```
import cv2
import os
import numpy as np
import matplotlib.pyplot as plt
import random
from keras.utils import np_utils

# 1. 資料前處理

'''
訓練集分為三大類：IKEA鯊魚、真鯊魚、其他各9000張，測試集各1000張
利用np.empty()去做出所需的空陣列，用來儲存圖片data以及其圖片label
所有圖片大小皆已轉為32x32大小
'''

ikeashark_train_data = np.empty((9000,3,32,32),dtype="uint8")
realshark_train_data = np.empty((9000,3,32,32),dtype="uint8")
others_train_data = np.empty((9000,3,32,32),dtype="uint8")

ikeashark_train_label = np.empty((9000,),dtype="uint8")
realshark_train_label = np.empty((9000,),dtype="uint8")
others_train_label = np.empty((9000,),dtype="uint8")

ikeashark_test_data = np.empty((1000,3,32,32),dtype="uint8")
realshark_test_data = np.empty((1000,3,32,32),dtype="uint8")
others_test_data = np.empty((1000,3,32,32),dtype="uint8")

ikeashark_test_label = np.empty((1000,),dtype="uint8")
realshark_test_label = np.empty((1000,),dtype="uint8")
others_test_label = np.empty((1000,),dtype="uint8")

# 1-1. 訓練集檔案輸入 ('@@'為路徑)

'''
先使用os.listdir()將資料夾中所有的檔名存成一串列，再利用不同資料夾去進行三種訓練集圖片資料的讀
取，並分別存在三個data陣列中
'''

ikeashark_train_names = os.listdir('./train/IKEA_shark_train')
for i in range(0,len(ikeashark_train_names)):
    img = cv2.imread('./train/IKEA_shark_train/'+ikeashark_train_names[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # 將圖片顏色轉為正常
    if img is None: # 讀取不到圖片則跳過此次迴圈，防bug用
        continue
    ikeashark_train_data[i,:,:,:] = [img[:,:,0] , img[:,:,1] , img[:,:,2]]
    ikeashark_train_label[i] = 0 # label:0 代表此張圖片為IKEA鯊魚

realshark_train_names = os.listdir('./train/Real_shark_train')
for i in range(0,len(realshark_train_names)):
    img = cv2.imread('./train/Real_shark_train/'+realshark_train_names[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if img is None:
        continue
    realshark_train_data[i,:,:,:] = [img[:,:,0] , img[:,:,1] , img[:,:,2]]
    realshark_train_label[i] = 1 # label:1 代表此張圖片為real鯊魚
```

```
others_train_names = os.listdir('./train/Others_train')
for i in range(0,len(others_train_names)):
    img = cv2.imread('./train/Others_train/'+others_train_names[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if img is None:
        continue
    others_train_data[i,:,:,:] = [img[:,:,0] , img[:,:,1] , img[:,:,2]]
    others_train_label[i] =2 # label:2 代表此張圖片屬於others

# 1-2. 測試集檔案輸入

'''
基本上與訓練集的蒐集方式相同
'''

ikeashark_test_names = os.listdir('./test/IKEA_shark_test')
for i in range(0,len(ikeashark_test_names)):
    img = cv2.imread('./test/IKEA_shark_test/'+ikeashark_test_names[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if img is None:
        continue
    ikeashark_test_data[i,:,:,:] = [img[:,:,0] , img[:,:,1] , img[:,:,2]]
    ikeashark_test_label[i] = 0

realshark_test_names = os.listdir('./test/Real_shark_test')
for i in range(0,len(realshark_test_names)):
    img = cv2.imread('./test/Real_shark_test/'+realshark_test_names[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if img is None:
        continue
    realshark_test_data[i,:,:,:] = [img[:,:,0] , img[:,:,1] , img[:,:,2]]
    realshark_test_label[i] = 1

others_test_names = os.listdir('./test/Others_test')
for i in range(0,len(others_test_names)):
    img = cv2.imread('./test/Others_test/'+others_test_names[i])
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if img is None:
        continue
    others_test_data[i,:,:,:] = [img[:,:,0] , img[:,:,1] , img[:,:,2]]
    others_test_label[i] = 2

# 1-3. 資料合併與打亂

'''
先進行三個data陣列與三個label陣列的合併，再利用shuffle()將其打亂 (x為資料，y為label，也就是
答案)
'''

x_train =
np.vstack((ikeashark_train_data,realshark_train_data,others_train_data))
x_test = np.vstack((ikeashark_test_data,realshark_test_data,others_test_data))
y_train =
