Ayush Ranjan

aranjan1@ucsc.edu | (831) 266-5973 | linkedin.com/in/ayuranjan
github.com/ayuranjan | ayuranjan.github.io

Summary

Engineer with **3.5 years of experience** across **backend**, **full-stack**, and **AI systems** (**2.5 years industry + 1 year AI research**). At **Capgemini**, I led **data modeling** and built **Java-based diagnostic tools** for Mercedes-Benz. At **UCSC**, I developed **AI agents**, **vector search**, and **multimodal AI applications** using **Python**, **LangChain**, and **FastAPI**. Currently researching **RAG system reliability** and **LLM fine-tuning** in the **AIEA Lab**. Skilled in **Java**, **Python**, **SQL**, and **LangGraph**, with a focus on **scalable, production-grade systems**.

Education

University of California, Santa Cruz **Sep 2023 – Aug 2025**
Master of Science (MS) in Computer Science CGPA: 3.92/4

- **Relevant Coursework:** Analysis of Algorithms, Design and Implementation of Database Systems, Deep Learning for Advanced Computer Vision, Artificial Intelligence (AI), Applied ML: Deep Learning (DL), Computer Networks
- **Teaching Assistant Roles: 4x TA for Database Systems (CSE-180/181/182)**: led labs/projects on **SQL**, **PostgreSQL internals**, **transaction management**, **indexing**, **stored functions**, **PL/pgSQL**, **ETL workflows**, **data modeling**, and **query optimization**. **1x TA for Software Engineering (CSE-115A)**: mentored student teams on **Agile development**, version control, and team collaboration practices.

Manipal University, Jaipur **July 2017 – May 2021**
Bachelor of Technology (B.Tech.) in Information Technology

- **Relevant Coursework:** Operating Systems, Data Mining and Warehousing, Data Science, Cryptography and Network Security, Advanced Data Structures, Natural Language Processing, Advance Machine Learning Techniques

Work Experience

AI Explainability and Accountability (AIEA) Lab, UCSC ***Santa Cruz, CA***
Graduate Researcher *Oct 2024 – Present*

- Conducted applied research to improve the reliability and explainability of LLM-based university chatbots, focusing on campus-wide use cases such as enrollment, deadlines, housing, and course queries, leading to enhanced user satisfaction.
- Designed and evaluated **10+ advanced RAG workflow** architectures (Classic RAG, Chain of Thought, Agentic RAG, Adaptive RAG, Corrective RAG, RAT RAG) using comprehensive Ragas evaluation framework.
- **Fine-tuning open-source LLMs** to align with university-specific tone, structure, and factual accuracy, enabling domain-adaptive generation for student and administrative queries.
- Achieved consistent performance improvements across all architectures, **with an average 35-50% enhancement** over baseline RAG systems, where different approaches excelled in specific metrics (**faithfulness, answer relevancy, context precision**) depending on query complexity and domain requirements.
- Developing production deployment pipeline using Docker containerization, Kubernetes orchestration, and FastAPI endpoints with automated CI/CD workflows, load balancing, and monitoring systems for scalable campus-wide chatbot implementation.

Information Retrieval and Knowledge Management Lab, UCSC ***Santa Cruz, CA***
AI Research Intern *July 2024 – Sep 2024*

- Partnered with a **stealth hardware startup** to develop a **0-to-1 multimodal AI agent** for smart wearable devices (camera-integrated earphones), implementing wake word detection, intent classification, and real-time audio-visual processing for calorie estimation, emergency response and video summarization.
- Designed intelligent query routing system with **95% accuracy** in classifying continuous vs. new queries, integrating **Dialogflow for 8+ pre-built workflows** (calorie estimation, contact calling, emergency location services) and custom **LangGraph agents** for open-domain conversations.
- Engineered real-time multimodal data fusion system combining **audio transcription** (Whisper), **computer vision** (food segmentation, depth estimation), and vector similarity search for intelligent fallback routing to **external tools (web search, OCR)** when confidence scores dropped below 0.8 threshold.
- Developed a multi-threaded memory manager to asynchronously encode and cache historical observations (images, transcripts) into vector embeddings using **Hugging Face transformers**, with storage in **Pinecone**.
- Integrated the prototype with a **local edge pipeline** (**FFmpeg**, Whisper, and custom vision models), **achieving sub-500ms inference latency for key commands** and enabling real-time calorie detection via food segmentation and depth estimation.

Capgemini Technology Services India Limited ***Mumbai, India***
Software Engineer I & Software Engineer II *July 2021 – Aug 2023*

- **Software Engineer II** (Oct 2022 – Aug 2023)
  - **Headed the Data Modeling Team** for Mercedes-Benz's XDIS platform, focusing on backend schema evolution for **vehicle network topology change requests** (e.g., ECU reconfigurations, bus architecture edits).

