TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA CƠ KHÍ CHẾ TẠO MÁY

~~~~~~~~*~~~~~~~~

BÀI TẬP VỀ NHÀ MÔN AI

GVHD: ThS. Nguyễn Trường Thịnh

SVTH:

Trần Anh Tú – 19146416

Thành phố Hồ Chí Minh, tháng 12 năm 2021
~~~~~~~~

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import glob
import cv2
import os
from os import listdir
from numpy import asarray
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from keras.layers import Dense,Activation,BatchNormalization, Dropout, LSTM, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from tensorflow.keras.optimizers import RMSprop, SGD
from keras.utils import np_utils
from tensorflow.keras.preprocessing.image import load_img, img_to_array
```

```
folder1 = '/content/drive/MyDrive/FACES/TAT'
folder2 = '/content/drive/MyDrive/FACES/TLUAN/dataSet'
folder3 = '/content/drive/MyDrive/TAN'
photos, labels = list(), list()
for file in listdir(folder1):
  photo = load_img(folder1+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(0)
for file in listdir(folder2):
  photo = load_img(folder2+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(1)
for file in listdir(folder3):
  photo = load_img(folder3+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(2)
photos = asarray(photos)
labels = asarray(labels)
class_name = ['Anh Tú','Tấn Luân','Nhật Tấn']
print(photos.shape, labels.shape, class_name)

```

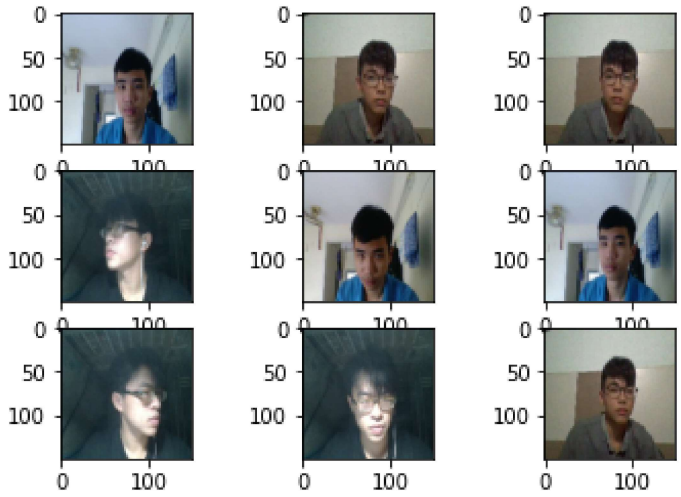
```
(596, 150, 150, 3) (596,) ['Anh Tú', 'Tấn Luân', 'Nhật Tấn']
```

Nhấp đúp (hoặc nhấn Enter) để chỉnh sửa

```
x_train, x_test, y_train, y_test = train_test_split(photos, labels, test_size=0.25, ranc
print(x_train.shape,x_test.shape)
print(y_train.shape,y_test.shape)
```

```
(447, 150, 150, 3) (149, 150, 150, 3)
(447,) (149,)
```

```
x_train_show = x_train.astype(int)
import matplotlib.pyplot as plt
from matplotlib.image import imread
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_train_show[i])
plt.show()
```



```
x_train = x_train.reshape(447, 67500)
x_test = x_test.reshape(149, 67500)
x_train/=255
x_test/=255
y_train=np_utils.to_categorical(y_train)
y_test=np_utils.to_categorical(y_test)

```

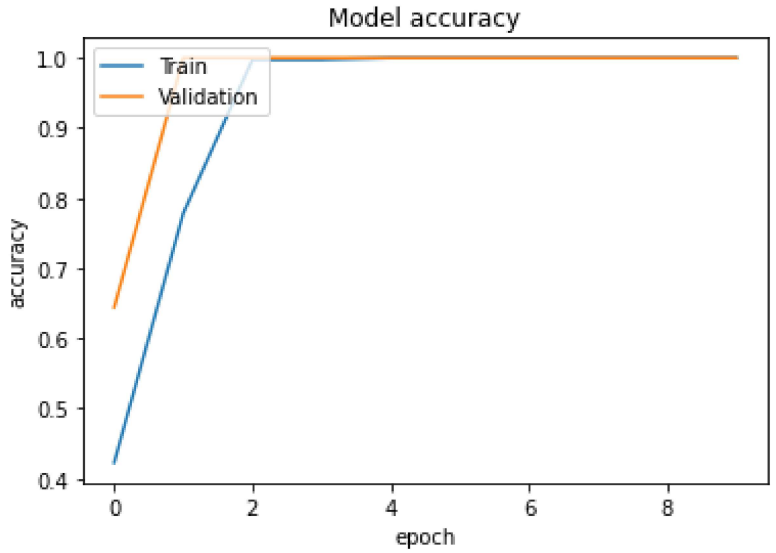
```
model = Sequential()
model.add(Dense(1024, kernel_initializer='normal',activation='relu', input_shape = (6750
model.add(Dense(128, activation='relu'))
model.add(Dense(3, activation='softmax'))

```

```
model.compile(loss='categorical_crossentropy', optimizer=RMSprop(), metrics=['accuracy']
history = model.fit(x_train,y_train, batch_size=64,epochs=10,verbose=1,validation_data =
```

```
Epoch 1/10
7/7 [==============================] - 4s 109ms/step - loss: 202.9695 - accuracy: 0.4
Epoch 2/10
7/7 [==============================] - 0s 45ms/step - loss: 12.9348 - accuracy: 0.778
Epoch 3/10
7/7 [==============================] - 0s 50ms/step - loss: 0.0438 - accuracy: 0.9978
Epoch 4/10
7/7 [==============================] - 0s 52ms/step - loss: 0.0042 - accuracy: 0.9978
Epoch 5/10
7/7 [==============================] - 0s 45ms/step - loss: 5.3337e-10 - accuracy: 1.
Epoch 6/10
7/7 [==============================] - 0s 45ms/step - loss: 5.3337e-10 - accuracy: 1.
Epoch 7/10
7/7 [==============================] - 0s 44ms/step - loss: 5.3337e-10 - accuracy: 1.
Epoch 8/10
7/7 [==============================] - 0s 61ms/step - loss: 5.3337e-10 - accuracy: 1.
Epoch 9/10
7/7 [==============================] - 0s 64ms/step - loss: 5.3337e-10 - accuracy: 1.
Epoch 10/10
7/7 [==============================] - 0s 51ms/step - loss: 5.3337e-10 - accuracy: 1.
```

```
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')
```

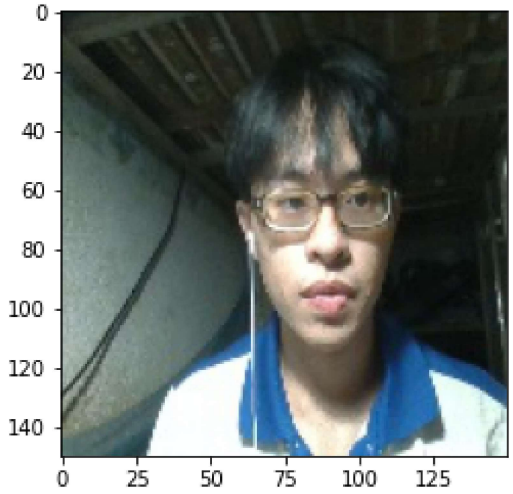
```
<matplotlib.legend.Legend at 0x7fc355f4b750>
```




```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
```

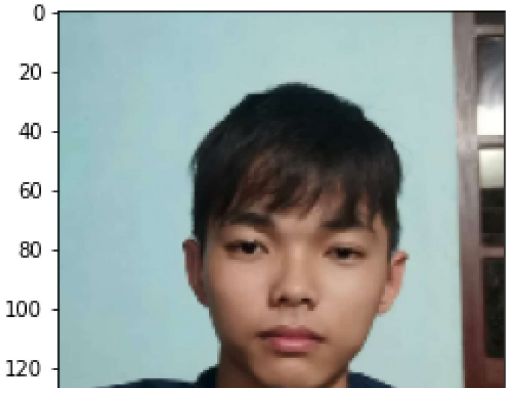
```
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('AnhTuTest1.jpg', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,67500)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('Á,',class_name[int(a)],'nè!')
```

```
[[1.0000000e+00 0.0000000e+00 1.4569816e-35]]
Á, Anh Tú nè!
```




```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('Luan0.png', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,67500)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('Á,',class_name[int(a)],'nè!')
```

```
[[3.0410925e-07 9.9999964e-01 6.4432492e-15]]
Á, Tấn Luân nè!
```




```
1
```

✓ 0 giây hoàn thành lúc 00:55

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import glob
import cv2
import os
from os import listdir
from numpy import asarray
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from keras.layers import Dense,Activation,BatchNormalization, Dropout, LSTM, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from tensorflow.keras.optimizers import RMSprop, SGD
from keras.utils import np_utils
from tensorflow.keras.preprocessing.image import load_img, img_to_array
```

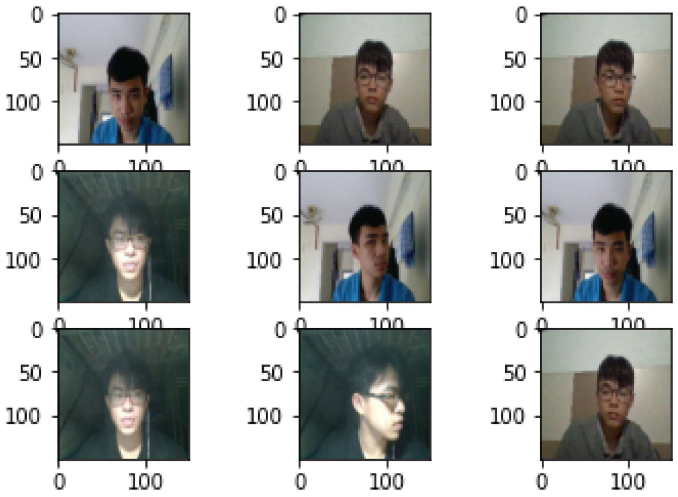
```
folder1 = '/content/drive/MyDrive/FACES/TAT'
folder2 = '/content/drive/MyDrive/FACES/TLUAN/dataSet'
folder3 = '/content/drive/MyDrive/TAN'
photos, labels = list(), list()
for file in listdir(folder1):
  photo = load_img(folder1+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(0)
for file in listdir(folder2):
  photo = load_img(folder2+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(1)
for file in listdir(folder3):
  photo = load_img(folder3+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(2)
photos = asarray(photos)            #lưu dữ liệu ảnh dưới dạng mảng
labels = asarray(labels)            #lưu dữ liệu phân loại dạng mảng
class_name = ['Anh Tú','Tấn Luân','Nhật Tấn']   #tạo tên class tương ứng với dữ liệu phâ
print(photos.shape, labels.shape, class_name)

```

```
    (596, 150, 150, 3) (596,) ['Anh Tú', 'Tấn Luân', 'Nhật Tấn']
```

```
1 x_train, x_test, y_train, y_test = train_test_split(photos, labels, test_size=0.25, rand
2 print(x_train.shape,x_test.shape)
3 print(y_train.shape,y_test.shape)
```

```
    (447, 150, 150, 3) (149, 150, 150, 3)
    (447,) (149,)
```

```
1 x_train_show = x_train.astype(int)
2 import matplotlib.pyplot as plt
3 from matplotlib.image import imread
4 for i in range(9):
5   plt.subplot(330+i+1)
6   plt.imshow(x_train_show[i])
7 plt.show()
```



```
1 x_train/=255
2 x_test/=255
3 y_train=np_utils.to_categorical(y_train)
4 y_test=np_utils.to_categorical(y_test)
```

```
1 model = Sequential()
2 model.add(Conv2D(32,(3,3), activation = 'relu', kernel_initializer='he_uniform',padding=
3 model.add(MaxPooling2D((2,2)))
4
5 model.add(Conv2D(64,(3,3), activation = 'relu', kernel_initializer='he_uniform',padding=
6 model.add(MaxPooling2D((2,2)))
7
8 model.add(Conv2D(128,(3,3), activation = 'relu', kernel_initializer='he_uniform',padding
9 model.add(MaxPooling2D((2,2)))
```

```
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))  #Dense la ful
model.add(Dense(3, activation='Softmax'))
opt = SGD(lr=0.01, momentum=0.9)
model.compile(optimizer=opt,loss='categorical_crossentropy',  metrics=['accuracy'])
history = model.fit(x_train,y_train, batch_size=64,epochs=10,verbose=1,validation_data =
model.save('/content/drive/MyDrive/TAT_Face_CNN.h5')
```

```
Epoch 1/10
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/gradient_descent.py:102: Use
  super(SGD, self).__init__(name, **kwargs)
7/7 [==============================] - 5s 354ms/step - loss: 2.4744 - accuracy: 0.6242
Epoch 2/10
7/7 [==============================] - 1s 115ms/step - loss: 0.2818 - accuracy: 0.9195
Epoch 3/10
7/7 [==============================] - 1s 116ms/step - loss: 0.0038 - accuracy: 1.0000
Epoch 4/10
7/7 [==============================] - 1s 115ms/step - loss: 1.3135e-04 - accuracy: 1.
Epoch 5/10
7/7 [==============================] - 1s 117ms/step - loss: 3.6820e-06 - accuracy: 1.
Epoch 6/10
7/7 [==============================] - 1s 118ms/step - loss: 1.8650e-05 - accuracy: 1.
Epoch 7/10
7/7 [==============================] - 1s 118ms/step - loss: 1.8532e-05 - accuracy: 1.
Epoch 8/10
7/7 [==============================] - 1s 120ms/step - loss: 1.0813e-05 - accuracy: 1.
Epoch 9/10
7/7 [==============================] - 1s 116ms/step - loss: 8.4112e-06 - accuracy: 1.
Epoch 10/10
7/7 [==============================] - 1s 125ms/step - loss: 6.2757e-06 - accuracy: 1.
```

