Đỗ Minh Triều_19146283_FruitsCNN

```
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2
import os
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
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.optimizers import Adam,SGD
from tensorflow.keras.utils import to_categorical
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
import pandas as pd
```

```
load_train_data='../input/fruitcnn/traicay'
```

```
train=ImageDataGenerator(rescale=1/255)
validation=ImageDataGenerator(rescale=1/255,validation_split=0.2,rotation_range=10)
```

```
train_data=train.flow_from_directory(
    load_train_data,
    target_size=(150,150),
    batch_size=100,
    class_mode='categorical',
    subset = 'training'
   )
validation_set=validation.flow_from_directory(
    load_train_data,
    target_size=(150,150),
    batch_size=100,
    class_mode='categorical',
    subset = 'validation'
    )
```

```
 Found 4755 images belonging to 10 classes.
 Found 949 images belonging to 10 classes.
```

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

```
{'Apple Braeburn': 0, 'Banana Lady Finger': 1, 'Blueberry': 2, 'Corn': 3, 'Fig': 4,
{'Apple Braeburn': 0, 'Banana Lady Finger': 1, 'Blueberry': 2, 'Corn': 3, 'Fig': 4,
```

```
model = Sequential()
model.add(Conv2D(32,(3,3), activation = 'relu', kernel_initializer = 'he_uniform', padding
model.add(Conv2D(32,(3,3), activation = 'relu', kernel_initializer = 'he_uniform', padding
model.add(Dropout(0.3))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64,(3,3), activation = 'relu', kernel_initializer = 'he_uniform', padding
model.add(Conv2D(64,(3,3), activation = 'relu', kernel_initializer = 'he_uniform', padding
model.add(Dropout(0.3))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(256,activation='relu',kernel_initializer='he_uniform'))
model.add(Dropout(0.3))
model.add(Dense(10,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
history=model.fit(train_data,batch_size=100,epochs=50,verbose=1,validation_data=validation
```

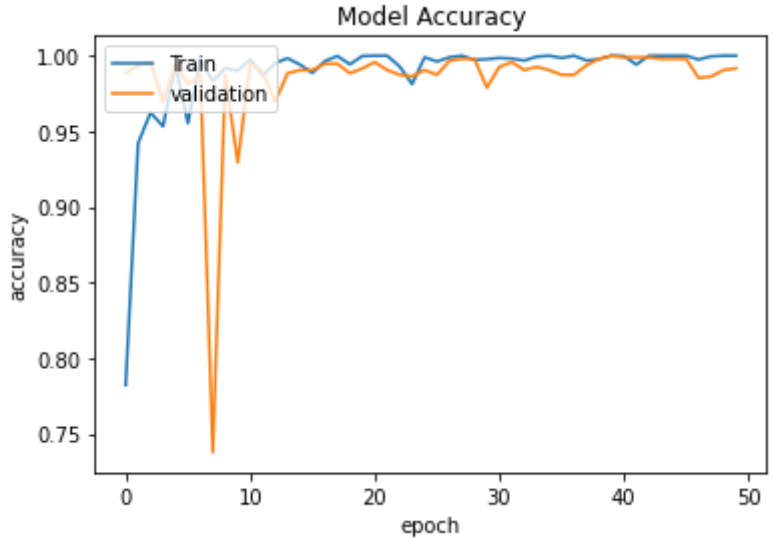
```
48/48 [==============================] - 13s 263ms/step - loss: 1.2712e-06 - accur
Epoch 22/50
48/48 [==============================] - 12s 248ms/step - loss: 7.1752e-07 - accur
Epoch 23/50
48/48 [==============================] - 12s 248ms/step - loss: 0.2286 - accuracy:
Epoch 24/50
48/48 [==============================] - 12s 257ms/step - loss: 3.7635 - accuracy:
Epoch 25/50
48/48 [==============================] - 13s 263ms/step - loss: 0.0280 - accuracy:
Epoch 26/50
48/48 [==============================] - 13s 271ms/step - loss: 0.1231 - accuracy:
Epoch 27/50
48/48 [==============================] - 13s 268ms/step - loss: 0.0116 - accuracy:
Epoch 28/50
48/48 [==============================] - 12s 256ms/step - loss: 6.4803e-06 - accur
Epoch 29/50
48/48 [==============================] - 13s 269ms/step - loss: 0.0952 - accuracy:
Epoch 30/50
48/48 [==============================] - 12s 259ms/step - loss: 0.2482 - accuracy:
Epoch 31/50
