Nguyễn Đình Khải

+84 37838454 nguyendinhkhaiqt2005@gmail.com ngwkhai.com github.com/ngwkhai Hà Nội, Việt Nam

TÓM TẮT

Sống và làm việc tại Việt Nam, tôi là một nghiên cứu viên và kỹ sư AI với niềm đam mê mạnh mẽ dành cho trí tuệ nhân tạo và các công nghệ dựa trên dữ liệu. Nền tảng học thuật về khoa học máy tính cùng kinh nghiệm thực chiến qua các dự án học máy, học sâu, xử lý ngôn ngữ tự nhiên và công nghệ tài chính đã cho tôi cả chiều sâu lý thuyết lẫn kỹ năng triển khai. Tôi đã làm việc với những bài toán trải rộng từ nhận dạng ảnh, phát hiện gian lận cho tới xây dựng chatbot thông minh và hệ thống thời gian thực. Hành trình đó mang lại cho tôi kinh nghiệm quý giá về phân tích dữ liệu, lập trình và giải quyết vấn đề. Tôi luôn được thôi thúc bởi mong muốn tạo ra những giải pháp AI không chỉ tiến bộ về mặt công nghệ mà còn tạo tác động thật trong đời sống. Đổi mới, học hỏi không ngừng và mở rộng giới hạn của những gì AI có thể làm là cốt lõi trong công việc của tôi.

HỌC VẤN

Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội 08/2023 - Hiện tại
Cử nhân Trí tuệ Nhân tạo GPA: 3.61/4

KINH NGHIỆM

VinSmart Future

Chuyên viên Ứng dụng AI 11/2026 - Hiện tại
- Đảm nhận mảng dữ liệu AI chuyên sâu và chịu trách nhiệm kiểm soát chất lượng dữ liệu, bảo đảm mỗi bộ dữ liệu đều đạt chuẩn trước khi được đưa vào khai thác.
- Hỗ trợ quản lý đội ngũ 12 thành viên, góp phần giữ vững tiến độ và chất lượng công việc chung của nhóm.

Kỹ sư tập sự Ứng dụng AI 09/2026 - 11/2026
- Thực hiện kiểm soát chất lượng dữ liệu: rà soát, đối chiếu với tiêu chuẩn đề ra và bảo đảm dữ liệu đạt yêu cầu trước khi đưa vào sử dụng.
- Hỗ trợ quản lý đội ngũ 6 thành viên trong công việc hằng ngày.

Thực tập sinh AI 05/2026 - 08/2026
- Xây dựng pipeline thu thập và kiểm soát chất lượng dữ liệu du lịch: từ khâu thu thập, nạp dữ liệu cho tới bước rà soát, xác minh để quyết định dữ liệu nào đủ điều kiện giữ lại.

Viettel Networks 09/2025 - 01/2026

Thực tập sinh AI
- Nghiên cứu sinh cấu trúc dự án từ đặc tả bằng ngôn ngữ tự nhiên, xử lý hiện tượng "mù cấu trúc" của các mô hình ngôn ngữ lớn khi phải sinh ra kết quả phân cấp.
- Xây dựng tập dữ liệu huấn luyện và bộ chỉ số đánh giá cho bài toán, tinh chỉnh các mô hình mở nhỏ gọn bằng LoRA cùng một hàm mất mát thiết kế đồng bộ với bộ chỉ số đó.

NGHIÊN CỨU

NLP Lab 06/2025 - Hiện tại

Thành viên
- Tham gia các hoạt động nghiên cứu và phát triển trong lĩnh vực xử lý ngôn ngữ tự nhiên, bao gồm khảo sát các mô hình AI, thực nghiệm trên dữ liệu văn bản và đóng góp ý tưởng cho các dự án nhóm.

DỰ ÁN

ResearchMap AI 2026

Một mình — sản phẩm, backend, pipeline AI, frontend, triển khai
Biến một câu mô tả lĩnh vực nghiên cứu thành bản đồ tài liệu gồm chủ đề, xu hướng và khoảng trống. Một agent trả lời câu hỏi về lĩnh vực và chỉ trích dẫn những bài báo nó thực sự truy xuất được.
Bản chạy thật: researchmap.ngwkhai.com · Chi tiết: ngwkhai.com/vi/projects/researchmap
- Tự xây dựng toàn bộ sản phẩm: backend FastAPI trên PostgreSQL và pgvector, frontend React, đăng nhập Google OAuth với xác thực hai lớp TOTP, các gói dịch vụ có hạn mức, và trang quản trị.
- Thiết kế pipeline phân tích: LLM lên kế hoạch truy vấn OpenAlex, bài báo được nhúng rồi phân cụm bằng KMeans, LLM đặt tên từng chủ đề, còn khoảng trống nghiên cứu được tính từ những cặp chủ đề không có bài chung.
- Xây dựng chat agentic RAG bằng tool calling của OpenAI, gồm 7 công cụ và tối đa 4 vòng. Trích dẫn chỉ lấy từ những bài mà công cụ trả về, và có ngưỡng liên quan được đo trên dữ liệu thật.
- Đánh giá hệ thống bằng bốn bộ benchmark có LLM làm giám khảo, và triển khai lên Cloud Run bằng GitHub Actions với Workload Identity Federation.

Kết quả chính: Bài trích dẫn được xác minh có thật **36/36** · Precision@5 của truy xuất **0.867** · Chi phí mỗi câu trả lời **$0.00032** (gpt-5.4-nano, đo thực tế)
Công nghệ: Python, FastAPI, PostgreSQL, pgvector, scikit-learn, OpenAI API, React, TypeScript, Cloud Run, GitHub Actions.

TaxMate AI 2026

Trưởng nhóm bốn người — phạm vi, tích hợp và phát hành; core Risk, API và xác thực, truy xuất pháp luật, triển khai GCP
Trợ lý tuân thủ thuế cho người bán hàng online tại Việt Nam: trả lời câu hỏi pháp luật kèm trích dẫn cho mọi nhận định, và đối chiếu doanh thu với số đã kê khai. Khải dẫn dắt nhóm bốn người xây dựng.
Bản chạy thật: taxmate.ngwkhai.com · Chi tiết: ngwkhai.com/vi/projects/taxmate
- Dẫn dắt nhóm bốn người xây dựng bốn core trong một monorepo: Legal RAG, Risk, Graph và tối ưu lượng tử. Chốt phạm vi, ngân sách và lịch phát hành, review từng core, rồi tích hợp tất cả thành một sản phẩm.
- Xây dựng core Risk: nhập file CSV rồi đối chiếu doanh thu, tiền về tài khoản và doanh thu kê khai qua bốn rule có phiên bản. Khi thiếu dữ liệu, hệ thống trả `insufficient_data` thay vì đưa ra một điểm số gây hiểu lầm.
- Làm lại phần truy xuất của trợ lý pháp luật bằng BM25 trên cặp âm tiết tiếng Việt, và buộc mỗi nhận định chỉ được trích dẫn các chunk đã thực sự truy xuất. Nhờ đó, recall@5 theo văn bản tăng từ 32 lên 52 trên 56 câu kiểm thử.
- Xây dựng xác thực đa tenant trên Supabase, triển khai hai môi trường dev và production trên Cloud Run sau load balancer và CDN, thông qua GitHub Actions.

Kết quả chính: Recall@5 của Legal **93%** (52 trên 56 câu, trước đó là 32) · Văn bản pháp luật đã lập chỉ mục **13** (1.118 chunk) · Core đã tích hợp **4**
Công nghệ: Python, FastAPI, PostgreSQL, Supabase Auth, OpenAI API, React, TypeScript, Cloud Run, GitHub Actions.

Detoura 2026

Làm một mình — kiến trúc, engine luật, giao thức, service giải đấu, thuật toán tìm kiếm, triển khai

Game cờ đối kháng hai người thời gian thực, xây dựng như một nền tảng: lõi luật tất định, bộ schema vừa kiểm tra, vừa tài liệu hoá và kiểm thử API, cùng service giải đấu riêng cho bot.

