

Lab 2: Convolutional Neural Network

(環境：google colab，以下是有修改或新增的程式碼)

● 準備資料

原程式碼內沒有「連結 google 雲端」的程式碼，新增下面程式碼在解壓縮 data.zip 上面，如圖 1：

準備資料

在做這個分類器之前，請先將train.zip解壓縮。解壓縮之後，會有很多貓的圖片和狗的圖片。貓的圖片是以cat開頭，狗的圖片是以dog開頭。

請創立一個資料夾，資料夾的結構如下：

- 將2000張貓(狗)的圖片移至 data/train/cats (data/train/dogs)
- 將1000張貓(狗)的圖片移至 data/validation/cats (data/validation/dogs)
- 將1000張貓(狗)的圖片移至 data/test/cats (data/test/dogs)

```
[ ] from google.colab import drive  
    drive.mount('/content/drive')
```

Mounted at /content/drive

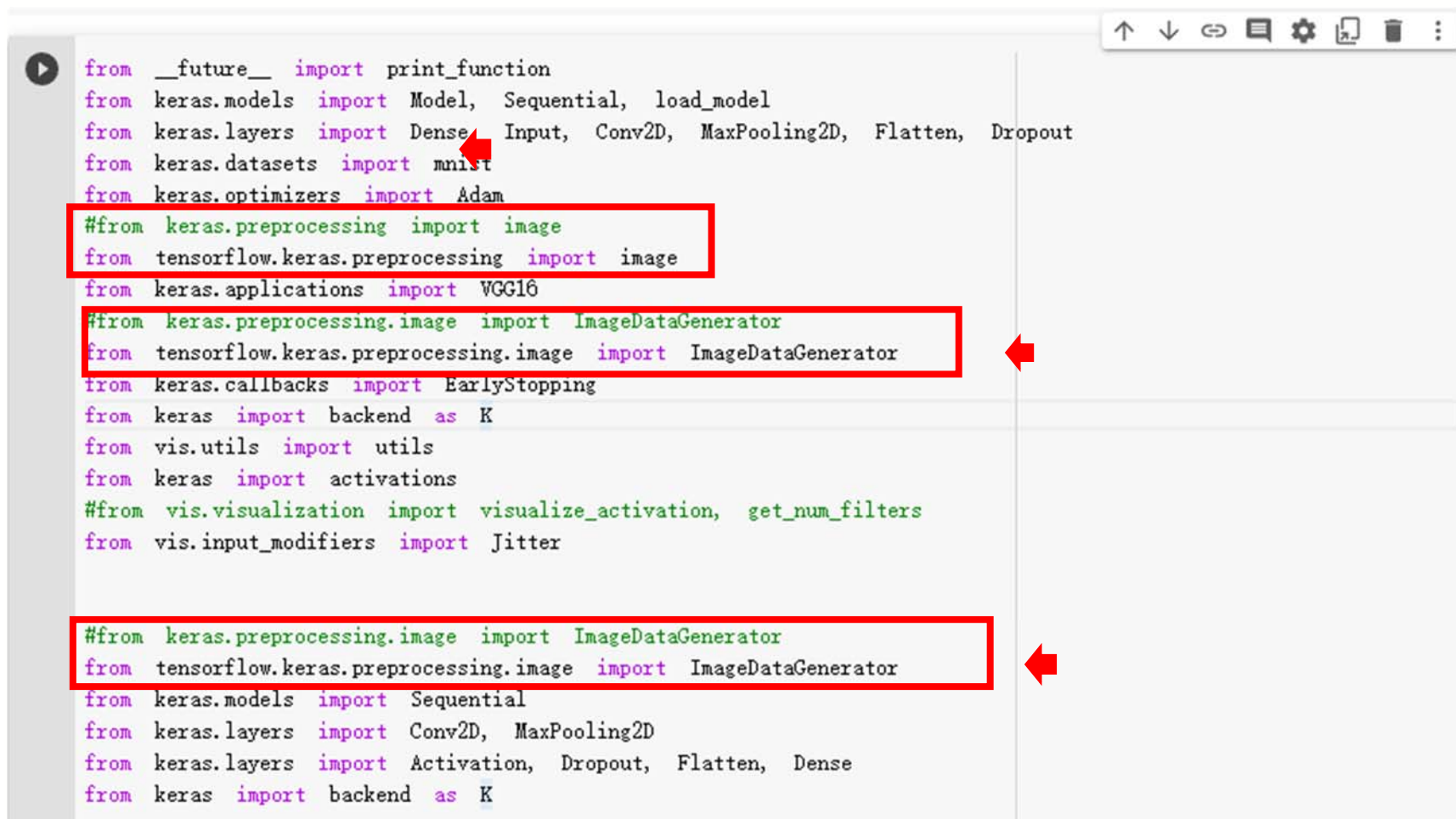
```
!unzip data.zip
```

```
Archive: data.zip  
replace data/test/cats/cat.151.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: n  
replace data/test/cats/cat.152.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: N
```

