# VU HUU AN

**AI ENGINEER**

Cau Giay Dist, Hanoi City, VN | 085.901.5893 | anvuleo1204@gmail.com | github.com/AnKun10

## EDUCATION

**Hanoi University of Science and Technology** **Sep 2022 - Sep 2026**

School of Information and Communication Technology

- Graduated student majors in "Data Science & Artificial Intelligence" (CPA: 3.3/4.0).

**Luong The Vinh Hanoi High School** **Sep 2019 - June 2022**

Science Specialization Class

- Math CPA: 9.6/10.0 | Math (National Highschool Examination): 10.0/10.0.

## PROJECTS

**Query-aware Visual Token Pruning for Efficient Multi-turn Vision-Language Conversation** **Mar 2026 - Jul 2026**

Graduation Thesis (Supervisor: Assoc. Prof. Le Thanh Huong) - Role: AI Research Engineer

- Overview: Proposed a training-free, query-aware, two-stage visual token pruning method for LLaVA-1.5-7B that cuts 576 visual tokens per image down to 32/64/128 with little accuracy loss, plus a retain-token cache for multi-turn conversation.
- Architecture: Stage 1 selects tokens at the CLIP vision encoder by combining CLS-attention dominant tokens with farthest-point-sampling diversity tokens; Stage 2 re-scores the survivors at a middle LLM layer by attention from the question tokens, with a "no-slice decode" that keeps RoPE position ids and the KV cache correct.
- Multi-turn Caching: A per-image LRU cache reuses Stage-1 tokens, so later turns on the same image skip the vision encoder and save up to ~15% of per-turn latency.
- Insight: Measured query-vision attention concentration (entropy, max-share, top-k% mass) across LLM layers to justify the middle-layer (l=12) pruning point; ablations showed accuracy plateaus at a 1:1 dominant-diversity ratio.
- Benchmark: Evaluated on 7 benchmarks at 3 retain budgets via lmms-eval, covering hallucination (POPE), visual reasoning (GQA), perception and cognition (MME), science QA (ScienceQA-IMG), text reading and OCR (TextVQA, OCRBench), and low-quality imagery (VizWiz), with both accuracy and stage-level latency recorded.
- Evaluation: Against 4 baselines, one per family of related work, vision-encoder compression (VisionZip, CVPR 2025), LLM-layer pruning (FastV, ECCV 2024), query-conditioned sparsification (SparseVLM, ICML 2025), and diversity selection (DivPrune, CVPR 2025), the method reached the highest average accuracy at every budget (0.918 / 0.958 / 0.982 vs. the unpruned model) with up to 2.73× prefill speedup, and beat the unpruned baseline on POPE (1.011).
- **Stacks:** PyTorch, HuggingFace Transformers, lmms-eval, Gradio

**MADTempo:** **Aug 2025 - Nov 2025**

**An Interactive System for Multi-Event Temporal Video Retrieval with Query Augmentation**

SOICT 2025 Proceddings (First Author) - Role: AI Research Engineer

- Overview: Developed an interactive video retrieval system featuring a temporal search framework for reasoning over ordered multi-event queries, integrating multimodal representations to support scalable retrieval across large video collections.
- Dataset: Curated and processed a ~300 GB Vietnamese video corpus spanning news, cooking, and entertainment domains.
- Data Preprocessing: Designed a scalable video preprocessing pipeline using TransNetV2 for scene segmentation and perceptual hashing + neighborhood similarity for deduplication, reducing the corpus from ~300 GB to ~10 GB (~96.7% reduction) after scene extraction and further to ~7 GB (~97.7% total reduction).
- Feature Extraction: Architected a multimodal feature extraction and storage pipeline, indexing CLIP-Laion2B keyframe embeddings in Milvus and storing rich metadata in MongoDB, including object detection (YOLOv8), ASR (PhoWhisper-large), OCR (Vintern-1B), and image captioning (Qwen2.5-VL-3B).
- Deployment: Built a FastAPI-based service and containerized the entire system using Docker Compose, with automated embedding and metadata ingestion pipelines and migration scripts to ensure reproducible deployment and indexing.
- Hightlights: Efficient temporal search pipeline for multi-event retrieval, web-scale visual grounding for OOD queries.
- Evaluation: Evaluated the system on KIS, QA, and TRAKE tasks from the HCMAI 2025 competition, achieving 75.4 points in the Preliminary Round and an “Excellent” rating on the TRAKE task in the Final Round.
- **Stacks:** Pytorch, FastAPI, Huggingface, Milvus, MongoDB, Docker, LangChain, OpenAI

