TensorFlow2教程-使用預訓練模型

In [1]:
```
#! pip install Pillow
! curl 'https://gfp-2a3tnpzj.stackpathdns.com/wp-content/uploads/2016/07/Dachshur
! ls
#參考 https://blog.gtwang.org/programming/keras-resnet-50-pre-trained-model-build
```

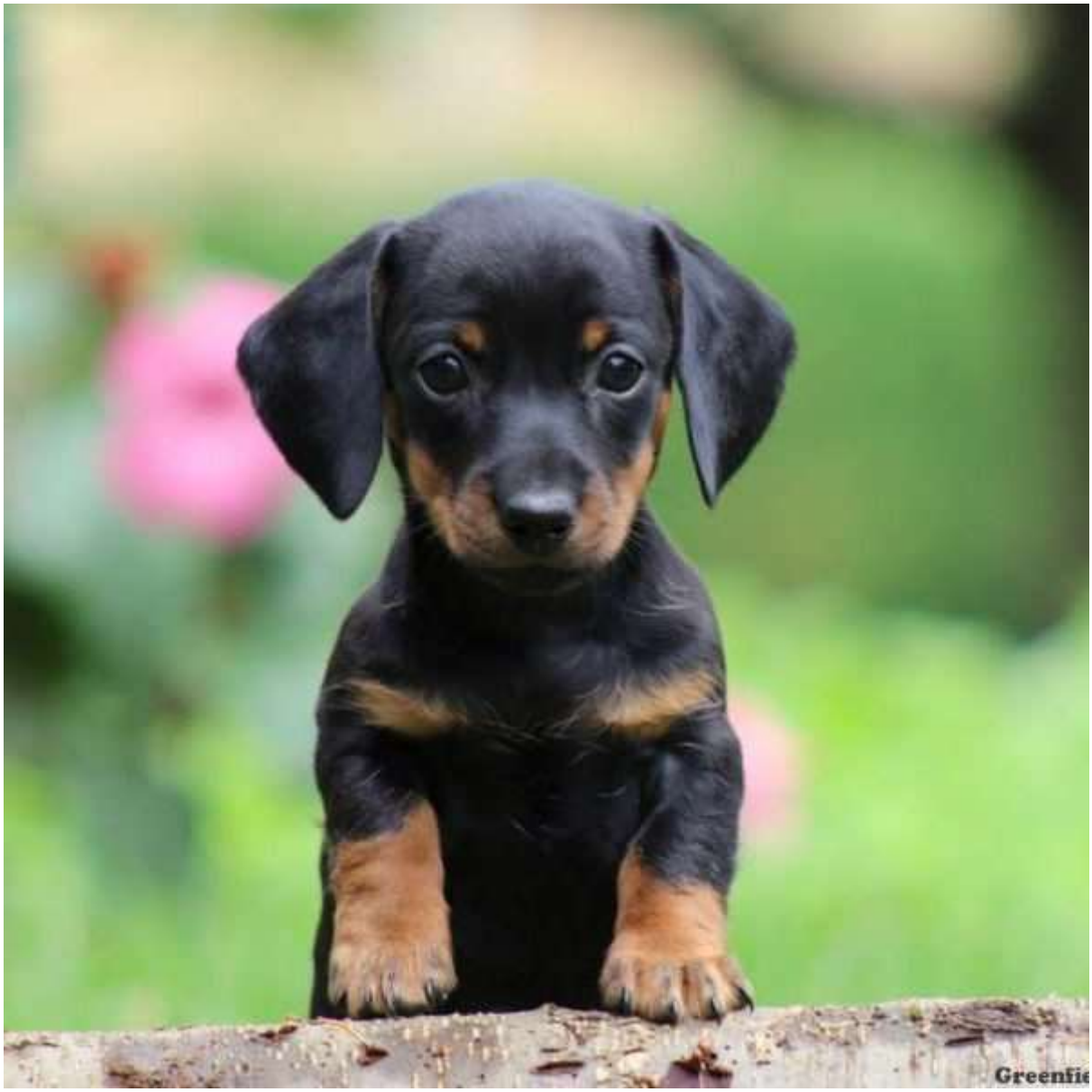
```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 39238  100 39238    0     0  28682      0  0:00:01  0:00:01 --:--:-- 28661
001-cnn.ipynb           003-text_cnn.ipynb        dog.jpg
002-cnn_variants.ipynb  004-pretrained_cnn.ipynb
```

In [1]:
```
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications import resnet50
```

In [2]:
```
img = image.load_img('dog.jpg')
print(image.img_to_array(img).shape)
img
```

```
(600, 600, 3)
```

Out[2]:



1.導入模型

目前看使用模型：

Import model

- Currently, seven models are supported
  - Xception
  - VGG16
  - VGG19
  - ResNet50
  - InceptionV3
  - InceptionResNetV2
  - MobileNet
  - MobileNetV2
  - DenseNet
  - nasnet

In [3]:
```
model = resnet50.ResNet50(weights='imagenet')
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applicati
ons/resnet/resnet50_weights_tf_dim_ordering_tf_kernels.h5 (https://storage.goog
leapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_orderin
g_tf_kernels.h5)
102973440/102967424 [==============================] - 35s 0us/step
```

In [9]:
```
img = image.load_img('dog.jpg', target_size=(224, 224))
img = image.img_to_array(img)
img = np.expand_dims(img, axis=0)
print(img.shape)
```

```
(1, 224, 224, 3)
```

2.模型預測

In [5]:
```
pred_class = model.predict(img)
```

In [10]:
```
n = 10
top_n = resnet50.decode_predictions(pred_class, top=n)
for c in top_n[0]:
    print(c)
```

```
('n02106550', 'Rottweiler', 0.7156204)
('n02107142', 'Doberman', 0.10712539)
('n02107312', 'miniature_pinscher', 0.051518817)
('n02107908', 'Appenzeller', 0.03773272)
('n02101006', 'Gordon_setter', 0.030471016)
('n02112706', 'Brabancon_griffon', 0.013199493)
('n02089078', 'black-and-tan_coonhound', 0.012116586)
('n02108000', 'EntleBucher', 0.0034771902)
('n02099712', 'Labrador_retriever', 0.002470969)
('n02093754', 'Border_terrier', 0.001997008)
```

In [7]:
```
# img = image.load_img('dog.jpg')
# img = image.img_to_array(img)
# print(img.shape)
img = resnet50.preprocess_input(img)
print(img.shape)
```

```
(1, 224, 224, 3)
```

In [12]:
```
pred_class = model.predict(img)
n = 10
top_n = resnet50.decode_predictions(pred_class, top=n)
for c in top_n[0]:
    print(c)
```

```
('n02099849', 'Chesapeake_Bay_retriever', 0.5004646)
('n02099712', 'Labrador_retriever', 0.19408423)
('n02088364', 'beagle', 0.055846117)
('n02105412', 'kelpie', 0.034197725)
('n02087394', 'Rhodesian_ridgeback', 0.02148086)
('n02090379', 'redbone', 0.019448025)
('n04409515', 'tennis_ball', 0.014889464)
('n02100236', 'German_short-haired_pointer', 0.013312437)
('n02107142', 'Doberman', 0.009854675)
('n02107908', 'Appenzeller', 0.008381209)
```

In [ ]: