TRƯỜNG ĐẠI HỌC ĐẠI NAM

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC

HỌC PHẦN: HỆ THỐNG NHÚNG

TÊN ĐỀ TÀI: HỆ THỐNG CẢNH BÁO KHÍ GAS

Sinh viên thực hiện: Ngô Đặng Tuấn Anh
Email: tuananh301105@gmail.com
Ngành: Công nghệ Thông tin
Chuyên ngành: Hệ thống thông tin

Giảng viên hướng dẫn: ThS. Lê Trung Hiếu, KS.Nguyễn Thái Khánh
Khoa: Công nghệ Thông tin

HÀ NỘI, 08/2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐẠI NAM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN

CHUYỂN ĐỔI SỐ

NHẬN DẠNG VÀ PHÂN LOẠI GIẤY TỜ

Sinh viên thực hiện : Ngô Đặng Tuấn Anh

Ngành : Công nghệ thông tin

Giảng viên hướng dẫn : ThS.Lê Trung Hiếu

KS.Nguyễn Thái Khánh

Lời cảm ơn

Trong bối cảnh các cơ quan, doanh nghiệp đẩy mạnh chuyển đổi số để tối ưu quy trình vận hành, nhu cầu số hóa và tự động hóa xử lý giấy tờ ngày càng cấp thiết. Công việc nhận diện và phân loại các loại văn bản—như CMND/CCCD, bằng lái, hoá đơn, hợp đồng, phiếu thu/chi, công văn—nếu làm thủ công thường tốn nhiều thời gian, dễ sai sót và khó truy xuất. Nhiều đơn vị hiện vẫn dựa vào nhập liệu tay hoặc tra cứu thủ công, dẫn đến độ trễ xử lý, thiếu minh bạch về nguồn dữ liệu và khó khăn trong việc kiểm soát phiên bản, lịch sử chỉnh sửa.

Xuất phát từ thực tế đó, đề tài "Xây dựng hệ thống nhận diện và phân loại giấy tờ" được thực hiện nhằm phát triển một giải pháp số hóa trọn vẹn vòng đời tài liệu: từ khâu tiếp nhận ảnh/chứng từ, tiền xử lý (chỉnh nghiêng, khử nhiễu, tăng tương phản), nhận dạng vùng văn bản, OCR trích xuất nội dung, đến phân loại tự động theo loại giấy tờ và trích xuất trường thông tin quan trọng (tên, số định danh, ngày cấp, số tiền...). Hệ thống hướng tới mục tiêu rút ngắn thời gian nhập liệu, nâng cao độ chính xác, tiêu chuẩn hóa cấu trúc dữ liệu đầu ra và hỗ trợ tra cứu–lưu trữ tập trung, phục vụ kiểm toán cũng như khai thác dữ liệu về sau.

Mặc dù đã nỗ lực để bảo đảm tính ứng dụng và độ tin cậy của mô hình, do hạn chế về thời gian và kinh nghiệm, đề tài khó tránh khỏi những thiếu sót—đặc biệt ở các trường hợp ảnh đầu vào chất lượng kém hoặc bố cục tài liệu không chuẩn. Em rất mong nhận được những ý kiến đóng góp của thầy cô để hoàn thiện hệ thống, nâng cao độ chính xác nhận dạng–phân loại và tối ưu hiệu năng triển khai trong môi trường thực tế.

Mục lục

Danh sách hình vẽ

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Trong bối cảnh các cơ quan, doanh nghiệp đang đẩy mạnh chuyển đổi số nhằm tối ưu hóa quy trình vận hành, nhu cầu số hóa và tự động hóa xử lý giấy tờ trở nên ngày càng cấp thiết. Công việc nhận diện và phân loại các loại văn bản—như CMND/CCCD, giấy phép lái xe, hóa đơn, hợp đồng, phiếu thu/chi hay công văn—nếu thực hiện thủ công thường mất nhiều thời gian, dễ xảy ra sai sót và khó khăn trong việc lưu trữ, tra cứu. Hiện nay, nhiều đơn vị vẫn còn dựa vào hình thức nhập liệu tay hoặc tìm kiếm truyền thống, dẫn đến quy trình xử lý chậm, thiếu minh bạch về dữ liệu và khó kiểm soát lịch sử thay đổi tài liệu.

