Steven (Shing) Kam

(949) 500-4113 | GitHub: Skam23 | shingk@andrew.cmu.edu | Linkedin: Stevenkam.com

Detail oriented and result driven computer science student at Carnegie Mellon University immersed in the field for more than half a decade. Passionate about the elegance of problem solving through computing and the use of machine learning in everyday life. I am seeking a software engineering role where I can efficiently contribute my skills as well as grow and develop my own abilities as an engineer.

Language: Python, Java, C/C++, SML, HTML/CSS, JavaScript, Julia, SQL, Swift | **Operating System:** MAC OS X; Linux
Tools: Jenkins; Docker; Kubernetes; Kafka; Spring Boot; Postman; Tensor Flow; Numpy; Git; LaTeX

EDUCATION

Carnegie Mellon University — **Pittsburgh, Pennsylvania**
Bachelor of Science, Computer Science — *August 2021 – Expected Dec. 2024*
Activities: Carnegie Mellon Autonomous Racing, CMU Blockchain group, NCAA Varsity Track and Field
Relevant Coursework: Artificial Intelligence; Computer Systems; Theoretical Computer Science; Probability Theory; Functional Programming; Imperative Computation; Matrix Algebra; Mathematical Foundations for Computer Science; Great Practical Ideas In CS

EXPERIENCE

Optum Technology — **Minneapolis, Minnesota**
Software Engineer Intern — *May 2022 – August 2022*

- Designed, prototyped, tested, and deployed an internal Java API that streamed secure provider demographic data attestation events to a central location
- Decreased turnaround time for the verification and suppression of demographic information by 80% by migrating and streaming 100,000+ rows and 80+ columns of data from separate databases into a centralized location, and ensuring 100% of provider demographic information was attested for within a 10 day period for 1.3 million healthcare professionals
- Technology: Java, SpringBoot, MySql, Jackson API, Apache POI, JDBC, Jenkins, Docker, Kubernetes, Kafka.

Carnegie Mellon Autonomous Racing — **Pittsburgh, Pennsylvania**
Autonomous Software Engineer — *August 2021 – Present*

- Working with 7 cross-functional teams to build autonomous driving capabilities for a fully functional Formula SAE race car and an accurate, Low-Latency Visual Perception for Autonomous Racing
- Utilized OpenCV and Python to develop a YOLOv5 Algorithm for detecting traffic cones used in the Formula Student Germany competition
- Achieved an average IOU threshold of 76%, with a less than 4% false positive and negative rate in YOLOv5 cone detection model utilizing the Pittsburgh Super Computer and over 50,000+ competition traffic cone images

The IPSF CubeSat Program — **Irvine, California**
Communication Team; Low-level laser software captain — *August 2018 – June 2021*

- One of 100 students selected from 6 public schools to build, test and successfully launch a solar-powered CubeSat into space
- Led a team of 5 students to integrate software for an optical laser downlink on Irvine02 to send vital data back to Earth from low earth orbit
- Worked with industry partners and laser suppliers to resolve software issues revolving the laser communication system.

PROJECT

MagicBin:

- Prototyped and developed a fully-functional trash detection system in a scrum team of 5 that blends machine learning and open-source software with the Raspberry Pi ecosystem
- Object detection model combines OpenCV and 10,000+ images to effectivity differentiate recyclable bottles from landfill waste

Dynamic Storage Allocator:

- Designed and developed a 64-bit general purpose dynamic storage allocator for C programs; with own versions of the malloc, free, realloc, and calloc functions.
- Operates using a struct-based explicit free list memory allocator with Performance Optimizations on design decisions with average Utilization 70% and throughput of 12054 Kops/sec.

Autonomous Donkey Car:

- Developed a real-time racing line detection software and implemented it on an autonomous RC racing car for competition.
- Multi-stage algorithm detects real time lane lines by using Histogram peak, Huff Lines, and Canny Edge detection algorithms in combination with the sliding window algorithm to output three real time functions for the controls team.