

# Yijia Gao

734-373-8675 | [yjiagao@umich.edu](mailto:yjiagao@umich.edu) | <https://github.com/YijiaG-Jessica>

## EDUCATION

### University of Michigan, Ann Arbor

- **Degree:** Master of Computer Science in Engineering **Expected Graduation: May 2025**
- **Coursework:** Operating Systems, Parallel Computing, Computer Networking, Privacy Enhancing Technology

### University of Michigan, Ann Arbor

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

### Shanghai Jiao Tong University – UM-SJTU Joint Institute

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

## WORK EXPERIENCE

### Zoox, Foster City, CA

#### Incoming System and Site Reliability Intern

**Expected: May 2024 - Aug. 2024**

- Expected to build a centralized debug data infrastructure that supports collection, organization, and analysis
- Expected to use tools such as TypeScript, Azure, Docker, Pandas, and other data analysis frameworks

### University of Michigan, Ann Arbor

#### Research Assistant: Graph Database Full Stack Development

**May 2022 - Present**

- Designed, developed, and maintained an end-end genomic information graph editor web application capable of querying genomic relationships using React, Flask, HTML, Neo4j, Vis.js (<https://gkb.dcmdb.med.umich.edu/>)
- Increased traffic to ~1500 active users from 49 countries, and continuing to attract more users
- Added information logging process and embedded Google Analytics APIs at front-end to track user traffic info
- Reduced load at the front-end and increased query API retrieval efficiency by caching duplicated user queries
- 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

### Client-Server Instagram Replica

- Implemented in Python, JavaScript, HTML, SQL, and React to build an Instagram replica to practice knowledge behind Client-Server applications, Rest APIs, and web system development
- Led team to build web architecture and full-stack design of the application

### Reddit Posting Helper

- Developed a Python-based tool utilizing web crawl, text rank, BERT, and tf-idf methods to enhance content creation and posting on r/uofm, with features including tag suggestion and post retrieval
- Benchmarked performance for keyword extraction methods: BERT(phrase and unigram) and text rank. Improved content relevance and search accuracy with BERT-phrase by ~30%

### Network File Server

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

### Thread Library

- Developed a kernel-level library in C++ focusing on synchronization mechanisms, including mutexes and conditional variables, and covered thread creation, context switching on CPUs and interrupt handling.
- Applied RAIL lock techniques for automated lock management and used smart pointers to prevent memory leaks.

### 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
- Implemented DNS server for load balancing based on geographical information through 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 testing dataset

## SKILLS

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

**Tools/Frameworks:** Git, React, Redux, Flask, NumPy, AWS, Linux, VS Code, IntelliJ, Docker