Từ thực tế đó, đề tài "Xây dựng hệ thống nhận diện và phân loại giấy tờ" được triển khai với mục tiêu phát triển một giải pháp số hóa toàn diện cho vòng đời tài liệu: từ khâu tiếp nhận ảnh/chứng từ, tiền xử lý (chỉnh nghiêng, khử nhiễu, tăng cường chất lượng), nhận dạng vùng văn bản, OCR trích xuất nội dung, cho đến phân loại tự động theo từng loại giấy tờ và trích xuất các trường thông tin quan trọng (họ tên, số định danh, ngày cấp, số tiền,...). Hệ thống hướng đến việc rút ngắn thời gian xử lý, nâng cao độ chính xác, chuẩn hóa cấu trúc dữ liệu và hỗ trợ việc lưu trữ – tra cứu tập trung, phục vụ cả công tác quản lý nội bộ lẫn khai thác dữ liệu lâu dài.

1.2 Mục tiêu của đề tài

Mục tiêu tổng quát: Xây dựng hệ thống số hóa giúp nhận diện (OCR) và phân loại tự động giấy tờ, rút ngắn thời gian xử lý, nâng cao độ chính xác và hỗ trợ lưu trữ – tra cứu hiệu quả.

Mục tiêu cụ thể:

- Tiếp nhận ảnh/chứng từ, tiền xử lý và nhận dạng văn bản bằng OCR.

- Phân loại tài liệu theo từng loại (CMND/CCCD, hóa đơn, hợp đồng, . . .).

- Trích xuất và chuẩn hóa thông tin quan trọng (họ tên, số định danh, ngày cấp, số tiền,. . .).

- Lưu trữ, tìm kiếm, quản lý dữ liệu tập trung và cung cấp API tích hợp.

- Mục tiêu kỹ thuật: Đảm bảo độ chính xác OCR cao

1.3 Yêu cầu phi chức năng

Yêu cầu phi chức năng (từ quy trình trên) - Tốc độ phân loại tài liệu nhanh, đảm bảo đáp ứng nhu cầu xử lý hàng loạt.

- OCR nhận diện ký tự chính xác cao

- Có thể mở rộng để hỗ trợ nhiều loại tài liệu khác nhau trong tương lai.

- Giao diện đơn giản, thông báo lỗi rõ ràng

1.4 Phương pháp nghiên cứu

- **Thu thập và nhập dữ liệu** Hình ảnh giấy tờ được chụp hoặc quét (scanner, camera).
- **Tiền xử lý ảnh**
  - Chuyển ảnh sang grayscale.
  - Lọc nhiễu, tăng độ tương phản.
  - Căn chỉnh, cắt viền, làm phẳng tài liệu.
  - Phân ngưỡng (binarization) để tách chữ khỏi nền.
- **Phân vùng và phát hiện văn bản**
  - Xác định vùng có chứa chữ.
  - Cắt tách thành từng dòng, từ hoặc ký tự.
- **Nhận dạng ký tự**
  - Sử dụng mô hình OCR (Tesseract, PaddleOCR, EasyOCR).
  - Chuyển hình ảnh chữ thành văn bản số hóa.
- **Hậu xử lý**

- Sửa lỗi nhận dạng (chính tả, font đặc biệt).
- Chuẩn hóa dữ liệu (định dạng ngày)

Chương 2

Cơ sở lý thuyết

2.1 Tổng quan về chuyển đổi số

Chuyển đổi số là quá trình ứng dụng công nghệ số vào mọi lĩnh vực của đời sống, kinh doanh, và quản lý. Nó không chỉ là việc số hóa dữ liệu, mà còn thay đổi cách thức tổ chức hoạt động, mô hình kinh doanh, và mang lại giá trị mới.

