

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN TRÍ TUỆ NHÂN TẠO



NGUYỄN THỊ PHÚ - 21110600

BÙI QUANG THIỆN - 21110656

Đề Tài :

**HỆ THỐNG THỦ THƯ THÔNG MINH HỖ TRỢ
SINH VIÊN TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ
THUẬT TP.HCM**

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ CNTT

GIÁO VIÊN HƯỚNG DẪN

ThS. LÊ MINH TÂN

KHÓA 2021 – 2025

LỜI CẢM ƠN

Để hoàn thành đề tài này, trước hết, chúng em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô trong khoa Công Nghệ Thông Tin – Trường Đại học Sư phạm Kỹ thuật TP.HCM, đã trang bị cho chúng em nền tảng kiến thức vững chắc cùng những kỹ năng cần thiết trong suốt quá trình học tập 4 năm tại trường.

Chúng em cũng xin gửi lời cảm ơn chân thành đến nhà trường đã tạo điều kiện thuận lợi về cơ sở vật chất, môi trường học tập và nghiên cứu, giúp chúng em có thể hoàn thành đề tài một cách tốt nhất.

Đặc biệt, chúng em xin trân trọng cảm ơn thầy ThS. Lê Minh Tân – người thầy đã tận tâm hướng dẫn, hỗ trợ chúng em trong suốt quá trình thực hiện đề tài. Những lời khuyên, góp ý sâu sắc cùng kinh nghiệm quý báu của thầy đã giúp chúng em định hướng đúng đắn, khắc phục khó khăn và hoàn thành đề tài đúng thời hạn.

Do thời gian và kiến thức còn hạn chế, đề tài không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý từ quý thầy cô để đề tài được hoàn thiện hơn trong tương lai.

Cuối cùng, chúng em kính chúc quý thầy cô luôn dồi dào sức khỏe, hạnh phúc và tiếp tục thành công trên con đường giảng dạy, ươm mầm tri thức.

Chúng em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, ngày 06 tháng 06 năm 2025

Nhóm Sinh viên thực hiện

Nguyễn Thị Phú

Bùi Quang Thiện

Trường ĐH Sư Phạm Kỹ Thuật TP.HCM

Khoa : CNTT

ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP

Họ và Tên SV thực hiện 1 : Nguyễn Thị Phú

Mã Số SV : 21110600

Ho và Tên SV thực hiện 2 : Bùi Quang Thiên

Mã Số SV : 21110656

Chuyên ngành : Trí Tuệ Nhân Tao

Tên luận văn : Hệ thống Thủ thư thông minh hỗ trợ sinh viên trường Đại Học Sư Phạm
Kỹ Thuật TP.HCM

GV hướng dẫn : ThS. Lê Minh Tân

Nhiệm Vụ Của Luận Văn :

1. Nghiên cứu các cơ sở lý thuyết về bài toán nhận diện khuôn mặt qua xử lý ảnh và học sâu, trích xuất thông tin qua nhận dạng ký tự quang học, gợi ý sách qua phương pháp học tăng cường và mô hình ngôn ngữ lớn qua phương pháp Retrieval Augmented Generation.
 2. Xây dựng cơ sở dữ liệu: thu thập xử lý các dữ liệu về khuôn mặt, về sách từ đó xây dựng cơ sở dữ liệu hoàn chỉnh giúp quản lý thông tin của hệ thống.
 3. Xây dựng hệ thống nhận diện khuôn mặt: Huấn luyện mô hình nhận diện khuôn mặt, đồng thời đưa ra giải pháp thay thế là quét mã QR nếu người dùng không thể nhận diện bằng khuôn mặt.
 4. Xây dựng hệ thống chatbot: Áp dụng phương pháp Retrieval Augmented Generation cho dữ liệu sách từ trang thư viện số trường Đại học Sư Phạm Kỹ Thuật TP.HCM.
 5. Xây dựng hệ thống quét bìa sách và lời mở đầu: Áp dụng kết hợp giữa 2 mô hình nhận dạng ký tự quang với mô hình ngôn ngữ lớn để trích xuất được tác giả và tiêu đề sách, từ đó thêm vào CSDL sách và vector database để chatbot truy vấn.
 6. Thiết kế hệ thống gợi ý sách: Áp dụng kỹ thuật Deep Q-Learning trong học tăng cường kết hợp để xuất phần thưởng để gợi ý sách cho người dùng.

- Xây dựng kiểm thử và tối ưu: Xây dựng kiểm thử với dữ liệu thực tế tối ưu hiệu năng của các hệ thống con.

Đề cương viết luận văn:

MỤC LỤC

PHẦN 1:MỞ ĐẦU

- TÍNH CẤP THIẾT CỦA ĐỀ TÀI
- ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU
- PHÂN TÍCH HƯỚNG NGHIÊN CỨU LIÊN QUAN
- KẾT QUẢ DỰ KIẾN

PHẦN 2:NỘI DUNG

CHƯƠNG 1: LÝ THUYẾT LIÊN QUAN

- MÔ HÌNH MULTI-TASK CASCADED CONVOLUTIONAL
 - Ý tưởng, hướng tiếp cận
 - Vai trò trong đề tài
 - Kiến trúc mô hình
 - Công thức
 - Siêu tham số
 - Mô phỏng
 - Kết quả thử nghiệm đã có
- MÔ HÌNH INCEPTIONRESNETV1
 - Ý tưởng, hướng tiếp cận
 - Cách hoạt động
 - Công thức
 - Siêu tham số
 - Mô phỏng
 - Kết quả thử nghiệm đã có
- KỸ THUẬT RAG (RETRIEVAL-AUGMENTED GENERATION)
 - Ý tưởng, hướng tiếp cận
 - Quy trình hoạt động của RAG
 - Các siêu tham số sử dụng trong RAG
 - Mô phỏng

1.3.5. Vai trò trong đề tài

1.4. MÔ HÌNH SENTECETRANSFORMER

1.4.1. Ý tưởng, hướng tiếp cận

1.4.2. Cách hoạt động

1.4.3. Các mô hình dựa trên SentenceTransformer

1.4.4. Mô phỏng

1.4.5. Kết quả thử nghiệm đã có

1.4.6. Vai trò trong đề tài

1.4.7. Mức độ phổ biến

1.5. HỌC TĂNG CƯỜNG

1.5.1. Ý tưởng, hướng tiếp cận

1.5.2. Công thức

1.5.3. Mô phỏng

1.6. PHƯƠNG PHÁP Q-LEARNING

1.6.1. Ý tưởng, hướng tiếp cận

1.6.2. Công thức

1.6.3. Siêu tham số

1.6.4. Mô phỏng

1.6.5. Kết quả thử nghiệm đã có

1.7. PHƯƠNG PHÁP DEEP Q-LEARNING

1.7.1. Ý tưởng, hướng tiếp cận

1.7.2. Công thức

1.7.3. Siêu tham số

1.7.4. Mô phỏng

1.7.5. Kết quả thử nghiệm đã có

1.7.6. Vai trò trong đề tài

1.8. CÔNG CỤ LLAMA.CPP

1.8.1. Giới thiệu

1.8.2. Tính tương thích

1.8.3. Mức độ phổ biến

1.9. CÔNG CỤ DJANGO

1.9.1. Giới thiệu

1.9.2. Cách hoạt động của Django

1.9.3. Tính tương thích

1.9.4. Mức độ phổ biến

1.10. CÔNG CỤ FLASK

1.10.1. Giới thiệu

1.10.2. Các tính năng nổi bật của Flask

1.10.3. Tính tương thích

1.10.4. Mức độ phổ biến

1.11. CÔNG NGHỆ LANGCHAIN

1.11.1. Giới thiệu

1.11.2. Các module của Langchain

1.11.3. Mức độ phổ biến

1.12. SQL SERVER

1.12.1. Giới thiệu

1.12.2. Các tính năng của SQL Server

1.12.3. Tính tương thích

1.13. CHROMA DB

1.13.1. Giới thiệu về cơ sở dữ liệu vector (vector database)

1.13.2. Định nghĩa về ChromaDB

1.13.3. Cách hoạt động của Chroma

1.13.4. Mức độ phổ biến

1.14. CÔNG CỤ PYMUPDF

1.14.1. Giới thiệu

1.14.2. Tính tương thích

1.14.3. Mức độ phổ biến

1.15. CÔNG CỤ PADDLEOCR

1.15.1. Giới thiệu

1.15.2. Tính tương thích

1.15.3. Mức độ phổ biến

1.16. CÔNG CỤ VIETOCR

1.16.1. Giới thiệu

1.16.2. Tính tương thích

1.16.3. Mức độ phổ biến

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. XÁC ĐỊNH YÊU CẦU KỸ THUẬT

2.2. CÁC CHỨC NĂNG CHÍNH

2.2.1. Chức năng đăng nhập bằng cách nhận diện khuôn mặt

2.2.2. Chức năng gợi ý sách cho người dùng

2.2.3. Chức năng trích xuất lời mở đầu sách

2.2.4. Chức năng dùng chatbot hỏi câu hỏi liên quan đến sách

2.2.5. Chức năng thêm sách bằng cách quét mã ISBN

2.2.6. Chức năng thêm sách bằng cách sử dụng công nghệ OCR

2.3. THIẾT KẾ HỆ THỐNG

2.3.1. Sơ đồ tổng quan

2.3.2. Sơ đồ chức năng đăng nhập bằng khuôn mặt

2.3.3. Sơ đồ chức năng dùng chatbot hỏi câu hỏi liên quan tới sách

2.3.4. Sơ đồ chức năng trích xuất lời mở đầu sách

2.3.5. Sơ đồ chi tiết chức năng gợi ý sách

2.3.6. Sơ đồ chức năng thêm sách bằng cách sử dụng công nghệ OCR

2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU

2.5. THIẾT KẾ GIAO DIỆN HỆ THỐNG

2.5.1. Trang đăng nhập

2.5.2. Trang chủ sách

2.5.3. Trang quản lý sách

2.5.4. Trang thêm sách bằng cách thủ công

2.5.5. Trang thêm sách bằng OCR

2.5.6. Trang chatbot tư vấn sách

2.5.7. Trang gửi yêu cầu xóa sách

2.5.8. Trang quản lý của thủ thư

2.5.9. Trang thêm sách bằng mã ISBN của thủ thư

2.5.10. Trang thêm sách bằng OCR của thủ thư

2.5.11. Trang quản lý sách của thủ thư

2.5.12. Trang xử lý yêu cầu của thủ thư

CHƯƠNG 3: HUẤN LUYỆN VÀ TINH CHỈNH HỆ THỐNG

3.1. DỮ LIỆU SỬ DỤNG

3.1.1. Tập dữ liệu các khuôn mặt

3.1.2. Tập dữ liệu sách

3.2. LÍ DO CHỌN VÀ ĐẶC ĐIỂM CỦA DỮ LIỆU

3.2.1. Tập dữ liệu khuôn mặt

3.2.2. Tập dữ liệu sách

3.3. TINH CHỈNH THAM SỐ

3.3.1. Tham số mô hình LLM Gemma 2.9b IT GGUF

3.3.2. Tham số mô hình embedding sentence-transformers/paraphrase-multilingual-mpnet-base-v2

3.3.3. Tham số mô hình cho hệ thống gợi ý sách

CHƯƠNG 4: ĐÁNH GIÁ HỆ THỐNG

4.1. ĐÁNH GIÁ HỆ THỐNG NHẬN DIỆN KHÔN MẶT

4.1.1. Mục tiêu đánh giá

4.1.2. Thiết kế hệ thống đánh giá

4.1.3. Kết quả

4.2. ĐÁNH GIÁ HỆ THỐNG CHATBOT TƯ VẤN SÁCH

4.2.1. Mục tiêu đánh giá

4.2.2. Thiết kế hệ thống đánh giá

4.2.3. Kết quả đánh giá

4.3. ĐÁNH GIÁ HỆ THỐNG OCR

4.3.1. Mục tiêu đánh giá

4.3.2. Thiết kế hệ thống đánh giá

4.3.3. Kết quả

4.4. ĐÁNH GIÁ HỆ THỐNG GỢI Ý SÁCH

4.4.1. Mục tiêu đánh giá

4.4.2. Phương pháp đánh giá

4.4.3. Kết quả

PHẦN 3: KẾT LUẬN

3.1. KẾT QUẢ ĐẠT ĐƯỢC

3.2. ƯU ĐIỂM

3.3. HẠN CHẾ

3.4. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

TÀI LIỆU THAM KHẢO

- [1] VTV.VN, “Ngày Sách và Văn hóa đọc Việt Nam 2024: Những tín hiệu tích cực từ cộng đồng,” 16 tháng 4 năm 2024. [Trực tuyến]. Có tại: <https://vtv.vn/doi-song/ngay-sach-va-van-hoa-doc-viet-nam-2024-nhung-tin-hieu-tich-cuc-tu-cong-dong-20240416073329563.htm>.
- [2] Châu Anh, “Nhiều thư viện tại Việt Nam như xác sống,” VnExpress, ngày 3 tháng 6 năm 2025. [Trực tuyến]. Có tại: <https://vnexpress.net/nhieu-thu-vien-tai-viet-nam-nhu-xac-song-4893373.html>.
- [3] Phan Trường Nhất, “Tác động của sự thay đổi, ứng dụng công nghệ thông tin và công nghệ mới trong dịch vụ thông tin – thư viện tại các trường đại học ở Việt Nam hiện nay,” Tạp chí Khoa học - Đại học Đồng Nai, số 27, 2023.
- [4] Y. Liu, “AI-powered Library Assistant Xiaotu at Tsinghua University”, in Proceedings of the International Symposium on Library Automation, Beijing, China, Aug. 2021, pp. 55–60.
- [5] Thư viện Quốc gia Việt Nam, “Ứng dụng A.I trong hoạt động phân loại tài liệu của thư viện” 2023. [Trực tuyến]. Có tại: <https://nvl.gov.vn/nghien-cuu-trao-doi/ung-dung-a.i-trong-hoat-dong-phan-loai-tai-lieu-cua-thu-vien.html>.
- [6] VinUni, “Nơi giáo dục và công nghệ gặp gỡ: Khám phá sách giáo khoa điện tử đầu tiên tích hợp AI của VinUni”, [Trực tuyến]. Có tại: <https://admissions.vinuni.edu.vn/vi/noi-giao-duc-va-cong-nghe-gap-go-kham-pha-sach-giao-khoa-dien-tu-dau-tien-tich-hop-ai-cua-vinuni/>.
- [7] J. D. Shank, “University of Arizona Library Integrates ChatGPT to Support Research,” in *Proceedings of the 2023 ACRL Conference*, Pittsburgh, PA, Mar. 15–18, 2023, pp. 210–215.
- [8] Y. Du, W. Qian, Y. Luo, et al., “PP-OCR: A Practical Ultra Lightweight OCR System”, *arXiv preprint arXiv:2009.09941*, 2020. [Trực tuyến]. Có tại: <https://arxiv.org/abs/2009.09941>.
- [9] K. Zhang, Z. Zhang, Z. Li, và Y. Qiao, “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3670–3679.
- [10] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”, in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec. 2020.
- [11] B. K. Iwana và S. Uchida, “BookCoverNet: A Dataset and Model for Book Cover Classification,” in *Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, Australia, Sep. 2019, pp. 144–149.
- [12] Z. Zhang và J. Liu, “Intelligent Library System Using RFID and AI-Based Search”, *IEEE Access*, vol. 9, pp. 144321–144329, 2021.

- [13] F. Schroff, D. Kalenichenko và J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.
- [14] C. Szegedy, S. Ioffe và V. Vanhoucke, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning” *arXiv preprint arXiv:1602.07261*, 2016. [Trực tuyến]. Có tại: <https://arxiv.org/abs/1602.07261>.
- [15] N. Reimers và I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, Nov. 2019, pp. 3982–3992.
- [16] C. J. C. H. Watkins, “Learning from Delayed Rewards” Luận án Tiến sĩ, Đại học Cambridge, Vương quốc Anh, 1989.
- [17] M. J. Kim và H. Peng, “Learning Time Reduction Using Warm Start Methods for a Reinforcement Learning Based Supervisory Control in Hybrid Electric Vehicle Applications” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 198–205, Jan. 2018.
- [18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg và D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015.
- [19] G. Gerganov, “llama.cpp: Port of LLaMA model in C/C++,” GitHub, 2023. [Trực tuyến]. Có tại: <https://github.com/ggerganov/llama.cpp>.
- [20] Phan Văn Tân, “Tìm hiểu về Django – Framework hỗ trợ Python trong lập trình Web” Viblo, 25/8/2022. [Trực tuyến]. Có tại: <https://viblo.asia/p/tim-hieu-ve-django-framework-ho-tro-python-trong-lap-trinh-web-QpmlexbkZrd>.
- [21] Nguyễn Hữu Dũng, “Flask python là gì, tính năng cơ bản và lý do vì sao nên sử dụng” Bizfly.vn, 1/4/2021. [Trực tuyến]. Có tại: <https://bizfly.vn/techblog/flask-python-la-gi.html>.
- [22] 200Lab Blog, “Tìm hiểu LangChain: Framework phát triển ứng dụng LLM mạnh mẽ,” 10/2/2025. [Trực tuyến]. Có tại: <https://200lab.io/blog/langchain-la-gi>.
- [23] Pum, “SQL Server là gì? Cách tải & cài đặt Microsoft SQL Server” 200Lab Blog, 16/9/2023. [Trực tuyến]. Có tại: <https://200lab.io/blog/sql-server-la-gi>.
- [24] A. A. Awan, “Learn How to Use Chroma DB: A Step-by-Step Guide” DataCamp, 28/9/2023. [Trực tuyến]. Có tại: <https://www.datacamp.com/tutorial/chromadb-tutorial-step-by-step-guide>.
- [25] Document Processing, “Thư viện Python nguồn mở để quản lý siêu dữ liệu PDF” 2025. [Trực tuyến]. Có tại: <https://products.documentprocessing.com/vi/metadata/python/pymupdf/>

[26] P. P., “What Is Optical Character Recognition (OCR)? Explained,” *Roboflow Blog*, ngày 21 tháng 11 năm 2023. [Trực tuyến]. Có tại: <https://blog.roboflow.com/what-is-optical-character-recognition-ocr/>.

[27] P. B. C. Quốc, “VietOCR: Transformer OCR,” GitHub, 2020. [Trực tuyến]. Có tại: <https://github.com/pbcquoc/vietocr>.

[28] Hugging Face. *sentence-transformers/paraphrase-multilingual-mpnet-base-v2*. [Trực tuyến]. Có tại: <https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2>.

[29] H. Burak, “Pins Face Recognition: Facial Recognition Dataset collected from Pinterest” Kaggle, 2018. [Trực tuyến]. Có tại: <https://www.kaggle.com/datasets/herveisburak/pins-face-recognition>.

[30] Thư viện số Trường Đại học Sư phạm Kỹ thuật TP.HCM, “Trang chủ Thư viện số,” Trường Đại học Sư phạm Kỹ thuật TP.HCM, [Trực tuyến]. Có tại: <https://thuvienso.hcmute.edu.vn/>.

KẾ HOẠCH THỰC HIỆN

STT	Thời gian	Công việc	Ghi chú
1	06/01 – 15/01/2025	Tìm hiểu đề tài và tìm đọc các bài báo liên quan	
2	16/01 – 21/01/2025	Crawl dữ liệu sách từ trang thư viện số, thu thập dữ liệu khuôn mặt từ Kaggle	
3	22/01 – 28/01/2025	Xây dựng Cơ sở dữ liệu và thêm dữ liệu.	
4	29/01 – 20/02/2025	Xây dựng chức năng đăng nhập bằng khuôn mặt.	
5	21/02 – 10/03/2025	Tìm hiểu Paddle OCR và VietOCR để trích xuất thông tin bìa sách và lời mở đầu.	
6	11/03 – 20/03/2025	Nghiên cứu triển khai quy trình RAG với dữ liệu là lời mở đầu	
7	21/03 – 07/04/2025	Nghiên cứu sử dụng LLM trong việc phân loại tác giả, tiêu đề	
8	08/04 – 14/04/2025	Tinh chỉnh các tham số mô hình lớn	

9	15/04 – 02/05/2025	Xây dựng chức năng gợi ý sách bằng phương pháp Deep Q-Learning	
10	03/05 – 19/05/2025	Nghiên cứu xây dựng UI/UX tương tác theo thời gian thực	
11	20/05 – 27/05/2025	Đánh giá các chức năng	
12	28/05 – 06/06/2025	Kiểm thử và tối ưu	

Ngày 06 tháng 06 năm 2025

Ý kiến của giáo viên hướng dẫn

Người viết đề cương

(ký và ghi rõ họ tên)

Bùi Quang Thiện

MỤC LỤC

DANH MỤC HÌNH ẢNH	1
DANH MỤC BẢNG	3
DANH MỤC TỪ VIẾT TẮT	5
Phần 1:MỞ ĐẦU	6
1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI	6
1.2. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU	8
1.3. PHÂN TÍCH HƯỚNG NGHIÊN CỨU LIÊN QUAN	9
1.4. KẾT QUẢ DỰ KIẾN	11
Phần 2:NỘI DUNG	13
CHƯƠNG 1: LÝ THUYẾT LIÊN QUAN	13
1.1. MÔ HÌNH MULTI-TASK CASCADED CONVOLUTIONAL	13
1.1.1. Ý tưởng, hướng tiếp cận	13
1.1.2. Vai trò trong đề tài	13
1.1.3. Kiến trúc mô hình	13
1.1.4. Công thức	18
1.1.5. Siêu tham số	20
1.1.6. Mô phỏng	21
1.1.7. Kết quả thử nghiệm đã có	24
1.2. MÔ HÌNH INCEPTIONRESNETV1	26
1.2.1. Ý tưởng, hướng tiếp cận	26
1.2.2. Cách hoạt động	26
1.2.3. Công thức	37
1.2.4. Siêu tham số	38
1.2.5. Mô phỏng	39
1.2.6. Kết quả thử nghiệm đã có	42
1.3. THUẬT RAG (RETRIEVAL-AUGMENTED GENERATION)	KỸ 43

1.3.1. Ý tưởng, hướng tiếp cận	43
1.3.2. Quy trình hoạt động của RAG	45
1.3.3. Các siêu tham số sử dụng trong RAG	47
1.3.4. Mô phỏng	47
1.3.5. Vai trò trong đê tài	48
1.4.	MÔ HÌNH SENTECETRANSFORMER
1.4.1. Ý tưởng, hướng tiếp cận	48
1.4.2. Cách hoạt động	49
1.4.3. Các mô hình dựa trên SentenceTransformer	49
1.4.4. Mô phỏng	50
1.4.5. Kết quả thử nghiệm đã có	50
1.4.6. Vai trò trong đê tài	51
1.4.7. Mức độ phổ biến	51
1.5.	HỌC TĂNG CƯỜNG
1.5.1. Ý tưởng, hướng tiếp cận	51
1.5.2. Công thức	53
1.5.3. Mô phỏng	55
1.6.	PHƯƠNG PHÁP Q-LEARNING
1.6.1. Ý tưởng, hướng tiếp cận	57
1.6.2. Công thức	57
1.6.3. Siêu tham số	58
1.6.4. Mô phỏng	58
1.6.5. Kết quả thử nghiệm đã có	60
1.7.	PHƯƠNG PHÁP DEEP Q-LEARNING
1.7.1. Ý tưởng, hướng tiếp cận	60
1.7.2. Công thức	61
1.7.3. Siêu tham số	61
1.7.4. Mô phỏng	62
1.7.5. Kết quả thử nghiệm đã có	63
1.7.6. Vai trò trong đê tài	63

1.8.	CỤ LLAMA.CPP	CÔNG
		64
1.8.1.	Giới thiệu	64
1.8.2.	Tính tương thích	64
1.8.3.	Mức độ phổ biến	65
1.9.	CỤ DJANGO	CÔNG
		65
1.9.1.	Giới thiệu	65
1.9.2.	Cách hoạt động của Django	66
1.9.3.	Tính tương thích	66
1.9.4.	Mức độ phổ biến	67
1.10.	CỤ FLASK	CÔNG
		67
1.10.1.	Giới thiệu	67
1.10.2.	Các tính năng nổi bật của Flask	67
1.10.3.	Tính tương thích	68
1.10.4.	Mức độ phổ biến	68
1.11.	NGHỆ LANGCHAIN	CÔNG
		68
1.11.1.	Giới thiệu	68
1.11.2.	Các module của Langchain	69
1.11.3.	Mức độ phổ biến	70
1.12.	SERVER	SQL
		71
1.12.1.	Giới thiệu	71
1.12.2.	Các tính năng của SQL Server	71
1.12.3.	Tính tương thích	72
1.13.	A DB	CHROM
		73
1.13.1.	Giới thiệu về cơ sở dữ liệu vector (vector database)	73
1.13.2.	Định nghĩa về ChromaDB	73
1.13.3.	Cách hoạt động của Chroma	74
1.13.4.	Mức độ phổ biến	75
1.14.	CỤ PYMUPDF	CÔNG
		75

1.14.1. Giới thiệu	75
1.14.2. Tính tương thích	75
1.14.3. Mức độ phổ biến	76
1.15.	CÔNG
CU PADDLEOCR	76
1.15.1. Giới thiệu	76
1.15.2. Tính tương thích	76
1.15.3. Mức độ phổ biến	76
1.16.	CÔNG
CU VIETOCR	77
1.16.1. Giới thiệu	77
1.16.2. Tính tương thích	77
1.16.3. Mức độ phổ biến	77
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	78
2.1. XÁC ĐỊNH YÊU CẦU KỸ THUẬT	78
2.2. CÁC CHỨC NĂNG CHÍNH	78
2.2.1. Chức năng đăng nhập bằng cách nhận diện khuôn mặt	78
2.2.2. Chức năng gợi ý sách cho người dùng	82
2.2.3. Chức năng trích xuất lời mở đầu sách	85
2.2.4. Chức năng dùng chatbot hỏi câu hỏi liên quan đến sách	88
2.2.5. Chức năng thêm sách bằng cách quét mã ISBN	90
2.2.6. Chức năng thêm sách bằng cách sử dụng công nghệ OCR	92
2.3. THIẾT KẾ HỆ THỐNG	95
2.3.1. Sơ đồ tổng quan	95
2.3.2. Sơ đồ chức năng đăng nhập bằng khuôn mặt	95
2.3.3. Sơ đồ chức năng dùng chatbot hỏi câu hỏi liên quan tới sách	97
2.3.4. Sơ đồ chức năng trích xuất lời mở đầu sách	98
2.3.5. Sơ đồ chi tiết chức năng gợi ý sách	99
2.3.6. Sơ đồ chức năng thêm sách bằng cách sử dụng công nghệ OCR	101
2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU	102
2.5. THIẾT KẾ GIAO DIỆN HỆ THỐNG	107
2.5.1. Trang đăng nhập	107
2.5.2. Trang chủ sách	108

2.5.3. Trang quản lý sách	108
2.5.4. Trang thêm sách bằng cách thủ công	109
2.5.5. Trang thêm sách bằng OCR	109
2.5.6. Trang chatbot tư vấn sách	110
2.5.7. Trang gửi yêu cầu xóa sách	110
2.5.8. Trang quản lý của thủ thư	110
2.5.9. Trang thêm sách bằng mã ISBN của thủ thư	111
2.5.10. Trang thêm sách bằng OCR của thủ thư	111
2.5.11. Trang quản lý sách của thủ thư	112
2.5.12. Trang xử lý yêu cầu của thủ thư	112
CHƯƠNG 3: HUẤN LUYỆN VÀ TINH CHỈNH HỆ THỐNG	113
3.1. DỮ LIỆU SỬ DỤNG	113
3.1.1. Tập dữ liệu các khuôn mặt	113
3.1.2. Tập dữ liệu sách	113
3.2. LÍ DO CHỌN VÀ ĐẶC ĐIỂM CỦA DỮ LIỆU	114
3.2.1. Tập dữ liệu khuôn mặt	114
3.2.2. Tập dữ liệu sách	115
3.3. TINH CHỈNH THAM SỐ	116
3.3.1. Tham số mô hình LLM Gemma 2.9b IT GGUF	116
3.3.2. Tham số mô hình embedding sentence-transformers/paraphrase-multilingual-mpnet-base-v2	116
3.3.3. Tham số mô hình cho hệ thống gợi ý sách	117
CHƯƠNG 4: ĐÁNH GIÁ HỆ THỐNG	119
4.1. ĐÁNH GIÁ HỆ THỐNG NHẬN DIỆN KHÔN MẶT	119
4.1.1. Mục tiêu đánh giá	119
4.1.2. Thiết kế hệ thống đánh giá	119
4.1.3. Kết quả	120
4.2. ĐÁNH GIÁ HỆ THỐNG CHATBOT TƯ VẤN SÁCH	122
4.2.1. Mục tiêu đánh giá	122
4.2.2. Thiết kế hệ thống đánh giá	122
4.2.3. Kết quả đánh giá	125
4.3. ĐÁNH GIÁ HỆ THỐNG OCR	127

4.3.1. Mục tiêu đánh giá	127
4.3.2. Thiết kế hệ thống đánh giá	127
4.3.3. Kết quả	129
4.4. ĐÁNH GIÁ HỆ THỐNG GỢI Ý SÁCH	131
4.4.1. Mục tiêu đánh giá	131
4.4.2. Phương pháp đánh giá	131
4.4.3. Kết quả	131
PHẦN 3: KẾT LUẬN	134
3.1. KẾT QUẢ ĐẠT ĐƯỢC	134
3.2. ƯU ĐIỂM	135
3.3. HẠN CHẾ	135
3.4. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI	136
TÀI LIỆU THAM KHẢO	137

DANH MỤC HÌNH ẢNH

Hình 1: Kiến trúc mạng P-Net.....	15
Hình 2: Kiến trúc mô hình InceptionResNet V1.....	27
Hình 3: Kiến trúc Inception-ResNet-A.....	30
Hình 4: Kiến trúc Reduction-A.....	31
Hình 5: Kiến trúc Inception-ResNet-B.....	33
Hình 6: Kiến trúc Reduction-B.....	34
Hình 7: Kiến trúc Inception-ResNet-C.....	36
Hình 8: Quy trình hoạt động của RAG.....	45
Hình 9: Cơ chế hoạt động của Học Tăng Cường.....	52
Hình 10: Mô hình quá trình học tăng cường.....	56
Hình 11: Công cụ Django	65
Hình 12: Cách hoạt động của Django.....	66
Hình 13: Công cụ Flask	67
Hình 14: Công cụ LangChain	68
Hình 15: Các module của Langchain.....	69
Hình 16: SQL Server	71
Hình 17: ChromaDB.....	73
Hình 18: Cách hoạt động của Chroma.....	74
Hình 19: Kết quả lưu vào ChromaDB khi trích xuất lời mở đầu sách.....	88
Hình 20: Sơ đồ tổng quan hệ thống	95
Hình 21: Sơ đồ chức năng đăng nhập bằng khuôn mặt.....	96
Hình 22: Sơ đồ chức năng đăng nhập bằng khuôn mặt.....	97
Hình 23: Sơ đồ chức năng dùng chatbot hỏi câu hỏi liên quan sách.....	98
Hình 24: Sơ đồ chức năng trích xuất lời mở đầu sách.....	99
Hình 25: Cấu trúc mạng mô hình chức năng gợi ý sách.....	99
Hình 26: Sơ đồ chức năng gợi ý sách	100
Hình 27: Sơ đồ quy trình đưa mạng Deep Q Learning vào hệ thống.....	101
Hình 28: Sơ đồ chức năng thêm sách bằng công nghệ OCR	101
Hình 29: Cơ sở dữ liệu hệ thống	102
Hình 30: Giao diện trang đăng nhập	107
Hình 31: Giao diện trang đăng nhập bằng khuôn mặt	107
Hình 32: Giao diện trang chủ sách.....	108
Hình 33: Giao diện trang quản lý sách.....	108
Hình 34: Giao diện trang thêm sách bằng thủ công	109
Hình 35: Giao diện trang thêm sách bằng OCR.....	109
Hình 36: Giao diện trang chatbot tư vấn sách.....	110
Hình 37: Giao diện trang gửi yêu cầu xóa sách	110
Hình 38: Giao diện trang quản lý của thủ thư	110
Hình 39: Giao diện trang thêm sách bằng mã ISBN	111
Hình 40: Giao diện trang thêm sách bằng OCR	111
Hình 41: Giao diện trang quản lý sách của thủ thư	112
Hình 42: Giao diện trang xử lý yêu cầu của thủ thư	112
Hình 43: Kết quả hệ thống nhận diện khuôn mặt qua các Metrics	120
Hình 44: Thời gian trung bình hệ thống nhận diện khuôn mặt	120
Hình 45: Phân bố Cosine Similarity trong chức năng nhận diện khuôn mặt	121

Hình 46: File Excel kết quả đánh giá chatbot.....	127
Hình 47: Kết quả ROUGE Scores trong hệ thống OCR.....	129
Hình 48: Kết quả LCS trong hệ thống OCR.....	130
Hình 49: Kết quả về độ bao phủ từ vựng trong hệ thống OCR.....	130
Hình 50: Đồ thị Reward qua các Episode của hệ thống gợi ý sách.....	132

DANH MỤC BẢNG

Bảng 1: Kiến trúc mạng P-net.....	14
Bảng 2: Kiến trúc mạng R-net.....	15
Bảng 3: Kiến trúc mạng O-net.....	17
Bảng 4: Siêu tham số mô hình MTCNN.....	20
Bảng 5: Tọa độ tương đối landmark khuôn mặt từ O-Net.....	23
Bảng 6: Kết quả thử nghiệm mô hình MTCNN.....	24
Bảng 7: Đặc điểm các dataset trong thử nghiệm mô hình MTCNN.....	25
Bảng 8: Kiến trúc Stem Block.....	27
Bảng 9: Kiến trúc Inception-ResNet-A.....	28
Bảng 10: Kiến trúc Reduction-A.....	30
Bảng 11: Kiến trúc Inception-Resnet-B.....	31
Bảng 12: Kiến trúc Reduction-B	33
Bảng 13: Kiến trúc Inception-ResNet-C.....	35
Bảng 14: Kiến trúc Output Block.....	36
Bảng 15: Siêu tham số mô hình InceptionResNet V1.....	38
Bảng 16: Mô phỏng InceptionResNet V1 Khối Stem.....	39
Bảng 17: Các nhánh của Khối Stem	40
Bảng 18: Mô phỏng InceptionResNet V1 Khối Inception-ResNet-A.....	40
Bảng 19: Mô phỏng InceptionResNet V1 Khối Reduction-A	40
Bảng 20: Mô phỏng InceptionResNet V1 Khối Inception-ResNet-B.....	41
Bảng 21: Mô phỏng InceptionResNet V1 Khối Reduction-B.....	41
Bảng 22: Mô phỏng InceptionResNet V1 Khối Inception-ResNet-C.....	42
Bảng 23: Kết quả thử nghiệm với một mô hình 1 lần cắt trên tập validation	42
Bảng 24: Kết quả thử nghiệm với một mô hình 12 lần cắt trên tập validation	42
Bảng 25: Kết quả thử nghiệm với một mô hình 144 lần cắt trên tập validation	43
Bảng 26: Siêu tham số của RAG - Retrieval	47
Bảng 27: Siêu tham số của RAG - Generation	47
Bảng 28: Siêu tham số của Q-Learning	58
Bảng 29: Trạng thái agent khi thực hiện các chuỗi hành động	59
Bảng 30: Trạng thái agent khi thực hiện lặp lại thêm 1 vòng	60
Bảng 31: Siêu tham số Deep Q-Learning	61
Bảng 32: Kết quả thử nghiệm đã có của Deep Q-Learning	63
Bảng 33: Các yêu cầu kỹ thuật cho hệ thống	78
Bảng 34: Quy trình thực hiện chức năng đăng nhập bằng nhận diện khuôn mặt.....	79
Bảng 35: Quy trình thực hiện chức năng đăng nhập bằng nhận diện khuôn mặt – giai đoạn 2	80
Bảng 36: Mô tả các file trong chức năng đăng nhập bằng nhận diện khuôn mặt.....	80
Bảng 37: Quy trình thực hiện chức năng đăng nhập bằng nhận diện khuôn mặt – giai đoạn 3	81
Bảng 38: Các biến ghi nhận người dùng đăng nhập	81
Bảng 39: Chức năng gợi ý sách cho người dùng – Giai đoạn 1	82
Bảng 40: Chức năng gợi ý sách cho người dùng – Giai đoạn 2	83
Bảng 41: Chức năng gợi ý sách cho người dùng – Giai đoạn 3	84
Bảng 42: Chức năng gợi ý sách cho người dùng – Giai đoạn 4	84
Bảng 43: Các thành phần trong chức năng dùng chatbot hỏi câu hỏi liên quan sách	88

Bảng 44: Cấu trúc mạng mô hình của chức năng gợi ý sách	100
Bảng 45: Bảng User trong CSDL	102
Bảng 46: Bảng Book trong CSDL	103
Bảng 47: Bảng DeleteRequest trong CSDL	104
Bảng 48: Bảng Department trong CSDL	104
Bảng 49: Bảng BookIntro trong CSDL	105
Bảng 50: Bảng BookQA trong CSDL	105
Bảng 51: Bảng UserBookInteraction trong CSDL	106
Bảng 52: Bảng QRCodeLoginToken trong CSDL	106
Bảng 53: Bảng Chat trong CSDL	106
Bảng 54: Mô tả tập dữ liệu khuôn mặt	114
Bảng 55: Mô tả tập dữ liệu sách	115
Bảng 56: Tham số mô hình Gemma 2.9b IT GGUF	116
Bảng 57: Tham số mô hình embedding sentence-transformers/paraphrase-multilingual-mmpnet-base-v2	116
Bảng 58: Tham số mô hình cho hệ thống gợi ý sách	117
Bảng 59: Kết quả đánh giá Chatbot	125
Bảng 60: Bảng kết quả hệ thống gợi ý sách qua các episode	131

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Nghĩa tiếng Anh	Tạm dịch
AI	Artificial Intelligence	Trí tuệ Nhân tạo
RAG	Retrieval Augmented Generation	Tạo sinh tăng cường truy xuất
LLM(s)	Large Language Model(s)	Mô hình ngôn ngữ lớn
API	Application Programming Interface	Giao diện lập trình ứng dụng
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
GPU	Graphics Processing Unit	Bộ xử lý đồ họa
RL	Reinforcement Learning	Học tăng cường
OCR	Optical Character Recognition	Nhận dạng ký tự quang học
CRNN	Convolutional Recurrent Neural Network	Mạng nơ-ron hồi quy tích chập
MTCNN	Multi-task Cascaded Convolutional Networks	Mạng tích chập xếp tầng đa tác vụ
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
DPR	Dense Passage Retrieval	Truy xuất đoạn dày đặc
RFID	Radio Frequency Identification	Nhận dạng đối tượng bằng sóng vô tuyến
PDF	Portable Document Format	Định dạng tài liệu di động
NSM	Non-Maximum Supression	Úc chế không tối đa
MDP(s)	Markov Decision Process(es)	Quy trình Markov
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản trên web
UX/UI	User Experience / User Interface	Trải nghiệm người dùng / Giao diện người dùng
PINS	Personalities IN Social Media	Hình ảnh cá nhân trên mạng xã hội
bbox	Bounding Box	Hộp giới hạn

PHẦN 1: MỞ ĐẦU

1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Trong thời đại công nghệ số, tri thức đã trở thành một trong những yếu tố cốt lõi quyết định sự phát triển của mỗi cá nhân và xã hội. Tuy nhiên, tại Việt Nam, văn hóa đọc – nguồn gốc của tri thức – vẫn chưa được chú trọng đúng mức. Theo thống kê từ VTV, chỉ 21% người Việt đọc sách mỗi năm, với mức trung bình chỉ 1,4 đầu sách/người/năm [1]. Thực trạng này không chỉ phản ánh sự thiếu quan tâm đến việc đọc sách mà còn cho thấy các rào cản trong việc tiếp cận tri thức, đặc biệt là trong môi trường giáo dục đại học.

Trong bối cảnh chuyển đổi số, các mô hình thư viện truyền thống đang đối mặt với nhiều thách thức lớn. Sự phát triển mạnh mẽ của các nền tảng trực tuyến và sự thay đổi trong cách tiếp cận của hệ thống thông tin đã khiến vai trò của thư viện dần bị lu mờ. Những hạn chế trong việc tiếp cận và khai thác hiệu quả tài nguyên thư viện không chỉ làm giảm tính hấp dẫn của thư viện mà còn cản trở sinh viên trong việc học tập và nghiên cứu [2]. Điều này đặt ra yêu cầu cấp thiết về việc đổi mới, tích hợp công nghệ để đáp ứng tốt hơn nhu cầu tri thức trong môi trường giáo dục hiện đại.

Việc tích hợp các công nghệ AI không chỉ giúp giải quyết các vấn đề về khai thác tài nguyên mà còn nâng cao trải nghiệm người dùng trong thư viện. Các tính năng này cho phép sinh viên nhanh chóng truy cập thông tin sách, hiểu rõ giá trị của từng tài liệu đối với ngành học của mình, từ đó thúc đẩy khả năng tự học và nghiên cứu một cách hiệu quả.

Tại các trường đại học, sự thiếu hụt trong việc ứng dụng công nghệ vào hệ thống thư viện đã khiến nhiều sinh viên gặp khó khăn trong việc khai thác hiệu quả nguồn tài nguyên sẵn có. Mặc dù các thư viện đại học đã triển khai một số ứng dụng công nghệ mới, nhưng việc áp dụng chưa đồng bộ và còn nhiều thách thức, đặc biệt trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ [3]. Điều này không chỉ ảnh hưởng đến khả năng tự học, tự nghiên cứu mà còn làm giảm động lực tìm hiểu và phát triển tri thức. Trong bối cảnh toàn cầu hóa và sự cạnh tranh gay gắt giữa các hệ thống giáo dục, đổi mới cách tiếp cận tri thức trong môi trường học thuật trở thành một yêu cầu tất yếu.