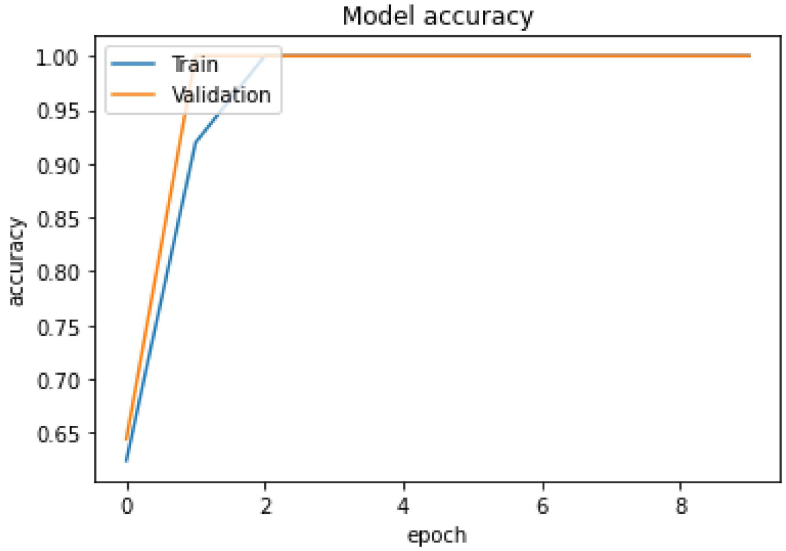
```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss: ',score[0])
print('test accuracy: ', score[1])

y_pred = model.predict(x_test)
print(y_pred.flatten())
print(y_test)
```

```
Test loss:  0.0
test accuracy:  1.0
[2.42502079e-25 2.96307195e-13 1.00000000e+00 1.00000000e+00
 1.71380388e-15 4.78676637e-26 1.00000000e+00 4.46625377e-18
 7.89727813e-20 1.00000000e+00 2.89172711e-14 2.29341984e-22
 1.83721609e-19 1.00000000e+00 1.45062873e-09 2.22566381e-25
 3.19910006e-13 1.00000000e+00 1.00000000e+00 5.79722902e-14
 2.76482790e-24 3.98048273e-26 2.74585584e-13 1.00000000e+00
 7.01852022e-20 1.00000000e+00 1.43615536e-10 9.38104111e-20
 1.00000000e+00 2.77284362e-10 1.00000000e+00 3.03576303e-10
 4.15878010e-20 9.48065692e-20 1.00000000e+00 2.65141770e-10
 5.88784420e-19 1.00000000e+00 8.44586623e-10 5.50168260e-25
```

```
1.01929511e-12 1.00000000e+00 1.00000000e+00 3.12495002e-10
4.38655268e-20 2.23705274e-25 4.36788938e-14 1.00000000e+00
5.87129327e-26 2.68605206e-13 1.00000000e+00 1.00000000e+00
1.01175206e-13 1.33517614e-21 1.55988557e-19 1.00000000e+00
6.57570276e-10 2.09848677e-19 1.00000000e+00 2.41365983e-10
1.00000000e+00 9.71807270e-14 2.82244122e-21 1.00000000e+00
2.73557827e-17 1.63532135e-17 1.00000000e+00 9.41458317e-15
4.14835237e-21 3.40598615e-26 3.05504320e-13 1.00000000e+00
1.00000000e+00 1.34199210e-15 5.07913458e-23 1.00000000e+00
1.61928102e-13 3.11341787e-19 1.60214894e-19 1.00000000e+00
3.60379282e-10 1.00000000e+00 1.44656276e-15 1.12973786e-24
2.34063834e-19 1.00000000e+00 1.68243142e-09 1.63851013e-27
1.79815998e-14 1.00000000e+00 1.00000000e+00 1.18177211e-15
4.02005695e-23 4.86422728e-25 7.86414716e-13 1.00000000e+00
2.87306815e-25 3.57107383e-13 1.00000000e+00 1.24327909e-19
1.00000000e+00 1.37589751e-09 9.31922810e-20 1.00000000e+00
2.01117220e-10 3.68717901e-25 3.85011738e-13 1.00000000e+00
1.00000000e+00 1.01493380e-16 1.60120639e-21 1.00000000e+00
1.08835899e-15 3.12206976e-23 7.95096345e-20 1.00000000e+00
1.12437601e-10 1.00000000e+00 1.76698052e-13 3.19859007e-19
7.29947147e-20 1.00000000e+00 1.40221793e-10 2.37724419e-19
1.00000000e+00 1.47524171e-09 1.62551770e-19 1.00000000e+00
8.59056049e-10 1.00000000e+00 1.04148603e-13 2.42001818e-20
1.00000000e+00 1.47628437e-11 3.16878206e-22 1.64495489e-19
1.00000000e+00 1.16644028e-09 1.00000000e+00 1.84173382e-13
1.21022379e-19 1.37627940e-26 1.91302283e-13 1.00000000e+00
1.00000000e+00 5.02522837e-16 1.34481265e-27 1.88328437e-25
2.84754804e-13 1.00000000e+00 1.98744357e-27 2.18824677e-14
1.00000000e+00 2.05605600e-25 3.13252381e-13 1.00000000e+00
1.73743187e-19 1.00000000e+00 9.59690771e-10 1.00000000e+00
1.78489562e-17 9.30432644e-24 3.87066197e-27 1.05032140e-13
1.00000000e+00 1.00000000e+00 3.51561194e-15 1.79662705e-22
9.80727371e-20 1.00000000e+00 4.19192264e-10 1.20799179e-26
8.04491022e-14 1.00000000e+00 1.00000000e+00 4.10963403e-15
5.43152014e-22 1.82558399e-19 1.00000000e+00 3.32690431e-10
5.87129327e-26 2.68605206e-13 1.00000000e+00 1.70276828e-26
2.21667465e-13 1.00000000e+00 9.48065692e-20 1.00000000e+00
2.65141770e-10 1.45853953e-19 1.00000000e+00 8.48736914e-10
1.68202499e-19 1.00000000e+00 1.68513226e-09 6.45539143e-19
1.00000000e+00 9.82739667e-10 1.00000000e+00 1.62806387e-13
2.37966653e-19 3.82074401e-25 4.91602256e-13 1.00000000e+00
2.97015375e-26 1.30334450e-13 1.00000000e+00 8.43742513e-26
2.97357408e-13 1.00000000e+00 1.00000000e+00 2.25425191e-12
1.16845446e-22 2.04887530e-26 2.94554718e-13 1.00000000e+00
1.00000000e+00 5.42000501e-17 6.44068444e-26 1.00000000e+00
```

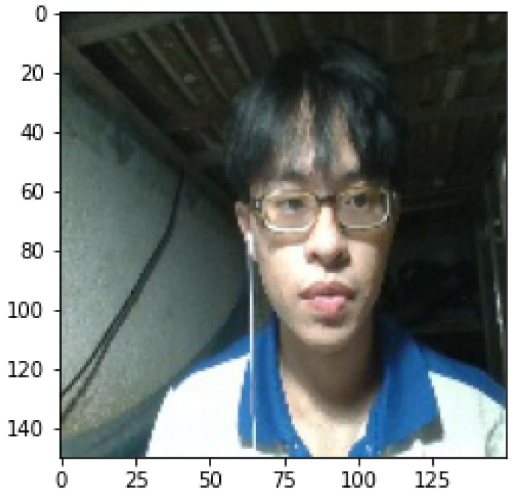
```
1 plt.plot(history.history['accuracy'])
2 plt.plot(history.history['val_accuracy'])
3 plt.title('Model accuracy')
4 plt.ylabel('accuracy')
5 plt.xlabel('epoch')
6 plt.legend(['Train','Validation'], loc='upper left')
```

```
<matplotlib.legend.Legend at 0x7fd3646a5910>
```




```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('AnhTuTest1.jpg', target_size=(150,150))     #Add ảnh test
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('Á,',class_name[int(a)],'nè!')
```

