### 1. Nhận diện chữ viết tay

**Kỹ thuật sử dụng:**

* **Mạng nơ-ron tích chập (CNN)**: Đây là một trong những kỹ thuật phổ biến cho nhận diện hình ảnh, bao gồm cả chữ viết tay.
* **Mô hình học sâu (Deep Learning)**: Sử dụng các mô hình như LSTM (Long Short-Term Memory) hoặc Transformer để nhận diện chuỗi ký tự.

### 2. Dữ liệu train

**Dữ liệu train cần thiết:**

* **Hình ảnh chữ viết tay**: Bạn cần một bộ dữ liệu chứa hình ảnh chữ viết tay từ nhiều người viết khác nhau. Bộ dữ liệu này nên bao gồm nhiều loại chữ cái, chữ số và ký hiệu.
* **Gán nhãn**: Mỗi hình ảnh cần có một nhãn tương ứng cho văn bản mà nó đại diện. Ví dụ: hình ảnh chữ viết tay "Hello" sẽ có nhãn là "Hello".

### 3. Mô hình để train

**Mô hình đề xuất:**

* **Cascaded CNN + RNN**: Kết hợp CNN để trích xuất đặc trưng từ hình ảnh và RNN để nhận diện chuỗi ký tự.
* **CRNN (Convolutional Recurrent Neural Network)**: Là mô hình thường được sử dụng cho nhận diện văn bản viết tay. Nó kết hợp các lớp CNN để trích xuất đặc trưng và các lớp RNN để xử lý chuỗi.
* **TensorFlow hoặc PyTorch**: Sử dụng một trong hai thư viện này để xây dựng và huấn luyện mô hình.

### 4. Quy trình thực hiện

1. **Thu thập dữ liệu**: Tìm kiếm và thu thập bộ dữ liệu chữ viết tay, như IAM Handwriting Database hoặc chữ viết tay tự thu thập.
2. **Tiền xử lý dữ liệu**: Làm sạch và chuẩn hóa hình ảnh, có thể bao gồm cắt xén, điều chỉnh kích thước và chuyển đổi màu sắc.
3. **Huấn luyện mô hình**: Sử dụng bộ dữ liệu đã chuẩn bị để huấn luyện mô hình CNN hoặc CRNN.
4. **Nhận diện**: Triển khai mô hình đã huấn luyện để nhận diện chữ viết tay từ hình ảnh.
5. **Chuyển đổi sang PDF**: Sử dụng thư viện như reportlab hoặc fpdf để tạo file PDF từ văn bản đã nhận diện.

### 5. Thư viện hữu ích

* **OpenCV**: Để xử lý hình ảnh.
* **Pytesseract**: Để nhận diện văn bản (nếu cần).
* **TensorFlow/PyTorch**: Để xây dựng và huấn luyện mô hình học sâu.
* **Pandas**: Để xử lý và quản lý dữ liệu.