Yuxi Luo

Email: yuxiluo2022@gmail.com | Phone: (917) 863-9829 | LinkedIn: linkedin.com/in/yuxiluo

Education

Carnegie Mellon University - School of Computer Science ***Pittsburgh, PA | 09/2020 - 12/2021***

Master of Computational Data Science | Systems Concentration *GPA: 4.00*

Relevant Coursework: Operating Systems | Compiler Design | Storage Systems | Advanced Cloud Computing | Databases

New York University - Tandon School of Engineering ***New York, NY | 09/2016 - 05/2020***

Bachelor of Science in Computer Science | Minors: Mathematics & Game Engineering *GPA: 3.93*

Relevant Coursework: Parallel and Distributed Systems | Unix System Programming | Programming Languages

Awards: Myron M.Rosenthal Merit Award 2019 - 2020 | Dean’s List 2016 - 2020

Technical Skills

Programming Languages: C++ | C | Python | Golang | OCaml | Java | Haskell | bash

Development Tools: Protocol Buffers | Bazel | Docker | Cassandra | Kibana | Promethehus | Grafana | NATS | git

Machine Learning Frameworks: PyTorch | Keras | NumPy | Pandas | Scikit-learn | NLTK | LaTeX

Work Experience

Software Engineer Intern - Backend Development | Robinhood ***Menlo Park, CA | 05/2021 - 08/2021***

- Restructured a money movement workflow that handles > 10k transfers per day into multiple daemon processes in Python and Django, which provides flexibility of handling transfers, improves stability of the workflow, and reduces on-call burdens
- Designed a failure-handling mechanism for daemon processes to keep track of failures and retry any of them if necessary

Software Engineer Intern - Backend Development | Genesis Global Trading ***New York, NY | 05/2020 - 08/2020***

- Improved database schema and updated symbol parsing algorithm to populate database with new market data in Golang, and equipped traders with comprehensive data to analyze spot and derivatives markets and perform algorithmic trading
- Developed a reliable data streamer of an exchange on a second-by-second basis for all provided markets (Bitcoin, LTC, etc)
- Designed and developed an LRU data structure to maintain a dynamic list of 100 tradable stock symbols for real-time trading

Software Engineer Intern - Full Stack Development | Goldman Sachs ***Hong Kong | 06/2019 - 08/2019***

- Developed a backend micro-service to track business approval status across an in-house web management system in Java
- Revamped and implemented a business-oriented workflow that communicates with backend services of the application

Undergraduate Independent Researcher | New York University Wireless Lab ***New York, NY | 02/2019 - 05/2019***

- Conducted a Multiple Sclerosis Lesion Segmentation research project supervised by Professor Yao Wang
- Built a 3D U-Net structure to process Magnetic Resonance Imaging (MRI) data in Python with Keras framework
- Implemented a sliding window approach to generate unique batches of training samples from only 15 available images
- Utilized Dice Score loss function to achieve a model with 71% cross-validation accuracy

Research Assistant | New York University Composite Materials and Mechanics Lab ***New York, NY | 06/2017 - 08/2017***

- Analyzed low-contrast images with histogram and used OpenCV techniques to upsample images
- Implemented a pixel-wise comparison algorithm between sample images and their corresponding reference images
- Published a paper in *Advanced Engineering Materials* titled “Embedding tracking codes in additive manufactured parts for product authentication” (https://doi.org/10.1002/adem.201800495)

Teaching Assistant | New York University Computer Science Department ***New York, NY | 09/2017 - 05/2020***

- Led Object-Oriented Programming in C++ from 2017 to 2019, and Unix System Programming in C in 2020
- Taught lab materials and assisted debugging for >170 students, held office hours, and graded assignments

Projects

C0 Compiler | Carnegie Mellon University - Compiler Design Course Group Project ***12/2021***

- Designed and implemented a C0 compiler from scratch in OCaml, which includes a number of optimizations to ensure that the generated assembly code is comparable to GCC -O1 performance, including Register Allocation, Single Static Assignment, Constant / Copy Propagation, Dead Code Elimination, Dataflow Analysis, and more

x86-32 Kernel | Carnegie Mellon University - Operating Systems Course Group Project ***04/2021***

- Designed and implemented an x86-32 kernel from scratch in C, which supports context switching, interrupt handling, task and thread management, scheduling, virtual memory, synchronization primitives, and device drivers

Honors and Awards

19th Place out of 65 Teams | 2019 ACM / ICPC Greater New York Contest *10/2019*