

# Image Processing : Activity 1 Reports

## 1.2) Image Reshape

Code :

```
x = rgbimg
print(x.shape)
y = np.moveaxis(x, -1, 0)
print(y.shape)
```

Output :

```
(400, 400, 3)
(3, 400, 400)
```

รับค่าของ **width, height, channel** มาเก็บไว้ในตัวแปร **x**

แล้วให้ **y** มีค่าเท่ากับ **np.moveaxis(x, -1, 0)** โดย **moveaxis** จะรับค่า **array, source, destination**

จากโค้ดจะเห็นว่า **array x** จะ **move** จาก **source = -1** ไปยัง **destination = 0** ก็จะเหมือนกับเลื่อนไปทางขวา(+1) 1 ครั้ง

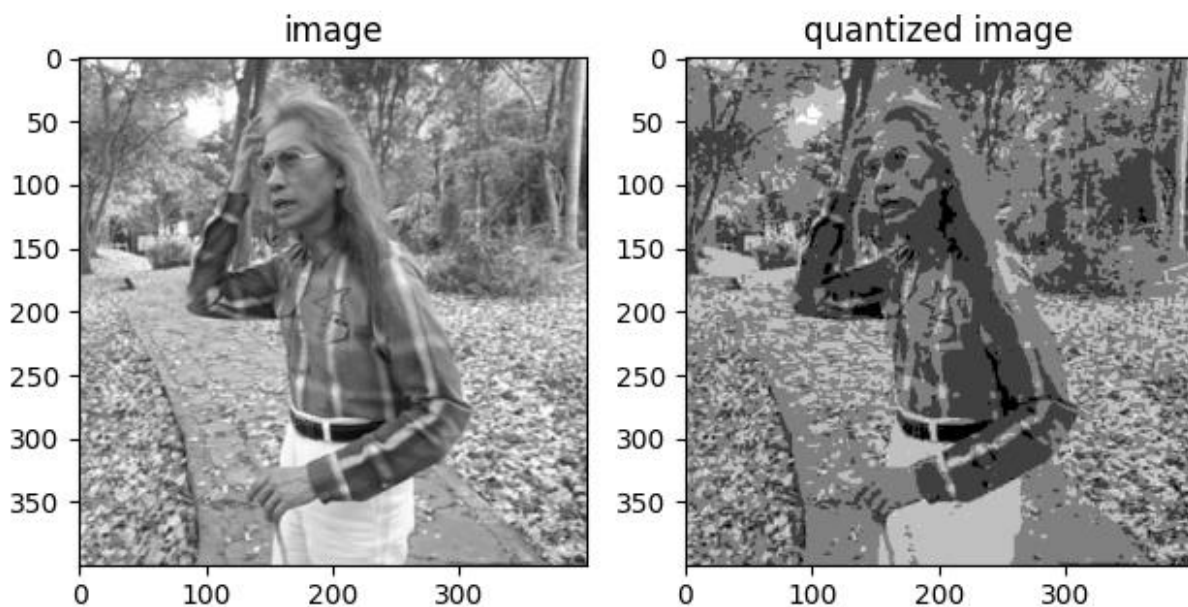
ทำให้ **width, height** เลื่อนไปทางขวา 1 ครั้ง และ **channel** ไปที่ตำแหน่งหน้าสุด มี **order** เป็น (**channel, height, width**)

### 1.3) Reduce Bit Depth using Quantization

Code :

```
import math
fig, axs = plt.subplots(1, 2)
unquanimg = cv2.imread('./BGR-ajred.jpg',0)
quanimg = unquanimg
level = 4
width = quanimg.shape[0]
height = quanimg.shape[1]
for i in range (height):
    for j in range (width):
        quanimg[i,j] = math.floor((quanimg[i,j]/255)*level)
cv2.imwrite('output.jpg', quanimg)
image = cv2.imread('./BGR-ajred.jpg',0)
axs[0].set_title('image')
axs[0].imshow(image,cmap="gray")
output = cv2.imread('./output.jpg',0)
axs[1].set_title('quantized image')
axs[1].imshow(output,cmap="gray")
plt.tight_layout()
plt.show()
```

Output :