圖 1：連結 google 雲端

● Import Library

第二段原程式碼，因為 google colab 的 keras 版本較新，無法使用 `load_img` 以及 `img_to_array`，所以這邊程式碼的部分改使用 `tensorflow.keras` 函式，先更改需要 import 的 library，註解(#綠色)的地方是原始碼，如圖 2：



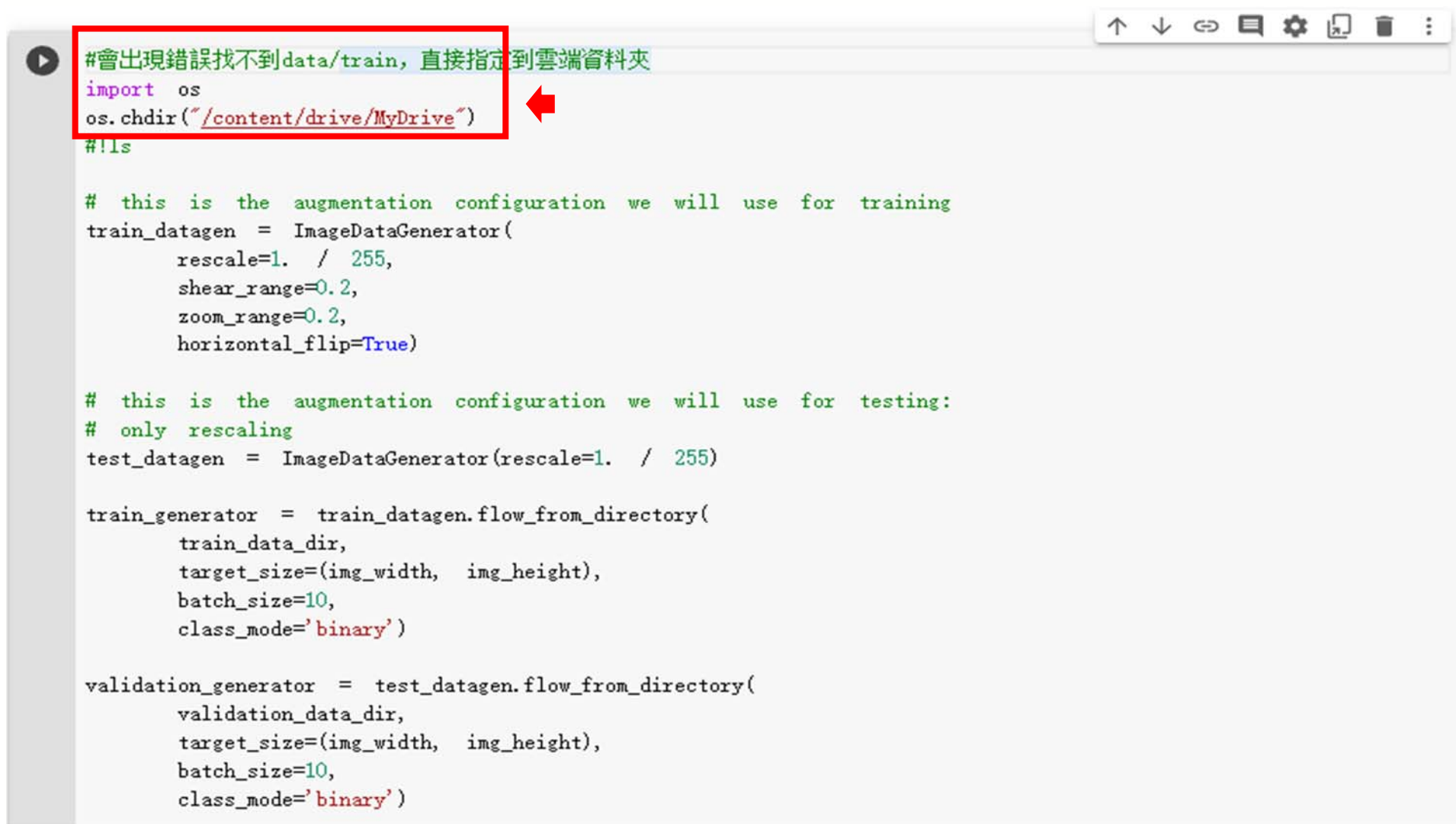
```
from __future__ import print_function
from keras.models import Model, Sequential, load_model
from keras.layers import Dense, Input, Conv2D, MaxPooling2D, Flatten, Dropout
from keras.datasets import mnist
from keras.optimizers import Adam
#from keras.preprocessing import image
from tensorflow.keras.preprocessing import image
from keras.applications import VGG16
#from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import EarlyStopping
from keras import backend as K
from vis.utils import utils
from keras import activations
#from vis.visualization import visualize_activation, get_num_filters
from vis.input_modifiers import Jitter

#from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras import backend as K
```

