Nguyễn Thái Bình - 19146050 - Nhóm 02CLC - 10_Fruit

```
import numpy as np
from tensorflow import keras
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img,img_to_array
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import os
import matplotlib.pyplot as plt
from skimage import io
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers import Dense,Activation,Dropout,LSTM,BatchNormalization
from keras.layers import Flatten
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.utils import to_categorical
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
```

```
trainset='/content/drive/MyDrive/10_Fruit_data/train'
validationset='/content/drive/MyDrive/10_Fruit_data/validation'
train=ImageDataGenerator(rescale=1/255.0,validation_split=0)
validation=ImageDataGenerator(rescale=1/255.0,validation_split=0.9)
```

```
train_data=train.flow_from_directory(trainset,target_size=(150,150),batch_size=10,class_mode=
validation_set=validation.flow_from_directory(validationset,target_size=(150,150),batch_size=
```

```
Found 148 images belonging to 10 classes.
Found 84 images belonging to 10 classes.
```

```
print(train_data.class_indices)
print(validation_set.class_indices)
```

```
{'cam': 0, 'dao': 1, 'du_du': 2, 'dua_hau': 3, 'khe': 4, 'le': 5, 'man': 6, 'oi': 7, 'sa
{'cam': 0, 'dao': 1, 'du_du': 2, 'dua_hau': 3, 'khe': 4, 'le': 5, 'man': 6, 'oi': 7, 'sa
```

```
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D((2,2)))
```

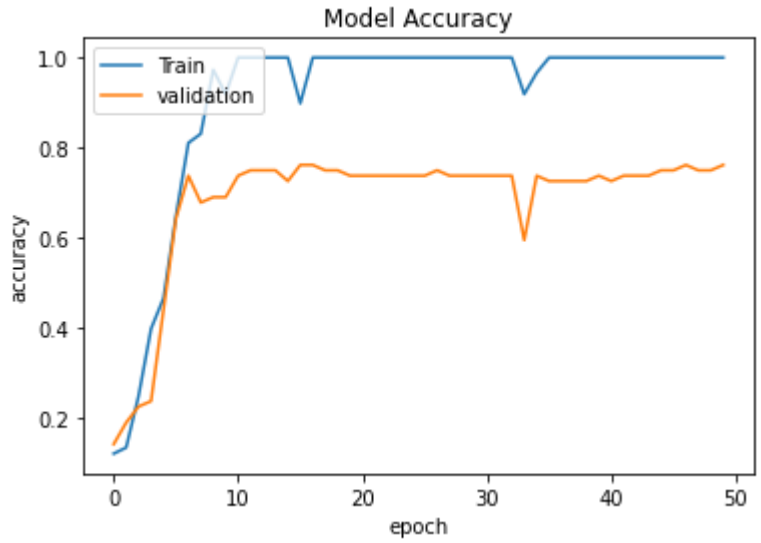
```
model.add(Flatten())
model.add(Dense(128,activation='relu'))
model.add(Dense(10,activation='softmax'))


model.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy'])
history=model.fit(train_data,batch_size=5,epochs=50,verbose=1,validation_data=validation_set)
```

```
Epoch 1/50
15/15 [==============================] - 64s 4s/step - loss: 2.8115 - accuracy: 0.121
Epoch 2/50
15/15 [==============================] - 9s 574ms/step - loss: 2.3121 - accuracy: 0.1
Epoch 3/50
15/15 [==============================] - 9s 571ms/step - loss: 2.1515 - accuracy: 0.2
Epoch 4/50
15/15 [==============================] - 8s 570ms/step - loss: 1.8987 - accuracy: 0.3
Epoch 5/50
15/15 [==============================] - 9s 574ms/step - loss: 1.4748 - accuracy: 0.4
Epoch 6/50
15/15 [==============================] - 9s 573ms/step - loss: 1.1428 - accuracy: 0.6
Epoch 7/50
15/15 [==============================] - 9s 575ms/step - loss: 0.6822 - accuracy: 0.8
Epoch 8/50
15/15 [==============================] - 9s 572ms/step - loss: 0.5630 - accuracy: 0.8
Epoch 9/50
15/15 [==============================] - 9s 572ms/step - loss: 0.1692 - accuracy: 0.9
Epoch 10/50
15/15 [==============================] - 9s 577ms/step - loss: 0.2094 - accuracy: 0.9
Epoch 11/50
15/15 [==============================] - 9s 579ms/step - loss: 0.0259 - accuracy: 1.0
Epoch 12/50
