

# Đinh Nhật Thành

0363956098 | [dingnhathanh248@gmail.com](mailto:dingnhathanh248@gmail.com) | [linkedin.com/in/d-nhatthanh248](https://www.linkedin.com/in/d-nhatthanh248) | [github.com/MathematicGuy](https://github.com/MathematicGuy)

## HỌC VẤN

### CMC University

Bằng Cử Nhân Công Nghệ Thông Tin / GPA: 3.09

Đạt 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

Hà Đông, Hà Nội

09/2022 – 10/2025

## 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 nổi 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 Đích: Tóm tắt hoặc Tạo sinh sinh nhiều câu trả lời trắc nhiệm 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 liệu 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.

Phân loại văn bản sử dụng Transformer (Self-Attention & Embeddings)
02/2024 – 02/2024

- Mục đích: triển khai mô hình phân loại văn bản từ báo bằng cơ chế tự chú ý (self-attention) và nhúng từ (word embeddings) sử dụng PyTorch, để hiểu hơn về kiến trúc và cách hoạt động của Transformer.
- Xây dựng lớp ‘Head’ tùy chỉnh để thực hiện cơ chế tự chú ý, bao gồm các phép biến đổi tuyến tính riêng biệt cho Key, Query, Value từ vector đầu vào, sau đó tính toán trọng số chú ý thông qua tích vô hướng và hàm softmax, cũng như scaling factor ‘sqrt(d\_k)’ để duy trì ổn định gradient.
- Thiết lập pipeline xử lý văn bản với ‘torchtext’ sử dụng ‘basic\_english’ tokenizer để xây dựng từ vựng với token
- Sử dụng lớp ‘nn.Embedding’ của PyTorch để ánh xạ các chỉ số token rồi rạc thành các vector nhúng, giúp mô hình nắm bắt được các mối quan hệ ngữ nghĩa tiềm ẩn giữa các từ trong không gian liên tục.
- Trực quan hóa các vector nhúng trong không gian 3D bằng ‘matplotlib’, giúp phân tích và đánh giá trực quan về cách các từ được biểu diễn và mối quan hệ ngữ nghĩa của chúng sau quá trình nhúng.

Naive Bayes cho Dự đoán Chơi Tennis
07/2024 – 07/2024

- Mục đích: Dự đoán khả năng chơi tennis dựa trên
- Xây dựng một mô hình Naive Bayes để dự đoán khả năng chơi tennis bằng cách phân tích các điều kiện thời tiết như Outlook, Temperature, Humidity và Wind từ dữ liệu dạng bảng, nhằm áp dụng Lý thuyết Xác suất để đưa ra quyết định dự đoán về việc chơi tennis.
- Thực hiện tiên xử lý dữ liệu để tính toán xác suất tiên nghiệm (prior probabilities) cho mỗi lớp (‘Yes’/‘No’) và xác suất có điều kiện (conditional probabilities) của từng thuộc tính (outlook, temperature, humidity, wind) cho mỗi lớp (yes/no.
- Triển khai thuật toán Naive Bayes từ đầu để kết hợp các xác suất này, đưa ra dự đoán cho các điều kiện thời tiết mới.
- Sử dụng thư viện NumPy để xử lý dữ liệu kiểu array trong toàn bộ quá trình tính toán và huấn luyện mô hình.

CHỨNG CHỈ

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

KỸ NĂNG CHUYÊN MÔN

<b>Languages:</b> Java, C#, Python, SQL (Postgres), MongoDB (NoSQL), JavaScript, HTML/CSS <b>AI/ML Frameworks &amp; Libraries:</b> PyTorch, TensorFlow, Scikit-learn, LangChain, OpenCV, MediaPipe, Streamlit, Spacy <b>Developer Tools:</b> Git, Docker, Flask, FastAPI, PostmanAPI <b>Big Data &amp; Cloud (Optional):</b> Hadoop, Google Cloud Platform	
---	--