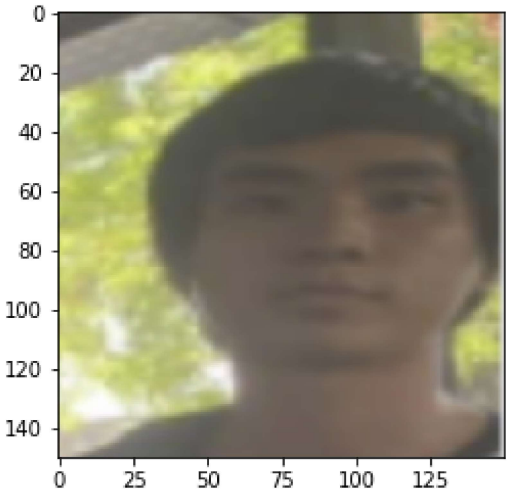
```
[[9.99896646e-01 9.35604946e-14 1.03387734e-04]]
Á, Anh Tú nè!
```




```
from keras.preprocessing.image import load_img
```

```
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('TESTTAN.png', target_size=(150,150))     #Add ảnh test
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('Á,',class_name[int(a)],'nè!')
```

```
[[6.3171893e-20 4.1428781e-11 1.0000000e+00]]
Á, Nhật Tấn nè!
```




```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('Luan0.png', target_size=(150,150))     #Add ảnh test
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('Á,',class_name[int(a)],'nè!')
```

```
[[2.0624676e-15 9.9980122e-01 1.9885239e-04]]
Á, Tấn Luân nè!
```


```
1
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mo
```

```
import glob
import cv2
import os
from os import listdir
from numpy import asarray
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from keras.layers import Dense,Activation,BatchNormalization, Dropout, LSTM, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from tensorflow.keras.optimizers import RMSprop, SGD
from keras.utils import np_utils
from tensorflow.keras.preprocessing.image import load_img, img_to_array
```

```
folder1 = '/content/drive/MyDrive/data_train/bun_thit_nuong'
folder2 = '/content/drive/MyDrive/data_train/bun_bo_hue'
folder3 = '/content/drive/MyDrive/data_train/banh_mi'
folder4 = '/content/drive/MyDrive/data_train/banh_xeo'
folder5 = '/content/drive/MyDrive/data_train/cao_lau'
folder6 = '/content/drive/MyDrive/data_train/mi_quang'
folder7 = '/content/drive/MyDrive/data_train/sup_cua'
folder8 = '/content/drive/MyDrive/data_train/pho'
folder9 = '/content/drive/MyDrive/data_train/com_tam'
folder10 = '/content/drive/MyDrive/data_train/banh_trang_tron'
photos, labels = list(), list()
for file in listdir(folder1):
  photo = load_img(folder1+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(0)
for file in listdir(folder2):
  photo = load_img(folder2+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(1)
for file in listdir(folder3):
  photo = load_img(folder3+'/'+file, target_size=(150,150))
```

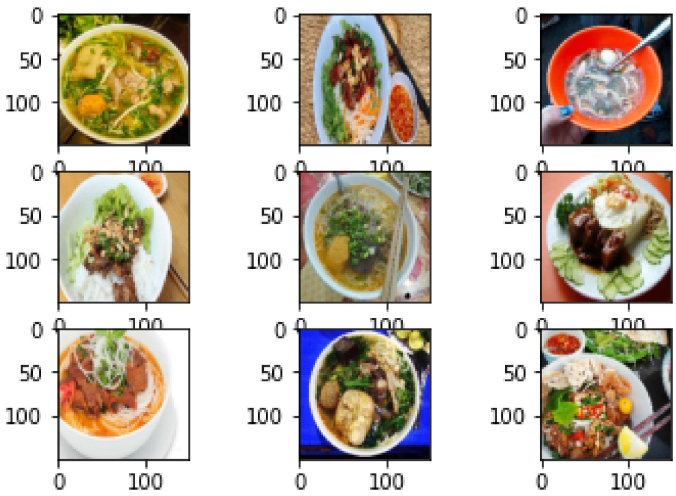
```
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(2)
for file in listdir(folder4):
  photo = load_img(folder4+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(3)
for file in listdir(folder5):
  photo = load_img(folder5+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(4)
for file in listdir(folder6):
  photo = load_img(folder6+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(5)
for file in listdir(folder7):
  photo = load_img(folder7+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(6)
for file in listdir(folder8):
  photo = load_img(folder8+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(7)
for file in listdir(folder9):
  photo = load_img(folder9+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(8)
for file in listdir(folder10):
  photo = load_img(folder10+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(9)
photos = asarray(photos)            #lưu dữ liệu ảnh dưới dạng mảng
labels = asarray(labels)            #lưu dữ liệu phân loại dạng mảng
class_name = ['Bún thịt nướng','Bún Bò Huế','Bánh Mì','Bánh Xèo','Cao Lầu','Mì Quảng','S
print(photos.shape, labels.shape, class_name)
```

```
(7379, 150, 150, 3) (7379,) ['Bún thịt nướng', 'Bún Bò Huế', 'Bánh Mì', 'Bánh Xèo', '(
```

```
x_train, x_test, y_train, y_test = train_test_split(photos, labels, test_size=0.25, rand
print(x_train.shape,x_test.shape)
print(y_train.shape,y_test.shape)
```

```
(5534, 150, 150, 3) (1845, 150, 150, 3)
(5534,) (1845,)
```

```
x_train_show = x_train.astype(int)
import matplotlib.pyplot as plt
from matplotlib.image import imread
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_train_show[i])
plt.show()
```



```
x_train/=255
x_test/=255
y_train=np_utils.to_categorical(y_train)
y_test=np_utils.to_categorical(y_test)
```

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

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

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

```
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))  #Dense la ful
model.add(Dense(10, activation='Softmax'))
opt = SGD(lr=0.01, momentum=0.9)
model.compile(optimizer=opt,loss='categorical_crossentropy',  metrics=['accuracy'])
history = model.fit(x_train,y_train, batch_size=64,epochs=10,verbose=1,validation_data =
model.save('/content/drive/MyDrive/TATSpe1.h5')
```

