

Cheng-Yi Tang

Irvine, CA 92614 | (626) 742-4068 | chengyit@uci.edu | [LinkedIn](#) | [Github](#) | [Personal Website](#)

EDUCATION

University of California, Irvine **Irvine, CA**
Master of Software Engineering, GPA: 3.88/4.00 Sept. 2024 - Dec. 2025 (Expected)

- Courses: Distributed Software Architecture, Cloud and Security Dependability, Software Testing and Debugging

National Chengchi University **Taipei, Taiwan**
Bachelor of Science in Management Information Systems, GPA: 3.61/4.30 Sept. 2019 - June 2023

- Research Assistant: Human-Automation Interaction Lab ([Research publications](#))
- Courses: Database Systems, Computer Network, Applied Machine Learning, Data Structures, Algorithms

EXPERIENCE

Raydium Semiconductor Corporation **Hsinchu, Taiwan**
Software Engineer Intern, Touch IC Hardware Design Team June 2024 - July 2024

Touch and Display Driver Integration Deep Learning Model

- Designed and implemented a lightweight depthwise CNN architecture (0.06 MB, 14K params, 18.46M MACs) for touchscreen environment classification, achieving 94% accuracy on 71K capacitance training samples using **PyTorch**.
- Optimized model architecture through 2D-to-1D tensor reshape, reducing parameters by 17%, MACs by 52%, and hardware resource utilization while maintaining 93% accuracy.
- Enhanced data quality through baseline canceling for noise reduction and augmentation for class balance.
- Validated model effectiveness through t-SNE visualization, demonstrating clear separation of operating environments.

Intel Corporation **Taipei, Taiwan**
Datacenter Technical Sales Specialist Intern, DCAI Platform Sales Enablement Team July 2022 - July 2023

Design-win Project Tracking Tool

- Developed **Python** Dash desktop application enabling PMs and FAEs to monitor over 500 design-win projects.
- Automated data analytics and report generation using **Pandas**, reducing project tracking time from 4 hours to 1 sec.
- Created interactive **Plotly** dashboard for visualization, enabling real-time data analysis and reporting.

SKILLS

Languages	Java, Python, JavaScript, TypeScript, C, C++, R, SQL
Web Development	React, Node.js, Django, Flask, Spring Boot, HTML, CSS
ML & Data Science	PyTorch, TensorFlow, Keras, NumPy, Pandas, Scikit-learn, OpenCV
DevOps & Databases	Git, Docker, CI/CD, AWS, Linux, Unix Shell, MySQL, MongoDB, Cassandra

SOFTWARE PROJECTS

Apache Cassandra CQL Test Suite Enhancement | Java, JUnit, JaCoCo, GitHub Actions, Docker [[GitHub](#)] Mar. 2025

- Implemented partition testing for CQL query using **JUnit**, covering boundary conditions across multiple data types.
- Designed tests using Finite State Machine modeling to verify all stages of CQL query processing pipeline.
- Improve code coverage from 29% to 60% using **JaCoCo**, with method coverage increasing from 52% to 88%.
- Set up **GitHub Actions** CI/CD workflow to automate testing and ensure code reliability.

Mailbag System | Node, React, TypeScript, Docker, AWS [[GitHub](#)] Nov. 2024

- Built a full-stack email system using **Node.js** and **Express.js** for REST API integration with IMAP/SMTP protocols.
- Engineered frontend client with **React** and **TypeScript**, implementing secure email operations and real-time updates.
- Dockerized application using **Docker Compose**, optimizing deployment workflow and ensuring consistency across environments. Deployed application on AWS EC2.

Efficient Capture & Helpful Output with LLM | Flask, React, Ollama, Python, JavaScript [[GitHub](#)] Oct. 2024

- Developed a full-stack platform (**Flask/React**) for processing multi-modal educational content (audio, video, PDF).
- Engineered LLaMA-based processing pipeline with custom prompts, generating educational materials within 15 sec.
- Implemented backend services using Flask, REST APIs, and **Ollama** for model interaction and content processing.

Real-time Sitting Posture Detection | Python, TensorFlow, Keras, OpenCV, MediaPipe [[GitHub](#)] Jan. 2023

- Developed real-time posture detection system with **TensorFlow** and **OpenCV**, achieving 98% accuracy (VGGNet16).
- Processed 450 images using **MediaPipe** for automated feature (skeleton) extraction and **Keras** for data augmentation.