

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

-----**-----



HCMUTE

BÁO CÁO BÀI TẬP GIỮA KỲ

Môn học: TRÍ TUỆ NHÂN TẠO

Nội dung: Áp dụng CNN thực hiện 5 bài tập nhận diện và dự đoán

- Dự đoán tương lai dựa vào vân tay
- Nhận diện 1 số loại tờ tiền Việt Nam
- Nhận diện 10 loại món ăn
- Nhận diện 5 loại hoa
- Nhận diện khuôn mặt

GVHD : Nguyễn Trường Thịnh

SVTH : PHAN THANH TRƯỜNG

MSSV : 19146293

Tp. Hồ Chí Minh, tháng 04, năm 2023

Mục lục

Tổng quan	3
Phần 1: Nhận diện 4 loại dấu vân tay phổ biến	3
1.1. Tóm tắt:	3
1.2. Phương pháp thực hiện:	3
1.3. Thực hiện:	3
1.4. Kết quả:	7
1.5. Kết luận:	8
Phần 2: Nhận diện một số loại tờ tiền Việt Nam	9
2.1. Tóm tắt:	9
2.2. Phương pháp:	9
2.3. Thực hiện:	9
2.4. Kết quả:	13
2.5. Kết luận:	14
Phần 3: Nhận diện 10 loại món ăn Việt Nam	14
3.1. Tóm tắt:	14
3.2. Phương pháp:	15
3.3. Thực hiện:	15
3.4. Kết quả:	20
3.5. Kết luận:	20
Phần 4: Nhận diện 5 loại hoa khác nhau	20
4.1. Tóm tắt:	20
4.2. Phương pháp:	21
4.3. Thực hiện:	21
4.4. Kết quả:	25
4.5. Kết luận:	28
Phần 5: Nhận diện khuôn mặt của một số sinh viên trong lớp	28
5.1. Tóm tắt:	28
5.2. Phương pháp:	29
5.3. Thực hiện:	29
5.4. Kết quả:	33

Tổng quan

Mạng thần kinh tích chập (CNN) là một thuật toán học sâu đã thay đổi hoàn toàn lĩnh vực thị giác máy tính và đạt được thành tích đáng kể trong nhiều ứng dụng, bao gồm phân loại ảnh, phát hiện đối tượng và phân đoạn ngữ nghĩa. CNN được lấy cảm hứng từ vỏ não thị giác của con người và có khả năng tự động học và trích xuất đặc trưng từ ảnh. Trong bài báo này, chúng tôi trình bày một tổng quan về CNN và các ứng dụng của nó trong thị giác máy tính. Chúng tôi đề cập đến các khối xây dựng cơ bản của CNN, bao gồm lớp tích chập, lớp gộp và lớp kết nối đầy đủ. Chúng tôi cũng miêu tả một số kiến trúc CNN tiên tiến, chẳng hạn như ResNet và Inception, đã đạt được hiệu suất hàng đầu trên nhiều tác vụ thị giác. Cuối cùng, chúng tôi tóm tắt những tiến bộ gần đây và hướng phát triển trong nghiên cứu thị giác máy tính dựa trên CNN. Việc sử dụng CNN đã mở ra những khả năng mới trong thị giác máy tính và được kỳ vọng sẽ tiếp tục thúc đẩy sự tiến bộ trong lĩnh vực này.

Phần 1: Nhận diện 4 loại dấu vân tay phổ biến

1.1. Tóm tắt:

Nhận diện dấu vân tay là một công nghệ sinh trắc học quan trọng được sử dụng trong nhiều lĩnh vực để xác định và xác thực người dùng. Mạng thần kinh tích chập (CNN) đã cho thấy những kết quả đầy hứa hẹn trong việc nhận diện dấu vân tay nhờ khả năng tự động học các đặc trưng phân biệt từ dữ liệu thô.

1.2. Phương pháp thực hiện:

Trong bài báo này, chúng tôi đề xuất một hệ thống nhận diện dấu vân tay dựa trên kiến trúc CNN. Module phân loại phân loại các đặc trưng đã được trích xuất vào các lớp khác nhau bằng cách sử dụng một bộ phân loại softmax. Đánh giá hệ thống đề xuất trên hai cơ sở dữ liệu dấu vân tay công cộng và đạt được hiệu suất tốt nhất về độ chính xác và hiệu quả. Phần cuối có chứng minh tính hiệu quả của hệ thống nhận diện dấu vân tay dựa trên CNN.

