

Đinh Nhật Thành

0363956098 | dinhnhatthanh248@gmail.com | linkedin.com/in/d-nhatthanh248 | github.com/MathematicGuy

HỌC VẤN

CMC University

Bằng Cử Nhân Công Nghệ Thông Tin / GPA: 3.19

Hà Đông, Hà Nội

09/2022 – 10/2025

Dạt giải khuyến khích cuộc thi nghiên cứu khoa học cấp trường

Các môn học tiêu biểu: Học sâu và ứng dụng, Thị giác máy tính, Xử lý ngôn ngữ tự nhiên, Phân tích dữ liệu lớn

KINH NGHIỆM

Thực tập sinh

CMC ATI

09/2024 – 12/2024

Hà Đông, Hà Nội

- Mục đích: Trích xuất các thông tin mặt trước của thẻ CCCD và lưu vào hệ thống Database của Postgres
- Phát triển hệ thống trích xuất thông tin thẻ CCCD sử dụng các mô hình VietOCR và YOLOv11 cho việc nhận diện góc và vùng thông tin (ROI) của thẻ.
- Tái dựng góc còn thiếu sử dụng Homography và áp dụng Perspective Transformation sử dụng 4 góc thẻ CCCD để cắt ảnh thẻ hiệu quả
- Tăng cường ảnh bằng Roboflow, thử nghiệm và so sánh các phương pháp tiền xử lý ảnh chính (như AHE, sCLAHE và Adaptive Thresholding) kết hợp với Morphological Operations cùng các kỹ thuật loại bỏ bóng, làm mờ, điều chỉnh độ sáng/tương phản tự động để xây dựng các module tiền xử lý ảnh giúp Yolov11 nâng cao độ chính xác nhận diện 4 góc thẻ CCCD lên **83%** và độ chính xác trung bình khi trích xuất vùng thông tin (ROI) lên đến **93%**.
- Triển khai các phương pháp hậu xử lý như Non-Maximum Suppression (NMS) để tinh chỉnh kết quả nhận diện vùng thông tin, đảm bảo dữ liệu đầu ra sạch và chính xác.
- Xây dựng API backend với FastAPI và PostgreSQL để quản lý, lưu trữ, và chuẩn hóa dữ liệu trích xuất bằng Regex, đảm bảo luồng thông tin liền mạch.

Nghiên Cứu Sinh

CMC University

03/2025 – 7/2025

Hà Đông, Hà Nội

- Học cách đọc báo nghiên cứu và viết báo cáo latex, docx, slides chuẩn format nghiên cứu khoa học cấp trường
- Phát triển Module nhận diện hành vi bạo lực trong môi trường học đường dưới sự hướng dẫn của ThS. Nguyễn Khánh Sơn.
- Thiết kế và tối ưu hệ thống nhận diện bạo lực sử dụng YoloPose11m và LSTM dựa trên bài báo "Student Behavior Recognition System for the Classroom Environment Based on Skeleton Pose Estimation and Person Detection"
- Xây dựng bộ dữ liệu tùy chỉnh sử dụng Matplotlib để hình dung kết quả và xác định vấn đề trong quy trình thử nghiệm trích xuất keypoints của Yolopose11m cho 74 videos, gồm góc quay (ngang cho Kicking/Punching, dọc cho Standing) và quay toàn thân/nửa thân trên cho Punching nhằm chuẩn hóa dữ liệu giúp mô hình nắm bắt được đặc trưng của mỗi hành động tốt hơn.
- Features Engineer qua trích xuất 17 keypoints theo chuẩn COCO từ video và tinh chỉnh các đặc trưng (tọa độ khớp nối chuẩn hóa, khoảng cách giữa các khớp, và góc giữa các khớp) giúp tăng Mean Accuracy từ **43%** lên **78%**. sử dụng Numpy để tối ưu hiệu suất.
- Phát triển & tối ưu mô hình học sâu: Xây dựng và tối ưu các mô hình (LSTM, GRU, DNN) trên TensorFlow áp dụng các kỹ thuật Dropout, L2 Regularization để giảm thiểu overfitting, tăng cường khả năng tổng quát hóa của mô hình và Adam Optimizer để tinh chỉnh tham số.
- Tiền xử lý dữ liệu video bằng OpenCV và chia tập huấn luyện/kiểm tra với K-Fold Cross-Validation sử dụng Scikit-learn, đảm bảo đánh giá hiệu suất mô hình đáng tin cậy và tận dụng tối đa dữ liệu hạn chế; tận dụng CUDA để tăng tốc các mô hình học sâu.

