TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

LÊ ÂU XUÂN DƯƠNG – 19120488

BÁO CÁO HW02

MÔN HỌC: NHẬN DẠNG

CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN LÝ THUYẾT

Lê Hoàng Thái

GIÁO VIÊN HƯỚNG DẪN THỰC HÀNH

Lê Thanh Phong

Tp. Hồ Chí Minh, ngày 31/05/2023

1. Xây dựng tập dữ liệu

Tập dữ liệu ảnh trong bài em sử dụng là tập FEI Face Database.

Link database: https://fei.edu.br/~cet/facedatabase.html

Dữ liệu của tập ảnh gồm tầm người, mỗi người sẽ gồm 14 ảnh nhiều góc cạnh và ánh sáng như ví dụ trên.

Tuy nhiên trong bài yêu cầu việc nhận dạng điểm danh 20 người nên em chỉ lấy dữ liệu của 20 người đầu tiên.

Vì tập dữ liệu là ít nên em tiến hành phân bố file dữ liệu trực tiếp và dán nhãn cho các ảnh.

Link drive:
https://drive.google.com/drive/folders/16JFW3u4XNytpWZBsbYafgZM_1FjwIovM?usp=sharing

2. Xây dựng model

CNN sử dụng phép tích chập (Convolution) để giải quyết vấn đề của ảnh có tập điểm ảnh lớn (có kích thước N x N) và trích xuất ra những đặc trưng của hình, từ đây có thể sử dụng để nhận diện.

Ngoài các lớp mạng neuron kết nối đầy đủ cơ bản, CNN còn có một số lớp tích chập trước quá trình xử lý. Cấu trúc cơ bản của CNN bao gồm các lớp như sau:

$$\text{Image} \rightarrow \text{Convolution layer(Conv)} + \text{Pooling layer(Pool)} + \text{Fully connected layer(FC)} \rightarrow \text{Output}$$

Trong bài này em sẽ tiến hành tạo model CNN cơ bản như trên.

```
print(X_train.shape)
print(X_val.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_val.shape)
print(Y_test.shape)
```

```
(199, 96, 96, 3)
(40, 96, 96, 3)
(40, 96, 96, 3)
(199, 20)
(40, 20)
(40, 20)
```

Ở đây, shape của iuput vào model sẽ là (96,96,3).

Model xây dựng:

```
def create_custom_model():
    clear_session()

    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(96, 96, 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(512, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(CLASS_TOTAL, activation='softmax'))

    model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=LEARNING_RATE), metrics=['accuracy'])
    return model
```

Có 3 lớp Conv trong mô hình. Mỗi lớp Conv2D thực hiện phép tích chập 2D bằng cách sử dụng bộ lọc tăng dần (32, 64 và cuối cùng là 128) với kích thước kernel là 3x3, hàm kích hoạt là relu(phi tuyến tính).

Sau mỗi lớp Conv2D, một lớp MaxPooling2D được thêm vào để thực hiện max pooling với kích thước pool là 2x2. Max pooling giúp giảm kích thước của bản đồ đặc trưng và trích xuất thông tin quan trọng nhất.

Sau các lớp này, sử dụng lớp Flatten để biến đổi các đặc trưng trên từ một vector 2D sang 1D.

Fully connected (dense) layer, lớp Dense đầu tiên có 512 đơn vị và sử dụng hàm kích hoạt ReLU. Lớp Dropout ngẫu nhiên đặt 50% các đơn vị đầu vào thành 0 trong quá trình huấn luyện, giúp giảm hiện tượng overfitting, điều này giúp ngăn mô hình phụ thuộc quá nhiều vào các tính năng cụ thể và cải thiện khả năng khái quát hóa của nó. Lớp Dense cuối cùng có số đơn vị bằng với số lượng lớp trong tập dữ liệu (CLASS_TOTAL=20). Nó sử dụng hàm kích hoạt softmax, chuyển đổi giá trị đầu ra thành xác suất.

Complie: sử dụng loss='categorical_crossentropy' - phù hợp để tranning data gồm nhiều class, optimizer=Adam(lr=LEARNING_RATE=0.00006).

3. Báo cáo kết quả

Load data:

```
from matplotlib import pyplot as plt
import os
import numpy as np
%matplotlib inline

BASE_DIR = os.path.join(os.getcwd(), '/content/drive/MyDrive/FEI/image')
CLASSES = os.listdir(BASE_DIR)
CLASS_TOTAL = len(CLASSES)
EPOCH = 40
BATCH_SIZE = 5
SIZE = (96, 96)
LEARNING_RATE = 0.00006
BRIGHTNESS = 1.1

print("CLASS TOTAL: ", CLASS_TOTAL)
```

```
CLASS TOTAL:  20
```

```
CLASSES
```

```
['john',
 'Caleb ',
 'Lawson',
 'Barnes',
 'Hughes',
 'Bennett',
 'William ',
 'Henry ',
 'Emma ',
 'Lucas ',
 'Sophia ',
 'Noah ',
 'Harper ',
 'Jackson ',
 'Ethan ',
 'Oliver',
 'Benjamin',
 'Maxine',
 'Stone',
 'Sullivan']
```

Preprocessing:

```
def preprocessing(path):
  img = load_img(path, target_size=SIZE, color_mode="grayscale")
  enhancer = ImageEnhance.Brightness(img)
  img = enhancer.enhance(BRIGHTNESS)
  img_array = img_to_array(img)
  img_array = np.dstack([img_array] * 3)
  return img_array
```

Ta sẽ load ảnh với kích thước 96x96 về gray scale, sau đó xử lý độ sáng, và chuyển đổi thông tin ảnh về dạng array 3 chiều.

