# Yijia Gao

734-373-8675 | yijiagao@umich.edu | https://github.com/YijiaG-Jessica

## EDUCATION

**University of Michigan, Ann Arbor** **Ann Arbor, MI**

- **Degree:** Master of Computer Science in Engineering ***Expected Graduation: Dec. 2024***
- **Coursework**: Distributed Systems, Networking, Operating Systems, Parallel Computing, Privacy Technologies, Compilers

**University of Michigan, Ann Arbor** **Ann Arbor, MI**

- **Degree:** Bachelor of Science in Data Science (Dual Degree) ***Sep. 2021 - May 2023***
- **Coursework**: Web Systems, Data Structures and Algorithms, Information Retrieval, Embedded Systems, Machine Learning, Database Management Systems, Software Engineering, Computer Security

**Shanghai Jiao Tong University – UM-SJTU Joint Institute** **Shanghai, China**

- **Degree:** Bachelor of Science in Electrical and Computer Engineering (Dual Degree) ***Sep. 2019 - Aug. 2023***
- **Coursework**: Signals and Systems, Circuits, Semiconductor, Linear Algebra, Probability, Discrete Math

## WORK EXPERIENCE

**Zoox** **Foster City, CA**

***Software Engineering Intern - Asset Management System*** ***May 2024 - Present***

- Designed and built an end-to-end source-of-truth system for hardware-in-loop test assets with a Typer CLI front-end, FastAPI back-end, and MongoDB storage, which enabled efficient CRUD operations for internal developers.
- Productionized the service by deploying the API and database as containerized Docker images running on Linux servers.
- Scaled service to be used by 50+ internal test engineers from the software, firmware, and hardware validation teams.
- Collected data from sources upon communication with stakeholders and established ETL process with robust model validation.
- Unit tested using PyTest mocking methods and integration tested by fuzz-testing the deployed API with a mocked database.
- Wrote detailed documentation such as a CLI handbook, system design documentation, and database ER diagrams.

**University of Michigan, Ann Arbor** **Ann Arbor, MI**

***Full-Stack Research Developer - Graph Database*** ***May 2022 -  Present***

- Designed, developed, and maintained an end-to-end genomic information graph editor web application capable of querying genomic relationships using React, Flask, HTML, Neo4j, Vis.js (https://gkb.dcmb.med.umich.edu/).
- Increased traffic to over 1.6K active users from 61 countries, and continuing to attract more users.
- Added information logging process and embedded Google Analytics APIs at the front-end to track user traffic data.
- Reduced load at the front-end and increased query API retrieval efficiency by caching historic user query information.
- Co-authored the paper "GenomicKB: a Knowledge Graph for the Human Genome". Published in "Nucleic Acids Research" Journal. (DOI: https://doi.org/10.1093/nar/gkac957)

## PROJECT EXPERIENCE

**Network File Server**

- Implemented a multithreaded network file server that supports concurrent requests for filesystem operations including reading, writing, creating, and deleting files/ directories from multiple clients.
- Maximized concurrency using Boost library's reader-writer lock.
- Minimized expensive disk I/O by eagerly detecting and rejecting invalid requests, as well as caching directory entry indices.

**Reddit Posting Helper**

- Developed a Reddit tag-suggestion and post-retrieval model using Bidirectional Encoder Representations from Transformers (BERT), web crawl, text rank, and tf-idf methods to improve content creation on r/uofm Reddit page.
- Improved content relevance and search accuracy with BERT-phrase by ~30%. Benchmarked performance for keyword extraction methods: BERT (phrase and unigram) and text rank.

**Thread Library**

- Developed a kernel-level library in C++ that supports thread creation, context switching, and interrupt handling.
- Implemented synchronization primitives such as mutex and condition variables.
- Designed a round-robin scheduler to maximize fairness, favor interactive workloads, and avoid starvation.

**Video Streaming via CDN & DNS**

- Implemented adaptive bitrate selection on HTTP proxy server to stream video at high bit rates from the server to clients.
- Built a DNS server for load balancing based on geographical information using Dijkstra's algorithm.

**Cloud-local Joint Energy Coordination Platform**

- Developed a cloud-based energy scheduling system simulator by integrating centralized cloud-based scheduling and prediction, networking layer, and distributed local controls based on Raspberry Pi.
- Validated communication framework and energy simulation knowledge under cloud-local energy management.

**Dog Breed Detection CNN Model**

- Developed a supervised deep learning algorithm using convolutional neural networks (CNNs) to classify dog images by breed, achieving accurate breed identification with an accuracy of ~65% on the testing dataset.
- Utilized transfer learning through supervised pre-training and data augmentation techniques to enhance classification abilities and improve model robustness by ~15% on the testing dataset.

## SKILLS

**Programming Languages:** Python, C/C++, JavaScript, Go, Rust, SQL, MongoDB, MATLAB, HTML/CSS, Shell

**Tools/Frameworks:** Git, Linux, AWS, React, Redux, Flask, FastAPI, Click, Typer, Docker, PyTest, MongoMock, MUI, NumPy