**Lina: A Personalized AI Learning Companion over an Agentic RAG Knowledge Hub** **Jul 2026 - Present**

Group Project - Role: AI Engineer

- Overview: Built and deployed to production an AI learning platform where students chat with a tutor, run deep research, generate quizzes, and follow study plans over a lecturer-curated document hub. Every answer carries page-level citations, and HITL checkpoints govern the platform (lecturers approve documents before indexing, and users approve research briefs and study plans before the agents act on them).
- Architecture: Designed the LangGraph multi-agent system. A tool-calling tutor routes each turn to grounded QA, exercise guidance, answer grading, or report synthesis; a deep research agent drafts a brief for HITL approval, then runs parallel researchers over the hub and whitelisted web search; a quiz pipeline plans, generates, and validates questions, rejecting any without citations.
- Personalization: Built long-term memory on mem0 (deterministic onboarding seeding, LLM fact extraction from chat, fail-soft throughout), a competency model that scores mastery from quiz results, and study plans that reason backwards from the learner's goal, with an approve-or-redesign loop, checkpoint quizzes, and scheduled email digests and deadline reminders.
- Retrieval & Ingestion: Hybrid search over Qdrant Cloud and BM25 with RRF, plus role, course, page-window, and private-session filters. Ingestion converts five document formats into heading-aware Markdown, takes browser uploads straight to GCS via signed URLs or Google Drive, and OCRs scanned Vietnamese pages through RapidOCR and a vLLM-served model.
- Safety: Rule-based guardrails against prompt injection and role confusion, PII and secret masking on output, a guide mode that walks students through graded exercises without revealing answers, and an LLM judge that verifies each answer against its retrieved context with one bounded regeneration before refusing.
- Frontend: Bilingual (Vietnamese/English) Next.js 15 workspace: streaming chat, a session sources panel, citations that open the exact source chunk and PDF page, a quiz hub with mastery tracking, and lecturer/admin dashboards for document review.
- Deployment: FastAPI with SSE streaming, JWT auth plus Google Sign-In, PostgreSQL with Alembic migrations; GitHub Actions builds Docker images to GHCR and deploys to Render and Vercel, with 175 test modules covering the 1,000+ commit codebase.
- Observability: Instrumented the full agent graph with Langfuse: per-node traces with input, output, latency, and token counts, user feedback and sender identity mirrored onto traces, and structured JSON logs sharing one request ID end to end, so any wrong answer can be debugged from a single trace.
- Evaluation: Wrote a RAGAS-based harness with 8 labeled benchmark suites, including red-team evals for direct/indirect prompt injection and data poisoning, plus multi-hop QA metrics and before/after routing reports; ran a role-played user test against production and shipped six waves of fixes for blocking bugs, plan quality, latency, and mobile UX.
- **Stacks:** LangGraph, Langfuse, LangChain, FastAPI, OpenAI, Qdrant, mem0, Tavily, RAGAS, Langfuse, PostgreSQL, Next.js, TypeScript, Docker, Google Cloud Storage, Render, Vercel.

**IPSP: An Image Processing and Storage Pipeline** **Oct 2025 - Dec 2025**

Group Project - Role: MLOps Engineer

- Overview: Developed a Lambda-based MLOps system for urban traffic analytics, with a primary focus on a distributed batch model training pipeline. The system integrates streaming analytics and large-scale batch processing to support scalable model training, versioning, and deployment in a production-like environment.
- Dataset: Utilized a dataset of 11,508 urban traffic images containing cars, buses, and trucks, collected from multi-camera traffic monitoring systems.
- Data Ingestion: Built a multi-threaded, partition-aware Kafka producer ingesting image metadata from LMDB files stored in MinIO, supporting 20 concurrent camera streams with configurable FPS and fault-tolerant, load-balanced message delivery.
- Stream Processing: Implemented PySpark Structured Streaming with 15-second micro-batches for real-time vehicle detection, classification, and speed estimation, using Redis-backed state management for cross-frame object tracking and writing enriched traffic analytics to MongoDB.
- Batch Processing: Designed a distributed YOLO training pipeline using a Kafka-based job dispatcher–worker pattern, where GPU-enabled workers consume training jobs asynchronously, support checkpoint resumption, perform automatic model versioning, and upload trained artifacts to MinIO for reproducible deployment.
- Visualization: Built Metabase dashboards on top of MongoDB to visualize real-time and batch traffic analytics, including vehicle counts, speed distributions, congestion trends, peak-hour analysis, and spatial heatmaps with low-latency querying.
- **Stacks:** Kubernetes, MinIO, Kafka, PySpark, Redis, MongoDB, Metabase, Ultralytics, Pytorch, OpenCV, Albumentations