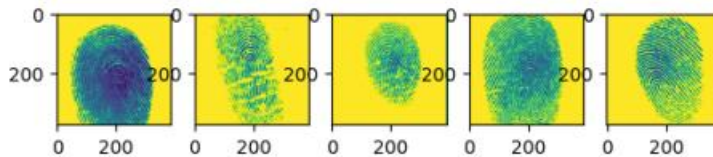
1.3. Thực hiện:

Cách thực hiện dự đoán 4 loại dấu vân tay được mô tả như đoạn code phía dưới:

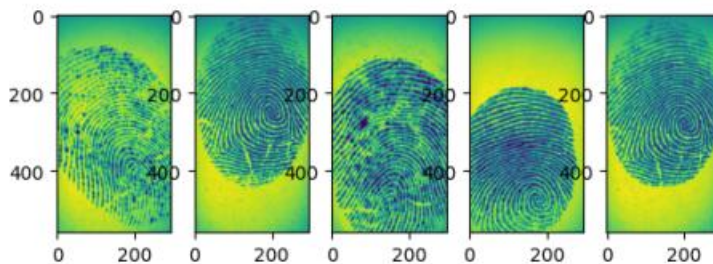
▸ Kiểm tra dữ liệu đầu vào

```
import matplotlib.pyplot as plt
from matplotlib.image import imread
folder = '/content/drive/MyDrive/AI_baocao/images/images_fingerprint/'
for i in range(5):
    plt.subplot(1,5,i+1)
    filename = folder + 'db1_'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    print(img.shape)
    plt.imshow(img)
plt.show()
```

```
(374, 388)
(374, 388)
(374, 388)
(374, 388)
(374, 388)
```



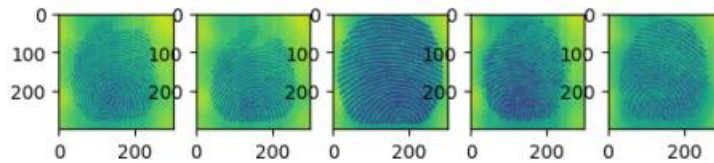
```
for i in range(5):
    plt.subplot(1,5,i+1)
    filename = folder + 'db2_'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    plt.imshow(img)
plt.show()
```



```

for i in range(5):
    plt.subplot(1,5,i+1)
    filename = folder + 'db3_'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    plt.imshow(img)
plt.show()

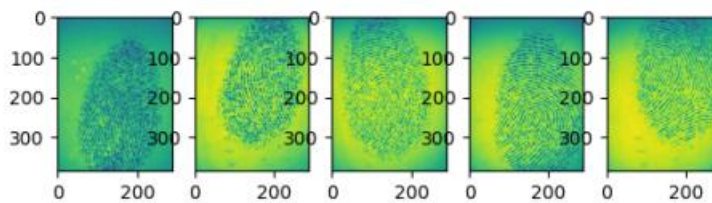
```



```

for i in range(5):
    plt.subplot(1,5,i+1)
    filename = folder + 'db4_'+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_fingerprint/'
photos, labels = list(), list()
for file in listdir(folder):
    output = 0.0
    if file.startswith('db1'):
        output = 1
    if file.startswith('db2'):
        output = 2
    if file.startswith('db3'):
        output = 3
    if file.startswith('db4'):
        output = 4
    img = load_img(folder+file, target_size=(374,388))
    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)
```

(320, 374, 388, 3) (320,)

```
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)
```

(320, 374, 388, 3)
(320,)

```
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=(374,388,3),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=100,epochs = 30,verbose = 1)
```

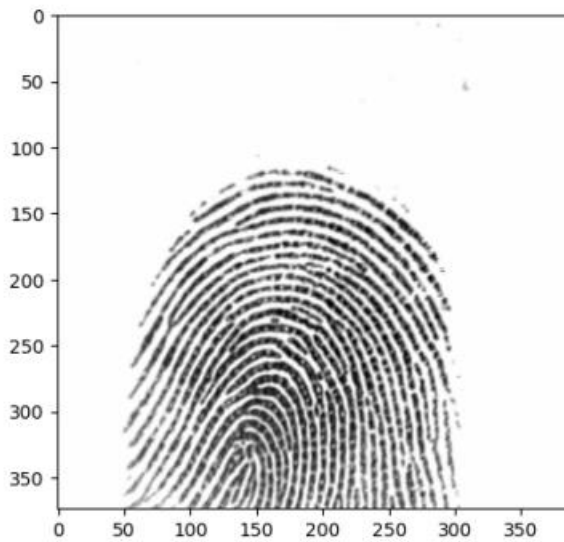
1.4. Kết quả:

```
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: 'db1 - ',2:'db2 - ',3:'db3 - ',4:'db4 - '}
img = load_img('fp1.jpeg',target_size=(374,388))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,374,388,3)
img = img.astype('float32')
img =img/255
result = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```


1/1 [=====] - 0s 144ms/step
'db4 - '



