Andrew Ngo

ango@udel.edu · (302) 357-5499 · github.com/andrewango

Education

B.S. Computer Science and Applied Mathematics, University of Delaware
GPA: 3.9 / 4.0
Graduate-Level Coursework: Machine Learning, AI, Mathematical Data Science, Linear Algebra, Stochastic Processes

Skills

- Languages: Python, Java, SQL, TypeScript, JavaScript, HTML/CSS, C/C++, MATLAB, ARM Assembly
- Libraries: React, Jest, NodeJS, Express.js, PyTorch, TensorFlow, OpenCV, NumPy, Pandas, Matplotlib, Scikit-learn
- Frameworks / Tools: Oracle, MySQL, Git, Docker, Agile, RESTful API, FastAPI, Linux, Unix, VIM, Bash, GDB, ETL

Experience

Research Engineer Intern, University of Delaware – Newark, DE — Sept. 2024 - May 2025

- Accelerated counterexample search in graph theory by 10x by building an AI reinforcement learning agent and novel graph embedding system in PyTorch for scalable experimentation, modularizing code into reusable components with documentation.
- Authored a research paper detailing implementation and experimental results, now used as a foundation for ongoing work.

ML/Software Engineer Intern, NASA – Mountain View, CA — June 2024 - Aug. 2024

- Built an ETL pipeline to combine overlapping images of the sun into a dataset of full-scale images for scientific analysis.
- Enhanced analysis of the sun by developing Python software that visualizes 2D solar images into 3D using OpenCV/AstroPy.
- Deployed the software for 3K+ researchers using Docker containerization.
- Improved research pipeline speed by 3x by training machine learning models in TensorFlow/Python to automate the alignment of 1K+ solar images, ensuring accurate measurements during scientific analysis.

Research Engineer Intern, Sensify Lab – Newark, DE — Sept. 2023 - May 2024

- Improved depression/anxiety prediction accuracy from 70% to 75% using feature engineering in Pandas and PyTorch.
- Achieved 80% sentiment classification accuracy on 50K+ mental health app reviews by training and validating a PyTorch NLP model bootstrapped from 3K manually labeled samples in JSON format, supporting a poster to CHASE '24.
- Reached 75% accuracy on rating prediction of 200K+ app reviews by engineering scalable neural network pipelines in PyTorch.
- Supported a publication to ICWSM '25 by building Python-based tooling and Amazon Mechanical Turk surveys to compare LLM-generated outputs to human review ratings.

ML/Software Engineer Intern, NASA – Mountain View, CA — June 2023 - Aug. 2023

- Built ETL pipelines for solar data downloads and visualizations by developing a data portal using Python and JavaScript with the software engineering team, supporting space researchers and an astrophysics poster at the AGU '23 conference.
- Increased query speed by 30% by optimizing database filters for category and time-based retrieval using a MySQL backend.
- Automated solar feature detection by building an image segmentation pipeline in OpenCV, achieving 75% accuracy on 4,680+ satellite images using machine learning models in TensorFlow to improve robustness.
- Reduced image processing latency by 150% using multithreaded data loading.

Projects

Restaurant Menu – React, Boostrap, TypeScript, HTML/CSS, Jest, Git

- Built a React/TypeScript website with session-based role access and state management, integrating CI using GitHub Actions.
- Ensured unit test coverage of at least 90% using Jest by leading code reviews with Git for Agile team of 5 developers.

Cora, Your Anatomy Assistant – JavaScript, FastAPI, Three.js, OpenAI Whisper, Figma

- Built a full-stack web app with a JavaScript/HTML frontend and FastAPI backend serving RESTful endpoints.
- Integrated Whisper AI and IBM Watson APIs to enable speech-to-text and NLP functionalities.

Employment Database System – PL/SQL, Oracle

- Created a schema with tables for employees, departments, salaries, and job titles.
- Enforced business rules by implementing triggers, functions, and packages.

2-Player Chess – Java, IntelliJ, Gradle

- Designed a custom two-player chess-inspired game in Java, featuring unique pieces and rules, player input handling, game logic enforcement, and test cases to verify functionality.