, 6))
plt.title('SCALE IMAGE')
plt.imshow(main_im_)
plt.show()
```




```
ig, axes = plt.subplots(1, 2, figsize=(16, 8))
ax = axes.ravel()
ax[0].imshow(main_im)
ax[0].set_title("Original")
ax[1].imshow(main_im_)
ax[1].set_title("Scaled")
plt.show()
```

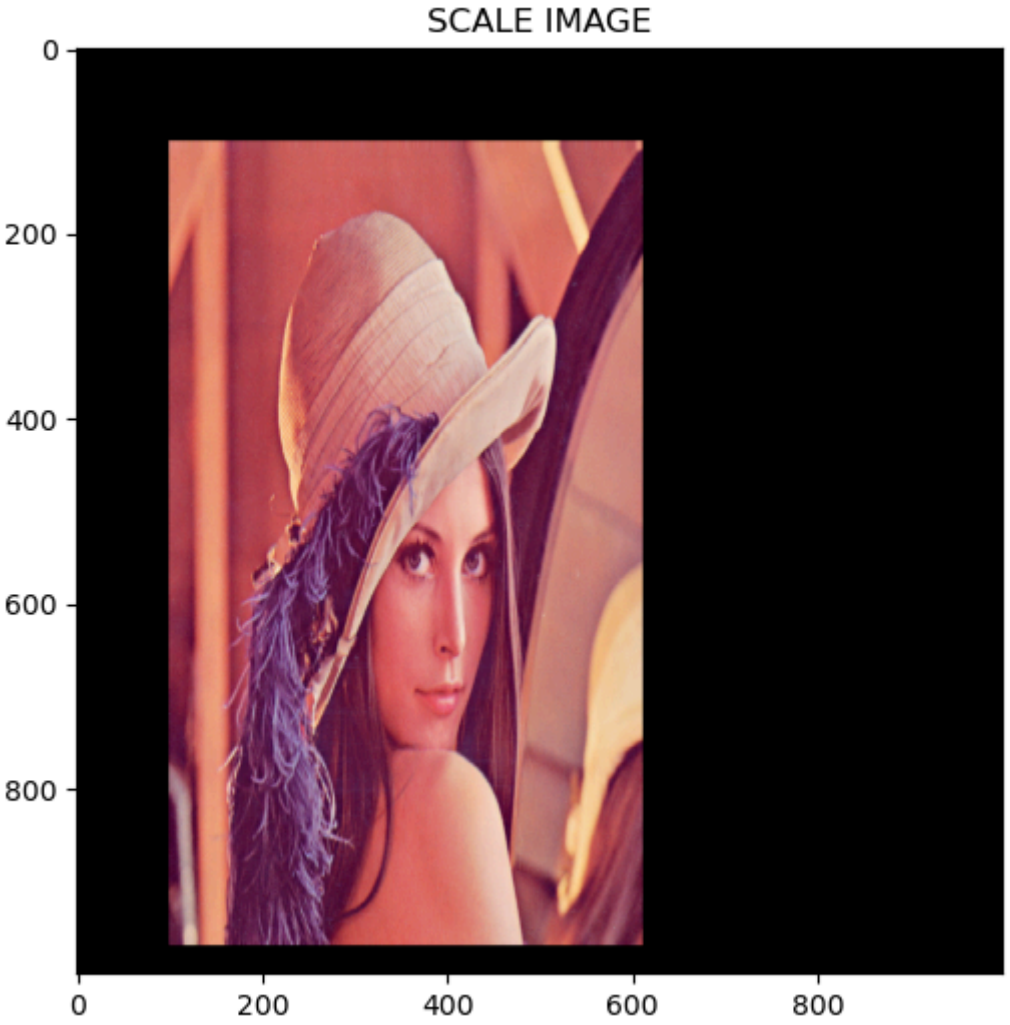


Масштабування по Y (S > 1)

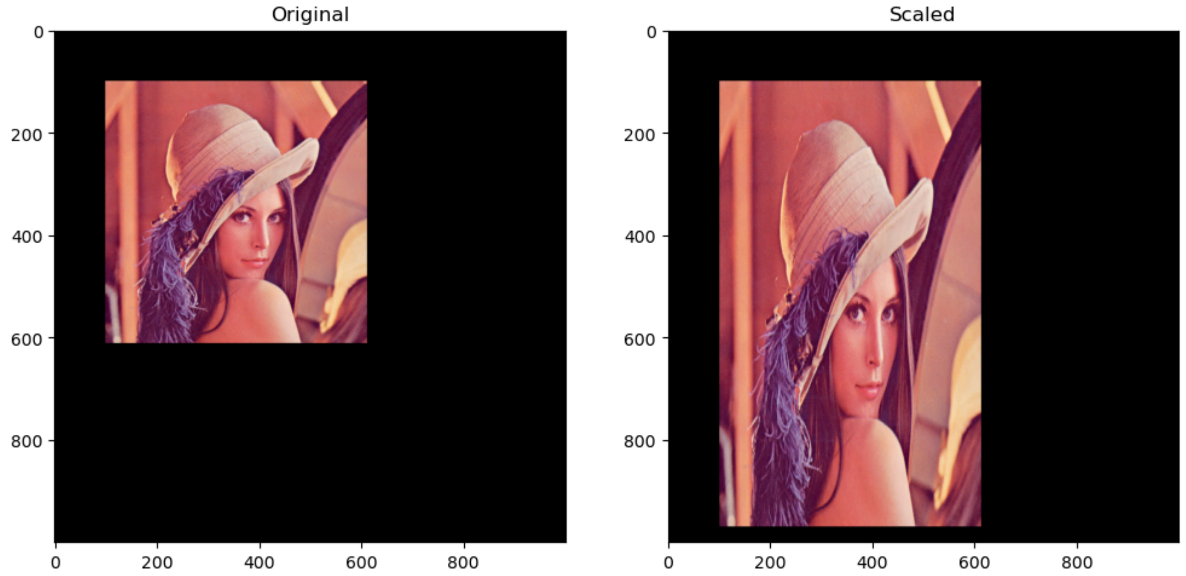
```
# Формуємо код Родштейна
scale = 1.7
Point_1 = np.array ([0, 0])
Point_2 = np.array ([np.uint(clms_num*scale), clms_num])
rod_cod = brez(Point_1,Point_2)
rod_cod_len = len(rod_cod)
print ('Длина кода Родштейна', rod_cod_len )
print (rod_cod)
```