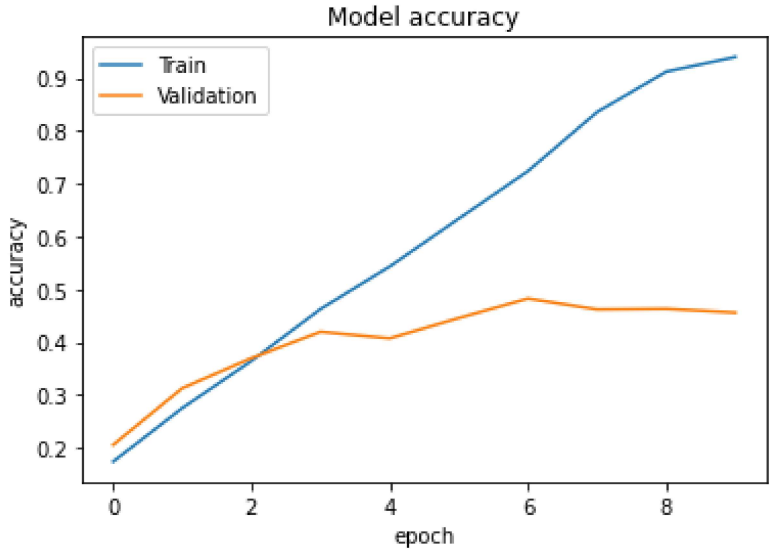
```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/gradient_descent.py:102: Use
  super(SGD, self).__init__(name, **kwargs)
Epoch 1/10
87/87 [==============================] - 23s 130ms/step - loss: 2.3968 - accuracy: 0.1
Epoch 2/10
87/87 [==============================] - 9s 105ms/step - loss: 2.0580 - accuracy: 0.27
Epoch 3/10
87/87 [==============================] - 9s 105ms/step - loss: 1.8420 - accuracy: 0.36
Epoch 4/10
87/87 [==============================] - 9s 105ms/step - loss: 1.5750 - accuracy: 0.46
Epoch 5/10
87/87 [==============================] - 9s 105ms/step - loss: 1.3382 - accuracy: 0.54
Epoch 6/10
87/87 [==============================] - 9s 105ms/step - loss: 1.1080 - accuracy: 0.63
Epoch 7/10
87/87 [==============================] - 9s 105ms/step - loss: 0.8106 - accuracy: 0.72
Epoch 8/10
87/87 [==============================] - 9s 107ms/step - loss: 0.4934 - accuracy: 0.83
Epoch 9/10
87/87 [==============================] - 9s 105ms/step - loss: 0.2653 - accuracy: 0.91
Epoch 10/10
87/87 [==============================] - 9s 107ms/step - loss: 0.1850 - accuracy: 0.94
```

```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss: ',score[0])
print('test accuracy: ', score[1])

y_pred = model.predict(x_test)
print(y_pred.flatten())
print(y_test)
```

```
Test loss:  2.621371269226074
test accuracy:  0.4563685655593872
[1.3257714e-05 9.9980479e-01 2.1513287e-08 ... 8.2390143e-06 2.3462098e-04
 2.7053281e-07]
[[0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 1. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 1. 0. 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')
```

```
<matplotlib.legend.Legend at 0x7f49d2d24d90>
```

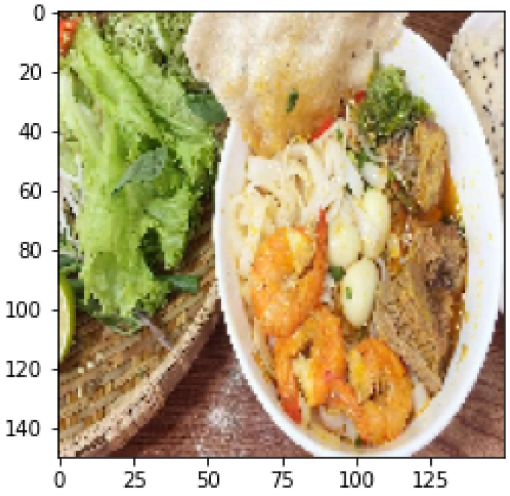



```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('BunBoHue.jpg', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('ĐÂY LÀ',class_name[int(a)])
```

```
[[2.15090779e-06 9.85642612e-01 1.0539476[illegible]
```

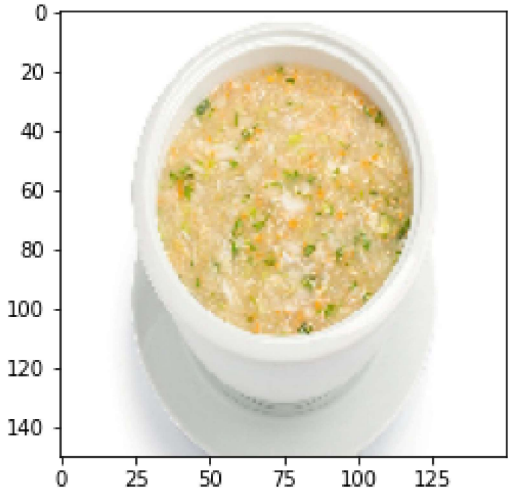
```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('MiQuang.jpg', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('ĐÂY LÀ',class_name[int(a)])
```

```
[[2.2313658e-04 2.9205035e-03 1.2268258e-06 1.1826056e-02 5.3729986e-05
  9.8425967e-01 3.4938697e-05 1.8737472e-05 2.8899212e-08 6.6197856e-04]]
ĐÂY LÀ Mì Quảng
```




```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('SupCua.jpg', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('ĐÂY LÀ',class_name[int(a)])
```

```
[[2.9952676e-04 1.0991320e-02 1.3102180e-01 1.5208359e-04 2.3008162e-05
  1.6783576e-03 6.5389138e-01 6.6619825e-05 1.4288205e-03 2.0044704e-01]]
ĐÂY LÀ Súp Cua
```