Trong bối cảnh toàn cầu hóa và sự cạnh tranh ngày càng gay gắt giữa các hệ thống giáo dục, đổi mới cách tiếp cận tri thức trong môi trường học thuật là một yêu cầu tất yếu. Việc xây dựng các hệ thống thư viện thông minh, áp dụng trí tuệ nhân tạo và các công nghệ tiên tiến nhằm cá nhân hóa việc tra cứu, hỗ trợ học thuật và nâng cao trải nghiệm người dùng đã trở thành xu hướng tại nhiều quốc gia phát triển.

Đề tài "Hệ thống thủ thư thông minh hỗ trợ sinh viên trường Đại Học Sư phạm Kỹ thuật TP.HCM" mang tính cấp thiết không chỉ bởi thực trạng nêu trên, mà còn bởi ý nghĩa chiến lược của việc ứng dụng công nghệ để nâng cao hiệu quả giáo dục. Một hệ thống thư viện thông minh sẽ không chỉ khắc phục những bất cập hiện tại mà còn mở ra cơ hội để thư viện trở thành trung tâm tri thức thực sự, nơi nuôi dưỡng văn hóa đọc và khuyến khích sinh viên chủ động tiếp cận tri thức. Đây là bước tiến quan trọng trong việc xây dựng một nền giáo dục hiện đại, gắn kết tri thức với công nghệ, góp phần đào tạo nguồn nhân lực chất lượng cao đáp ứng yêu cầu của thời đại.

Nhóm đã tiến hành khảo sát một số hệ thống thư viện tại các trường đại học trong và ngoài nước, tập trung vào mức độ ứng dụng công nghệ trí tuệ nhân tạo (AI), đặc biệt là khả năng tích hợp các mô hình ngôn ngữ lớn (LLM) vào hệ thống tra cứu. Qua khảo sát, nhóm nhận thấy hiện nay chưa có hệ thống thư viện nào tích hợp đồng thời các chức năng như trích xuất nội dung sách bằng OCR, hỗ trợ trả lời câu hỏi qua giao diện chat, và gợi ý sách cá nhân hóa dựa trên nội dung người dùng quan tâm. Tuy nhiên, một số hệ thống thư viện tiên tiến đã áp dụng AI vào một số khía cạnh như phân loại tài liệu, hỗ trợ học tập và cung cấp dịch vụ tham khảo thông minh. Cụ thể:

- Đại học Thanh Hoa (Tsinghua University), Trung Quốc: Trường đã phát triển hệ thống thư viện AI mang tên *Xiaotu*, cho phép người dùng tương tác qua ứng dụng di động hoặc mạng xã hội. *Xiaotu* cung cấp dịch vụ tham khảo ảo theo thời gian thực, có khả năng tự học từ phản hồi người dùng và liên kết với các nguồn tri thức như Wikipedia (phiên bản tiếng Trung), tài nguyên do giáo sư kiểm duyệt, FAQ từ thư viện, và internet nội địa. Hệ thống này cũng tạo điều kiện cho sinh viên tham gia nhóm đọc sách và truy cập thông tin thuận tiện [4].
- Thư viện Quốc gia Việt Nam (NLV): Tại Việt Nam, AI được ứng dụng vào việc phân loại tài liệu. AI hỗ trợ tự động nhận diện tiêu đề và nội dung chính, từ đó phân loại tài liệu một cách chính xác và nhanh chóng, giúp tiết kiệm thời gian xử lý thủ công và nâng cao hiệu quả quản lý tài nguyên [5].

- Đại học VinUni, Việt Nam: VinUni đã cho ra mắt sách giáo khoa điện tử đầu tiên tích hợp AI mang tên “*Đại số tuyến tính và ứng dụng*”. Cuốn sách tích hợp trợ lý ảo AI NaviAI hỗ trợ tương tác, tìm kiếm thông tin và tạo trải nghiệm học tập cá nhân hóa. AI có thể tạo câu hỏi luyện tập và flashcard, đồng thời hướng đến việc mở rộng khả năng truy cập dữ liệu trực tuyến trong tương lai [6].
- Đại học Arizona (University of Arizona), Hoa Kỳ: Thư viện của trường đã triển khai tích hợp ChatGPT để hỗ trợ sinh viên trong quá trình nghiên cứu. ChatGPT có thể giúp xác định chủ đề, đề xuất từ khóa, hỗ trợ tìm kiếm tài liệu và nâng cao hiệu quả học tập [7].

1.2. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu của đề tài là hệ thống thư viện thông minh, được phát triển nhằm hỗ trợ người dùng – đặc biệt là sinh viên – trong việc tra cứu, tìm kiếm và khai thác thông tin sách một cách hiệu quả, tự động và thuận tiện. Hệ thống này ứng dụng các công nghệ hiện đại như nhận diện ký tự quang học (OCR), nhận diện mã ISBN, và đặc biệt là mô hình ngôn ngữ lớn (LLM) để nâng cao khả năng xử lý và tương tác với dữ liệu thư viện.

Cụ thể, đề tài tập trung vào các hướng nghiên cứu và ứng dụng sau:

- Nhận diện tiêu đề và tác giả từ ảnh bìa sách thông qua OCR kết hợp LLM, cho phép tự động thêm sách vào CSDL.
- Nhận diện mã ISBN của sách sử dụng kỹ thuật xử lý ảnh, giúp truy xuất thông tin sách từ các cơ sở dữ liệu bên ngoài (như Google Books API), hỗ trợ quá trình nhập liệu nhanh chóng và chính xác.
- Xây dựng hệ thống nhận diện khuôn mặt giúp người dùng thuận tiện hơn trong việc đăng nhập.
- Trích xuất phần giới thiệu sách từ tài liệu PDF bằng OCR, xử lý văn bản, phân đoạn nội dung, và lưu vào cơ sở dữ liệu vector (ChromaDB) nhằm phục vụ truy vấn ngữ nghĩa bằng LLM.
- Ứng dụng mô hình LLM để trả lời câu hỏi về sách, cho phép người dùng đặt câu hỏi tự nhiên (vd: “Sách A viết về chủ đề gì?”, “Hãy gợi ý sách về lập trình”) và nhận được phản hồi chính xác dựa trên phần giới thiệu sách đã được trích xuất.
- Hiển thị sách gợi ý cho người dùng, dựa trên nội dung câu hỏi và lịch sử truy vấn trước đó. Mô hình sẽ phân tích ngữ nghĩa các tương tác, từ đó đề xuất những cuốn

sách có liên quan về nội dung hoặc chủ đề, giúp người dùng khám phá thêm nhiều tài liệu hữu ích.

- Phát triển hệ thống quản lý thư viện, cho phép thủ thư nhập liệu, chỉnh sửa thông tin sách, và tự động gán thẻ phân loại dựa trên nội dung sách hoặc từ khóa xuất hiện trong metadata.
- Xây dựng giao diện web thân thiện với người dùng, hỗ trợ các chức năng như tìm kiếm sách theo tên, tác giả, từ khóa hoặc bằng truy vấn ngôn ngữ tự nhiên. Giao diện cũng tích hợp tính năng xác thực qua khuôn mặt để đảm bảo bảo mật khi người dùng đăng nhập hệ thống.

1.3. PHÂN TÍCH HƯỚNG NGHIÊN CỨU LIÊN QUAN

Trong quá trình nghiên cứu và xây dựng hệ thống thư viện thông minh hỗ trợ thêm sách bằng ảnh bìa và tra cứu nội dung bằng ngôn ngữ tự nhiên, đề tài có sự kế thừa và tham khảo từ nhiều công trình nghiên cứu có giá trị trong các lĩnh vực như nhận dạng văn bản (OCR), xử lý ngôn ngữ tự nhiên (NLP), và ứng dụng trí tuệ nhân tạo trong quản lý thư viện. Dưới đây là một số công trình liên quan có ảnh hưởng trực tiếp đến định hướng và giải pháp kỹ thuật của đề tài.

Bài báo đầu tiên là “PP-OCR: A Practical Ultra Lightweight OCR System” của Du et al., công bố trên arXiv vào năm 2020 [8]. Đây là một trong những công trình nền tảng quan trọng trong lĩnh vực nhận dạng văn bản từ ảnh, đặc biệt với các ứng dụng yêu cầu hiệu suất cao và tốc độ xử lý nhanh. PP-OCR được thiết kế với mục tiêu xây dựng một hệ thống OCR siêu nhẹ, có thể hoạt động trên các thiết bị với tài nguyên hạn chế. Hệ thống này gồm ba thành phần chính: phát hiện văn bản bằng mô hình Differentiable Binarization (DB), nhận dạng văn bản bằng mô hình CRNN (Convolutional Recurrent Neural Network), và cuối cùng là bước hậu xử lý dùng để kết nối lại các dòng văn bản rời rạc. Nhờ kỹ thuật knowledge distillation, mô hình được rút gọn mà vẫn giữ được độ chính xác cao. PP-OCR là nền tảng cho PaddleOCR – công cụ được sử dụng trong đề tài để nhận dạng tiêu đề và tên tác giả từ ảnh bìa sách. Việc lựa chọn PP-OCR là do khả năng cân bằng giữa hiệu quả xử lý và tốc độ, phù hợp với các yêu cầu thực tế khi người dùng tải ảnh lên hệ thống.

Một nghiên cứu quan trọng khác liên quan đến việc xây dựng hệ thống thư viện thông minh là mô hình Multi-task Cascaded Convolutional Networks (MTCNN) được giới thiệu trong bài báo “*Joint Face Detection and Alignment using Multi-task*

Cascaded Convolutional Networks” của Zhang et al. (CVPR 2016) [9]. Đây là mô hình nổi bật trong lĩnh vực thị giác máy tính, đặc biệt được sử dụng rộng rãi trong các ứng dụng nhận diện và xác định đặc điểm khuôn mặt. MTCNN kết hợp ba mạng neuron tích chồng theo dạng tầng (P-Net, R-Net và O-Net), hoạt động theo cơ chế tuần tự nhằm tăng độ chính xác trong việc phát hiện khuôn mặt và các điểm đặc trưng như mắt, mũi, miệng. Mô hình được huấn luyện theo hướng đa nhiệm (multi-task learning), cho phép đồng thời thực hiện hai tác vụ: phát hiện khuôn mặt và căn chỉnh các điểm đặc trưng, giúp cải thiện hiệu suất tổng thể. Trong đề tài này, MTCNN được tích hợp để phục vụ chức năng đăng nhập bằng khuôn mặt dành cho thủ thư và người dùng hệ thống, giúp tăng cường tính bảo mật và tiện lợi, đồng thời là một bước tiến hướng tới thư viện không tiếp xúc (contactless).

Trong lĩnh vực xử lý ngôn ngữ tự nhiên, nghiên cứu “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks” của Lewis et al., được công bố tại hội nghị NeurIPS năm 2020, là một công trình có ảnh hưởng lớn trong việc kết hợp truy xuất thông tin với mô hình sinh ngôn ngữ [10]. Mô hình RAG (Retrieval-Augmented Generation) mà bài báo đề xuất là sự kết hợp giữa hai thành phần: một retriever (dựa trên Dense Passage Retrieval - DPR) có nhiệm vụ tìm kiếm các đoạn văn liên quan trong một cơ sở tri thức lớn, và một generator (dựa trên BART) để sinh câu trả lời hoàn chỉnh dựa trên thông tin truy xuất được. Hướng tiếp cận này giúp mô hình trả lời chính xác hơn và tránh tình trạng "hallucination" – khi mô hình tự bịa nội dung không có trong dữ liệu gốc. Trong đề tài này, RAG được áp dụng cho chức năng tra cứu nội dung sách: dữ liệu phần giới thiệu sách được lưu trữ trong cơ sở dữ liệu vector (ChromaDB), và khi người dùng đặt câu hỏi, hệ thống sẽ truy xuất các đoạn văn phù hợp để cung cấp cho LLM sinh câu trả lời. Cách tiếp cận này giúp mô hình LLM hiểu rõ ngữ cảnh và cung cấp câu trả lời sát với nội dung sách đã lưu.

Về ứng dụng thực tiễn, bài báo “BookCoverNet: A Dataset and Model for Book Cover Classification” của Iwana và Uchida, trình bày tại hội nghị ICDAR 2019 [11], là một nghiên cứu tiêu biểu trong việc sử dụng ảnh bìa sách để phân loại và nhận dạng thông tin sách. Bài báo xây dựng một tập dữ liệu lớn gồm hàng chục nghìn ảnh bìa sách kèm theo nhãn thể loại, sau đó huấn luyện mô hình CNN để phân loại ảnh theo thể loại (ví dụ: khoa học, tiểu thuyết, lịch sử). Ngoài ra, nghiên cứu cũng chứng minh rằng từ ảnh bìa có thể trích xuất được các đặc trưng quan trọng liên quan đến nội dung

cuốn sách. Kết quả này mở ra hướng đi cho việc sử dụng ảnh bìa không chỉ để nhận diện tiêu đề, mà còn để hỗ trợ gợi ý sách hoặc phân loại tự động. Đề tài kế thừa ý tưởng này bằng cách triển khai hệ thống OCR từ ảnh bìa, trích xuất các thông tin quan trọng như tiêu đề, tác giả, và gán thẻ loại dựa trên dữ liệu từ thư viện.

Cuối cùng, nghiên cứu “Intelligent Library System Using RFID and AI-Based Search” của Zhang và Liu, công bố trên tạp chí IEEE Access năm 2021 [12], là một ví dụ điển hình về việc tích hợp công nghệ AI vào hệ thống quản lý thư viện truyền thống. Hệ thống trong nghiên cứu này sử dụng RFID để theo dõi vị trí và trạng thái sách, đồng thời tích hợp công cụ tìm kiếm sử dụng AI để gợi ý tài liệu dựa trên hành vi người dùng. Mặc dù không sử dụng OCR hay mô hình sinh ngôn ngữ, nghiên cứu này cung cấp một mô hình tổng thể về cách ứng dụng công nghệ thông minh trong môi trường thư viện. Từ đó, đề tài đã mở rộng thêm chức năng gợi ý sách dựa trên lịch sử tra cứu và nội dung đã hỏi trước đó của người dùng, góp phần nâng cao trải nghiệm tra cứu và hỗ trợ học tập.

Tổng hợp lại, các nghiên cứu kể trên đã cung cấp nền tảng vững chắc cả về lý thuyết lẫn ứng dụng thực tiễn cho việc xây dựng hệ thống thư viện thông minh trong đề tài. Việc kết hợp các mô hình OCR hiện đại, công nghệ NLP dựa trên LLM, cùng định hướng ứng dụng thực tế giúp đề tài vừa đảm bảo tính học thuật, vừa đáp ứng tốt các nhu cầu thực tiễn tại các thư viện đại học hiện nay.

1.4. KẾT QUẢ DỰ KIẾN

Phần mềm đầu ra của đề tài có tên gọi là “Hệ thống Thư viện Thông minh hỗ trợ nhập liệu và tra cứu sách bằng ảnh bìa và mô hình ngôn ngữ lớn (LLM)”. Đây là một hệ thống web tích hợp nhiều chức năng hỗ trợ quản lý, nhập liệu và tra cứu sách trong môi trường thư viện đại học. Hệ thống cho phép người dùng thêm sách mới thông qua ảnh chụp bìa bằng công nghệ OCR kết hợp với mô hình ngôn ngữ, đồng thời cung cấp chức năng hỏi – đáp thông minh về nội dung sách bằng giao diện hội thoại tự nhiên.

Các chức năng chính của phần mềm gồm có:

- Tự động điền thông tin sách khi tải file PDF vào biểu mẫu để giúp thủ thư nhập liệu nhanh chóng.
- Chức năng đăng nhập bằng cách nhận diện khuôn mặt.
- Chức năng tra cứu thông tin sách bằng ngôn ngữ tự nhiên, thông qua mô hình LLM kết hợp với dữ liệu phần mở đầu sách lưu trong cơ sở tri thức vector.

- Gợi ý sách liên quan dựa trên nội dung người dùng đã tương tác trước đó.
- Quản lý sách và người dùng thông qua hệ thống tài khoản và phân quyền (thủ thư, sinh viên).
- Hiển thị kết quả trích xuất, xác nhận thủ công, và lưu vào cơ sở dữ liệu nội bộ.

Đối tượng người dùng chính của hệ thống là thủ thư thư viện và sinh viên đại học. Thủ thư sẽ sử dụng hệ thống để nhập liệu sách mới một cách nhanh chóng, chính xác mà không cần gõ tay, trong khi sinh viên có thể sử dụng để tra cứu thông tin sách và đặt câu hỏi liên quan đến nội dung học thuật. Hệ thống cũng có thể mở rộng cho giảng viên hoặc cán bộ quản lý thư viện trong tương lai.

Ý nghĩa thực tiễn của đề tài thể hiện ở khả năng tăng tốc và tự động hóa quy trình quản lý sách trong môi trường học thuật. Đối với thủ thư, hệ thống giúp giảm thiểu thời gian và sai sót trong nhập liệu thủ công. Đối với sinh viên, hệ thống mang đến trải nghiệm tra cứu hiện đại, thông minh hơn so với các công cụ tìm kiếm truyền thống. Đặc biệt, trong bối cảnh khối lượng sách ngày càng lớn, giải pháp này góp phần số hóa thư viện hiệu quả và tiết kiệm chi phí nhân sự.

Hệ thống được thiết kế theo mô hình client-server, trong đó phần nhận diện ảnh và xử lý ngôn ngữ có thể chạy độc lập trên server hỗ trợ GPU (nếu cần thiết). Cụ thể:

- GPU: Không bắt buộc trong giai đoạn đầu triển khai. Các mô hình OCR như PadleOCR và VietOCR có thể chạy hiệu quả trên CPU. Tuy nhiên, nếu tích hợp mô hình LLM lớn (ví dụ như LLaMA-3 hoặc GPT qua API nội bộ), thì nên triển khai trên server Zeppelin có GPU (NVIDIA Tesla P100 PCIe 16GB và MSI RTX 3090 SUPRIM X 24GB).
- Cơ sở dữ liệu: Dữ liệu giới thiệu sách được lưu trong cơ sở dữ liệu vector (ChromaDB), với dung lượng trung bình khoảng vài MB cho 250 sách. Vì vậy, hệ thống không yêu cầu một cơ sở dữ liệu quá lớn; có thể vận hành tốt trên các hệ thống lưu trữ phổ thông.
- Quy mô triển khai: Trong giai đoạn đầu, hệ thống được triển khai nội bộ cho một thư viện đại học, quy mô khoảng 250 đầu sách và phục vụ đồng thời khoảng 50 – 100 người dùng (bao gồm thủ thư và sinh viên).

PHẦN 2: NỘI DUNG

CHƯƠNG 1: LÝ THUYẾT LIÊN QUAN

1.1. MÔ HÌNH MULTI-TASK CASCADED CONVOLUTIONAL

1.1.1. Ý tưởng, hướng tiếp cận

MTCNN (Multi-task Cascaded Convolutional Networks) là một mô hình deep learning được thiết kế để phát hiện khuôn mặt và các điểm đặc trưng trên khuôn mặt (landmark) trong ảnh. Mô hình này được giới thiệu trong bài báo "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks" (Zhang et al., 2016) [9]. Ý tưởng chính của MTCNN là sử dụng một mạng nơ-ron tích chập (CNN) theo kiểu cascade (tầng tầng lớp lớp) để giải quyết đồng thời hai nhiệm vụ: phát hiện khuôn mặt và xác định vị trí các điểm landmark - các vùng nổi bật của khuôn mặt như: mắt trái-phải, mũi, miệng trái-phải.

MTCNN hoạt động theo ba giai đoạn chính:

- P-Net (Proposal Network): Mạng này nhanh chóng tạo ra các ứng viên (candidate boxes) có thể chứa khuôn mặt bằng cách quét ảnh ở nhiều tỷ lệ khác nhau.
- R-Net (Refinement Network): Mạng này loại bỏ các hộp giả (false positives) và tinh chỉnh lại vị trí các hộp ứng viên từ P-Net.
- O-Net (Output Network): Mạng cuối cùng này tiếp tục tinh chỉnh vị trí khuôn mặt, đồng thời dự đoán vị trí các điểm landmark (như mắt, mũi, miệng).

MTCNN tận dụng phương pháp multi-task learning, trong đó mỗi giai đoạn không chỉ học để phát hiện khuôn mặt mà còn học để dự đoán các điểm landmark và hiệu chỉnh bounding box.

1.1.2. Vai trò trong đề tài

- Vai trò: Tìm ra bounding box của người dùng trong khi đăng nhập bằng khuôn mặt.
- Lý do chọn: MTCNN phù hợp cho đề tài nhận diện khuôn mặt real-time do cân bằng tốt giữa tốc độ và độ chính xác, đồng thời cung cấp sẵn landmarks cho các bước tiền xử lý (căn chỉnh ảnh).

1.1.3. Kiến trúc mô hình

MTCNN gồm ba mạng con: P-Net, R-Net, O-Net xử lý tuần tự để phát hiện khuôn mặt và landmark.

1.1.3.1. Mạng P-net

Mục đích của phương pháp này là tạo ra các ứng viên (candidate boxes) chứa khuôn mặt một cách nhanh chóng bằng cách quét ảnh đầu vào ở nhiều tỷ lệ khác nhau. Bằng việc áp dụng kỹ thuật này, hệ thống có thể phát hiện các khuôn mặt ở nhiều kích thước, từ lớn đến nhỏ, trong khi vẫn đảm bảo tốc độ xử lý cao. Điều này đặc biệt hữu ích trong các ứng dụng thời gian thực, nơi cần cân bằng giữa độ chính xác và hiệu suất.

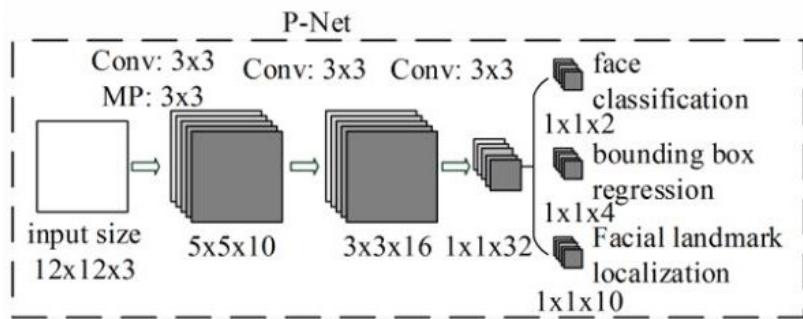
Kiến trúc mạng sử dụng một mạng CNN nông (shallow CNN) bao gồm các lớp tích chập, lớp MaxPooling và hàm kích hoạt PReLU. Mạng này được thiết kế đơn giản để tối ưu tốc độ xử lý, đồng thời vẫn đảm bảo khả năng phát hiện khuôn mặt hiệu quả. Các lớp Conv giúp trích xuất đặc trưng, MaxPooling giảm kích thước dữ liệu và PReLU đem lại tính phi tuyến, giúp mạng học tốt hơn mà không quá phức tạp.

Bảng 1: Kiến trúc mạng P-net

Layer	Kernel Size	Stride	Filter	Output Size	Input Size
Input	-	-	-	12x12x3	12x12x3
Conv1+PReLU	3x3	1	10	10x10x10	12x12x3
MaxPool1	2x2	2	-	5x5x10	10x10x10
Conv2+PReLU	3x3	1	16	3x3x16	5x5x10
Conv3+PReLU	3x3	1	32	1x1x32	3x3x16
Conv4 (cls)	1x1	1	2	1x1x2	1x1x32
Conv4 (bbox)	1x1	1	4	1x1x4	1x1x32
Conv4 (landmark)	1x1	1	10	1x1x10	1x1x32

- Đầu ra gồm 3 nhánh:

- Classification: Dự đoán xem vùng ảnh có phải khuôn mặt hay không (face/non-face).
- Bounding Box Regression: Hiệu chỉnh tọa độ bounding box.
- Landmark Localization (tùy chọn): Dự đoán 5 điểm landmark (mắt, mũi, miệng).



Hình 1: Kiến trúc mạng P-Net

- Kích thước đầu vào: Ảnh được chia thành nhiều cửa sổ con (sliding windows) với kích thước 12×12 pixel.
- Kích thước đầu ra:
 - Classification: $(1, 1, 2)$ (xác suất face/non-face).
 - Bounding Box Regression: $(1, 1, 4)$ (tọa độ $[x_1, y_1, x_2, y_2]$).
 - Landmark (nếu có): $(1, 1, 10)$ (5 điểm landmark, mỗi điểm có (x, y)).
- Xử lý sau P-Net:
 - Áp dụng Non-Maximum Suppression để loại bỏ các hộp trùng lặp. Non-Maximum Suppression (NMS) là bước quan trọng trong phát hiện khuôn mặt nhằm loại bỏ các hộp dự đoán trùng lặp, chỉ giữ lại hộp có độ tin cậy cao nhất cho mỗi khuôn mặt. Thuật toán hoạt động bằng cách chọn hộp có xác suất cao nhất, sau đó loại bỏ các hộp còn lại có độ chồng lấp (IoU) vượt ngưỡng cho phép. Nhờ đó, NMS giúp kết quả phát hiện rõ ràng, chính xác và không bị rối bởi các hộp dư thừa.
 - Chỉ giữ lại các hộp có độ tin cậy cao (threshold ~ 0.6).

1.1.3.2. Mạng R-net

- Mục đích: Lọc bỏ các hộp giả (false positives) và tinh chỉnh lại vị trí bounding box.
- Kiến trúc:
 - Mạng CNN sâu hơn P-Net, gồm các lớp Conv + MaxPooling + Fully Connected.
 - Tương tự P-Net, có 3 nhánh đầu ra (classification, bbox regression, landmark).

Bảng 2: Kiến trúc mạng R-net

Layer	Kernel Size	Stride	Filter	Output Size	Input Size
Input	-	-	-	24x24x3	24x24x3
Conv1+PReLU	3x3	1	28	22x22x28	24x24x3
MaxPool1	3x3	2	-	11x11x28	22x22x28
Conv2+PReLU	3x3	1	48	9x9x48	11x11x28
MaxPool2	3x3	2	-	4x4x48	9x9x48
Conv3+PReLU	2x2	1	64	3x3x64	4x4x48
Dense1	-	-	128	128	3x3x64(flatten)
Dense2(cls)	-	-	2	2	128
Dense2(bbox)	-	-	4	4	128
Dense2(landmark)	-	-	10	10	128

- Kích thước đầu vào: Các hộp ứng viên từ P-Net được resize về 24×24 pixel.
- Kích thước đầu ra:
 - o Classification: (1, 2) (face/non-face).
 - o Bounding Box Regression: (1, 4) (tọa độ sau hiệu chỉnh).
 - o Landmark (nếu có): (1, 10) (5 điểm landmark).
- Xử lý sau R-Net:
 - o Tiếp tục áp dụng NMS và lọc theo ngưỡng tin cậy (~ 0.7).

1.1.3.3. Mang O-net

- Mục đích:
 - o Tinh chỉnh chính xác bounding box.
 - o Dự đoán 5 điểm landmark.
- Kiến trúc:

- Mạng CNN sâu nhất trong MTCNN, gồm nhiều lớp Conv + FC.
- Có 3 nhánh đầu ra tương tự P-Net và R-Net.

Bảng 3: Kiến trúc mạng O-net

Layer	Kernel Size	Stride	Filter	Output Size	Input Size
Input	-	-	-	-	48x48x3
Conv1+PReLU	3x3	1	-	-	48x48x3
MaxPool1	3x3	2	-	-	46x46x32
Conv2+PReLU	3x3	1	-	-	23x23x32
MaxPool2	3x3	2	-	-	21x21x64
Conv3+PReLU	3x3	1	-	-	10x10x64
MaxPool3	2x2	2	-	-	8x8x64
Conv4+PReLU	2x2	1	-	-	4x4x64
Dense1	-	-	256	256	3x3x128 (flatten)
Dense2(cls)	-	-	2	2	256
Dense2(bbox)	-	-	4	4	256
Dense2(landmark)	-	-	10	10	256

- Kích thước đầu vào:
 - Các hộp từ R-Net được resize về 48×48 pixel.
- Kích thước đầu ra:
 - Classification: (1, 2) (face/non-face).
 - Bounding Box Regression: (1, 4) (tọa độ cuối cùng $[x_1, y_1, x_2, y_2]$).
 - Landmark: (1, 10) (5 điểm landmark $[x_1, y_1, x_2, y_2, \dots, x_5, y_5]$).
- Xử lý cuối cùng:

- Áp dụng NMS lần cuối với ngưỡng tin cậy cao (~0.7).
- Trả về danh sách các khuôn mặt đã được căn chỉnh và landmark (nếu có)

1.1.4. Công thức

Công thức chung:

$$L = \alpha_{cls} L_{cls} + \alpha_{box} L_{box} + \alpha_{landmark} L_{landmark}$$

Trong đó:

- L_{cls} : Cross-entropy cho classification (face/non-face).
- L_{box} : MSE cho bounding box regression.
- $L_{landmark}$: MSE cho tọa độ landmarks (5 điểm)
- $\alpha_{cls}, \alpha_{box}, \alpha_{landmark}$: Trọng số (ví dụ: 1, 0.5, 0.5)

Chức năng: Tổng loss của MTCNN là một hàm mất mát đa nhiệm. Nó đóng vai trò trung tâm trong việc huấn luyện mô hình để thực hiện đồng thời ba nhiệm vụ quan trọng trong nhận diện khuôn mặt: phân loại, dự đoán hộp giới hạn và dự đoán vị trí điểm đặc trưng của khuôn mặt.

1.1.4.1. Classification loss (Softmax loss)

Công thức:

$$\mathcal{L}_{cls} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Trong đó

- $y \in \{0,1\}$: nhãn thực tế (0 = non-face, 1 = face)
- \hat{y} : xác suất dự đoán là khuôn mặt, đầu ra từ mạng

Chức năng: Giúp mô hình học cách phân biệt giữa khuôn mặt và không phải khuôn mặt. Đây là bài toán phân loại nhị phân. Nếu mô hình dự đoán sai nhãn, hàm loss này sẽ tăng để phạt lỗi và điều chỉnh trọng số trong quá trình huấn luyện.

1.1.4.2. Bounding box regression loss (Smooth L1 Loss)

$$\text{SmoothL1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$$

Áp dụng cho 4 thành phần của bounding box:

$$\mathcal{L}_{\text{bbox}} = \sum_{i \in \{x_1, y_1, x_2, y_2\}} \text{SmoothL1}(t_i - \hat{t}_i)$$

Trong đó:

- t_i : Giá trị thật (ground truth) của tọa độ bounding box (tọa độ được chuẩn hóa)
- \hat{t}_i : Giá trị thật của mạng

Chức năng: Huấn luyện mô hình để dự đoán vị trí và kích thước chính xác của bounding box bao quanh khuôn mặt. Smooth L1 Loss giúp ổn định quá trình học bằng cách xử lý tốt cả lỗi nhỏ và lớn (so với L2 loss, vốn nhạy cảm với outliers).

1.1.4.3. Landmark regression loss

$$\mathcal{L}_{\text{landmark}} = \sum_{i=1}^5 [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

Trong đó:

- 5 landmark: 2 mắt, mũi, 2 miệng, tổng cộng 10 giá trị.
- x_i, y_i : Tọa độ thực của điểm landmark thứ i
- \hat{x}_i, \hat{y}_i : Tọa độ dự đoán từ O-Net

Chức năng: Huấn luyện mô hình để dự đoán chính xác các điểm đặc trưng trên khuôn mặt, từ đó phục vụ cho các tác vụ như căn chỉnh khuôn mặt.

1.1.4.4. Bounding box regression

Từ hộp dự đoán ban đầu $B = (x_1, y_1, x_2, y_2)$, và offset $(\Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2)$ do mạng dự đoán → cập nhật lại hộp.

$$\begin{aligned}\hat{x}_1 &= x_1 + \Delta x_1 \cdot w \\ \hat{y}_1 &= y_1 + \Delta y_1 \cdot h \\ \hat{x}_2 &= x_2 + \Delta x_2 \cdot w \\ \hat{y}_2 &= y_2 + \Delta y_2 \cdot h\end{aligned}$$

Trong đó:

- $w = x_2 - x_1, h = y_2 - y_1$: Chiều rộng và chiều cao hộp gốc.
- $\Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2$: output từ mạng (regression head).

Chức năng: Giúp tinh chỉnh lại bounding box ban đầu bằng các giá trị dịch chuyển ($\Delta x, \Delta y$) và tỉ lệ thay đổi kích thước ($\Delta w, \Delta h$), làm cho vùng dự đoán khớp hơn với vị trí thật sự của khuôn mặt.

1.1.4.5. Facial Landmark Regression

$$\hat{x}_i = x_1 + w \cdot \delta x_i, \quad \hat{y}_i = y_1 + h \cdot \delta y_i$$

for $i = 1 \dots 5$

Trong đó:

- (x_1, y_1) : góc trên trái của hộp mặt.
- w, h : chiều rộng/chiều cao của hộp.
- $(\delta x_i, \delta y_i)$: tọa độ landmark chuẩn hóa do O-Net dự đoán (giá trị từ 0 đến 1).

Chức năng: Dự đoán và tính toán chính xác vị trí các điểm đặc trưng trên khuôn mặt (mắt, mũi, miệng...) bằng cách chuyển tọa độ landmark từ dạng chuẩn hóa sang tọa độ thật dựa trên bounding box. Đây là bước quan trọng để căn chỉnh và phân tích khuôn mặt sau khi đã phát hiện.

1.1.4.6. Non-Maximum Suppression(NMS)

Công thức:

$$\text{IoU} = \frac{\text{Area}(B_1 \cap B_2)}{\text{Area}(B_1 \cup B_2)}$$

Nếu $\text{IoU} > \text{threshold}$ (thường 0.7 hoặc 0.6), loại bỏ hộp có score thấp hơn.

Chức năng: Loại bỏ các bounding box trùng lặp hoặc quá gần nhau, chỉ giữ lại vùng có xác suất chứa khuôn mặt cao nhất. NMS giúp đảm bảo mô hình không dự đoán nhiều khuôn mặt cho cùng một người.

1.1.5. Siêu tham số

Bảng 4: Siêu tham số mô hình MTCNN

Hyperparameter	P-Net	R-Net	O-Net
Learning Rate	$0.001 \rightarrow 0.0001$	$0.001 \rightarrow 0.0001$	$0.001 \rightarrow 0.0001$
Batch Size	128	64	16

Training Iterators	600K	400K	200K
Confidence Threshold	0.6	0.7	0.7
NMS Threshold	0.7	0.7	0.7
Minimum Face Size	20px	20px	20px

1.1.6. Mô phỏng

Mục tiêu: Phát hiện khuôn mặt trên một ảnh đen trắng kích thước 60×60 , chứa duy nhất 1 khuôn mặt hình vuông kích thước 20×20 tại vị trí ($x = 20, y = 20$).

Giả định:

- Mỗi stage sử dụng kernel 3×3 , stride = 1.
- Đầu vào cho P-Net là ảnh gốc 60×60 .
- Ngưỡng xác suất xác định khuôn mặt là 0.6 ở mỗi stage.
- Tại mỗi pixel, P-Net dự đoán:
 - o Xác suất có mặt: p
 - o Hộp đề xuất - Bounding box: $[\Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2]$ là phần hiệu chỉnh vị trí.

Bước 1: P-Net

- Đầu vào: Ảnh grayscale có kích thước 60×60 pixel.
- Output: Với mỗi vị trí trên ảnh (sau khi qua kernel 3×3 , stride=1), P-Net dự đoán:
 - o Xác suất có khuôn mặt p
 - o Bounding box Offset: $[\Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2]$ để hiệu chỉnh hộp đề xuất.
- Giả sử tại vị trí: ($i = 20, j = 20$), P-Net dự đoán:
 - o $p = 0.85$, vượt ngưỡng threshold mặc định 0.6 nên giữ lại.
 - o Offset được dự đoán:
 - $\Delta x_1 = -0.1, \Delta y_1 = -0.1$ (dịch mép trái, mép trên)
 - $\Delta x_2 = 0.1, \Delta y_2 = 0.1$ (dịch mép phải, mép dưới)
- Tính toán Bounding box ban đầu trước khi điều chỉnh:
 - o Vì P-Net quét với kernel 3×3 và stride = 1, nên mỗi vị trí (i, j) tương ứng với một vùng 20×20 trên ảnh gốc (theo thiết kế giả lập của đề bài).
 - o Với ($i = 20, j = 20$), vùng tương ứng trên ảnh gốc là:

- Original box: $(x_1, y_1, x_2, y_2) = (20, 20, 40, 40) \rightarrow$ Một hộp vuông 20x20
- Áp dụng Offset để hiệu chỉnh Bounding box
 - o Chiều cao và chiều rộng ban đầu:
 - width = $40 - 20 = 20$
 - height = $40 - 20 = 20$
 - o Điều chỉnh:
 - new_x₁ = $20 - 0.1 * 20 = 18$
 - new_y₁ = $20 - 0.1 * 20 = 18$
 - new_x₂ = $40 + 0.1 * 20 = 42$
 - new_y₂ = $40 + 0.1 * 20 = 42$
- Kết quả → Hộp mới: (18, 18, 42, 42)

Bước 2: R-Net

- Crop ảnh trong vùng (18,18) – (42,42), resize về 24x24.
- R-Net dự đoán p = 0.9, Bounding box Offset: [-0.05, -0.05, 0.05, 0.05] → Tiếp tục điều chỉnh hộp
- Điều chỉnh:
 - o dx₁ = $-0.05 * 24 = -1.2$
 - o x₁ = $18 - 1.2 \approx 16.8$
 - o dx₂ = $0.05 * 24 = 1.2$
 - o x₂ = $42 + 1.2 \approx 43.2$
- Kết quả → Hộp mới: (16.8, 16.8, 43.2, 43.2)

Bước 3: O-Net

- Đầu vào: Sử dụng vùng đã được điều chỉnh từ bước 2 phía trước, cụ thể là một Bounding box: ($x_1 = 16.8, y_1 = 16.8$) đến ($x_2 = 43.2, y_2 = 43.2$)
- Thao tác: Crop vùng ảnh này từ ảnh gốc kích thước 60×60, sau đó resize về kích thước 48×48 để đưa vào O-Net.
- O-Net dự đoán:
 - o Xác suất có mặt: p = 0.95, vượt ngưỡng threshold mặc định 0.6 nên giữ lại.
 - o Bounding box Offset mới để điều chỉnh lại hộp.
 - o Landmark: O-Net không trả về tọa độ tuyệt đối mà trả về tọa độ tương đối chuẩn hóa trong hộp vào

Bảng 5: Tọa độ tương đối landmark khuôn mặt từ O-Net

Landmark	x' (tương đối)	y' (tương đối)
Mắt trái	0.31	0.31
Mắt phải	0.69	0.31
Mũi	0.50	0.50
Miệng trái	0.35	0.69
Miệng phải	0.65	0.69

- Chuyển tọa độ landmark về ảnh gốc bằng công thức

$$\begin{aligned} width &= x_2 - x_1 \\ height &= y_2 - y_1 \end{aligned}$$

$$\begin{aligned} landmark_x &= x_1 + x' * width \\ landmark_y &= y_1 + y' * height \end{aligned}$$

- Ta được: $width = x_2 - x_1 = 26.4$, $height = y_2 - y_1 = 26.4$
- Các điểm landmark:
 - Mắt trái: $x = 16.8 + 0.31 \times 26.4 = 25$, $y = 16.8 + 0.31 \times 26.4 = 25 \rightarrow (25, 25)$
 - Mắt phải: $x = 16.8 + 0.69 \times 26.4 = 35$, $y = 16.8 + 0.31 \times 26.4 = 25 \rightarrow (35, 25)$
 - Mũi: $x = 16.8 + 0.50 \times 26.4 = 30$, $y = 16.8 + 0.50 \times 26.4 = 30 \rightarrow (30, 30)$
 - Miệng trái: $x = 16.8 + 0.35 \times 26.4 = 26$, $y = 16.8 + 0.69 \times 26.4 = 35 \rightarrow (26, 35)$

- Miệng phải: $x = 16.8 + 0.65 \times 26.4 = 34$, $y = 16.8 + 0.69 \times 26.4 = 35$
 $\rightarrow (34, 35)$

1.1.7. Kết quả thử nghiệm đã có

Trong bài báo gốc “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks” (Zhang et al., 2016) [9] mô hình đạt được các kết quả đáng kinh ngạc như sau:

Bảng 6: Kết quả thử nghiệm mô hình MTCNN

Phương diện đánh giá	Dataset	Chỉ số đo lường	Kết quả	Điều kiện thử nghiệm	Ghi chú
Phát hiện khuôn mặt	FDDB	True Positive Rate (TPR)	92.1%	1000 false positives	Đạt recall cao nhất trong các phương pháp cùng thời
Phát hiện khuôn mặt	WIDER FACE	Average Preci- sion (AP)	Easy: 94.8% Medium: 93.4% Hard: 76.1%	IoU = 0.5	Vượt trội trên cả 3 mức độ khó
Định vị landmark	AFLW	Normalized Mean Error (NME)	5.39%	5 điểm landmark	Kết quả mô hình đạt được đã vượt qua phương pháp tốt nhất trước đó đến 20%
Định vị landmark	AFLW	Failure Rate	1.84%	NME > 10%	Tỉ lệ thất bại rất thấp
Tốc độ xử lý		Thời gian xử lý	99ms/ảnh	GPU Titan X Ảnh 640x480	Nhanh hơn 12 lần so với phương pháp trước
Khả năng		Average Preci-	62.3%	Mức occlu-	Vẫn phát hiện tốt

chịu occlusion (Mức độ mô hình vẫn có thể phát hiện hoặc định vị khuôn mặt chính xác, ngay cả khi khuôn mặt đó bị che khuất một phần)		sion (AP) với occlusion		sion 40-70%	khuôn mặt bị che khuất
Kích thước khuôn mặt nhỏ	WIDER FACE	Tỉ lệ phát hiện	85.7%	Khuôn mặt < 30x30px	Hiệu quả tốt với khuôn mặt rất nhỏ

Mỗi bộ dữ liệu đề cập phía trên có tiêu chuẩn và độ khó khác nhau, dùng để kiểm tra mô hình trong các khía cạnh cụ thể: phát hiện, định vị, chịu che khuất...

