

```
import numpy as np
from tensorflow import keras
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img, img_to_array
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import os
import matplotlib.pyplot as plt
from skimage import io
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers import Dense, Activation, Dropout, LSTM, BatchNormalization
from keras.layers import Flatten
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.utils import to_categorical
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
```

```
trainset='/content/drive/MyDrive/10_Food_data/train'
validationset='/content/drive/MyDrive/10_Food_data/validation'
train=ImageDataGenerator(rescale=1/255.0, validation_split=0)
validation=ImageDataGenerator(rescale=1/255.0, validation_split=0.9)
```

```
train_data=train.flow_from_directory(trainset, target_size=(150,150), batch_size=10, class_mode=
validation_set=validation.flow_from_directory(validationset, target_size=(150,150), batch_size=
```

```
    Found 127 images belonging to 10 classes.
    Found 79 images belonging to 10 classes.
```

```
print(train_data.class_indices)
print(validation_set.class_indices)
```

```
{'banh_mi': 0, 'banh_xeo': 1, 'bun_dau': 2, 'che': 3, 'coffee': 4, 'com_tam': 5, 'goi_cu
{'banh_mi': 0, 'banh_xeo': 1, 'bun_dau': 2, 'che': 3, 'coffee': 4, 'com_tam': 5, 'goi_cu
```



```
model=Sequential()
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D((2,2)))
```

```

model.add(Flatten())
model.add(Dense(128,activation='relu'))
model.add(Dense(10,activation='softmax'))

```

```

model.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy'])
history=model.fit(train_data,batch_size=5,epochs=30,verbose=1,validation_data=validation_set)

```

```

Epoch 2/30
13/13 [=====] - 17s 1s/step - loss: 2.2494 - accuracy: 0.189 ^
Epoch 3/30
13/13 [=====] - 17s 1s/step - loss: 2.1580 - accuracy: 0.291
Epoch 4/30
13/13 [=====] - 17s 1s/step - loss: 1.5704 - accuracy: 0.551
Epoch 5/30
13/13 [=====] - 18s 1s/step - loss: 0.9625 - accuracy: 0.677
Epoch 6/30
13/13 [=====] - 17s 1s/step - loss: 0.4220 - accuracy: 0.897
Epoch 7/30
13/13 [=====] - 18s 1s/step - loss: 0.1903 - accuracy: 0.952
Epoch 8/30
13/13 [=====] - 17s 1s/step - loss: 0.1031 - accuracy: 0.984
Epoch 9/30
13/13 [=====] - 18s 1s/step - loss: 0.0020 - accuracy: 1.000
Epoch 10/30
13/13 [=====] - 17s 1s/step - loss: 7.1054e-04 - accuracy: 1
Epoch 11/30
13/13 [=====] - 17s 1s/step - loss: 3.2282e-04 - accuracy: 1
Epoch 12/30
13/13 [=====] - 17s 1s/step - loss: 1.4619e-04 - accuracy: 1
Epoch 13/30
13/13 [=====] - 18s 1s/step - loss: 7.1377e-05 - accuracy: 1
Epoch 14/30
13/13 [=====] - 17s 1s/step - loss: 3.0874e-05 - accuracy: 1
Epoch 15/30
13/13 [=====] - 18s 1s/step - loss: 1.4938e-05 - accuracy: 1
Epoch 16/30
13/13 [=====] - 18s 1s/step - loss: 8.0676e-06 - accuracy: 1
Epoch 17/30
13/13 [=====] - 17s 1s/step - loss: 3.3059e-06 - accuracy: 1
Epoch 18/30
13/13 [=====] - 17s 1s/step - loss: 1.7750e-06 - accuracy: 1
Epoch 19/30
13/13 [=====] - 17s 1s/step - loss: 8.6075e-07 - accuracy: 1
Epoch 20/30
13/13 [=====] - 17s 1s/step - loss: 4.4586e-07 - accuracy: 1
Epoch 21/30
13/13 [=====] - 17s 1s/step - loss: 1.9712e-07 - accuracy: 1
Epoch 22/30
13/13 [=====] - 17s 1s/step - loss: 1.0325e-07 - accuracy: 1
Epoch 23/30
13/13 [=====] - 17s 1s/step - loss: 6.6645e-08 - accuracy: 1
Epoch 24/30
13/13 [=====] - 17s 1s/step - loss: 3.6608e-08 - accuracy: 1
Epoch 25/30