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import glob
import cv2
import os
from os import listdir
from numpy import asarray
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from keras.layers import Dense,Activation,BatchNormalization, Dropout, LSTM, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from tensorflow.keras.optimizers import RMSprop, SGD
from keras.utils import np_utils
from tensorflow.keras.preprocessing.image import load_img, img_to_array
```

```
folder1 = '/content/drive/MyDrive/Training/Apple'
folder2 = '/content/drive/MyDrive/Training/Avocado'
folder3 = '/content/drive/MyDrive/Training/Banana'
                                    /Training/Cherry'
                                    /Training/Grape_Pink'
                                    /Training/Litchi'
folder7 = '/content/drive/MyDrive/Training/Orange'
folder8 = '/content/drive/MyDrive/Training/Peach'
folder9 = '/content/drive/MyDrive/Training/Pineapple'
folder10 = '/content/drive/MyDrive/Training/Strawberry'
photos, labels = list(), list()
for file in listdir(folder1):
  photo = load_img(folder1+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(0)
for file in listdir(folder2):
  photo = load_img(folder2+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(1)
for file in listdir(folder3):
  photo = load_img(folder3+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
```

Đã lưu thành công!

```
  photos.append(photo)
  labels.append(2)
for file in listdir(folder4):
  photo = load_img(folder4+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(3)
for file in listdir(folder5):
  photo = load_img(folder5+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(4)
for file in listdir(folder6):
  photo = load_img(folder6+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(5)
for file in listdir(folder7):
  photo = load_img(folder7+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(6)
for file in listdir(folder8):
  photo = load_img(folder8+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(7)
for file in listdir(folder9):
  photo = load_img(folder9+'/'+file, target_size=(150,150))
```

Đã lưu thành công!

```
  labels.append(8)
for file in listdir(folder10):
  photo = load_img(folder10+'/'+file, target_size=(150,150))
  photo = img_to_array(photo)
  photos.append(photo)
  labels.append(9)
photos = asarray(photos)            #lưu dữ liệu ảnh dưới dạng mảng
labels = asarray(labels)            #lưu dữ liệu phân loại dạng mảng
class_name = ['Bom','Bơ','Chuối','Sorry','Nho','Vải','Cam','Đào','Dứa','Dâu']   #tạo tên
print(photos.shape, labels.shape, class_name)
```

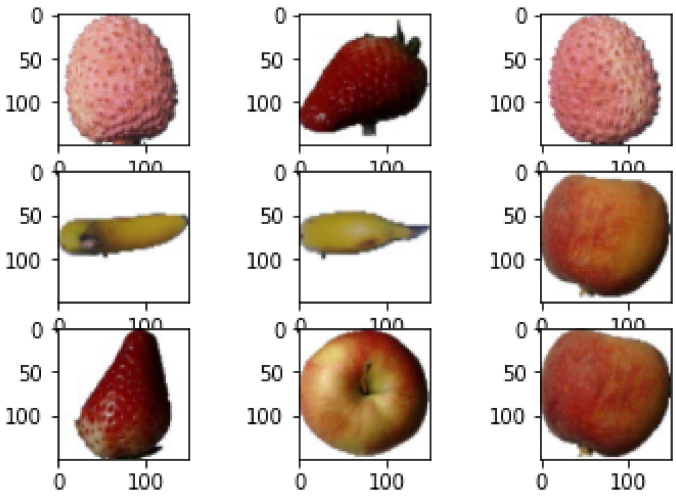
```
(5082, 150, 150, 3) (5082,) ['Bom', 'Bơ', 'Chuối', 'Sorry', 'Nho', 'Vải', 'Cam', 'Đào
```

```
x_train, x_test, y_train, y_test = train_test_split(photos, labels, test_size=0.25, rand
print(x_train.shape,x_test.shape)
print(y_train.shape,y_test.shape)
```

```
(3811, 150, 150, 3) (1271, 150, 150, 3)
```

```
(3811,) (1271,)
```

```
x_train_show = x_train.astype(int)
import matplotlib.pyplot as plt
from matplotlib.image import imread
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x_train_show[i])
plt.show()
```



```
x_train/=255
x_test/=255
y_train=np_utils.to_categorical(y_train)
                                test)
```

Đã lưu thành công!

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

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

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