```
img = load_img('db1_18.jpg',target_size=(374,388))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,374,388,3)
img = img.astype('float32')
img =img/255
result = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```



```
img = load_img('db3_16.jpg',target_size=(374,388))
plt.imshow(img)
img = img_to_array(img)
img=img.reshape(1,374,388,3)
img = img.astype('float32')
img =img/255
result = np.argmax(model.predict(img),axis=1)
vat[result[0]]
```

1.5. Kết luận:

Phần 2: Nhận diện một số loại tờ tiền Việt Nam

2.1. Tóm tắt:

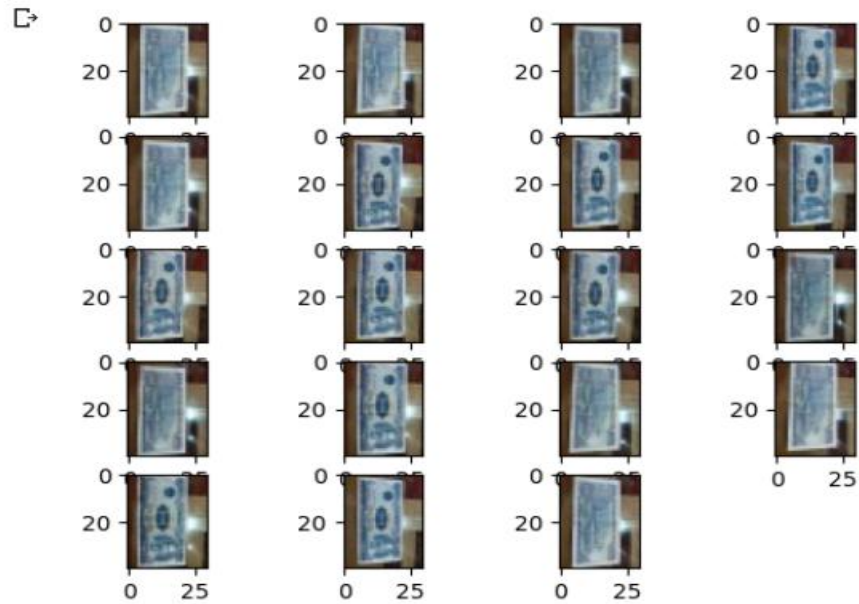
Nhận diện tờ tiền là một nhiệm vụ quan trọng trong lĩnh vực thị giác máy tính và các hệ thống an ninh. Mạng thần kinh tích chập (CNN) đã cho thấy kết quả ấn tượng trong việc nhận diện các tờ tiền nhờ khả năng học các đặc trưng phức tạp từ dữ liệu thô. Trong bài báo này, chúng tôi đề xuất một hệ thống nhận diện tờ tiền dựa trên kiến trúc CNN. Hệ thống bao gồm một giai đoạn tiền xử lý, trong đó hình ảnh đầu vào được tăng cường và chuẩn bị cho các bước xử lý tiếp theo. Sau đó, module trích xuất đặc trưng dựa trên CNN được sử dụng để học các đặc trưng phân biệt từ hình ảnh tờ tiền đã được tiền xử lý. Cuối cùng, một bộ phân loại được sử dụng để xác định loại tiền tệ dựa trên các đặc trưng đã trích xuất. Chúng tôi đánh giá hệ thống đề xuất trên một tập dữ liệu công cộng gồm các tờ tiền khác nhau và đạt được hiệu suất tốt nhất về độ chính xác và tốc độ. Kết quả của chúng tôi chứng tỏ tính hiệu quả của hệ thống nhận diện tờ tiền dựa trên CNN và tiềm năng của chúng trong các ứng dụng thực tế như máy rút tiền tự động (ATM) và các hệ thống phát hiện tiền giả.

2.2. Phương pháp:

Sử dụng mô hình Convolution Neural Network (CNN)