圖 2：import library

● Data Generator

第一段原程式碼，在 google colab 上有時候會找不到 data/train，因此直接指定路徑到連線的雲端資料夾裡，如圖 3：



```
#會出現錯誤找不到data/train, 直接指定到雲端資料夾
import os
os.chdir("/content/drive/MyDrive")
#!ls

# this is the augmentation configuration we will use for training
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

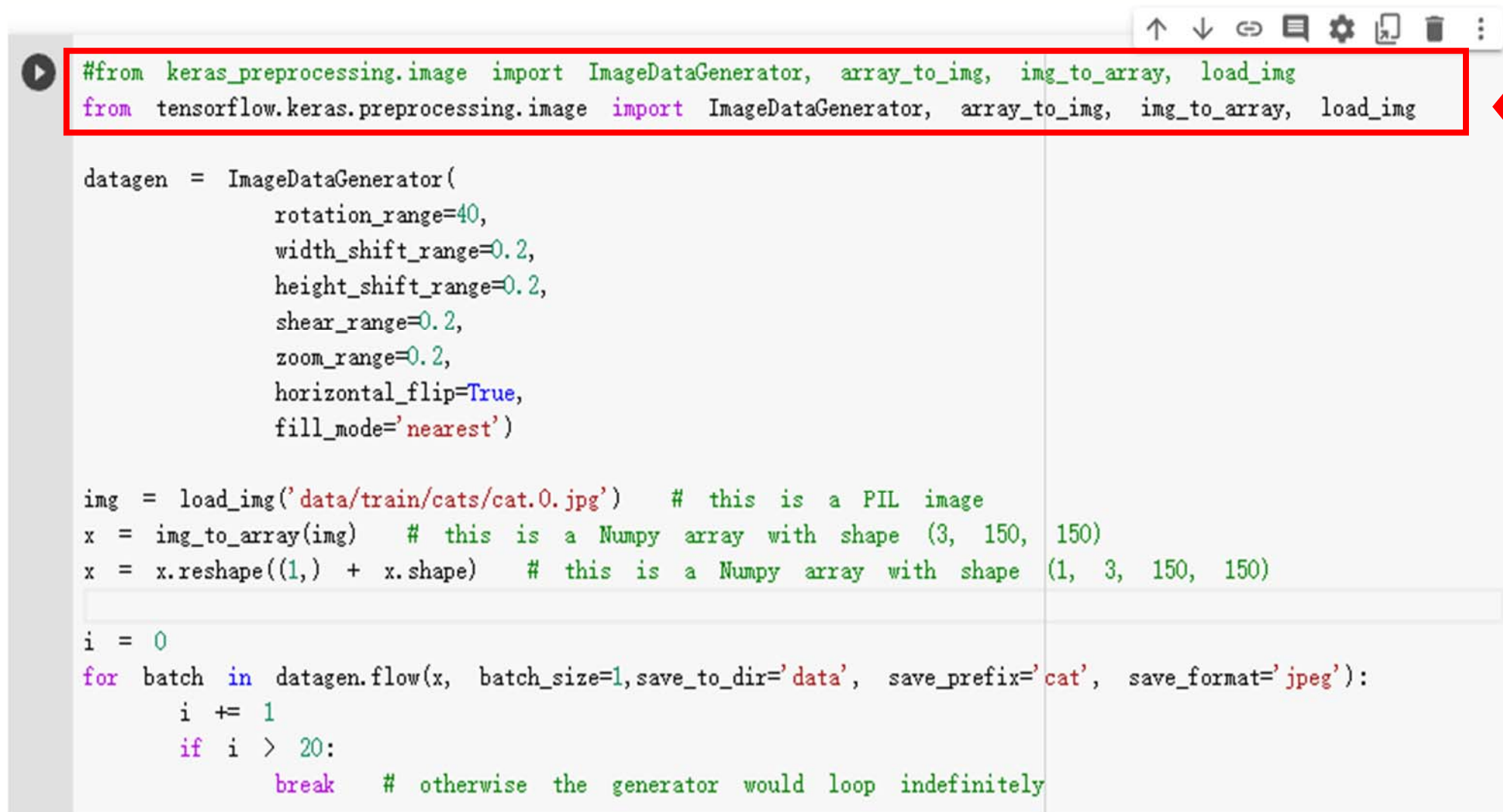
# this is the augmentation configuration we will use for testing:
# only rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=10,
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(img_width, img_height),
    batch_size=10,
    class_mode='binary')
```

