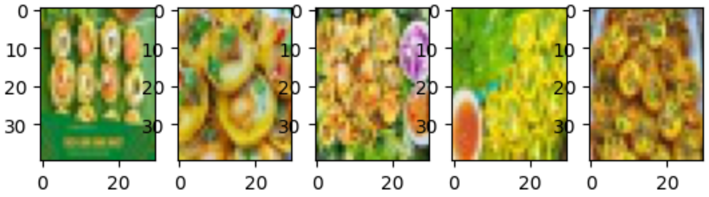
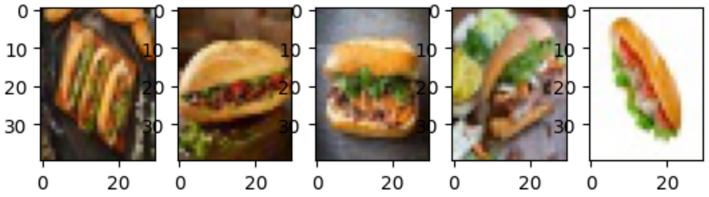
Kiem tra du lieu dau vao

```
import matplotlib.pyplot as plt
from matplotlib.image import imread
folder = '/content/drive/MyDrive/AI_baocao/images/images_food/'
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'banhkhot'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```

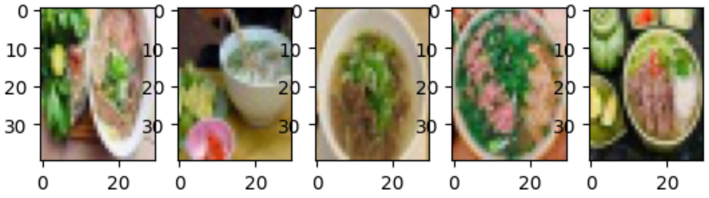


```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'banhmi'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```

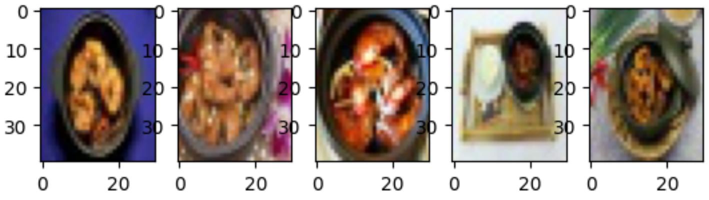


```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'bdmt'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```

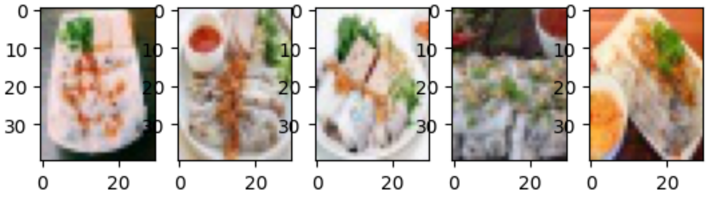
```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'pho'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



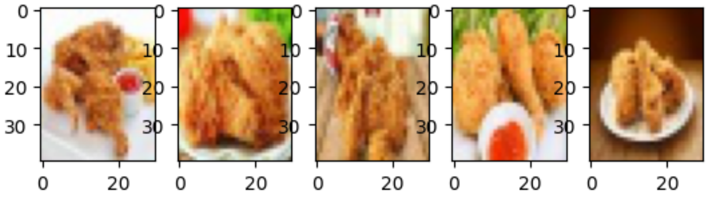
```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'cakhoto'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



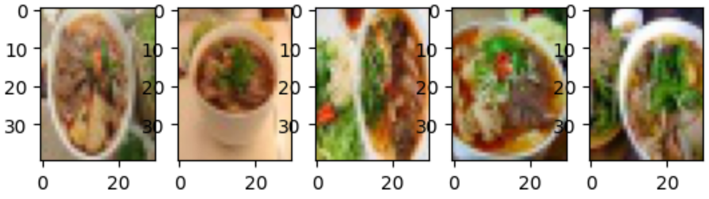
```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'banhuot'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