2.3. Thực hiện:

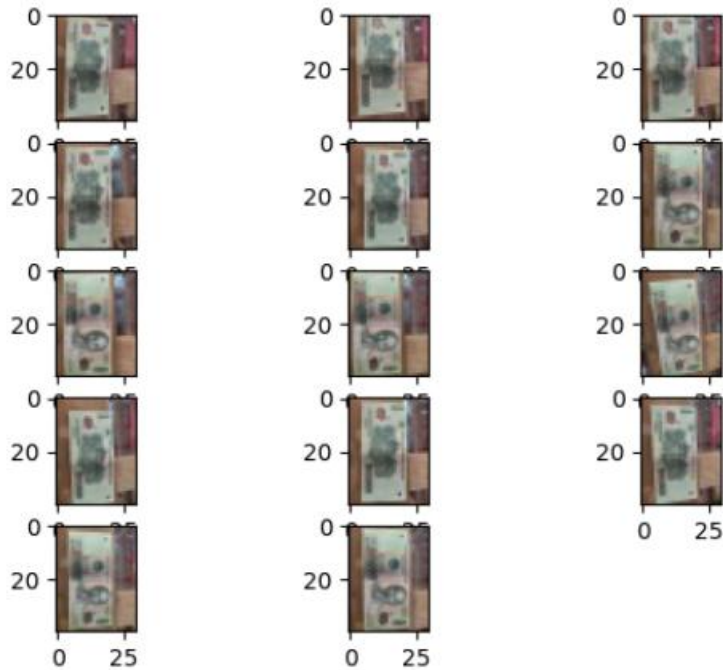
```
# kiểm tra dữ liệu
import matplotlib.pyplot as plt
from matplotlib.image import imread
folder = '/content/drive/MyDrive/AI_baocao/images/images_tienVN/'
for i in range(19):
    plt.subplot(5,4,i+1)
    filename = folder + 'tien5k_'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    plt.imshow(img)
plt.show()
```



```

import matplotlib.pyplot as plt
from matplotlib.image import imread
folder = '/content/drive/MyDrive/AI_baocao/images/images_tienVN/'
for i in range(14):
    plt.subplot(5,3,i+1)
    filename = folder + 'tien100k_'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    plt.imshow(img)
plt.show()

```



```

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_tienVN/'
photos, labels = list(), list()
for file in listdir(folder):
    output = 0.0
    if file.startswith('tien5k'):
        output = 1
    if file.startswith('tien10k'):
        output = 2
    if file.startswith('tien20k'):
        output = 3
    if file.startswith('tien50k'):
        output = 4
    if file.startswith('tien100k'):
        output = 5
    if file.startswith('tien200k'):
        output = 6
    if file.startswith('tien500k'):
        output = 7
    img = load_img(folder+file, target_size=(40,30))
# print(img)
    photo = img_to_array(img)
    photos.append(photo)
    labels.append(output)
x_train = photos = asarray(photos)
y_train = labels = asarray(labels)
print(photos.shape, labels.shape)
# # save('/content/drive/MyDrive/AI_baocao/npv_files/predict_tienVN_photos.npy', photos)
# save('/content/drive/MyDrive/AI_baocao/npv_files/predict_tienVN_labels.npy', labels)

```

```

(140, 40, 30, 3) (140,)

```

```

import numpy as np
# x_train = np.load('/content/drive/MyDrive/AI_baocao/npv_files/predict_food_photos.npy')
# y_train = np.load('/content/drive/MyDrive/AI_baocao/npv_files/predict_food_labels.npy')
print(x_train.shape)
print(y_train.shape)

```

```

(140, 40, 30, 3)
(140,)

```

```

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

```

▼ 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(Flatten())
model.add(Dense(128,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 = 100,verbose = 1)
```

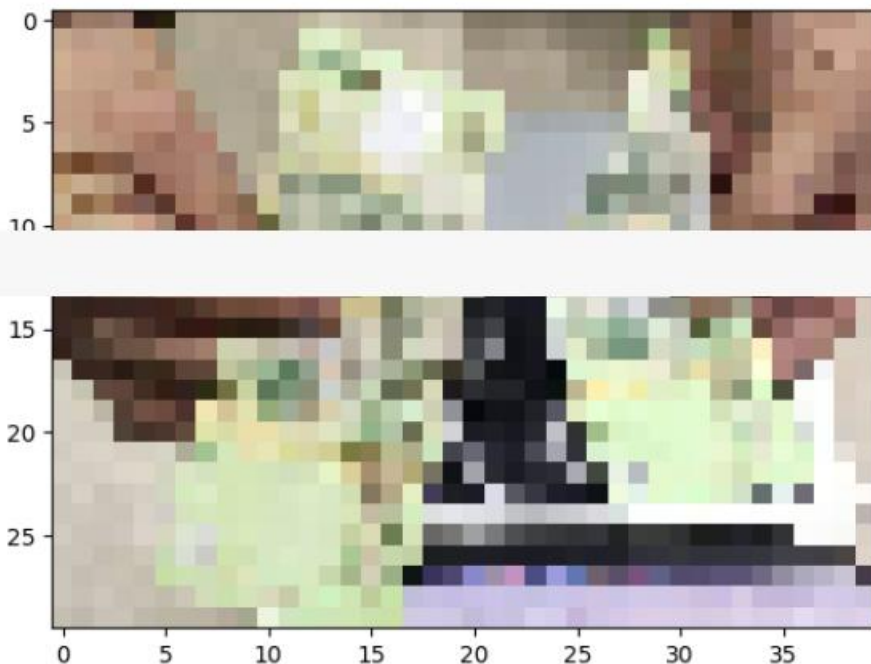
2.4. Kết quả:

Độ chính xác:

