YUHENG DING

706-238-1952 ◇ github.com/AaXDing ◇ linkedin.com/in/yuheng-ding

EDUCATION

Carnegie Mellon University Expected 12/2026

MS, Intelligent Information Systems

University of California - Los Angeles 09/2021 – 06/2025

BS, Computer Science & Applied Mathematics, GPA: 3.88 / 4.0

• Courses: Algorithm and Complexity, Computer Network, Machine Learning, Database Systems, Data Structure, Computer Organizations, Deep Learning, Computer Vision, Natural Language Processing, Graph Theory, Linear Algebra.

TECHNICAL SKILLS

Java (Tomcat, Spring Boot), **Python** (Django, PyTorch, NumPy), **C++**, **SQL** (MySQL, PostgreSQL, SQLite), **Redis**, **MongoDB**, **JavaScript** (React), **AWS**, **Azure**, **Git**, **Bash**, **Linux**, **Docker**, **Kubernetes**, **HTML**, **CSS, Spark**

PROFESSIONAL EXPERIENCE

VisionX | Full-Stack Software Engineer Intern 07/2023 - 10/2023

- Collaborated with a team of four engineers to architect a **full-stack** e-commerce platform, with a focus on promoting a healthy lifestyle and fostering environmental sustainability.
- Engineered a product page UI template using **Python Django** framework for backend and **MongoDB** for data persistence; streamlined automatic page creation from database; enhanced scalability and amplified product creation efficiency by 400% over manual page inputting.
- Implemented user information gathering, payment workflow with **Stripe API**, and email order confirmations; achieved a 30% reduction in checkout process response time.
- Collaborated with a colleague to enhance online consulting session reservation system, integrating voice and video chat functionalities by using **WebSocket** and **WebRTC** technologies, resulting in 60% decrease in reservation confirmation time.

RESEARCH EXPERIENCE

UCLA-NLP Mentor: Kai-Wei Chang, Nanyun Peng | Undergrad Researcher 06/2024 - Present

- Developed a comprehensive benchmark for evaluating visual language models (VLMs) on perception and reasoning quality in QA tasks, with testing on over 25 VLMs to assess self-critique and correction abilities. (**CVPR 25' Accepted**) Link
- Developed an intuitive annotation tool that improved the annotator experience, enabling efficient curation of over 1,600 fine-grained annotated samples for a high-quality dataset.
- Introduced a novel evaluation metric, namely VISCore, which incorporated answer accuracy, step-wise critique accuracy, and critique quality, leading to more accurate assessments.
- Implemented an agent-based critique generation method at each reasoning step, achieving an 8% increase in critique accuracy.

UCLA ARNI Lab Mentor: Christina Fragouli | Research Intern 07/2023 - 09/2023

- Discussed and refined a graph theory solution to minimize tests and time used in identifying infection within a community network using **Python**.
- Developed data pre-generation script using **Python**, resulting in a 70% reduction in test time for a population size of 20k.
- Improved testing script with Jupyter and **JSON** that enabled test creation and logging with 20% overhead reduction.
- Originated a community-aware diagonal splitting algorithm using a divide-and-conquer strategy which factored in community structure; the new algorithm needed 10% less tests compared to 5 other commonly-used algorithms.

PROJECT EXPERIENCE

Job-Search | Independent Developer 10/2023 - 01/2024

- Constructed a job search platform that personalized job recommendations merely based on user views and likes; incorporated Search, Favorite, and Recommendation functionalities in **Java Servlets** with **REST APIs** to efficiently handle HTTP requests and responses.
- Implemented frontend UI components using **JavaScript**, **HTML**, **CSS**, and **Ajax**, facilitating user login and enabling users to efficiently search and apply for positions.
- Designed AI-based keyword extraction algorithms and history-based recommendation to implement a job recommendation engine; Prioritized user safety through session-based authentication.
- Configured and deployed on **AWS EC2** virtual machine, combining **Redis** and **Amazon RDS** to cache and persist search and favorite related data; scaled the system to be able to serve 10k queries per second (QPS) and reduced latency by 90%.