Bảng 7: Đặc điểm các dataset trong thử nghiệm mô hình MTCNN

Dataset	Mục đích chính	Đặc điểm nổi bật
FDDB	Phát hiện khuôn mặt	Gồm nhiều ảnh thực tế, không gắn khung vuông chính xác; dùng để đo TPR (True Positive Rate).
WIDER FACE	Phát hiện khuôn mặt	Rất khó, gồm nhiều tình huống như nhỏ, nghiêng đầu, che khuất, ánh sáng xáu...
AFLW	Định vị landmark khuôn mặt	Chứa thông tin về vị trí các điểm mốc (mắt, mũi, miệng) trên khuôn mặt.

1.2. MÔ HÌNH INCEPTIONRESNETV1

1.2.1. Ý tưởng, hướng tiếp cận

InceptionResnetV1 là một mô hình deep learning kết hợp tinh hoa của hai kiến trúc mạng: Inception và ResNet (Residual Network). Mô hình này được thiết kế để tối ưu hóa hiệu suất trong các bài toán nhận dạng khuôn mặt bằng cách cân bằng giữa độ sâu của mạng (ResNet) và khả năng trích xuất đa chiều thông tin (Inception).

InceptionResnetV1 được giới thiệu trong nghiên cứu về FaceNet [13] và các phiên bản cải tiến sau này. Mục tiêu chính của mô hình là tạo ra embedding - vector đặc trưng cho ảnh khuôn mặt, giúp so sánh hoặc nhận dạng khuôn mặt dựa trên khoảng cách Euclidean hoặc cosine similarity giữa các embedding.

1.2.2. Cách hoạt động

InceptionResnetV1 là một mô hình deep learning kết hợp giữa kiến trúc Inception và ResNet, được tối ưu cho bài toán nhận dạng khuôn mặt. Dưới đây là cấu trúc chi tiết và cách thức hoạt động của mô hình:

Mô hình gồm 3 phần chính:

- Stem Block: Lớp tiền xử lý ảnh, giảm kích thước và chuẩn bị đặc trưng cơ bản.
- Các khối Inception-ResNet (A, B, C): Kết hợp Inception module và residual connection để trích xuất đặc trưng đa tỷ lệ.
- Reduction Block (A, B): Giảm kích thước không gian (spatial dimension) giữa các khối.

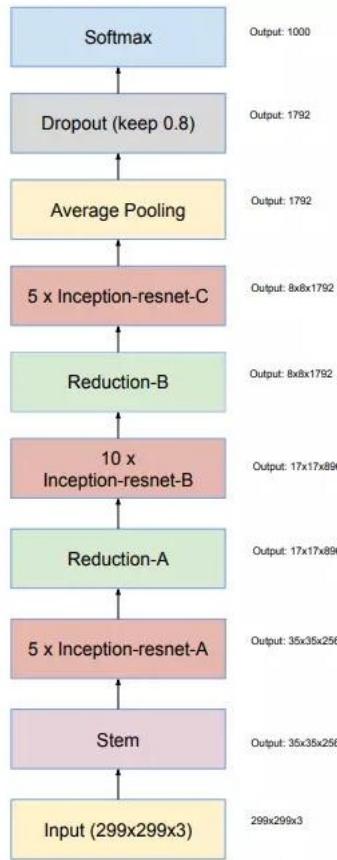
Global Average Pooling & Fully Connected Layer: Tổng hợp đặc trưng và tạo embedding.

InceptionResnetV1 là một mô hình deep learning kết hợp giữa kiến trúc Inception và ResNet, được tối ưu cho bài toán nhận dạng khuôn mặt. Dưới đây là cấu trúc chi tiết và cách thức hoạt động của mô hình:

Mô hình gồm 3 phần chính:

- Stem Block: Lớp tiền xử lý ảnh, giảm kích thước và chuẩn bị đặc trưng cơ bản.
- Các khối Inception-ResNet (A, B, C): Kết hợp Inception module và residual connection để trích xuất đặc trưng đa tỷ lệ.
- Reduction Block (A, B): Giảm kích thước không gian (spatial dimension) giữa các khối.

Global Average Pooling & Fully Connected Layer: Tổng hợp đặc trưng và tạo embedding.



Hình 2: Kiến trúc mô hình InceptionResNet V1

1.2.2.1. Stem Block

Mô tả: Khởi tạo quá trình trích xuất đặc trưng, giảm kích thước ảnh từ 299x299 xuống 35x35 qua các lớp Conv và MaxPooling, đồng thời tăng dần chiều sâu kênh (3 → 256).
Đặc điểm:

- Kết hợp Conv stride=2 và MaxPooling để giảm kích thước nhanh.
- Dùng padding 'valid' để tránh dư thừa thông tin.
- Mục đích: Trích xuất đặc trưng cấp thấp (edges, textures) và giảm chiều dữ liệu.

Bảng 8: Kiến trúc Stem Block

Layer	Filter	Stride	Padding	Activation	Output Shape	Ghi chú
Input	-	-	-	-	299×299×3	Ảnh đầu vào

Conv2D 3x3	32	2	valid	ReLU	$149 \times 149 \times 32$	Giảm 50% kích thước ảnh (299→149)
Conv2D 3x3	32	1	valid	ReLU	$147 \times 147 \times 32$	
Conv2D 3x3	64	1	same	ReLU	$147 \times 147 \times 64$	Giữ nguyên kích thước
MaxPool 3x3	-	2	valid	-	$73 \times 73 \times 64$	Giảm kích thước (147→73)
Conv2D 1x1	80	1	valid	ReLU	$73 \times 73 \times 80$	Tăng chiều sâu kênh (64→80)
Conv2D 3x3	192	1	valid	ReLU	$71 \times 71 \times 192$	
Conv2D 3x3	256	2	valid	ReLU	$35 \times 35 \times 256$	Chuẩn bị đưa vào khối Inception

1.2.2.2. Inception-ResNet-A(lặp lại 5 lần)

Mô tả: Gồm 5 khối liên tiếp, mỗi khối kết hợp 3 nhánh Conv (1x1, 3x3, và chuỗi 3x3 lồng nhau) để trích xuất đặc trưng đa tỷ lệ, sau đó cộng với đầu vào (residual connection).

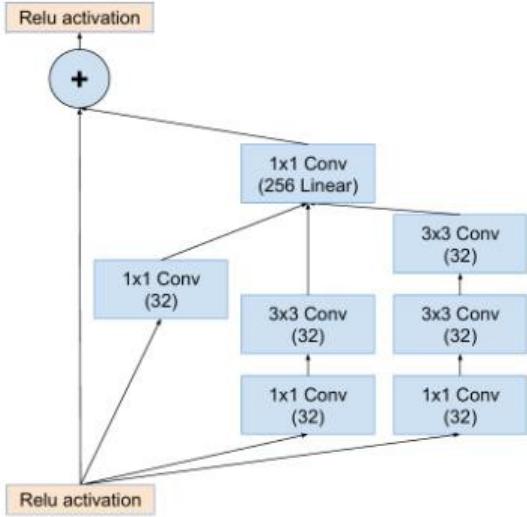
Đặc điểm:

- Dùng Conv 1x1 để nén thông tin trước khi áp dụng Conv lớn hơn.
- Projection layer (Conv 1x1) điều chỉnh chiều kênh cho khớp residual.

Bảng 9: Kiến trúc Inception-ResNet-A

Branch	Layer	Filter	Stride	Padding	Activation	Output Shape	Ghi chú
Branch 1	Conv2D 1x1	32	1	same	ReLU	$35 \times 35 \times 32$	Trích xuất đặc trưng

							tuyên tính
Branch 2	Conv2D 1x1	32	1	same	ReLU	35×35×32	
Branch 2	Conv2D 3x3	32	1	same	ReLU	35×35×32	Bắt đặc trung không gian 3x3
Branch 3	Conv2D 1x1	32	1	same	ReLU	35×35×32	
Branch 3	Conv2D 3x3	32	1	same	ReLU	35×35×32	
Branch 3	Conv2D 3x3	32	1	same	ReLU	35×35×32	Mở rộng recep- tive field
Concate- nate	-	-	-	-	-	35×35×96	Ghép 3 nhánh
Projection	Conv2D 1x1	256	1	same	Linear	35x35x25 6	Không dùng ReLU để giữ nguyên giá trị residual
Residual Connection	Add([input,output])	-	-	-	-	35x35x25 6	Giải quyết vanish- ing gra- dient



Hình 3: Kiến trúc Inception-ResNet-A

1.2.2.3. Reduction-A

Mô tả: Giảm kích thước không gian từ 35×35 xuống 17×17 qua 3 nhánh song song: Conv stride=2, chuỗi Conv phức tạp, và MaxPooling.

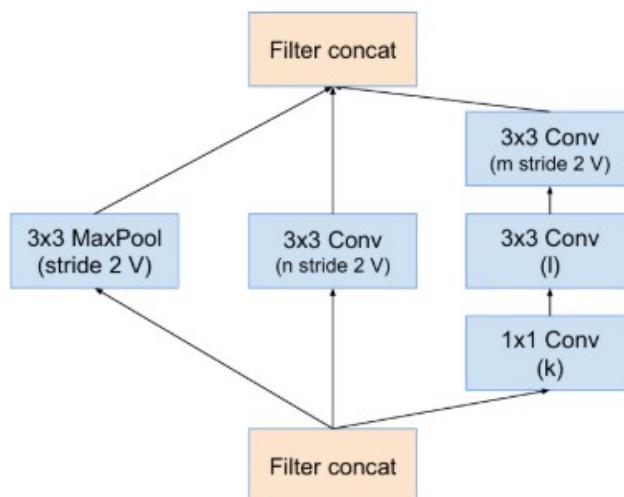
Đặc điểm:

- Kết hợp nhiều phương pháp downsampling để bảo toàn thông tin.
- Output ghép từ 3 nhánh \rightarrow 896 kênh.

Bảng 10: Kiến trúc Reduction-A

Branch	Layer	Filter	Stride	Padding	Activation	Output Shape	Ghi chú
Branch 1	Conv2D 3x3	384	2	valid	ReLU	$17 \times 17 \times 384$	Giảm kích thước bằng stride=2
Branch 2	Conv2D 1x1	192	1	same	ReLU	$35 \times 35 \times 192$	
Branch 2	Conv2D 3x3	224	1	same	ReLU	$35 \times 35 \times 224$	

Branch 2	Conv2D 3x3	256	2	valid	ReLU	$17 \times 17 \times 256$	Nhánh phức tạp để bảo toàn thông tin
Branch 3	MaxPool 3x3	-	2	valid	-	$17 \times 17 \times 256$	Downsample đơn giản
Concatenate	-	-	-	-	-	$17 \times 17 \times 896$	Kết hợp đặc trưng từ 3 nhánh



Hình 4: Kiến trúc Reduction-A

1.2.2.4. Inception-Resnet-B (Lặp lại 10 lần)

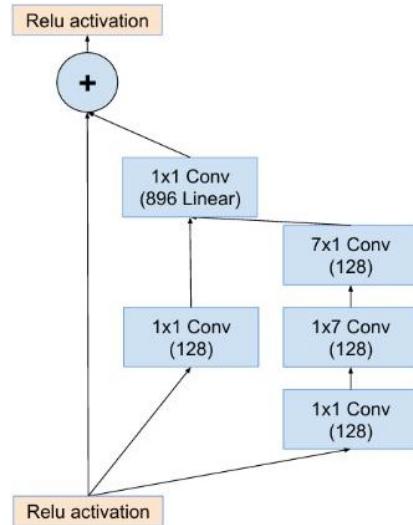
Mô tả: Gồm 10 khối, mỗi khối sử dụng Conv 1x7 và 7x1 thay vì Conv 7x7 để tiết kiệm tham số, vẫn giữ residual connection.

Đặc điểm:

- Tách tích chập lớn (7×7) thành $1 \times 7 + 7 \times 1 \rightarrow$ hiệu quả tính toán.
- Phù hợp cho đặc trưng ở kích thước trung bình (17×17).

Bảng 11: Kiến trúc Inception-Resnet-B

Branch	Layer	Filter	Stride	Padding	Activation	Output Shape	Ghi chú
Branch 1	Conv2D 1x1	128	1	same	ReLU	17×17×128	Nén thông tin
Branch 2	Conv2D 1x1	128	1	same	ReLU	17×17×128	
Branch 2	Conv2D 1x7	128	1	same	ReLU	17×17×128	
Branch 2	Conv2D 7x1	128	1	same	ReLU	17×17×128	Tách tích chập 7x7 → tiết kiệm tham số
Concatenate						17×17×256	
Projection	Conv2D 1x1	896	1	same	Linear	17×17×896	Không dùng ReLU để khớp residual
Residual Connection	Add([input, output])					17×17×896	Ôn định quá trình huấn luyện



Hình 5: Kiến trúc Inception-ResNet-B

1.2.2.5. Reduction-B

Mô tả: Tiếp tục giảm kích thước từ 17x17 xuống 8x8 qua 2 nhánh Conv stride=2 và 1 nhánh MaxPooling.

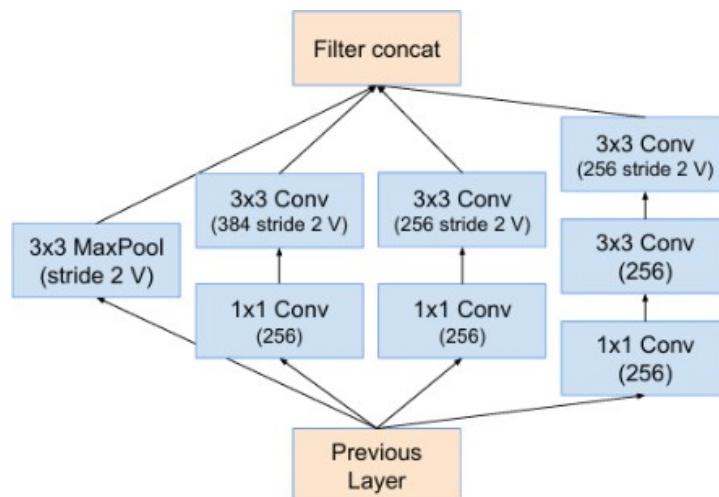
Đặc điểm:

- Tương tự Reduction-A nhưng tối ưu cho kích thước nhỏ hơn.
- Output ghép 3 nhánh → 896 kênh.

Bảng 12: Kiến trúc Reduction-B

Branch	Layer	Filter	Stride	Padding	Activation	Output Shape	Ghi chú
Branch 1	Conv2D 1x1	256	1	same	ReLU	17x17x256	
Branch 1	Conv2D 3x3	384	2	valid	ReLU	8×8×384	Giảm kích thước phức tạp

Branch 2	Conv2D 1x1	256	1	same	ReLU	17x17x256	
Branch 2	Conv2D 3x3	256	2	valid	ReLU	8x8x256	
Branch 3	MaxPool 3x3	-	2	valid	ReLU	8x8x256	
Concatenate	-	-	-	-	-	8x8x896	Chuẩn bị cho khối cuối cùng



Hình 6: Kiến trúc Reduction-B

1.2.2.6. Inception-ResNet-C (Lắp lại 5 lần)

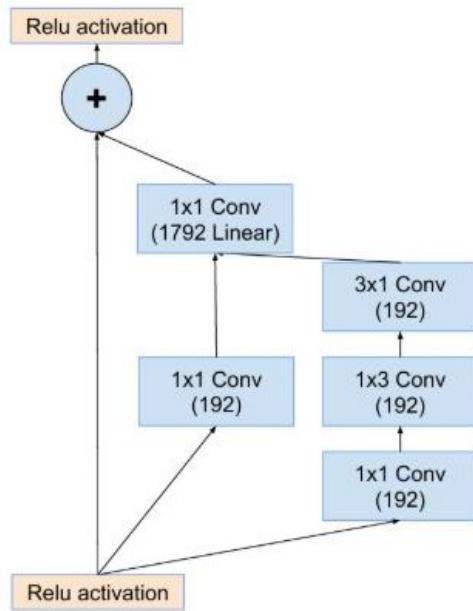
Mô tả: Gồm 5 khối cuối, dùng Conv 1x3 và 3x1 thay vì Conv 3x3 để trích xuất đặc trưng mịn ở kích thước nhỏ (8x8).

Đặc điểm:

- Tích chập tách rời giúp bắt các đặc trưng theo chiều ngang/dọc riêng biệt.
- Output 1792 kênh → đủ phong phú cho GAP.

Bảng 13: Kiến trúc Inception-ResNet-C

Branch	Layer	Filter	Stride	Padding	Activation	Output Shape	Ghi chú
Branch 1	Conv2D 1x1	128	1	same	ReLU	8x8x128	
Branch 2	Conv2D 1x1	128	1	same	ReLU	8x8x128	
Branch 2	Conv2D 1x3	128	1	same	ReLU	8x8x128	
Branch 2	Conv2D 3x1	128	1	same	ReLU	8x8x128	Tách tích chập 3x3 → tối ưu tính toán
Concatenate	-	-	-	-	-	8x8x256	
Projection	Conv2D 1x1	1792	1	same	Linear	$8 \times 8 \times 1792$	
Residual Connection	Add([input, output])	-	-	-	-	$8 \times 8 \times 1792$	



Hình 7: Kiến trúc Inception-ResNet-C

1.2.2.7. Output Block

Mô tả: Chuyển đặc trưng $8 \times 8 \times 1792$ thành vector 512D qua Global Average Pooling và Fully Connected, sau đó chuẩn hóa L2.

Đặc điểm:

- GAP thay thế FC để giảm overfitting.
- Embedding 512D dùng cho so sánh cosine similarity trong face recognition.

Bảng 14: Kiến trúc Output Block

Layer	Tham số	Activation	Output Shape	Ghi chú
Global Average Pooling	-	-	$1 \times 1 \times 1792$	Thay thế Fully Connected để giảm overfitting
Fully Connected	512 units	L2 Norm	512	Embedding chuẩn hóa cho so sánh cosine similarity

1.2.3. Công thức

1.2.3.1. Embedding Function

FaceNet học một hàm ánh xạ (embedding function) để ánh xạ một ảnh khuôn mặt vào một không gian Euclidean

$$f(x) \in \mathbb{R}^d$$

Trong đó:

- x : ảnh đầu vào (kích thước ban đầu là $H \times W \times C$)
- $f(x)$: vector đặc trưng (embedding) của ảnh. ($f(x) \in \mathbb{R}^d$)
- d : kích thước của embedding (thường là 128)

$$f(x) = \frac{W \cdot \text{GAP}(\text{CNN}(x))}{\|W \cdot \text{GAP}(\text{CNN}(x))\|_2}$$

Trong đó:

- $\text{CNN}(x)$: Đặc trưng trích xuất từ các lớp tích chập.
- GAP: Global Average Pooling, chuyển đặc trưng thành vector R^{1792} (ví dụ với Inception-ResNet).
- W : Trọng số lớp FC $\in R^{128 \times 1792}$

1.2.3.2. Triplet Loss

Công thức chính được sử dụng để huấn luyện là **Triplet Loss**:

$$\mathcal{L} = \sum_{i=1}^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

Trong đó:

- x_i^a : ảnh anchor – khuôn mặt gốc
- x_i^p : ảnh positive – cùng người với anchor

- x_i^n : ảnh negative – người khác với anchor
- $f(x) \in Rd$: embedding vector
- $\|\cdot\|_2^2$: norm Euclidean bình phương
- α : **margin** (thường chọn khoảng 0.2)
- $[\cdot] +$: hàm **ReLU**: $[z] += max(z, 0)$
- N: số lượng triplet trong batch

Buộc embedding của anchor và positive gần nhau hơn so với anchor và negative ít nhất là một khoảng cách α

$$f(x_i^a), f(x_i^p), f(x_i^n) \in \mathbb{R}^{128}$$

1.2.3.3. Hard Negative Mining

Để tăng hiệu quả, FaceNet chọn các triplet "khó" (hard triplets):

$$\|f(x^a) - f(x^p)\|_2^2 + \alpha < \|f(x^a) - f(x^n)\|_2^2$$

1.2.4. Siêu tham số

Bảng 15: Siêu tham số mô hình InceptionResNet V1

Hyperparameter	Giá trị trong bài báo	Khoảng giá trị tối ưu	Mô tả chi tiết
Embedding size	128	64 - 512	Kích thước vector đặc trưng. Nhỏ hơn → tốc độ nhanh, lớn hơn → độ chính xác cao.
Margin (α)	0.2	0.1 - 0.5	Biên trong Triplet Loss. Giá trị lớn hơn làm mô hình học cách phân biệt rõ hơn.
Batch size	1800 - 2600 ảnh/batch	512 - 3000	Mỗi batch chứa ~40 triplet. Batch lớn hơn giúp ổn định huấn luyện.
Learning rate	Ban đầu 0.05	0.001 - 0.1	Dùng SGD với momentum. Có thể

	rồi giảm dần		dùng lịch giảm theo epoch.
Momentum	0.9	0.8 - 0.99	Hệ số momentum cho SGD. Giá trị cao giúp thoát local optimum tốt hơn.
Dropout	Không dùng	0.2 - 0.5 (nếu có)	FaceNet gốc không dùng dropout, nhưng có thể thêm để tránh overfitting.
Số lượng triplet/batch	40	20 - 100	Số triplet được chọn trong mỗi batch. Hard mining hiệu quả hơn với nhiều triplet.
Optimizer	SGD với momentum	Adam, RMSprop (tùy biến)	SGD được dùng trong bài báo, nhưng Adam có thể thử nếu dữ liệu nhỏ.
Weight decay	5e-4	1e-5 - 1e-3	Điều chỉnh trọng số để tránh overfitting.
Epochs	500 - 1000	100 - 1500	Số epoch phụ thuộc vào dữ liệu. FaceNet cần huấn luyện lâu do dùng triplet loss.

1.2.5. Mô phỏng

- Giả sử đầu vào là một ảnh màu kích thước nhỏ $75 \times 75 \times 3$, thay vì $299 \times 299 \times 3$ như chuẩn gốc.
- Mục tiêu: Theo dõi cách ảnh được biến đổi qua các khối chính của mô hình Inception-ResNet-v1, giảm chiều, và tăng số lượng đặc trưng.

Giai đoạn 1: Khối Stem

Bảng 16: Mô phỏng InceptionResNet V1 Khối Stem

Bước	Mô tả	Kích thước đầu ra
1	Conv 3×3 , stride=2, valid	$\frac{75-3}{2} + 1 = 37 \rightarrow 37 \times 37 \times 32$
2	Conv 3×3 , stride=1, valid	$\frac{37-3}{1} + 1 = 35 \rightarrow 35 \times 35 \times 32$
3	Conv 3×3 , stride=1, valid	$\frac{35-3}{1} + 1 = 33 \rightarrow 33 \times 33 \times 64$

- Sau đó tách thành 2 nhánh

Bảng 17: Các nhánh của Khối Stem

Nhánh	Phép toán	Kích thước
Nhánh 1	Maxpool 3×3 , stride=2	$\frac{33-3}{2} + 1 = 16 \rightarrow 16 \times 16 \times 64$
Nhánh 2	Conv 3×3 , stride=2	$\frac{33-3}{2} + 1 = 16 \rightarrow 16 \times 16 \times 96$

- Nối 2 nhánh bằng phép toán **Concat** $\rightarrow 16 \times 16 \times (64+96) = 16 \times 16 \times 160$

Giai đoạn 2: Khối Inception-ResNet-A

- Giữ nguyên kích thước không gian, thay đổi số channels.
- Đầu vào: $16 \times 16 \times 160$

Bảng 18: Mô phỏng InceptionResNet V1 Khối Inception-ResNet-A

Nhánh	Phép toán	Kết quả
Nhánh 1	1×1 conv $\rightarrow 32$	$16 \times 16 \times 32$
Nhánh 2	1×1 conv $\rightarrow 32 \rightarrow 3 \times 3$ conv $\rightarrow 32$	$16 \times 16 \times 32$
Nhánh 3	1×1 conv $\rightarrow 32 \rightarrow 3 \times 3$ conv $\rightarrow 48 \rightarrow 3 \times 3$ conv $\rightarrow 64$	$16 \times 16 \times 64$

- Nối 3 nhánh bằng phép toán Concat: $32 + 32 + 64 = 128 \rightarrow 16 \times 16 \times 128$
- Dùng 1×1 conv để nâng từ 128 $\rightarrow 160 \rightarrow$ Cộng với đầu vào residual $\rightarrow 16 \times 16 \times 160$
- Hàm kích hoạt: ReLU
- Kết quả: $16 \times 16 \times 160$

Giai đoạn 3: Khối Reduction-A

- Đầu vào: $16 \times 16 \times 160$
- Mục tiêu: giảm không gian từ $16 \times 16 \rightarrow 8 \times 8$

Bảng 19: Mô phỏng InceptionResNet V1 Khối Reduction-A

Nhánh	Phép toán	Kết quả

Nhánh 1	Conv 3×3 , stride=2	$(16 - 3)/2 + 1 = 7 \rightarrow 7 \times 7 \times 160$
Nhánh 2	1×1 conv $\rightarrow 96 \rightarrow 3 \times 3$ conv $\rightarrow 96 \rightarrow 3 \times 3$ conv stride=2 $\rightarrow 128$	$7 \times 7 \times 128$
Nhánh 3	Maxpool 3×3 stride=2	$(16 - 3)/2 + 1 = 7 \rightarrow 7 \times 7 \times 160$ (giữ nguyên số channels ban đầu)

- Nối 3 nhánh bằng phép toán Concat: $7 \times 7 \times (160 + 128 + 160) = 7 \times 7 \times 448$

Giai đoạn 4: Khối Inception-ResNet-B

Đầu vào: $7 \times 7 \times 448$

Bảng 20: Mô phỏng InceptionResNet V1 Khối Inception-ResNet-B

Nhánh	Phép toán	Kết quả
Nhánh 1	1×1 conv $\rightarrow 128$	$7 \times 7 \times 128$
Nhánh 2	1×1 conv $\rightarrow 128 \rightarrow 1 \times 7$ conv $\rightarrow 160 \rightarrow 7 \times 1$ conv $\rightarrow 192$	$7 \times 7 \times 192$

- Nối 2 nhánh bằng phép toán Concat: $7 \times 7 \times (128 + 192) = 7 \times 7 \times 320$
- Dùng 1×1 conv: $320 \rightarrow 448$ để cộng với shortcut
- Hàm kích hoạt: ReLU
- Kết quả: $7 \times 7 \times 448$

Giai đoạn 5: Khối Reduction-B

Đầu vào: $7 \times 7 \times 448$

Mục tiêu: giảm từ $7 \times 7 \rightarrow 3 \times 3$

Bảng 21: Mô phỏng InceptionResNet V1 Khối Reduction-B

Nhánh	Phép toán	Kết quả
Nhánh 1	1×1 conv $\rightarrow 256 \rightarrow 3 \times 3$ conv stride=2 $\rightarrow 384$	$3 \times 3 \times 384$
Nhánh 2	1×1 conv $\rightarrow 256 \rightarrow 3 \times 3$ conv $\rightarrow 288 \rightarrow 3 \times 3$ conv stride=2 $\rightarrow 320$	$3 \times 3 \times 320$
Nhánh 3	Maxpool 3×3 stride=2	$3 \times 3 \times 448$

- Nối 3 nhánh bằng phép toán Concat: $3 \times 3 \times (384 + 320 + 448) = 3 \times 3 \times 1152$

Giai đoạn 6: Khối Inception-ResNet-C

- Đầu vào: $3 \times 3 \times 1152$

Bảng 22: Mô phỏng InceptionResNet V1 Khối Inception-ResNet-C

Nhánh	Phép toán	Kết quả
Nhánh 1	1×1 conv $\rightarrow 192$	$3 \times 3 \times 192$
Nhánh 2	1×1 conv $\rightarrow 192 \rightarrow 1 \times 3$ conv $\rightarrow 224 \rightarrow 3 \times 1$ conv $\rightarrow 256$	$3 \times 3 \times 256$

- Nối 2 nhánh bằng phép toán Concat: $192 + 256 = 448 \rightarrow 3 \times 3 \times 448$
- Dùng 1×1 conv: $448 \rightarrow 1152$
- Hàm kích hoạt: ReLU
- Kết quả cuối cùng: $3 \times 3 \times 1152$

1.2.6. Kết quả thử nghiệm đã có

Trong bài báo Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning của Christian Szegedy Google Inc., Sergey Ioffe và Vincent Vanhoucke [14] đã cho thấy hiệu suất cuối cùng của mô hình tương đương với Inception-v3, khả năng hội tụ nhanh hơn khiến Inception-ResNet-v1 trở thành một lựa chọn hấp dẫn cho các ứng dụng yêu cầu đào tạo nhanh chóng.

Bảng 23: Kết quả thử nghiệm với một mô hình 1 lần cắt trên tập validation

Mô hình	Top-1 Error (%)	Top-5 Error (%)
BN-Inception	25.2	7.8
Inception-v3	21.2	5.6
Inception-Resnet-v1	21.3	5.5
Inception-v4	20.0	5.0
Inception-Resnet-v2	19.9	4.9

Bảng 24: Kết quả thử nghiệm với một mô hình 12 lần cắt trên tập validation

Mô hình	Top-1 Error (%)	Top-5 Error (%)
Resnet-151	21.4	5.7
Inception-v3	19.8	4.6
Inception-Resnet-v1	19.8	4.6
Inception-v4	18.7	4.2
Inception-Resnet-v2	18.7	4.1

Bảng 25: Kết quả thử nghiệm với một mô hình 144 lần cắt trên tập validation

Mô hình	Top-1 Error (%)	Top-5 Error (%)
Resnet-151	19.4	4.5
Inception-v3	18.9	4.3
Inception-Resnet-v1	18.8	4.3
Inception-v4	17.7	3.8
Inception-Resnet-v2	17.8	3.7

1.3. KỸ THUẬT RAG (RETRIEVAL-AUGMENTED GENERATION)

1.3.1. Ý tưởng, hướng tiếp cận

Sự phát triển mạnh mẽ của các mô hình ngôn ngữ lớn (Large Language Models - LLMs) như ChatGPT, LLaMA-2, Qwen hay Mistral đã mở ra nhiều tiềm năng ứng dụng trong xử lý ngôn ngữ tự nhiên. Tuy nhiên, một hạn chế đáng kể mà các mô hình này thường gặp phải là hiện tượng "hallucination", tức là tạo ra các nội dung nghe có vẻ hợp lý và trôi chảy nhưng lại thiếu chính xác hoặc không có cơ sở trong thực tế.

Từ vấn đề trên, một hướng tiếp cận hiệu quả đã được đề xuất nhằm giảm thiểu hiện tượng này, đó là kỹ thuật RAG (Retrieval-Augmented Generation). Đây là một phương pháp kết hợp giữa hai lĩnh vực quan trọng trong NLP: truy xuất thông tin (retrieval) và sinh ngôn ngữ tự nhiên (generation). Ý tưởng chính là không để mô hình ngôn ngữ tự "biến ra" câu trả lời, mà thay vào đó, cung cấp cho nó thông tin có liên quan từ một cơ sở tri thức đã được xác thực.

Về mặt cấu trúc, RAG vận hành theo hai giai đoạn chính:

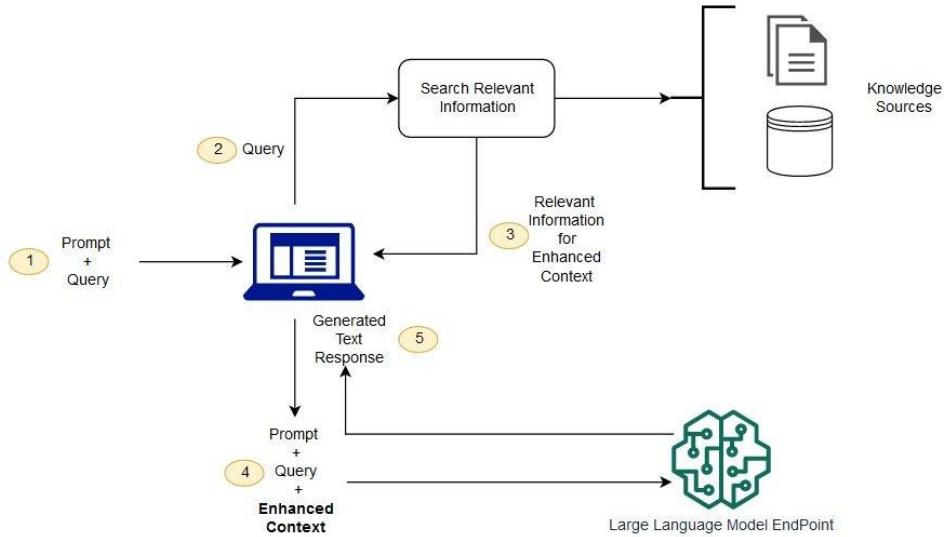
- Truy xuất thông tin (Retrieval Module): Giai đoạn này sử dụng một mô hình truy vấn để tìm kiếm các đoạn văn bản có liên quan từ kho dữ liệu có sẵn (chẳng hạn như tài liệu PDF, cơ sở tri thức nội bộ, hoặc dữ liệu đã được nhúng vào vector database). Kết quả của bước này là các đoạn văn bản chứa thông tin chính xác và có ngữ cảnh phù hợp với câu hỏi đầu vào.
- Sinh văn bản (Generation Module): Sau khi có được các đoạn văn liên quan, mô hình ngôn ngữ lớn (LLM) sẽ sử dụng các thông tin đó để tạo ra câu trả lời dưới dạng ngôn ngữ tự nhiên. Mục tiêu là tạo ra câu trả lời rõ ràng, chính xác và dễ hiểu cho người dùng.

Trong năm 2020, ý tưởng Retrieval-Augmented Generation (RAG) lần đầu tiên được giới thiệu trong bài báo khoa học "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" [10] của nhóm tác giả tại Facebook AI Research (FAIR) do Patrick Lewis đứng đầu. Bài báo này đề xuất một kiến trúc lai (hybrid architecture) nhằm tận dụng cả khả năng truy xuất tri thức của các mô hình tìm kiếm và sức mạnh tạo văn bản của các mô hình sinh (generative models) như BART.

Mô hình RAG không yêu cầu toàn bộ tri thức phải được “học thuộc” trong tham số của mô hình ngôn ngữ như trước đây, thay vào đó, nó cho phép mô hình truy xuất động các thông tin liên quan trong lúc sinh phản hồi, nhờ đó giúp giảm thiểu vấn đề “hallucination” và nâng cao độ chính xác của câu trả lời.

Kết quả trong bài báo gốc cho thấy RAG đạt hiệu năng vượt trội trên nhiều tác vụ yêu cầu kiến thức như: trả lời câu hỏi mở (Open-domain QA), liên kết thực thể (Entity linking), và xác minh thông tin (Fact verification). Chính điều này đã mở ra một hướng đi mới cho việc xây dựng các hệ thống thông minh có khả năng kết hợp tri thức thực tế với khả năng sinh ngôn ngữ tự nhiên linh hoạt.

1.3.2. Quy trình hoạt động của RAG



Hình 8: Quy trình hoạt động của RAG

Bước 1: Thu thập dữ liệu

Quá trình đầu tiên và quan trọng nhất của Retrieval Augmented Generation là thu thập dữ liệu. Đây chính là “nguyên liệu thô” quyết định mức độ chính xác của câu trả lời sau này. Nguồn dữ liệu trong hệ thống được chia thành hai nhóm:

- Kho tư liệu nội bộ: bao gồm hướng dẫn vận hành, báo cáo kỹ thuật, cơ sở dữ liệu khách hàng, nhật ký trao đổi e-mail hoặc hồ sơ giao dịch do doanh nghiệp sở hữu. Nhóm tài liệu này cung cấp thông tin có độ tin cậy cao, hỗ trợ hệ thống trả lời những câu hỏi liên quan đến quy trình và nghiệp vụ bên trong.
- Kho tư liệu bên ngoài: là các ấn phẩm học thuật, bài viết chuyên ngành, diễn đàn cộng đồng hoặc nguồn mở trên Internet. Việc bổ sung dữ liệu ngoại vi giúp hệ thống cập nhật tri thức mới, mở rộng góc nhìn và tránh “thiếu nguồn” đối với các chủ đề thời sự.

Bước 2: Tổ chức và phân loại dữ liệu

Sau khi thu thập, toàn bộ tài liệu được sắp xếp lại để tối ưu hóa tốc độ truy xuất:

- Cơ chế phân loại: tài liệu được nhóm theo *chủ đề*, *loại nội dung* hoặc *tình huống sử dụng*. Ví dụ một công ty dịch vụ CNTT có thể tách dữ liệu thành “Bảo trì hệ thống”, “Giải pháp bảo mật”, “Hỗ trợ khách hàng”, v.v.
- Kết quả phân loại: mỗi tài liệu được gắn nhãn (metadata) về từ khóa, phạm vi và độ ưu tiên. Nhờ vậy, khi truy vấn tới, hệ thống chỉ tìm kiếm trong phạm vi hẹp, rút ngắn đáng kể thời gian đáp ứng.

Bước 3: Nhúng tài liệu

Để máy tính hiểu được ngữ nghĩa, văn bản thuần phải chuyển thành vector số:

- Quy trình nhúng: áp dụng các mô hình ngôn ngữ sẵn huấn luyện (BERT, RoBERTa, hoặc Sentence-Transformers). Mỗi đoạn văn sau khi đi qua mạng Transformer sẽ trở thành một vector kích thước cố định, phản ánh ý nghĩa của đoạn.
- Lợi ích: khi truy vấn của người dùng được chuyển sang cùng không gian vector, hệ thống có thể đo *độ tương đồng ngữ nghĩa* thay vì so khớp từ khóa đơn thuần. Ví dụ, dù người dùng hỏi “làm sao để sửa lỗi hệ thống?” hay “khắc phục lỗi IT như thế nào?”, hệ thống đều hiểu đây là cùng một ý định và truy xuất các tài liệu liên quan.

Bước 4: Xử lý truy vấn người dùng

Khi hệ thống tiếp nhận một truy vấn từ người dùng, quá trình xử lý sẽ diễn ra theo hai bước chủ đạo: phân tích ngữ nghĩa và so khớp thông tin.

- Phân tích ngữ nghĩa: Truy vấn được chuyển thành vector số bằng kỹ thuật mã hóa ngữ nghĩa, tương tự như cách biểu diễn tài liệu đã được thực hiện trước đó. Việc này giúp hệ thống “nắm bắt” được mục đích thực sự ẩn sau câu hỏi, kể cả khi người dùng sử dụng từ ngữ khác nhau. Chẳng hạn, câu hỏi “Làm sao để bảo trì máy chủ hiệu quả?” sẽ được phân tích để xác định các thành phần chính như “bảo trì” và “máy chủ” – từ đó truy ra ý định tìm tài liệu liên quan đến bảo trì hệ thống.
- So khớp thông tin: Sau khi biểu diễn truy vấn dưới dạng vector, hệ thống sử dụng phép đo khoảng cách (thường là cosine similarity) để tìm các đoạn văn bản có nội dung gần nhất trong không gian vector. Những kết quả tương đồng nhất sẽ được chọn lọc và chuyển tiếp đến bước sinh phản hồi.

Nhờ tích hợp kỹ thuật xử lý ngôn ngữ tự nhiên (NLP), hệ thống không chỉ dừng lại ở việc trả lời đúng trọng tâm mà còn có khả năng mở rộng thông tin, giúp người dùng có thêm bối cảnh và kiến thức liên quan đến câu hỏi ban đầu.

Bước 5: Sinh phản hồi bằng mô hình LLM

Các đoạn văn liên quan được ghép với câu hỏi gốc và cung cấp cho mô hình ngôn ngữ lớn (LLM):

- Hợp nhất ngữ cảnh: LLM đọc đồng thời truy vấn và các trích đoạn đã truy xuất, từ đó tổng hợp thông tin.
- Sinh văn bản tự nhiên: mô hình tạo câu trả lời mạch lạc, đầy đủ và điều chỉnh văn phong sao cho phù hợp với người dùng.

Ví dụ: khi người dùng yêu cầu “Hướng dẫn bảo mật mạng nội bộ”, phản hồi không chỉ liệt kê quy trình cơ bản mà còn đề xuất các công cụ bảo mật hiện đại, nhờ dữ liệu ngoại vi đã được thu thập và nhúng từ trước.

1.3.3. Các siêu tham số sử dụng trong RAG

1.3.3.1. Retrieval (*Truy xuất*)

Bảng 26: Siêu tham số của RAG - Retrieval

Hyperparameter	Mô tả	Giá trị thường gặp
retriever_top_k	Số lượng tài liệu gần nhất (most relevant documents) được truy xuất	Từ 3 đến 10
similarity_metric	Phương pháp đo độ tương đồng giữa câu truy vấn và tài liệu trong DB	cosine, dot-product, euclidean
embedding_model	Mô hình dùng để chuyển văn bản thành vector	Sentence-transformers, OpenAI,...
vectorstore	Loại database dùng lưu trữ và truy xuất vector	Chroma, FAISS, Weaviate,...

1.3.3.2. Generation (*Sinh câu trả lời*)

Bảng 27: Siêu tham số của RAG - Generation

Hyperparameter	Mô tả	Giá trị thường gặp
temperature	Mức độ ngẫu nhiên của đầu ra	Từ 0.3 đến 0.7
max_tokens	Giới hạn số token tối đa mô hình được phép sinh	Từ 512 đến 4096
model_name	Tên mô hình sinh	Gemma, Llama 3,...
n_ctx	Bối cảnh tối đa đầu vào mô hình (context window)	Từ 2048 đến 8192

1.3.4. Mô phỏng

Ví dụ: Giả sử người dùng đặt câu hỏi: "Ai là hiệu trưởng Trường Đại học ABC vào năm 2020?"

Bước 1 – Truy xuất (Retrieval): RAG sẽ sử dụng một retriever để truy vấn kho tri thức (Wikipedia hoặc dữ liệu vector hóa bằng FAISS, Chroma,...).