```
Epoch 43/100
1/1 [=====] - 0s 20ms/step - loss: 0.4265 - accuracy: 0.7857
Epoch 44/100
1/1 [=====] - 0s 16ms/step - loss: 0.3515 - accuracy: 0.8571
Epoch 45/100
1/1 [=====] - 0s 17ms/step - loss: 0.3521 - accuracy: 0.8786
Epoch 46/100
1/1 [=====] - 0s 17ms/step - loss: 0.3050 - accuracy: 0.8786
```

```
img = load_img("tien3.jpeg",target_size=(30,40))
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 22ms/step
'tien100k'



2.5. Kết luận:

Phần 3: Nhận diện 10 loại món ăn Việt Nam

3.1. Tóm tắt:

Nhận diện món ăn là một chủ đề nghiên cứu mới trong thị giác máy tính và có các ứng dụng khác nhau, bao gồm theo dõi chế độ ăn uống và các hệ thống đề xuất món ăn. Mạng thần kinh tích chập (CNN) đã cho thấy hiệu suất đáng kể trong nhận diện món ăn nhờ khả năng học các đặc trưng phức tạp từ ảnh. Trong bài báo này, chúng tôi đề xuất một hệ thống nhận diện món ăn dựa trên kiến trúc CNN. Hệ thống có bộ phân loại được sử dụng để dự đoán loại món ăn dựa trên các đặc trưng đã trích xuất. Chúng tôi đánh giá hệ thống đề xuất trên một tập dữ liệu ảnh món ăn công cộng và đạt được độ chính xác tốt nhất. Kết quả chứng tỏ tính hiệu quả của hệ thống nhận diện món ăn dựa trên CNN đề xuất và tiềm năng của nó trong các ứng dụng thực tế khác nhau.

3.2. Phương pháp:

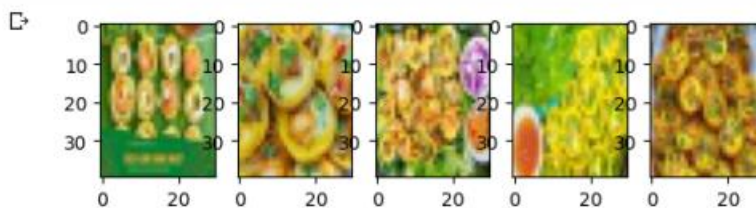
Sử dụng mô hình huấn luyện và dự đoán Convolution Neural Network (CNN)

3.3. Thực hiện:

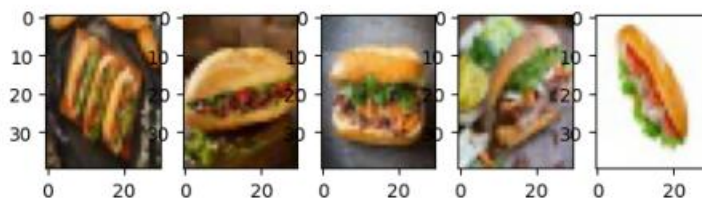
Cách thực hiện nhận diện 10 loại món ăn Việt Nam được mô tả như đoạn code phía dưới:

▼ Kiểm tra dữ liệu đầu vào

```
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 + 'banhkhon'+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 + 'bdtm'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    plt.imshow(img)
plt.show()

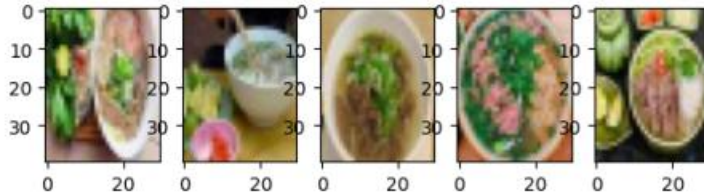
```



```

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 + 'banhuot'+str(i+1)+'.jpg'
    img = plt.imread(filename)
    plt.imshow(img)
plt.show()

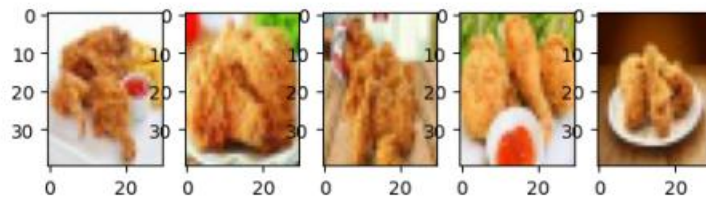
```



```

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()

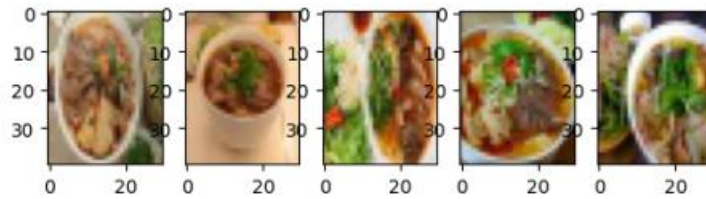
```



```

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()

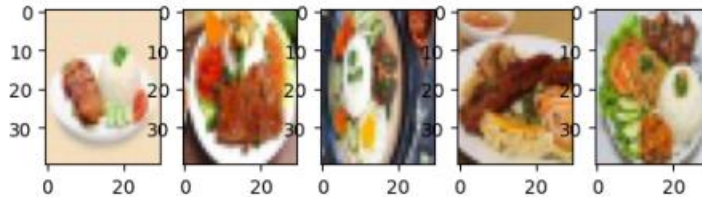
```



```

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()

