

```
# import library
import cv2
import numpy as np
from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import preprocess_input
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.models import Model
from matplotlib import pyplot as plt
from numpy import expand_dims
```

```
"""
เป็นการแสดงพารามิเตอร์โครงสร้างของโมเดล VGG16
"""

# โหลด VGG16 Model
model = VGG16()
# สรุปรวมโครงสร้างและจำนวนพารามิเตอร์ของโมเดล VGG16
model.summary()
# ดึง array ของ weight และ bias ของ VGG16 ใน Layer 1
kernels, biases = model.layers[1].get_weights()
# ดึงโครงสร้างการทำงานทั้งหมดของ VGG16 Model ใน Layer 1 มาแสดง
model.layers[1].get_config()
```

```
# อ่านไฟล์ภาพที่ต้องการนำเข้า Model
img = cv2.imread("bird.jpg")
img_ori = img

# ปรับโครงสร้าง Image Array จาก 3D to 4D
# แปลงรูปภาพให้อยู่ในรูปแบบ Array
img = img_to_array(img)
```

```
"""
expand_dims เป็นฟังก์ชันที่ใช้ในการเพิ่มมิติเข้าไปในอาร์เรย์ และมีพารามิเตอร์สำคัญสองตัวคืออาร์เรย์
ที่ต้องการเพิ่มมิติ (ในที่นี้คือ img) และแกนที่ต้องการเพิ่มมิติเข้าไป (ในที่นี้คือ axis=0 ซึ่งหมายถึงเพิ่มมิติที่ 0)
ดังนั้นผลลัพธ์ของคำสั่งนี้คือ img จะกลายเป็นอาร์เรย์ 4 มิติที่มีขนาด (1, H, W, Ch) ซึ่งเป็นรูปแบบที่เหมาะสม
สำหรับการนำเข้ารูปภาพเข้าสู่โมเดล CNN เพื่อทำการประมวลผลและคำนวณผลลัพธ์ในแต่ละตัวอย่างของรูปภาพ
"""
```

```
img = expand_dims(img, axis=0)
```

```
"""
preprocess_input คือฟังก์ชันที่ทำการปรับปรุงภาพที่กำลังจะนำเข้าให้เหมาะสมกับโมเดล VGG16 หรือโมเดล
CNN อื่น ๆ โดยจะทำการปรับค่าสีและการปรับปรุงค่าอื่น ๆ ของภาพ
"""
```

```
img = preprocess_input(img)
```

```
fig, axes = plt.subplots(2, 2, figsize=(10, 10))
```

```
# แสดงภาพ Original
```

```
axes[0,0].imshow(img_ori)
```

```
axes[0,0].set_title('Original Image')
```

```
# แสดงภาพหลังทำ preprocess
```

```
axes[0,1].imshow(img[0])
```

```
axes[0,1].set_title('Preprocess Image')
```

```
# แสดงภาพหลังทำ preprocess
```

```
axes[1,0].imshow(img[0])
```

```
axes[1,0].set_title('Preprocess Image')
```

```
# แสดงภาพหลังทำ preprocess
```

```
axes[1,1].imshow(img[0])
```

```
axes[1,1].set_title('Preprocess Image')
```

```
f = plt.figure(figsize=(16,16))
```

```

model = VGG16()

# ดึงโมเดล VGG16 เฉพาะ CNN Layer 1 มาใช้
model = Model(inputs=model.inputs, outputs=model.layers[1].output)
model.summary()

# โหลด Image โดยเอาขนาดตามที่เรากำลังต้องการคือ 224x224
img = load_img('bird.jpg', target_size=(224, 224))
img = img_to_array(img)
img = expand_dims(img, axis=0)
img = preprocess_input(img)

# ดึง Feature map ของ CNN Layer 1 มา โดยทำ model prediction
feature_maps = model.predict(img)

# แสดงภาพ Feature map ผลลัพธ์จาก 64 node ของ CNN Layer 1 แสดงเป็น 8x8
square = 8
ix = 1
for _ in range(square):
    for _ in range(square):
        # specify subplot and turn of axis
        ax = plt.subplot(square, square, ix)
        ax.set_xticks([])
        ax.set_yticks([])
        # plot filter channel in grayscale
        plt.imshow(feature_maps[0, :, :, ix-1], cmap='viridis')
        ix += 1

plt.show()

```