2.1.1 Ví dụ đơn giản

- Trước đây: đi làm thủ tục hành chính phải xếp hàng, nộp giấy tờ trực tiếp.
- Sau chuyển đổi số: có thể đăng ký dịch vụ công trực tuyến, ký số, thanh toán online $\rightarrow$ nhanh hơn, tiết kiệm thời gian, minh bạch hơn.

2.1.2 Khác biệt giữa "chuyển đổi số" và "số hóa"

- Số hóa (Digitization): chỉ là chuyển dữ liệu giấy sang file điện tử (PDF, Excel).
- Chuyển đổi số: là thay đổi cách làm việc, đưa công nghệ vào toàn bộ hoạt động (quản lý, sản xuất, kinh doanh, chăm sóc khách hàng...).

2.2 Các trụ cột của Chuyển đổi số

Chuyển đổi số được xây dựng dựa trên ba trụ cột chính:

- Chính phủ số: Ứng dụng công nghệ để hiện đại hóa hoạt động quản lý nhà nước, tăng tính minh bạch và phục vụ người dân tốt hơn.

- Kinh tế số: Sử dụng công nghệ để tạo ra các sản phẩm, dịch vụ và mô hình kinh doanh mới.
- Xã hội số: Thúc đẩy sự phát triển của cộng đồng thông minh, nơi mọi người được hưởng lợi từ tiện ích số trong học tập, y tế, giao thông và đời sống hằng ngày.

2.3 Mục tiêu của chuyển đổi số trong lĩnh vực giáo dục và ngôn ngữ

2.1.3 Vai trò trong giáo dục

- Hỗ trợ dạy và học hiệu quả: Công cụ số giúp sinh viên, học sinh viết đúng chính tả, từ đó nâng cao chất lượng bài tập, báo cáo, luận văn.
- Giảm tải cho giảng viên: Thay vì phải sửa lỗi chính tả thủ công, giảng viên có thể tập trung vào nội dung, tư duy và cách trình bày.
- Cá nhân hóa việc học: Người học dễ dàng nhận diện lỗi chính tả phổ biến của bản thân, từ đó cải thiện kỹ năng viết tiếng Việt.

2.1.4 Vai trò trong xử lý ngôn ngữ

- Tự động hóa công việc: Các thuật toán giúp phát hiện và gợi ý từ đúng nhanh chóng.
- Đảm bảo chuẩn ngôn ngữ: Văn bản học thuật, hành chính hay truyền thông đều yêu cầu chính tả chính xác.
- Tích hợp đa nền tảng: Công cụ kiểm tra lỗi có thể nhúng vào hệ thống quản lý học tập,ứng dụng chat, trình soạn thảo văn bản, hoặc website.

2.5 Lợi ích của Chuyển đổi số

Lợi ích mang lại của chuyển đổi số rất đa dạng:

- Đối với doanh nghiệp: tối ưu quy trình, tăng năng suất lao động, mở rộng thị trường, nâng cao trải nghiệm khách hàng.
- Đối với chính phủ: quản lý hiệu quả hơn, giảm chi phí hành chính, nâng cao mức độ hài lòng của người dân.
- Đối với xã hội: tạo cơ hội tiếp cận bình đẳng, thúc đẩy sáng tạo, hỗ trợ giáo dục và chăm sóc sức khỏe từ xa.