Bản chạy thật: detoura.ngwkhai.com · Chi tiết: ngwkhai.com/vi/projects/detoura

- Thiết kế monorepo TypeScript gồm chín package xoay quanh một lõi luật tất định, không phụ thuộc thư viện nào. Nhờ vậy cùng một đoạn code chạy được trên trình duyệt, trên server và khi phát lại ván đấu.
- Tách nền tảng khỏi từng game cụ thể: luật game cắm vào qua một interface có kiểu duy nhất, còn server coi mỗi ván đấu là dữ liệu đóng, không đọc vào bên trong.
- Dùng chung một bộ schema Zod để kiểm tra mọi tin nhắn và sinh đặc tả OpenAPI lẫn WebSocket. Một bộ kiểm thử chống lệch hợp đồng chạy trực tiếp trên bản đang triển khai.
- Xây dựng server WebSocket giữ quyền quyết định cuối cùng, có giới hạn tần suất và đồng hồ do server quản lý, thêm service giải đấu trên Postgres, và pipeline triển khai lên Cloud Run không lưu khoá.

Kết quả chính: Workspace **9** · Interface có hợp đồng **5** · Tin nhắn giao thức **14**

Công nghệ: TypeScript, Zod, ws, React, Vite, PostgreSQL, Docker, Cloud Run, GitHub Actions.

Bacera 2026

Nhóm hai người — plugin Pancake POS, shop và checkout, thanh toán, tìm kiếm, bản demo và hạ tầng (theme, đăng nhập và workshop do người còn lại làm)

Website bán hàng và đặt chỗ workshop cho một xưởng gốm ở Hà Nội, xây dựng trên WordPress, dùng Pancake POS làm nguồn dữ liệu gốc. Bản demo công khai phát lại dữ liệu POS đã được ghi sẵn.

Bản chạy thật: bacera.ngwkhai.com · Mã nguồn: github.com/ngwkhai/bacera-pancake-integration · Chi tiết: ngwkhai.com/vi/projects/bacera

- Xây dựng plugin WordPress tích hợp Pancake POS qua sáu module. Sản phẩm, tồn kho và đơn hàng đều nằm trong POS, nên website không phải giữ một kho hàng thứ hai.
- Giữ API key ở phía server: checkout gọi các REST route có kiểm tra nonce để làm việc với POS, còn ảnh sản phẩm được proxy qua tên miền của chính xưởng.
- Xác minh thanh toán chuyển khoản, thẻ và PayPal bằng payment intent có hạn dùng, cùng một endpoint quy các tín hiệu "đã thanh toán" không thống nhất của POS về một kết quả duy nhất.
- Biến dự án khách hàng thành bản demo công khai an toàn: phát lại dữ liệu POS đã ghi, chặn container kết nối tới POS thật, và tự reset mỗi đêm trên Compute Engine.

Kết quả chính: Module tích hợp **6** · Bảng SQL tự thiết kế **15** · Endpoint POS được phát lại **12**

Công nghệ: PHP 8.3, WordPress, REST API, Tailwind CSS, Alpine.js, MariaDB, Docker, Caddy, Compute Engine.

Flow Center 2026

Một mình, cho khách hàng — full stack, tính năng AI, triển khai

Đặt lịch, khảo sát thể trạng và trợ lý chat cho một cơ sở chăm sóc đau bụng kinh theo Y học Cổ truyền. AI chỉ được hoạt động trong phạm vi nội dung đã được chuyên gia duyệt.

Bản chạy thật: flowcenter.ngwkhai.com · Chi tiết: ngwkhai.com/vi/projects/flowcenter

- Một mình xây dựng toàn bộ website của cơ sở trên Next.js 16, PostgreSQL và Drizzle, gồm đặt lịch kèm quy trình xử lý của nhân viên, khảo sát thể trạng, trợ lý chat, xác thực và trang quản trị, triển khai trên Vercel và Neon.
- Bảo đảm đặt lịch an toàn khi nhiều người đặt cùng lúc bằng advisory lock và các bước chuyển trạng thái ràng buộc ngay trong SQL. Có test bắn tám lượt đặt song song vào hai chỗ để kiểm chứng.
- Giữ AI trong phạm vi nội dung đã duyệt: khảo sát được chấm bằng rule cố định, gặp dấu hiệu cảnh báo thì dừng lại, và mô hình chỉ được diễn đạt lại mẫu lời khuyên của cơ sở, gặp lỗi thì quay về dùng mẫu.
- Xây dựng trợ lý chat RAG với tìm kiếm tiếng Việt kiểu lai: từ khoá bỏ dấu kết hợp pgvector, trộn kết quả bằng RRF. Trợ lý chỉ đọc dữ liệu, và chỉ đặt lịch khi chính người dùng bấm nút.

