Họ và Tên : Võ Nguyễn Phú Thịnh
MSSV : 19146271
Link-Github : https://github.com/chjf123456789/19146271.git

CNNFruit

May 18, 2022

[53]:
```
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.layers import Flatten, Dense, Dropout, Activation
from google.colab import drive
```

[54]:
```
drive.mount('/content/drive',force_remount=True)
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
train=train_datagen.flow_from_directory('/content/drive/MyDrive/train',
                                              target_size=(256,256),
                                              batch_size=32,
                                              class_mode ='categorical')
test=train_datagen.flow_from_directory('/content/drive/MyDrive/test1',
                                             target_size=(256,256),
                                             batch_size=32,
                                             class_mode ='categorical')
```

```
Mounted at /content/drive
Found 48 images belonging to 10 classes.
Found 20 images belonging to 10 classes.
```

[55]:
```
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[56]:
```
model=Sequential()
model.
 ↪add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',input
model.add(MaxPooling2D(pool_size=(2,2)))
model.
 ↪add(Conv2D(64,(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.add(Flatten())
model.add(Dense(128,activation='relu',kernel_initializer = 'he_uniform'))
#model.add(Dropout(0,2))
model.add(Dense(10,activation='Softmax'))
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.callbacks import EarlyStopping
#opt = SGD(lr = 0.01, momentum = 0.9)
model.compile(optimizer = 'adam', loss ='categorical_crossentropy',metrics =␣
 ↪['accuracy'])
callbacks=[EarlyStopping(monitor='val_loss',patience=100)]
history=model.fit(train,
                  steps_per_epoch=len(train),
                  batch_size = 64,
                  epochs=50,
                  validation_data=test,
                  validation_steps=len(test),
                  callbacks=callbacks,
                  verbose = 1)
```

```
Epoch 1/50
2/2 [==============================] - 7s 5s/step - loss: 26.9320 - accuracy:
0.0833 - val_loss: 22.7313 - val_accuracy: 0.1500
Epoch 2/50
2/2 [==============================] - 5s 2s/step - loss: 19.0183 - accuracy:
0.1250 - val_loss: 6.1188 - val_accuracy: 0.1000
Epoch 3/50
2/2 [==============================] - 3s 2s/step - loss: 3.8251 - accuracy:
0.1042 - val_loss: 2.8134 - val_accuracy: 0.1000
Epoch 4/50
2/2 [==============================] - 2s 985ms/step - loss: 2.5122 - accuracy:
0.1458 - val_loss: 2.2533 - val_accuracy: 0.2000
Epoch 5/50
2/2 [==============================] - 2s 1s/step - loss: 2.2528 - accuracy:
0.1042 - val_loss: 2.2392 - val_accuracy: 0.1500
Epoch 6/50
2/2 [==============================] - 2s 2s/step - loss: 2.2668 - accuracy:
0.1667 - val_loss: 2.2018 - val_accuracy: 0.2000
Epoch 7/50
2/2 [==============================] - 4s 3s/step - loss: 2.0472 - accuracy:
0.3542 - val_loss: 2.0598 - val_accuracy: 0.2000
Epoch 8/50
2/2 [==============================] - 2s 2s/step - loss: 2.3673 - accuracy:
0.2500 - val_loss: 2.2420 - val_accuracy: 0.1000
Epoch 9/50
```