圖 3：指定路徑

第二段原程式碼，因為 google colab 的 keras 版本較新，無法使用 load_img 以及 img_to_array，所以這邊程式碼的部分改使用 tensorflow.keras 函式，註解(#綠色)的地方是原始碼，如圖 4：



```
#from keras_preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
from tensorflow.keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img

datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

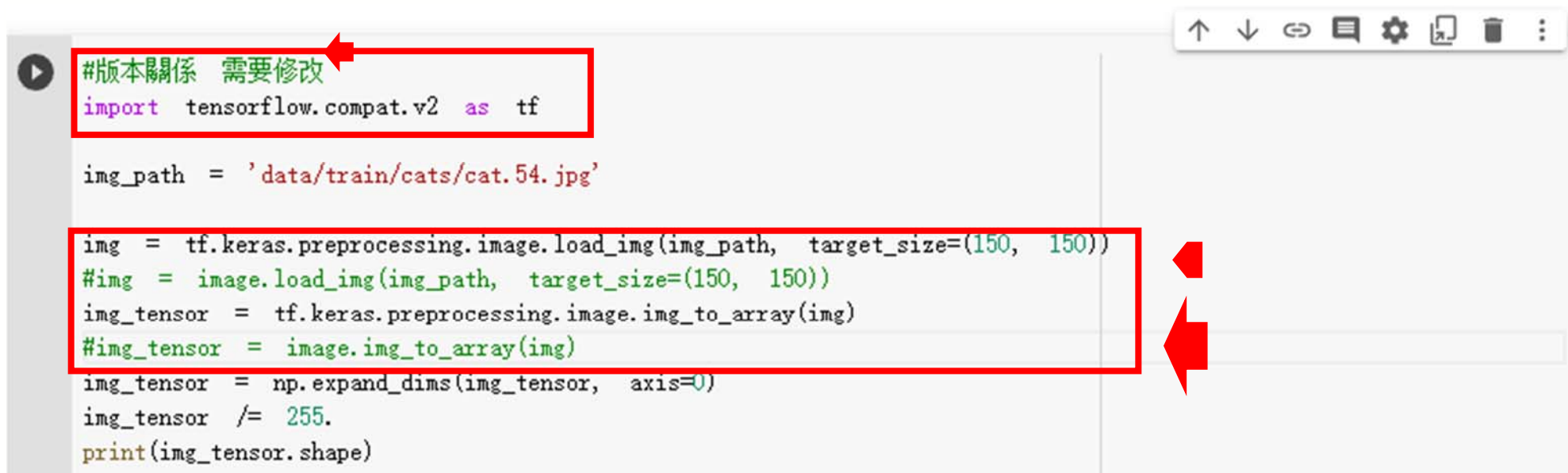
img = load_img('data/train/cats/cat.0.jpg') # this is a PIL image
x = img_to_array(img) # this is a Numpy array with shape (3, 150, 150)
x = x.reshape((1,) + x.shape) # this is a Numpy array with shape (1, 3, 150, 150)

i = 0
for batch in datagen.flow(x, batch_size=1, save_to_dir='data', save_prefix='cat', save_format='jpeg'):
    i += 1
    if i > 20:
        break # otherwise the generator would loop indefinitely
```