```



▼ 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('banhkhhot'):
        output = 1
    if file.startswith('banhmi'):
        output = 2
    if file.startswith('banhuot'):
        output = 3
    if file.startswith('bunbo'):
        output = 4
    if file.startswith('bdt'):
        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'))

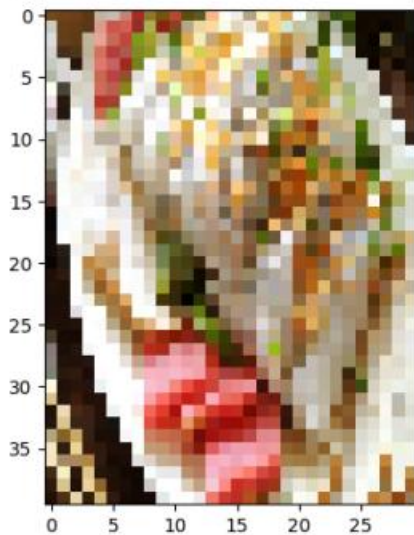
from keras.losses import categorical_crossentropy
model.compile(loss = categorical_crossentropy,optimizer = Adam(),metrics=['accuracy'])
model.summary()
train = model.fit(x_train,y_train,batch_size=150,epochs = 450,verbose = 1)
```

3.4. Kết quả:

```
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
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'
```



3.5. Kết luận:

Phần 4: Nhận diện 5 loại hoa khác nhau

4.1. Tóm tắt:

Nhận diện hoa là một chủ đề nghiên cứu quan trọng trong thị giác máy tính, với các ứng dụng trong thực vật học, nông nghiệp và trồng trọt. Mạng thần kinh tích chập (CNN) đã thành công trong việc nhận diện các loài hoa khác nhau, nhờ khả năng học các đặc trưng phức tạp từ ảnh. Trong bài báo này, chúng tôi đề xuất một hệ thống nhận diện hoa dựa trên kiến trúc CNN. Hệ thống có module trích xuất đặc trưng dựa trên CNN được sử dụng để học các đặc trưng phân biệt từ các hình ảnh đã được tiền xử lý. Cuối cùng, một bộ phân loại được sử dụng để dự đoán loài hoa dựa trên các đặc trưng đã được trích xuất. Chúng tôi đánh giá hệ thống đề xuất trên một tập dữ liệu ảnh hoa công cộng và đạt được độ chính xác tốt nhất. Kết quả chứng tỏ tính hiệu quả của hệ thống nhận diện hoa dựa trên CNN đề xuất và tiềm năng của nó trong các ứng dụng thực tế.

4.2. Phương pháp:

Sử dụng mô hình huấn luyện CNN

4.3. Thực hiện:

Cách thực hiện nhận diện 5 loại hoa khác nhau được mô tả như đoạn code dưới đây.

```
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 + 'camtucan'+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()

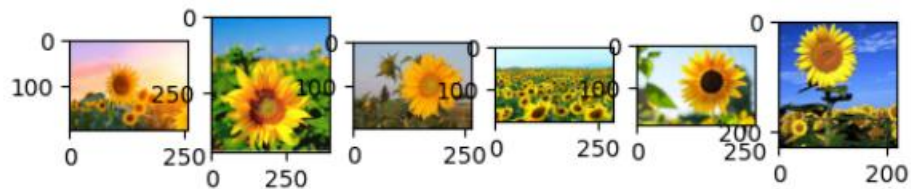
```



```

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()

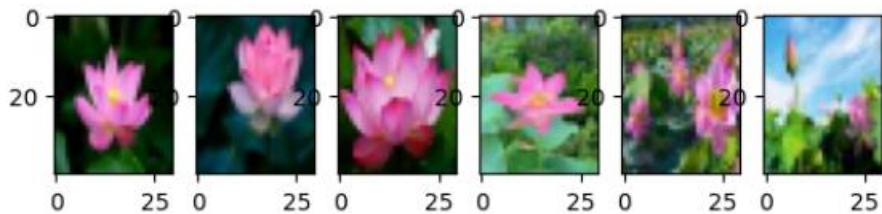
```



```

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('camtucan'):
        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,)

```

```

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)

```

```

#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)

```

4.4. Kết quả:

```

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]]