```
2/2 [==============================] - 2s 2s/step - loss: 2.1531 - accuracy:
0.1875 - val_loss: 2.1130 - val_accuracy: 0.2500
Epoch 10/50
2/2 [==============================] - 2s 1s/step - loss: 1.8890 - accuracy:
0.3750 - val_loss: 1.9804 - val_accuracy: 0.2500
Epoch 11/50
2/2 [==============================] - 2s 2s/step - loss: 1.7227 - accuracy:
0.4167 - val_loss: 1.8396 - val_accuracy: 0.2500
Epoch 12/50
2/2 [==============================] - 2s 2s/step - loss: 1.4337 - accuracy:
0.4792 - val_loss: 1.6758 - val_accuracy: 0.3000
Epoch 13/50
2/2 [==============================] - 2s 996ms/step - loss: 1.2073 - accuracy:
0.6250 - val_loss: 1.8741 - val_accuracy: 0.2000
Epoch 14/50
2/2 [==============================] - 2s 1s/step - loss: 1.0129 - accuracy:
0.6250 - val_loss: 1.9103 - val_accuracy: 0.3000
Epoch 15/50
2/2 [==============================] - 2s 2s/step - loss: 0.7869 - accuracy:
0.7917 - val_loss: 1.7886 - val_accuracy: 0.3500
Epoch 16/50
2/2 [==============================] - 2s 2s/step - loss: 0.6814 - accuracy:
0.8125 - val_loss: 1.6292 - val_accuracy: 0.3000
Epoch 17/50
2/2 [==============================] - 2s 1s/step - loss: 0.5875 - accuracy:
0.8333 - val_loss: 1.9560 - val_accuracy: 0.3500
Epoch 18/50
2/2 [==============================] - 2s 2s/step - loss: 0.4534 - accuracy:
0.8750 - val_loss: 1.6596 - val_accuracy: 0.3000
Epoch 19/50
2/2 [==============================] - 2s 2s/step - loss: 0.4009 - accuracy:
0.9167 - val_loss: 1.9364 - val_accuracy: 0.3500
Epoch 20/50
2/2 [==============================] - 2s 1s/step - loss: 0.2458 - accuracy:
0.9583 - val_loss: 1.7112 - val_accuracy: 0.5000
Epoch 21/50
2/2 [==============================] - 2s 1s/step - loss: 0.3895 - accuracy:
0.8542 - val_loss: 1.8460 - val_accuracy: 0.5000
Epoch 22/50
2/2 [==============================] - 2s 2s/step - loss: 0.1189 - accuracy:
1.0000 - val_loss: 1.7692 - val_accuracy: 0.4500
Epoch 23/50
2/2 [==============================] - 2s 1s/step - loss: 0.1110 - accuracy:
1.0000 - val_loss: 1.6497 - val_accuracy: 0.3500
Epoch 24/50
2/2 [==============================] - 2s 2s/step - loss: 0.0992 - accuracy:
0.9792 - val_loss: 2.4550 - val_accuracy: 0.4500
Epoch 25/50
```

```
2/2 [==============================] - 2s 2s/step - loss: 0.1386 - accuracy:
0.9583 - val_loss: 2.0278 - val_accuracy: 0.5000
Epoch 26/50
2/2 [==============================] - 2s 2s/step - loss: 0.1242 - accuracy:
0.9583 - val_loss: 1.8718 - val_accuracy: 0.3500
Epoch 27/50
2/2 [==============================] - 2s 2s/step - loss: 0.1125 - accuracy:
0.9792 - val_loss: 2.5232 - val_accuracy: 0.3500
Epoch 28/50
2/2 [==============================] - 2s 1s/step - loss: 0.2167 - accuracy:
0.9167 - val_loss: 2.2803 - val_accuracy: 0.4000
Epoch 29/50
2/2 [==============================] - 3s 1s/step - loss: 0.1482 - accuracy:
0.9583 - val_loss: 2.5548 - val_accuracy: 0.4500
Epoch 30/50
2/2 [==============================] - 2s 2s/step - loss: 0.1397 - accuracy:
0.9375 - val_loss: 2.3721 - val_accuracy: 0.3500
Epoch 31/50
2/2 [==============================] - 2s 2s/step - loss: 0.2261 - accuracy:
0.9583 - val_loss: 2.2710 - val_accuracy: 0.3000
Epoch 32/50
2/2 [==============================] - 2s 2s/step - loss: 0.2396 - accuracy:
0.9375 - val_loss: 2.2715 - val_accuracy: 0.5000
Epoch 33/50
2/2 [==============================] - 2s 1s/step - loss: 0.0762 - accuracy:
0.9583 - val_loss: 2.7368 - val_accuracy: 0.4500
Epoch 34/50
2/2 [==============================] - 2s 2s/step - loss: 0.1704 - accuracy:
0.9583 - val_loss: 3.0861 - val_accuracy: 0.3500
Epoch 35/50
2/2 [==============================] - 2s 2s/step - loss: 0.2951 - accuracy:
0.8750 - val_loss: 2.9272 - val_accuracy: 0.4000
Epoch 36/50
2/2 [==============================] - 2s 2s/step - loss: 0.2649 - accuracy:
0.9167 - val_loss: 2.4136 - val_accuracy: 0.4000
Epoch 37/50
2/2 [==============================] - 2s 2s/step - loss: 0.7057 - accuracy:
0.7708 - val_loss: 1.6669 - val_accuracy: 0.4000
Epoch 38/50
2/2 [==============================] - 2s 1s/step - loss: 0.2149 - accuracy:
0.9375 - val_loss: 2.5152 - val_accuracy: 0.4000
Epoch 39/50
2/2 [==============================] - 2s 2s/step - loss: 0.2266 - accuracy:
0.9167 - val_loss: 3.4246 - val_accuracy: 0.3500
Epoch 40/50
2/2 [==============================] - 2s 2s/step - loss: 0.4013 - accuracy:
0.8750 - val_loss: 2.2826 - val_accuracy: 0.4500
Epoch 41/50
```

