

```

import numpy as np #tinh toan
import pandas as pd #xu li bang
import seaborn as sns #do thi du lieu
import matplotlib.pyplot as plt # ve
from sklearn.preprocessing import StandardScaler # xu li xu lieu cho dong deu
from sklearn.model_selection import train_test_split # chia du lieu
from keras.layers import Dense,Activation,Dropout,BatchNormalization,LSTM # nhom chuan
from keras.models import Sequential #
from tensorflow.keras.utils import to_categorical ##### chỉ cho 1 giá trị (1) hiện lên
from keras import callbacks #
from sklearn.metrics import precision_score, recall_score, confusion_matrix, classification_r
from tensorflow.keras.optimizers import RMSprop
from keras.callbacks import EarlyStopping
from sklearn.preprocessing import scale
from keras.utils import np_utils
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import RMSprop
import numpy as np
from tensorflow.keras.optimizers import SGD,Adam
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
import os

```

```
url='/content/drive/MyDrive/Colab Notebooks/FOOD_TOUR_VN'
```

```

generator= ImageDataGenerator(rescale=1./255,validation_split=0.1)
train_data=generator.flow_from_directory(url,target_size=(150,150),batch_size=10,class_mode='
validation_dataset=generator.flow_from_directory(url,target_size=(150,150),batch_size=10,clas

```

```

    Found 964 images belonging to 10 classes.
    Found 104 images belonging to 10 classes.

```

```
validation_dataset.class_indices
```

```

{'BANH_CHUNG': 0,
 'BANH_MI': 1,
 'BANH_XEO': 2,
 'BUN_BO': 3,

```

```
'CANH_CHUA': 4,
'CAO_LAU': 5,
'CHAO_LONG': 6,
'COM_TAM': 7,
'NEM_CHUA': 8,
'PHO': 9}
```

```
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',ir
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model=Sequential()
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model=Sequential()
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model.add(Flatten())
model.add(Dense(256,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

```
opt=SGD(learning_rate=0.002,momentum=0.9)
model.compile(optimizer=opt,loss='categorical_crossentropy',metrics=['accuracy'])
history=model.fit(train_data,epochs=5,batch_size=128,validation_data=validation_dataset,verbo
score=model.evaluate(validation_dataset,verbose=0)
print('Sai số kiểm tra là:',score[0])
print('Độ chính xác kiểm tra là:',score[1])
```

```
Epoch 1/5
97/97 [=====] - 607s 6s/step - loss: 2.8765 - accuracy: 0.1981
Epoch 2/5
97/97 [=====] - 604s 6s/step - loss: 1.7725 - accuracy: 0.3869
Epoch 3/5
97/97 [=====] - 606s 6s/step - loss: 1.0179 - accuracy: 0.6857
Epoch 4/5
97/97 [=====] - 609s 6s/step - loss: 0.3295 - accuracy: 0.9035
Epoch 5/5
97/97 [=====] - 608s 6s/step - loss: 0.0897 - accuracy: 0.9793
Sai số kiểm tra là: 2.8845608234405518
Độ chính xác kiểm tra là: 0.3365384638309479
```

```
model.save('/content/drive/MyDrive/Colab Notebooks/MOHINH_H5/FOOD_TOUR_VN.h5')
```

```
img=load_img('banhchung_test.jpg',target_size=(150,150)) #'BANH_CHUNG': 0,'BANH_MI': 1,'BANH_XEC'  
#'CAO_LAU': 5,'CHAO_LONG': 6,'COM_TAM': 7,'NEM_CHUA': 8,'PHO': 9  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=1)
```

```
array([0])
```

```
img=load_img('banhmi_test.jpg',target_size=(150,150)) #'BANH_CHUNG': 0,'BANH_MI': 1,'BANH_XEC'  
#'CAO_LAU': 5,'CHAO_LONG': 6,'COM_TAM': 7,'NEM_CHUA': 8,'PHO': 9  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=1)
```

```
array([7])
```

```
img=load_img('comtam_test.jpg',target_size=(150,150)) #'BANH_CHUNG': 0,'BANH_MI': 1,'BANH_XEC'  
#'CAO_LAU': 5,'CHAO_LONG': 6,'COM_TAM': 7,'NEM_CHUA': 8,'PHO': 9  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=1)
```