Kết quả chính: Lượt đặt song song trong test tranh chấp **8** (cho 2 chỗ, không vượt chỗ) · Dấu hiệu cảnh báo dừng phân tích **10** · Số ngày giữ hội thoại **90**

Công nghệ: Next.js 16, TypeScript, PostgreSQL, Drizzle ORM, pgvector, OpenAI API, Vercel, Neon.

AI Pipeline Mastery 2026

Làm một mình — kiến trúc nền tảng, điều phối GPU, bộ công cụ build, sản xuất nội dung

Một khoá học được xây dựng như phần mềm: pipeline Python tự sinh học liệu rồi chạy thử để kiểm tra, còn nền tảng Next.js chạy code của người học ngay trên trình duyệt hoặc trên một nhóm GPU từ xa.

Chi tiết: ngwkhai.com/vi/projects/ai-pipeline-mastery

- Xây dựng bộ công cụ Python sinh slide, tài liệu và bài tập từ mã nguồn. Đến nay bộ công cụ đã tạo ra 383 slide và 129 hình vẽ từ dữ liệu thật.
- Kiểm soát chất lượng bằng cách chạy thử: mọi lời giải mẫu phải pass, mọi file khởi đầu phải fail đúng như dự kiến, và các file sinh ra được build lại rồi so sánh từng byte trước mỗi lần phát hành.
- Xây dựng nền tảng học trên Next.js 16 với server action và không có tầng API nội bộ. Trình duyệt không bao giờ truy cập trực tiếp kho dữ liệu, và file được phát qua URL có chữ ký.
- Thiết kế hàng đợi job GPU cho môi trường serverless: mượn tài khoản từ một nhóm, theo dõi job bằng polling kết hợp quét định kỳ, và có chế độ mô phỏng để chạy toàn bộ luồng ngay trên máy local.

Kết quả chính: Slide đã sinh **383** · Hình đã render **129** · Đường chạy code **2** (trong trình duyệt và GPU từ xa)

Công nghệ: Next.js 16, Firebase Auth, Firestore, Cloudflare R2, Pyodide, Kaggle API, Tailwind v4.

Phát hiện rác phòng học 2026

Toàn hệ thống — kiến trúc phân tán, client, service suy luận, tối ưu runtime, triển khai

Hệ thống thị giác máy tính tách làm hai phần qua mạng: client giữ trạng thái lo phần camera và dashboard, còn một service FastAPI không trạng thái chạy YOLOv8 trên GPU.

Mã nguồn: github.com/ngwkhai/AI_Server · Chi tiết: ngwkhai.com/vi/projects/ai-server

- Tách một ứng dụng chạy trong một tiến trình thành hai phần: client giữ trạng thái và service suy luận không trạng thái chạy trên GPU, nối với nhau bằng một hợp đồng HTTP duy nhất có xác thực.
- Giảm tải cho mạng và GPU bằng cách thu nhỏ frame về 640 px rồi quy đổi lại toạ độ trả về, đồng thời chỉ suy luận trên các frame lấy mẫu và dùng lại kết quả đã cache cho các frame ở giữa.
- Thêm cơ chế tự chạy model tại chỗ khi service từ xa không phản hồi, và một API trạng thái cho biết mỗi request do backend nào xử lý và mất bao lâu.
- Xuất model sang ONNX nửa độ chính xác, nạp model ngay khi khởi động, và triển khai cùng một ứng dụng FastAPI lên GPU notebook, Docker image và GPU serverless.

Kết quả chính: Định dạng model **3** (PyTorch, ONNX, TensorRT) · Bề ngang gửi lên **640 px** · Nơi triển khai **3**

Công nghệ: Python, YOLOv8, ONNX Runtime, FastAPI, Flask, OpenCV, SQLite, Docker, Modal.