np.concatenate((ikeashark_train_label,realshark_train_label,others_train_label))
y_test =
np.concatenate((ikeashark_test_label,realshark_test_label,others_test_label))
```

```
x_train = x_train.transpose(0,2,3,1)
x_test = x_test.transpose(0,2,3,1)

index_1 = [i for i in range(len(x_train))]
random.shuffle(index_1)
x_train = x_train[index_1]
y_train = y_train[index_1]

index_2 = [i for i in range(len(x_test))]
random.shuffle(index_2)
x_test = x_test[index_2]
y_test = y_test[index_2]

# 1-4. 將資料做正規化，且將label做onehot編碼

x_train_normalize = x_train.astype('float32') / 255.0
x_test_normalize = x_test.astype('float32') / 255.0

y_train_OneHot = np_utils.to_categorical(y_train,3) # 3->三種分類
y_test_OneHot = np_utils.to_categorical(y_test,3)


# 2. 建立模型

from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D

model = Sequential()

# 卷積層1與池化層1

model.add(Conv2D(filters=64,kernel_size=(3,3),
                 input_shape=(32,32,3),
                 activation='relu',
                 padding='same'))

model.add(Dropout(rate=0.25))

model.add(MaxPooling2D(pool_size=(2, 2)))

# 卷積層2與池化層2

model.add(Conv2D(filters=128, kernel_size=(3, 3),
                 activation='relu', padding='same'))

model.add(Dropout(0.25))

model.add(MaxPooling2D(pool_size=(2, 2)))

# 卷積層3與池化層3

model.add(Conv2D(filters=256, kernel_size=(3, 3),
                 activation='relu', padding='same'))

model.add(Dropout(0.25))

model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
# 3. 建立神經網路(平坦層、隱藏層、輸出層)

model.add(Flatten())
model.add(Dropout(rate=0.25))

model.add(Dense(128, activation='relu'))
model.add(Dropout(rate=0.25))

model.add(Dense(3, activation='softmax'))

print(model.summary())

# 4. 訓練模型

model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=
['accuracy'])
train_history=model.fit(x_train_normalize, y_train_OneHot,
                        validation_split=0.3,
                        epochs=200, batch_size=512, verbose=1)

# 5. 為訓練集作圖

import matplotlib.pyplot as plt
def show_train_history(train_history):
    plt.plot(train_history.history['accuracy'])
    plt.plot(train_history.history['val_accuracy'])
    plt.title('Train History')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['train', 'test'], loc='upper left')
    plt.show()

show_train_history(train_history)

# 6. 評估模型準確率

scores = model.evaluate(x_test_normalize,y_test_OneHot,verbose=0)
print(scores[:10])


# 7. 進行預測

prediction=model.predict_classes(x_test_normalize)
prediction[:10]

# 7-1. 查看預測結果

label_dict={0:"ikeashark",1:"realshark",2:"other"}

print(label_dict)

import matplotlib.pyplot as plt
def plot_images_labels_prediction(images,labels,prediction,idx,num=3):
    fig = plt.gcf()
    fig.set_size_inches(12,14)
    if num>25: num=25
    for i in range(0, num):
        ax=plt.subplot(5,5, 1+i)
        ax.imshow(images[idx],cmap='binary')

        title=str(i)+','+label_dict[labels[i]]
        if len(prediction)>0:
            title+='=>'+label_dict[prediction[i]]

        ax.set_title(title,fontsize=10)
        ax.set_xticks([]);ax.set_yticks([])
        idx+=1
    plt.show()

plot_images_labels_prediction(x_test_normalize,y_test,prediction,0,20)

# 7-2. 查看預測機率

Predicted_Probability=model.predict(x_test_normalize)

def show_Predicted_Probability(y,prediction,x_img,Predicted_Probability,i):
    print('label:',label_dict[y[i]],
          'predict:',label_dict[prediction[i]])
    plt.figure(figsize=(2,2))
    plt.imshow(np.reshape(x_test[i],(32,32,3)))
    plt.show()
    for j in range(3):
        print(label_dict[j]+ ' Probability:%1.9f'%(Predicted_Probability[i][j]))

for i in range(0,4)

 show_Predicted_Probability(y_test,prediction,x_test_normalize,Predicted_Probability,i)
```