Giả sử retriever tìm được 2 đoạn văn có liên quan sau:

- Đoạn 1: “Ông Nguyễn Văn A được bổ nhiệm làm hiệu trưởng Trường Đại học ABC vào tháng 6 năm 2020.”
- Đoạn 2: “Bà Trần Thị B giữ chức hiệu trưởng Trường Đại học ABC từ năm 2015 đến giữa năm 2020.”

Bước 2 – Sinh (Generation): RAG sử dụng mô hình sinh (ví dụ: BART hoặc T5) để đưa ra câu trả lời dựa trên câu hỏi + các đoạn truy xuất.

[Question:"AilàhiệutrưởngTrườngĐạihọcABCvào năm
2020?"|Context:"ÔngNguyễnVănA...";"BàTrầnThịB..."]

Mô hình sau đó sinh ra câu trả lời như: "Ông Nguyễn Văn A là hiệu trưởng Trường Đại học ABC vào năm 2020."

1.3.5. Vai trò trong đề tài

Kỹ thuật RAG đóng vai trò là thành phần trung tâm trong hệ thống chatbot thủ thư, chịu trách nhiệm tạo ra câu trả lời chính xác và có dẫn chứng cho các câu hỏi của sinh viên. Mô hình sẽ tìm kiếm và trích xuất thông tin phù hợp từ cơ sở dữ liệu lời mở đầu của sách đã được xử lý từ PDF, sau đó sử dụng mô hình ngôn ngữ lớn (LLM) để tổng hợp thông tin thành câu trả lời hoàn chỉnh, rõ ràng và dễ hiểu.

Kỹ thuật RAG được lựa chọn vì có khả năng kết hợp giữa tìm kiếm thông tin và sinh ngôn ngữ tự nhiên, giúp tạo ra câu trả lời chính xác, có dẫn chứng từ cơ sở dữ liệu lời mở đầu sách. RAG cũng hỗ trợ tiếng Việt và phù hợp với mục tiêu xây dựng chatbot thư viện chuyên sâu, hiệu quả.

1.4. MÔ HÌNH SENTECETRANSFORMER

1.4.1. Ý tưởng, hướng tiếp cận

Sentence Transformers là một thư viện Python được xây dựng dựa trên PyTorch và thư viện Hugging Face Transformers. Mục đích chính của nó là đơn giản hóa việc tính toán sentence embeddings (vector biểu diễn cho toàn bộ câu hoặc đoạn văn). Thay vì chỉ lấy vector từ token cuối cùng của một mô hình (như cách đôi khi được thực hiện

với các mô hình chỉ dựa trên token), Sentence Transformers sử dụng các kỹ thuật tổng hợp (pooling) trên đầu ra của các mô hình transformer để tạo ra một vector cố định duy nhất cho toàn bộ đơn vị văn bản (câu, đoạn).

SentenceTransformer được giới thiệu lần đầu trong bài báo "*Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*" (Reimers & Gurevych, 2019) [15]. Mục tiêu của mô hình là tạo ra vector biểu diễn cho câu (sentence embeddings) giúp so sánh ngữ nghĩa giữa các câu một cách hiệu quả.

1.4.2. Cách hoạt động

- Sử dụng một mô hình transformer cơ bản (thường là BERT, RoBERTa, MPNet, MiniLM, ...)
- Văn bản đầu vào được đưa qua mô hình transformer này để lấy ra vector biểu diễn cho từng token.
- Thay vì chỉ lấy vector của token đặc biệt [CLS] hoặc [SEP], Sentence Transformers áp dụng một lớp pooling (tổng hợp), thường là Mean Pooling (tính trung bình cộng các vector token), để tạo ra một vector embedding duy nhất có kích thước cố định cho toàn bộ câu hoặc đoạn.
- Vector embedding này có thể được sử dụng cho các tác vụ so sánh ngữ nghĩa (sử dụng cosine similarity), phân cụm, phân loại, ...

1.4.3. Các mô hình dựa trên SentenceTransformer

- all-MiniLM-L6-v2: Một trong những mô hình mặc định và được sử dụng rộng rãi nhất. Nó nhỏ gọn, nhanh và cung cấp hiệu suất cân bằng tốt trên nhiều tác vụ tiếng Anh. Kích thước embedding là 384 chiều.
- paraphrase-mpnet-base-v2: Thường cung cấp hiệu suất tốt hơn all-MiniLM-L6-v2, nhưng lớn hơn một chút. Kích thước embedding 768 chiều. Được huấn luyện đặc biệt để nhận diện các câu có cùng ngữ nghĩa (paraphrases).
- multi-qa-MiniLM-L6-cos-v1: Được tối ưu hóa cho các tác vụ Hỏi-Đáp (Question-Answering) và tìm kiếm ngữ nghĩa.
- multi-qa-mpnet-base-dot-v1: Phiên bản lớn hơn của mô hình trên, thường cho hiệu suất tốt hơn trên tác vụ QA.

1.4.4. Mô phỏng

Giả sử ta có câu: "Tôi thích học môn trí tuệ nhân tạo".

SentenceTransformer sẽ mã hóa câu này thành một vector gồm 384 giá trị thực (float), mỗi giá trị biểu diễn một đặc trưng ngữ nghĩa tiềm ẩn của câu. Ví dụ, vector kết quả có thể có dạng: [-0.12, 0.31, 0.08, ..., 0.45]

Đây là biểu diễn số học cho nội dung của câu, cho phép sử dụng trong các tác vụ như tìm kiếm ngữ nghĩa (semantic search), phân cụm (clustering), hoặc phân loại văn bản (text classification). Câu càng giống nhau về ý nghĩa, thì các vector tương ứng sẽ càng gần nhau trong không gian vector.

1.4.5. Kết quả thử nghiệm đã có

Mô hình Sentence-BERT (SBERT) đã được đề xuất bởi Reimers và Gurevych (2019) như một giải pháp cải tiến cho bài toán biểu diễn câu (sentence embedding) phục vụ các tác vụ so sánh ngữ nghĩa, đặc biệt là Semantic Textual Similarity (STS). Trong các thử nghiệm được trình bày trong bài báo gốc “*Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*” [15], nhóm tác giả đã đánh giá SBERT trên một loạt các tập dữ liệu chuẩn cho bài toán STS như STS-Benchmark, SICK-Relatedness, và STS 2012–2016.

Kết quả thực nghiệm cho thấy SBERT vượt trội so với các phương pháp embedding truyền thống. Cụ thể, trên tập STS Benchmark, SBERT đạt hệ số tương quan Spearman lên đến 0.876, cao hơn so với InferSent - GloVe (0.855) và Universal Sentence Encoder (0.851). Ngoài ra, nhờ kiến trúc Siamese hoặc Triplet Network, SBERT cho phép so sánh hàng triệu cặp câu một cách hiệu quả về mặt tính toán, trong khi BERT gốc không thể thực hiện điều này do chi phí tính toán quá cao khi encode cả hai câu cùng lúc.

Hiệu suất cao và khả năng tính toán song song hiệu quả giúp SBERT trở thành một lựa chọn lý tưởng trong các hệ thống tìm kiếm ngữ nghĩa, phân cụm văn bản, hoặc truy xuất thông tin thông minh dựa trên ngữ nghĩa. Với ứng dụng vào hệ thống thư viện thông minh, SBERT có thể được dùng để so sánh văn bản truy vấn của người dùng với phần nội dung sách đã được trích xuất, từ đó phục vụ chức năng gợi ý sách hoặc trả lời câu hỏi dựa trên dữ liệu đã lưu.

1.4.6. Vai trò trong đề tài

Trong đề tài, SentenceTransformer được sử dụng để chuyển các đoạn văn bản của lời mở đầu sách thành vector để lưu vào ChromaDB. Việc này giúp dễ dàng truy vấn ngữ nghĩa (semantic search) khi tích hợp với RAG.

1.4.7. Mức độ phổ biến

- GitHub: Thư viện sentence-transformers trên GitHub có hơn 13.000 sao.
- Ứng dụng thực tế & hệ thống sử dụng: SentenceTransformer được dùng rộng rãi trong các hệ thống tìm kiếm ngữ nghĩa, chatbot, phân cụm câu hỏi, khuyến nghị văn bản, và phân loại ngữ nghĩa.
- Mức độ phổ biến trong sách báo (Google Books Ngram): Từ khóa “Sentence-BERT” hoặc “SentenceTransformer” chưa phổ biến trong sách xuất bản, do đây là công nghệ ra đời từ năm 2019 và mang tính chuyên sâu, nhưng xuất hiện rất nhiều trong các blog kỹ thuật, bài báo học thuật, và tài liệu hướng dẫn trực tuyến.

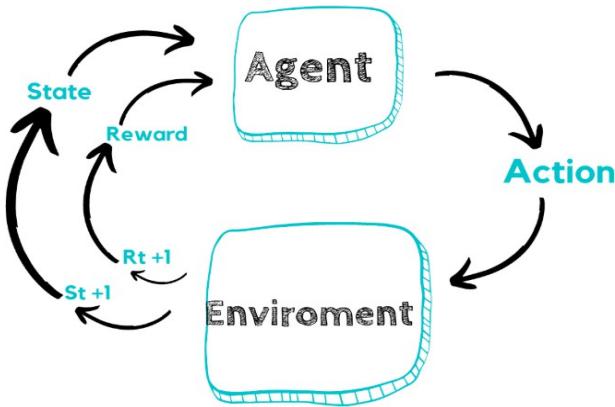
1.5. HỌC TĂNG CƯỜNG

1.5.1. Ý tưởng, hướng tiếp cận

Học tăng cường – Reinforcement Learning là một hướng tiếp cận trong học máy dựa trên quá trình tác nhân (agent) tự học cách hành động trong một môi trường sao cho tối ưu hóa phần thưởng tích lũy theo thời gian. Thuật toán học tăng cường đầu tiên được trình bày một cách có hệ thống bởi Christopher Watkins trong luận án tiến sĩ của ông năm 1989 tại Đại học Cambridge với tiêu đề “*Learning from Delayed Rewards*” [16]. Tuy nhiên, ý tưởng nền tảng của học tăng cường đã được nhắc đến trước đó trong lĩnh vực điều khiển tối ưu (optimal control) và lý thuyết trò chơi.

Học tăng cường là một kỹ thuật học máy, mà agent tự khám phá, tìm hiểu, thăm dò hành vi của mình thông qua tương tác với môi trường. Agent đưa ra quyết định về một hành động sẽ thực hiện dựa trên nguyên tắc học từ phản hồi và phần thưởng để tối đa hóa một hàm phần thưởng được xác định trước.

Học tăng cường hoạt động dựa trên cơ chế khen thưởng hoặc phạt để củng cố các hành vi tốt và hạn chế hành vi xấu. Mục đích chính của nó là xây dựng một chiến lược hành động tối ưu nhằm đạt được tổng phần thưởng cao nhất có thể. Trong phương pháp này, hệ thống tự động học cách đưa ra các quyết định liên tiếp để tối ưu hóa lợi ích mà không cần con người chỉ dẫn chi tiết hay lập trình sẵn các bước thực hiện.



Hình 9: Cơ chế hoạt động của Học Tăng Cường

Các thuật ngữ của học tăng cường:

- Tác nhân (agent) – Thực thể chịu trách nhiệm đưa ra quyết định và là trung tâm của quá trình học.
- Môi trường (environment) – Không gian mà tác nhân tương tác, bao gồm các điều kiện và quy tắc ảnh hưởng đến quá trình học.
- Hành động (action) – Tập hợp các thao tác mà tác nhân có thể thực hiện để tác động đến môi trường.
- Trạng thái (state) – Tình huống hiện tại của tác nhân trong môi trường tại một thời điểm cụ thể.
- Phản thưởng (reward) – Phản hồi từ môi trường sau mỗi hành động của tác nhân, thường là một giá trị số thể hiện mức độ thành công.
- Chính sách (policy) – Chiến lược mà tác nhân sử dụng để chọn hành động nhằm tối đa hóa lợi ích dài hạn.
- Hàm giá trị (value function) – Ước lượng mức độ "tốt" của một trạng thái hoặc hành động, thường dựa trên phản thưởng kỳ vọng trong tương lai.
- Mô hình (model) – Biểu diễn của môi trường giúp tác nhân dự đoán kết quả của các hành động, có thể là mô phỏng hoặc dựa trên dữ liệu thực tế.

Các nền tảng lý thuyết của học tăng cường bao gồm:

- Chuỗi quyết định Markov (Markov Decision Processes – MDPs): Mô hình hóa môi trường mà trong đó hành động hiện tại chỉ phụ thuộc vào trạng thái hiện tại (không phụ thuộc vào lịch sử trước đó).

- Lý thuyết điều khiển động (Dynamic Programming): Được phát triển bởi Richard Bellman vào thập niên 1950, cung cấp công cụ toán học để xử lý tối ưu hóa theo từng bước thời gian.
- Học thử - sai (Trial-and-error learning): Được kế thừa từ tâm lý học hành vi (behaviorist psychology), đặc biệt từ công trình của B.F. Skinner.

1.5.2. Công thức

Học tăng cường vận hành dựa trên một mô hình toán học có tên là MDP – Markov Decision Process, được định nghĩa bởi 5 thành phần:

$$\mathcal{M} = (S, A, P, R, \gamma)$$

Trong đó:

- S: Tập hợp tất cả các trạng thái có thể xảy ra
- A: Tập hợp các hành động mà tác nhân có thể thực hiện.
- $P(s'|s, a)$: Xác suất chuyển trạng thái, tức là xác suất chuyển từ trạng thái s sang s' khi thực hiện hành động a.
- $R(s, a)$: Hàm phần thưởng kỳ vọng, cho biết phần thưởng nhận được khi thực hiện hành động a tại trạng thái s.
- $\gamma \in [0, 1]$: Hệ số chiết khấu, thể hiện mức độ ưu tiên phần thưởng tương lai.

1.5.2.1. Policy

Chính sách - Policy là chiến lược chọn hành động của tác nhân tại mỗi trạng thái

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$$

Trong đó:

- $\pi(a | s)$: Xác suất chọn hành động a khi ở trạng thái s
- S_t : Trạng thái tại thời điểm t
- A_t : Hành động tại thời điểm t

Nếu chính sách là xác định (deterministic), ta có:

$$\pi(s) = a$$

1.5.2.2. Hàm giá trị trạng thái

Hàm giá trị trạng thái đo lường phần thưởng kỳ vọng tích lũy nếu bắt đầu từ trạng thái s , sau đó tuân theo chính sách π .

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s \right]$$

Trong đó:

- $V^\pi(s) \in \mathbb{R}$
- $\gamma \in [0,1]$: Hệ số chiết khấu, điều chỉnh mức độ ưu tiên phần thưởng tương lai
- r_t : Phần thưởng tại thời điểm t
- \mathbb{E}_π : Kỳ vọng theo chính sách π

1.5.2.3. Hàm giá trị hành động

Hàm giá trị hành động ký hiệu là $Q^\pi(s,a)$, đo lường phần thưởng tích lũy kỳ vọng nếu tác nhân ở trạng thái s , thực hiện hành động a , rồi tiếp tục theo chính sách π .

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s, A_0 = a \right]$$

Về kích thước, nếu $|S| = n, |A| = m \rightarrow Q^\pi(s, a) \in \mathbb{R}^{n \times m}$

1.5.2.4. Phương trình Bellman

(a) Cho hàm giá trị trạng thái $V^\pi(s)$:

$$V^\pi(s) = \sum_{a \in A} \pi(a|s) \left[R(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^\pi(s') \right]$$

Trong đó:

- $R(s, a)$: Phần thưởng kỳ vọng khi thực hiện hành động a tại trạng thái s .
- $P(s' \mid s, a)$: Xác suất chuyển từ trạng thái s đến s' sau khi thực hiện hành động a .
- Tổng trong tổng: đi qua mọi hành động có thể, và mọi trạng thái tiếp theo.

(b) Cho hàm giá trị hành động $Q^\pi(s, a)$:

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \sum_{a'} \pi(a'|s') Q^\pi(s', a')$$

1.5.2.5. Hàm tối ưu

(a) Trạng thái:

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

(b) Hành động:

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a')$$

1.5.2.6. Chính sách tối ưu

Chính sách tối ưu π^* đạt được giá trị cao nhất tại mọi trạng thái:

$$\pi^*(s) = \arg \max_a Q^*(s, a)$$

1.5.2.7. Tổng phần thưởng tích lũy

Đây là mục tiêu tối đa hóa của học tăng cường

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

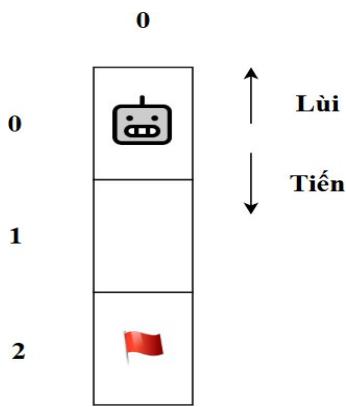
Trong đó:

- γ : Hệ số chiết khấu, cho biết mức độ ưu tiên giữa phần thưởng hiện tại và phần thưởng tương lai.
- k : Số bước trong tương lai, tính từ thời điểm t .
- R_{t+k+1} : Phần thưởng nhận được tại bước $t+k+1$. Đây là phần thưởng mà agent nhận được khi thực hiện hành động trong môi trường.

Kích thước: $G_t \in \mathbb{R}$, là một đại lượng vô hướng đại diện cho tổng phần thưởng từ thời điểm t .

1.5.3. Mô phỏng

Giả sử ta cho tác nhân di chuyển trên một lưới kích thước 3×1 , tác nhân có thể di chuyển lên hoặc xuống tùy ý, lúc đầu tác nhân ở vị trí 0, mục tiêu là đến được vị trí cuối cùng ở vị trí 2.



Hình 10: Mô hình quá trình học tăng cường

Thông tin môi trường:

- Tập trạng thái: $S = \{0,1,2\}$
- Tác nhân bắt đầu tại $s = 0$, mục tiêu là tới $s = 2$
- Tập hành động: $A = \{\text{Tiến}, \text{Lùi}\}$
- Reward:
 - o $R(2) = +1$ khi tới đích
 - o $R(s \neq 2) = 0$
- Hệ số chiết khấu: $\gamma = 0.9$
- Chính sách ban đầu: chọn hành động ngẫu nhiên với xác suất đều.

Các bước thực hiện

Tính giá trị theo Value Iteration:

1. Khởi tạo $V(s) = 0$ với mọi s
2. Cập nhật theo Bellman:

$$V(s) \leftarrow \max_{a \in A} \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a) V(s') \right]$$

3. Giả sử xác suất chuyển là chắc chắn (deterministic), sau vài bước cập nhật:

$$V(2) = 1 \quad V(1) = \gamma \cdot V(2) = 0.9 \quad V(0) = \gamma \cdot V(1) = 0.81$$

4. Chính sách tối ưu là: luôn chọn **Tiến** từ $0 \rightarrow 1 \rightarrow 2$.

1.6. PHƯƠNG PHÁP Q-LEARNING

1.6.1. Ý tưởng, hướng tiếp cận

Q-Learning được đề xuất lần đầu tiên bởi Christopher Watkins trong luận án tiến sĩ năm 1989 tại Đại học Cambridge, với tiêu đề “Learning from Delayed Rewards” [16] với người hướng dẫn là Peter Dayan.

Thuật toán Q-Learning dựa trên nền tảng lý thuyết của quy hoạch động, đặc biệt là phương trình Bellman. Không giống như các phương pháp truyền thống yêu cầu mô hình xác suất chuyển trạng thái (transition model), Q-Learning học trực tiếp giá trị hành động (action-value function) thông qua tương tác với môi trường.

1.6.2. Công thức

1.6.2.1. Hàm Q và mục tiêu

Hàm Q (hàm giá trị hành động) được định nghĩa là:

$$Q(s, a) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, a_0 = a \right]$$

Trong đó:

- s : trạng thái hiện tại, thuộc không gian trạng thái S
- a : hành động tại trạng thái s , thuộc không gian hành động A
- r_t : phần thưởng nhận được tại bước t
- $\gamma \in [0,1]$: hệ số chiết khấu, thể hiện mức độ ưu tiên hiện tại so với tương lai
- $Q(s,a)$: giá trị kỳ vọng phần thưởng tích lũy khi thực hiện hành động a tại trạng thái s

1.6.2.2. Cập nhật Q theo phương trình Bellman

Thuật toán Q-Learning cập nhật giá trị Q theo công thức sau:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right]$$

Trong đó:

- s_t : trạng thái hiện tại
- a_t : hành động được chọn tại s_t
- r_{t+1} : phần thưởng nhận được sau khi thực hiện a_t

- s_{t+1} : trạng thái mới sau khi thực hiện hành động
- $\alpha \in (0,1]$: tốc độ học
- $\max_a Q(s_{t+1}, a')$: ước lượng giá trị hành động tốt nhất tại trạng thái mới

Kích thước của Q là $|S| \times |A|$, tức là ma trận với hàng là các trạng thái, cột là các hành động.

1.6.3. Siêu tham số

Bảng 28: Siêu tham số của Q-Learning

Hyperparameter	Khoảng giá trị gợi ý	Ý nghĩa
γ	0.90 - 0.99	Hệ số chiết khấu phần thưởng tương lai
θ	$1e-3 - 1e-6$	Ngưỡng sai số hội tụ trong thuật toán Value Iteration/Policy Iteration
ϵ	0.01 - 0.2 (nếu dùng)	Xác suất hành động ngẫu nhiên trong khám phá (exploration)
Số vòng lặp N	Tùy thuộc độ lớn không gian trạng thái	Số lần lặp tối đa cho đến khi hội tụ

1.6.4. Mô phỏng

Giả sử môi trường là một lưới 2×2 , mỗi ô là một trạng thái. Mỗi hành động đưa agent sang trạng thái khác (nếu hợp lệ). Mục tiêu là đến được ô $(1,1)$, nơi agent nhận phần thưởng +1, các trạng thái khác thưởng bằng 0.

Thông tin môi trường:

- Trạng thái: $S = \{(0,0), (0,1), (1,0), (1,1)\}$
- Hành động: $A = \{\text{lên, xuống, phải, trái}\}$
- Phần thưởng:
 - $R(s = (1,1)) = +1$ nếu đến $(1,1)$
 - $R(s \neq (1,1)) = 0$

- Trạng thái (1,1) là trạng thái kết thúc (terminal).

Thiết lập ban đầu:

- Khởi tạo $Q(s,a) = 0$ với mọi trạng thái và hành động.
- Hệ số học $\alpha = 0.5$.
- Hệ số chiết khấu $\gamma = 1.0$.
- Chính sách ϵ -greedy (không nêu chi tiết ở đây vì chọn hành động cố định để minh họa).

Các bước thực hiện:

1. Giả sử agent thực hiện chuỗi hành động sau:

- (0,0) → phải → (0,1) — nhận thưởng $r = 0$
- (0,1) → xuống → (1,1) — nhận thưởng $r = +1$

Cập nhật 1: tại trạng thái (0,1), hành động “xuống”

$$Q((0, 1), \text{xuống}) \leftarrow 0 + 0.5 \left[1 + 1 \cdot \max_{a'} Q((1, 1), a') - 0 \right]$$

Vì (1,1) là trạng thái kết thúc nên $\max_{a'} Q((1, 1), a') = 0$

$$Q((0, 1), \text{xuống}) \leftarrow 0.5 \cdot (1 + 0) = 0.5$$

Cập nhật 2: tại trạng thái (0,0), hành động “phải”

$$Q((0, 0), \text{phải}) \leftarrow 0 + 0.5 \left[0 + 1 \cdot \max_{a'} Q((0, 1), a') - 0 \right]$$

$$Q((0, 1), \text{xuống}) = 0.5 \Rightarrow \max_{a'} Q((0, 1), a') = 0.5$$

$$Q((0, 0), \text{phải}) \leftarrow 0.5 \cdot 0.5 = 0.25$$

Bảng 29: Trạng thái agent khi thực hiện các chuỗi hành động

Trạng thái	Hành động	Q-Value
(0,0)	phải	0.25
(0,1)	xuống	0.5
Các cặp khác		0.0

2. Thực hiện lặp lại thêm 1 vòng nữa

Hành động 1: $(0,0) \rightarrow$ phải $\rightarrow (0,1)$ ($r = 0$)

$$\max_{a'} Q((0,1), a') = 0.5$$

$$Q((0,0), \text{phải}) \leftarrow 0.25 + 0.5 \cdot (0 + 1 \cdot 0.5 - 0.25) = 0.25 + 0.125 = 0.375$$

Hành động 2: $(0,1) \rightarrow$ xuống $\rightarrow (1,1)$ ($r = 1$)

$$\max_{a'} Q((1,1), a') = 0$$

$$Q((0,1), \text{xuống}) \leftarrow 0.5 + 0.5 \cdot (1 - 0.5) = 0.5 + 0.25 = 0.75$$

Bảng Q sau 2 vòng cập nhật:

Bảng 30: Trạng thái agent khi thực hiện lặp lại thêm 1 vòng

Trạng thái	Hành động	Q-Value
$(0,0)$	phải	0.375
$(0,1)$	xuống	0.75

Sau nhiều vòng lặp, giá trị Q sẽ hội tụ về gần 1 cho lộ trình tối ưu đến $(1,1)$.

1.6.5. Kết quả thử nghiệm đã có

Phương pháp Q-Learning đạt hiệu quả cao trong các môi trường game giả lập của OpenAI Gym như: Taxi-v3, FrozenLake-v1, CliffWalking-v0, CartPole-v0,...

Trong thực tế Q-Learning cũng đạt hiệu quả trong ứng dụng xe điện lai khi giảm tiêu thụ nhiên liệu 16% so với cách truyền thống, và tốc độ hội tụ nhanh hơn với warm-start theo bài báo “Learning Time Reduction Using Warm Start Methods for a Reinforcement Learning Based Supervisory Control in Hybrid Electric Vehicle Applications” của tác giả Tiancheng Shen et al. (2020) [17].

1.7. PHƯƠNG PHÁP DEEP Q-LEARNING

1.7.1. Ý tưởng, hướng tiếp cận

Deep Q-Learning là một phương pháp học tăng cường được phát triển nhằm giải quyết hạn chế của thuật toán Q-Learning truyền thống trong các môi trường có không gian trạng thái lớn hoặc liên tục.

Deep Q-Learning là sự kết hợp giữa thuật toán Q-Learning truyền thống và mạng nơ-ron sâu. Thuật toán này lần đầu tiên được giới thiệu trong bài báo "Playing

"Atari with Deep Reinforcement Learning" của Volodymyr Mnih và cộng sự vào năm 2013.

Deep Q-Learning sử dụng một mạng nơ-ron sâu, gọi là Deep Q-Network, để xác định hàm giá trị hành động (Q-function), cho phép tác nhân học chính sách tối ưu từ dữ liệu đầu vào có chiều cao như hình ảnh.

Mạng nơ-ron được sử dụng trong Deep Q-Learning bao gồm:

- Input layer: Nhận đầu vào là trạng thái s.
- Hidden layers: Tích hợp các lớp ẩn để học các đặc trưng từ trạng thái.
- Output layer: Cung cấp giá trị $Q(s,a)$ cho mỗi hành động a.

Deep Q-Learning có thể kết hợp các kỹ thuật như:

- Replay buffer: Lưu trữ kinh nghiệm từ các lần chơi trước, giúp giảm sự phụ thuộc vào chuỗi dữ liệu gần nhất.
- Target network: Sử dụng một mạng nơ-ron mục tiêu tạm thời để làm giảm sự không ổn định trong quá trình huấn luyện.

1.7.2. Công thức

1.7.2.1. Hàm mất mát

$$L(\theta) = \mathbb{E}_{(s,a,r,s') \sim D} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]$$

Trong đó:

- θ : Tham số của mạng Q hiện tại.
- θ^- : Tham số của mạng Q mục tiêu (target network), được cập nhật định kỳ từ θ .
- D: Bộ dữ liệu kinh nghiệm (replay buffer).
- s,a,r,s': Trạng thái hiện tại, hành động, phần thưởng và trạng thái tiếp theo.
- γ : Hỗn số chiết khấu.

1.7.2.2. Cập nhật tham số

Tham số θ được cập nhật bằng cách giảm thiểu hàm mất mát $L(\theta)$ thông qua thuật toán gradient descent.

1.7.3. Siêu tham số

Bảng 31: Siêu tham số Deep Q-Learning

Hyperparameter	Ký hiệu	Khoảng giá trị đề xuất
Hệ số chiết khấu	γ	0.95 - 0.99
Tốc độ học	α	0.00025 – 0.001
Kích thước batch	-	32 - 64
Kích thước replay buffer	-	1000 - 1,000,000
Epsilon (khám phá)	ϵ	Giảm từ 1.0 đến 0.1
Tần suất cập nhật target	-	1000 - 10,000 bước

1.7.4. Mô phỏng

Môi trường: CartPole - trong môi trường này, một chiếc xe đẩy có thể di chuyển qua lại trên một đường ray ngang, và trên xe có một cây cột được gắn vào khớp bản lề. Mục tiêu của agent là giữ cho cây cối luôn ở trạng thái cân bằng (góc nghiêng không vượt quá ngưỡng cho phép) và đồng thời không để xe di chuyển ra khỏi giới hạn của đường ray. Agent có thể thực hiện các hành động đẩy xe sang trái hoặc phải. Môi trường cung cấp các thông tin quan sát như vị trí xe, vận tốc xe, góc nghiêng của cột và vận tốc góc của cột.

Thiết lập:

- Trạng thái: Vector 4 chiều (vị trí xe, vận tốc xe, góc cột, tốc độ góc).
- Hành động: 0 (trái), 1 (phải).
- Phần thưởng: +1 cho mỗi bước giữ cột thẳng bằng.

Mạng Q:

- Kiến trúc: Mạng nơ-ron với 2 lớp ẩn, mỗi lớp 24 nơ-ron, hàm kích hoạt ReLU.
- Đầu ra: 2 giá trị Q tương ứng với 2 hành động.

Quy trình huấn luyện:

1. Khởi tạo replay buffer và mạng Q với tham số θ .
2. Tại mỗi bước:
 - Chọn hành động a dựa trên chính sách ϵ -greedy.
 - Thực hiện hành động a , nhận phần thưởng r và trạng thái mới s' .
 - Lưu (s, a, r, s') vào replay buffer.
 - Lấy một minibatch từ replay buffer và cập nhật θ bằng cách giảm thiểu $L(\theta)$.
3. Cập nhật mạng Q mục tiêu θ^- sau mỗi 1000 bước.

1.7.5. Kết quả thử nghiệm đã có

Nghiên cứu gốc: "*Human-level control through deep reinforcement learning*" (Nature, 2015) [18]. Trong nghiên cứu này, thuật toán Deep Q-Learning được áp dụng cho 49 trò chơi trên hệ máy Atari 2600. Kết quả cho thấy Deep Q-Learning đạt hiệu suất ngang bằng hoặc vượt qua người chơi chuyên nghiệp trong nhiều trò chơi.

Bảng 32: Kết quả thử nghiệm đã có của Deep Q-Learning

Trò chơi	(Deep Q-Learning/Người) %
Breakout	1324%
Video Pinball	426%
Space Invaders	197%
Pong	118%
Q*bert	89%
Seaquest	77%
Montezuma's Revenge	0%

Các trò chơi trên không phải là bản gốc do hãng Atari phát hành mà là các môi trường mô phỏng được xây dựng lại trong thư viện Arcade Learning Environment (ALE) và thường được tích hợp qua OpenAI Gym hoặc Gymnasium. Mặc dù tên gọi và cách chơi tương tự trò chơi gốc, nhưng độ khó, quy tắc, cũng như thiết kế màn chơi hoàn toàn phụ thuộc vào cách mà môi trường mô phỏng được xây dựng.

Thuật toán Deep Q-Learning cho thấy khả năng vượt trội ở những trò chơi như Breakout và Video Pinball, tuy nhiên lại gặp nhiều khó khăn trong các trò chơi đòi hỏi lập kế hoạch dài hạn và khám phá phức tạp, như Montezuma's Revenge.

1.7.6. Vai trò trong đế tài

- Vai trò: Xây dựng chức năng gợi ý sách cho người dùng
- Lý do chọn: Deep Q-Learning cho phép hệ thống học từ dữ liệu hành vi của người dùng theo thời gian, thay vì chỉ dựa vào dữ liệu tĩnh như trong các phương pháp học có giám sát truyền thống. Bên cạnh đó, Với khả năng học từ trạng thái đầu vào phức tạp như embedding của người dùng, embedding sách và thời gian tương tác, Deep Q-Learning sử dụng mạng nơ-ron sâu để xử lý hiệu quả các đặc trưng phi

tuyến, vốn là một điểm yếu của các phương pháp Q-learning truyền thống dựa trên bảng tra cứu. Cuối cùng, so với các kỹ thuật gợi ý phổ biến như collaborative filtering hay content-based filtering, Deep Q-Learning có ưu điểm vượt trội trong việc tối ưu hóa phần thưởng dài hạn và tự học chính sách gợi ý tốt nhất thông qua dữ liệu lịch sử.

1.8. CÔNG CỤ LLAMA.CPP

1.8.1. Giới thiệu

Llama.cpp là một thư viện mã nguồn mở được phát triển bởi Georgi Gerganov, nhằm mục tiêu cho phép chạy các mô hình LLaMA (Meta AI) trên CPU một cách hiệu quả, không cần GPU. Công cụ này biên dịch mô hình xuống định dạng nhẹ GGUF (trước đây là GGML) để tiết kiệm bộ nhớ và tối ưu hóa quá trình suy luận (inference). Nó hỗ trợ các mô hình LLaMA 1, LLaMA 2, LLaMA 3 và nhiều biến thể như Alpaca, Vicuna, Mistral, Qwen... Công cụ này đặc biệt phù hợp với các hệ thống tài nguyên thấp, như laptop cá nhân hoặc máy chủ không có GPU [19].

1.8.2. Tính tương thích

- Hệ điều hành hỗ trợ: Windows, Linux, macOS (hỗ trợ tốt cả ARM như M1/M2 trên Mac).
- Môi trường chạy: Chạy trên môi trường C++ hoặc gọi từ Python thông qua thư viện llama-cpp-python.
- Input đầu vào:
 - o Mô hình ngôn ngữ định dạng .gguf: Đây là định dạng nén và tối ưu hóa được Llama.cpp sử dụng, giúp giảm kích thước mô hình và tăng tốc độ xử lý trên CPU. Các mô hình này thường được chuyển đổi từ các mô hình gốc như LLaMA, Mistral, Qwen thông qua công cụ chuyển đổi chuyên dụng.
 - o Prompt đầu vào dạng văn bản thuần (plain text): Là chuỗi câu hỏi hoặc yêu cầu từ người dùng mà mô hình sẽ xử lý để tạo ra phản hồi.
 - o Tùy chọn cấu hình (nếu cần): Người dùng có thể điều chỉnh các thông số như số lượng luồng CPU (n_threads), số token sinh ra tối đa (max_tokens), nhiệt độ (temperature), hoặc context window (ctx_size) để kiểm soát quá trình sinh văn bản.
- Công cụ/phụ thuộc cần cài thêm: C++ compiler, thư viện llama-cpp-python để dùng trong các framework như LangChain.

1.8.3. Mức độ phổ biến

- GitHub: llama.cpp có khoảng 80.000 sao trên Github.
- Hệ thống lớn sử dụng: được tích hợp trong nhiều framework như Ollama, LM Studio, LangChain, llama-index.
- Mức độ phổ biến trong sách báo: Từ khóa "Llama.cpp" không xuất hiện nhiều trên Google Books Ngram vì đây là công cụ mới, chủ yếu phổ biến qua GitHub và cộng đồng.

1.9. CÔNG CỤ DJANGO

1.9.1. Giới thiệu

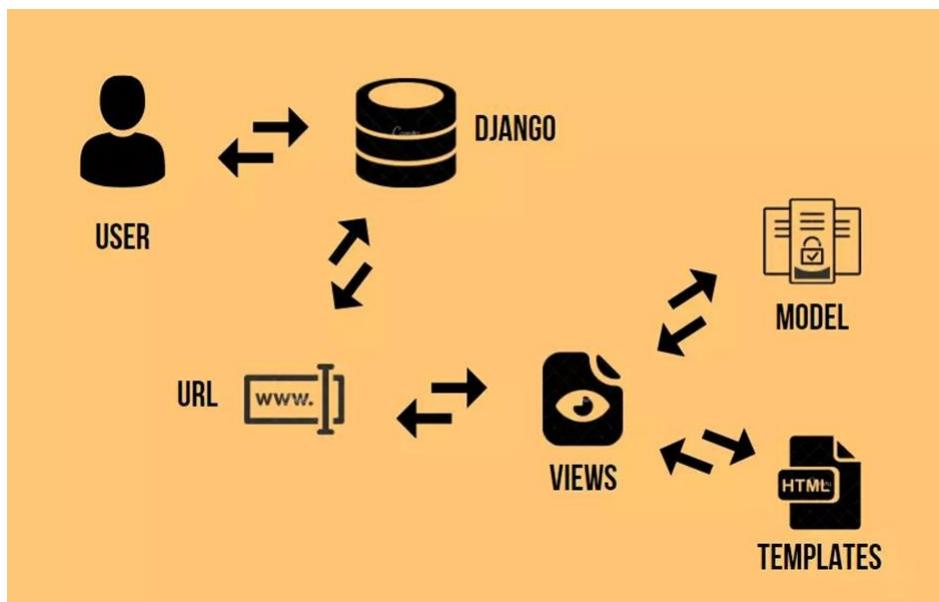


Hình 11: Công cụ Django

Django là một framework mã nguồn mở dùng để phát triển web, được xây dựng trên ngôn ngữ lập trình Python. Ra đời từ năm 2005, Django đã nhanh chóng trở nên phổ biến và được đông đảo cộng đồng lập trình viên cũng như doanh nghiệp tin dùng trong việc xây dựng các ứng dụng web.

Mục tiêu chính khi phát triển Django là giảm thiểu độ phức tạp và rút ngắn thời gian phát triển ứng dụng. Framework này cung cấp sẵn nhiều thành phần như hệ thống định tuyến, cơ sở dữ liệu ORM, và cơ chế xác thực, giúp lập trình viên không phải xây dựng lại từ đầu những chức năng phổ biến. Thay vào đó, họ có thể tập trung vào thiết kế logic nghiệp vụ và phát triển tính năng đặc thù cho sản phẩm của mình [20].

1.9.2. Cách hoạt động của Django



Hình 12: Cách hoạt động của Django

Django vận hành dựa trên mô hình kiến trúc MVT (Model – View – Template), một biến thể của kiến trúc MVC truyền thống, nhằm phân tách rõ ràng giữa dữ liệu, xử lý và giao diện. Cụ thể:

- Model (Mô hình dữ liệu): Đại diện cho cấu trúc và logic lưu trữ dữ liệu. Đây là nơi định nghĩa các bảng cơ sở dữ liệu dưới dạng các lớp Python, giúp quản lý và truy vấn dữ liệu một cách trực quan và hiệu quả.
- View (Xử lý logic): Đóng vai trò tiếp nhận và xử lý các yêu cầu từ người dùng (HTTP request). Tại đây, hệ thống có thể tương tác với Model để lấy dữ liệu, sau đó gửi dữ liệu đó tới Template để hiển thị.
- Template (Giao diện người dùng): Là các tệp HTML có thể kết hợp với cú pháp đặc biệt để hiển thị dữ liệu động. Template chịu trách nhiệm định dạng và trình bày nội dung mà người dùng nhìn thấy trên trình duyệt.

1.9.3. Tính tương thích

- Hệ điều hành tương thích: Windows, macOS, Linux.
- Môi trường: Python ≥ 3.8 , chạy tốt trong môi trường virtualenv, Docker, hoặc hệ thống cloud (AWS, Azure, GCP).
- Yêu cầu đầu vào: Cần cài đặt Python, pip và hệ quản trị CSDL (SQLite, PostgreSQL, MySQL, SQL Server, v.v.).
- Cách cài đặt: pip install django.

1.9.4. Mức độ phổ biến

- GitHub: Django có khoảng 83.000 sao trên Github.
- Hệ thống lớn sử dụng: Instagram , Mozilla, Pinterest, ...
- Mức độ phổ biến trong sách báo: tăng dần đều từ năm 2000 đến 2017 và giảm dần từ năm 2018. (theo Google Books Ngram).

1.10. CÔNG CỤ FLASK

1.10.1. Giới thiệu



Hình 13: Công cụ Flask

Flask được tạo ra và phát triển bởi Armin Ronacher khi ông là sinh viên tại Đại học Kỹ thuật Viện Bremen, Đức. Flask được phát hành lần đầu vào năm 2010 và đã trở thành một trong những framework web Python phổ biến nhất từ đó.

Flask là một microframework web được phát triển bằng ngôn ngữ Python, nổi bật với thiết kế tối giản nhưng vẫn đảm bảo hiệu suất và tính mở rộng. Khác với Django – vốn đi kèm với nhiều thành phần tích hợp sẵn – Flask mang lại cho lập trình viên sự chủ động trong việc lựa chọn công cụ và thư viện bên ngoài, từ đó dễ dàng điều chỉnh cấu trúc ứng dụng theo nhu cầu cụ thể. Với triết lý "đơn giản và linh hoạt", Flask đặc biệt phù hợp với những ai mong muốn xây dựng ứng dụng web một cách linh động, tránh sự ràng buộc từ các cấu trúc framework cứng nhắc.

1.10.2. Các tính năng nổi bật của Flask

Flask là một framework phát triển web dựa trên Python, nổi bật nhờ sự tối giản và linh hoạt. Nó cung cấp nhiều tính năng hữu ích giúp các nhà phát triển dễ dàng xây dựng ứng dụng web một cách hiệu quả. Một số điểm nổi bật của Flask có thể kể đến như sau:

- Thiết kế gọn nhẹ, dễ tiếp cận: Flask được xây dựng với mục tiêu đơn giản hóa quy trình phát triển. Với cấu trúc rõ ràng và mã nguồn dễ hiểu, các lập trình viên, kể cả người mới, có thể nhanh chóng làm quen và tùy chỉnh theo yêu cầu của dự án.

