Maahir Gupta

maahirgupta.com | m63gupta@uwaterloo.ca | in/maahirgupta | github/MaahirG | 519 590 0150

Education

University of Waterloo, Software Engineering 2019 - 2024
Courses: Data Structures, Algorithms, Software Design, Operating Systems, Concurrency, Compilers, Linear Algebra
Activities: Self-driving car and EcoCar Student Teams, Varsity Baseball, Mentor, Career Fair Director

Skills

Languages: Python, C++, Golang
Tools: AWS, GCP, Kubernetes, Docker, gRPC, Linux, MongoDB, MERN Stack, PyData Stack, ROS, PyTorch
Fields: Backend, Distributed Systems, Cloud, ML Backend/Ops, Deep Learning, Computer Vision, Robotics

Work Experience

Software Engineering Intern
May - Aug 2022
Google - X, Everyday Robots Mountain View, USA

- Designed and built a distributed cloud system to stream and aggregate perception data from **100+ robots** into a single view, enabling realtime fleet wide insights and collaborative robotics in a production environment.
- Integrated unified view into cloud visualizer, leveraged pub-sub system to capture vision streams, setup cloud infrastructure, designed APIs, tracked metrics, and deployed in a Kubernetes environment.

Technologies: Golang, C++, Python, GCP, K8s, Microservices, gRPC, Docker, Distributed Systems, Robotics Programming, Git

Software Engineering Intern
Sept - Dec 2021
Wish - Search & Recommendations San Francisco, USA

- Designed, proposed, and built an expandable gRPC microservice paired with a client-side API to enable simple & centralized retrieval of user data; resulting in consistent low latency retrievals, minimized dev lead time and a cleaner codebase.
- Led development emphasizing 4 key design pillars: concurrency, expandability, modularity, and usability; resulting in project rollout with **~1B queries/day** from 3 critical business use cases (homepage, for you page, search).

Technologies: Golang, Python, gRPC, Kubernetes, Grafana, Flink ETL, Airflow, SQS, Docker, MongoDB, Memcached, Fluent HTTP, Git

Machine Learning Engineer Intern
Jan - Apr 2021
PerkinElmer, Inc. Waterloo, Canada

- Developed an end-to-end deep learning recommendation system to streamline analysis of time series data from medical devices.
- Leveraged Python, PyTorch, AWS, and Databricks MLFlow, to execute ML lifecycle tasks - research, data handling, model implementation (LSTM, Transformer), systems design, application building, metric tracking, and cloud deployment.
- Worked towards projected savings of **x . . hrs/week** for labs across multiple countries by reducing human intervention by **~98%**.

Technologies: Python, PyTorch, Linux, PyData Stack, Dash (Webapp), AWS, MLFlow, MongoDB, Git, Img Segment: CNN, U-Net
Link: [System Design Diagram]

Software Engineering Intern
May - Aug 2020
Swap Robotics Waterloo, Canada

- Built a robot-mounted image classification system to flag sidewalk defects, deployed by **6 municipalities** for over **x . . km.**
- Developed a modular C++ backend for depth camera interfacing, image processing and real-time CNN model use.
- Implemented robotic controls in C++ enabling use of in-house hardware, resulting in **78% cost savings** by eliminating controller.

Technologies: C++, Python, Linux, TensorRT, Embedded Systems, ROS, TensorFlow, Convolutional Neural Net (CNN), Git

University Teams & Projects

Software Engineer
Apr 2020 – Jan 2022
Self Driving Car Teams (WATonomous + EcoCar) Waterloo, Canada

- Facilitated driver scene understanding and simplified complex driving modules into human-like queries – built a real time model to logically structure traffic systems (signs, lights, obstacles) and their effects on the environment into a routing graph in C++.
- Developed vehicle tracking SLAM algorithm by fusing clustered camera & radar data with predicted position data at 86% accuracy.
- Co-authored research paper accepted into ICRA.

Technologies: C++, ROS, Rviz, Carla Vehicle Simulator, Linux, CMake, Docker, CAN, Nearest Neighbor, Kalman Filter, Git
Links: [Demo], [Publication], [Sensor Fusion Paper], [Vehicle Tracking System Diagram]

Autonomous Mini Car Project
Aug 2020 - May 2021

- Built an autonomous mini car along with, top-down driving visualization, remote video streaming & manual control, I/O interfacing.
- Implemented: perception pipeline that maps new obstacles (object detection) to a dynamically updating occupancy grid, which communicates with the path planning module (multiprocessing), resulting in realtime updates to the shortest path that is followed.

Technologies: Python, C++, Linux, TensorRT, OpenCV, ROS, Docker, PyData Stack, CMake, Jetson Nano Board, Git
Links: [Demo], [Short Explainer Video]