```
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))  #Dense la ful
model.add(Dense(10, activation='Softmax'))
opt = SGD(lr=0.01, momentum=0.9)
model.compile(optimizer=opt,loss='categorical_crossentropy',  metrics=['accuracy'])
history = model.fit(x_train,y_train, batch_size=64,epochs=10,verbose=1,validation_data =
model.save('/content/drive/MyDrive/TAT.h5')
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/gradient_descent.py:102: Use
  super(SGD, self).__init__(name, **kwargs)
Epoch 1/10
60/60 [==============================] - 5s 71ms/step - loss: 1.4781 - accuracy: 0.569
Epoch 2/10
60/60 [==============================] - 4s 60ms/step - loss: 0.5117 - accuracy: 0.859
Epoch 3/10
60/60 [==============================] - 4s 60ms/step - loss: 0.0051 - accuracy: 0.999
Epoch 4/10
60/60 [==============================] - 4s 59ms/step - loss: 0.0640 - accuracy: 0.978
Epoch 5/10
60/60 [==============================] - 4s 60ms/step - loss: 0.0041 - accuracy: 0.999
Epoch 6/10
60/60 [==============================] - 4s 62ms/step - loss: 1.4766e-04 - accuracy: 1
Epoch 7/10
60/60 [==============================] - 4s 60ms/step - loss: 6.8961e-05 - accuracy: 1
Epoch 8/10
60/60 [==============================] - 4s 60ms/step - loss: 5.6389e-05 - accuracy: 1
Epoch 9/10
60/60 [==============================] - 4s 59ms/step - loss: 4.8297e-05 - accuracy: 1
Epoch 10/10
60/60 [==============================] - 4s 60ms/step - loss: 4.2342e-05 - accuracy: 1
```

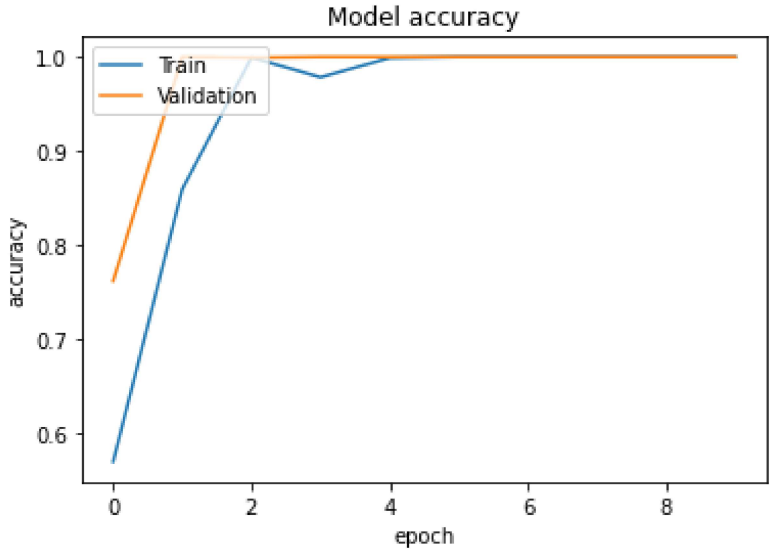
```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss: ',score[0])
print('test accuracy: ', score[1])

y_pred = model.predict(x_test)
print(y_pred.flatten())
```

Đã lưu thành công!

```
Test loss:  6.150678229903527e-05
test accuracy:  1.0
[1.37853677e-13 9.99999642e-01 1.01157026e-17 ... 1.56589935e-10
 1.89187599e-31 2.71841882e-21]
[[0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 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')
```

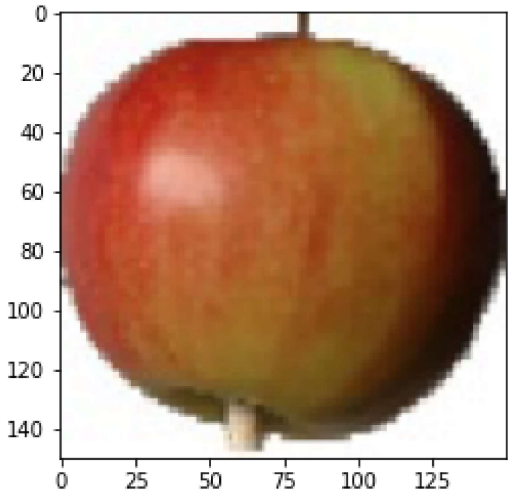
```
<matplotlib.legend.Legend at 0x7fb83c7ec710>
```




```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('/content/drive/MyDrive/Testing/Apple Red.jpg', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)

)
```

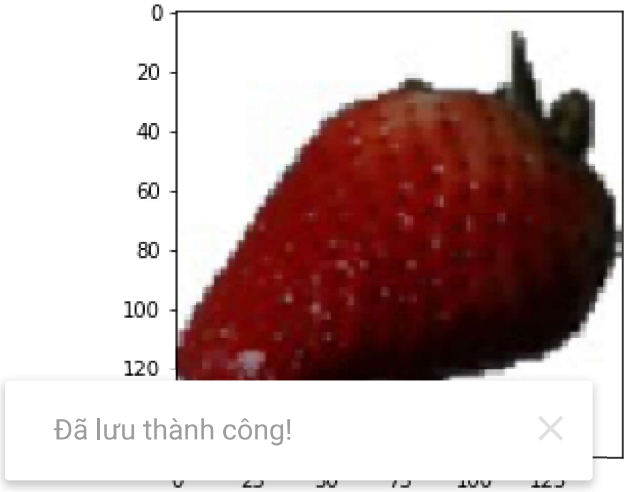
Đã lưu thành công!

```
[[9.9999976e-01 7.7524887e-14 7.1870964e-20 1.4203889e-19 1.3226227e-12
  3.9093903e-20 1.4691177e-12 1.8416968e-07 1.5498678e-18 1.6611409e-20]]
ĐÂY LÀ Bom
```



```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img
from google.colab.patches import cv2_imshow

img = load_img('/content/drive/MyDrive/Testing/Strawberry.jpg', target_size=(150,150))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img/=255.0

a = np.argmax(model.predict(img), axis=-1)
print(model.predict(img))
print('ĐÂY LÀ',class_name[int(a)])
```

```
[[8.9505452e-24 7.5862387e-24 4.3450059e-14 1.7459591e-12 3.0808638e-22
  1.9423627e-17 3.1212837e-23 2.0505505e-13 2.9095576e-22 1.0000000e+00]]
ĐÂY LÀ Dâu
```



Đã lưu thành công!

Đã lưu thành công!