- Hệ thống định tuyến linh hoạt: Flask cho phép định nghĩa các tuyến đường (URL routes) và ánh xạ chúng tới các hàm xử lý tương ứng. Điều này giúp kiểm soát luồng xử lý yêu cầu HTTP một cách linh hoạt và hiệu quả.
- Khả năng mở rộng mạnh mẽ: Dù bản chất là một micro-framework, Flask vẫn hỗ trợ tích hợp nhiều tiện ích mở rộng (extensions) để bổ sung các tính năng như xác thực người dùng, kết nối cơ sở dữ liệu, quản lý phiên làm việc, v.v.
- Tích hợp sẵn máy chủ phát triển: Flask cung cấp sẵn một server thử nghiệm giúp lập trình viên kiểm tra và gỡ lỗi ứng dụng trong quá trình phát triển mà không cần cấu hình phức tạp.
- Hỗ trợ xây dựng RESTful API: Với Flask, việc triển khai các dịch vụ web tuân theo chuẩn REST trở nên đơn giản, hỗ trợ xây dựng các hệ thống backend hiện đại.
- Cộng đồng phát triển năng động: Flask được sử dụng rộng rãi và có một cộng đồng lập trình viên lớn luôn sẵn sàng chia sẻ tài liệu, hướng dẫn và hỗ trợ kỹ thuật.

1.10.3. Tính tương thích

- Hệ điều hành tương thích: Windows, macOS, Linux.
- Môi trường: Python ≥ 3.7 , Flask hỗ trợ tích hợp với SQLAlchemy, MongoDB, RESTful API và các công nghệ khác một cách dễ dàng.
- Yêu cầu đầu vào: Cần cài đặt Python và pip.
- Cài đặt: pip install flask.

1.10.4. Mức độ phổ biến

- GitHub: tính đến nay Flask đã thu hút hơn 69.000 sao trên Github.
- Các hệ thống lớn sử dụng: Netflix, Reddit, ...
- Mức độ phổ biến trong sách báo: tăng dần đều từ năm 2012(theo Google Books Ngram).

1.11. CÔNG NGHỆ LANGCHAIN

1.11.1. Giới thiệu



Hình 14: Công cụ LangChain

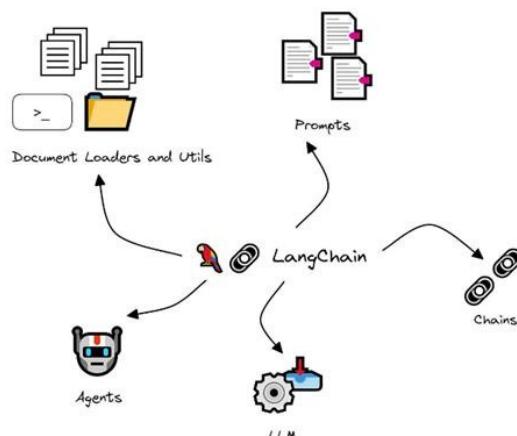
LangChain là một framework được viết bằng Python và JavaScript, nó cung cấp các công cụ để thao tác và xây dựng ứng dụng dựa trên LLMs. LangChain được biết đến là một mã nguồn mở, cho phép các nhà phát triển dễ dàng xây dựng các ứng dụng dựa trên các mô hình ngôn ngữ lớn, có thể kể đến như Bard, GPT-3,...

Theo đó, mã nguồn mở này hoạt động như một cầu nối giữa LLM và các ứng dụng.

Chúng cung cấp các công cụ để:

- Kết nối LLM với các nguồn dữ liệu bên ngoài: Cho phép người dùng truy cập dữ liệu từ các API, cơ sở dữ liệu, tập tin,... và sử dụng dữ liệu này để cung cấp cho LLM.
- Xử lý đầu ra của LLM: Mã nguồn mở này cung cấp các công cụ để phân tích, hiểu và chuyển đổi đầu ra của LLM thành định dạng phù hợp với ứng dụng của bạn.

1.11.2. Các module của Langchain



Hình 15: Các module của Langchain

LangChain là một framework mạnh mẽ hỗ trợ xây dựng ứng dụng sử dụng mô hình ngôn ngữ lớn (LLM) bằng cách chia thành nhiều thành phần chức năng riêng biệt, cho phép kiểm soát tốt hơn trong quá trình phát triển. Một số module nổi bật của LangChain bao gồm:

- **Model I/O:** Đây là phần cốt lõi để giao tiếp với các mô hình ngôn ngữ. Nó cung cấp giao diện chuẩn hóa để gửi yêu cầu và nhận phản hồi từ bất kỳ LLM nào. Việc tạo đầu vào được thực hiện thông qua các Prompt — bao gồm hướng dẫn và dữ liệu cần thiết để mô hình hiểu và phản hồi chính xác. LangChain hỗ trợ cả hai loại mô hình: LLM thuần và Chat Model.

- **Retrieval:** Thành phần này hỗ trợ việc kết hợp thông tin từ cơ sở dữ liệu của người dùng vào quá trình sinh văn bản (generation) thông qua kỹ thuật RAG (Retrieval-Augmented Generation). LangChain cung cấp công cụ như bộ tải tài liệu, bộ biến đổi văn bản, mô hình embedding và các cơ chế tìm kiếm — tất cả được thiết kế để xử lý và lưu trữ dữ liệu dạng vector một cách tối ưu theo ngữ cảnh sử dụng.
- **Chains:** Cho phép tạo ra các quy trình xử lý phức tạp bằng cách liên kết nhiều thành phần hoặc nhiều LLM lại với nhau theo chuỗi. Cách tiếp cận này giúp dễ dàng thiết kế các hệ thống linh hoạt và dễ bảo trì, đặc biệt là khi cần mở rộng quy mô hoặc tích hợp logic tùy chỉnh.
- **Agents:** Module này cung cấp cơ chế ra quyết định động cho ứng dụng, cho phép LLM lựa chọn hành động tiếp theo dựa trên kết quả và bối cảnh hiện tại. LangChain hỗ trợ nhiều loại agent khác nhau, có thể kết hợp với tools, chains hoặc thậm chí là các agent khác để giải quyết các bài toán phức tạp theo hướng tự chủ.
- **Memory:** Đóng vai trò quan trọng trong các ứng dụng hội thoại, module này giúp lưu trữ và truy xuất thông tin từ các lượt tương tác trước đó. Điều này cho phép mô hình duy trì được ngữ cảnh, mang lại trải nghiệm hội thoại mạch lạc và liên tục. LangChain hỗ trợ nhiều dạng memory khác nhau, từ đơn giản đến phức tạp, phù hợp với từng tình huống sử dụng.
- **Callbacks:** Đây là cơ chế ghi nhận và xử lý các sự kiện trong quá trình thực thi, rất hữu ích trong việc theo dõi, log hoặc hiển thị quá trình xử lý theo thời gian thực. LangChain cho phép bạn tùy chỉnh các callback handler, đồng thời cung cấp sẵn một số tùy chọn như StdOutCallbackHandler để dễ dàng quan sát dòng sự kiện đang diễn ra [22].

1.11.3. Mức độ phổ biến

- Github: Tính đến năm 2025, Langchain đã thu hút hơn 70.000 sao trên GitHub, phản ánh sự quan tâm mạnh mẽ từ cộng đồng nhà phát triển.
- Các hệ thống lớn sử dụng: Chatbot và trợ lý ảo của OpenAI, Microsoft, Google,...
- Mức độ phổ biến trong sách báo: Mức độ xuất hiện của từ “LangChain” trong các tài liệu sách hiện còn tương đối thấp (do framework mới xuất hiện từ 2022), tuy nhiên tần suất đang tăng mạnh từ năm 2023 trở đi(theo Google Books Ngram).

1.12. SQL SERVER

1.12.1. Giới thiệu



Hình 16: SQL Server

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ-RDBMS được dùng để lưu trữ và truy xuất dữ liệu theo yêu cầu từ các ứng dụng khác. Thay vì phải viết mã từ đầu để quản lý cơ sở dữ liệu, các nhà phát triển phần mềm thường sử dụng SQL Server để tận dụng các chức năng sẵn có, giúp quá trình phát triển ứng dụng nhanh chóng, tiết kiệm chi phí, đồng thời tăng tính ổn định, bảo mật và khả năng mở rộng.

SQL Server có cả phiên bản mã nguồn mở và mã nguồn đóng, giúp tăng cường tính linh hoạt và khả năng tùy chỉnh trong việc triển khai cơ sở dữ liệu. Tuy nhiên, phiên bản chính của SQL Server chủ yếu là mã nguồn đóng, điều này có nghĩa là mã nguồn đóng này không được công khai và người dùng cần phải tuân theo các điều khoản và điều kiện sử dụng của Microsoft.

Tương tự như các RDBMS khác, trong SQL Server, dữ liệu được lưu trữ dưới dạng các bảng (tables) với các hàng (rows) và cột (columns). SQL Server đảm bảo rằng dữ liệu được lưu trữ hiệu quả và có thể được tìm kiếm, truy xuất nhanh chóng. Ngoài ra nó còn quản lý tốt việc truy cập đồng thời từ nhiều người dùng để tránh xung đột dữ liệu.

1.12.2. Các tính năng của SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu mạnh mẽ của Microsoft, không chỉ hỗ trợ lưu trữ và truy vấn dữ liệu hiệu quả mà còn tích hợp các công cụ phân tích và trí tuệ doanh nghiệp (Business Intelligence – BI) nhằm hỗ trợ các tổ chức đưa ra quyết định chính xác. Một số tính năng nổi bật của SQL Server bao gồm:

- Quản lý chất lượng dữ liệu (Data Quality Services): SQL Server tích hợp các dịch vụ hỗ trợ đánh giá và nâng cao chất lượng dữ liệu, bao gồm phát hiện lỗi, sửa chữa và chuẩn hóa dữ liệu đầu vào.
- Quản lý dữ liệu chủ (Master Data Services): Tính năng này giúp tổ chức và duy trì thông tin cốt lõi (master data) một cách đồng bộ và nhất quán, từ đó đảm bảo tính chính xác và toàn vẹn cho các hệ thống khác nhau cùng sử dụng dữ liệu.
- SQL Server Data Tools (SSDT): Đây là bộ công cụ phát triển cơ sở dữ liệu tích hợp trong môi trường Visual Studio, hỗ trợ thiết kế lược đồ, viết truy vấn, kiểm thử và triển khai cơ sở dữ liệu một cách dễ dàng và hiệu quả.
- SQL Server Management Studio (SSMS): Là công cụ quản trị toàn diện giúp người dùng thao tác với cơ sở dữ liệu thông qua giao diện đồ họa hoặc dòng lệnh. SSMS hỗ trợ thực hiện các tác vụ như giám sát hệ thống, sao lưu, khôi phục và tối ưu hiệu suất truy vấn.
- SQL Server Analysis Services (SSAS): Cung cấp khả năng xây dựng các mô hình phân tích dữ liệu đa chiều hoặc theo mô hình tabular, từ đó phục vụ các truy vấn phân tích chuyên sâu và hỗ trợ ra quyết định.
- SQL Server Reporting Services (SSRS): Giải pháp mạnh mẽ để tạo và phân phối báo cáo từ dữ liệu trong SQL Server. SSRS cho phép người dùng thiết kế báo cáo tùy chỉnh và xuất chúng sang nhiều định dạng khác nhau, đồng thời dễ dàng chia sẻ qua web hoặc hệ thống nội bộ [23].

1.12.3. Tính tương thích

- Hệ điều hành hỗ trợ: Windows, Linux,...
- Môi trường hoạt động: Chạy trên máy chủ hoặc dịch vụ cloud (Azure SQL, Amazon RDS, Docker...).
- Yêu cầu đầu vào: schema cơ sở dữ liệu, dữ liệu đầu vào và truy vấn SQL.
- Công cụ phụ thuộc:
 - SQL Server Management Studio (SSMS): Là công cụ chính thức của Microsoft để quản lý SQL Server. SSMS cung cấp giao diện người dùng đồ họa để quản lý cơ sở dữ liệu, viết và thực thi truy vấn, và thực hiện các tác vụ quản trị.

- Azure Data Studio: Là công cụ mã nguồn mở, đa nền tảng, hỗ trợ quản lý SQL Server và Azure SQL Database. Kể từ phiên bản SSMS 18.7, Azure Data Studio được cài đặt tự động cùng với SSMS.

1.13. CHROMA DB

1.13.1. Giới thiệu về cơ sở dữ liệu vector (vector database)

Trong thời đại hiện nay, trí tuệ nhân tạo (AI) đang ngày càng ảnh hưởng mạnh mẽ đến lĩnh vực phát triển phần mềm, kéo theo sự gia tăng đáng kể về nhu cầu xử lý và lưu trữ dữ liệu. Sự tiến bộ trong các kiến trúc AI và học máy (Machine Learning – ML) đã mở ra khả năng xây dựng các mô hình nhúng (embedding models), cho phép chuyển đổi nhiều loại dữ liệu như văn bản, hình ảnh hay âm thanh thành các vector số học phản ánh ý nghĩa và ngữ cảnh tiềm ẩn của chúng.

Cơ sở dữ liệu vector là một loại hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) được thiết kế chuyên biệt để lưu trữ và truy xuất các vector nhiều chiều với hiệu suất cao. Mỗi vector là một biểu diễn số hóa của dữ liệu, mang thông tin đặc trưng về đối tượng như đặc điểm, ngữ nghĩa, hoặc các thuộc tính khác. Ví dụ, trong các hệ thống tìm kiếm hình ảnh, khi người dùng tải lên một ảnh, hệ thống sẽ chuyển ảnh đó thành vector và so sánh với các vector khác để tìm ra những hình ảnh tương đồng – đó chính là một ứng dụng thực tế của cơ sở dữ liệu vector.

Một số cơ sở dữ liệu vector mã nguồn mở hiện nay được đánh giá cao không chỉ bởi tính dễ sử dụng mà còn nhờ khả năng tích hợp linh hoạt với hệ sinh thái Python, phục vụ hiệu quả cho các tác vụ AI và ML hiện đại.

1.13.2. Định nghĩa về ChromaDB



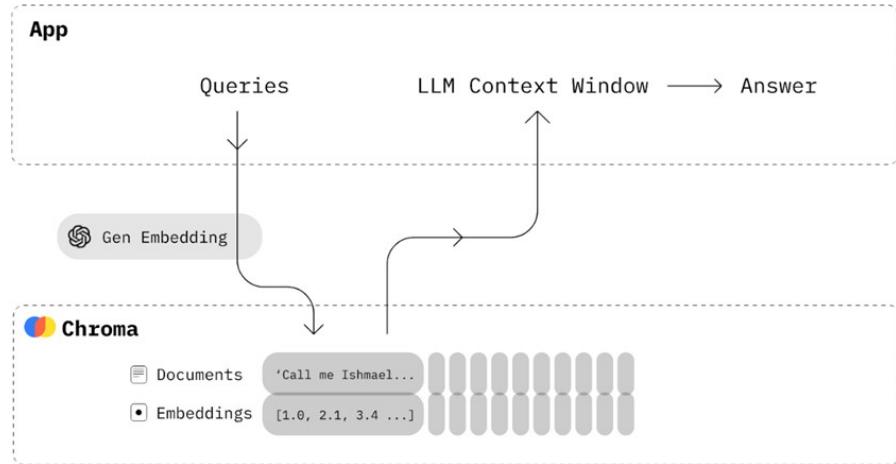
Chroma

Hình 17: ChromaDB

Chroma là một kho lưu trữ vector nguồn mở được sử dụng để lưu trữ và truy xuất các nhúng vector. Công dụng chính của nó là lưu các nhúng cùng với siêu dữ liệu để các mô hình ngôn ngữ lớn sử dụng sau này. Ngoài ra, nó cũng có thể được sử dụng cho các công cụ tìm kiếm ngữ nghĩa trên dữ liệu văn bản.

1.13.3. Cách hoạt động của Chroma

ChromaDB là một hệ quản trị cơ sở dữ liệu vector được thiết kế để lưu trữ và truy vấn các biểu diễn vector của tài liệu, phục vụ các ứng dụng học sâu và truy hồi thông tin dựa trên ngữ nghĩa.



Hình 18: Cách hoạt động của Chroma

Bước 1: Khởi tạo Collection

Tương tự như bảng (table) trong cơ sở dữ liệu quan hệ truyền thống, ChromaDB sử dụng khái niệm collection để tổ chức dữ liệu. Đây là nơi lưu trữ các tài liệu (documents) cùng với các vector embedding tương ứng. Khi khởi tạo, người dùng có thể lựa chọn mô hình embedding tùy chỉnh hoặc sử dụng mô hình mặc định được cung cấp bởi ChromaDB là all-MiniLM-L6-V2 – một mô hình nhẹ nhưng hiệu quả cho việc biểu diễn ngữ nghĩa của văn bản.

Bước 2: Thêm tài liệu vào collection

Sau khi tạo collection, người dùng có thể thêm các tài liệu văn bản vào hệ thống. Nếu tài liệu ở dạng văn bản thô, ChromaDB sẽ tự động chuyển đổi chúng thành các vector embedding bằng mô hình đã định sẵn. Mỗi tài liệu cần có một ID duy nhất để định danh, và người dùng cũng có thể tùy chọn thêm các tmetadata để phục vụ cho việc lọc hoặc truy vấn nâng cao sau này.

Bước 3: Truy vấn cơ sở dữ liệu vector

Khi collection đã được bổ sung đầy đủ dữ liệu, người dùng có thể bắt đầu thực hiện truy vấn nhằm tìm kiếm các tài liệu có nội dung tương đồng với một câu truy vấn

cho trước (query). Việc tìm kiếm này được thực hiện dựa trên khoảng cách ngữ nghĩa giữa các vector, thường sử dụng các chỉ số như cosine similarity.

Bước 4: Lọc tài liệu với meta-filtering

Một trong những tính năng mạnh mẽ của ChromaDB là hỗ trợ meta-filtering – cho phép lọc tài liệu theo các tiêu chí cụ thể dựa trên metadata đã lưu trước đó. Tính năng này rất hữu ích trong các trường hợp cần giới hạn truy vấn theo ngữ cảnh, danh mục, hoặc nguồn dữ liệu cụ thể [24].

1.13.4. Mức độ phổ biến

- Github: ChromaDB thu hút khoảng 20.000 sao trên GitHub.
- Các hệ thống lớn sử dụng: Được sử dụng trong nhiều dự án mã nguồn mở và tích hợp trong các hệ thống RAG phổ biến.
- Phổ biến trong sách báo: ChromaDB ngày càng được nhắc đến nhiều trong tài liệu về vector database và hệ thống AI tích hợp LLM, nhưng vẫn còn mới so với các hệ quản trị cơ sở dữ liệu truyền thống.

1.14. CÔNG CỤ PYMUPDF

1.14.1. Giới thiệu

PyMuPDF, còn được gọi là Fitz, là một thư viện Python mã nguồn mở cung cấp một bộ công cụ toàn diện để làm việc với các tệp PDF. Với PyMuPDF, người dùng có thể thực hiện hiệu quả các tác vụ như mở tệp PDF, trích xuất văn bản, hình ảnh và bảng, thao tác các thuộc tính trang như xoay và cắt, tạo tài liệu PDF mới và chuyển đổi các trang PDF thành hình ảnh [25].

PyMuPDF hỗ trợ một số tính năng được liệt kê dưới đây:

- Đọc tài liệu PDF: PyMuPDF có thể mở và đọc tài liệu PDF, cho phép bạn truy cập vào văn bản, hình ảnh và nội dung khác trong đó.
- Trích xuất văn bản: Bạn có thể trích xuất văn bản từ tài liệu PDF, bao gồm nội dung văn bản, phông chữ và thông tin bố cục.
- Trích xuất hình ảnh: Bạn có thể trích xuất hình ảnh từ tài liệu PDF ở nhiều định dạng khác nhau, chẳng hạn như JPEG hoặc PNG.
- Trích xuất bảng: Bạn cũng có thể trích xuất bảng từ tài liệu PDF.

1.14.2. Tính tương thích

- Hệ điều hành hỗ trợ: Windows, Linux, macOS.

- Môi trường hoạt động: Cài đặt qua pip (pip install pymupdf), yêu cầu Python 3.8.0 trở lên.
- Yêu cầu đầu vào: Tập PDF hoặc các định dạng tài liệu khác.

1.14.3. Mức độ phổ biến

- Github: Langchain có khoảng 7000 sao trên GitHub.
- Các hệ thống lớn sử dụng: Được sử dụng trong các dự án như hệ thống hỏi đáp dựa trên truy xuất (RAG) và tích hợp trong các pipeline xử lý tài liệu.
- Mức độ phổ biến trong sách báo: Chưa có dữ liệu cụ thể trên Google Books Ngram.

1.15. CÔNG CỤ PADDLEOCR

1.15.1. Giới thiệu

PaddleOCR là một công cụ mã nguồn mở được phát triển bởi nhóm PaddlePaddle thuộc Baidu, phục vụ cho tác vụ nhận dạng ký tự quang học (OCR). PaddleOCR tích hợp đầy đủ pipeline bao gồm phát hiện văn bản, nhận dạng, layout analysis và hỗ trợ hơn 80 ngôn ngữ (trong đó có tiếng Việt).

PaddleOCR thường được ứng dụng trong số hóa tài liệu, trích xuất thông tin từ hóa đơn, bảng biểu, ảnh chụp, và là nền tảng phổ biến cho các hệ thống như tìm kiếm ngữ nghĩa hay chatbot tài liệu [26].

1.15.2. Tính tương thích

- Hệ điều hành: Windows, Linux, macOS.
- Môi trường hoạt động: Chạy được trên máy tính cá nhân, máy chủ, cloud (Docker, Colab, Azure...).
- Yêu cầu đầu vào: Hình ảnh chứa văn bản; có thể là ảnh chụp sách, tài liệu, hóa đơn...
- Công cụ phụ thuộc: Cần cài đặt PaddlePaddle framework, OpenCV, Numpy.

1.15.3. Mức độ phổ biến

- GitHub: PaddleOCR có khoảng 49.000 sao trên Github.
- Các hệ thống lớn sử dụng: Baidu, một số hệ thống OCR thương mại và cộng đồng nguồn mở.

- Mức độ phổ biến trên sách báo: Từ khóa "PaddleOCR" không xuất hiện trên Google Books Ngram.

1.16. CÔNG CỤ VIETOCR

1.16.1. Giới thiệu

VietOCR là một công cụ mã nguồn mở phục vụ nhận dạng ký tự tiếng Việt từ hình ảnh. VietOCR hỗ trợ cả hai mô hình: CRNN truyền thống và Transformer hiện đại. Công cụ này tập trung chủ yếu vào tiếng Việt, cho phép người dùng fine-tune hoặc sử dụng mô hình sẵn có cho các tài liệu in ấn, sách báo, v.v.

VietOCR có khả năng xử lý ảnh đầu vào với các bước như nhị phân hóa, lọc nhiễu, và cắt vùng văn bản trước khi thực hiện nhận dạng, giúp cải thiện độ chính xác đáng kể với tài liệu tiếng Việt có chất lượng kém. Công cụ này phù hợp cho các ứng dụng số hóa tài liệu hành chính, thư viện điện tử, và đặc biệt hữu ích trong các dự án OCR quy mô nhỏ hoặc không có GPU mạnh. Mặc dù không phổ biến bằng PaddleOCR, VietOCR vẫn được cộng đồng nghiên cứu tiếng Việt sử dụng rộng rãi [27].

1.16.2. Tính tương thích

- Hệ điều hành hỗ trợ: Windows, Linux, macOS.
- Môi trường hoạt động: Cài đặt qua pip (pip install vietocr)
- Yêu cầu đầu vào: Tệp PDF hoặc các định dạng tài liệu khác.

1.16.3. Mức độ phổ biến

- GitHub: VietOCR có khoảng 700 sao trên Github.
- Các hệ thống lớn sử dụng: Một số dự án xử lý tài liệu tiếng Việt, OCR sách thư viện.
- Mức độ phổ biến trên sách báo: Từ khóa "VietOCR" không xuất hiện trên Google Books Ngram.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. XÁC ĐỊNH YÊU CẦU KỸ THUẬT

Bảng 33: Các yêu cầu kỹ thuật cho hệ thống

Thành phần	Cấu hình chi tiết	Chú thích
GPU	NVIDIA Tesla P100 PCIe 16GB MSI RTX 3090 SUPRIM X 24GB	Tối ưu n_layers cho mô hình ngôn ngữ lớn cũng như tăng tốc quá trình tính toán embedding
CPU	Intel Core i9 10900X	Các tác vụ thông thường
RAM	64GB	Đảm bảo khả năng xử lý các tập dữ liệu lớn trong bộ nhớ
VRAM	16GB	Hỗ trợ GPU trong việc lưu trữ và xử lý các tensor lớn khi chạy mô hình ngôn ngữ lớn
Laptop máy khách	MSI GF63 Thin 11UC-1228VN - Bộ vi xử lý Intel Core i7-13800H - Card đồ họa GEFORCE GTX - NVIDIA RAM 8GB DDR4 và ổ cứng SSD 512GB	Gọi API từ server, thực hiện các chức năng xử lý ảnh, nhận diện khuôn mặt, chức năng gọi ý sách và xây dựng backend, frontend cho phần app

2.2. CÁC CHỨC NĂNG CHÍNH

2.2.1. Chức năng đăng nhập bằng cách nhận diện khuôn mặt

2.2.2.1. Mục tiêu

Chức năng đăng nhập bằng nhận diện khuôn mặt được xây dựng nhằm cung cấp phương thức xác thực nhanh chóng, an toàn và tiện lợi cho người dùng hệ thống thư viện thông minh. Thay vì sử dụng tên đăng nhập và mật khẩu truyền thống, người dùng – cụ thể là sinh viên hoặc thủ thư – có thể truy cập hệ thống chỉ bằng cách chụp ảnh khuôn mặt qua webcam tích hợp trên thiết bị.

2.2.2.2. Quy trình thực hiện

Hệ thống gồm 2 giai đoạn chính:

- Trích xuất đặc trưng khuôn mặt (Face Embedding) từ dữ liệu ảnh người dùng.

- Nhận diện khuôn mặt theo thời gian thực qua webcam và xác thực danh tính.

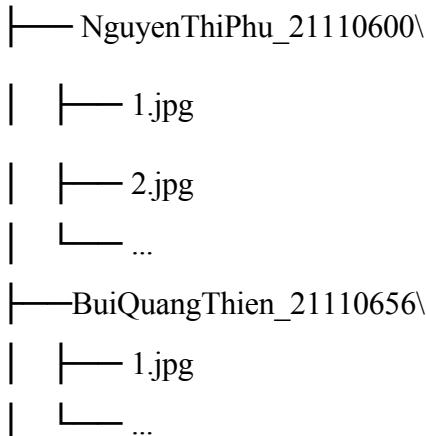
Bảng 34: Quy trình thực hiện chức năng đăng nhập bằng nhận diện khuôn mặt

Giai đoạn	Mục đích	Mô tả ngắn
1. Chuẩn bị dữ liệu	Tạo thư mục ảnh người dùng	Mỗi người 1 thư mục chứa nhiều ảnh chụp mặt
2. Trích xuất đặc trưng (embeddings)	Vector hóa khuôn mặt	Dùng mô hình pre-trained để tạo vectors
3. Nhận diện khuôn mặt	So sánh vector ảnh webcam với vector đã lưu. Nếu Cosine Similarity lớn hơn hoặc bằng 91% thì xem như trùng khớp khuôn mặt	Gắn nhãn người dùng hoặc “Unknown”
4. Ghi nhận trạng thái người dùng	Xác định ai đang đăng nhập	Biến current_user lưu thông tin

Giai đoạn 1: Chuẩn bị dữ liệu

Cấu trúc thư mục ảnh:

C:\FACE_LOGIN\image\



Mỗi thư mục chứa các ảnh chân dung khác nhau của một người. Tên thư mục là tên người dùng và mã số sinh viên.

Giai đoạn 2: Trích xuất Embeddings khuôn mặt

Bảng 35: Quy trình thực hiện chức năng đăng nhập bằng nhận diện khuôn mặt – giai đoạn 2

Bước	Diễn giải
1	Load mô hình InceptionResnetV1 (được huấn luyện trước trên tập dữ liệu VGG-Face2)
2	Duyệt qua từng thư mục con trong IMAGE_DIR (tức mỗi người dùng)
3	Với mỗi ảnh, thực hiện:
→	a. Đọc ảnh và chuyển về kích thước 160x160
→	b. Tiền xử lý ảnh: normalize về khoảng [-1, 1]
→	c. Truyền ảnh qua mô hình để thu được embedding vector kích thước 512
4	Lưu tất cả vector vào mảng embeddings, và tên tương ứng vào names

Bảng 36: Mô tả các file trong chức năng đăng nhập bằng nhận diện khuôn mặt

Tên file	Mô tả	Dạng dữ liệu
known_embeddings.npy	Ma trận chứa vector khuôn mặt	numpy.ndarray kích thước (số ảnh, 512)
known_names.npy	Danh sách tên ứng với từng vector	numpy.ndarray kiểu chuỗi

Giai đoạn 3: Nhận diện khuôn mặt theo thời gian thực

Bảng 37: Quy trình thực hiện chức năng đăng nhập bằng nhận diện khuôn mặt – giai đoạn 3

Bước	Mô tả
1	Dùng mô hình MTCNN để phát hiện khuôn mặt trong mỗi frame video từ webcam
2	Cắt vùng ảnh chứa khuôn mặt từ frame
3	Resize ảnh khuôn mặt về 160x160 và normalize như khi huấn luyện
4	Dùng InceptionResnetV1 để tạo embedding từ ảnh cắt
5	Tính cosine similarity giữa embedding mới và từng vector trong known_embeddings.npy
6	Nếu similarity cao nhất $> 0.91 \rightarrow$ gán nhãn người đó, ngược lại \rightarrow gán “Unknown”
7	Vẽ khung và tên người lên ảnh hiển thị

Giai đoạn 4: Ghi nhận người dùng đăng nhập

Ở giai đoạn này hệ thống ghi lại ai là người được nhận diện gần nhất

Bảng 38: Các biến ghi nhận người dùng đăng nhập

Biến	Mô tả
current_user	Chuỗi lưu tên người vừa được nhận diện
get_user_status()	Trả về tên người dùng hiện tại
reset_user_status()	Xóa thông tin người dùng hiện tại (khi logout)

2.2.2. Chức năng gợi ý sách cho người dùng

2.2.2.1. Mục tiêu

Chức năng gợi ý sách nhằm hỗ trợ người dùng, đặc biệt là sinh viên, nhanh chóng tìm được những đầu sách phù hợp với nhu cầu học tập và nghiên cứu của mình dựa trên các hành động xem, tải và hỏi đáp về sách.

2.2.2.2. Quy trình thực hiện

Quy trình gồm 4 giai đoạn chính, mỗi giai đoạn có mục tiêu và nhiệm vụ riêng nhằm chuẩn hóa dữ liệu, thiết kế môi trường học tăng cường, xây dựng mô hình học sâu, và tiến hành huấn luyện mô hình.

Giai đoạn 1: Chuẩn bị dữ liệu

Bảng 39: Chức năng gợi ý sách cho người dùng – Giai đoạn 1

Bước	Mô tả chi tiết
1	Tải mô hình SentenceTransformer "all-MiniLM-L6-v2" để chuyển nội dung sách thành vector embedding.
2	Ghép nối tiêu đề và mô tả sách thành một đoạn văn duy nhất. Duyệt qua tất cả sách trong cơ sở dữ liệu Book, kết hợp mô tả (desc) và giới thiệu (intro_text) để tạo văn bản đầu vào.
3	Mã hóa văn bản thành vector embedding và lưu vào book_embeddings[book.id].
4	Trích xuất dữ liệu tương tác từ UserBookInteraction: gồm user_id, book_id, action, timestamp
5	Với mỗi tương tác, ghép nối embedding của sách tương ứng và đưa vào danh sách data
6	Ghi dữ liệu ra file: user_book_data.pkl chứa DataFrame; book_embeddings.npy chứa từ điển book_id → embedding.

Giai đoạn 2: Xây dựng phần thưởng

Bảng 40: Chức năng gợi ý sách cho người dùng – Giai đoạn 2

Bước	Mô tả chi tiết
1	<p>Nhận đầu vào: hành động (view, download, ask), thời gian tương tác, embedding sách gợi ý và sách người dùng đã xem.</p>
2	<p>Quy ước phần thưởng base cho từng hành động:</p> <ul style="list-style-type: none"> • view: 1 • ask: 2 • download: 3 $base = \begin{cases} 1 & : view \\ 2 & : ask \\ 3 & : download \end{cases}$ <p>Công thức decay suy giảm theo thời gian để đánh giá mức độ "cũ" của một tương tác :</p> $\text{decay} = \frac{1}{1 + \frac{\Delta t}{86400}}$ <p>Công thức cosine similarity để so sánh sự giống nhau giữa vector embedding 2 quyển sách:</p> $\text{sim} = \frac{\vec{a} \cdot \vec{b}}{ \vec{a} \vec{b} }$
3	<p>Do 1 người dùng có thể tương tác với nhiều sách, nên ta chọn ra phần thưởng lớn nhất trong các phần thưởng cho từng sách của người dùng đó.</p> <p>Phần thưởng cho từng sách: Kết hợp phần thưởng base, hệ số suy giảm thời gian và độ tương đồng nội dung để tính phần thưởng cho 1 tương tác.</p> $r = base \times \text{decay} + 0.5 \times \text{sim}$ <p>Xác định phần thưởng lớn nhất ứng với người dùng đó</p>

	$R = \max(r_1, r_2, \dots, r_n)$
--	----------------------------------

Giai đoạn 3: Xây dựng mô hình Deep Q-Learning

Bảng 41: Chức năng gợi ý sách cho người dùng – Giai đoạn 3

Bước	Mô tả chi tiết
1	Xây dựng lớp DQNModel kế thừa torch.nn.Module
2	Kiến trúc mạng gồm 2 tầng: Linear(input_dim → 256), ReLU, Linear(256 → output_dim).

Giai đoạn 4: Huấn luyện mô hình

Bảng 42: Chức năng gợi ý sách cho người dùng – Giai đoạn 4

Bước	Mô tả chi tiết
1	Nạp dữ liệu đã lưu: DataFrame df và từ điển book_embeddings
2	Gán chỉ số cho từng book_id, xác định input_dim = embedding_dim + 1 cụ thể $385 = 384 + 1$
3	Khởi tạo mô hình DQN, tối ưu Adam, hàm mất mát MSELoss, bộ nhớ deque, và siêu tham số (gamma, epsilon, v.v)
4	Định nghĩa hàm get_state(user_id, book_emb) tạo tensor trạng thái đầu vào.
5	Với mỗi epoch, lặp qua dữ liệu ngẫu nhiên: <ul style="list-style-type: none"> - Tạo trạng thái hiện tại từ user_id và book_emb. - Chọn hành động: ngẫu nhiên (ϵ) hoặc hành động có Q-value cao nhất - Lấy embedding sách kế tiếp, tính phần thưởng bằng công thức đề cập ở giai đoạn 2 - Tạo trạng thái kế tiếp và lưu vào bộ nhớ.

6	<p>Nếu đủ dữ liệu trong bộ nhớ:</p> <ul style="list-style-type: none"> • Lấy mẫu minibatch • Dự đoán Q-value hiện tại và Q-value mục tiêu • Tính toán loss và cập nhật mô hình
---	---

2.2.3. Chức năng trích xuất lời mở đầu sách

2.2.3.1. Mục tiêu

Chức năng này nhằm tự động hóa quá trình thu thập phần lời mở đầu của sách – vốn thường mang tính tóm lược và định hướng nội dung – để lưu trữ vào vector database ChromaDB. Thông tin này được sử dụng như một nguồn dữ liệu đầu vào cho hệ thống RAG (Retrieval-Augmented Generation), giúp chatbot có thể trả lời các câu hỏi về nội dung sơ lược của sách một cách hiệu quả và chính xác hơn.

2.2.3.2. Quy trình thực hiện

Bước 1: Chuyển đổi các trang PDF thành ảnh

- File PDF đầu vào được xử lý bằng thư viện pdf2image để chuyển đổi từng trang thành ảnh định dạng JPEG hoặc PNG.
- Lý do: Các mô hình OCR như PaddleOCR chỉ nhận ảnh làm đầu vào, không xử lý được trực tiếp file PDF.
- Cấu hình chuyển đổi đảm bảo độ phân giải (DPI) cao (thường là 300–400 DPI) để giữ được độ sắc nét của chữ, giúp tăng độ chính xác khi nhận dạng sau này.

Bước 2: Phát hiện vùng chứa văn bản (Text Detection)

- Ảnh đầu vào được đưa vào mô hình PaddleOCR với lang="vi" để phát hiện khối văn bản.
- PaddleOCR sử dụng các mô hình detection như DBNet, EAST để xác định các vùng có chứa văn bản và trả về danh sách bounding boxes (tọa độ x, y của các khung chữ).
- Việc tách từng khối văn bản giúp sau này trích xuất dễ dàng từng dòng hoặc đoạn cụ thể thay vì toàn bộ ảnh.

Bước 3: Tiền xử lý ảnh (Image Preprocessing)

- Mục tiêu là tăng độ tương phản, giảm nhiễu, làm rõ nét chữ giúp quá trình OCR chính xác hơn.
- Các bước tiền xử lý:
 - Grayscale: Chuyển ảnh sang ảnh xám để đơn giản hóa.
 - Làm sạch nhiễu: Sử dụng các phép toán hình thái học (opening, closing) bằng OpenCV.
 - Tăng tương phản: Sử dụng adaptive threshold hoặc Otsu thresholding để phân biệt chữ và nền.
 - Làm dày hoặc mỏng nét chữ: Tùy vào đặc điểm ảnh, có thể sử dụng erosion hoặc dilation.
- Những khung ảnh chứa chữ sau khi xử lý sẽ được đưa vào bước OCR nhận dạng.

Bước 4: Nhận dạng văn bản (Text Recognition)

- Với mỗi vùng văn bản đã được xác định (bounding box), hệ thống thực hiện cắt ảnh tại vị trí đó và đưa vào mô hình VietOCR (sử dụng kiến trúc vgg_transformer hoặc CRNN).
- Mô hình sẽ chuyển hình ảnh thành văn bản Unicode tiếng Việt.
- Sau khi nhận dạng xong tất cả khối văn bản trong trang, các dòng chữ được sắp xếp lại theo vị trí để đảm bảo đúng thứ tự đọc từ trên xuống dưới, trái sang phải.

Bước 5: Tiền xử lý và trích xuất phần lời mở đầu

- Loại bỏ các phần không liên quan: Nhiều file PDF chứa mục lục, trang bìa quyền, giới thiệu nhà xuất bản ở đầu.
- Tìm tiêu đề phần “Lời mở đầu”
 - Tiếp theo, hệ thống tìm kiếm các dòng chứa các cụm từ như: LỜI MỞ ĐẦU, GIỚI THIỆU, LỜI NÓI ĐẦU, PREFACE, INTRODUCTION,...
 - Nếu phát hiện dòng tiêu đề này, văn bản ở phía sau nó sẽ được giữ lại để tiếp tục xử lý.

Bước 6: Chia đoạn, sinh vector nhúng và lưu vào vector database

a. Chia đoạn văn bản (Chunking)

- Văn bản lời mở đầu thường dài, cần chia nhỏ thành các đoạn ngắn để xử lý hiệu quả hơn trong hệ thống truy vấn và sinh câu trả lời.
- Hàm chunk_text_nltk() sử dụng bộ tokenizer của thư viện NLTK để tách văn bản thành các câu và tạo ra các đoạn (chunk) có tối đa 10 câu, với 3 câu chòng lấp giữa các đoạn kế tiếp.
- Cách chia đoạn này giúp giảm rủi ro mất mát thông tin quan trọng ở ranh giới các đoạn văn bản.

b. Mã hóa văn bản thành vector nhúng (Embedding)

- Mỗi đoạn sau khi chia nhỏ được chuyển thành vector embedding sử dụng mô hình paraphrase-multilingual-mpnet-base-v2 từ thư viện HuggingFace.
- Đây là mô hình đa ngôn ngữ, phù hợp với tiếng Việt, và được tối ưu cho nhiệm vụ so khớp ngữ nghĩa văn bản.
- Vector nhúng có 768 chiều, đảm bảo cân bằng giữa độ chính xác và tốc độ xử lý.

c. Lưu vào vector database (ChromaDB)

- Mỗi đoạn sẽ được lưu với:
 - Trường chroma:document: chứa văn bản gốc.
 - Metadata: gồm book_title, chunk_id, source (tên file PDF).
- Các vector này sẽ phục vụ chức năng tìm kiếm ngữ nghĩa trong hệ thống hỏi đáp.
- Kết quả khi lưu vào ChromaDB:

The screenshot shows the DB Browser for SQLite interface with the 'embedding_metadata' table selected. The table contains approximately 34 rows of data. The columns are: id, key, string_value, int_value, float_value, and bool_value. The string_value column contains various text snippets from books, such as '167_mách_diện_tù_ứng_dụng.pdf', '421_mách_diện_ứng_dụng_của_dòng_hồ_đ.', and 'Báo_du้อง_và_thú_nghiêm_thiết_bị_trò...'. The int_value, float_value, and bool_value columns are mostly NULL or 0. On the right side of the interface, there is a large text area displaying a summary of the extracted text, which includes sections like 'LỜI NGÓI BÀU', 'Phản 2 gồm các mách kiểm tra và thiết bị do luồn', and 'Phản 3 gồm các mách trò chơi và ứng dụng trong c'. Below this text area, there is a note about 'Editing row=4, column=3' and a 'Remote' section.

Hình 19: Kết quả lưu vào ChromaDB khi trích xuất lời mở đầu sách

2.2.4. Chức năng dùng chatbot hỏi câu hỏi liên quan đến sách

2.2.4.1. Mục tiêu

Mục tiêu của chức năng này là xây dựng một hệ thống trợ lý ảo (chatbot) có khả năng trả lời các câu hỏi của người dùng liên quan đến nội dung sách, dựa trên phần lời mở đầu đã được trích xuất và lưu trữ trong cơ sở dữ liệu vector. Chatbot có thể hỗ trợ người đọc tra cứu thông tin nhanh, tóm tắt nội dung, hoặc giải đáp các thắc mắc như: “Cuốn sách này nói về điều gì?” hay “Tác giả đề cập đến vấn đề gì trong sách?”. Mục tiêu cuối cùng là nâng cao trải nghiệm người dùng và tối ưu hóa việc tra cứu tài liệu trong hệ thống thư viện số.