เริ่มจาก Import math library เพื่อใช้ math.floor() แล้วสร้างพล็อตที่ใช้แสดงผลพล็อต แล้วนำค่าของรูปที่ใช้ทดลองมาเก็บไว้ในตัวแปร quaning ให้ Qlevel ที่ต้องการคือ 4 เพราะต้องการภาพที่ Bit\_Depth = 2 หลังจากนั้นนำค่า width, height ของรูปมาเก็บไว้ในตัวแปรแล้วทำการวนลูป เพื่อเปลี่ยนค่าของแต่ละ pixel ของภาพให้เป็นไปตาม

$$Qlevel = 2^{Bit\_Depth}$$

$$Q = \text{floor}(\text{NormValue}(Si) * Qlevel) \\ = \text{floor}\left(\left(\frac{Si - Smin}{Smax - Smin}\right) * Qlevel\right)$$

แล้วเก็บข้อมูลภาพไว้ใน output.jpg และทำการ read รูปทั้งสองมาเพื่อแสดงผล

## 1.4) 3D Image Surface

Code :

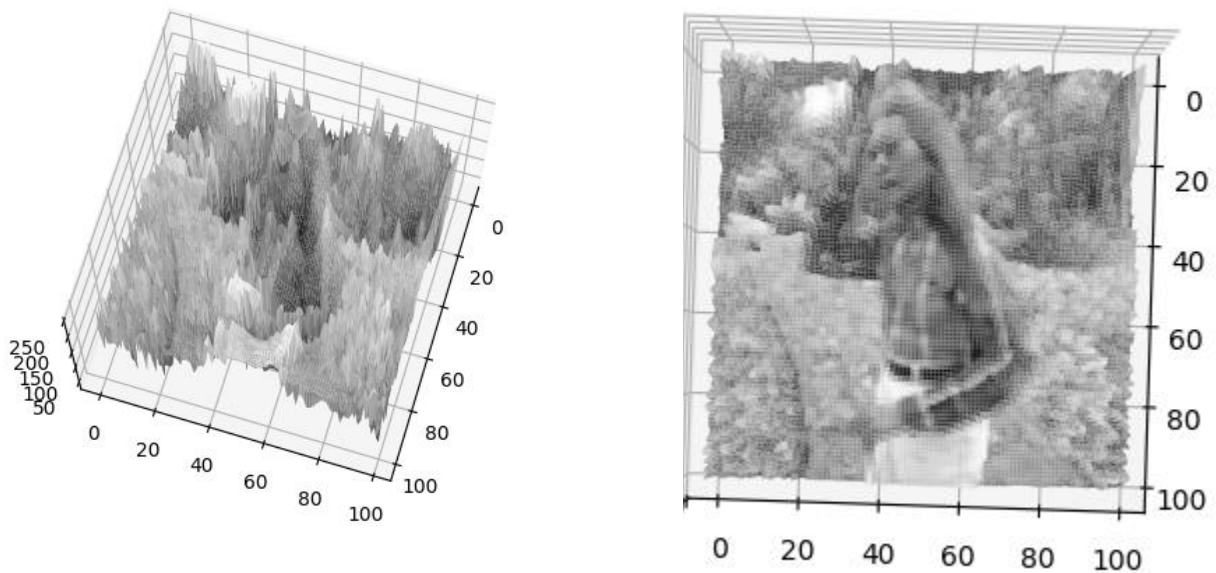
```
data = cv2.imread('./BGR-ajred.jpg',0)
data = cv2.resize(data,(100,100))

xx, yy = np.mgrid[0:data.shape[0], 0:data.shape[1]]

fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot_surface(xx, yy, data ,rstride=1, cstride=1, cmap="gray",linewidth=0)

plt.show()
```

Output :



ให้ **data** คือข้อมูลที่อ่านจากรูปที่นำมาทดลองโดยจะอ่านค่า **grayscale** แล้วทำการ **resize width,height** ให้เป็น **100x100** ให้ **xx, yy** เป็นพิกัดแกน **x** และ **y** จาก **0** ถึง **100** สร้าง **figure** และ **axes** โดยที่ **axes** มีการ **projection** แกนแบบสามมิติ หลังจากนั้นทำการ **plot** ข้อมูลลงไปด้วยคำสั่ง **ax.plot\_surface()** โดยให้ **plot** ค่าเป็น **grayscale** แล้วแสดงผล