BCHT

Phần 1: đọc hiểu của file code

**BÁO CÁO PHÂN TÍCH VÀ SO SÁNH MÔ HÌNH CNN**

## 1. Quy trình chung của các mô hình CNN

Cả ba file đều tuân theo quy trình chung sau:

1. **Load dữ liệu**: Import dữ liệu và chia thành tập train/test.
2. **Tiền xử lý**: Chuẩn hóa, thay đổi kích thước ảnh, chuyển đổi dữ liệu.
3. **Xây dựng mô hình CNN**: Dùng các layer Conv2D, MaxPooling2D, Flatten, Dense,...
4. **Compile và huấn luyện**: Sử dụng optimizer, loss function phù hợp.
5. **Đánh giá mô hình**: Tính accuracy, loss và trực quan hóa kết quả.

## 2. Chi tiết các mô hình

### 2.1. Mô hình nhận dạng chữ số viết tay (****1\_MNIST\_CNN****)

* **Dữ liệu**: Bộ dữ liệu MNIST gồm các ảnh chữ số (28x28, grayscale).
* **Mô hình**:
  + 2 lớp Convolutional (Conv2D) + MaxPooling2D.
  + Lớp Flatten, Fully Connected (Dense) với Softmax đầu ra (10 lớp - số 0-9).
* **Huấn luyện**: Adam optimizer, categorical cross-entropy loss.
* **Kết quả**: Accuracy cao (>98%), thời gian huấn luyện nhanh do dữ liệu đơn giản.

**Trích đoạn code:**

* Load dữ liệu: mnist = keras.datasets.mnist.load\_data()
* Chuẩn hóa dữ liệu: x\_train, x\_test = x\_train / 255.0, x\_test / 255.0
* Xây dựng mô hình CNN:

model = Sequential([

Conv2D(32, kernel\_size=(3,3), activation='relu', input\_shape=(28,28,1)),

MaxPooling2D(pool\_size=(2,2)),

Flatten(),

Dense(128, activation='relu'),

Dense(10, activation='softmax')

])

* Compile và huấn luyện:

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.fit(x\_train, y\_train, epochs=10, validation\_data=(x\_test, y\_test))

### 2.2. Mô hình phân loại giới tính từ ảnh khuôn mặt (****2\_Human\_Gender\_CNN****)

* **Dữ liệu**: Bộ dữ liệu ảnh khuôn mặt với nhãn Nam/Nữ.
* **Mô hình**:
  + 3 lớp Convolutional (Conv2D) + MaxPooling2D.
  + Thêm Dropout để giảm overfitting.
  + Lớp Fully Connected với Softmax đầu ra (2 lớp - Nam/Nữ).
* **Huấn luyện**: Adam optimizer, binary cross-entropy loss.
* **Kết quả**: Accuracy khá tốt (~90%), cần nhiều dữ liệu hơn để tối ưu.

**Trích đoạn code:**

* Load dữ liệu: train\_datagen = ImageDataGenerator(rescale=1./255)
* Tiền xử lý dữ liệu ảnh:

train\_generator = train\_datagen.flow\_from\_directory(

'dataset/train',

target\_size=(128, 128),

batch\_size=32,

class\_mode='binary'

)

* Xây dựng mô hình CNN:

model = Sequential([

Conv2D(32, kernel\_size=(3,3), activation='relu', input\_shape=(128,128,3)),

MaxPooling2D(pool\_size=(2,2)),

Dropout(0.25),

Flatten(),

Dense(128, activation='relu'),

Dropout(0.5),

Dense(1, activation='sigmoid')

])

* Compile và huấn luyện:

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(train\_generator, epochs=20)

### 2.3. Mô hình nhận diện cảm xúc từ ảnh khuôn mặt (****3\_Emotion\_Reg\_CNN****)

* **Dữ liệu**: Bộ dữ liệu ảnh khuôn mặt có nhiều biểu cảm (buồn, vui, giận dữ,...).
* **Mô hình**:
  + 4 lớp Convolutional (Conv2D) + MaxPooling2D.
  + Thêm Batch Normalization và Dropout để cải thiện generalization.
  + Lớp Fully Connected với Softmax đầu ra (n lớp - số lượng cảm xúc).
* **Huấn luyện**: Adam optimizer, categorical cross-entropy loss.
* **Kết quả**: Accuracy trung bình (~65-75%), do bài toán phức tạp và dữ liệu khó phân biệt.

**Trích đoạn code:**

* Load dữ liệu: data = pd.read\_csv('fer2013.csv')
* Tiền xử lý dữ liệu:

x\_train = x\_train.reshape(-1, 48, 48, 1) / 255.0

y\_train = to\_categorical(y\_train, num\_classes=7)

* Xây dựng mô hình CNN:

model = Sequential([

Conv2D(64, kernel\_size=(3,3), activation='relu', input\_shape=(48,48,1)),

BatchNormalization(),

MaxPooling2D(pool\_size=(2,2)),

Dropout(0.25),

Flatten(),

Dense(256, activation='relu'),

Dropout(0.5),

Dense(7, activation='softmax')

])

* Compile và huấn luyện:

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.fit(x\_train, y\_train, epochs=30, validation\_data=(x\_test, y\_test))

## 3. So sánh các mô hình

| **Tiêu chí** | **1\_MNIST\_CNN** | **2\_Human\_Gender\_CNN** | **3\_Emotion\_Reg\_CNN** |
| --- | --- | --- | --- |
| **Bài toán** | Nhận dạng chữ số viết tay | Phân loại giới tính | Nhận diện cảm xúc |
| **Dữ liệu** | MNIST (28x28, grayscale) | Ảnh khuôn mặt | Ảnh khuôn mặt có biểu cảm |
| **Số lớp CNN** | 2 lớp Conv2D | 3 lớp Conv2D | 4 lớp Conv2D |
| **Có Dropout?** | Không | Có | Có |
| **Có BatchNorm?** | Không | Không | Có |
| **Số lớp đầu ra** | 10 (0-9) | 2 (Nam/Nữ) | Nhiều hơn (cảm xúc) |
| **Ứng dụng** | Nhận dạng chữ số | Xác định giới tính | Phân tích cảm xúc |
| **Accuracy** | >98% | ~90% | 65-75% |

## 4. Nhận xét

* **MNIST** dễ nhận dạng nhất do chữ số có hình dạng đơn giản, ít nhiễu.
* **Phân loại giới tính** khó hơn do có nhiều đặc điểm khuôn mặt cần nhận dạng.
* **Nhận diện cảm xúc** là bài toán khó nhất, đòi hỏi nhiều dữ liệu và mô hình sâu hơn để đạt độ chính xác cao hơn.
* Việc thêm **Dropout** và **Batch Normalization** giúp mô hình tránh overfitting nhưng cũng có thể làm giảm accuracy nếu không tinh chỉnh đúng.

## 5. Kết luận

Báo cáo này giúp nhận diện sự khác nhau giữa các bài toán CNN và từ đó có thể tối ưu mô hình phù hợp với từng bài toán.