**Restora Vision** **Oct 2024 - Dec 2024**

Group Project (Supervisor: Prof. Nguyen Hung Son) - Role: AI Engineer

- Overview: Developed an end-to-end image super-resolution framework targeting ×4 upscaling, benchmarking classical and transformer-based models (U-Net, FSRCNN, SwinIR) and implementing a custom SRGAN architecture with enhanced training losses, evaluated on standard SR benchmarks using PSNR and SSIM.
- Dataset: Trained models on DF2K (DIV2K + Flickr2K), comprising 3,450 training images and 100 validation images, and evaluated on standard SR benchmarks including Set5, Set12, Set14, and Urban100.
- Data Preprocessing: Applied standard SR data augmentation, including random cropping and horizontal flipping (p = 0.5), to improve model generalization and robustness.
- Training Loss: Designed a multi-component loss function for SRGAN, combining content loss (MSE for warm-up epochs followed by VGG perceptual loss), adversarial loss, and an additional total variation loss to promote spatial smoothness and reduce artifacts in generated images.
- Evaluation: Evaluated x4 super-resolution performance on standard benchmarks (Set5, Set12, Set14, Urban100) using PSNR and SSIM, achieving an average of 26.75 dB PSNR and 0.79 SSIM, with consistent performance across diverse image datasets.
- **Stacks:** Pytorch, Albumentations, Wandb, Streamlit

**Car Evaluation** **Apr 2024 - Jun 2024**

Group Project (Supervisor: Assoc. Prof. Than Quang Khoat) - Role: ML Engineer

- Overview: Built a machine learning regression pipeline to predict car prices in Vietnam, combining large-scale real-world data with feature enrichment, missing-value handling, and systematic model selection and hyperparameter optimization.
- Dataset: Utilized a Kaggle dataset sourced from bonbanh.com, containing 30,652 vehicles with diverse car attributes and timestamped prices, characterized by high missingness across feature dimensions.
- Data Preprocessing: Addressed ~6.7% feature sparsity in the original dataset by enriching vehicle attributes through web scraping with Selenium (from “fueleconomy.gov” and “fuelly.com”), followed by KNN-based imputation to handle remaining missing values.
- Train/Val/Test Split: Modeling & Training: Applied an 80/20 train–test split, hold-out cross-validation, and Optuna-driven hyperparameter optimization, combining grid search and random search strategies.
- Modelling: Implemented and benchmarked 10 regression models, with a primary focus on tree-based and boosting approaches (Decision Tree, Random Forest, AdaBoost, Gradient Boosting, XGBoost), alongside classical baselines (KNN, Polynomial Regression, SVR, Bagging, Stacking).
- EDA: Analyzed feature importance across multiple tree-based and boosting models, consistently identifying “year of manufacture”, “engine capacity”, “grade”, “brand”, and “mileage” as the strongest price drivers, while structural attributes (e.g., “doors”, “engine type”, “color”) showed comparatively lower impact.
- Evaluation: Evaluated tree-based and boosting models using $R^2$ and RMSE, achieving up to 0.96 $R^2$ and 0.075 RMSE on the test set (XGBoost). Boosting methods (AdaBoost, Gradient Boosting, XGBoost) consistently outperformed standalone tree models, demonstrating strong generalization with minimal train–test performance gap.
- **Stacks:** Selenium, Beautifulsoup, Pandas, Numpy, Sklearn, Optuna, Streamlit

## CERTIFICATES

- **Vietnam STEAM Challenge (2021):** AI-Powered Autonomous Car (Top 10)
- **IELTS (2021):** 7.0/9.0
- **SAT (2022):** 1460/1600
- **Foundation of ML (2024):** VietAI & New Turing Institute (Issuer)
- **Deep Learning Specialization (2024):** DeepLearning.AI (Issuer)
- **Retrieval Augmented Generation (2025):** DeepLearning.AI (Issuer)
- **AI All in One (2025):** AI Viet Nam (Issuer)
- **HCMAI Challenge (2025):** MADTempo (Finalist - Promising Solution Prize)
- **AWS Cloud Practitioner (2026):** AWS (Issuer)

## ACTIVITIES

- **LTV STEAM Club (2021):** Member
- **STEAM for Vietnam - Train the Trainers - AI for SuperTeacher (2025):** Teacher Assistant