2.2.4.2. Quy trình thực hiện

Bảng 43: Các thành phần trong chức năng dùng chatbot hỏi câu hỏi liên quan sách

Thành phần	Mô tả ngắn
LLM	Sử dụng mô hình gemma-2-9b-bit (qua LlamaCpp) để sinh câu trả lời.
Vector DB	Dữ liệu được nhúng và lưu bằng Chroma, sử dụng sentence-transformers/paraphrase-multilingual-mpnet-base-v2 [28].
Prompt Template	Câu lệnh hướng dẫn mô hình trả lời (tiếng Việt, chi tiết, thân thiện).

Retriever	Lấy top k=6 văn bản liên quan từ dữ liệu vector.
RAG Pipeline	Chuỗi kết nối các bước: lấy dữ liệu → tạo prompt → gọi LLM → xuất ra.

Giai đoạn 1: Tải mô hình ngôn ngữ lớn (LLM)

Công cụ: LlamaCpp từ LangChain.

Mô hình: gemma-2-9b-it-Q8_0.gguf (nhưng từ HuggingFace Hub).

Giai đoạn 2: Tạo Vector Database (ChromaDB)

- Embedding model: mô hình sentence-transformers/paraphrase-multilingual-mpnet-base-v2 của HuggingFace. Mô hình này hỗ trợ đa ngôn ngữ (trong đó có tiếng Việt), được huấn luyện cho nhiệm vụ hiểu ngữ nghĩa nên rất phù hợp cho tìm kiếm ngữ nghĩa (semantic search), có độ chính xác cao và tốc độ xử lý tốt, dễ tích hợp với ChromaDB thông qua thư viện sentence-transformers, và được cộng đồng đánh giá cao với khả năng bảo trì ổn định.
- Dữ liệu: Phân mỏ đầu sách, đã chia thành các đoạn nhỏ.
- Lưu trữ: Dưới thư mục DB_HCMUTELibrary.

Giai đoạn 3: Triển khai Retriever

Mục tiêu: Truy xuất k=6 đoạn văn có liên quan đến câu hỏi sinh viên.

Cơ chế: Tính toán khoảng cách cosine giữa vector câu hỏi và các vector đoạn văn.

Giai đoạn 4: Thiết kế Prompt Template

Cấu trúc xuất hiện trong prompt:

Bạn là một thủ thư thông minh tại Đại học Sư phạm Kỹ thuật TP.HCM, đang hỗ trợ sinh viên hiểu rõ hơn về nội dung sách.

Ngôn ngữ trả lời: Tiếng Việt chuẩn, dễ hiểu, có tính hướng dẫn.

Dưới đây là thông tin được hệ thống trích xuất từ lời mở đầu của các cuốn sách có liên quan:

[THÔNG TIN SÁCH]\nTiêu đề: {title}\n\n[PHẦN LỜI MỞ ĐẦU]\n

Yêu cầu:

- Trả lời câu hỏi của sinh viên một cách mạch lạc, súc tích và chính xác.
- Khi người dùng hỏi nội dung của sách, hãy tìm trong dữ liệu để trả lời. Không trả lời lan man.
- Khi người dùng kêu gọi ý sách về một chủ đề cụ thể, bạn hãy xem chủ đề đó thuộc lĩnh vực/ngành nào và chỉ gọi ý sách trong lĩnh vực đó. Chỉ gọi ý sách đã có trong dữ liệu, không được gọi ý sách bên ngoài dữ liệu.
- Văn phong thân thiện, chuyên nghiệp, tránh liệt kê khô khan hoặc máy móc.
- Nếu có đủ thông tin từ dữ liệu, hãy ưu tiên dùng chúng.
- Nếu không đủ thông tin, hãy viết câu trả lời dựa trên kiến thức phổ thông, và ghi chú rõ là "theo kiến thức phổ thông".
- Hãy trình bày câu trả lời chi tiết và mở rộng, ít nhất 1000 từ nếu có thể (hoặc càng dài càng tốt trong giới hạn cho phép).
- Nếu nội dung dài, chia ra nhiều đoạn hoặc mục để dễ theo dõi.

Câu hỏi của sinh viên:

{question}

Câu trả lời:

Giai đoạn 5: Kết nối các thành phần bằng LangChain RAG Chain

- Pipeline: Câu hỏi → Truy xuất dữ liệu → Format lại ngữ cảnh → Chèn vào Prompt → Trả lời từ LLM

2.2.5. Chức năng thêm sách bằng cách quét mã ISBN

2.2.5.1. Mục tiêu

- Mục tiêu của chức năng này là hỗ trợ người dùng (thủ thư hoặc quản trị viên) có thể dễ dàng thêm sách mới vào hệ thống thư viện chỉ bằng cách nhập mã ISBN hoặc quét mã vạch ISBN từ bìa sách. Thay vì nhập tay toàn bộ thông tin sách (tên sách, tác giả, nhà xuất bản, năm xuất bản,...), hệ thống sẽ tự động truy xuất thông tin từ các nguồn dữ liệu sách trực tuyến dựa vào ISBN và lưu vào cơ sở dữ liệu nội bộ.
- Việc này giúp tiết kiệm thời gian, đảm bảo độ chính xác của dữ liệu, đồng thời hỗ trợ thao tác nhanh chóng trong thực tế.

2.2.5.2. Quy trình thực hiện

Bước 1: Nhận ảnh đầu vào từ client

- Ảnh chụp mã vạch sách được gửi từ client (web/app) thông qua HTTP request dạng POST.

- Ảnh được mã hóa dưới dạng chuỗi base64 và truyền trong body của request.

Bước 2: Giải mã ảnh và xử lý ảnh

- Server sử dụng thư viện base64 để giải mã chuỗi ảnh thành dữ liệu nhị phân (bytes).
- Sau đó, dùng numpy để chuyển bytes thành mảng dữ liệu ảnh (numpy array) và sử dụng OpenCV (cv2.imdecode) để đọc ảnh.
- Ảnh được chuyển sang ảnh xám bằng cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) nhằm tăng độ tương phản và khả năng nhận diện mã vạch.

Bước 3: Nhận diện và giải mã mã vạch

- Sử dụng thư viện pyzbar để quét và giải mã các mã vạch trong ảnh (pyzbar.decode()).
- Với mỗi mã vạch được nhận diện:
 - o Giải mã dữ liệu thành chuỗi bằng obj.data.decode('utf-8').
 - o Kiểm tra định dạng chuỗi: nếu chuỗi là dãy số và có độ dài hợp lệ (10 hoặc 13 ký tự), hệ thống xác định đây là một mã ISBN hợp lệ.

Bước 4: Tra cứu thông tin sách theo ISBN

- Khi đã xác định được mã ISBN, hệ thống gọi API tra cứu thông tin sách.
- Hệ thống sẽ gửi yêu cầu tới Google Books API tra cứu mã ISBN và lấy về các thông tin như: tên sách, tác giả, nhà xuất bản, mô tả, ảnh bìa,...

Bước 5: Lưu thông tin sách vào hệ thống

- Nếu tìm thấy thông tin sách, hệ thống khởi tạo một đối tượng Book mới với các trường: ISBN, tiêu đề (title), tác giả (author), nhà xuất bản (publisher), năm xuất bản (year), mô tả (desc), người đăng (uploaded_by), và ID người dùng (user_id).
- Nếu có đường dẫn ảnh bìa (cover_url), hệ thống tiếp tục xử lý tải ảnh về.

Bước 6: Phản hồi kết quả cho client

- Trả về JSON chứa thông tin ISBN hoặc toàn bộ thông tin sách.
- Nếu không nhận diện được mã vạch hoặc ISBN không hợp lệ, trả về thông báo lỗi tương ứng.

2.2.6. Chức năng thêm sách bằng cách sử dụng công nghệ OCR

2.2.6.1. Mục tiêu

Mục tiêu của chức năng này là hỗ trợ thủ thư hoặc người dùng có thể thêm sách mới vào hệ thống thư viện chỉ bằng cách tải lên file PDF sách để thêm sách vào CSDL mà không cần nhập tay thông tin. Cụ thể:

- Tự động trích xuất tên sách và tên tác giả từ ảnh bìa sử dụng kỹ thuật OCR (Optical Character Recognition).
- Xử lý văn bản trích xuất để xác định chính xác các trường thông tin cần thiết như “Tiêu đề” và “Tác giả” bằng mô hình ngôn ngữ lớn (LLM).
- Hỗ trợ các ảnh bìa có chất lượng in ấn hoặc scan khác nhau, kể cả khi văn bản bị nghiêng hoặc có nhiều nhiễu.
- Tăng tốc và đơn giản hóa quy trình nhập liệu thủ công, nâng cao hiệu suất làm việc của thủ thư.

2.2.6.2. Quy trình thực hiện

Quy trình thực hiện chức năng này bao gồm nhiều bước xử lý từ frontend đến backend, từ việc xử lý ảnh cho đến phân tích ngôn ngữ tự nhiên. Cụ thể:

Bước 1: Tải file PDF sách lên hệ thống

Chọn file PDF có sẵn từ thiết bị: Người dùng sử dụng giao diện web để chọn một file PDF sách đã lưu trong máy tính. Ảnh phải ở định dạng hợp lệ có đuôi .pdf. Sau khi người dùng tải file lên.

Bước 2: Tiền xử lý ảnh

Sau khi file được tải lên, hệ thống tiến hành các bước xử lý ảnh bìa cơ bản nhằm cải thiện chất lượng đầu vào cho bước nhận dạng văn bản:

- Khử nhiễu hình ảnh (noise removal) để loại bỏ các chi tiết không mong muốn.
- Đảo màu và làm dày nét chữ để tăng độ tương phản, giúp mô hình OCR nhận diện tốt hơn.
- Chuẩn hóa kích thước ảnh đầu vào để đảm bảo tương thích với mô hình nhận dạng.

Các bước xử lý ảnh này được thực hiện bằng thư viện OpenCV và một số thuật toán xử lý ảnh truyền thống.

Bước 3: Nhận dạng văn bản từ ảnh (OCR)

Hệ thống sử dụng thư viện PaddleOCR để thực hiện nhận dạng văn bản tiếng Việt trên ảnh. PaddleOCR được cấu hình với các tham số phù hợp để tối ưu cho ảnh bìa sách, bao gồm:

- Sử dụng mô hình nhận dạng SVTR_LCNet.
- Kích hoạt tính năng phát hiện góc xoay văn bản để đảm bảo tính chính xác dù ảnh bị lệch.
- Giới hạn chiều dài văn bản đầu ra và số lượng ảnh xử lý theo lô để cân bằng hiệu suất.

Kết quả của bước này là một chuỗi văn bản (ocr_text) được hệ thống trích xuất từ nội dung trên ảnh bìa.

Bước 4: Phân tích văn bản để trích xuất thông tin sách

Văn bản trích xuất từ ảnh thường chứa nhiều thông tin như tiêu đề, tác giả, nhà xuất bản, mã vạch, v.v. Do đó, hệ thống cần thực hiện một bước phân tích ngữ nghĩa để xác định đâu là tiêu đề sách và đâu là tên tác giả.

Cụ thể:

- Văn bản OCR sẽ được gửi đến một mô hình ngôn ngữ lớn (LLM) đã được huấn luyện và tích hợp sẵn trên server Zeppelin.
- Một prompt bằng tiếng Việt được tạo ra nhằm hướng dẫn mô hình trích xuất hai trường thông tin chính: Tiêu đề và Tác giả, đồng thời bỏ qua các phần không cần thiết như nhà xuất bản, mã vạch, số ISBN.
- Server sử dụng mô hình gemma-2-9b-it-GGUF chạy bằng LlamaCpp để xử lý và trả về kết quả dưới dạng văn bản.

Thông tin đầu ra gồm:

- Tiêu đề: tên cuốn sách.
- Tác giả: tên người viết sách.

Prompt chi tiết:

Bạn là chuyên gia phân tích văn bản OCR từ bìa sách và có khả năng hiệu chỉnh lỗi chính tả, dấu câu trong văn bản bị sai lệch.

Nhiệm vụ của bạn:

1. Xác định chính xác Tiêu đề sách:

- Là cụm từ mô tả nội dung chính của cuốn sách.
- Thường có độ dài từ 3 từ trở lên, dễ nhận biết qua cách trình bày nổi bật (in hoa, phông lớn, giữa trang).
- Ưu tiên cụm từ có tính học thuật hoặc mang nội dung chuyên môn.

2. Xác định Tác giả:

- Là tên người (hoặc nhóm), có thể đứng gần các từ như “Tác giả”, “Author”, “Editor”, hoặc xuất hiện ở cuối văn bản.

- Có thể bao gồm học hàm: “GS.”, “TS.”, “PGS.”,...

Bạn phải tự động sửa lỗi chính tả nếu văn bản có sai sót.

Không được phép ghi "Không rõ", bạn bắt buộc phải đưa ra dự đoán hợp lý nhất cho cả Tiêu đề và Tác giả, ngay cả khi thông tin không rõ ràng.

Chi trả lời đúng theo định dạng sau (không thêm mô tả, không xuống dòng thừa):

Tiêu đề: ...

Tác giả: ...

Văn bản OCR đầu vào:

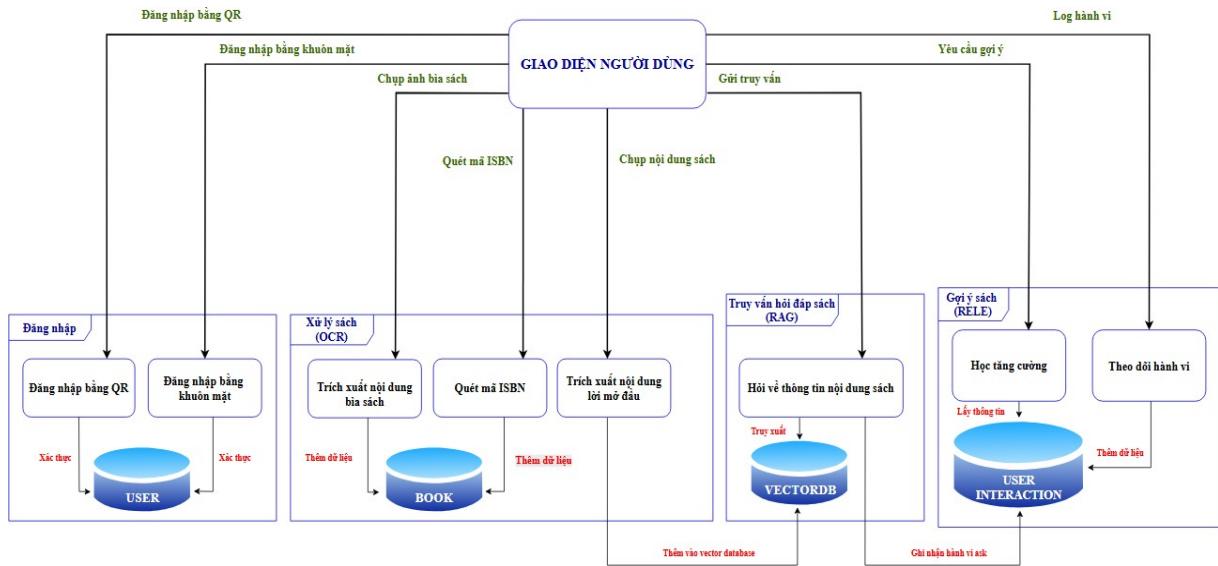
{ocr_text}

Bước 5: Lưu dữ liệu vào cơ sở dữ liệu

Thông tin sách sau khi được xác nhận sẽ được lưu vào cơ sở dữ liệu của hệ thống quản lý thư viện. Cấu trúc dữ liệu lưu trữ bao gồm các trường cơ bản như: tiêu đề sách, tên tác giả, đường dẫn sách, người thêm, ...

2.3. THIẾT KẾ HỆ THỐNG

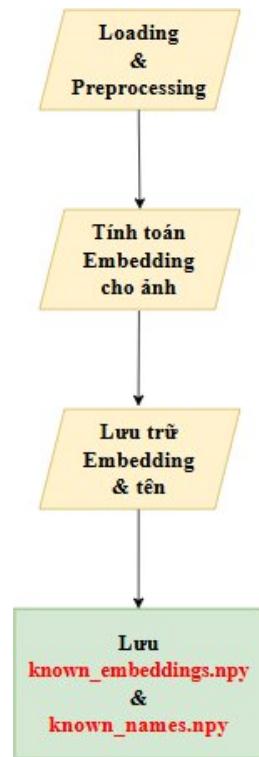
2.3.1. Sơ đồ tổng quan



Hình 20: Sơ đồ tổng quan hệ thống

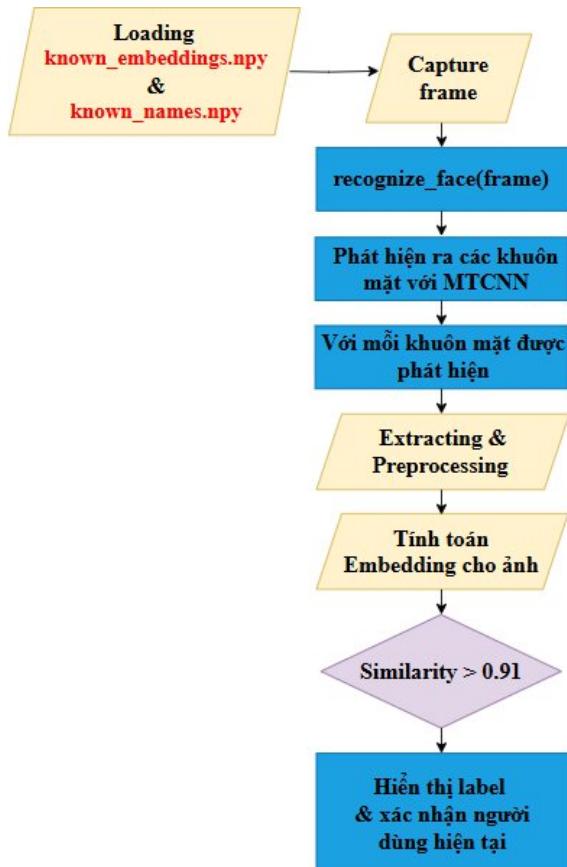
2.3.2. Sơ đồ chức năng đăng nhập bằng khuôn mặt

Sơ đồ họa quy trình trích xuất đặc trưng khuôn mặt từ ảnh tĩnh để lưu trữ làm cơ sở dữ liệu so sánh. Quá trình bắt đầu bằng cách truy cập vào thư mục chứa ảnh của từng người dùng. Mỗi ảnh sẽ được xử lý bằng thư viện PIL để chuyển sang dạng RGB, sau đó resize về kích thước chuẩn 160x160 và chuẩn hóa bằng transform. Sau bước xử lý ảnh, mô hình nhận dạng khuôn mặt InceptionResnetV1 - được huấn luyện trước trên tập VGGFace2 sẽ được sử dụng để trích xuất vector đặc trưng cho từng khuôn mặt. Các vector này được lưu trữ kèm theo tên người tương ứng trong hai file *known_embeddings.npy* và *known_names.npy*, phục vụ cho bước nhận diện khuôn mặt về sau.



Hình 21: Sơ đồ chức năng đăng nhập bằng khuôn mặt

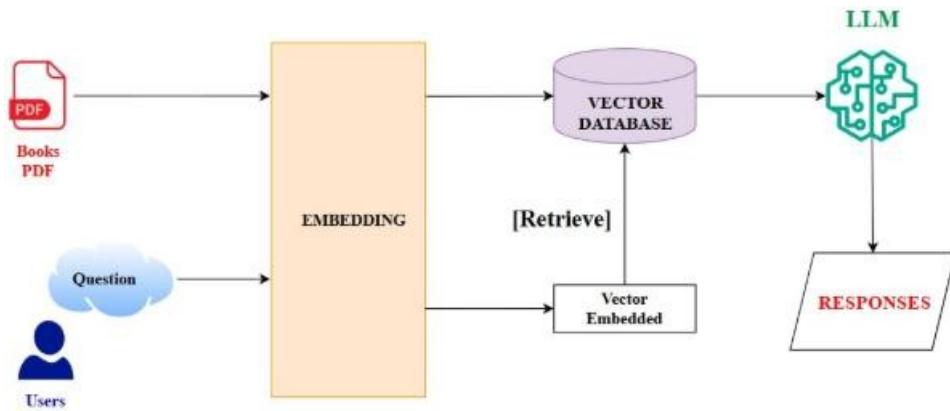
Sơ đồ mô tả chi tiết quy trình nhận diện khuôn mặt trong thời gian thực thông qua camera. Đầu vào là khung hình từ webcam, sau đó được xử lý bằng mô hình MTCNN để phát hiện vị trí khuôn mặt. Với mỗi khuôn mặt được phát hiện, hệ thống cắt ảnh và xử lý lại theo kích thước và chuẩn hóa tương tự như bước trích xuất. Ảnh khuôn mặt đã xử lý được đưa qua mô hình InceptionResnetV1 để tạo vector đặc trưng, sau đó so sánh với các vector đã lưu trong *known_embeddings.npy* bằng cosine similarity. Nếu độ tương đồng vượt ngưỡng 0.91, hệ thống xác định được tên người dùng và hiển thị tên ngay tại khung hình. Ngoài ra, biến toàn cục *current_user* được cập nhật để lưu trạng thái người dùng hiện tại, hỗ trợ cho các chức năng đăng nhập hoặc phân quyền sau này.



Hình 22: Sơ đồ chức năng đăng nhập bằng khuôn mặt

2.3.3. Sơ đồ chức năng dùng chatbot hỏi câu hỏi liên quan tới sách

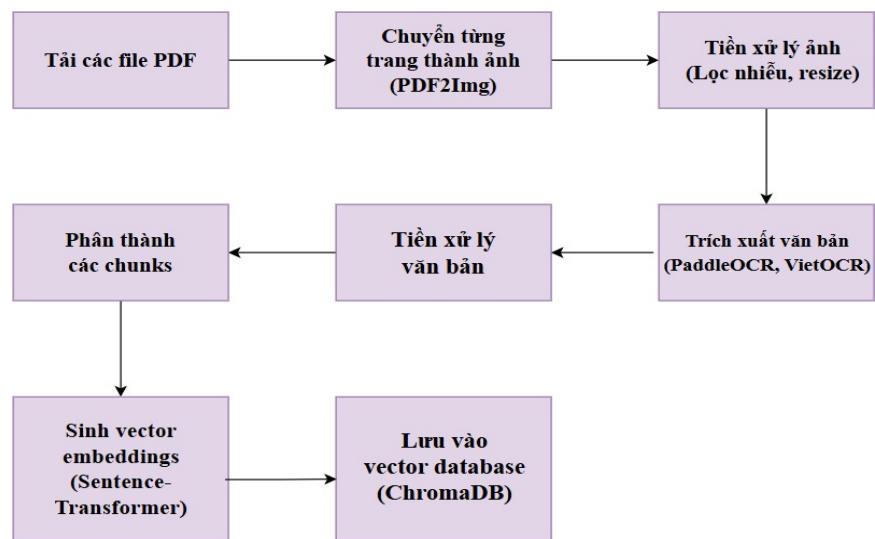
Sơ đồ minh họa luồng hoạt động của hệ thống RAG sử dụng dữ liệu là phần lời mở đầu của các sách trong thư viện số. Trước tiên, nội dung từ các file PDF của sách được trích xuất và chuyển đổi thành vector embedding thông qua một mô hình mã hóa ngữ nghĩa. Các vector này sau đó được lưu trữ trong một Vector Database. Khi người dùng đặt câu hỏi, hệ thống cũng biến câu hỏi đó thành vector tương ứng và truy vấn vào cơ sở dữ liệu để tìm kiếm những đoạn văn bản có ngữ nghĩa gần nhất. Các đoạn văn bản được truy xuất sẽ được kết hợp với câu hỏi đầu vào và đưa vào mô hình ngôn ngữ lớn để tạo ra câu trả lời phù hợp. Nhờ cơ chế kết hợp truy xuất và sinh phản hồi, hệ thống có thể đưa ra câu trả lời chính xác và sát với nội dung sách.



Hình 23: Sơ đồ chức năng dùng chatbot hỏi câu hỏi liên quan sách

2.3.4. Sơ đồ chức năng trích xuất lời mở đầu sách

Quy trình trích xuất lời mở đầu từ sách PDF và lưu vào vector database được thực hiện qua nhiều bước liên tiếp. Trước tiên, hệ thống sẽ tải các tệp PDF chứa sách từ thư mục đầu vào và chuyển từng trang PDF thành ảnh bằng thư viện pdf2image. Sau đó, các ảnh này được tiền xử lý để cải thiện chất lượng (lọc nhiễu, thay đổi kích thước, tăng tương phản) nhằm nâng cao độ chính xác của bước nhận dạng văn bản. Tiếp theo, hệ thống sử dụng công cụ OCR như VietOCR hoặc PaddleOCR để trích xuất nội dung văn bản từ các ảnh. Phần văn bản thu được sẽ trải qua bước tiền xử lý ngôn ngữ tự nhiên để loại bỏ các ký tự lỗi, chuẩn hóa định dạng và làm sạch nội dung. Sau đó, văn bản được phân chia thành các đoạn nhỏ (chunk) có độ dài khoảng 1000 tokens để phù hợp với giới hạn đầu vào của mô hình nhúng. Mỗi đoạn sẽ được đưa vào mô hình sinh vector nhúng (chẳng hạn như sentence-transformers) để mã hóa thành vector đa chiều, phản ánh ngữ nghĩa của nội dung. Cuối cùng, các vector cùng với thông tin bổ sung như tiêu đề sách và số thứ tự đoạn sẽ được lưu trữ vào cơ sở dữ liệu vector (ChromaDB), sẵn sàng phục vụ cho các truy vấn ngữ nghĩa trong hệ thống RAG.

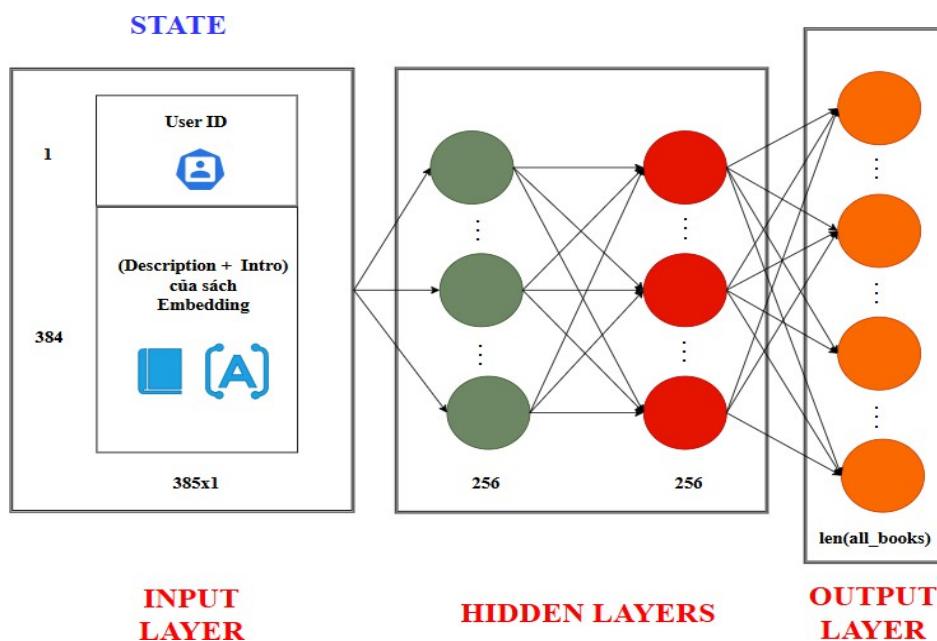


Hình 24: Sơ đồ chức năng trích xuất lời mở đầu sách

2.3.5. Sơ đồ chi tiết chức năng gợi ý sách

Mạng nơ-ron được thiết kế với 3 layers:

- Input layer: nhận trạng thái của môi trường dưới dạng vector 1D.
- Hidden layers: tầng thứ nhất có 256 nodes, tầng sau có 256 nodes và sử dụng hàm kích hoạt ReLU.
- Output layer: trả về giá trị Q tương ứng với các hành động khả thi trong trạng thái hiện tại, cụ thể là gợi ý các sách trên nội dung và tương tác của người dùng trong lịch sử.

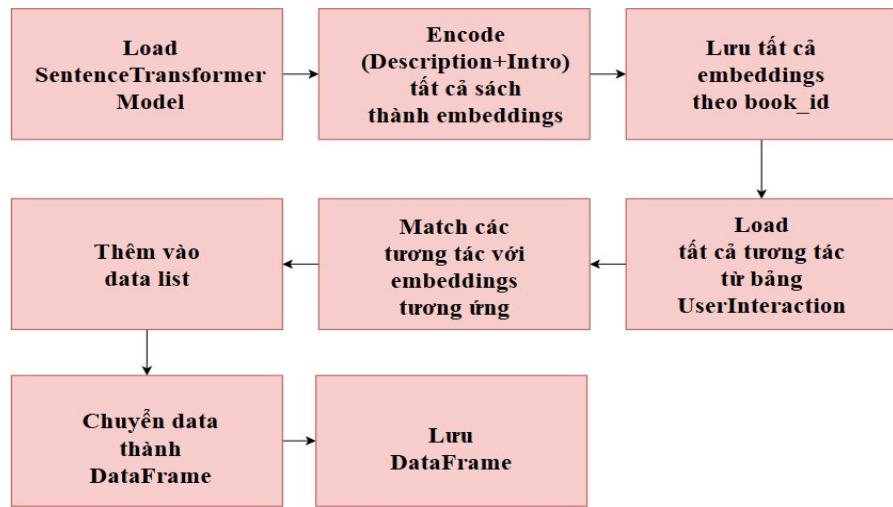


Hình 25: Cấu trúc mạng mô hình chức năng gợi ý sách

Bảng 44: Cấu trúc mạng mô hình của chức năng gợi ý sách

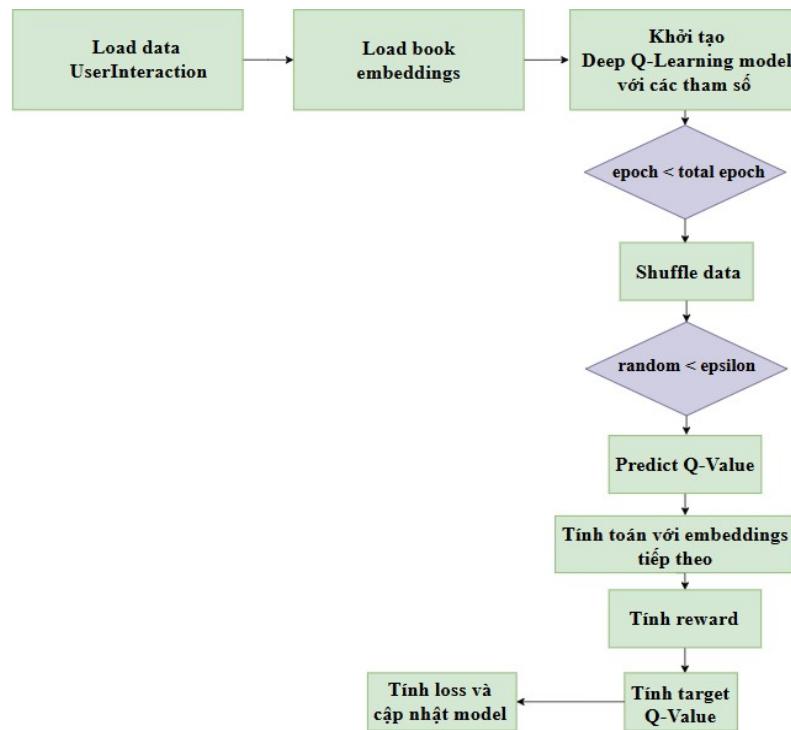
Layer	Input Size	Output Size	Activation
Input Layer	state_dim (385x1)	256	-
Fully Connected 1	256	256	RELU
Fully Connected 2	256	len(all_books)	-

Input layer là 1 vector 1D được nối bằng phép toán Concatenate giữa [user_id] và vector embeddings mô tả sách và phần Intro của sách đó. Cụ thể quy trình như sau:



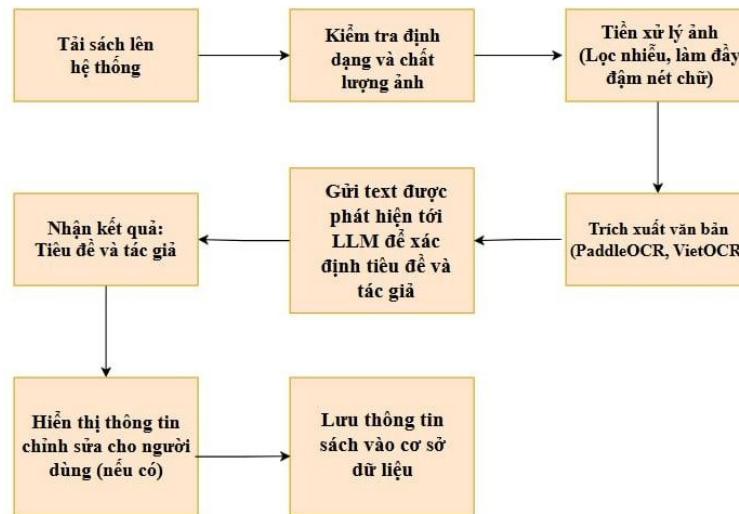
Hình 26: Sơ đồ chức năng gợi ý sách

Sau đó tiến hành đưa vào mạng Deep Q-Learning để xử lý theo quy trình:



Hình 27: Sơ đồ quy trình đưa mạng Deep Q Learning vào hệ thống

2.3.6. Sơ đồ chức năng thêm sách bằng cách sử dụng công nghệ OCR

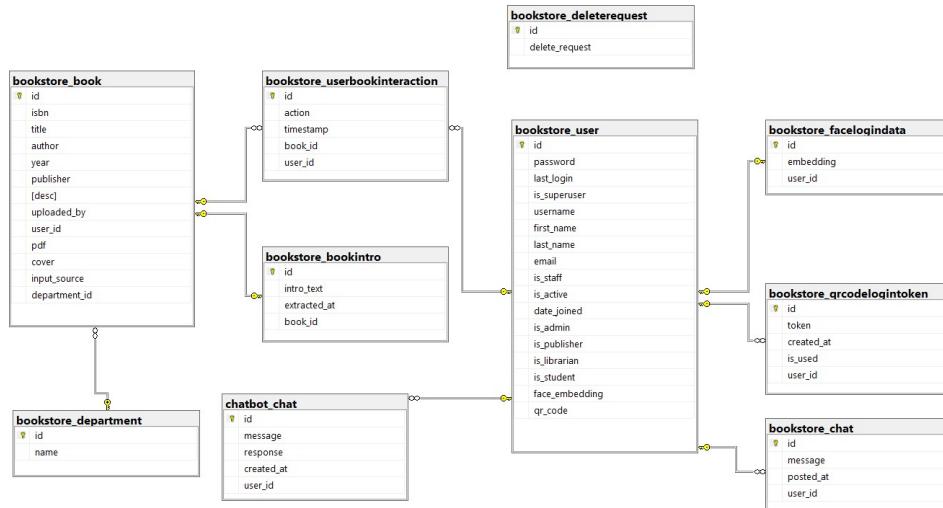


Hình 28: Sơ đồ chức năng thêm sách bằng công nghệ OCR

Chức năng “Thêm sách bằng cách sử dụng công nghệ OCR” cho phép người dùng (thủ thư) nhanh chóng nhập dữ liệu sách vào hệ thống bằng cách chụp ảnh hoặc tải lên file sách dạng PDF. Sau khi người dùng gửi ảnh, hệ thống sẽ kiểm tra định dạng và chất lượng ảnh, sau đó thực hiện các bước tiền xử lý như loại bỏ nhiễu và làm rõ chữ. Ảnh sau xử lý được đưa vào hệ thống OCR để trích xuất văn bản. Văn bản thu được tiếp tục được gửi đến mô hình ngôn ngữ lớn (LLM) để phân tích và xác định tiêu

để cùng tác giả sách. Kết quả này được hiển thị cho người dùng xác nhận hoặc chỉnh sửa lại nếu cần, trước khi lưu vào cơ sở dữ liệu thư viện. Quy trình này giúp đơn giản hóa thao tác nhập liệu, nâng cao tính chính xác và tiết kiệm thời gian cho người quản lý thư viện.

2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU



Hình 29: Cơ sở dữ liệu hệ thống

User(bookappuser)

Bảng này lưu thông tin người dùng của hệ thống, bao gồm các loại tài khoản như sinh viên, thủ thư, quản trị viên và nhà xuất bản. Ngoài thông tin cơ bản như tên, email, và mật khẩu, bảng còn hỗ trợ xác thực qua khuôn mặt và mã QR.

Bảng 45: Bảng User trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
password	CharField	Mật khẩu đã mã hóa
last_login	DateTimeField	Lần đăng nhập gần nhất
is_superuser	BooleanField	Là superuser (admin toàn hệ thống)
username	CharField	Tên đăng nhập (duy nhất)

first_name	CharField	Tên
last_name	CharField	Họ
email	EmailField	Email
is_staff	BooleanField	Có quyền vào admin site
is_active	BooleanField	Trạng thái hoạt động tài khoản
date_joined	DateTimeField	Ngày tạo tài khoản
is_admin	BooleanField	Cờ đánh dấu người dùng là quản trị viên
is_librarian	BooleanField	Cờ đánh dấu người dùng là thủ thư
is_student	BooleanField	Cờ đánh dấu người dùng là sinh viên (mặc định là True)

Book (bookstore_book)

Chứa thông tin về sách, bao gồm tiêu đề, tác giả, năm xuất bản, nhà xuất bản, mô tả, file PDF và ảnh bìa. Mỗi sách có thể được liên kết với một bộ môn và nguồn nhập liệu (nhập tay, OCR hoặc ISBN).

Bảng 46: Bảng Book trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
isbn	CharField	Số ISBN (tùy chọn)
title	CharField	Tên sách
author	CharField	Tác giả

year	CharField	Năm xuất bản
publisher	CharField	Nhà xuất bản
desc	CharField	Mô tả sách
uploaded_by	CharField	Tên người tải lên
user_id	CharField	ID người tải lên (không phải ForeignKey)
pdf	FileField	File PDF sách
cover	ImageField	Ảnh bìa sách
department_id	ForeignKey	Mã khoa
input_source	CharField	Nguồn nhập liệu (ISBN, OCR, Manual)

DeleteRequest (bookstore_deleterequest)

Xử lý yêu cầu xóa sách của người dùng. Yêu cầu này sẽ gửi cho thủ thư.

Bảng 47: Bảng DeleteRequest trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AuoField	Khóa chính
delete_request	CharField	Nội dung yêu cầu

Department (bookstore_department)

Lưu thông tin về các bộ môn/khoa trong hệ thống. Mỗi bộ môn có thể liên kết với nhiều sách.

Bảng 48: Bảng Department trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
name	CharField	Tên khoa

BookIntro (bookstore_bookintro)

Lưu lời giới thiệu của sách, được tạo tự động từ nội dung sách hoặc nhập thủ công. Mỗi sách chỉ có một lời giới thiệu tương ứng.

Bảng 49: Bảng BookIntro trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
book_id	OneToOneField	Liên kết một sách
intro_text	TextField	Nội dung lời giới thiệu sách
extracted_at	DateTimeField	Ngày tạo lời giới thiệu (auto_now_add)

BookQA (bookapp_bookqa)

Lưu các câu hỏi và câu trả lời của người dùng liên quan đến sách. Mỗi câu hỏi có thẻ hoặc không liên kết với một quyền sách cụ thể.

Bảng 50: Bảng BookQA trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
user_id	ForeignKey	Người đặt câu hỏi
book_id	ForeignKey	Sách được hỏi (nullable)
question	TextField	Nội dung câu hỏi
answer	TextField	Câu trả lời
asked_at	DateTimeField	Thời gian hỏi (auto_now_add)

UserBookInteraction (bookapp_userbookinteraction)

Lưu lịch sử tương tác giữa người dùng và sách, bao gồm các hành động như xem, hỏi và tải sách. Giúp phân tích hành vi người dùng và cải thiện đề xuất.

Bảng 51: Bảng UserBookInteraction trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
user_id	ForeignKey	Người tương tác
book_id	ForeignKey	Sách được tương tác
action	CharField	Hành động: view, ask, download
timestamp	DateTimeField	Thời gian thực hiện hành động

QRCodeLoginToken (bookapp_qrcodelogintoken)

Lưu mã token đăng nhập tạm thời được tạo cho người dùng khi họ quét mã QR. Token này chỉ dùng một lần và có thời hạn sử dụng nhất định.

Bảng 52: Bảng QRCodeLoginToken trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích
id	AutoField	Khóa chính
user_id	ForeignKey	Người dùng sở hữu token
token	UUIDField	Mã định danh duy nhất dùng để đăng nhập bằng QR
created_at	DateTimeField	Ngày tạo token
is_used	BooleanField	Trạng thái đã sử dụng hay chưa

Chat(chatbot_chat)

Lưu các câu hỏi và câu trả lời của người dùng liên quan đến sách. Mỗi câu hỏi có thể hoặc không liên kết với một quyền sách cụ thể.