```
array([7])
```

✓ 0s completed at 11:20 AM

● ✕

```

import numpy as np #tinh toan
import pandas as pd #xu li bang
import seaborn as sns #do thi du lieu
import matplotlib.pyplot as plt # ve
from sklearn.preprocessing import StandardScaler # xu li xu lieu cho dong deu
from sklearn.model_selection import train_test_split # chia du lieu
from keras.layers import Dense,Activation,Dropout,BatchNormalization,LSTM # nhom chuan
from keras.models import Sequential #
from tensorflow.keras.utils import to_categorical ##### chỉ cho 1 giá trị (1) hiện lên
from keras import callbacks #
from sklearn.metrics import precision_score, recall_score, confusion_matrix, classification_r
from tensorflow.keras.optimizers import RMSprop
from keras.callbacks import EarlyStopping
from sklearn.preprocessing import scale
from keras.utils import np_utils
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import RMSprop
import numpy as np
from tensorflow.keras.optimizers import SGD,Adam
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
import os

```

```
url='/content/drive/MyDrive/Colab Notebooks/FRUIT'
```

```

generator= ImageDataGenerator(rescale=1./255,validation_split=0.1)
train_data=generator.flow_from_directory(url,target_size=(150,150),batch_size=10,class_mode='
validation_dataset=generator.flow_from_directory(url,target_size=(150,150),batch_size=10,clas

```

```

    Found 865 images belonging to 10 classes.
    Found 90 images belonging to 10 classes.

```

```
validation_dataset.class_indices
```

```

{'APPLE': 0,
 'BANANA': 1,
 'CORN': 2,
 'LEMON': 3,

```

```
'LYCHEE': 4,
'PEACH': 5,
'RASBERRY': 6,
'STRAWBERRY': 7,
'TOMATO': 8,
'WATERMELON': 9}
```

```
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',in
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model=Sequential()
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model=Sequential()
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model.add(Flatten())
model.add(Dense(256,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

```
opt=Adam(lr=0.002)
model.compile(optimizer=opt,loss='categorical_crossentropy', metrics=['accuracy'])
history=model.fit(train_data,epochs=5,batch_size=256,validation_data=validation_dataset,verbo
score=model.evaluate(validation_dataset,verbose=0)
print('Sai số kiểm tra là:',score[0])
print('Độ chính xác kiểm tra là:',score[1])
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The
super(Adam, self).__init__(name, **kwargs)
Epoch 1/5
87/87 [=====] - 512s 6s/step - loss: 23.7610 - accuracy: 0.7376
Epoch 2/5
87/87 [=====] - 502s 6s/step - loss: 0.2155 - accuracy: 0.9734
Epoch 3/5
87/87 [=====] - 505s 6s/step - loss: 0.0998 - accuracy: 0.9884
Epoch 4/5
87/87 [=====] - 500s 6s/step - loss: 9.4639e-04 - accuracy: 1.0000
Epoch 5/5
87/87 [=====] - 505s 6s/step - loss: 6.1335e-05 - accuracy: 1.0000
Sai số kiểm tra là: 0.6242494583129883
Độ chính xác kiểm tra là: 0.9111111164093018
```

```
#'APPLE': 0, 'BANANA': 1, 'CORN': 2, 'LEMON': 3, 'LYCHEE': 4, 'PEACH': 5, 'RASBERRY': 6, 'STRAWBERRY'  
img=load_img('peach_test.jpg',target_size=(150,150))  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=1)
```

```
array([5])
```

```
img=load_img('tomato_test.jpg',target_size=(150,150))  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=1)
```

```
array([6])
```

```
img=load_img('watermelon_test.jpg',target_size=(150,150))  
img=img_to_array(img)  
img=img.reshape(1,150,150,3)  
img=img.astype('float32')  
img=img/255  
np.argmax(model.predict(img),axis=1)
```

```
📄 array([9])
```

```

import numpy as np #tinh toan
import pandas as pd #xu li bang
import seaborn as sns #do thi du lieu
import matplotlib.pyplot as plt # ve
from sklearn.preprocessing import StandardScaler # xu li xu lieu cho dong deu
from sklearn.model_selection import train_test_split # chia du lieu
from keras.layers import Dense,Activation,Dropout,BatchNormalization,LSTM # nhom chuan
from keras.models import Sequential #
from tensorflow.keras.utils import to_categorical ##### chỉ cho 1 giá trị (1) hiện lên
from keras import callbacks #
from sklearn.metrics import precision_score, recall_score, confusion_matrix, classification_r
from tensorflow.keras.optimizers import RMSprop
from keras.callbacks import EarlyStopping
from sklearn.preprocessing import scale
from keras.utils import np_utils
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import RMSprop
import numpy as np
from tensorflow.keras.optimizers import SGD,Adam
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
import os

```

Saved successfully!



otebooks/CURRENT_VN'

```

generator= ImageDataGenerator(rescale=1./255,validation_split=0.1)
train_data=generator.flow_from_directory(url,target_size=(150,150),batch_size=10,class_mode='
validation_dataset=generator.flow_from_directory(url,target_size=(150,150),batch_size=10,clas

```

```

Found 101 images belonging to 11 classes.
Found 11 images belonging to 11 classes.

