```
import numpy as np
import os

from keras.models import Sequential
from keras.layers import Dense, Conv2D
from keras.layers import Dropout
from keras.layers import Flatten
from keras.constraints import maxnorm
from tensorflow.keras.optimizers import Adam

from keras.layers.convolutional import Convolution2D
from keras.layers.convolutional import MaxPooling2D
from keras.callbacks import ModelCheckpoint, LearningRateScheduler
from keras.callbacks import ReduceLROnPlateau
from keras.callbacks import EarlyStopping


from keras.utils import np_utils
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator


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

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

```
x_train= '/content/drive/MyDrive/10typesoffruits/x_train'
x_test= '/content/drive/MyDrive/10typesoffruits/x_test'
```

```
x_train = ImageDataGenerator(rescale=1/255)
x_test = ImageDataGenerator(rescale=1/255)
```

```
x_train_data = x_train.flow_from_directory(
    directory=r"/content/drive/MyDrive/10typesoffruits/x_train",
    target_size=(224, 224),
    batch_size=3,
    class_mode='categorical',
    )
x_test_data = x_test.flow_from_directory(
    directory=r"/content/drive/MyDrive/10typesoffruits/x_test",
    target_size=(224, 224),
    batch_size=3,
    class_mode= "categorical",
    )
```

```
     Found 97 images belonging to 10 classes.
     Found 28 images belonging to 10 classes.
```

```
x_train_data.class_indices
```

```
{'apple': 0,
 'banana': 1,
 'cherry': 2,
 'coconut': 3,
 'durian': 4,
 'kiwi': 5,
 'mango': 6,
 'orange': 7,
 'pomelo': 8,
 'water melon': 9}
```

```
model = Sequential()

model.add(Conv2D(32,(3,3),input_shape=(224,224,3),padding='same',activation='relu'))
model.add(Dropout(0.2))

model.add(Conv2D(32,(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(4,4)))

model.add(Conv2D(64,(3,3),activation='relu',padding='same'))
model.add(Dropout(0.2))

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

model.add(Conv2D(128,(3,3),activation='relu',padding='same'))
model.add(Dropout(0.2))

model.add(Conv2D(128,(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(Dropout(0.2))

model.add(Dense(1026,activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(512,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(100,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(10,activation='softmax'))
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_36 (Conv2D)          (None, 224, 224, 32)      896       
                                                                 
 dropout_37 (Dropout)        (None, 224, 224, 32)      0         
 conv2d_37 (Conv2D)          (None, 224, 224, 32)      9248      
                                                                 
 max_pooling2d_18 (MaxPoolin  (None, 56, 56, 32)       0         
 g2D)                                                            
                                                                 
 conv2d_38 (Conv2D)          (None, 56, 56, 64)        18496     
                                                                 
 dropout_38 (Dropout)        (None, 56, 56, 64)        0         
                                                                 
 conv2d_39 (Conv2D)          (None, 56, 56, 64)        36928     
                                                                 
 max_pooling2d_19 (MaxPoolin  (None, 28, 28, 64)       0         
 g2D)                                                            
                                                                 
 conv2d_40 (Conv2D)          (None, 28, 28, 128)       73856     
                                                                 
 dropout_39 (Dropout)        (None, 28, 28, 128)       0         
                                                                 
 conv2d_41 (Conv2D)          (None, 28, 28, 128)       147584    
                                                                 
 max_pooling2d_20 (MaxPoolin  (None, 14, 14, 128)      0         
 g2D)                                                            
                                                                 
 flatten_6 (Flatten)         (None, 25088)             0         
                                                                 
 dropout_40 (Dropout)        (None, 25088)             0         
                                                                 
 dense_19 (Dense)            (None, 1026)              25741314  
                                                                 
 dropout_41 (Dropout)        (None, 1026)              0         
                                                                 
 dense_20 (Dense)            (None, 512)               525824    
                                                                 
 dropout_42 (Dropout)        (None, 512)               0         
                                                                 
 dense_21 (Dense)            (None, 100)               51300     
                                                                 
 dropout_43 (Dropout)        (None, 100)               0         
                                                                 
 dense_22 (Dense)            (None, 10)                1010      
                                                                 
=================================================================
Total params: 26,606,456
Trainable params: 26,606,456
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow.keras.optimizers import SGD
#opt = SGD(lr = 0.01, momentum= 0.9)
model.compile(optimizer=Adam(learning_rate=0.0005), loss='categorical_crossentropy', metri
history=model.fit(x_train_data,
                  epochs=10,
                  batch_size=32,
                  verbose=1,
                  validation_data= x_test_data)
```

```
Epoch 1/10
33/33 [==============================] - 3s 60ms/step - loss: 1.1497 - accuracy: 0.5
```

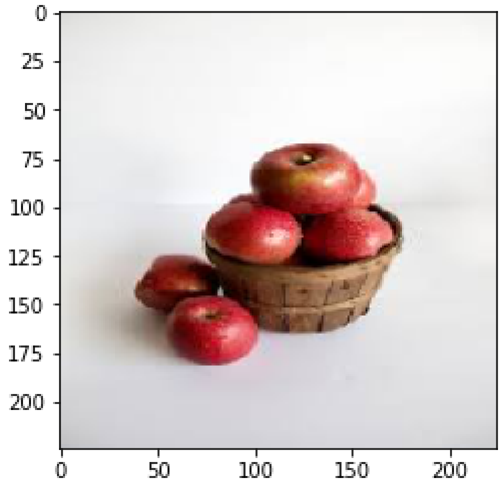
```
Epoch 2/10
33/33 [==============================] - 2s 53ms/step - loss: 0.7143 - accuracy: 0.76
Epoch 3/10
33/33 [==============================] - 2s 52ms/step - loss: 0.7868 - accuracy: 0.76
Epoch 4/10
33/33 [==============================] - 2s 48ms/step - loss: 0.4610 - accuracy: 0.87
Epoch 5/10
33/33 [==============================] - 2s 51ms/step - loss: 0.3766 - accuracy: 0.86
Epoch 6/10
33/33 [==============================] - 2s 48ms/step - loss: 0.3488 - accuracy: 0.87
Epoch 7/10
33/33 [==============================] - 2s 50ms/step - loss: 0.1310 - accuracy: 0.96
Epoch 8/10
33/33 [==============================] - 2s 47ms/step - loss: 0.2828 - accuracy: 0.92
Epoch 9/10
33/33 [==============================] - 2s 51ms/step - loss: 0.2502 - accuracy: 0.96
Epoch 10/10
33/33 [==============================] - 2s 51ms/step - loss: 0.1068 - accuracy: 0.96
```

```
model.save('nhandientraicay.h5')
```

```
from keras.models import load_model
nhandangtraicay = load_model('nhandientraicay.h5')
```

```
from keras.preprocessing.image import load_img, img_to_array
img = load_img('/content/drive/MyDrive/10typesoffruits/x_test/apple/1.jpg', target_size =(
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,224,224,3)
img = img.astype('float32')
img = img/255
img.shape
```

```
(1, 224, 224, 3)
```



```
np.argmax(nhandangtraicay.predict(img),axis= 1)
```

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

```
{'apple': 0, 'banana': 1, 'cherry': 2, 'coconut': 3, 'durian': 4, 'kiwi': 5, 'mango': 6, 'orange': 7, 'pomelo': 8,
'water melon': 9}
```

✓ 0 giây hoàn thành lúc 12:29 ● ×