```
2/2 [==============================] - 2s 2s/step - loss: 0.3389 - accuracy:
0.9167 - val_loss: 1.8615 - val_accuracy: 0.4500
Epoch 42/50
2/2 [==============================] - 2s 2s/step - loss: 0.0846 - accuracy:
0.9792 - val_loss: 1.9804 - val_accuracy: 0.5500
Epoch 43/50
2/2 [==============================] - 2s 1s/step - loss: 0.0419 - accuracy:
1.0000 - val_loss: 2.7068 - val_accuracy: 0.4000
Epoch 44/50
2/2 [==============================] - 2s 2s/step - loss: 0.1346 - accuracy:
0.9792 - val_loss: 2.6812 - val_accuracy: 0.4500
Epoch 45/50
2/2 [==============================] - 2s 2s/step - loss: 0.0623 - accuracy:
0.9792 - val_loss: 3.1316 - val_accuracy: 0.4000
Epoch 46/50
2/2 [==============================] - 2s 2s/step - loss: 0.1741 - accuracy:
0.9792 - val_loss: 3.1748 - val_accuracy: 0.3500
Epoch 47/50
2/2 [==============================] - 2s 1s/step - loss: 0.0872 - accuracy:
0.9792 - val_loss: 3.1480 - val_accuracy: 0.3000
Epoch 48/50
2/2 [==============================] - 2s 2s/step - loss: 0.1012 - accuracy:
0.9792 - val_loss: 2.7172 - val_accuracy: 0.3500
Epoch 49/50
2/2 [==============================] - 2s 1s/step - loss: 0.0459 - accuracy:
0.9792 - val_loss: 2.5316 - val_accuracy: 0.4500
Epoch 50/50
2/2 [==============================] - 2s 2s/step - loss: 0.0075 - accuracy:
1.0000 - val_loss: 2.7316 - val_accuracy: 0.4000
```

[57]:
```
#đánh giá chất lượng của mô hình và vẽ lại
score = model.evaluate(test,verbose=0)
print('Sai số kiểm tra là: ',score[0])
print('Độ chính xác kiểm tra là: ',score[1])
```

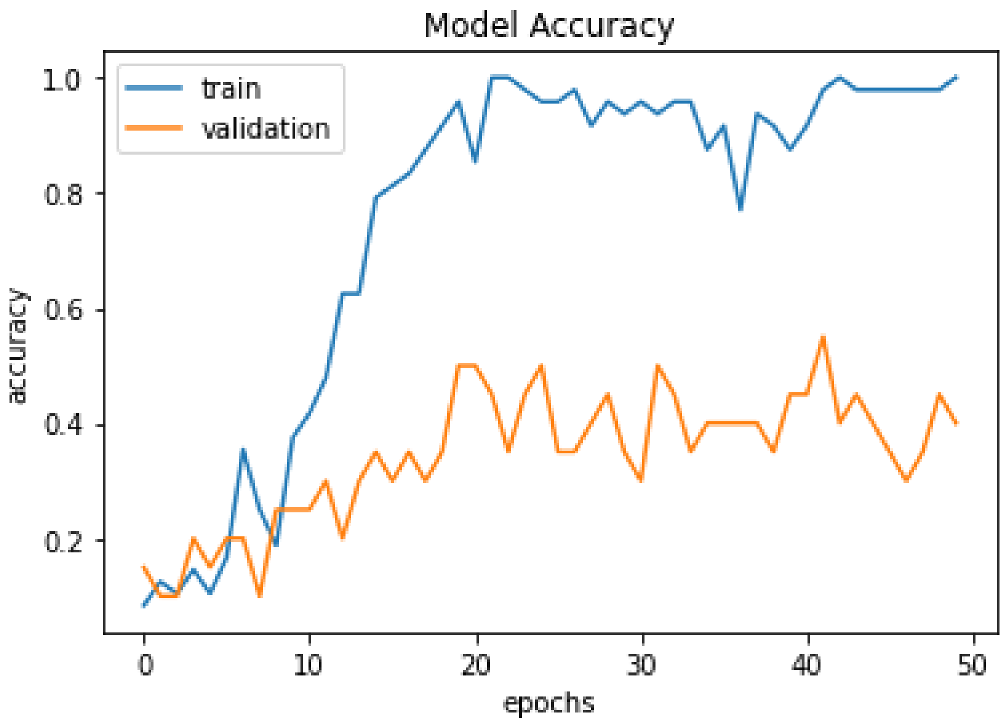
```
Sai số kiểm tra là:  2.5663399696350098
Độ chính xác kiểm tra là:  0.44999998807907104
```

[58]:
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epochs')
plt.legend(['train','validation'],loc='upper_left')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6:
MatplotlibDeprecationWarning: Unrecognized location 'upper_left'. Falling back
on 'best'; valid locations are
```

```
        best
        upper right
        upper left
        lower left
        lower right
        right
        center left
        center right
        lower center
        upper center
        center
This will raise an exception in 3.3.
```