```

```
validation_dataset.class_indices
```

```

{'100K': 0,
 '10K': 1,
 '1K': 2,
 '200K': 3,

```



```
'200Đ': 4,
'20K': 5,
'2K': 6,
'500K': 7,
'500Đ': 8,
'50K': 9,
'5K': 10}
```

```
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',ir
model.add(MaxPooling2D(2,2))
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model=Sequential()
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

```
model=Sequential()
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
```

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

```
opt=Adam(lr=0.002)
model.compile(optimizer=opt,loss='categorical_crossentropy', metrics=['accuracy'])
history=model.fit(train_data,epochs=15,batch_size=128,validation_data=validation_dataset,vert
dataset,verbose=0)
print('Độ chính xác kiểm tra là:',score[1])
```

Saved successfully!

Epoch 1/15

/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The
super(Adam, self).__init__(name, **kwargs)

11/11 [=====] - 51s 5s/step - loss: 0.1596 - accuracy: 0.9703

Epoch 2/15

11/11 [=====] - 50s 5s/step - loss: 0.4443 - accuracy: 0.9505

Epoch 3/15

11/11 [=====] - 50s 5s/step - loss: 0.6753 - accuracy: 0.8911

Epoch 4/15

11/11 [=====] - 54s 5s/step - loss: 0.7201 - accuracy: 0.9307

Epoch 5/15

11/11 [=====] - 50s 5s/step - loss: 0.6343 - accuracy: 0.8317

```

Epoch 6/15
11/11 [=====] - 50s 5s/step - loss: 0.5325 - accuracy: 0.8713
Epoch 7/15
11/11 [=====] - 50s 5s/step - loss: 0.1790 - accuracy: 0.9703
Epoch 8/15
11/11 [=====] - 50s 5s/step - loss: 0.0156 - accuracy: 1.0000
Epoch 9/15
11/11 [=====] - 50s 5s/step - loss: 0.0386 - accuracy: 1.0000
Epoch 10/15
11/11 [=====] - 50s 5s/step - loss: 0.0087 - accuracy: 1.0000
Epoch 11/15
11/11 [=====] - 50s 5s/step - loss: 2.1827e-04 - accuracy: 1.0
Epoch 12/15
11/11 [=====] - 51s 5s/step - loss: 1.6688e-06 - accuracy: 1.0
Epoch 13/15
11/11 [=====] - 50s 5s/step - loss: 1.6375e-05 - accuracy: 1.0
Epoch 14/15
11/11 [=====] - 50s 5s/step - loss: 1.0327e-06 - accuracy: 1.0
Epoch 15/15
11/11 [=====] - 50s 5s/step - loss: 1.0977e-07 - accuracy: 1.0
Sai số kiểm tra là: 10.136160850524902
Độ chính xác kiểm tra là: 0.4545454680919647

```

```

# '100K': 0, '10K': 1, '1K': 2, '200K': 3, '200Đ': 4, '20K': 5, '2K': 6, '500K': 7, '500Đ': 8, '50K': 9
img=load_img('20k_test.jpg',target_size=(150,150))
img=img_to_array(img)
img=img.reshape(1,150,150,3)
img=img.astype('float32')
img=img/255
np.argmax(model.predict(img),axis=1)

```

```
array([5])
```

```

img=load_img('50k_test.jpg',target_size=(150,150))
img=img_to_array(img)

```

Saved successfully!

```

img=img/255
np.argmax(model.predict(img),axis=1)

```

```
array([7])
```

```

img=load_img('100k_test.jpg',target_size=(150,150))
img=img_to_array(img)
img=img.reshape(1,150,150,3)
img=img.astype('float32')
img=img/255
np.argmax(model.predict(img),axis=1)

```

```
array([0])
```

✓ 0s completed at 4:17 PM ● ✕

Saved successfully! ✕