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, Distributed Systems, Operating Systems, Networking, Concurrency, Compilers, Controls, DBs
Activities: Varsity Baseball, Career Fair Director, Student Mentor, Self-driving and EcoCar Student Teams

Skills

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

Work Experience

Software Engineering Intern Sept - Dec 2023
Tesla – Autopilot Palo Alto, USA

- Incoming: scaling inference for machine learning development.

Software Engineering Intern Jan - May 2023
Tesla – Fleet Robotics Palo Alto, USA

- Designed and built a fleet management subsystem which streams sensor data and video from Tesla factories, performs congestion analysis, and supports real time re-routing and command of **1000s** of operational vehicles, assets, and robots.
- Built a high throughput data analytics and dashboarding pipeline using Grafana, Presto and Prometheus enabling distributed SQL queries.
- Packaged pipeline as a microservice and added to cloud deployments Kustomize workflow, enabling scalable data analysis per deployment.

Technologies: Golang, GraphQL, K8s, Kustomize, Prometheus, Grafana, Trino, Cortex, Microservices, Docker, JavaScript, React, Git

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 100+ robot 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.

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

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.

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

University Teams & Projects

Self Driving Car Teams (WATonomous + EcoCar) Apr 2020 – Jan 2022

- 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++. Paper **accepted** to ICRA.

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

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.

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