Ruoyu (Lanny) Wang

Irvine, CA 92617
(949) 254-8225 | ruoyuw15@uci.edu | linkedin.com/in/lannywong2000

Education

University of California, Irvine Irvine, California
Master of Computer Science, GPA: 4.0/4.0 *Sept 2022 – Dec 2023*

University of California, Berkeley Berkeley, California
Berkeley International Study Program, Data Structures: A, Machine Structures: A *Aug 2019 – Dec 2019*

South China University of Technology Guangzhou, China
B.Eng. in Computer Science & Technology, GPA: 3.8/4.0 *Sept 2018 – July 2022*

Technical Skills

Languages: C++, Java, Python, JavaScript/HTML/CSS, C
Technologies: MySQL, MongoDB, Redis, Node.js, Express.js, React.js, Vue.js, Flask, Tensorflow, Keras
Tools: Git, Linux, Docker, AWS, Visual Studio Code, IntelliJ, Jupyter Notebook, Agile, Scrum

Professional Certificates

AWS Certified Developer – Associate *Jan 2023 – Jan 2026*
Amazon Web Services (AWS)

Professional Scrum Master™ I (PSM I) *Jan 2023 – No Expiration*
Scrum.org

Internship Experience

Tencent Technology (Guangzhou) Co., Ltd. *July 2021 – Aug 2021*
Software Engineer Intern, WeChat Channels Recommender System Team *Guangzhou, China*

- Implemented a filter to remove candidate videos of duplicate authors during candidate generation using **C++**
- Collaborated on maintaining the fast response time of online recommendation service under the rapid growth of video quantity, designed a candidate retrieval strategy to evenly distribute workload among multiple models
- Completed an online concurrent candidate generation tool of the recommender system using **OpenMP**

Publications

Active Learning-Based Optimization of Scientific Experimental Design
2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)

- Proposed a regression prediction model framework of matrix factorization algorithm Alternating Least Square combined with Deep Neural Network for non-characteristic datasets
- Proposed an Active Learning Query Strategy based on Expected Loss Minimization (ELM)
- Trained the proposed prediction model by the proposed Active Learning framework using **Tensorflow**
- Results showed that the experimental design was automatically and actively optimized by the ELM Query Strategy

Projects

PeterDB, A Database Management System for SQL | *C++, GDB, Valgrind, GoogleTest*

- Implemented a DBMS supporting CRUD operations and versioning of tables and records from scratch using **C++**
- Designed a hierarchy with paged file, record-based file and relation managers for bottom-up implementation
- Accelerated the process of range searching by maintaining self-implemented **B+ tree** index files
- Built a query engine realizing several relational operators, including filtering, projecting, joining, and aggregation
- Used **GDB** for debugging, **Valgrind** for checking memory leak, and **GoogleTest** for writing unit tests

TechieHR, An Online Video Interviews Platform | *React.js, Node.js, Express.js MongoDB, Redis, Docker, AWS*

- Completed a website allowing user login, chatting, and video conferencing through WebSocket
- Implemented frontend using **React.js**, and used **Node.js** and **Express.js** as backend support
- Used **MongoDB** to store user information and chat history, and used **Redis** to store login sessions
- Developed using **Docker** to ease the process of deployment, deployed on **AWS** EC2 server

Semantic Segmentation of 2D Satellite Images | *Python, Flask, React.js, Docker, OpenCV, Keras*

- Programmed a web crawler in **Python** to collect sample 2D satellite images from the data source website
- Implemented a web-based semi-automatic image annotation tool using **React.js** as frontend and **Flask** as backend based on an open-source tool called LOST. Used **Docker** for container building and deployment on a cloud server
- Reduced annotation time by using **OpenCV** for semi-automatic object contour detection during image annotation
- Designed a U-Net-based semantic segmentation model with Focal Loss and trained the model using **Keras**, reaching the highest classification accuracy mIoU of **67.15%** among other models using FCN, SegNet, and baseline U-Net

Logistics Location Planning (LLP) based on Local Greedy Genetic Algorithm | *Evolutionary Algorithm, Matlab*

- Modeled the LLP problem as distance minimization between different locations of distributors and retailers
- Proposed an intelligent Genetic Algorithm (GA) variant with a local greedy crossover operator, implemented the algorithm in **Matlab**, and reduced the solution time by **99.9%** compared with the Brute Force algorithm
- Analyzed the convergence rate of the algorithm, investigated the hyperparameters, and determined the combination of crossover probability and mutation probability to optimize the algorithm's performance

Online Chatbox | *Vue.js, MySQL, Redis*

- Developed a chatting website allowing user login and communication through WebSocket
- Implemented frontend using **Vue.js**, and used BaaS provided by ByteDance as backend support
- Used **MySQL** to store user information and chat history, and used **Redis** to store login sessions

Online E-Commerce Website | *Java Web, MySQL*

- Built a **Java Web** project and used an **MVC** design pattern as the overall framework
- Created tables for sellers, customers, products, shopping carts, etc. in **MySQL** database
- Deployed the project on the **Tomcat** server and connected to the MySQL database

Mountable Linux Filesystem in Userspace | *C, Linux, FUSE, Makefile*

- Designed a storage space for the file system, divided the space into equal-sized disk blocks, including a superblock for system description, a bitmap block for block usage records, and data blacks for data storage
- Implemented **C** functions realizing **Linux** file operations, including mkdir, mknod, write, read, etc
- Implemented the **FUSE** operations structure with the pointers of the implemented functions
- Wrote a **Makefile** specifying file dependencies and GCC command for disk initialization and executable generation

Gitlet, A Simplified Version-Control System | *Java*

- Used SHA-1 for unique identification of files and maintained a directed acyclic graph of commits
- Implemented the basic features of **Git** including saving, storing, branching, and merging files in **Java**

A RISC-V CPU Simulation based on LogiSim | *RISC-V, Assembly*

- Created a processor comprised of an ALU supporting basic arithmetic operations, a PC, and nine registers
- Implemented the datapaths and controls for a subset of **RISC-V** ISA using a two-stage pipeline
- Wrote the scripts of unit tests, integration tests, and edge-case tests for functional verification