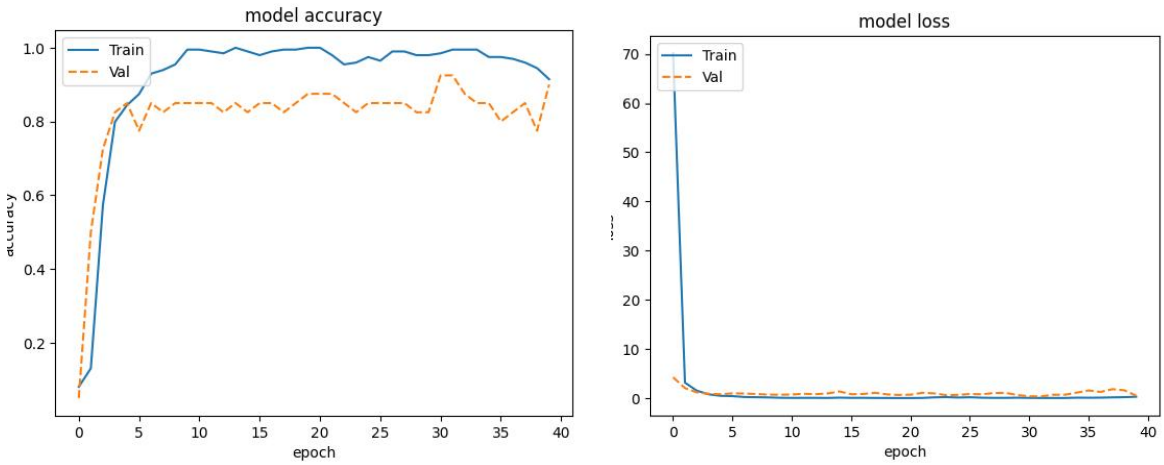
Split data:

```
print(X_train.shape)
print(X_val.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_val.shape)
print(Y_test.shape)
```

```
(199, 96, 96, 3)
(40, 96, 96, 3)
(40, 96, 96, 3)
(199, 20)
(40, 20)
(40, 20)
```

Dữ liệu sẽ được chia ra train, test và val bằng hàm train_test_split() sau đó được xáo trộn lên.

Model:



Ở đây, ta thấy model vẫn không được tốt lắm (bị overfitting) một phần khiến dữ liệu bị như vậy là vì dataset. Dữ liệu ảnh của ta là rất ít, mỗi người chỉ gồm tầm 14 sample, ngoài ra mô hình chỉ được học về những dữ liệu về đúng những

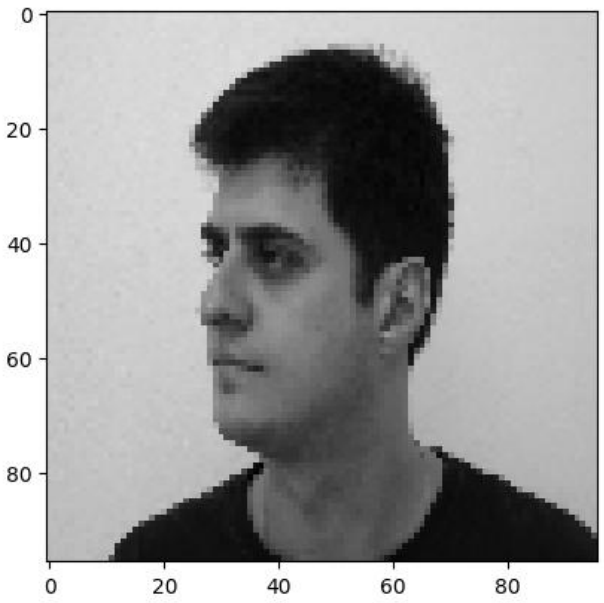
người này, không có ngoại cảnh cũng như phải phân biệt những người khác(trong cùng 1 sample chỉ đúng người đó).

Kết quả:

```
loss, accuracy = model.evaluate(X_test, Y_test)

print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
```

```
2/2 [==============================] - 0s 74ms/step - loss: 0.3356 - accuracy: 0.9250
Test Loss: 0.33559590578079224
Test Accuracy: 0.925000011920929
```




```
1/1 [==============================] - 0s 32ms/step
Predicted class: john
Predicted probabilities: [9.9920219e-01 1.5709993e-09 2.3112156e-07 6.2994472e-13 5.0062499e-10
 2.0522136e-11 4.2661913e-10 7.7166824e-06 4.7989852e-11 1.0972894e-08
 4.2007636e-10 7.5864858e-07 7.8765838e-04 3.6846842e-08 1.3401682e-06
 4.6394247e-12 1.1141141e-12 4.6297254e-13 1.0710450e-08 2.2683895e-11]
```

Kết luận

Model có tỉ lệ chính xác rất cao, tuy nhiên, việc sử dụng trong thực tế lại không khả thi vì model bị over fitting. Khi test trên các dữ liệu khác ngoài tập dữ liệu của ta (có thể cùng các người trong tập dữ liệu nhưng khác góc chụp, độ sáng, …) thì mức độ chính xác lại không đáng kể.