Xiangwan (Sunny) Sun

(480)-207-0591 | xs275@cornell.edu | linkedin.com/in/sunny-sun25/ | github.com/sunny525s

Education

Cornell University Ithaca, NY
Bachelor of Science, Computer Science *Aug. 2021 – May 2025*
Relevant Courses: Object-Oriented Programming & Data Structures, Functional Programming, Algorithms, Discrete Structures, Computer System Organization, Machine Learning, Computer Vision, Databases Systems, Data Science, Web Applications
Honors: John McMullen Dean's Scholar, Grace Hopper Scholarship (GHC '23), NCWIT Aspirations in Computing Winner
Activities: Corporate Officer @ Association of Computer Science Undergraduates (ACSU), Women in Computing at Cornell

Experience

Software Engineer Intern June 2023 – August 2023
Merck & Co., Inc. *Boston, MA*
- Managing the deployment of Machine Learning and Image processing workflows running in a cloud high-performance computing infrastructure using **Python** and **AWS** with pipelines to automate data flows and implementation architectures
- Implementing backend and UI of web dashboard to display real-time GPU and memory usage metrics and graphs using R data visualization libraries, **SQL**, and API calls to optimize the performance of Deep Learning workflows

Software Engineer Oct 2021 - Present
Cornell University Unmanned Air Systems *Ithaca, NY*
- Designed an unmanned aerial system capable of target detection and classification, image processing, and waypoint task navigation with a Ground Control Station supported with **Docker**, **React**, and **Typescript** frontend and **Python** backend
- Optimized Python algorithms up to **95% accuracy** to automatically localize targets, avoid obstacles, and classify attributes such as color, shape, and alphanumeric with **ResNet** and **CNN machine learning** methods from plane captured images
- Implemented a **ReactJS** component that outputs the coordinates of target shape corners to create masks for training a machine learning shape segmenter pipeline and tested the functionality in **Node.js** with Jasmine
- Developed a recruitment portal frontend with React and created markups on **Figma** to boost application review efficiency

Machine Learning Intern May 2022 - August 2022
Rensselaer Polytechnic Institute *Albany, NY*
- Simulated real-world voting behavior and decision making using multiagent reinforcement learning algorithms, improving the efficiency of **Python** algorithms by over **30%** and increased understanding of ranking axioms and scenario sets
- Developed a custom learning framework with **Stable Baselines** and **OpenAI gym class** to perform accurate predictive analysis on the behaviors of voters by optimizing the Pareto frontier to model the informational complexity of various environments, agent payoffs, and improved metrics to benefit all stakeholders involved in a voting scenario

Software Engineer Intern June 2021 - August 2021
Brookhaven National Laboratory *Upton, NY*
- Collaborated with industry professionals to apply Fourier reconstruction and Machine Learning algorithms to optimize collections of diffraction patterns and improved Ptychographic imaging techniques from synchrotron X-ray data over 85%
- Specialized in developing analysis methods for reconstructing 1000 sets of data into images and optimized data-processing pipelines, image quality, and processing speeds through simulations with **Python** (TensorFlow, Numpy, Matplotlib, SciPy)
- Presented major findings with a poster presentation at a research symposium with over 50 Brookhaven scientists

Projects

Predicting whether a Dog's Legs are Lame | *Python, PyTorch, XGBoost, scikit-learn, numpy*
- Developed an XGBoost classifier model to predict if a dog's leg is limp based on numerical results of the Force Plate Gait Analysis for a research project at the Cornell Vet School
- Built 4 python classifiers to predict the probability for each leg and improved model accuracy up to 87% through tuning hyperparameters and employing k-fold Cross-Validation

Chinese Caml Checkers | *OCaml, Git*
- Developed an interactive multiplayer terminal-based version of the game Chinese Checkers that supports turn-based gameplay using OCaml and JSON files, focusing on functional programming principles and test-driven development
- Implemented features to integrate different game mode difficulties, marble hopping movements, command line flexibility, colored terminal using ANSITerminal module, and followed agile software practices by using Git for version control

Technical Skills

Languages: Python, Java, TypeScript/JavaScript, OCaml, SQL, HTML/CSS, C, R
Tools: Git, Docker, LaTeX, AWS
Libraries/Frameworks: TensorFlow, PyTorch, NumPy, Matplotlib, SciPy, pandas, React, Node.js