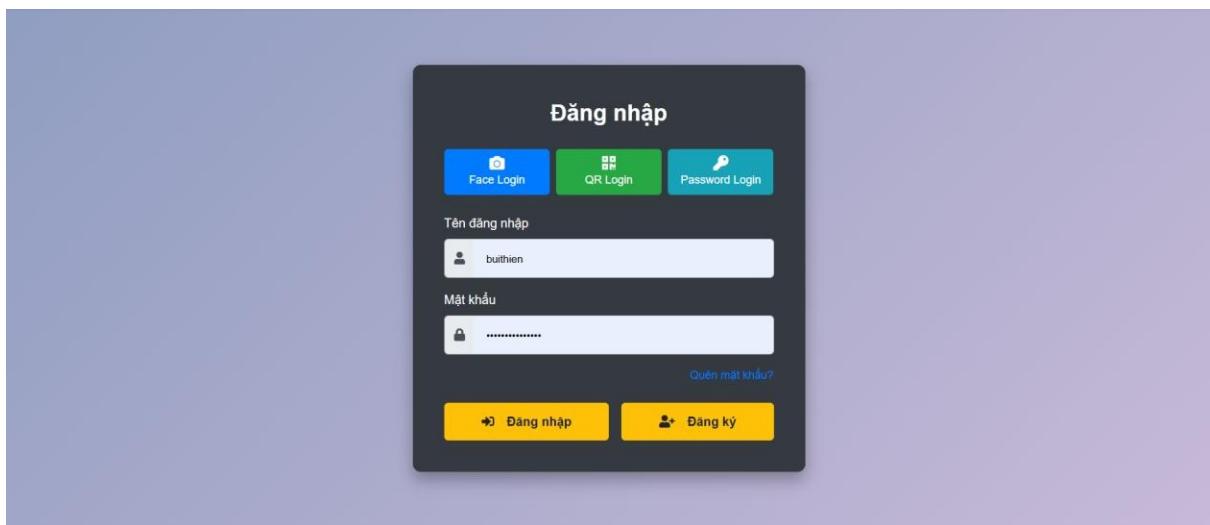
Bảng 53: Bảng Chat trong CSDL

Tên trường	Kiểu dữ liệu	Chức năng/Chú thích

id	AutoField	Khóa chính
message	TextField	Nội dung câu hỏi
response	TextField	Nội dung câu trả lời
create_at	DateTimeField	Thời gian hỏi
user_id	ForeignKey	ID người hỏi

2.5. THIẾT KẾ GIAO DIỆN HỆ THỐNG

2.5.1. Trang đăng nhập



Hình 30: Giao diện trang đăng nhập



Hình 31: Giao diện trang đăng nhập bằng khuôn mặt

2.5.2. Trang chủ sách

The screenshot shows the HCMUTE Library website. At the top, there's a navigation bar with links for Home, Library, Chat, and Yêu cầu sách. A search bar is also at the top. On the left, a sidebar menu includes 'Tất Cá' (All), 'Sách gợi ý' (Recommended Books), 'Tất cả sách' (All books), 'Khoa' (Faculty), 'Cơ Khí Chế Tạo Máy' (Mechanical Engineering), 'Cơ Khí Động Lực' (Mechanics), 'Công Nghệ In' (Printing Technology), 'Công Nghệ Thông Tin' (Information Technology), 'Công Nghệ Thực Phẩm' (Food Technology), and 'Điện - Điện Tử' (Electronics). The main content area features a section titled 'Sách được gợi ý cho bạn' (Books recommended for you) with three book cards:

- BẢO VỆ VÀ PHÁT TRIỂN MÔI TRƯỜNG CẢNH QUAN TRONG XÂY DỰNG ĐƯỜNG Ô TÔ**
Báo cáo và phát triển môi trường cảnh quan trong xây dựng...
- KIỂM ĐỊNH - SỬA CHỮA VÀ TĂNG CƯỜNG CẦU**
Kiểm định sửa chữa và tăng cường cầu
- HƯỚNG DẪN CÁCH TIẾNG ANH**
Hướng dẫn cách tiếng Anh

To the right, there's a blue box titled 'Trợ lý ảo HCMUTE' containing text about free resources on Coursera, edX, or Udemy, followed by a question about flood prevention. Below it is another box asking for feedback on the flood prevention book.

Hình 32: Giao diện trang chủ sách

2.5.3. Trang quản lý sách

The screenshot shows the library management system interface. At the top, there's a navigation bar with links for Home, Library, Chat, and Yêu cầu sách. A search bar is also at the top. The main content area is titled 'Sách Mới Thêm Gần Đây' (Recently Added Books) and includes buttons for '+ Thêm Sách' (Add Book) and 'Thêm Sách Bằng OCR' (Add Book by OCR). It shows a list of three books:

Mã Sách	Ảnh Bìa	Tên Sách	Khoa	Đọc	Tải Xuống
157		Sửa chữa máy lạnh và điều hòa không khí	Cơ Khí Động Lực	Xem PDF	Tải PDF
158		Sửa chữa điện ô tô	Cơ Khí Động Lực	Xem PDF	Tải PDF
159		Thiết bị tiết lưu và thiết bị phụ Sổ tay kỹ thuật lạnh	Cơ Khí Động Lực	Xem PDF	Tải PDF

Hình 33: Giao diện trang quản lý sách

2.5.4. Trang thêm sách bằng cách thủ công

The screenshot shows a web-based form titled "Thêm Sách Mới". The form fields are as follows:

- Tên Sách**: Nhập tên sách
- Tác Giả**: Nhập tên tác giả
- Nhà Xuất Bản**: Nhập tên nhà xuất bản
- Năm Xuất Bản**: Nhập năm xuất bản
- Mô Tả**: Nhập mô tả sách
- Ảnh Bìa**: A file input field showing "No file chosen" with a blue upload icon.
- File PDF**: A file input field showing "No file chosen" with a blue upload icon.

Hình 34: Giao diện trang thêm sách bằng thủ công

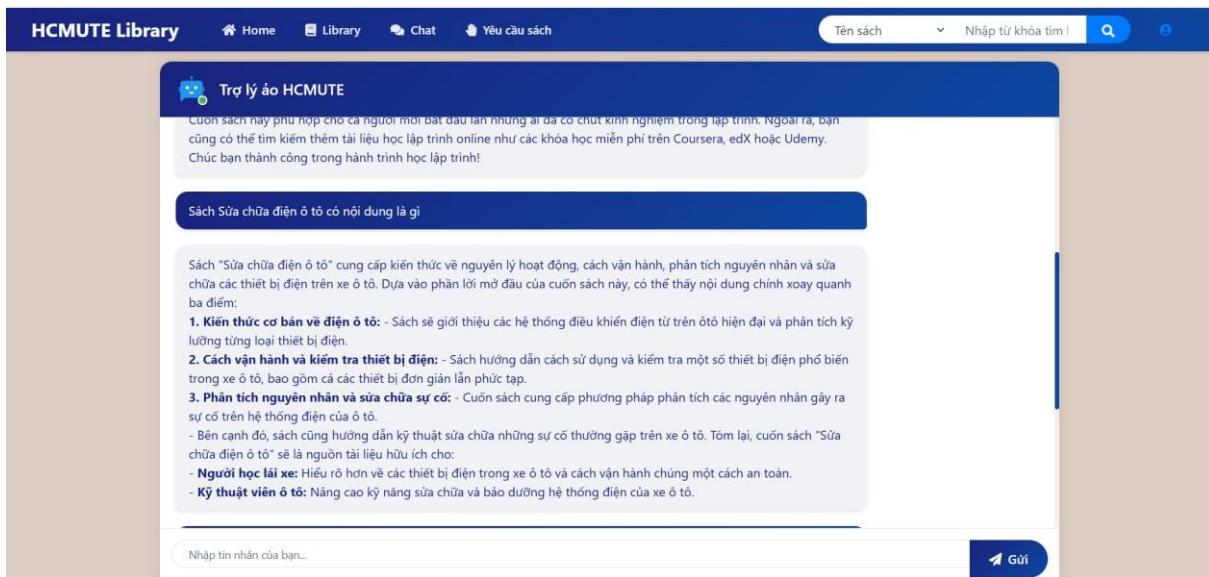
2.5.5. Trang thêm sách bằng OCR

The screenshot shows a web-based form titled "Thêm Sách Bằng PDF". The form includes the following features:

- Hướng dẫn sử dụng**:
 - Tải lên file PDF của sách. Hệ thống sẽ tự động:
 - Trích xuất trang đầu tiên làm ảnh bìa
 - Phân tích nội dung để lấy thông tin sách
 - Tự động điền các thông tin cần thiết
- Chọn File PDF**: A file input field showing "No file chosen" with a blue upload icon.
- Trích Xuất Thông Tin**: A button with a document icon.

Hình 35: Giao diện trang thêm sách bằng OCR

2.5.6. Trang chatbot tư vấn sách



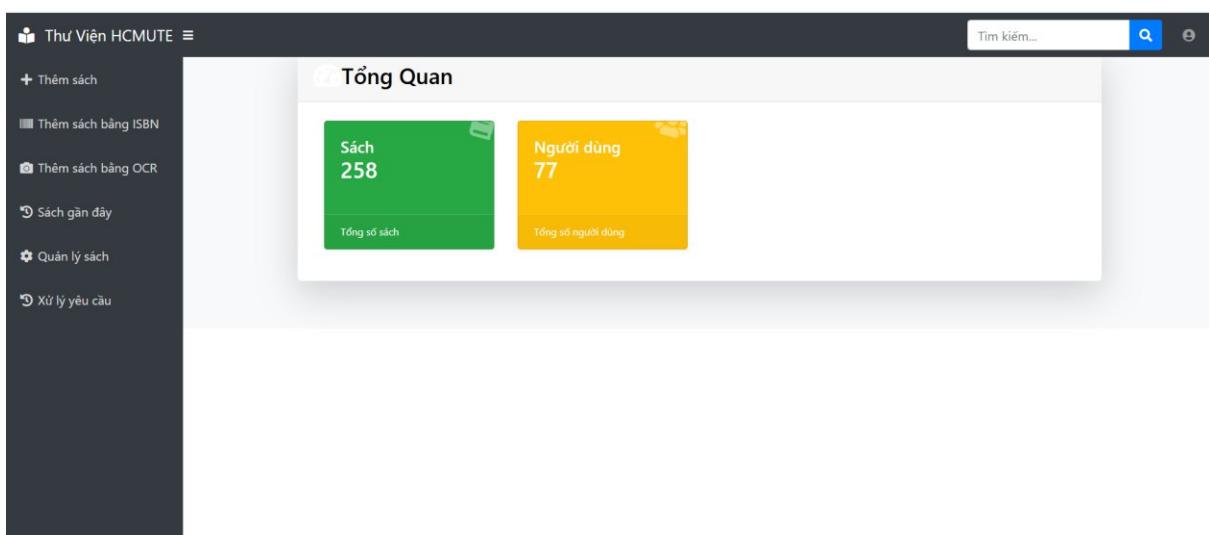
Hình 36: Giao diện trang chatbot tư vấn sách

2.5.7. Trang gửi yêu cầu xóa sách



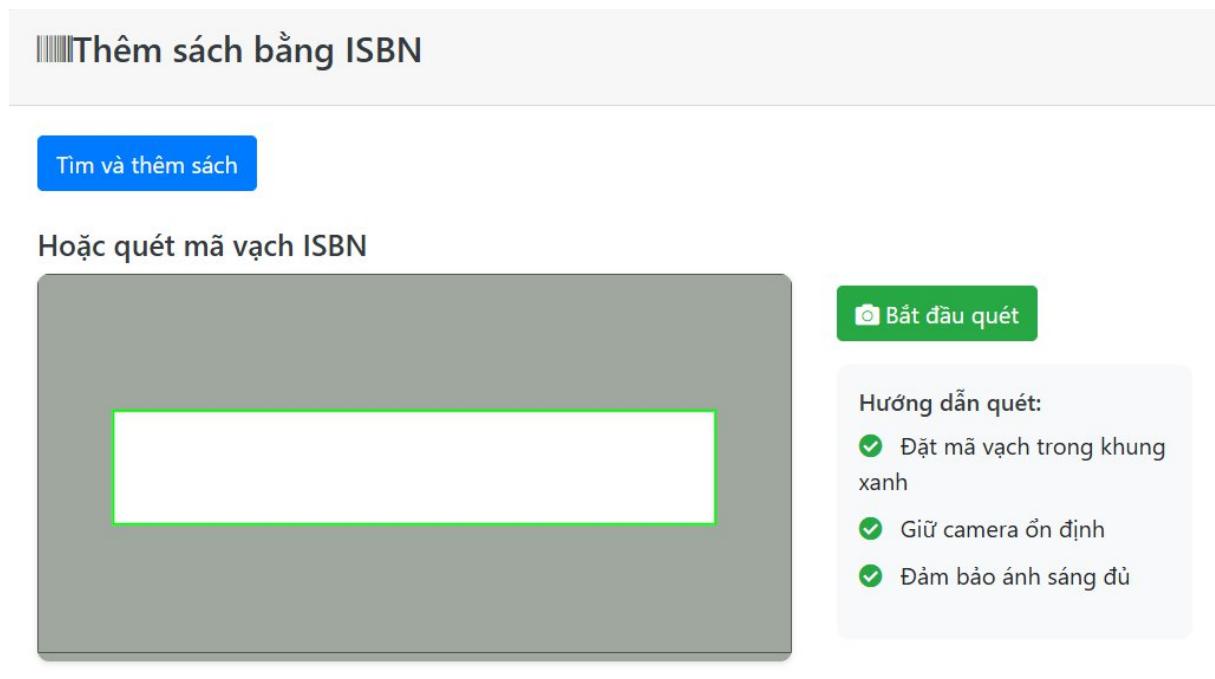
Hình 37: Giao diện trang gửi yêu cầu xóa sách

2.5.8. Trang quản lý của thủ thư



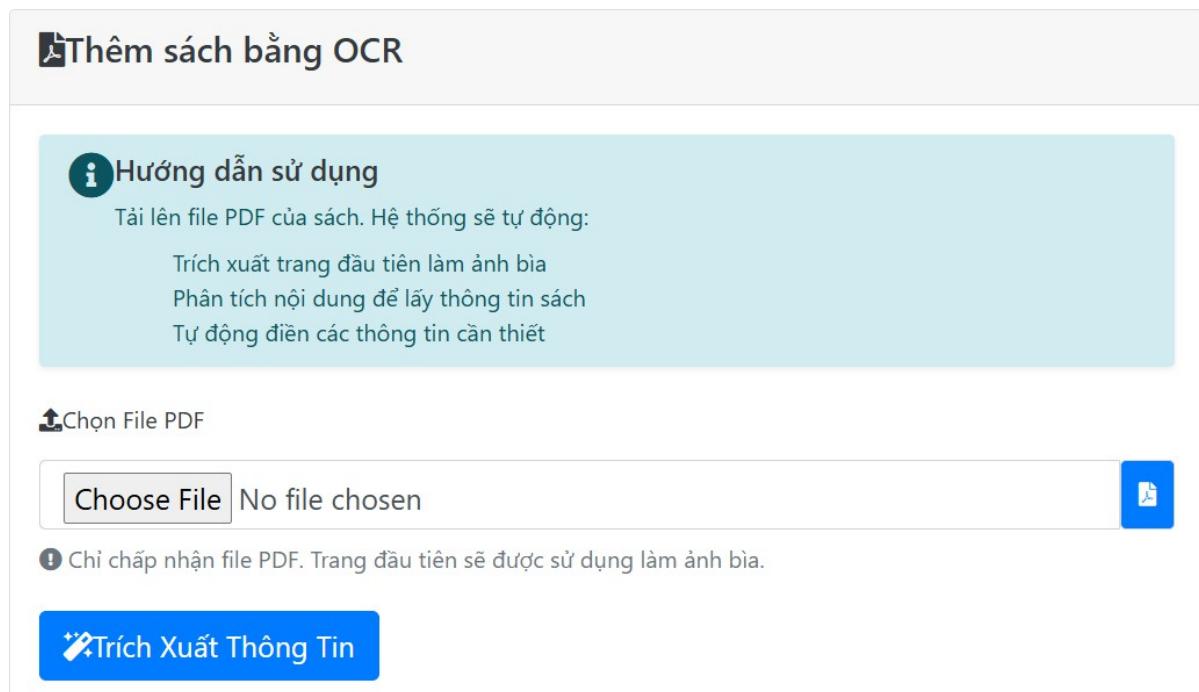
Hình 38: Giao diện trang quản lý của thủ thư

2.5.9. Trang thêm sách bằng mã ISBN của thủ thư



Hình 39: Giao diện trang thêm sách bằng mã ISBN

2.5.10. Trang thêm sách bằng OCR của thủ thư



Hình 40: Giao diện trang thêm sách bằng OCR

2.5.11.Trang quản lý sách của thủ thư

Mã Sách	Ảnh Bìa	Tên Sách	Xem	Sửa	Xóa
248		Vận hành hệ thống điện	<button>Xem</button>	<button>Sửa</button>	<button>Xóa</button>
247		Tính toán kỹ thuật điện đơn giản	<button>Xem</button>	<button>Sửa</button>	<button>Xóa</button>
246		Trang bị điện tử Máy gia công kim loại	<button>Xem</button>	<button>Sửa</button>	<button>Xóa</button>

Hình 41: Giao diện trang quản lý sách của thủ thư

2.5.12.Trang xử lý yêu cầu của thủ thư

Danh sách yêu cầu xóa sách
<p>Yêu cầu xóa: buiquangthien muốn xóa sách có mã 273</p>

Hình 42: Giao diện trang xử lý yêu cầu của thủ thư

CHƯƠNG 3: HUẤN LUYỆN VÀ TINH CHỈNH HỆ THỐNG

3.1. DỮ LIỆU SỬ DỤNG

3.1.1. Tập dữ liệu các khuôn mặt

Tên tập dữ liệu: PINS (Personalities IN Social Media) [29]

Mục đích sử dụng: Tập dữ liệu được sử dụng trong đề tài nhằm phục vụ nhiệm vụ nhận dạng khuôn mặt sinh viên trong hệ thống thư viện thông minh. Cụ thể, hệ thống sử dụng nhận dạng khuôn mặt để hỗ trợ đăng nhập tự động cho người dùng.

Sửa đổi: Trong quá trình triển khai, nhóm đã thực hiện đổi tên thư mục và nhãn dữ liệu từ tên người nổi tiếng gốc sang tên sinh viên kèm mã số sinh viên để phù hợp với ứng dụng thực tế. Việc thay đổi tên thư mục không ảnh hưởng đến hiệu suất hay chất lượng của hệ thống nhận diện.

Nguồn gốc: Tập dữ liệu PINS được thu thập từ nhiều nguồn công khai trên internet như Pinterest, IMDb, Wikipedia, và một số mạng xã hội khác.

Cấu trúc dữ liệu:

- Số lớp (classes): 77 (tương ứng với 77 người khác nhau).
- Tổng số ảnh: ~17.710 ảnh.
- Trung bình mỗi lớp: Khoảng 40–60 ảnh.
- Độ phân giải ảnh: Không đồng nhất, phổ biến trong khoảng 100×100 đến 300×300 pixels.
- Định dạng ảnh: jpg

Phân chia dữ liệu:

- Tập huấn luyện (Train): 80% ảnh (≈ 14.176 ảnh).
- Tập kiểm tra (Test): 20% ảnh (≈ 3.544 ảnh).
- Ảnh được chọn ngẫu nhiên từ từng lớp để đảm bảo phân bố đều trong cả hai tập.

3.1.2. Tập dữ liệu sách

Nguồn dữ liệu: tập dữ liệu được thu thập từ Thư viện số Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh [30].

Phương pháp thu thập: dữ liệu được thu thập tự động bằng phương pháp web crawling. Cụ thể, hệ thống crawler thực hiện quét toàn bộ nội dung của thư viện số, bao gồm danh sách tài liệu, siêu dữ liệu mô tả (metadata), và các tệp đính kèm (file PDF). Quá trình thu thập được thực hiện theo cơ chế duyệt tuần tự theo từng chuyên ngành/khoa.

Phạm vi dữ liệu: tập dữ liệu bao gồm toàn bộ tài liệu học thuật được phân loại theo 8 khoa hoặc chuyên ngành, với nội dung chủ yếu là sách, luận văn, giáo trình, báo cáo, và các tài liệu tham khảo chuyên ngành.

Quy mô dữ liệu

- Tổng số tài liệu: xấp xỉ 250 đầu sách và tài liệu.
- Phân bổ theo khoa: Mỗi khoa/chuyên ngành có trung bình khoảng 40 tài liệu, dao động tùy theo mức độ phong phú của tài nguyên học thuật từng đơn vị.

Cấu trúc dữ liệu: dạng tệp PDF chứa các trang mở đầu của sách

3.2. LÍ DO CHỌN VÀ ĐẶC ĐIỂM CỦA DỮ LIỆU

3.2.1. Tập dữ liệu khuôn mặt

Bảng 54: Mô tả tập dữ liệu khuôn mặt

Tiêu chí	Ưu điểm	Nhược điểm
Phạm vi dữ liệu	- Bao gồm 77 người nổi tiếng đa dạng, phù hợp cho bài toán nhận dạng khuôn mặt.	- Thiếu đa dạng chủng tộc, độ tuổi (chủ yếu người da trắng, trung niên).
Số lượng ảnh	- ~17,000 ảnh, trung bình 40-60 ảnh/class, đủ để huấn luyện mô hình CNN.	- Mất cân bằng: một số lớp có ít ảnh hơn (<40), gây bias trong phân loại.
Chất lượng ảnh	- Ảnh chân dung thực tế từ mạng xã hội (Pinterest, IMDb), đa góc chụp, biểu cảm.	- Độ phân giải không đồng nhất (100x100 – 300x300), một số ảnh mờ/nhiều.
Tính ứng dụng	- Phù hợp cho face recognition, transfer learning (ResNet, FaceNet).	- Không có annotation chi tiết (landmark, bounding box), khó áp dụng cho bài toán face alignment.
Nguồn gốc	- Dữ liệu công khai từ mạng xã hội, dễ tiếp cận.	- Vấn đề bản quyền: chưa rõ giấy phép sử dụng.

Độ phức tạp	- Background đa dạng (sự kiện, tự nhiên), giúp mô hình học được đặc trưng robust.	- Nhiều ảnh bị che khuất (kính râm, tóc) hoặc ánh sáng yếu.
Cập nhật	- Phản ánh xu hướng người nổi tiếng trên truyền thông.	- Không được bổ sung thường xuyên, có thể lỗi thời theo thời gian.

3.2.2. Tập dữ liệu sách

Bảng 55: Mô tả tập dữ liệu sách

Tiêu chí	Ưu điểm	Nhược điểm
Phạm vi dữ liệu	- Bao phủ 8 khoa/chuyên ngành đa dạng (kỹ thuật, kinh tế, xã hội).	- Một số ngành ít tài liệu (VD: Nông nghiệp, Y học).
Số lượng	~250 tài liệu, đủ lớn để phân tích.	Phân bố không đều: CNTT, Điện-Điện tử chiếm đa số (>150), các ngành khác <100.
Loại tài liệu	Đa dạng: giáo trình, luận văn, báo cáo, tài liệu tham khảo.	Thiếu tài liệu đặc thù (VD: kỹ yếu hoi thảo chỉ có 10 tài liệu).
Metadata	Có cấu trúc rõ ràng (tiêu đề, tác giả, năm xuất bản, từ khóa).	Một số metadata thiếu hoặc không đồng nhất (thiếu năm xuất bản, tác giả).
Tính ứng dụng	Phù hợp xây dựng hệ thống tìm kiếm, gợi ý sách, phân tích xu hướng.	Khó áp dụng cho nghiên cứu chuyên sâu do hạn chế về chuyên ngành nhỏ.
Truy cập	Dữ liệu công khai, có thể crawl từ website.	Một số file yêu cầu đăng nhập để tải về.

Cập nhật	Tập trung vào lĩnh vực "hot" (CNTT, Điện - Điện tử).	Dữ liệu cũ, thiếu tài liệu mới trong các ngành mới nổi (AI, Block-chain).
Chất lượng dữ liệu	Nguồn tin cậy từ thư viện trường đại học.	Chưa được làm sạch (có thể tồn tại trùng lặp, nhiễu).

3.3. TINH CHỈNH THAM SỐ

3.3.1. Tham số mô hình LLM Gemma 2.9b IT GGUF

Bảng 56: Tham số mô hình Gemma 2.9b IT GGUF

Hyperparameter	Tên biến	Mô tả	Giá trị
Temperature	temperature	Điều chỉnh mức độ sáng tạo của mô hình khi sinh đầu ra.	0.3
Number of context tokens	n_ctx	Số lượng token ngữ cảnh mà mô hình có thể xử lý trong một lần chạy.	8192
Maximum number of tokens	max_tokens	Giới hạn số lượng token đầu ra mà mô hình có thể sinh ra.	2500
Number of GPU layers	n_gpu_layers	Quy định số lớp đầu tiên của mô hình sẽ được tải lên GPU để tăng tốc xử lý.	90

3.3.2. Tham số mô hình embedding sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Bảng 57: Tham số mô hình embedding sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Hyperparameter	Tên biến	Mô tả	Giá trị
Mô hình embedding	model_name	Tên mô hình dùng để mã hóa văn bản thành vector ngữ nghĩa	"sentence-transformers/paraphrase-multilingual-mpnet-base-v2"
Số câu tối đa mỗi đoạn	max_sentences	Số lượng câu tối đa trong một chunk văn bản	10
Số câu chồng lặp giữa các đoạn	overlap	Số lượng câu sẽ được lặp lại giữa các chunk để giữ ngữ cảnh	3

3.3.3. Tham số mô hình cho hệ thống gợi ý sách

Bảng 58: Tham số mô hình cho hệ thống gợi ý sách

Hyperparameter	Giá trị	Ý nghĩa
Learning Rate	0.0001	Tốc độ cập nhật trọng số mô hình trong mỗi bước huấn luyện.
Gamma	0.95	Hệ số chiết khấu cho phần thưởng tương lai
epsilon	0.1	Xác suất thực hiện hành động ngẫu nhiên (exploration).
batch_size	128	Kích thước của một minibatch trong huấn luyện.
memory	10000	Dung lượng tối đa của bộ nhớ kinh nghiệm (experience replay).
Số lượng nơ-ron	256/lớp	Số lượng nodes ở từng lớp ẩn
Số lớp hidden layer	2	Số lượng lớp ẩn
Episodes	200	Số lần huấn luyện với các

		người dùng khác nhau.
input_dim	384+1	Số chiều đầu vào cho mạng DQN (gồm embedding sách + user_id).
output_dim	len(all_books)	Số chiều đầu ra (tương ứng với tổng số sách có thể gợi ý)

CHƯƠNG 4: ĐÁNH GIÁ HỆ THỐNG

4.1. ĐÁNH GIÁ HỆ THỐNG NHẬN DIỆN KHUÔN MẶT

4.1.1. Mục tiêu đánh giá

- Đánh giá độ chính xác của mô hình xem xét sự phù hợp cho chức năng đăng nhập của ứng dụng.
- Đánh giá thời gian xử lý trung bình của hệ thống từ đó suy ra sự phù hợp cho hệ thống yêu cầu phản hồi thời gian thực.
- Đánh giá điểm tương đồng cosine similarity nào là tối ưu nhất cho hệ thống, đảm bảo tính bảo mật và khả năng dự đoán chính xác.

4.1.2. Thiết kế hệ thống đánh giá

Tập dữ liệu

Lấy ra ngẫu nhiên 20% số lượng từ tập dataset 105_classes_pins_dataset , tức khoảng 3500 ảnh để đánh giá.

Các phương pháp đánh giá

Hệ thống đánh giá được thiết kế dựa trên nhiều tiêu chí khác nhau như:

- Các metrics cho bài toán phân loại ảnh: Accuracy, F1-Score, Precision, Recall.
- Giá trị trung bình thời gian xử lý của hệ thống nhận diện khuôn mặt.
- Phân bố điểm tương đồng cosine similarity giữa 2 nhóm dự đoán đúng và dự đoán sai.

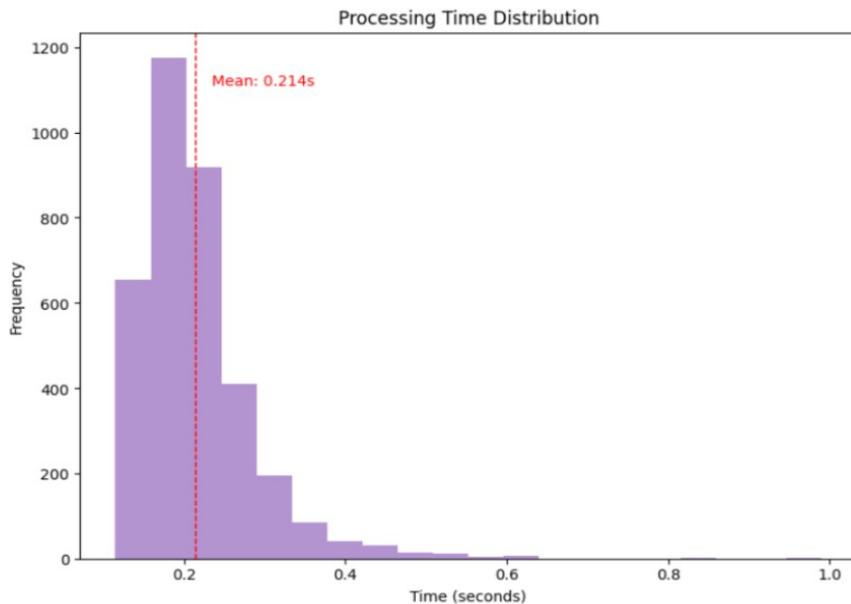
4.1.3. Kết quả



Hình 43: Kết quả hệ thống nhận diện khuôn mặt qua các Metrics

Tất cả metrics đều đạt trên 98%, vượt xa ngưỡng 90% thường được coi là tốt trong các hệ thống nhận dạng khuôn mặt. Sự đồng đều giữa các metrics cho thấy mô hình cân bằng giữa các khía cạnh đánh giá

Sự chênh lệch chỉ 0.002 giữa các metrics, là do chất lượng dữ liệu huấn luyện tốt, ngưỡng similarity được tối ưu.

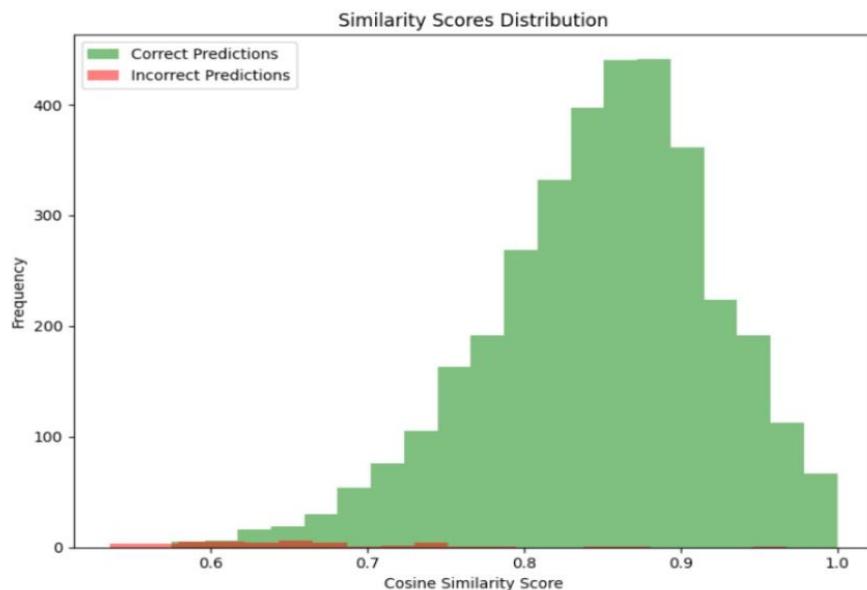


Hình 44: Thời gian trung bình hệ thống nhận diện khuôn mặt

Giá trị trung bình của thời gian xử lý là 0.214 giây. Phần lớn các truy vấn được xử lý trong khoảng 0.1 – 0.25 giây, thể hiện khả năng xử lý nhanh chóng và ổn định

của hệ thống. Một số ít trường hợp có thời gian xử lý lớn hơn, nhưng chiếm tỷ lệ không đáng kể.

Mô hình đảm bảo hiệu năng xử lý cao và có thể ứng dụng trong các hệ thống yêu cầu phản hồi gần thời gian thực.



Hình 45: Phân bố Cosine Similarity trong chức năng nhận diện khuôn mặt

Biểu đồ histogram so sánh phân bố điểm tương đồng cosine giữa hai nhóm: Dự đoán đúng - màu xanh lá, dự đoán sai - màu đỏ xét trên thang điểm cosine similarity từ 0.6 đến 1.0.

Ta thấy xu hướng chính phân tách rõ ràng giữa 2 nhóm:

- Dự đoán đúng tập trung ở khoảng 0.85-1.0: cao.
- Dự đoán sai chủ yếu ở khoảng 0.6-0.8: thấp hơn rõ rệt.

Giữa 2 nhóm có độ chênh lấp thấp: Chỉ xảy ra ở khoảng 0.75-0.85 qua đó ta thấy hệ thống có ngưỡng phân tách tốt.

Nếu đặt threshold tại 0.85 thì True Positive Rate $\approx 98\%$ chứng tỏ hầu hết điểm đúng >0.85 và False Positive Rate $\approx 10\%$ chứng tỏ một ít điểm sai >0.85 .

Độ nhạy hệ thống: Khoảng cách lớn giữa 2 peak (0.95 vs 0.7) cho thấy model học được đặc trưng rất riêng biệt

Các trường hợp sai có điểm >0.8 thường do:

- Người trong ảnh có đặc điểm gần giống nhau: anh em sinh đôi, cùng dân tộc, chủng tộc,...

- Điều kiện ảnh xấu: ánh sáng yếu, góc nghiêng >45 độ, background phức tạp.
- Độ phân giải thấp: các ảnh có pixel nhỏ hơn 50px giữa 2 mắt.

4.2. ĐÁNH GIÁ HỆ THỐNG CHATBOT TƯ VẤN SÁCH

4.2.1. Mục tiêu đánh giá

Mục tiêu của quá trình đánh giá này là:

- Đánh giá hiệu quả của chatbot trong việc hiểu và phản hồi chính xác các truy vấn của người dùng liên quan đến sách trong thư viện.
- Xác định khả năng truy xuất thông tin chính xác của chatbot trong 2 tình huống phổ biến:
 - o Người dùng đã biết tên sách và cần hỏi về nội dung, tác giả, ứng dụng,...
 - o Người dùng chưa biết sách nào, chỉ nêu ra chủ đề quan tâm (ví dụ: "giải thuật", "mạng máy tính",...).
- Đo lường hiệu quả của pipeline RAG hiện tại (OCR + Chroma + Sentence-Transformer + LLM) trong việc phục vụ truy vấn.

4.2.2. Thiết kế hệ thống đánh giá

Hệ thống đánh giá được xây dựng gồm 2 thành phần chính:

- Hệ thống sinh dữ liệu đầu vào (câu hỏi và câu trả lời từ mô hình).
- Hệ thống chấm điểm tự động, sử dụng một mô hình LLM độc lập để đánh giá độ đúng/sai của các câu trả lời dựa trên ngữ cảnh thực tế (được trích xuất từ CSDL ChromaDB).

4.2.2.1. Tập câu hỏi đánh giá

Tập đánh giá bao gồm tổng cộng 99 câu hỏi, chia thành 2 nhóm:

Nhóm 1: Câu hỏi liên quan đến sách cụ thể (49 câu)

Trong nhóm này, người dùng đã biết rõ tên sách và đặt các câu hỏi chi tiết liên quan đến nội dung hoặc ứng dụng thực tế của sách. Mục tiêu của nhóm câu hỏi này là đánh giá khả năng truy xuất thông tin của chatbot dựa trên dữ liệu lời mở đầu đã được lưu trữ trong cơ sở dữ liệu (ChromaDB).

Để đảm bảo tính đại diện và khách quan, các câu hỏi được sinh ra bằng phương pháp bắn tự động theo quy trình sau:

- Hệ thống có sẵn danh sách khoảng 250 đầu sách đã được xử lý dữ liệu phân lời mở đầu.

- Từ danh sách này, hệ thống chọn ngẫu nhiên 7 cuốn sách làm đại diện cho tập câu hỏi.
- Với mỗi cuốn sách được chọn, hệ thống sử dụng một trong 7 mẫu câu hỏi (template) sau để tạo thành 7 câu hỏi khác nhau.

Danh sách 7 mẫu câu hỏi sử dụng trong đánh giá:

1. "Sách {tên sách} có nội dung chính là gì?": đánh giá khả năng chatbot tóm tắt được trọng tâm của cuốn sách.
2. "Cuốn {tên sách} nói về vấn đề gì?": tương tự như câu 1 nhưng sử dụng cách diễn đạt khác để đánh giá khả năng hiểu ngôn ngữ đa dạng.
3. "Sách {tên sách} phù hợp với đối tượng nào?": kiểm tra khả năng suy luận và trích xuất thông tin liên quan đến độc giả mục tiêu.
4. "Cuốn sách {tên sách} có ứng dụng thực tế ra sao?": đánh giá khả năng mô tả các ứng dụng hoặc phạm vi áp dụng của kiến thức trong sách.
5. "Sách {tên sách} đề cập đến công nghệ nào?": kiểm tra khả năng xác định các khái niệm kỹ thuật hoặc công nghệ liên quan trong nội dung sách.
6. "Bạn hãy tóm tắt sách {tên sách} cho tôi.": đánh giá khả năng tổng hợp và diễn đạt ngắn gọn toàn bộ nội dung của cuốn sách.
7. "Sách {tên sách} có phù hợp cho người mới học không?": đánh giá khả năng phân tích mức độ phức tạp và yêu cầu đầu vào của sách đối với người học.

Tổng cộng, nhóm này tạo ra 49 câu hỏi (7 cuốn sách \times 7 mẫu câu hỏi), đảm bảo độ đa dạng và tính đại diện cao trong quá trình đánh giá hiệu quả truy xuất thông tin của hệ thống chatbot.

Nhóm 2: Câu hỏi gợi ý theo chủ đề (50 câu)

Trong nhóm này, người dùng không chỉ định rõ tên sách, mà chỉ nêu ra chủ đề hoặc lĩnh vực quan tâm, với mục tiêu đề nghị hệ thống chatbot tư vấn sách phù hợp. Đây là một tình huống phổ biến trong môi trường thư viện thực tế, khi người dùng không nhớ rõ tiêu đề mà chỉ biết chủ đề cần tra cứu.

Phương pháp xây dựng câu hỏi

- Dữ liệu đầu vào được xây dựng dựa trên danh sách các khoa hoặc bộ môn tại một trường đại học, mỗi khoa được xem như một chủ đề học thuật cụ thể.
- Với mỗi chủ đề này, hệ thống tạo ngẫu nhiên 1 câu hỏi yêu cầu chatbot đề xuất sách phù hợp.

- Tổng cộng có 50 câu hỏi, tương ứng với 50 chủ đề thuộc các khoa khác nhau như: Công nghệ thông tin, Xây dựng, Cơ khí chế tạo máy, Cơ khí động lực,...

Một số ví dụ về dạng câu hỏi trong nhóm này:

- "Có sách nào hướng dẫn tự học lập trình C từ cơ bản không?"
- "Tôi cần tài liệu học lập trình Python, bạn có gợi ý nào không?"
- "Có cuốn sách nào phù hợp với sinh viên ngành môi trường không?"
- "Tôi muốn học tính toán truyền động cơ khí, bạn gợi ý sách nào?"

Mục tiêu đánh giá:

- Kiểm tra khả năng của chatbot trong việc tìm kiếm và đề xuất sách phù hợp dựa trên ngữ nghĩa chủ đề, thay vì phụ thuộc vào từ khóa trùng khớp với tiêu đề.
- Đánh giá năng lực của hệ thống RAG (Retrieval-Augmented Generation) trong việc trích xuất thông tin liên quan từ lời mở đầu sách, sau đó kết hợp cùng mô hình ngôn ngữ lớn (LLM) để tạo ra phản hồi gợi ý phù hợp.

4.2.2.2. Phương pháp đánh giá

Để đảm bảo tính khách quan và tái lập trong đánh giá, hệ thống sử dụng mô hình ngôn ngữ lớn LLaMA-3 8B Instruct (truy cập thông qua nền tảng OpenRouter) đóng vai trò như một "trọng tài" đánh giá chất lượng câu trả lời do chatbot sinh ra.

Cơ chế hoạt động:

Với mỗi cặp câu hỏi – câu trả lời được sinh bởi hệ thống, một đoạn văn bản ngữ cảnh (context) được truy xuất từ cơ sở dữ liệu ChromaDB, bao gồm tiêu đề, tác giả, và lời mở đầu của các sách liên quan. Trong trường hợp không có sách nào phù hợp, một thông báo rỗng được chèn vào để mô hình đánh giá biết rằng không có dữ liệu liên quan đến truy vấn.

Mô hình đánh giá sẽ nhận vào ba thành phần chính:

- Ngữ cảnh sách liên quan (nếu có).
- Câu hỏi của người dùng.
- Câu trả lời do chatbot sinh ra.

Với mỗi câu hỏi trong tập đánh giá: Pipeline sinh câu trả lời từ hệ thống RAG (gồm OCR, ChromaDB, Sentence-Transformer và LLM).

Prompt dùng cho mô hình đánh giá:

Systemprompt:

"Bạn là một hệ thống đánh giá các câu trả lời của chatbot. Hãy đọc câu hỏi và câu trả lời, rồi trả về 1 nếu câu trả lời đúng hoặc hợp lý, 0 nếu sai hoặc không liên quan. Chỉ trả về 0 hoặc 1, không cần giải thích."

Userprompt:

Dữ liệu sách liên quan:

{context_text}

Câu hỏi: {question}

Câu trả lời của chatbot: {answer}

Đánh giá:

Trong context_text, mỗi sách sẽ được hiển thị như sau:

[THÔNGTINSÁCH]

Tiêu đề: Tên sách A

[PHÀNLỜIMỞĐẦU]

Đoạn 1...

Đoạn 2...

...