DỰ ÁN

Naive RAG for Question Answering & Multiple Choice Question Generation

06/2024 – 09/2024

- Mục Dích: Tóm tắt hoặc Tạo sinh sinh nhiều câu trả lời trắc nghiệm dựa trên nội dung PDF
- Xử lý và phân tích files sử dụng PDFReader, Pandas để xác định tần suất token, đảm bảo context window của mô hình LLM
- So sánh phương pháp Chunking Recursive Chunking và Dynamic Chunking để cải thiện quá trình Chunking

- Áp dụng Flash Attention, CUDA và nf4 quantization để tăng tốc mô hình LLM cho tạo sinh và Embedding cho quá trình so sánh các mô hình embedding trên các tập dữ STS-B, QQP và MRPC
- Nghiên cứu và so sánh hiệu quả giữa 2 phương pháp Chunking (Recursive Chunking và Dynamic Chunking) để tìm ra thuật toán Chunking tối ưu cho việc trích xuất ngữ cảnh.
- Tinh chỉnh đầu ra LLM sử dụng Prompt Engineering để đảm bảo câu hỏi trắc nghiệm được tạo sinh có cấu trúc chuẩn và dễ dàng trích xuất bằng Regex.

Agentic RAG for Multiple Choice Question Validation with Automatic Validation) 05/2024 – 09/2024

- Mục đích: Xây dựng quy trình RAG cho API tạo sinh câu hỏi trắc nghiệm cho chủ đề học thuật từ file pdf tiếng việt có cơ chế kiểm tra chất lượng và ước tính độ khó câu hỏi tự động
- Tiền xử lý PDF sang markdown và phân tích cú pháp PDF với LLM, thiết kế prompt để trích xuất khái niệm, công thức, ví dụ và tóm tắt tự động
- Thiết lập chuẩn đánh giá chất lượng câu hỏi và độ khó với rule-based dựa trên ngữ cảnh truy hỏi và độ tương đồng giữa phương án đúng và sai
- Trực quan hóa các tiêu chuẩn độ khó, giúp LLM đánh giá và phân loại câu hỏi trắc nghiệm đúng độ khó mong muốn
- Chất lượng truy hỏi đúng dữ liệu của môn học từ cơ sở dữ liệu vector đạt 89% sử dụng mô hình Embedding đa ngôn ngữ
- Xây dựng và Triển khai API cho RAG trên huggingface với Docker đảm bảo tốc độ tạo sinh 15 câu/phút, chi phí cho 50 câu dưới 1000 đồng

CHỨNG CHỈ

Samsung Innovation Campus AI Program <i>Samsung Innovation Campus</i>	06/2023 – 09/2024 <i>Samsung</i>
Linear Algebra for Machine Learning and Data Science <i>DeepLearning.ai</i>	07/2024 – 09/2024 <i>Coursera</i>
Calculus for Machine Learning and Data Science <i>DeepLearning.ai</i>	09/2024 – 10/2024 <i>Coursera</i>

KỸ NĂNG CHUYÊN MÔN

Languages: Java, C#, Python, SQL (Postgres), MongoDB (NoSQL), JavaScript, HTML/CSS

AI/ML Frameworks & Libraries: PyTorch, TensorFlow, , LangChain, OpenCV, MediaPipe, Streamlit,

Developer Tools: Git, Docker, FastAPI, PostmanAPI

Big Data & Cloud: Hadoop, Apache Spark