[58]:
```
<matplotlib.legend.Legend at 0x7f119f783190>
```



[59]:
```python
model.save('model_fruit.h5')
```

[60]:
```python
from tensorflow.keras.models import load_model
model=load_model('model_fruit.h5')
```

[61]:
```python
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math
```

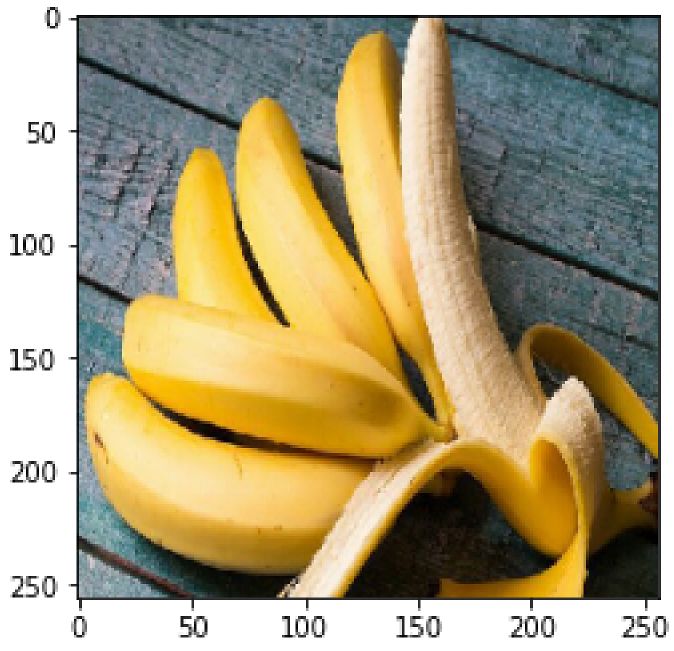
```
[79]: from keras.preprocessing.image import load_img
      from keras.preprocessing.image import img_to_array
      from google.colab import files
      uploadfile=files.upload()
```

```
<IPython.core.display.HTML object>

Saving chuoi.jpg to chuoi (1).jpg
```

```
[64]: fruit = ['bưởi','cam','chuối','kiwi','mận','mít','nhãn','táo','thanh␣
       ↪long','xoài']
```

```
[80]: img = load_img("chuoi.jpg",target_size = (256,256))
      plt.imshow(img)
      img=img_to_array(img)
      img=img.reshape(1,256,256,3)
      img=img.astype('float32')
      img=img/255
```



```
[81]: np.argmax(model.predict(img),axis=1)
```

```
[81]: array([2])
```

```
[ ]: from google.colab import drive
     drive.mount('/content/drive')
     !wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
     from colab_pdf import colab_pdf
     colab_pdf('CNNFruit.ipynb')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
--2022-05-18 07:43:52--  https://raw.githubusercontent.com/brpy/colab-
pdf/master/colab_pdf.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.108.133, 185.199.109.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 1864 (1.8K) [text/plain]
Saving to: ‘colab_pdf.py’

colab_pdf.py        100%[===================>]   1.82K  --.-KB/s    in 0s

2022-05-18 07:43:53 (27.8 MB/s) - ‘colab_pdf.py’ saved [1864/1864]


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```

```
[ ]:
```