15/15 [==============================] - 9s 577ms/step - loss: 0.0034 - accuracy: 1.0
Epoch 13/50
15/15 [==============================] - 9s 577ms/step - loss: 8.5527e-04 - accuracy:
Epoch 14/50
15/15 [==============================] - 9s 579ms/step - loss: 3.0581e-04 - accuracy:
Epoch 15/50
15/15 [==============================] - 9s 580ms/step - loss: 4.1033e-04 - accuracy:
Epoch 16/50
15/15 [==============================] - 9s 574ms/step - loss: 0.9884 - accuracy: 0.8
Epoch 17/50
15/15 [==============================] - 9s 572ms/step - loss: 0.0074 - accuracy: 1.0
Epoch 18/50
15/15 [==============================] - 9s 572ms/step - loss: 0.0033 - accuracy: 1.0
Epoch 19/50
15/15 [==============================] - 9s 576ms/step - loss: 6.4213e-04 - accuracy:
Epoch 20/50
15/15 [==============================] - 9s 578ms/step - loss: 2.0876e-04 - accuracy:
Epoch 21/50
15/15 [==============================] - 9s 582ms/step - loss: 8.9724e-05 - accuracy:
Epoch 22/50
15/15 [==============================] - 9s 580ms/step - loss: 4.0042e-05 - accuracy:
Epoch 23/50
15/15 [==============================] - 9s 575ms/step - loss: 1.3967e-05 - accuracy:
```

```
Epoch 24/50
15/15 [==============================] - 9s 579ms/step - loss: 7.4529e-06 - accuracy:
Epoch 25/50
15/15 [==============================] - 9s 579ms/step - loss: 3.1236e-06 - accuracy:
Epoch 26/50
15/15 [==============================] - 9s 576ms/step - loss: 1.4015e-06 - accuracy:
Epoch 27/50
15/15 [==============================] - 9s 587ms/step - loss: 6.0249e-07 - accuracy:
Epoch 28/50
15/15 [==============================] - 9s 578ms/step - loss: 3.0527e-07 - accuracy:
Epoch 29/50
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train','validation'],loc='upper left')
plt.show()
```




```
model.save('/content/drive/MyDrive/BT AI/10_Fruit.h5')
```

```
load_model('/content/drive/MyDrive/BT AI/10_Fruit.h5')
```

```
<keras.engine.sequential.Sequential at 0x7f97f6c4fb10>
```

```
generator= ImageDataGenerator(rescale=1./255)
generator_data=generator.flow_from_directory('/content/drive/MyDrive/Test fruit',batch_size=5

fruit={0:'Cam',
       1:'Dao',
       2:'Dudu',
       3:'Duahau',
       4:'Khe',
```

```
      5:'Le',
      6:'Man',
      7:'Oi',
      8:'Sapoche',
      9:'Xoai'}
plt.figure(figsize=(12,12))
for i in range(len(generator_data.filenames)):
    plt.subplot(8,5,i+1)
    plt.imshow(io.imread(os.path.join(generator_data.directory,generator_data.filenames[i])))
    plt.xticks([])
    plt.yticks([])
    img=load_img('/content/drive/MyDrive/Test fruit/'+generator_data.filenames[i],target_size
    img=img_to_array(img)
    img=img.reshape(1,150,150,3)
    img=img.astype('float')
    img=img/255
    plt.xlabel(fruit[np.argmax(model.predict(img))])
plt.show()
```

```
Found 23 images belonging to 1 classes.
```

8 giây hoàn thành lúc 14:31