Robin Lee

(510) 248-9750 rblee@scu.edu github.com/roblee04 roblee04.github.io linkedin.com/in/robinblee

Education

Santa Clara University — Spring 2026 **GPA 3.96**
B.S / M.S. Computer Science and Engineering, School of Engineering

Experience

Santa Clara University Machine Learning Research Lab – Santa Clara, CA Jun. 2025 — Current
Graduate Researcher

- Utilized triangulation, multiobject tracking, and ReID techniques to project Toyota's Woven Traffic Safety (WTS) traffic camera dataset footage into a 3D space, leading to the creation of high fidelity digital twins using Unreal Engine and NVIDIA Composer
- Expanded the WTS dataset with synthetic data generation by **400%** via NVIDIA IsaacSim
- Explored generative video-to-video models to generate synthetic accident data based on camera views from the WTS dataset
- Training models for accident anomaly anticipation; aiming to predict and detect an accident's occurrence within a future time horizon

Santa Clara University IoT Research Lab – Santa Clara, CA
Undergraduate Researcher Jun. 2023 — Sep. 2023

- Conducted supervised independent research on IoT security, internet architecture and protocols, and SOA network hardware
- Leveraged WiFi spectrum analysis and passive collection techniques to create machine learning models (DB-SCAN, SVM, LSTM) that can predict malicious traffic with a **70% recall** with a focus on features in WiFi and lower-level protocols
- Simulated active malicious traffic flows via Ostinato traffic generator and NetFlow
- Hardened device security with malicious agent detection and a dynamic firewall. Mitigated attack surface with network segmentation

Santa Clara University Engineering Department – Santa Clara, CA
Graduate Teaching Assistant Aug. 2024 — Current

- Conducted 3 weekly lab sections, guiding over **500 students** through topics in **advanced data structures and algorithms,** memory management, and **C programming** fundamentals
- Additionally taught over **100 students** on modern software engineering principles including **Agile**, Scrum, Unit and Performance testing, **DevOps** practices (CI/CD, Git, Docker), and **cloud-native** patterns (Monolith and Microservices)

Projects

Applied Auto-tuning on LoRA Hyperparameters Sept. 2023 – Jun. 2024
Technologies Used: llama.cpp, GPTune, QLoRA, Llama3, Google Cloud

- Optimized the fine-tuning of Llama3 8B, Qwen2 8B, and Mistral 7B large language models using quantization techniques (QLoRA)
- Implemented a pipeline utilizing GPTune, a Bayesian auto-tuning framework, to efficiently discover optimal QLoRA hyperparameters such as rank, alpha, and learning rate. Thus improving the trade-off between training time and model loss (performance and resource efficiency)
- Achieved a **30% reduction in training loss** compared to a random search baseline without increasing computational time

Online Code Sandbox Jan. 2024 – Mar. 2024
Technologies Used: Flask, Jinja2, Bootstrap, AWS EC2, Amazon S3 Bucket, Docker, Virtualization, Vagrant

- Enabled running C++ and C code online in a browser with multiple compiler versions
- Utilized serverless **AWS Lambda FaaS** for program execution, **Docker Containers** for pseudo-isolation, and **AWS S3 Buckets** for storage
- Supported **100+ concurrent users** via modularization and auto-scaling, cloud-based architecture

Distributed File System Jan. 2024 – Mar. 2024
Technologies Used: Flask, AWS EC2

- Constructed a distributed file system along with a client side library that enables users to seamlessly execute idempotent file system operations concurrently on a distributed network of storage devices as if it were one logical machine
- Incorporated support for **200+ concurrent users** with an average operation speed of **200ms**, scaling sublinearly with number of users
- Implemented fault-tolerance mechanisms, dynamic file distribution methods, and concurrency control via idempotent file operations

Eye Type, *Hack For Humanity* Nov. 2023
Technologies Used: Google Cloud, OpenCV, TensorFlow, Flask, Docker, Jinja2, Werkzeug, Numpy, Pandas, Matplotlib

- Led the design of an intelligent, on-screen keyboard for patients with Motor Neuron Disease or are unable to speak or type
- Enabled patients to type with their gaze up to **8 words per minute**, the 2023 state-of-the-art average
- Eliminated the need for specialized hardware typically costing **$300-$1,000**, requiring only a webcam or phone camera

Skills

Languages: Python, C, C++, Java, Typescript, SQL, HTML/CSS, Verilog, Bash/Zsh
Frameworks: Flask, Django, FastAPI, SQLAlchemy, Node.js, React.js, Pytorch, Tensorflow
Tools: Git, Linux (Arch), Jira, AWS, Docker, Kubernetes, QEMU, Jupyter