Nguyễn Thái Bình - 19146050 - Nhóm 02CLC - 11_Money

```
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/11_Money_data/train'
validationset='/content/drive/MyDrive/11_Money_data/validation'
train=ImageDataGenerator(rescale=1/255.0,validation_split=0.1)
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 341 images belonging to 11 classes.
Found 80 images belonging to 11 classes.
```

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

```
{'0.2k': 0, '0.5k': 1, '100k': 2, '10k': 3, '1k': 4, '200k': 5, '20k': 6, '2k': 7, '500k
{'0.2k': 0, '0.5k': 1, '100k': 2, '10k': 3, '1k': 4, '200k': 5, '20k': 6, '2k': 7, '500k
```

```
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)))
```

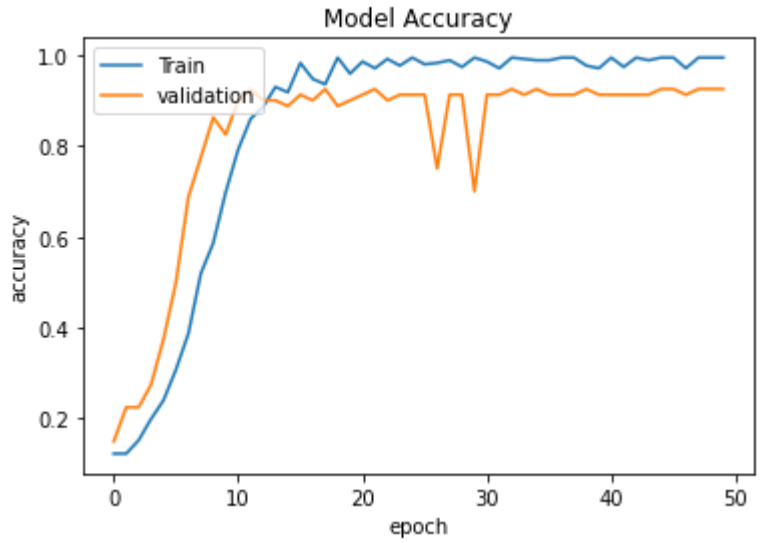
```
model.add(Flatten())
model.add(Dense(128,activation='relu'))
model.add(Dense(11,activation='softmax'))


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)
```

```
35/35 [==============================] - 21s 580ms/step - loss: 0.1737 - accuracy: 0.
Epoch 23/50
35/35 [==============================] - 21s 583ms/step - loss: 0.0431 - accuracy: 0.
Epoch 24/50
35/35 [==============================] - 21s 580ms/step - loss: 0.1778 - accuracy: 0.
Epoch 25/50
35/35 [==============================] - 20s 577ms/step - loss: 0.0442 - accuracy: 0.
Epoch 26/50
35/35 [==============================] - 20s 598ms/step - loss: 0.1006 - accuracy: 0.
Epoch 27/50
35/35 [==============================] - 21s 582ms/step - loss: 0.0720 - accuracy: 0.
Epoch 28/50
35/35 [==============================] - 21s 582ms/step - loss: 0.0772 - accuracy: 0.
Epoch 29/50
35/35 [==============================] - 21s 598ms/step - loss: 0.0807 - accuracy: 0.
Epoch 30/50
35/35 [==============================] - 21s 598ms/step - loss: 0.0552 - accuracy: 0.
Epoch 31/50
35/35 [==============================] - 20s 597ms/step - loss: 0.0950 - accuracy: 0.
Epoch 32/50
35/35 [==============================] - 21s 580ms/step - loss: 0.1578 - accuracy: 0.
Epoch 33/50
35/35 [==============================] - 20s 579ms/step - loss: 0.0507 - accuracy: 0.
Epoch 34/50
35/35 [==============================] - 21s 583ms/step - loss: 0.0742 - accuracy: 0.
Epoch 35/50
35/35 [==============================] - 21s 583ms/step - loss: 0.0419 - accuracy: 0.
Epoch 36/50
35/35 [==============================] - 21s 584ms/step - loss: 0.0487 - accuracy: 0.
Epoch 37/50
35/35 [==============================] - 21s 583ms/step - loss: 0.0198 - accuracy: 0.
Epoch 38/50
35/35 [==============================] - 21s 582ms/step - loss: 0.0231 - accuracy: 0.
Epoch 39/50
35/35 [==============================] - 21s 584ms/step - loss: 0.0911 - accuracy: 0.
Epoch 40/50
35/35 [==============================] - 21s 584ms/step - loss: 0.1010 - accuracy: 0.
Epoch 41/50
35/35 [==============================] - 21s 581ms/step - loss: 0.0213 - accuracy: 0.
Epoch 42/50
35/35 [==============================] - 21s 580ms/step - loss: 0.1352 - accuracy: 0.
Epoch 43/50
35/35 [==============================] - 21s 584ms/step - loss: 0.0155 - accuracy: 0.
Epoch 44/50
35/35 [==============================] - 21s 584ms/step - loss: 0.0751 - accuracy: 0.
Epoch 45/50
35/35 [==============================] - 21s 583ms/step - loss: 0.0125 - accuracy: 0.
```

```
Epoch 46/50
35/35 [==============================] - 21s 583ms/step - loss: 0.0098 - accuracy: 0.
Epoch 47/50
35/35 [==============================] - 21s 582ms/step - loss: 0.2558 - accuracy: 0.
Epoch 48/50
35/35 [==============================] - 21s 581ms/step - loss: 0.0159 - accuracy: 0.
Epoch 49/50
35/35 [==============================] - 21s 582ms/step - loss: 0.0084 - accuracy: 0.
Epoch 50/50
35/35 [==============================] - 21s 585ms/step - loss: 0.0086 - accuracy: 0.
```

```
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/11_Money.h5')
```

```
load_model('/content/drive/MyDrive/BT AI/11_Money.h5')
```

```
<keras.engine.sequential.Sequential at 0x7fc462b33290>
```

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

```
money={0:'200',
       1:'500',
       2:'100k',
       3:'10k',
       4:'1k',
```

```
      5:'200k',
      6:'20k',
      7:'2k',
      8:'500k',
      9:'50k',
      10:'5k'}

plt.figure(figsize=(15,15))
for i in range(len(generator_data.filenames)):
    plt.subplot(5,4,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 money/'+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(money[np.argmax(model.predict(img))])
plt.show()
```

```
Found 17 images belonging to 1 classes.
```

500k

200


200


200

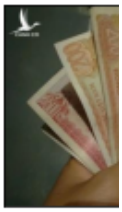
20k


5k

100k

5k

1k



✓ 5 giây hoàn thành lúc 15:15