Neural Machine Translation 2025

Xây dựng dữ liệu, cài đặt mô hình, thiết kế đánh giá

Xây dựng bộ dữ liệu 12,5 triệu cặp câu cho bài toán khôi phục dấu tiếng Việt và cài đặt từ đầu các mô hình seq2seq RNN, GRU, LSTM và Transformer bằng PyTorch.

Mã nguồn: github.com/ngwkhai/Neural-Machine-Translation · Chi tiết: ngwkhai.com/vi/projects/neural-machine-translation

- Xây dựng bộ dữ liệu khôi phục dấu tiếng Việt quy mô lớn đầu tiên (12,5 triệu cặp câu không dấu → có dấu).
- Cài đặt các mô hình Seq2Seq (RNN, GRU, LSTM, Transformer) từ đầu bằng PyTorch.
- Thiết kế và thực hiện đánh giá mô hình toàn diện với các chỉ số BLEU và ChrF++, đạt BLEU 81.31 và ChrF++ 88.57 với Transformer, vượt trội đáng kể so với các mốc cơ sở RNN và GRU.
- Đóng góp xuyên suốt từ xây dựng dữ liệu, phát triển mô hình tới đánh giá hiệu năng, tạo ra một hệ thống nguyên mẫu vững vàng cho khôi phục văn bản tiếng Việt.

Kết quả chính: BLEU **81.31** (Transformer) · ChrF++ **88.57** (Transformer) · Cặp câu **12.5M** (không dấu → có dấu)

Công nghệ: PyTorch, Transformer, LSTM, GRU, RNN, sacreBLEU.

Optimization for AI Inference Engines on GPUs 2025

Phương pháp đo đạc, profiling, phân tích

Tối ưu suy luận trên GPU cho ResNet50 và DistilBERT bằng FP16, INT8, TorchScript/torch.compile và ONNX.

Bài báo: PDF · Chi tiết: ngwkhai.com/vi/projects/gpu-inference-optimization

- Dự án tập trung tối ưu suy luận AI trên GPU cho ResNet50 và DistilBERT thông qua FP16, INT8, TorchScript/torch.compile và ONNX.
- Kết quả cho thấy FP16 mang lại cải thiện đáng kể, đặc biệt với DistilBERT: thông lượng tăng từ 305 → 1202 mẫu/s ở batch size 8 (≈4×).
- INT8 (TensorRT) đạt tăng tốc 2–5× nhưng đòi hỏi hiệu chuẩn cẩn thận để giữ được độ chính xác.
- Tối ưu đồ thị cho tăng tốc 1,2–1,8×, trong khi profiling CUDA xác định các nút thắt chính nằm ở kernel GEMM và ReLU.
- Dự án chứng minh mixed-precision và lượng tử hoá là những cách tiếp cận hiệu quả nhất cho suy luận quy mô lớn.

Kết quả chính: Thông lượng FP16 **≈4×** (DistilBERT, 305 → 1202 mẫu/s với batch size 8) · Tăng tốc INT8 **2–5×** (TensorRT, cần hiệu chuẩn cẩn thận) · Tối ưu đồ thị **1.2–1.8×**

Công nghệ: TensorRT, ONNX Runtime, TorchScript, torch.compile, Nsight, CUDA.

License Plate Recognition System 2025

Thiết kế hệ thống, tầng API và cơ sở dữ liệu, triển khai

Hệ thống nhận diện biển số xe thời gian thực: mô hình YOLOv5 fine-tune kết hợp OCR, backend FastAPI và PostgreSQL, giao diện React, triển khai bằng Docker.

Mã nguồn: github.com/ngwkhai/License-Plate-Recognition · Chi tiết: ngwkhai.com/vi/projects/license-plate-recognition

- Phát triển hệ thống nhận diện biển số xe thời gian thực bằng cách tích hợp mô hình YOLOv5 đã fine-tune với OCR để trích xuất văn bản.
- Tập trung vào phát triển hệ thống: thiết kế pipeline, xây dựng REST API bằng FastAPI, quản lý dữ liệu trong PostgreSQL và tạo giao diện dựa trên ReactJS. Triển khai hệ thống bằng Docker để dễ mở rộng và tích hợp.
- Đạt xử lý thời gian thực ở mức ~12 FPS với độ chính xác nhận diện đáng tin cậy.
- Nâng cao hiệu quả quản lý giao thông và giúp người dân thuận tiện tra cứu hồ sơ vi phạm giao thông trực tuyến.

