```
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')
```

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

```
x_train= '/content/drive/MyDrive/FOOD/x_train'
x_test= '/content/drive/MyDrive/FOOD/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/FOOD/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/FOOD/x_test",
    target_size=(224, 224),
    batch_size=3,
    class_mode= "categorical",
    )
```

```
Found 108 images belonging to 10 classes.
```

```
Found 86 images belonging to 10 classes.
```

```
x_train_data.class_indices
```

```
{'BANH MI': 0,
 'BUN BO': 1,
 'BUN CHA HA NOI': 2,
 'BUN DAU MAM TOM': 3,
 'BUN RIEU': 4,
 'CA LOC KHO': 5,
 'COM GA': 6,
 'COM TAM': 7,
 'GOI CUON': 8,
 'PHO': 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=(2,2)))

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

model.add(Dense(218,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_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
```

```
 conv2d_18 (Conv2D)          (None, 224, 224, 32)      896

 dropout_18 (Dropout)        (None, 224, 224, 32)      0

 conv2d_19 (Conv2D)          (None, 224, 224, 32)      9248

 max_pooling2d_9 (MaxPooling  (None, 112, 112, 32)     0
 2D)

 conv2d_20 (Conv2D)          (None, 112, 112, 64)      18496

 dropout_19 (Dropout)        (None, 112, 112, 64)      0

 conv2d_21 (Conv2D)          (None, 112, 112, 64)      36928

 max_pooling2d_10 (MaxPoolin  (None, 56, 56, 64)       0
 g2D)

 conv2d_22 (Conv2D)          (None, 56, 56, 128)       73856

 dropout_20 (Dropout)        (None, 56, 56, 128)       0

 conv2d_23 (Conv2D)          (None, 56, 56, 128)       147584

 max_pooling2d_11 (MaxPoolin  (None, 28, 28, 128)      0
 g2D)

 flatten_3 (Flatten)         (None, 100352)            0

 dropout_21 (Dropout)        (None, 100352)            0

 dense_9 (Dense)             (None, 512)               51380736

 dropout_22 (Dropout)        (None, 512)               0

 dense_10 (Dense)            (None, 218)               111834

 dropout_23 (Dropout)        (None, 218)               0

 dense_11 (Dense)            (None, 100)               21900

 dropout_24 (Dropout)        (None, 100)               0

 dense_12 (Dense)            (None, 10)                1010

=================================================================
Total params: 51,802,488
Trainable params: 51,802,488
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=1,
                  batch_size=32,
```

```
                    verbose=1,
                    validation_data= x_test_data)
```

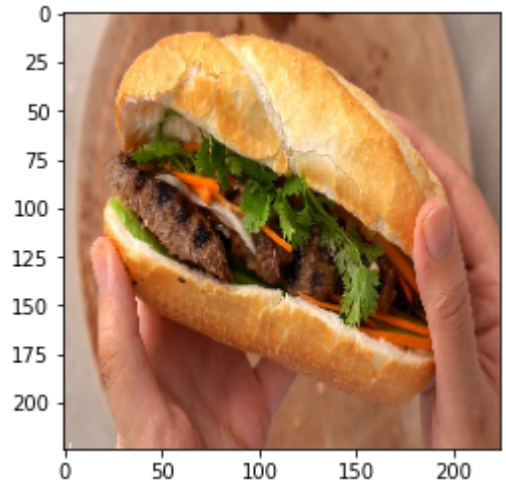
```
36/36 [==============================] - 5s 107ms/step - loss: 0.0894 - accuracy: 0.
```

```
model.save('10loaithucan.h5')
```

```
from keras.models import load_model
nhandangthucan = load_model('10loaithucan.h5')
```

```
from keras.preprocessing.image import load_img, img_to_array
img = load_img('/content/drive/MyDrive/FOOD/x_test/BANH MI/1.jpg', target_size =(224,224))
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(nhandangthucan.predict(img),axis= 1)
```

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

{'BANH MI': 0, 'BUN BO': 1, 'BUN CHA HA NOI': 2, 'BUN DAU MAM TOM': 3, 'BUN RIEU': 4, 'CA LOC KHO': 5, 'COM GA': 6, 'COM TAM': 7, 'GOI CUON': 8, 'PHO': 9}

✓ 0 giây hoàn thành lúc 11:53