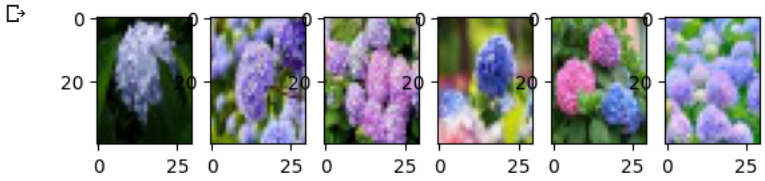
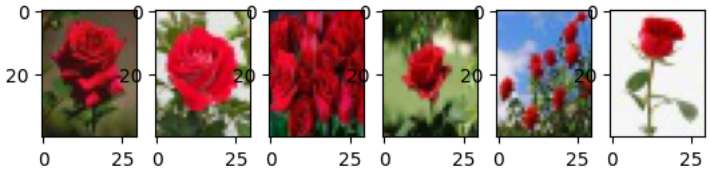
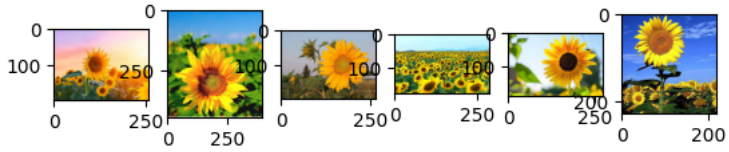
```
import matplotlib.pyplot as plt
from matplotlib.image import imread
folder ='/content/drive/MyDrive/AI_baocao/images/images_flowers/'
for i in range(6):
  plt.subplot(1,6,i+1)
  filename = folder + 'camtucau'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



```
folder ='/content/drive/MyDrive/AI_baocao/images/images_flowers/'
for i in range(6):
  plt.subplot(1,6,i+1)
  filename = folder + 'hong'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



```
folder ='/content/drive/MyDrive/AI_baocao/images/images_flowers/'
for i in range(6):
  plt.subplot(1,6,i+1)
  filename = folder + 'hduong'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```



```
folder ='/content/drive/MyDrive/AI_baocao/images/images_flowers/'
for i in range(6):
  plt.subplot(1,6,i+1)
  filename = folder + 'sen'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```

```
folder ='/content/drive/MyDrive/AI_baocao/images/images_flowers/'
for i in range(6):
  plt.subplot(1,6,i+1)
  filename = folder + 'tulip'+str(i+1)+'.jpg'
  img = plt.imread(filename)
  plt.imshow(img)
plt.show()
```

```
from os import listdir
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_flowers/'
photos, labels = list(), list()
for file in listdir(folder):
  output = 0.0
  if file.startswith('camtucau'):
    output = 1.0
  if file.startswith('hduong'):
    output = 2.0
  if file.startswith('hong'):
    output = 3.0
  if file.startswith('sen'):
    output = 4.0
  if file.startswith('tulip'):
    output = 5.0
  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('hooa_photos.npy', photos)
save('hooa_labels.npy', labels)
```

```
(50, 40, 30, 3) (50,)
```

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

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

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

```
(50, 40, 30, 3)
(50,)
```

```
x_train = x_train.astype('float32')/255
from keras.utils import to_categorical
y_train = to_categorical(y_train,10)
```

```
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))

model.add(Flatten())
model.add(Dense(64,activation = 'relu'))
model.add(Dropout(0.25))
model.add(Dense(10,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=250,epochs = 40,verbose = 1)
```