Nếu hệ thống không tìm thấy dữ liệu sách liên quan trong cơ sở dữ liệu, phần context_text sẽ được đặt là: "Không có dữ liệu sách liên quan đến câu hỏi này"

4.2.3. Kết quả đánh giá

Sau khi thực hiện đánh giá tự động 99 câu hỏi bằng hệ thống RAG tích hợp LLM chấm điểm (LLaMA-3 8B Instruct), kết quả như sau:

Bảng 59: Kết quả đánh giá Chatbot

Nhóm câu hỏi	Số lượng câu hỏi	Số câu đúng	Tỉ lệ chính xác
Câu hỏi liên quan đến sách cụ thể	49	42	85.7%
Câu hỏi gợi ý sách theo chủ đề	50	42	84%
Tổng cộng	99	84	84.8%

Nhóm 1 – Câu hỏi liên quan đến sách cụ thể: Hệ thống đạt độ chính xác 85.7%, cho thấy khả năng hiểu đúng câu hỏi và truy xuất thông tin phù hợp từ lời mở đầu của sách là khá ổn định. Hầu hết các lỗi sai đến từ các trường hợp câu hỏi quá chi tiết hoặc yêu cầu thông tin không có trong phần dữ liệu lưu trữ (ví dụ: hỏi về phần nội dung giữa hoặc cuối sách, trong khi chỉ có lời mở đầu).

Nhóm 2 – Gợi ý sách theo chủ đề: Với độ chính xác 84%, chatbot cho thấy khả năng gợi ý sách phù hợp với chủ đề chung là khá tốt. Những câu trả lời đúng thường liên quan đến những sách có lời mở đầu mang tính khái quát cao, bao phủ các khái niệm chung về lĩnh vực đó. Một số lỗi xảy ra khi chủ đề quá rộng hoặc có ít sách tương ứng trong cơ sở dữ liệu.

Toàn bộ kết quả đánh giá đã được ghi nhận và lưu trữ dưới dạng hai tệp Excel riêng biệt, tương ứng với hai nhóm câu hỏi đánh giá:

- **Tệp DanhgiaChatbot_Nhom01.xlsx:** Chứa kết quả đánh giá cho nhóm 49 câu hỏi liên quan đến sách cụ thể.
- **Tệp DanhgiaChatbot_Nhom02.xlsx:** Chứa kết quả đánh giá cho nhóm 50 câu hỏi gợi ý sách theo chủ đề.

Mỗi tệp đều gồm ba cột chính:

- question: Câu hỏi được người dùng đặt ra cho chatbot.
- answer: Câu trả lời do hệ thống chatbot sinh ra cho câu hỏi đó.
- evalution: Giá trị đánh giá do hệ thống chấm điểm trả về (1 = đúng, 0 = sai)

The screenshot shows a Microsoft Excel spreadsheet titled "Sheet1". The table has three columns: "question", "answer", and "evaluation". The "question" column contains 27 different search queries. The "answer" column contains responses from a chatbot to these queries. The "evaluation" column contains numerical values ranging from 0 to 1, likely representing the quality or relevance of the generated answers. The Excel ribbon at the top includes tabs for File, Home, Insert, Draw, Page Layout, Formulas, Data, Review, View, Automate, Help, and Power Pivot. The "Home" tab is selected.

Hình 46: File Excel kết quả đánh giá chatbot

4.3. ĐÁNH GIÁ HỆ THỐNG OCR

4.3.1. Mục tiêu đánh giá

Mục tiêu của quá trình đánh giá là đo lường mức độ chính xác và đầy đủ của hệ thống OCR trong việc chuyển đổi lời mở đầu từ định dạng ảnh sang văn bản số, cụ thể là:

- Đánh giá khả năng nhận dạng từ vựng của mô hình OCR.
- Đánh giá mức độ bảo toàn nội dung ý nghĩa giữa bản OCR và văn bản gốc.
- Đánh giá khả năng giữ nguyên cấu trúc câu chữ và độ bao phủ nội dung.

Qua đó, xác định được mức độ phù hợp của mô hình OCR hiện tại cho nhiệm vụ xử lý sách PDF trong hệ thống thư viện thông minh.

4.3.2. Thiết kế hệ thống đánh giá

4.3.2.1. Tập dữ liệu

Tập đánh giá gồm 40 file PDF chứa lời mở đầu của 40 đầu sách. Sau quá trình trích xuất bằng hệ thống OCR, thu được 40 file văn bản kết quả. Để đánh giá chính xác, nhóm nghiên cứu đã chuẩn bị bộ 40 file văn bản gốc (Ground Truth) tương ứng, được xử lý thủ công nhằm đảm bảo độ chính xác tuyệt đối.

4.3.2.1. Các phương pháp đánh giá

Hệ thống đánh giá được thiết kế dựa trên nhiều tiêu chí khác nhau nhằm phản ánh toàn diện chất lượng đầu ra của OCR. Ba phương pháp chính được sử dụng như sau:

a. Đánh giá dựa trên ROUGE Scores

Chỉ số ROUGE (Recall-Oriented Understudy for Gisting Evaluation) là một bộ chỉ số đánh giá mức độ trùng khớp giữa hai văn bản, thường được sử dụng trong đánh giá tóm tắt tự động và dịch máy. Trong nghiên cứu này, ba biến thể chính của ROUGE được sử dụng:

- **ROUGE-1**: đo lường mức độ trùng khớp theo từ đơn (unigram).
- **ROUGE-2**: đo mức độ trùng khớp theo cụm từ hai từ liên tiếp (bigram).
- **ROUGE-L**: đo độ dài chuỗi con chung dài nhất giữa hai văn bản (Longest Common Subsequence).

Các chỉ số ROUGE giúp đánh giá khả năng của OCR trong việc giữ lại đúng từ vựng và trật tự từ trong văn bản.

b. Đánh giá độ bao phủ từ vựng (Vocabulary Coverage)

Độ bao phủ từ vựng là phương pháp đánh giá khả năng của hệ thống OCR trong việc tái hiện lại chính xác các đơn vị từ vựng trong văn bản gốc. Trong nghiên cứu này, độ bao phủ được tính bằng tỷ lệ số lượng từ đơn (unigram) và cặp từ liên tiếp (bigram) trong kết quả OCR khớp với văn bản gốc. Chỉ số này giúp phản ánh mức độ trùng khớp bề mặt giữa hai văn bản và được tính toán theo công thức:

- Unigram coverage = (Số unigram trùng khớp / Tổng số unigram trong ground truth).
- Bigram coverage = (Số bigram trùng khớp / Tổng số bigram trong ground truth).

Phương pháp này giúp kiểm tra xem hệ thống OCR có nhận diện được đúng các từ và cụm từ trong tài liệu gốc hay không, từ đó phản ánh độ chính xác ở mức từ vựng mà không xét đến ngữ nghĩa.

c. Đánh giá theo chuỗi con chung dài nhất (LCS)

Phương pháp Longest Common Subsequence (LCS) đo độ dài chuỗi từ liên tục dài nhất xuất hiện trong cả hai văn bản theo đúng thứ tự. Điểm LCS được chuẩn hóa về thang điểm từ 0 đến 1:

- Điểm gần 1 thể hiện mức độ tương đồng cao giữa hai văn bản.

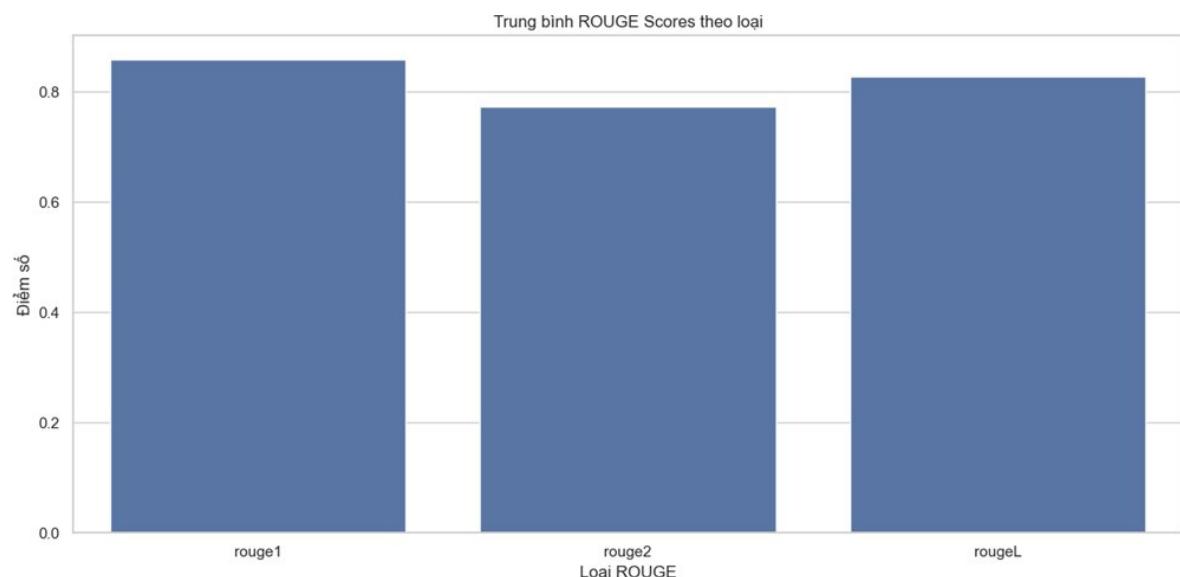
- Điểm thấp thể hiện có nhiều khác biệt về cấu trúc và trật tự thông tin.

Chỉ số này giúp kiểm tra khả năng giữ lại cấu trúc logic của đoạn văn OCR, đặc biệt có giá trị trong đánh giá các lỗi mất đoạn, nhảy dòng hoặc lặp lại từ.

4.3.3. Kết quả

Sau khi áp dụng ba phương pháp đánh giá (ROUGE, LCS, Semantic Similarity), kết quả được tổng hợp từ 40 cặp văn bản (đầu ra từ mô hình OCR và Ground Truth). Một số thống kê tổng quát được thể hiện như sau:

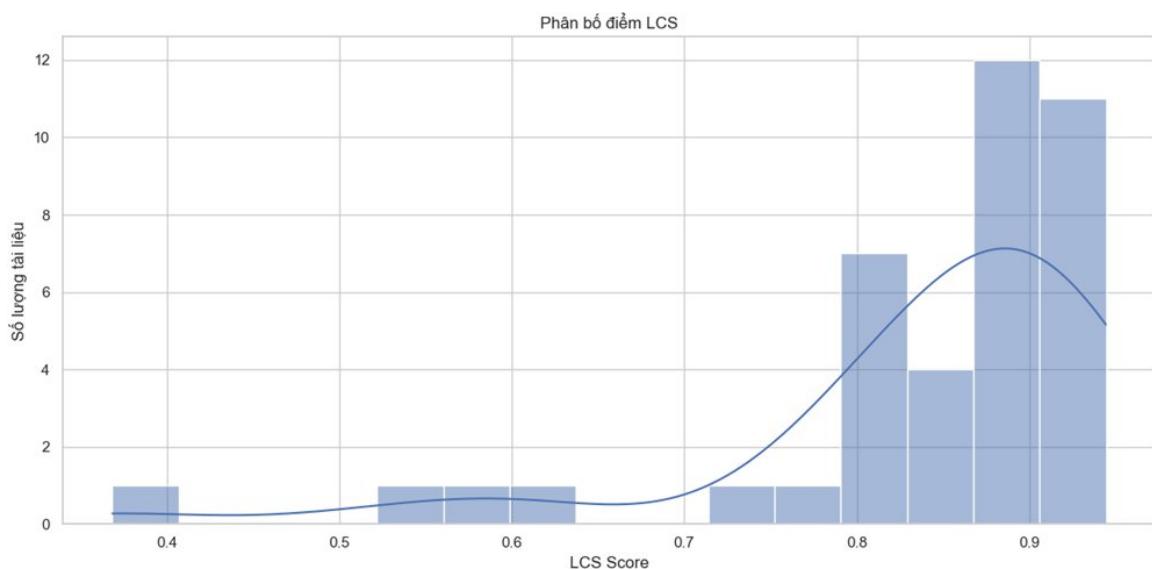
ROUGE Scores:



Hình 47: Kết quả ROUGE Scores trong hệ thống OCR

- Trung bình ROUGE-1: 0.831
- Trung bình ROUGE-2: 0.789
- Trung bình ROUGE-L: 0.823
- Các giá trị cao cho thấy mô hình OCR tái tạo khá tốt các từ/cụm từ trong văn bản gốc. Tuy nhiên, ROUGE-2 thấp hơn ROUGE-1 phản ánh hiện tượng sai sót trong chuỗi từ liền nhau.

Longest Common Subsequence (LCS):

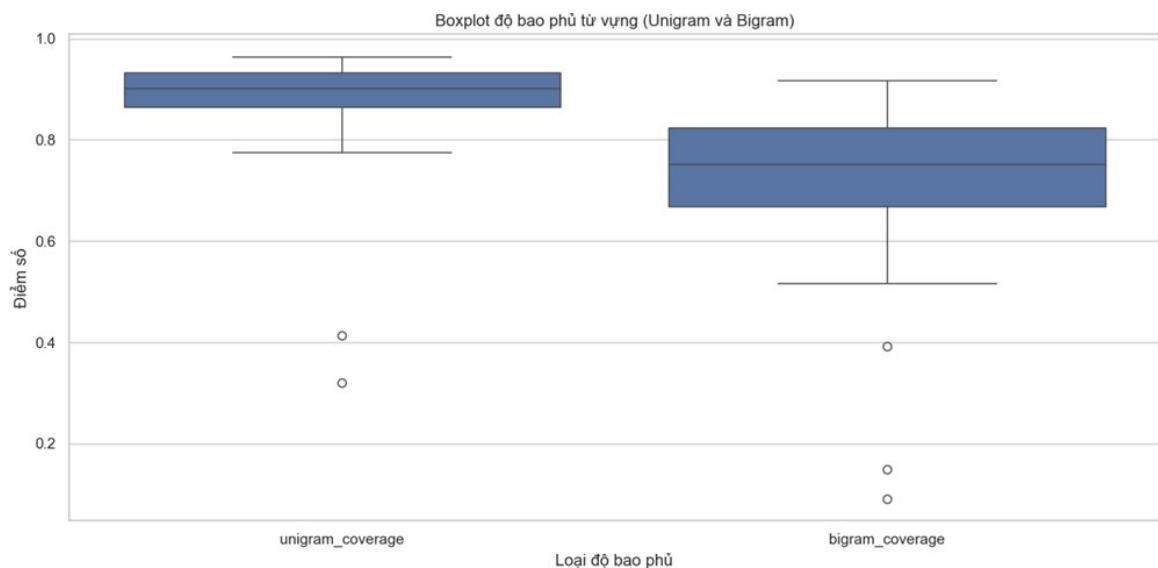


Hình 48: Kết quả LCS trong hệ thống OCR

- Trung bình điểm LCS: 0.812
- Điểm cao nhất: 0.953
- Điểm thấp nhất: 0.368

Qua kết quả này cho thấy một số tài liệu có độ khớp gần như hoàn hảo, nhưng một số khác gặp lỗi OCR đáng kể khiến thứ tự từ bị ảnh hưởng hoặc mất đoạn.

Độ bao phủ từ vựng (Coverage):



Hình 49: Kết quả về độ bao phủ từ vựng trong hệ thống OCR

- Unigram coverage (trùng từ đơn): Trung bình 0.845
- Bigram coverage (trùng cụm 2 từ): Trung bình 0.733

Kết quả này phản ánh rằng phần lớn từ trong văn bản gốc đã được tái hiện trong văn bản OCR, nhưng việc giữ đúng cụm từ liên tiếp vẫn còn hạn chế do lỗi OCR rải rác.

4.4. ĐÁNH GIÁ HỆ THỐNG GỢI Ý SÁCH

4.4.1. Mục tiêu đánh giá

Đo lường hiệu quả học của mô hình Deep Q-Learning thông qua tổng phần thưởng nhận được sau mỗi vòng lặp huấn luyện.

4.4.2. Phương pháp đánh giá

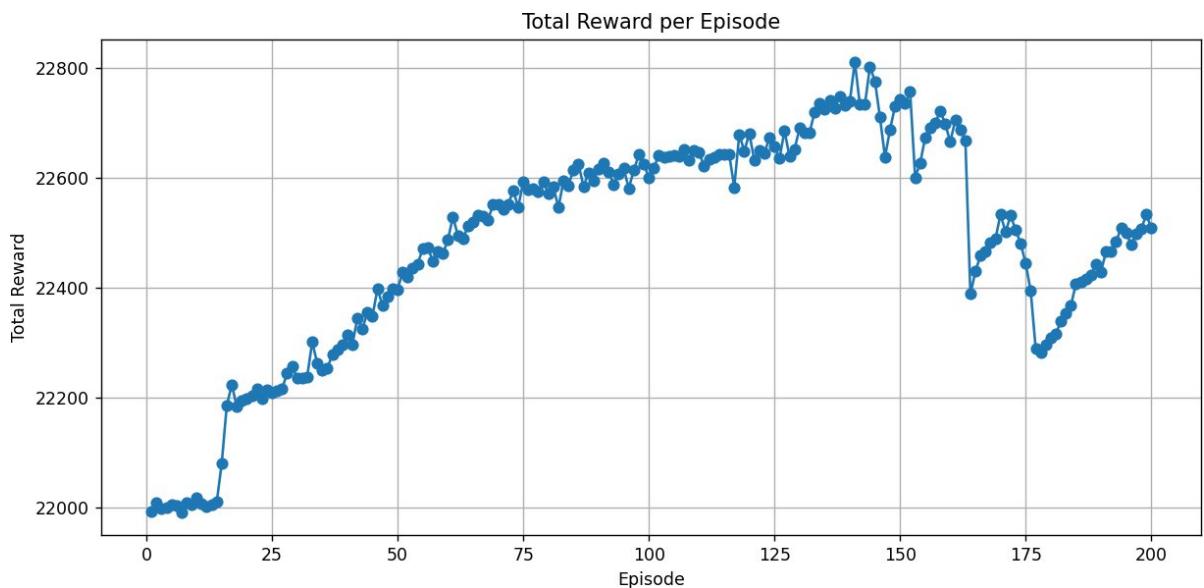
Phương pháp đánh giá: Vẽ đồ thị mối tương quan giữa phần thưởng qua các episode.

4.4.3. Kết quả

Bảng 60: Bảng kết quả hệ thống gợi ý sách qua các episode

Epoch 1: Total Reward = 21993.45	Epoch 34: Total Reward = 22263.17	Epoch 68: Total Reward = 22523.73
Epoch 2: Total Reward = 22009.46	Epoch 35: Total Reward = 22249.73	Epoch 69: Total Reward = 22552.23
Epoch 3: Total Reward = 21998.08	Epoch 36: Total Reward = 22254.02	Epoch 70: Total Reward = 22553.03
Epoch 4: Total Reward = 22001.18	Epoch 37: Total Reward = 22278.61	Epoch 71: Total Reward = 22543.81
Epoch 5: Total Reward = 22006.64	Epoch 38: Total Reward = 22288.38	Epoch 72: Total Reward = 22552.30
Epoch 6: Total Reward = 22003.47	Epoch 39: Total Reward = 22296.75	Epoch 73: Total Reward = 22577.62
Epoch 7: Total Reward = 21991.20	Epoch 40: Total Reward = 22315.35	Epoch 74: Total Reward = 22547.37
Epoch 8: Total Reward = 22008.94	Epoch 41: Total Reward = 22297.69	Epoch 75: Total Reward = 22593.36
Epoch 9: Total Reward = 22006.27	Epoch 42: Total Reward = 22344.97	Epoch 76: Total Reward = 22579.70
Epoch 10: Total Reward = 22018.93	Epoch 43: Total Reward = 22326.20	Epoch 77: Total Reward = 22580.27
Epoch 11: Total Reward = 22007.04	Epoch 44: Total Reward = 22355.73	Epoch 78: Total Reward = 22575.72
Epoch 12: Total Reward = 22002.24	Epoch 45: Total Reward = 22349.25	Epoch 79: Total Reward = 22593.69
Epoch 13: Total Reward = 22005.30	Epoch 46: Total Reward = 22398.80	Epoch 80: Total Reward = 22572.64
Epoch 14: Total Reward = 22010.34	Epoch 47: Total Reward = 22367.68	Epoch 81: Total Reward = 22584.86
Epoch 15: Total Reward = 22081.52	Epoch 48: Total Reward = 22384.47	Epoch 82: Total Reward = 22546.27
Epoch 16: Total Reward = 22186.35	Epoch 49: Total Reward = 22398.42	Epoch 83: Total Reward = 22594.42
Epoch 17: Total Reward = 22224.30	Epoch 50: Total Reward = 22397.50	Epoch 84: Total Reward = 22587.08
Epoch 18: Total Reward = 22183.63	Epoch 51: Total Reward = 22429.45	Epoch 85: Total Reward = 22615.36
Epoch 19: Total Reward = 22195.44	Epoch 52: Total Reward = 22420.53	Epoch 86: Total Reward = 22626.24
Epoch 20: Total Reward = 22198.66	Epoch 53: Total Reward = 22436.29	Epoch 87: Total Reward = 22584.72
Epoch 21: Total Reward = 22203.80	Epoch 54: Total Reward = 22444.25	Epoch 88: Total Reward = 22609.96
Epoch 22: Total Reward = 22217.14	Epoch 55: Total Reward = 22471.51	Epoch 89: Total Reward = 22595.36
Epoch 23: Total Reward = 22199.53	Epoch 56: Total Reward = 22473.81	Epoch 90: Total Reward = 22617.07
Epoch 24: Total Reward = 22214.07	Epoch 57: Total Reward = 22449.00	Epoch 91: Total Reward = 22626.73
Epoch 25: Total Reward = 22209.96	Epoch 58: Total Reward = 22465.87	Epoch 92: Total Reward = 22611.94
Epoch 26: Total Reward = 22212.79	Epoch 59: Total Reward = 22462.85	Epoch 93: Total Reward = 22588.68
Epoch 27: Total Reward = 22216.82	Epoch 60: Total Reward = 22487.25	Epoch 94: Total Reward = 22607.90
Epoch 28: Total Reward = 22244.36	Epoch 61: Total Reward = 22529.46	Epoch 95: Total Reward = 22617.81
Epoch 29: Total Reward = 22258.20	Epoch 62: Total Reward = 22495.94	Epoch 96: Total Reward = 22581.47
Epoch 30: Total Reward = 22235.96	Epoch 63: Total Reward = 22489.42	Epoch 97: Total Reward = 22614.24
Epoch 31: Total Reward = 22236.85	Epoch 64: Total Reward = 22512.72	Epoch 98: Total Reward = 22642.70
Epoch 32: Total Reward = 22238.72	Epoch 65: Total Reward = 22520.39	Epoch 99: Total Reward = 22625.61
Epoch 33: Total Reward = 22301.87	Epoch 66: Total Reward = 22532.02	Epoch 100: Total Reward = 22601.00
	Epoch 67: Total Reward = 22530.46	

Epoch 101: Total Reward = 22618.7	Epoch 134: Total Reward = 22736.4	Epoch 167: Total Reward = 22466.5
Epoch 102: Total Reward = 22642.2	Epoch 135: Total Reward = 22726.1	Epoch 168: Total Reward = 22483.31
Epoch 103: Total Reward = 22638.7	Epoch 136: Total Reward = 22742.1	Epoch 169: Total Reward = 22490.41
Epoch 104: Total Reward = 22639.5	Epoch 137: Total Reward = 22728.1	Epoch 170: Total Reward = 22534.13
Epoch 105: Total Reward = 22641.2	Epoch 138: Total Reward = 22749.4	Epoch 171: Total Reward = 22501.74
Epoch 106: Total Reward = 22639.8	Epoch 139: Total Reward = 22732.7	Epoch 172: Total Reward = 22532.15
Epoch 107: Total Reward = 22651.8	Epoch 140: Total Reward = 22739.5	Epoch 173: Total Reward = 22505.54
Epoch 108: Total Reward = 22632.2	Epoch 141: Total Reward = 22811.9	Epoch 174: Total Reward = 22481.33
Epoch 109: Total Reward = 22651.3	Epoch 142: Total Reward = 22733.7	Epoch 175: Total Reward = 22444.96
Epoch 110: Total Reward = 22647.5	Epoch 143: Total Reward = 22733.8	Epoch 176: Total Reward = 22394.81
Epoch 111: Total Reward = 22621.1	Epoch 144: Total Reward = 22803.0	Epoch 177: Total Reward = 22289.07
Epoch 112: Total Reward = 22634.5	Epoch 145: Total Reward = 22776.4	Epoch 178: Total Reward = 22282.45
Epoch 113: Total Reward = 22637.8	Epoch 146: Total Reward = 22711.0	Epoch 179: Total Reward = 22296.68
Epoch 114: Total Reward = 22644.0	Epoch 147: Total Reward = 22638.2	Epoch 180: Total Reward = 22309.74
Epoch 115: Total Reward = 22643.6	Epoch 148: Total Reward = 22688.2	Epoch 181: Total Reward = 22316.03
Epoch 116: Total Reward = 22642.8	Epoch 149: Total Reward = 22731.0	Epoch 182: Total Reward = 22340.17
Epoch 117: Total Reward = 22583.4	Epoch 150: Total Reward = 22743.5	Epoch 183: Total Reward = 22353.45
Epoch 118: Total Reward = 22679.2	Epoch 151: Total Reward = 22736.1	Epoch 184: Total Reward = 22367.97
Epoch 119: Total Reward = 22649.4	Epoch 152: Total Reward = 22757.7	Epoch 185: Total Reward = 22408.02
Epoch 120: Total Reward = 22680.9	Epoch 153: Total Reward = 22601.4	Epoch 186: Total Reward = 22410.94
Epoch 121: Total Reward = 22632.5	Epoch 154: Total Reward = 22626.9	Epoch 187: Total Reward = 22415.93
Epoch 122: Total Reward = 22651.2	Epoch 155: Total Reward = 22674.0	Epoch 188: Total Reward = 22423.62
Epoch 123: Total Reward = 22646.0	Epoch 156: Total Reward = 22691.4	Epoch 189: Total Reward = 22443.75
Epoch 124: Total Reward = 22673.5	Epoch 157: Total Reward = 22701.0	Epoch 190: Total Reward = 22429.79
Epoch 125: Total Reward = 22657.2	Epoch 158: Total Reward = 22722.0	Epoch 191: Total Reward = 22467.12
Epoch 126: Total Reward = 22635.8	Epoch 159: Total Reward = 22698.2	Epoch 192: Total Reward = 22467.16
Epoch 127: Total Reward = 22686.1	Epoch 160: Total Reward = 22666.0	Epoch 193: Total Reward = 22484.48
Epoch 128: Total Reward = 22639.3	Epoch 161: Total Reward = 22706.2	Epoch 194: Total Reward = 22510.11
Epoch 129: Total Reward = 22652.5	Epoch 162: Total Reward = 22688.7	Epoch 195: Total Reward = 22499.95
Epoch 130: Total Reward = 22690.8	Epoch 163: Total Reward = 22668.8	Epoch 196: Total Reward = 22479.34
Epoch 131: Total Reward = 22682.6	Epoch 164: Total Reward = 22388.9	Epoch 197: Total Reward = 22498.02
Epoch 132: Total Reward = 22683.4	Epoch 165: Total Reward = 22430.2	Epoch 198: Total Reward = 22507.26
Epoch 133: Total Reward = 22720.3	Epoch 166: Total Reward = 22459.2	Epoch 199: Total Reward = 22534.26
		Epoch 200: Total Reward = 22509.32



Hình 50: Đồ thị Reward qua các Episode của hệ thống gợi ý sách

- **Giai đoạn khởi đầu (episode 0–10):** Reward thấp và dường như không cải thiện, loanh quanh ở mức 22000.

- **Giai đoạn 2 (episode 10-15):** Reward tăng mạnh từ mức 22000 đến 22200, chứng tỏ mô hình thay đổi chính sách để có thể học tốt hơn.
- **Giai đoạn 3 (episode 15–100):** Reward tiếp tục tăng và đạt ngưỡng ~22600, cho thấy mô hình đang dần ổn định với chính sách hiệu quả hơn.
- **Giai đoạn 4 (episode 100-125):** Reward tăng nhưng khá chậm thậm chí bị bão hòa quanh mức 22650.
- **Giai đoạn cuối (episode 125-200):** Reward biến động mạnh đồng thời giảm, cho thấy mô hình không còn khả năng cải thiện nữa.

Xu hướng tăng rõ rệt sau đó tăng ít hoặc rất ít rồi giảm: Tổng reward tăng dần theo số episode, đặc biệt từ episode ~10 đến 15 có sự tăng mạnh, sau đó vẫn tiếp tục tăng nhưng chậm hơn và có dao động nhỏ, cho thấy mô hình học có hiệu quả sau đó đạt tới ngưỡng cực đại rồi bão hòa như tính chất thường thấy ở Học tăng cường.

PHẦN 3: KẾT LUẬN

3.1. KẾT QUẢ ĐẠT ĐƯỢC

Qua quá trình nghiên cứu và phát triển hệ thống thủ thư thông minh, nhóm sinh viên chúng em đã thu được nhiều kết quả đáng khích lệ cả về mặt kiến thức, kỹ năng và phương pháp làm việc nhóm.

Về kiến thức chuyên môn, chúng em đã vận dụng hiệu quả các nền tảng lý thuyết đã học như xử lý ảnh, các mô hình học sâu như (MTCNN, InceptionResNetV1), phương pháp Deep Q-Learning trong học tăng cường và đặc biệt là phương pháp Retrieval-Augmented Generation nhằm tích hợp Trí tuệ nhân tạo vào hệ thống thư viện. Những kiến thức này không chỉ giúp giải quyết các bài toán kỹ thuật cụ thể của đề tài mà còn mở rộng tư duy tiếp cận vấn đề theo hướng ứng dụng công nghệ vào thực tiễn.

Về kỹ năng, đề tài đã giúp chúng em rèn luyện khả năng lập trình, thiết kế hệ thống, xử lý dữ liệu và triển khai web. Ngoài ra, nhóm còn phát triển được kỹ năng

viết tài liệu kỹ thuật, trình bày logic, và đặc biệt là tư duy giải quyết vấn đề một cách hệ thống. Việc làm việc nhóm, trao đổi ý tưởng, phân công công việc hợp lý cũng là một kỹ năng quan trọng mà chúng em đã trau dồi được xuyên suốt quá trình thực hiện.

Về đề tài, hệ thống được xây dựng đã chứng minh được tính khả thi và mang lại giá trị thực tiễn cao trong môi trường đại học. Hệ thống không chỉ hỗ trợ tra cứu thông tin sách thông qua quét mã ISBN, nhận diện tiêu đề sách, gợi ý sách, mà còn cho phép người dùng đặt câu hỏi tự nhiên và nhận được câu trả lời chính xác nhờ tích hợp mô hình ngôn ngữ lớn. Bên cạnh đó, chức năng gợi ý sách thông minh và quản lý thư viện cũng góp phần hoàn thiện trải nghiệm người dùng một cách toàn diện.

Về phối hợp nhóm, nhóm chúng em đã làm việc ăn ý, hỗ trợ nhau vượt qua khó khăn kỹ thuật và thảo luận tích cực để đưa ra hướng giải quyết hiệu quả. Sự đoàn kết và chủ động là yếu tố then chốt giúp nhóm hoàn thành đề tài đúng tiến độ và đạt chất lượng tốt.

Tổng kết lại, đề tài không chỉ là cơ hội để vận dụng kiến thức đã học vào thực tiễn mà còn là hành trình giúp chúng em trưởng thành hơn về mặt tư duy, kỹ năng và trách nhiệm. Đây sẽ là tiền đề quan trọng để chúng em tự tin bước vào môi trường làm việc chuyên nghiệp trong tương lai.

3.2. ƯU ĐIỂM

- Tính ứng dụng cao: Hệ thống giải quyết được nhu cầu thực tế trong môi trường học thuật – hỗ trợ sinh viên tra cứu thông tin sách một cách nhanh chóng, chính xác và thân thiện với người dùng.
- Tích hợp công nghệ hiện đại: Đề tài kết hợp tất cả các khía cạnh lớn của Trí tuệ nhân tạo như: xử lý ảnh kết hợp nhận diện khuôn mặt, học tăng cường trong gợi ý sách và xử lý ngôn ngữ tự nhiên trong việc tạo chatbot thông minh– giúp tăng độ tiện dụng và tự động hóa cho hệ thống.
- Khả năng mở rộng: Hệ thống được thiết kế linh hoạt, có thể tích hợp thêm các chức năng như đánh giá sách, đăng ký mượn sách, hoặc mở rộng sang các lĩnh vực ngoài thư viện.
- Làm việc nhóm hiệu quả: Các thành viên trong nhóm phối hợp tốt, phân công rõ ràng, cùng nhau vượt qua khó khăn kỹ thuật để hoàn thành sản phẩm đúng tiến độ.

3.3. HẠN CHẾ

- Dữ liệu chưa phong phú: Hệ thống vẫn phụ thuộc nhiều vào phần giới thiệu sách, chưa tích hợp toàn bộ nội dung hoặc nhiều nguồn tri thức đa dạng hơn để phục vụ việc trả lời câu hỏi sâu.
- Tốc độ xử lý còn chậm: Khi truy vấn các câu hỏi ngôn ngữ tự nhiên, thời gian phản hồi có thể chậm nếu sử dụng mô hình lớn mà không có tối ưu hóa hoặc caching hiệu quả.
- Chưa triển khai thực tế: Hệ thống chưa được triển khai thử nghiệm rộng rãi trong môi trường thư viện thực tế, do đó chưa thể đánh giá toàn diện trải nghiệm người dùng và hiệu quả vận hành.
- Phụ thuộc vào API bên ngoài: Việc truy xuất thông tin sách từ Google Books API hoặc các nguồn dữ liệu ngoài có thể bị giới hạn hoặc phụ thuộc vào kết nối internet.
- Giao diện còn đơn giản: Tuy đáp ứng chức năng chính, giao diện hệ thống chưa thật sự tối ưu về mặt thẩm mỹ và trải nghiệm người dùng so với các hệ thống thương mại.

3.4. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

Để nâng cao hơn nữa tính hiệu quả và khả năng ứng dụng của hệ thống thư thư thông minh, nhóm đề xuất một số hướng phát triển trong tương lai như sau:

- Mở rộng nguồn dữ liệu: Tích hợp thêm nhiều nguồn tài nguyên học thuật như thư viện số, tài liệu chuyên ngành, luận văn, sách giáo trình... nhằm tăng độ bao phủ và chiều sâu cho các câu trả lời từ hệ thống.
- Tối ưu hóa tốc độ và độ chính xác: Nâng cấp mô hình xử lý ngôn ngữ, áp dụng các kỹ thuật rút gọn mô hình hoặc triển khai cơ chế cache và embedding vector để cải thiện thời gian phản hồi và giảm tải cho hệ thống.
- Nâng cấp trải nghiệm người dùng: Cải tiến giao diện web hiện đại hơn, tối ưu hiển thị trên thiết bị di động, bổ sung các chức năng như đánh giá sách, tạo danh sách yêu thích, hoặc hỗ trợ giọng nói để tăng tính tương tác.
- Triển khai và kiểm thử thực tế: Thực hiện thử nghiệm hệ thống trong môi trường thư viện thật của trường để thu thập phản hồi người dùng, từ đó điều chỉnh chức năng phù hợp hơn với nhu cầu thực tiễn.
- Kết nối hệ thống quản lý thư viện hiện có: Tích hợp hệ thống với phần mềm quản lý thư viện hiện hành (ví dụ Koha, Aleph...) để đảm bảo tính liên thông dữ liệu và tăng khả năng ứng dụng trong vận hành thực tế.

TÀI LIỆU THAM KHẢO

- [1] VTV.VN, “Ngày Sách và Văn hóa đọc Việt Nam 2024: Những tín hiệu tích cực từ cộng đồng,” 16 tháng 4 năm 2024. [Trực tuyến]. Có tại: <https://vtv.vn/doi-song/ngay-sach-va-van-hoa-doc-viet-nam-2024-nhung-tin-hieu-tich-cuc-tu-cong-dong-20240416073329563.htm>.
- [2] Châu Anh, “Nhiều thư viện tại Việt Nam như xác sống,” VnExpress, ngày 3 tháng 6 năm 2025. [Trực tuyến]. Có tại: <https://vnexpress.net/nhieu-thu-vien-tai-viet-nam-nhu-xac-song-4893373.html>.
- [3] Phan Trường Nhất, “Tác động của sự thay đổi, ứng dụng công nghệ thông tin và công nghệ mới trong dịch vụ thông tin – thư viện tại các trường đại học ở Việt Nam hiện nay,” Tạp chí Khoa học - Đại học Đồng Nai, số 27, 2023.
- [4] Y. Liu, “AI-powered Library Assistant Xiaotu at Tsinghua University”, in Proceedings of the International Symposium on Library Automation, Beijing, China, Aug. 2021, pp. 55–60.
- [5] Thư viện Quốc gia Việt Nam, “Ứng dụng A.I trong hoạt động phân loại tài liệu của thư viện” 2023. [Trực tuyến]. Có tại: <https://nlv.gov.vn/nghien-cuu-trao-doi/ung-dung-a.i-trong-hoat-dong-phan-loai-tai-lieu-cua-thu-vien.html>.
- [6] VinUni, “Nơi giáo dục và công nghệ gặp gỡ: Khám phá sách giáo khoa điện tử đầu tiên tích hợp AI của VinUni”, [Trực tuyến]. Có tại: <https://admissions.vinuni.edu.vn/vi/noi-giao-duc-va-cong-nghe-gap-go-kham-pha-sach-giao-khoa-dien-tu-dau-tien-tich-hop-ai-cua-vinuni/>.
- [7] J. D. Shank, “University of Arizona Library Integrates ChatGPT to Support Research,” in *Proceedings of the 2023 ACRL Conference*, Pittsburgh, PA, Mar. 15–18, 2023, pp. 210–215.
- [8] Y. Du, W. Qian, Y. Luo, et al., “PP-OCR: A Practical Ultra Lightweight OCR System”, *arXiv preprint arXiv:2009.09941*, 2020. [Trực tuyến]. Có tại: <https://arxiv.org/abs/2009.09941>.
- [9] K. Zhang, Z. Zhang, Z. Li, và Y. Qiao, “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3670–3679.
- [10] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”, in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec. 2020.
- [11] B. K. Iwana và S. Uchida, “BookCoverNet: A Dataset and Model for Book Cover Classification,” in *Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, Australia, Sep. 2019, pp. 144–149.
- [12] Z. Zhang và J. Liu, “Intelligent Library System Using RFID and AI-Based Search”, *IEEE Access*, vol. 9, pp. 144321–144329, 2021.
- [13] F. Schroff, D. Kalenichenko và J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.
- [14] C. Szegedy, S. Ioffe và V. Vanhoucke, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning” *arXiv preprint arXiv:1602.07261*, 2016. [Trực tuyến]. Có tại: <https://arxiv.org/abs/1602.07261>.
- [15] N. Reimers và I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, Nov. 2019, pp. 3982–3992.
- [16] C. J. C. H. Watkins, “Learning from Delayed Rewards” Luận án Tiến sĩ, Đại học Cambridge, Vương quốc Anh, 1989.

- [17] M. J. Kim và H. Peng, “Learning Time Reduction Using Warm Start Methods for a Reinforcement Learning Based Supervisory Control in Hybrid Electric Vehicle Applications” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 198–205, Jan. 2018.
- [18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg và D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015.
- [19] G. Gerganov, “llama.cpp: Port of LLaMA model in C/C++,” GitHub, 2023. [Trực tuyến]. Có tại: <https://github.com/ggerganov/llama.cpp>.
- [20] Phan Văn Tán, “Tìm hiểu về Django – Framework hỗ trợ Python trong lập trình Web” Viblo, 25/8/2022. [Trực tuyến]. Có tại: <https://viblo.asia/p/tim-hieu-ve-django-framework-ho-tro-python-trong-lap-trinh-web-QpmlexbkZrd>.
- [21] Nguyễn Hữu Dũng, “Flask python là gì, tính năng cơ bản và lý do vì sao nên sử dụng” Bizfly.vn, 1/4/2021. [Trực tuyến]. Có tại: <https://bizfly.vn/techblog/flask-python-la-gi.html>.
- [22] 200Lab Blog, “Tìm hiểu LangChain: Framework phát triển ứng dụng LLM mạnh mẽ,” 10/2/2025. [Trực tuyến]. Có tại: <https://200lab.io/blog/langchain-la-gi>.
- [23] Pum, “SQL Server là gì? Cách tải & cài đặt Microsoft SQL Server” 200Lab Blog, 16/9/2023. [Trực tuyến]. Có tại: <https://200lab.io/blog/sql-server-la-gi>.
- [24] A. A. Awan, “Learn How to Use Chroma DB: A Step-by-Step Guide” DataCamp, 28/9/2023. [Trực tuyến]. Có tại: <https://www.datacamp.com/tutorial/chromadb-tutorial-step-by-step-guide>.
- [25] Document Processing, “Thư viện Python nguồn mở để quản lý siêu dữ liệu PDF” 2025. [Trực tuyến]. Có tại: <https://products.documentprocessing.com/vi/metadata/python/pymupdf/>
- [26] P. P., “What Is Optical Character Recognition (OCR)? Explained,” Roboflow Blog, ngày 21 tháng 11 năm 2023. [Trực tuyến]. Có tại: <https://blog.roboflow.com/what-is-optical-character-recognition-ocr/>.
- [27] P. B. C. Quốc, “VietOCR: Transformer OCR,” GitHub, 2020. [Trực tuyến]. Có tại: <https://github.com/pbcquoc/vietocr>.
- [28] Hugging Face. *sentence-transformers/paraphrase-multilingual-mpnet-base-v2*. [Trực tuyến]. Có tại: <https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2>.
- [29] H. Burak, “Pins Face Recognition: Facial Recognition Dataset collected from Pinterest” Kaggle, 2018. [Trực tuyến]. Có tại: <https://www.kaggle.com/datasets/herveisburak/pins-face-recognition>.
- [30] Thư viện số Trường Đại học Sư phạm Kỹ thuật TP.HCM, “Trang chủ Thư viện số,” Trường Đại học Sư phạm Kỹ thuật TP.HCM, [Trực tuyến]. Có tại: <https://thuvienso.hcmute.edu.vn/>.