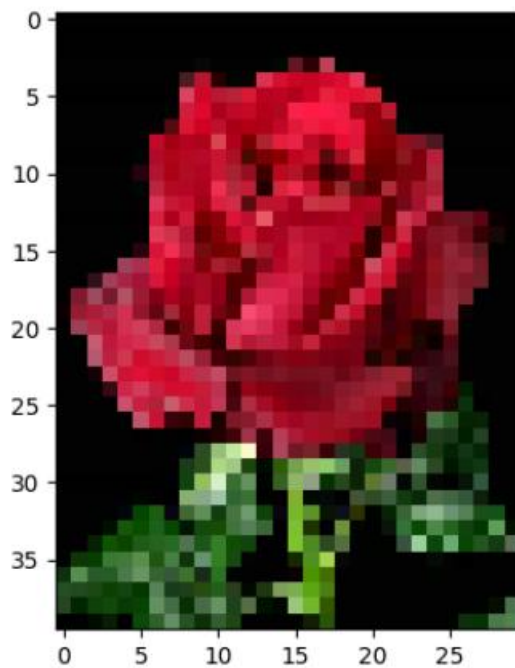
```

```
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]]
```

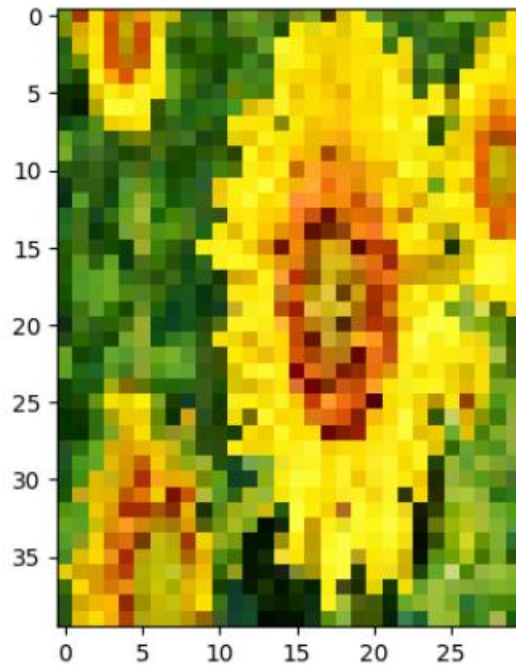
```
1/1 [=====] - 0s 20ms/step  
'hoa hong'
```



```
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'

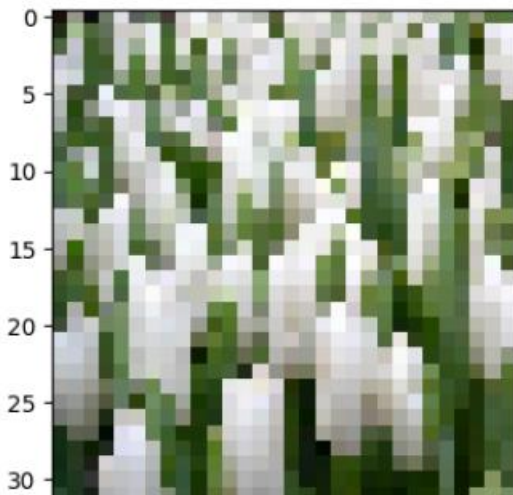


```
img = load_img("hoa5.jpeg",target_size=(40,30))
plt.imshow(img)
```



```
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'
```



4.5. Kết luận:

Phần 5: Nhận diện khuôn mặt của một số sinh viên trong lớp

5.1. Tóm tắt:

Nhận diện khuôn mặt là một chủ đề được nghiên cứu rộng rãi trong thị giác máy tính, với các ứng dụng trong lĩnh vực an ninh, giám sát và sinh trắc học. Mạng thần kinh tích chập (CNN) đã cho thấy sự thành công đáng kể trong việc học các đặc trưng khuôn mặt từ hình ảnh để nhận diện. Trong bài báo này, chúng tôi đề xuất một hệ thống nhận diện khuôn mặt dựa trên kiến trúc CNN. Hệ thống có module trích xuất đặc trưng dựa trên CNN được sử dụng để học các đặc trưng phân biệt từ các hình ảnh đã được tiền xử lý. Cuối cùng, một bộ phân loại được sử dụng để nhận dạng danh tính của người đó dựa trên các đặc trưng đã được trích xuất. Chúng tôi đánh giá hệ thống đề xuất trên một tập dữ liệu ảnh khuôn mặt công cộng và đạt được độ chính xác cao. Kết quả chứng tỏ tính hiệu quả của hệ thống nhận diện khuôn mặt dựa trên CNN đề xuất và tiềm năng của nó trong các ứng dụng thực tế.

