Alp Buz

alpbuz03@outlook.com | (631) 944-2401 | linkedin.com/in/alpbuz | github.com/AlpBuz

Education

University at Buffalo – B.S. in Computer Science, GPA: 3.5 (2021–2025)
Relevant Courses: Data Structures, Systems Programming, Algorithms, AI, Data Mining, Web Apps, Software Engineering, Data Models and Query Languages
Stony Brook University – M.S. in Computer Science, AI/ML Focus (Expected Start: Fall 2025)

Projects

Self-Order Kiosk Web App

React, PHP, MySQL, TailwindCSS, Git | Spring 2024 | Team of 4

- Developed a full stack restaurant self-ordering web app with real-time financial dashboards.
- Built secure login and signup features with session management and input validation to prevent SQL injection.
- Designed front-end components such as menu, checkout, orders, rewards using React and Tailwind CSS.
- Created backend PHP APIs to manage menu items, transactions, and profit tracking with MySQL integration.
- Implemented a Profits dashboard showing weekly revenue trends and top-selling menu items.

RoutineFlex Social Media Web App

Flask, HTML/CSS/JS, WebSocket's, Docker, Digital Ocean | Fall 2024 | Team of 4 | routineflex.social | GitHub Repo

- Built and deployed a social fitness platform where users share and discuss workout routines.
- Developed live chat functionality using WebSocket's for real-time user communication between all users
- Containerized the app with Docker and deployed on Digital Ocean with an SSL-secured domain.
- Implemented user profiles, routine cards, and a Suggested Connections system for user engagement.

Steganography

C, Makefile, Criterion Testing Framework | Spring 2025 | Team of 3

- Improved and extended an open-source steganography tool to support image encoding and text embedding.
- Fixed critical bugs and wrote comprehensive test cases, achieving 100% code coverage with Criterion.
- Added support for US-ASCII and UTF-8 encoding and expanded image format compatibility to JPEG and PNG.
- Automated the testing, coverage, and build process using a custom Makefile.

Plant Species Classifier using CNN

Python, TensorFlow, Kaggle *| July 2025 | Solo Project | GitHub Repo*

- Built and trained a Convolutional Neural Network on 1,000+ plant images across 100+ species, using TensorFlow.
- Handled data preprocessing by Normalizing image pixel values from 0–255 to 0–1, improving CNN training stability and speed.
- Built and trained a CNN using convolution, pooling, and dense layers with Adam optimizer.
- trained and validation performance over 10 epochs, visualized model via loss/accuracy plots using Matplotlib. Which allowed me to detect my model was overfitting.
- Evaluated model with a classification report. Found high accuracy in well-represented classes (Cassava: F1 = 0.94) and low scores in rare classes (<0.10 F1), revealing more data is needed.

Skills

Programming languages: Python, C, C++, PHP, Golang, JavaScript, Java
AI/ML: PyTorch, TensorFlow, Scikit-learn
Web Development: HTML, CSS, React, TailwindCSS
Databases: MySQL, MongoDB, PostgreSQL
Operating Systems: Windows, Linux (Ubuntu)
Tools: Git, Visual Studio Code, IntelliJ IDEA, Docker