Một số khái niệm cơ bản

a) Ứng dụng công nghệ số

- Đề tài sử dụng AI (Trí tuệ nhân tạo) để xử lý hình ảnh, giúp nhận diện biển số xe chính xác và nhanh chóng.
- Hệ thống được kết nối với cơ sở dữ liệu tập trung để kiểm tra thông tin phương tiện, tài khoản thanh toán của chủ xe.
- Đây chính là biểu hiện rõ rệt của chuyển đổi số: thay đổi cách thức vận hành truyền thống sang quy trình số hóa, hiện đại và tự động.

b) Thúc đẩy kinh tế số

- Khi áp dụng thu phí không dừng, tiền được thanh toán qua ví điện tử, ngân hàng số, tài khoản giao thông $\rightarrow$ khuyến khích giao dịch không tiền mặt.
- Góp phần mở rộng hệ sinh thái thanh toán số, thúc đẩy kinh tế số phát triển.

c) Chính phủ số – Minh bạch, hiệu quả

- Dữ liệu xe ra vào trạm được lưu trữ và đồng bộ, hạn chế gian lận trong thu phí.
- Cơ quan quản lý có thể giám sát trực tuyến, phân tích dữ liệu giao thông theo thời gian thực.
- Đây là bước đi quan trọng trong việc xây dựng Chính phủ số trong lĩnh vực giao thông.

Chương 3

Phân tích yêu cầu và thiết kế

3.1 Mô tả thuật toán

Trong thực tế, các cơ quan và doanh nghiệp thường phải xử lý khối lượng lớn giấy tờ như CMND/CCCD, giấy phép lái xe, hóa đơn, hợp đồng, phiếu thu/chi hay công văn. Việc nhập liệu và phân loại thủ công tốn nhiều thời gian, dễ sai sót và khó khăn trong việc lưu trữ, tra cứu. Do đó cần một hệ thống tự động có khả năng nhận diện, phân loại và trích xuất thông tin từ ảnh hoặc file scan để chuẩn hóa dữ liệu và quản lý tập trung.

Bài toán đặt ra là: với đầu vào là hình ảnh hoặc PDF tài liệu có thể khác nhau về chất lượng (mờ, nghiêng, thiếu sáng), hệ thống phải thực hiện chuỗi xử lý gồm tiền xử lý ảnh, OCR nhận dạng ký tự, chuẩn hóa văn bản, phân loại tài liệu và trích xuất các trường thông tin quan trọng. Kết quả đầu ra là văn bản số hóa có cấu trúc, kèm nhãn loại giấy tờ, được lưu trữ trong cơ sở dữ liệu để phục vụ tìm kiếm và chia sẻ.

Mục tiêu chính là rút ngắn thời gian xử lý, giảm thiểu nhập liệu thủ công và nâng cao độ chính xác. Thách thức của bài toán nằm ở chất lượng ảnh đầu vào, sự đa dạng bố cục tài liệu và yêu cầu vừa đảm bảo độ chính xác cao, vừa giữ hiệu năng xử lý đáp ứng nhu cầu thực tế.

3.2 Phương pháp hoạt động

Hệ thống được xây dựng trên **Streamlit** với các thành phần chính:

- **OCR**: sử dụng PaddleOCR (tiếng Việt) để trích xuất văn bản từ ảnh.
- **Ngôn ngữ**: PhoBERT (transformers) tạo vector ngữ nghĩa của văn bản.
- **Phân loại**: Logistic Regression huấn luyện trên dữ liệu mẫu để dự đoán loại giấy tờ.

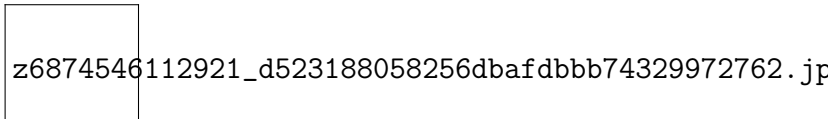
- **Giao diện**: cho phép upload ảnh, hiển thị kết quả OCR + loại giấy tờ, và tải kết quả dưới dạng CSV/Excel.

3.2.1 Luồng xử lý

- Người dùng upload ảnh (JPG/PNG).
- Ảnh được đưa vào OCR → xuất chuỗi văn bản.
- Văn bản chuyển sang vector embedding bằng PhoBERT.
- Classifier dự đoán nhãn tài liệu (ví dụ: hóa đơn, quyết định, đơn, thông báo).
- Kết quả (loại + nội dung OCR) được hiển thị và cho phép tải về.