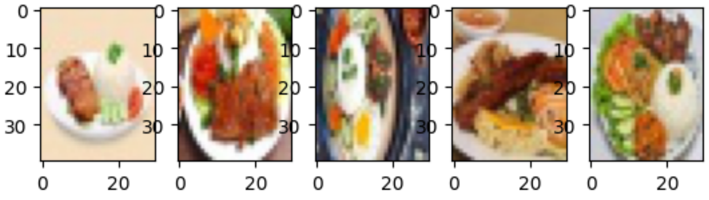
```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'gachien'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



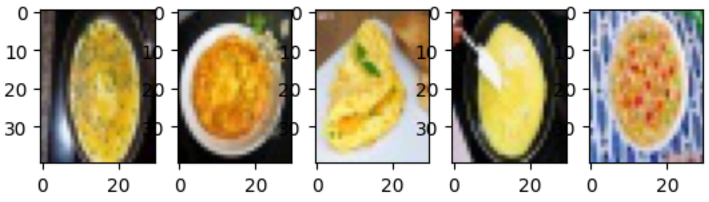
```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'bunbo'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'comsuon'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



```
for i in range(5):
  plt.subplot(1,5,i+1)
  filename = folder + 'trungchien'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



Gan nhan

```
from os import listdir
from os.path import isdir
from numpy import asarray
from numpy import save
from keras.utils import load_img, img_to_array
folder = '/content/drive/MyDrive/AI_baocao/images/images_food/'
photos, labels = list(), list()
for file in listdir(folder):
  output = 0.0
  if file.startswith('banhkhot'):
    output = 1
  if file.startswith('banhmi'):
    output = 2
  if file.startswith('banhuot'):
    output = 3
  if file.startswith('bunbo'):
    output = 4
  if file.startswith('bdmt'):
    output = 5
  if file.startswith('cakhoto'):
    output = 6
  if file.startswith('comsuon'):
    output = 7
  if file.startswith('gachien'):
    output = 8
  if file.startswith('pho'):
    output = 9
  if file.startswith('trungchien'):
    output = 10
  img =load_img(folder+file,target_size=(40,30))
  photo = img_to_array(img)
  photos.append(photo)
  labels.append(output)
photos = asarray(photos)
labels = asarray(labels)
print(photos.shape,labels.shape)
save('predict_food_photos.npy', photos)
save('predict_food_labels.npy', labels)
```

```
(150, 40, 30, 3) (150,)
```

```
import numpy as np
x_train = np.load('predict_food_photos.npy')
y_train = np.load('predict_food_labels.npy')
print(x_train.shape)
print(y_train.shape)
```

```
(150, 40, 30, 3)
(150,)
```

```
x_train = x_train.astype('float32')/255
from keras.utils import to_categorical
y_train = to_categorical(y_train,11)
```

Tao lop tich chap - CNN

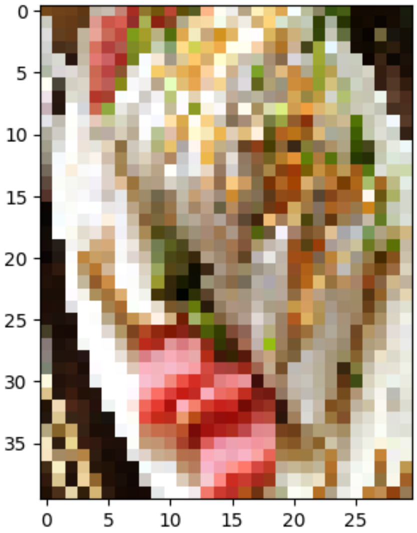
```
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,Conv2D,MaxPooling2D,Normalization,LeakyReLU
from keras.optimizers import Adam
```

```
#32 lan tich chap
model = Sequential()
model.add(Conv2D(32,kernel_size = (3,3),activation = 'relu',input_shape=(40,30,3),padding='Same'))
model.add(MaxPooling2D((2,2),padding='same'))
model.add(Dropout(0.25))

#64 lan tich chap
model.add(Conv2D(64,(3,3),activation ='relu',padding ='same'))
model.add(MaxPooling2D((2,2),padding='same'))
model.add(Dropout(0.25))

#128 lan tich chap
model.add(Conv2D(128,(3,3),activation ='relu',padding ='same'))
model.add(MaxPooling2D((2,2),padding='same'))
model.add(Dropout(0.25))

model.add(Conv2D(256,(3,3),activation ='relu',padding ='same'))
model.add(MaxPooling2D((2,2),padding='same'))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256,activation = 'relu'))
model.add(Dropout(0.25))
model.add(Dense(11,activation='softmax'))

from keras.losses import categorical_crossentropy
model.compile(loss = categorical_crossentropy,optimizer = Adam(),metrics=['accuracy'])
model.summary()
train = model.fit(x_train,y_train,batch_size=150,epochs = 450,verbose = 1)
```