Kết quả chính: Thông lượng **~12 FPS** (xử lý thời gian thực)

Công nghệ: YOLOv5, OCR, FastAPI, PostgreSQL, React, Docker.

Quantitative Trading System 2025

Toàn bộ — kỹ thuật đặc trưng, mô hình, pipeline huấn luyện

Hệ thống giao dịch định lượng viết từ đầu bằng PyTorch: mô hình Transformer dự đoán tín hiệu MUA/BÁN/GIỮ từ các chỉ báo tài chính và dữ liệu giao dịch lịch sử.

Mã nguồn: github.com/ngwkhai/Quantitative_trading_system · Chi tiết: ngwkhai.com/vi/projects/quantitative-trading-system

- Hệ thống giao dịch định lượng xây dựng từ đầu bằng PyTorch, dùng mô hình Transformer để dự đoán tín hiệu MUA/BÁN/GIỮ từ các chỉ báo tài chính và dữ liệu giao dịch lịch sử, với pipeline xử lý 3,4 triệu mẫu huấn luyện và 351 nghìn mẫu phát triển.
- Hệ thống tích hợp các đặc trưng như log-volume, chỉ báo kỹ thuật (MACD, RSI, Bollinger Bands, ATR) và Time2Vec với chuẩn hoá hai bước.
- Đạt Validation Accuracy 92.75% và Test Accuracy 92.50%.

Kết quả chính: Độ chính xác validation **92.75%** · Độ chính xác test **92.50%** · Mẫu huấn luyện **3.4M** (cùng 351K mẫu dev)

Công nghệ: PyTorch, Transformer, Time2Vec, MACD, RSI, Bollinger Bands, ATR.

Credit Card Fraud Detection 2024

Tiền xử lý, tái lấy mẫu, so sánh mô hình, đánh giá

Phát hiện gian lận trên bộ dữ liệu thẻ tín dụng của Kaggle, nơi chỉ 0,172% giao dịch là gian lận, so sánh bốn bộ phân loại cùng hai cách tái lấy mẫu SMOTE và ADASYN.

Mã nguồn: github.com/ngwkhai/Card-fraud-detection · Chi tiết: ngwkhai.com/vi/projects/credit-card-fraud-detection

- Thực hiện phát hiện gian lận trên bộ dữ liệu Kaggle (284.807 giao dịch, chỉ 492 giao dịch gian lận → 0,172%).
- Áp dụng SMOTE và ADASYN để xử lý mất cân bằng lớp nghiêm trọng.
- Cài đặt và so sánh Logistic Regression, Decision Tree, Random Forest và XGBoost.
- Đạt ROC-AUC 0.9817 và Recall 0.735 với XGBoost trên dữ liệu mất cân bằng.
- Cải thiện Recall lên 0.816 với ADASYN trong khi vẫn giữ ROC-AUC cao (0.9744), vượt các mô hình còn lại.
- Đóng góp vào tiền xử lý dữ liệu, tái lấy mẫu, phát triển mô hình và đánh giá, tạo ra một nguyên mẫu vững vàng cho phát hiện gian lận tài chính.

Kết quả chính: ROC-AUC **0.9817** (XGBoost trên dữ liệu mất cân bằng) · Recall **0.816** (với ADASYN, tăng từ 0.735) · Tỷ lệ mẫu dương **0.172%** (492 gian lận trong 284.807)

Công nghệ: XGBoost, Random Forest, Decision Tree, Logistic Regression, SMOTE, ADASYN.

Large-Scale Text Sentiment Classification 2024

Cài đặt pipeline và đo đạc so sánh

Phân loại cảm xúc 1,6 triệu tweet (Sentiment140) bằng Naive Bayes và SVM, chạy trên Apache Hadoop và Apache Spark.