```

```

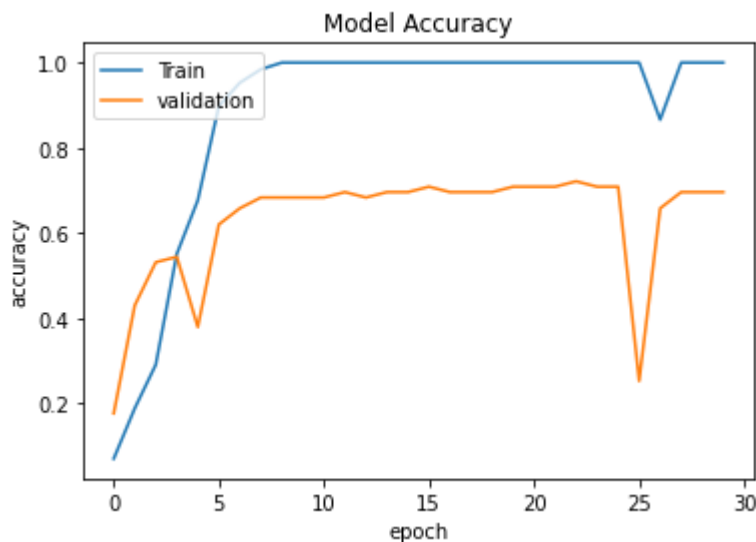
13/13 [=====] - 17s 1s/step - loss: 3.7546e-08 - accuracy: 1
Epoch 26/30
13/13 [=====] - 18s 1s/step - loss: 1.1264e-07 - accuracy: 1
Epoch 27/30
13/13 [=====] - 17s 1s/step - loss: 1.9462 - accuracy: 0.866
Epoch 28/30
13/13 [=====] - 17s 1s/step - loss: 0.0183 - accuracy: 1.000
Epoch 29/30
13/13 [=====] - 18s 1s/step - loss: 0.0034 - accuracy: 1.000
Epoch 30/30
13/13 [=====] - 17s 1s/step - loss: 0.0013 - accuracy: 1.000

```

```

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')
plt.show()

```



```
model.save('/content/drive/MyDrive/BT AI/10_Food.h5')
```

```
load_model('/content/drive/MyDrive/BT AI/10_Food.h5')
```

```
<keras.engine.sequential.Sequential at 0x7f7fd2d6cd90>
```

```
generator= ImageDataGenerator(rescale=1./255)
```

```
generator_data=generator.flow_from_directory('/content/drive/MyDrive/Test food',batch_size=5,
```

```

food={0: 'banh_mi',
      1: 'banh_xeo',
      2: 'bun_dau',
      3: 'che',
      4: 'coffee',

```

```

5:'com_tam',
6:'goi_cuon',
7:'hot_vit_lon',
8:'pho',
9:'xoi'}
plt.figure(figsize=(12,12))
for i in range(len(generator_data_filenames)):
    plt.subplot(4,4,i+1)
    plt.imshow(io.imread(os.path.join(generator_data_directory,generator_data_filenames[i])))
    plt.xticks([])
    plt.yticks([])
    img=load_img('/content/drive/MyDrive/Test food/'+generator_data_filenames[i],target_size=
    img=img_to_array(img)
    img=img.reshape(1,150,150,3)
    img=img.astype('float')
    img=img/255
    plt.xlabel(food[np.argmax(model.predict(img))])
plt.show()

```

☞ Found 10 images belonging to 1 classes.



bun_dau



bun_dau



banh_mi



banh_mi



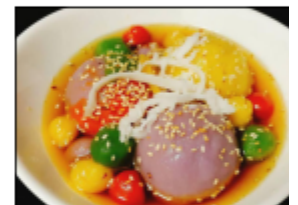
banh_xeo



banh_xeo



che



che



xoi



xoi

✓ 4 giây hoàn thành lúc 11:24

