Isamu Arthur Poy

858-361-8596 | poyisamu@gmail.com | linkedin.com/in/isamu510 | github.com/isamumu

Education

University of California, San Diego Sep. 2021 – June 2023
Master of Science in Computer Engineering *CGPA: 3.7/4.0*

University of Toronto Sep. 2017 – May 2021
Bachelor of Applied Science in Computer Engineering with honours (Minors in A.I. and Robotics) *CGPA: 3.5/4.0*

Technical Skills

Languages: Python, C/C++, Java, Go, Rust, Typescript, Javascript, SQL, MATLAB, Swift
Tools/Libraries: PyTorch, AWS, Docker, Git, CUDA, Tensorflow, NumPy, gRPC, Linux, Jest, React

Work Experience

R&D Software Engineer April 2023 – Present
Qualcomm Institute / JASR Systems *San Diego, California*

- Developed a $6 million data security system for U.S. intelligence (IARPA), for RF anomaly detection in complex environments.
- Designed a Rust classifier which achieved greater than 80% accuracy across 8 different modulations on complex signals.
- Implemented inter-process communication in Rust using USRP radios for multi-threaded radio scheduling software with TCP.
- Deployed Docker containers for CI development via Github actions for testing infrastructure, and managed testing on the cloud.
- Contribution **published to MILCOM '23** for signal detection and localization of wireless activity using software defined radios.

Software Development Engineering Intern June 2022 – Sep. 2022
Amazon *San Diego, California*

- Spearheaded a data traffic management project using AWS services, to develop from scratch, with Buyer Risk Prevention team.
- Designed a Typescript back-end tool to enable traffic control on over 2000 transactions/second per order for real-time and historical data point collection, obtained from buyer receipts, by moving data between DynamoDB and SQS queues.
- Built a REST API with API Gateway for routing data to custom Lambda functions to start/stop/pauses data traffic between buyer receipts and AWS Step Functions for ML data processing.
- Enabled 50 to 100 developers to efficiently and instantly modify traffic patterns, and save at least 2-3 hours of wait-time per processed receipt where previously, engineers were unable to independently modify traffic without permission.

Engineering Intern May 2021 – Aug. 2021
Nagasaki University *Nagasaki, Japan*

- Utilized Pytorch DeepLabV3 network to detect sick trees at 95% accuracy, enabling automatic tree hazard inspections in Nagasaki.
- Designed and implemented an Swift iOS app, using core ML, for iPhone cameras to be used in real-time image segmentation.
- Research **published to Japan Society of Civil Engineers**, which drew over 5000 attendees including Japan Railways (JR) officials.

Summer Researcher June 2019 – Aug. 2019
Hong Kong University of Science and Technology *Hong Kong*

- Introduced a novel user interface for an appearance-based eye tracking algorithm to enable hands-free robotic control.
- Designed a threaded Python interface for an eye-gaze controlled (via webcam) robot simulated in several ROS environments.
- Improved navigation times up to 26% by mapping eye-gaze to coordinates on a monitor displaying the environment, allowing for directions to be inferred; this allowed users to issue motor commands by just gazing at a desired destination in their field of view.
- Coauthored a paper **published to EMBC '21**, the world's largest international conference for biomedical engineering research.

Projects

SurfStore | *Go, gRPC*

- Implemented a distributed, scalable networked file storage application in Golang based on the design of Dropbox system.
- Designed a gRPC interface between client programs and file storage server such that files can be modified and synced.
- Incorperated RAFT consensus algorithm to allow for fault-tolerant file storage in case of crashed servers.

Custom HTTP server | *Go*

- Implemented an HTTP web server from scratch which supported headers and handled concurrent, overlapping client requests.
- Enabled virtual hosting by allowing for hosting of multiple servers that are mapped to unique host names and directories.
- Incorporated a timeout mechanism to identify when to close the connection for invalid requests.

Fast Matrix Multiply | *C/C++, CUDA, AWS*

- Implemented C/C++ cache-aware, SIMD matrix multiply for Amazon EC2 Graviton ARM CPU to achieve at least 90% of OpenBLAS.
- Optimized with NVIDIA Turing GPU from AWS by implementing blocked matrix multiply based on CUTLASS using CUDA.
- Optimized GPU matrix multiply to achieve up to 20x more performance than the baseline, and reaching up to 3500 GFlops.

GIS Map Application | *C++, Git*

- Programmed a GIS similar to that of Google Maps from scratch with C++ and solved the travelling salesman problem.
- Software was capable of mapping various cities around the world such as Toronto, France, New York City, Tokyo, and Cairo.
- Implemented multithreaded A* search algorithm and multi-path Djikstra to determine optimal paths between two intersections.