Mã nguồn: github.com/ngwkhai/Text-Sentiment-Classification-Hadoop-Spark · Chi tiết: ngwkhai.com/vi/projects/text-sentiment-classification

- Dự án khảo sát khai phá văn bản, phân tích cảm xúc và tính toán song song/phân tán với Apache Hadoop và Apache Spark.
- Hệ thống được triển khai trên bộ dữ liệu Sentiment140 (1,6 triệu tweet) dùng Naive Bayes và SVM.
- Kết quả cho thấy Naive Bayes đạt 75% độ chính xác với tốc độ xử lý nhanh hơn, trong khi SVM đạt độ chính xác cao hơn (80%–82%) nhưng cần thời gian huấn luyện lâu hơn trên Hadoop.
- Spark vượt Hadoop về thời gian thực thi và khả năng mở rộng với dữ liệu quy mô lớn.

Kết quả chính: Dữ liệu **1.6M** (tweet, Sentiment140) · Độ chính xác SVM **80–82%** (huấn luyện lâu hơn trên Hadoop) · Độ chính xác Naive Bayes **75%** (xử lý nhanh hơn)

Công nghệ: Apache Hadoop, Apache Spark, Naive Bayes, SVM, Sentiment140.

YOLOv1 With CBAM From Scratch 2024

Cài đặt toàn bộ — dataset, hàm mất mát, module attention, huấn luyện

Cài đặt lại YOLOv1 từ đầu bằng PyTorch, tự viết lớp dataset và hàm mất mát, rồi bổ sung CBAM attention vào phần trích xuất đặc trưng.
Chi tiết: ngwkhai.com/vi/projects/yolov1-cbam

- Dự án dựng lại mô hình YOLOv1 từ đầu bằng PyTorch, trong đó tôi cài đặt FruitDataset và YoloLoss, đồng thời tích hợp CBAM (Convolutional Block Attention Module) để tăng cường trích xuất đặc trưng, với tổng cộng 269.168.434 tham số.
- Mô hình sử dụng bộ lập lịch ReduceLROnPlateau để tối ưu quá trình huấn luyện.
- Kết quả huấn luyện cho thấy Train mAP đạt 0.8829 và Best Validation mAP đạt 0.6994.
- Những kết quả này chứng minh CBAM mang lại cải thiện đáng kể cho hiệu năng phát hiện đối tượng.

Kết quả chính: mAP huấn luyện **0.8829** · mAP validation tốt nhất **0.6994** · Tham số **269.2M** (269.168.434)
Công nghệ: PyTorch, YOLOv1, CBAM, ReduceLROnPlateau.

KỸ NĂNG

Lập trình	Python, C/C++, Java, SQL, TypeScript
Học máy	PyTorch, Transformer, Seq2Seq, YOLOv5 / YOLOv1, CBAM, XGBoost, SMOTE / ADASYN, scikit-learn
Xử lý ngôn ngữ tự nhiên	Vietnamese NLP, Diacritic restoration, BLEU / ChrF++, Sentiment classification
Hệ thống & Triển khai	FastAPI, Docker, PostgreSQL, React, Apache Spark, Apache Hadoop
Suy luận & Hiệu năng	TensorRT, ONNX Runtime, TorchScript, torch.compile, FP16 / INT8, CUDA profiling
Ngôn ngữ	Tiếng Việt: Bản ngữ, Tiếng Anh: Thành thạo

GIẢI THƯỞNG & THÀNH TÍCH

Giải Ba, Giải thưởng Nghiên cứu khoa học sinh viên — Trí tuệ Nhân tạo, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội	2026
Giải Ba, Bảng Sinh viên — Humanitarian Logistics Hackathon	2026
Học bổng Khuyến khích học tập, Trường Đại học Công nghệ	2024
Giải Nhì, Kỳ thi Học sinh giỏi cấp tỉnh môn Vật lý, tỉnh Thái Nguyên	2023
Giải Ba, Kỳ thi Học sinh giỏi cấp tỉnh môn Tin học, tỉnh Thái Nguyên	2021

HOẠT ĐỘNG & LÃNH ĐẠO

Viện Trí tuệ Nhân tạo 09/2024 - Hiện tại

Lớp phó & Thành viên Ban Chấp hành

- Đảm nhiệm vai trò lớp phó và thành viên tích cực của Ban Chấp hành, điều phối các dự án sinh viên, tổ chức họp và tham gia hoạch định chiến lược của viện.