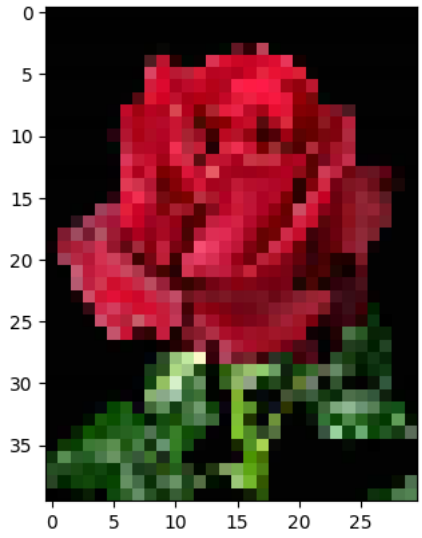
```
Epoch 12/40
1/1 [==============================] - 0s 12ms/step - loss: 1.2575 - accuracy: 0.5200
Epoch 13/40
1/1 [==============================] - 0s 14ms/step - loss: 1.2011 - accuracy: 0.5400
Epoch 14/40
1/1 [==============================] - 0s 14ms/step - loss: 1.2051 - accuracy: 0.6000
Epoch 15/40
1/1 [==============================] - 0s 14ms/step - loss: 1.2490 - accuracy: 0.5600
Epoch 16/40
1/1 [==============================] - 0s 13ms/step - loss: 1.1364 - accuracy: 0.6400
Epoch 17/40
1/1 [==============================] - 0s 11ms/step - loss: 1.0307 - accuracy: 0.7200
Epoch 18/40
1/1 [==============================] - 0s 12ms/step - loss: 0.9173 - accuracy: 0.7600
Epoch 19/40
1/1 [==============================] - 0s 12ms/step - loss: 0.8415 - accuracy: 0.7800
Epoch 20/40
1/1 [==============================] - 0s 18ms/step - loss: 0.8475 - accuracy: 0.7400
Epoch 21/40
1/1 [==============================] - 0s 14ms/step - loss: 0.6881 - accuracy: 0.8600
Epoch 22/40
1/1 [==============================] - 0s 13ms/step - loss: 0.7309 - accuracy: 0.8000
Epoch 23/40
1/1 [==============================] - 0s 12ms/step - loss: 0.7278 - accuracy: 0.7400
Epoch 24/40
1/1 [==============================] - 0s 13ms/step - loss: 0.7100 - accuracy: 0.7600
Epoch 25/40
1/1 [==============================] - 0s 12ms/step - loss: 0.6569 - accuracy: 0.6800
Epoch 26/40
1/1 [==============================] - 0s 12ms/step - loss: 0.7897 - accuracy: 0.7400
Epoch 27/40
1/1 [==============================] - 0s 12ms/step - loss: 0.5542 - accuracy: 0.8400
Epoch 28/40
1/1 [==============================] - 0s 12ms/step - loss: 0.4567 - accuracy: 0.8800
Epoch 29/40
1/1 [==============================] - 0s 13ms/step - loss: 0.5901 - accuracy: 0.8400
Epoch 30/40
1/1 [==============================] - 0s 12ms/step - loss: 0.4271 - accuracy: 0.8600
Epoch 31/40
1/1 [==============================] - 0s 18ms/step - loss: 0.4874 - accuracy: 0.9000
Epoch 32/40
1/1 [==============================] - 0s 13ms/step - loss: 0.4097 - accuracy: 0.8800
Epoch 33/40
1/1 [==============================] - 0s 12ms/step - loss: 0.3522 - accuracy: 0.9200
Epoch 34/40
1/1 [==============================] - 0s 12ms/step - loss: 0.3014 - accuracy: 0.8800
Epoch 35/40
1/1 [==============================] - 0s 17ms/step - loss: 0.4051 - accuracy: 0.8600
Epoch 36/40
1/1 [==============================] - 0s 13ms/step - loss: 0.3808 - accuracy: 0.8600
Epoch 37/40
1/1 [==============================] - 0s 12ms/step - loss: 0.3537 - accuracy: 0.8600
Epoch 38/40
1/1 [==============================] - 0s 13ms/step - loss: 0.2743 - accuracy: 0.9200
Epoch 39/40
1/1 [==============================] - 0s 13ms/step - loss: 0.2795 - accuracy: 0.9000
Epoch 40/40
1/1 [==============================] - 0s 13ms/step - loss: 0.2394 - accuracy: 0.9400
```

```
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: 'cam tu cau',2:'huong duong',3:'hoa hong',4:'hoa sen',5:'hoa tulip'}
img = load_img("hoa1.jpg",target_size=(40,30))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,40,30,3)
img = img.astype('float32')
img =img/255
result  = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```

```
1/1 [==============================] - 0s 20ms/step
'cam tu cau'
```

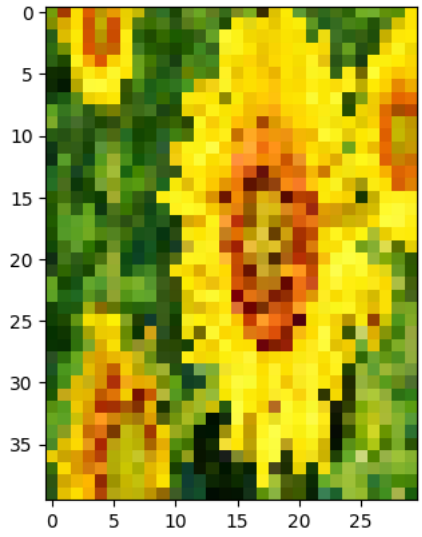
```
img = load_img("hoa2.jpeg",target_size=(40,30))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,40,30,3)
img = img.astype('float32')
img =img/255
result  = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```

```
1/1 [==============================] - 0s 20ms/step
'hoa hong'
```



```
img = load_img("hoa3.jpg",target_size=(40,30))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,40,30,3)
img = img.astype('float32')
img =img/255
result  = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```

```
1/1 [==============================] - 0s 19ms/step
'huong duong'
```



```
img = load_img("hoa5.jpeg",target_size=(40,30))
plt.imshow(img)
```

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

```
1/1 [==============================] - 0s 31ms/step
'hoa tulip'
```