```
Epoch 422/450
1/1 [==============================] - 0s 22ms/step - loss: 0.0837 - accuracy: 0.9800
Epoch 423/450
1/1 [==============================] - 0s 23ms/step - loss: 0.1185 - accuracy: 0.9333
Epoch 424/450
1/1 [==============================] - 0s 22ms/step - loss: 0.2113 - accuracy: 0.9333
Epoch 425/450
1/1 [==============================] - 0s 23ms/step - loss: 0.1713 - accuracy: 0.9400
Epoch 426/450
1/1 [==============================] - 0s 22ms/step - loss: 0.1129 - accuracy: 0.9467
Epoch 427/450
1/1 [==============================] - 0s 27ms/step - loss: 0.1609 - accuracy: 0.9400
Epoch 428/450
1/1 [==============================] - 0s 24ms/step - loss: 0.0887 - accuracy: 0.9667
Epoch 429/450
1/1 [==============================] - 0s 22ms/step - loss: 0.1624 - accuracy: 0.9333
Epoch 430/450
1/1 [==============================] - 0s 22ms/step - loss: 0.0928 - accuracy: 0.9733
Epoch 431/450
1/1 [==============================] - 0s 22ms/step - loss: 0.1158 - accuracy: 0.9667
Epoch 432/450
1/1 [==============================] - 0s 24ms/step - loss: 0.0907 - accuracy: 0.9667
Epoch 433/450
1/1 [==============================] - 0s 24ms/step - loss: 0.1229 - accuracy: 0.9600
Epoch 434/450
1/1 [==============================] - 0s 20ms/step - loss: 0.1054 - accuracy: 0.9733
Epoch 435/450
1/1 [==============================] - 0s 21ms/step - loss: 0.1091 - accuracy: 0.9533
Epoch 436/450
1/1 [==============================] - 0s 22ms/step - loss: 0.0911 - accuracy: 0.9667
Epoch 437/450
1/1 [==============================] - 0s 24ms/step - loss: 0.1250 - accuracy: 0.9600
Epoch 438/450
1/1 [==============================] - 0s 24ms/step - loss: 0.1747 - accuracy: 0.9533
Epoch 439/450
1/1 [==============================] - 0s 23ms/step - loss: 0.1074 - accuracy: 0.9533
Epoch 440/450
1/1 [==============================] - 0s 23ms/step - loss: 0.1223 - accuracy: 0.9667
Epoch 441/450
1/1 [==============================] - 0s 19ms/step - loss: 0.1085 - accuracy: 0.9667
Epoch 442/450
1/1 [==============================] - 0s 23ms/step - loss: 0.0961 - accuracy: 0.9733
Epoch 443/450
1/1 [==============================] - 0s 20ms/step - loss: 0.1004 - accuracy: 0.9600
Epoch 444/450
1/1 [==============================] - 0s 18ms/step - loss: 0.1178 - accuracy: 0.9667
Epoch 445/450
1/1 [==============================] - 0s 24ms/step - loss: 0.1105 - accuracy: 0.9600
Epoch 446/450
1/1 [==============================] - 0s 20ms/step - loss: 0.0952 - accuracy: 0.9600
Epoch 447/450
1/1 [==============================] - 0s 21ms/step - loss: 0.1191 - accuracy: 0.9667
Epoch 448/450
1/1 [==============================] - 0s 23ms/step - loss: 0.0848 - accuracy: 0.9667
Epoch 449/450
1/1 [==============================] - 0s 20ms/step - loss: 0.1370 - accuracy: 0.9533
Epoch 450/450
1/1 [==============================] - 0s 21ms/step - loss: 0.0988 - accuracy: 0.9733
```

```
import matplotlib.pyplot as plt
from keras.utils import load_img
from keras.utils.image_utils import img_to_array
import numpy as np
vat = {1: 'banh khot',2:'banh mi',3:'banh uot',4:'bun bo',5:'bun dau mam tom',6:'ca kho to',7:'com suon',8:'ga chien',9:'pho
img = load_img('food4.jpg',target_size=(40,30))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,30,40,3)
img = img.astype('float32')
img =img/255
result  = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```

```
1/1 [==============================] - 1s 802ms/step
'com suon'
```



```
img = load_img('food3.jpeg',target_size=(40,30))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,30,40,3)
img = img.astype('float32')
img =img/255
result  = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```

```
1/1 [==============================] - 0s 18ms/step
'com suon'
```

✓ 0 giây hoàn thành lúc 23:43 ● ×