48/48 [==============================] - 12s 254ms/step - loss: 0.0284 - accuracy:
Epoch 32/50
48/48 [==============================] - 13s 271ms/step - loss: 0.0319 - accuracy:
Epoch 33/50
48/48 [==============================] - 12s 260ms/step - loss: 0.1608 - accuracy:
Epoch 34/50
48/48 [==============================] - 13s 272ms/step - loss: 0.0567 - accuracy:
Epoch 35/50
48/48 [==============================] - 12s 259ms/step - loss: 1.7549e-10 - accur
Epoch 36/50
48/48 [==============================] - 12s 258ms/step - loss: 0.1012 - accuracy:
Epoch 37/50
48/48 [==============================] - 13s 268ms/step - loss: 7.5211e-11 - accur
Epoch 38/50
48/48 [==============================] - 13s 265ms/step - loss: 0.1469 - accuracy:
Epoch 39/50
```

```
Epoch 39/50
48/48 [==============================] - 13s 275ms/step - loss: 0.0844 - accuracy:
Epoch 40/50
48/48 [==============================] - 12s 256ms/step - loss: 0.0000e+00 - accur
Epoch 41/50
48/48 [==============================] - 12s 253ms/step - loss: 0.0122 - accuracy:
Epoch 42/50
48/48 [==============================] - 13s 265ms/step - loss: 0.7324 - accuracy:
Epoch 43/50
48/48 [==============================] - 12s 259ms/step - loss: 0.0000e+00 - accur
Epoch 44/50
48/48 [==============================] - 13s 260ms/step - loss: 0.0000e+00 - accur
Epoch 45/50
48/48 [==============================] - 13s 268ms/step - loss: 0.0000e+00 - accur
Epoch 46/50
48/48 [==============================] - 13s 266ms/step - loss: 0.0000e+00 - accur
Epoch 47/50
48/48 [==============================] - 13s 271ms/step - loss: 0.2323 - accuracy:
Epoch 48/50
48/48 [==============================] - 13s 263ms/step - loss: 0.0255 - accuracy:
Epoch 49/50
48/48 [==============================] - 12s 257ms/step - loss: 0.0000e+00 - accur
Epoch 50/50
```

```
score=model.evaluate(validation_set,verbose=0)
print('Sai số khi kiểm tra của mô hình là:',score[0])
print('Độ chính xác khi kiểm tra của mô hình là:',score[1])
```

```
Sai số khi kiểm tra của mô hình là: 0.2886275351047516
Độ chính xác khi kiểm tra của mô hình là: 0.9884088635444641
```

```
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()
```




```
fruit={0:'Apple Braeburn',
```

```
         1:'Banana Lady Finger',
         2:'Blueberry',
         3:'Corn',
         4:'Fig',
         5:'Ginger Root',
         6:'Lychee',
         7:'Onion Red Peeled',
         8:'Strawberry',
         9:'Watermelon'}

import os
filenames=os.listdir("../input/fruittest/fruittest")
df=pd.DataFrame({'filename':filenames})
url='../input/fruittest/fruittest/'+df['filename']

plt.figure(figsize=(20,20))
for i in range(df.shape[0]):
  plt.subplot(10,10,i+1)
  plt.grid(False)
  plt.xticks([])
  plt.yticks([])
  img=load_img(url[i],target_size=(150,150))
  plt.imshow(img)
  img=img_to_array(img)
  img=img.reshape(1,150,150,3)
  img=img/255.0
  img_pred = model.predict(img)
  plt.xlabel(fruit[np.argmax(img_pred)])
plt.show()
```


```
model.save('./fruitcnn.h5')
```