```
Длина кода Родштейна 870
[1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1
 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1
 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1
 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1
 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1
 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0
 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1
 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0
 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1
 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1
 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1]
```

```
rws_num = 1000 ## кількість рядків
cms_num = 1000 ## кількість колонок
main_im_ = np.zeros ((rws_num, cms_num, 3), dtype=np.uint8)

##
Start_X = 100
Start_Y = 100
for j in  range (clms_num):
    rod_cod_index = 0

    for i in  range (rows_num):
        main_im_ [Start_X+rod_cod_index, Start_Y + j, : ] = test_im [i,j,:]
        rod_cod_index += 1
        if rod_cod_index < rod_cod_len-1:
            if rod_cod[rod_cod_index] == 0 : # код ротштейна вставляет
ДОПОЛНИТЕЛЬНУЮ строку
                main_im_ [Start_X+rod_cod_index, Start_Y+j, : ] = test_im [i,j,:]
                rod_cod_index += 1


fig = plt.figure(figsize=(6, 6))
plt.title('SCALE IMAGE')
plt.imshow(main_im_)
plt.show()
```




```
ig, axes = plt.subplots(1, 2, figsize=(12, 6))
ax = axes.ravel()
ax[0].imshow(main_im)
ax[0].set_title("Original")
ax[1].imshow(main_im_)
ax[1].set_title("Scaled")
plt.show()
```



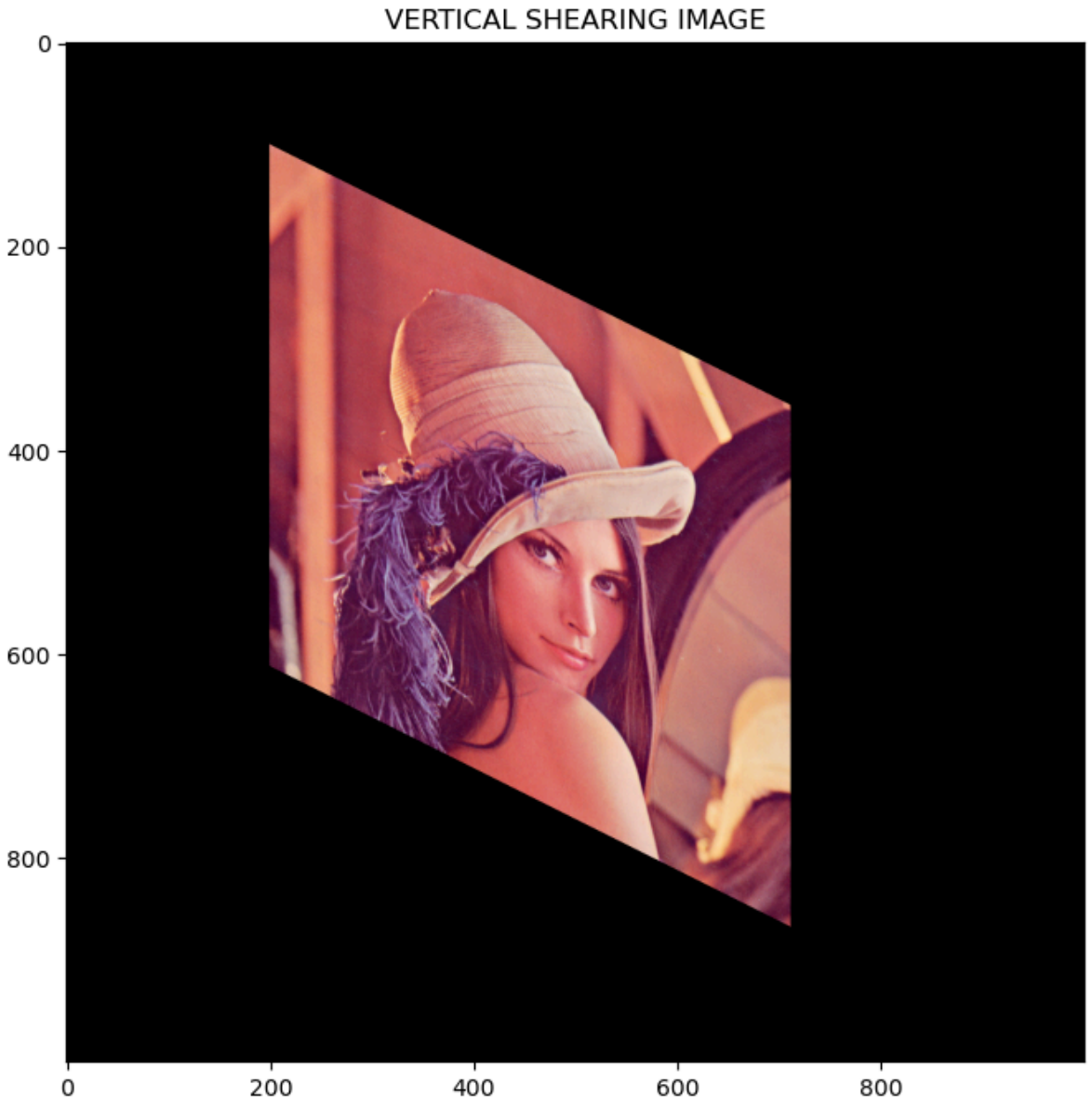
Вертикальний скос

```python
# Вертикальний скос донизу
# формуємо код Родштейна
shearing = 1.5
Point_1 = np.array ([0, 0])
Point_2 = np.array ([clms_num, np.uint(rows_num*(shearing-1.))])
print (Point_2)
rod_cod = brez(Point_1,Point_2)
rod_cod_len = len(rod_cod)
print ('Длина кода Родштейна', rod_cod_len )
print (rod_cod)
```

```
[512 256]
Длина кода Родштейна 512
[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
```

```python
# Вертикальный скос вниз
rws_num = 1000 ## кількість рядків
cms_num = 1000 ## кількість колонок
```

```
main_im_ = np.zeros ((rws_num, cms_num, 3), dtype=np.uint8)

##
Start_X = 100
Start_Y = 200

shift_row_index = 0 # Индекс скоса
for j in  range (clms_num):

    if rod_cod[j] == 1 : # Код = 1 Индекс скоса --> столдбец смещаем вниз
            shift_row_index += 1
    for i in  range (rows_num):
        main_im_ [Start_X + i + shift_row_index, Start_Y + j, : ] = test_im
[i,j,:]

fig = plt.figure(figsize=(8, 8))
plt.title('VERTICAL SHEARING IMAGE')
plt.imshow(main_im_)
plt.show()
```



```
ig, axes = plt.subplots(1, 2, figsize=(12, 6))
ax = axes.ravel()
ax[0].imshow(main_im)
ax[0].set_title("Original")
ax[1].imshow(main_im_)
ax[1].set_title("Shearing")
plt.show()
```