3.3 Sơ đồ hoạt động

3.3.1 Sơ đồ khối



Hình 3.1: Sơ đồ khối mô tả hoạt động.

3.3.2 Quy trình xử lý

- **Đầu vào – Hình ảnh tài liệu** Người dùng cung cấp ảnh scan hoặc ảnh chụp tài liệu (hợp đồng, hóa đơn, quyết định, CCCD, giấy khám sức khỏe...). Hệ thống tiếp nhận và đưa vào bước tiền xử lý.
- **Tiền xử lý ảnh** Chuẩn hoá kích thước ảnh, xoay đúng chiều, loại bỏ nhiễu, tăng độ tương phản để cải thiện chất lượng OCR. Nếu ảnh quá mờ hoặc hỏng → hệ thống báo lỗi và yêu cầu nhập lại.
- **Nhận dạng ký tự quang học (OCR – PaddleOCR)** PaddleOCR quét toàn bộ ảnh, tách thành từng dòng văn bản. Kết quả trả về là chuỗi text thô, ghép từ các dòng đã nhận diện.
- **Chuẩn hoá và trích xuất văn bản** Văn bản OCR được làm sạch: xoá ký tự đặc biệt, chuẩn hóa font/encoding.
- **Phân loại tài liệu** Văn bản sau OCR được đưa vào mô hình PhoBERT để biến thành vector ngữ nghĩa. Bộ phân loại dựa trên embedding này xác định loại tài liệu (ví dụ: hợp đồng,

hóa đơn, quyết định, CCCD...). Nếu phân loại thất bại → gán nhãn thủ công để cải thiện mô hình.

- **Kết quả phân loại** Hệ thống trả về loại tài liệu.
- **Lưu trữ vào cơ sở dữ liệu** Thông tin OCR + nhãn loại tài liệu được lưu vào CSDL quản lý. Tài liệu có thể được tìm kiếm, trích xuất hoặc chia sẻ sau này.

Chương 4

Triển khai

4.1 Môi trường và phát triển công cụ

4.1.1 Môi trường lập trình

Ngôn ngữ lập trình: Python 3.11. Python được lựa chọn vì tính linh hoạt, cú pháp dễ đọc và hệ sinh thái thư viện phong phú, hỗ trợ mạnh mẽ cho các tác vụ khoa học dữ liệu, xử lý ảnh và phát triển ứng dụng web.

Công cụ phát triển: Visual Studio Code.

1. Cài đặt các thư viện

Sử dụng các lệnh sau để cài đặt toàn bộ thư viện cần thiết cho dự án:

```
pip install torch torchvision torchaudio
pip install transformers
pip install scikit-learn
pip install joblib
pip install paddleocr
pip install paddlepaddle -U
pip install opencv-python
pip install pillow
pip install pytesseract
pip install streamlit
```

2. Cài đặt Tesseract OCR Engine

Hệ thống sử dụng Tesseract OCR Engine, do đó cần tải xuống và cài đặt thêm công cụ này.

- Đối với Windows: tải bộ cài đặt từ trang GitHub chính thức của Tesseract
- Sau khi cài đặt, thiết lập đường dẫn đến file `tesseract.exe` trong mã nguồn. Cụ thể, thêm dòng lệnh sau vào đầu file:

```
import pytesseract
pytesseract.pytesseract.tesseract_cmd =
    r"C:\Program Files\Tesseract-OCR\tesseract.exe"
```

Ứng dụng được xây dựng trên Python 3.10 với nhiều thư viện hỗ trợ, mỗi thư viện đảm nhận một vai trò riêng trong toàn bộ quy trình xử lý ảnh và phân loại giấy tờ:

- Streamlit: Cung cấp môi trường nhanh gọn để phát triển giao diện web. Người dùng có thể tải ảnh, xem kết quả OCR, loại giấy tờ và tải dữ liệu xuất ra mà không cần lập trình front-end phức tạp.
- OpenCV: Thư viện xử lý ảnh mạnh mẽ, đảm nhiệm các bước làm sạch ảnh như chuyển sang thang xám, khử nhiễu, tăng tương phản và phân ngưỡng, giúp cải thiện chất lượng đầu vào cho OCR.
- Pillow: Công cụ xử lý ảnh cơ bản, hỗ trợ mở, chuyển đổi và chuẩn bị dữ liệu hình ảnh trước khi đưa qua OpenCV hoặc OCR.
- PyTesseract: Giao diện Python cho Tesseract OCR Engine. Cho phép trích xuất văn bản số hóa từ ảnh tài liệu, đặc biệt hữu ích với văn bản in rõ nét.
- PaddleOCR & PaddlePaddle: Hệ thống OCR tối ưu cho nhiều ngôn ngữ, trong đó có tiếng Việt. PaddleOCR giúp nhận dạng chính xác hơn với ảnh phức tạp, chữ in mờ hoặc bố cục nhiều cột.
- Torch: Nền tảng tính toán tensor và deep learning. Đóng vai trò backend cho PhoBERT, hỗ trợ tính toán embedding văn bản trên CPU/GPU.
- Transformers: Thư viện của HuggingFace. Trong dự án, PhoBERT được sử dụng để biến đổi văn bản OCR thành vector đặc trưng, phục vụ phân loại.
- Scikit-learn: Cung cấp các thuật toán machine learning. Logistic Regression được sử dụng để dự đoán loại tài liệu dựa trên embedding từ PhoBERT.
- Joblib: Dùng để lưu và tải lại mô hình Logistic Regression, giúp triển khai nhanh mà không cần huấn luyện lại.
- Pandas & XlsxWriter: Hỗ trợ xử lý kết quả OCR, quản lý bảng dữ liệu và xuất file CSV/Excel để lưu trữ hoặc chia sẻ.

- Regex (re): Thư viện chuẩn của Python, được dùng để lọc và chuẩn hóa văn bản (ví dụ: loại bỏ ký tự đặc biệt, định dạng ngày tháng).

4.1.2 Cấu trúc và luồng xử lý hệ thống

Giao diện được xây dựng bằng Streamlit, một thư viện giúp tạo các ứng dụng web tương tác một cách nhanh chóng. Giao diện người dùng được thiết kế đơn giản và trực quan, gồm các thành phần chính sau:

Gồm trang chính:

Tải ảnh: Cho phép người dùng tải lên một hoặc nhiều file hình ảnh hóa đơn từ máy tính.

Vùng hiển thị kết quả: Một cột hiển thị ảnh gốc và ảnh sau khi đã được tiền xử lý, giúp người dùng dễ dàng so sánh chất lượng. Cột còn lại hiển thị kết quả OCR và các trường dữ liệu đã được trích xuất.

Vùng phân loại kết quả dự đoán: Hiển thị kết quả dự đoán phân loại giấy tờ sau khi phân tích hình ảnh.

4.1.3 Cấu trúc và luồng xử lý hệ thống

- **Nhận đầu vào:**
  - Hệ thống tiếp nhận hình ảnh từ người dùng.
- **Tiền xử lý ảnh:**
  - Hình ảnh được đưa vào hàm `preprocess()`.
  - Tự động điều chỉnh để tối ưu hóa cho OCR:
    - Chuyển ảnh sang định dạng xám.
    - Khử nhiễu.
    - Áp dụng CLAHE + Otsu hoặc Adaptive Thresholding để chuyển thành ảnh nhị phân.
- **Thực hiện OCR:**
  - Kết quả tiền xử lý được đưa vào hàm `run_ocr()`.
  - Thử nghiệm nhiều chế độ **PSM (Page Segmentation Mode)** của Tesseract để chọn chế độ phù hợp nhất với bố cục hóa đơn.

  - Trả về toàn bộ văn bản thô được nhận dạng.