圖 4：改為 tensorflow.keras 函式

● Quiz 2: Prediction

第一段原程式碼，如上述所說，這部分的原程式碼，因 google colab 的 keras 與 tensorflow 的版本較高，所以原本可使用的 `load_img` 以及 `img_to_array`，都無法使用，所以改用 `tf.keras.preprocessing.image.load_img` 以及 `tf.keras.preprocessing.image.img_to_array`，如圖 5：



```
#版本關係 需要修改
import tensorflow.compat.v2 as tf

img_path = 'data/train/cats/cat.54.jpg'

img = tf.keras.preprocessing.image.load_img(img_path, target_size=(150, 150))
#img = image.load_img(img_path, target_size=(150, 150))
img_tensor = tf.keras.preprocessing.image.img_to_array(img)
#img_tensor = image.img_to_array(img)
img_tensor = np.expand_dims(img_tensor, axis=0)
img_tensor /= 255.
print(img_tensor.shape)
```

圖 5：load_img、img_to_array

第二段原程式碼，因為 predict_classes 在 TensorFlow 2.6 版就被刪除了，所以小改了一下程式碼，如圖 6：

```
[ ] # put your code here
    # hint: use "train_generator.class_indices" to confirm the coding of each class

    prediction = (model.predict(img_tensor) > 0.5)*1
    #prediction = model.predict_classes(img_tensor)
    #predict_classes在TensorFlow 2.6 版就被刪除了

    if prediction == 0:
        print("It is a cat!")
    else:
        print("It is a dog!")
```

圖 6：predict_classes 問題

● Quiz 3

將上面 Quiz 2 在第一段及第二段更改過的程式碼，在這一段也進行修改，因程式碼較長，所以只擷取需要修改的程式碼區塊，如：

```
[ ] for cat in os.listdir(path='data/test/cats'):  
    img_path=os.path.join('data/test/cats', cat)  
    img = tf.keras.preprocessing.image.load_img(img_path, target_size=(150, 150))  
    img_tensor = tf.keras.preprocessing.image.img_to_array(img)  
    #img = image.load_img(img_path, target_size=(150, 150))  
    #img_tensor = image.img_to_array(img)  
    img_tensor = np.expand_dims(img_tensor, axis=0)  
    img_tensor /= 255.  
    prediction = (model.predict(img_tensor) > 0.5)*1  
    #prediction = model.predict_classes(img_tensor)  
    if prediction == 0:  
        correct_img=correct_img+1  
        correct_img_cat=correct_img_cat+1  
    total_img=total_img+1  
    total_img_cat=total_img_cat+1  
  
print("correct_predict_cat: "+str(correct_img_cat))  
print("total_image_cat: "+str(total_img_cat))  
print("prediction accuracy_cat: "+str(correct_img_cat/total_img_cat))  
  
for dog in os.listdir(path='data/test/dogs'):  
    img_path=os.path.join('data/test/dogs', dog)  
    img = tf.keras.preprocessing.image.load_img(img_path, target_size=(150, 150))  
    img_tensor = tf.keras.preprocessing.image.img_to_array(img)  
    #img = image.load_img(img_path, target_size=(150, 150))  
    #img_tensor = image.img_to_array(img)  
    img_tensor = np.expand_dims(img_tensor, axis=0)  
    img_tensor /= 255.  
    #test_tensors.append(img_tensor)  
    #test_classes.append(1)  
    prediction = (model.predict(img_tensor) > 0.5)*1  
    #prediction = model.predict_classes(img_tensor)  
    if prediction == 1:  
        correct_img=correct_img+1  
        correct_img_dog=correct_img_dog+1  
    total_img=total_img+1  
    total_img_dog=total_img_dog+1
```

圖 7：訓練結果