    - Designed a lightweight **ETL pipeline in Java** to process large XML diagnostic files: extracted raw telemetry data, transformed it into updated entity structures, and loaded it into IBM **Db2** tables—supporting seamless data migration.
    - Wrote and tuned complex **SQL queries and views** in Db2 to support schema validation, relational consistency checks, and historical topology comparisons for Change Request(CR) automation workflows.
    - Achieved **3rd Place at Innocircle 2022**, Mercedes-Benz Internal Innovation Forum by implementing a micro frontend architecture that enabled users to modify their vehicle network topology and review changes, reducing process time by over 50%
    - Developed an **AI-assisted validation system** for over 2,500 historical Change Requests by embedding symbolic vehicle network topologies using custom **Word2Vec** and **Sentence-BERT models**, which flagged rare configurations and recommended optimal topologies, improving validation accuracy.
- **Software Engineer I** (July 2021 – Sep 2022)
    - **Role**: **Java Backend Developer**
    - Worked on **XDIS**, a **SOAP**-based diagnostic tool structured around a three-tier monolithic Java architecture used by **Mercedes-Benz** service teams for vehicle automation and diagnostics.
    - Implemented and maintained backend modules using **Core Java**, **JAXB**, and **JDBC**; performed **data modeling** for diagnostic entities; applied **design patterns** and contributed to performance-critical sections of the legacy system.
    - Reduced XML migration time by **67%**, improving daily workflow efficiency for 50+ Mercedes-Benz service teams. Also implemented indexing strategies for associated IBM Db2 database tables.
    - Optimized export testing by creating a wrapper around the Autosar framework and implementing an **XML file** import strategy, **reducing overall testing time by 40%** and speeding up export time for **individual modules by 17% on average**.
- **Software Engineer Intern** (Jan 2021 – May 2021)
    - **Role**: **Java Full Stack Developer**
    - Built a full-stack **Medical Portal**: **Spring Boot REST MVC** backend exposing 17 JSON CRUD/search endpoints documented with **Swagger**, paired with a **React** single-page application using **Redux** and **Axios**, giving the team faster turnaround on new features.
    - Integrated **MySQL** via JPA/Hibernate and secured the APIs with **Spring Security** and JWT, implementing role-based access control (**RBAC**) for Admin, Doctor, and Patient roles; added automated tests with **JUnit**, **Jest**, and **React Testing Library** for reliable releases.
    - Containerized the stack with **Docker Compose**, set up a **GitHub Actions** CI pipeline, and deployed to a local **Minikube Kubernetes** cluster to provide quick, environment-consistent demos for QA and stakeholders.

Selected Projects

PgVector+ | ***C/C++, PL/pgSQL, Database Systems, Vector Similarity*** Ppt ***Jan 2024 – Mar 2024***

- Designed and built a **custom PostgreSQL extension** on top of pgvector to support hybrid similarity-dissimilarity search and low-level query composition, bridging gaps in vector DB functionality seen in systems like Pinecone and Qdrant.
- Prototyped a `compound_similarity()` operator in **PL/pgSQL** to support similarity search queries like "similar to X, but unlike Y" using cosine and inner-product thresholds.
- Prototyped PL/pgSQL-based `search_similar_vectors()` function to simulate centroid-based multi-query composition and validate set-based similarity retrieval.
- Earned an **A grade** in CSE 215 for system-level innovation in vector search acceleration and database extensibility.

Unveiling Glitches in CLIP | ***Hugging Face, Python, Vector Database, Prompt Engineering*** Arxiv ***Jan 2024 – March 2024***

- Conducted in-depth analysis of the CLIP model's image comprehension capabilities. Identified and documented **14 systemic faults**, including **four novel faults**, impacting CLIP's interpretation of images using **two novel methodologies**.
- Implemented the Discrepancy Analysis Framework (**DAF**) to analyze discrepancies in image similarity rankings between CLIP and **DINOv2** and utilized **OpenAI's GPT API** to identify and analyze faults systematically. Utilized the Transformative Caption Analysis for CLIP (**TCAC**) approach to evaluate CLIP's response to transformations applied to images.
- Achieved **A+ grade** in CSE 290D Neural Computation at UCSC for this project.

Video to MP3 Converter Service | ***Python, Flask, Docker, Kubernetes, RabbitMQ, MongoDB*** GitHub***Dec 2023 – Jan 2024***

- Designed a modular **microservices-based** system with four components: auth service, API gateway, uploader, and converter.
- Used **FastAPI** and **Flask** for building secure, high-performance REST endpoints, authenticated with **JWT tokens** and **role-based access control**; ensured protected video operations via the centralized API gateway.
- Enabled asynchronous video-to-MP3 conversion using **RabbitMQ** and **MoviePy**, supporting non-blocking task execution.
- Deployed services in **Docker** containers and orchestrated with **Kubernetes** via **Minikube** for scalable local development.

Technical Skills

Programming Languages: Python, Java, TypeScript, JavaScript, C/C++, Go, Rust, SQL

AI/ML Tooling: NLTK, spaCy, transformers library, Sentence-BERT, Word2Vec, MCP Protocols, MLflow, Prompt Engineering

Backend Development: Spring Boot, REST APIs, GraphQL, FastAPI, Flask, PL/pgSQL, JDBC, Node.js, Express.js

Frontend Development: React, Next.js, Redux, Axios, HTML/CSS, Swagger / OpenAPI, JWT, RBAC

Databases: PostgreSQL, MySQL, Oracle, IBM Db2, MongoDB, Redis, Vector Databases (pgvector, Pinecone, Milvus, Qdrant)

Data Engineering: ETL (Java/XML), Liquibase, Pandas, NumPy, Apache Kafka, Apache Spark

DevOps & MLOps: Docker, Docker Compose, Kubernetes, Minikube, Git, GitHub Actions, Jenkins, Maven, Gradle, Terraform

Testing and Automation: JUnit, Mockito, Jest, React Testing Library, Playwright, TDD, BDD

Cloud and Distributed Systems: AWS (EC2, S3, RDS, Lambda, DynamoDB), Amazon Bedrock, Google Cloud(GCP), Azure

Practices & Miscellaneous: Agile/SAFe, CI/CD, SDLC Lifecycle, Prometheus, Grafana, Apache Airflow