- **Phân loại tài liệu:**
  - Các đặc trưng được đưa vào mô hình phân loại.
  - Hệ thống tự động xác định loại giấy tờ dựa trên **nội dung, cấu trúc và định dạng**.
  - Trả về kết quả phân loại.

4.2 Phân tích mã nguồn

4.2.1 Ý tưởng bài toán

Trong thực tế, các cơ quan, doanh nghiệp hay cá nhân thường phải xử lý một lượng lớn giấy tờ, văn bản hành chính như: hóa đơn, quyết định, đơn từ, thông báo,... Việc phân loại và trích xuất thông tin từ các loại giấy tờ này nếu làm thủ công sẽ mất thời gian, dễ nhầm lẫn và tốn nhiều nhân lực.

- Nhận diện chữ trong ảnh giấy tờ (OCR – Optical Character Recognition).
  - Sử dụng PaddleOCR để quét và trích xuất văn bản tiếng Việt từ ảnh chụp/tệp scan.
- Biểu diễn nội dung văn bản dưới dạng vector số.
  - Dùng PhoBERT – mô hình ngôn ngữ tiếng Việt hiện đại – để mã hóa văn bản, tạo ra vector đặc trưng phản ánh ngữ nghĩa của tài liệu.
- Phân loại tài liệu theo từng nhóm.
  - Ứng dụng mô hình học máy (Logistic Regression) để dự đoán loại giấy tờ dựa trên vector đặc trưng.
- Triển khai giao diện trực quan để người dùng có thể tải ảnh giấy tờ, xem kết quả OCR, loại tài liệu và xuất ra file (CSV/Excel).

4.2.2 Huấn luyện mô hình phân loại giấy tờ

- Khởi tạo PhoBERT:

```
MODEL_NAME = "vinai/phobert-base"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModel.from_pretrained(MODEL_NAME)
```

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
```

- Giải thích:

- PhoBERT là mô hình ngôn ngữ tiếng Việt dựa trên BERT.
- tokenizer: Chuyển văn bản thành các token số để mô hình hiểu.
- model: PhoBERT để sinh vector đặc trưng.
- device: tự động chọn GPU nếu có, tăng tốc huấn luyện.

- Khởi tạo OCR:

```
ocr = PaddleOCR(use_angle_cls=True, lang="vi")
```

- Giải thích:

- Sử dụng PaddleOCR để đọc chữ trong ảnh tiếng Việt.
- use_angle_cls=True: hỗ trợ xoay chữ.
- Kết quả OCR sẽ được dùng làm đầu vào cho PhoBERT.

- Hàm OCR ảnh:

```
def extract_text(image_path: str) -> str:
    result = ocr.ocr(image_path)
    lines = []
    if result:
        for res in result:
            for line in res:
                lines.append(line[1][0])
    return " ".join(lines)
```

- Giải thích:

- Đọc ảnh và trích xuất các dòng văn bản.
- Kết quả trả về là một chuỗi text.

- Hàm sinh vector từ văn bản:

```
def embed_text(text: str) -> np.ndarray:
    if not text.strip():
        return np.zeros(768)
    inputs = tokenizer(text, return_tensors="pt", padding=True, truncation=True, ma
    with torch.no_grad():
        outputs = model(**inputs)
    return outputs.last_hidden_state.mean(dim=1).cpu().numpy().flatten()
```

- Giải thích:

- Nếu text rỗng → trả về vector 768 số 0.
- Với text có nội dung:
  - tokenizer: mã hóa văn bản.
  - model: PhoBERT tạo ra biểu diễn vector.
  - mean(dim=1): lấy trung bình embedding của các token → vector duy nhất (768 chiều).

- Hàm xây dựng dataset:

```
def build_dataset(dataset_dir: str):
    X, y = [], []
    labels = sorted(os.listdir(dataset_dir))
    for idx, label in enumerate(labels):
        folder = os.path.join(dataset_dir, label)
        ...
                text = extract_text(img_path)
                vec = embed_text(text)
                X.append(vec)
                y.append(idx)
    return np.array(X), np.array(y), labels
```

- Giải thích:

- Duyệt qua thư mục dữ liệu (theo từng nhãn).
- Với mỗi ảnh:
  - OCR → văn bản.
  - Văn bản → vector PhoBERT.
  - Lưu vector vào X, nhãn vào y.

- Huấn luyện và lưu mô hình:

```
clf = LogisticRegression(max_iter=2000)
clf.fit(X, y)

joblib.dump(clf, "doc_classifier.pkl")
with open("labels.json", "w", encoding="utf-8") as f:
    json.dump(labels, f, ensure_ascii=False, indent=2)
```

- Giải thích:

- Logistic Regression được chọn làm mô hình phân loại.
- Sau khi train:
  - Lưu model (doc_classifier.pkl).
  - Lưu danh sách nhãn (labels.json).

4.3 Chạy chương trình

Sử sụng lệnh để chạy: streamlit run app.py

Khi chạy lệnh sẽ hiển thị ra trang giao diện như sau:

Sau đó, người dùng tải ảnh cần phân loại lên, ví dụ:

Kết quả sau khi quét, nhận dạng được đây là giấy tờ hóa đơn

Hình 4.1: Dao diện chương trình.

Hình 4.2: Hình ảnh cần phân loại.

Hình 4.3: Kết quả sau khi nhận dạng.

Chương 5

Kết luận

Đề tài "Xây dựng hệ thống nhận diện và phân loại giấy tờ" đã cho thấy hiệu quả của việc ứng dụng công nghệ xử lý ảnh và học máy trong việc số hóa quy trình quản lý tài liệu. Thông qua việc kết hợp PaddleOCR để trích xuất văn bản, PhoBERT để biểu diễn ngữ nghĩa và mô hình phân loại để nhận dạng loại giấy tờ, hệ thống đã minh họa được khả năng tự động hóa các bước từ nhập liệu, xử lý đến lưu trữ dữ liệu. Kết quả đạt được giúp tiết kiệm thời gian, nâng cao độ chính xác và tạo thuận lợi cho việc quản lý, tìm kiếm, cũng như khai thác thông tin trong thực tế.

Trong quá trình thực hiện, nhóm đã tích lũy được nhiều kinh nghiệm về xử lý dữ liệu đầu vào, thiết kế quy trình OCR – phân loại, lựa chọn mô hình học máy phù hợp và triển khai hệ thống qua giao diện web. Bên cạnh đó, đề tài cũng chỉ ra những hạn chế nhất định như dữ liệu huấn luyện còn ít, chưa có bước tiền xử lý ảnh chuyên sâu và chưa tận dụng đầy đủ thông tin bố cục (layout) của tài liệu.

Trong tương lai, hệ thống có thể được mở rộng với tập dữ liệu lớn và đa dạng hơn, bổ sung các kỹ thuật tiền xử lý nâng cao, tích hợp trích xuất thông tin có cấu trúc (tên, số định danh, ngày, số tiền…) và tối ưu hiệu năng để xử lý trên quy mô lớn. Ngoài ra, việc triển khai thực tế có thể đi kèm cơ chế bảo mật, phân quyền và API tích hợp, nhằm đưa hệ thống trở thành một giải pháp hỗ trợ quản lý tài liệu số hóa hiệu quả cho các cơ quan, doanh nghiệp.

Tài liệu tham khảo

1. Smith, R. (2007). *An Overview of the Tesseract OCR Engine*. Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR).

2. Baevski, A., et al. (2022). *Donut: Document Understanding Transformer without OCR*. arXiv:2203.11904.

3. Xu, Y., et al. (2020). *LayoutLM: Pre-training of Text and Layout for Document Image Understanding*. Proceedings of the 26th ACM SIGKDD Conference.

4. PaddleOCR Team. (2023). *PaddleOCR: Practical Ultra Lightweight OCR System*. GitHub Repository: https://github.com/PaddlePaddle/PaddleOCR.