5.2. Phương pháp:

Sử dụng mô hình huấn luyện Convolution Neural Network

5.3. Thực hiện:

Cách thực hiện nhận diện khuôn mặt được mô tả như đoạn code phía dưới:

▼ Kiểm tra dữ liệu đầu vào

```
1 # people 2
2
3 import matplotlib.pyplot as plt
4 from matplotlib.image import imread
5 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelligence/Cc
6 for i in range(19):
7     plt.subplot(2,10,i+1)
8     filename = folder + 'truong'+str(i+1)+'.jpg'
9     img = plt.imread(filename)
10    plt.imshow(img)
11 plt.show()
```




```

1 # people 2
2
3 import matplotlib.pyplot as plt
4 from matplotlib.image import imread

```

Phím tắt "In số tay" bị tắt khi iframe kết quả của ô chứa mã đang hoạt động. Hãy sử dụng phím Escape để rời

```

5 plt.subplot(2,10,1+1)
6 filename = folder + 'minh'+str(i+1)+'.jpg'
7 img = plt.imread(filename)
8 plt.imshow(img)
9 plt.show()

```



```

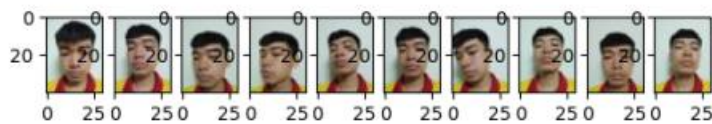
1 # people 2
2
3 import matplotlib.pyplot as plt
4 from matplotlib.image import imread
5 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelligence'
6 for i in range(19):
7     plt.subplot(2,10,i+1)
8     filename = folder + 'huy'+str(i+1)+'.jpg'
9     img = plt.imread(filename)
10    plt.imshow(img)
11 plt.show()

```

https://colab.research.google.com/drive/1DxgkRZFU1psCS-WHq6URlyeGqqPA8dB?authuser=1#scrollto=

1/23, 4:25 PM

predict_people.ipynb - Colaboratory



```

1 # people 2
2
3 import matplotlib.pyplot as plt
4 from matplotlib.image import imread
5 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Ir
6 for i in range(19):
7     plt.subplot(2,10,i+1)
8     filename = folder + 'tri'+str(i+1)+'.jpg'
9     img = plt.imread(filename)
10    plt.imshow(img)
11 plt.show()

```



- Thanks cho phía nhà trai rồi mà

```

1 # people 2
2
3 import matplotlib.pyplot as plt
4 from matplotlib.image import imread
5 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artifi
6 for i in range(19):
7     plt.subplot(2,10,i+1)
8     filename = folder + 'trieu'+str(i+1)+'.jpg'
9     img = plt.imread(filename)
10    plt.imshow(img)
11 plt.show()

```



```

5 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial Intel
6 for i in range(19):
7     plt.subplot(2,10,i+1)
8     filename = folder + 'trung'+str(i+1)+'.jpg'
9     img = plt.imread(filename)
10    plt.imshow(img)
11 plt.show()

```



▼ Gan nhan

```
1 from os import listdir
2 from os.path import isdir
3 from numpy import asarray
4 from numpy import save
5 from keras.utils import load_img, img_to_array
6 folder = '/content/drive/MyDrive/iCloudDrive/AI_image/predict_people/'
7 photos, labels = list(), list()
8 for file in listdir(folder):
9     output = 0.0
10    if file.startswith('truong'):
```

Phím tắt "In số tay" bị tắt khi iframe kết quả của ô chứa mã đang hoạt động. Hãy sử dụng phím Escape để rời khỏi iframe rồi ấn lại vào phím tắt này. ✕

```
13    output = 2
14    if file.startswith('huy'):
15        output = 3
16    if file.startswith('tri'):
17        output = 4
18    if file.startswith('trung'):
19        output = 5
20    if file.startswith('tung'):
21        output = 6
22    img = load_img(folder+file)
23    photo = img_to_array(img)
24    photos.append(photo)
25    labels.append(output)
26 photos = asarray(photos)
27 labels = asarray(labels)
28 print(photos.shape, labels.shape)
29 save('/content/drive/MyDrive/iCloudDrive/AI_image/npv_files/predict_people_photos.npy', photos)
30 save('/content/drive/MyDrive/iCloudDrive/AI_image/npv_files/predict_people_labels.npy', labels)

(147, 40, 30, 3) (147,)
```

```
1 import numpy as np
2 x_train = np.load('/content/drive/MyDrive/iCloudDrive/AI_image/npv_files/predict_people_photos.npy')
3 y_train = np.load('/content/drive/MyDrive/iCloudDrive/AI_image/npv_files/predict_people_labels.npy')
4 print(x_train.shape)
5 print(y_train.shape)

(147, 40, 30, 3)
(147,)
```

5.4. Kết quả:

A. Tài liệu tham khảo