Andrew Vattuone

San Mateo, CA 94402 | (650) 445-2787 | andrewvattuone@yahoo.com |
www.linkedin.com/in/andrewvattuone https://github.com/andrewvattuone

EDUCATION

Bachelor of Science in Computer Science and Engineering Expected Graduation: 06/2027
Santa Clara University, *Santa Clara, CA*

GPA: 4.0

- **Relevant Coursework Completed:** Machine Learning and Data Mining; Object-Oriented Programming and Advanced Data Structures; Computer Networks; Operating Systems; Theory of Algorithms; Introduction to Information Storage; Abstract Data Types and Data Structures; Introduction to Embedded Systems; Linear Algebra; Probability and Statistics; Discrete Mathematics; Differential Equations; Logic Design; Electric Circuits I
- **Relevant Coursework In Progress:** Electric Circuits II
- **Planned Fall 2025 Coursework:** Computer Architecture, Principles of Design and Implementation of Programming Languages, Digital Signal Processing

PROJECTS

Machine Learning Histopathological Image Classification of Gastric Cancer | *Python, ResNet50, VGG-16, TensorFlow, PyTorch, scikit-learn, XGBoost, Support Vector Machine, Truncated SVD, Pandas*

- Created a machine learning pipeline to identify cancerous vs non-cancerous gastric tissue and the tissue composition of each image based on 8 different types of gastric tissue. Used CNN-based models for feature extraction and traditional ML models for classification. Achieved an accuracy of 95% for cancerous vs non-cancerous tissue and 79.24% accuracy for tissue composition classification. Created a report summarizing findings and methods used, as well as a website that allows for images to be uploaded for cancerous vs non-cancerous determination and tissue composition classification. Created for Machine Learning and Data Mining class.

Smartware | *Python, Deepgram, ChromaDB, LangChain, LLaMA*

- AI conversation recorder/summarizer and general assistant. Can record conversations and stop when keyword "stop conversation" is heard, can summarize conversations, can remember conversations and find requested details in them, can set reminders, and can be asked about general facts or advice. Built using Python. Used Deepgram for audio recording & transcription, ChromaDB vector database to store transcriptions, LangChain to prompt LLaMA, and LLaMA to find conversations and create a response based on user request. Created for Hack for Humanity 2025.

Reducing Latency in AI Workloads: File System and Storage Optimization Strategies | Academic Research

- Research report completed for Introduction to Information Storage class. Investigated which file systems and object stores would be best for reducing latency in AI workloads that involve large volumes of small, random reads, such as those found in LLM inference. Research was conducted through a literature review of candidate file systems and object store features. Findings were that although file systems have some useful features for reducing latency in AI workloads, object stores have more helpful features overall for reducing latency for these workloads.

Eco Car Finder | *HTML, CSS, JavaScript*

- Website built for the AWS x INRIX Hackathon at SCU. User inputs information about what kind of car they want, including price, minimum miles per gallon, fuel type, color, and brand. Based on inputted information, program searches for appropriate cars stored in a CSV file containing information about cars currently for sale (downloaded from Kaggle). Top 10 cars that match all of the user's inputs are displayed. Cars with higher minimum MPG and electric vehicles are put to top of list to promote purchase of eco-friendly cars. Built with HTML, CSS, and JavaScript.

Window Maze | *Arduino, C++*

- Maze game built with photoresistors and controlled by an Arduino Uno. Maze is randomly generated and stored in a 4x3 array, with movement controlled through a 2x3 grid of photoresistors, acting as a moving “window” showing the user’s current row and the row ahead. Each photoresistor represents one maze cell, and player makes a move by tapping a photoresistor. Player loses if they make 3 incorrect moves or take too long.

Othello Game | *C*

- Othello game played using the command line. User plays against AI, which determines the best possible move by calculating what will give it the highest score 3 moves later. Total score determined by board location of each tile as well as opportunities to capture more tiles. AI assumes the user makes best possible move in each scenario.

WORK EXPERIENCE

Undergraduate Student Researcher 06/2025-Present
Santa Clara University, *Santa Clara, CA*

- Currently creating synthetic datasets for ML usage using IsaacSim with Omniverse
- Datasets include:
    - Digital twin of videos of car accidents filmed at Toyota factory in Japan
    - Typical warehouse environment of both humans and robots
    - Typical manufacturing environment

Design Intern Every June-September from 2022-Present
ISA Corporation, *Union City, CA*

- Updated and maintained company website with WordPress to improve site Search Engine Optimization and keep customers informed on the latest products
- Created solar array cost calculator tool for website using HTML, CSS, and JavaScript
- Created designs for solar installations using AutoCAD that aided in generating sales
- Monitored and organized inventory to streamline order fulfillment
- Collaborated on a 4-person team to design, develop, and build solar panel clamping device to enhance efficiency of solar equipment installation

ACTIVITIES

Member of Alpha Sigma Nu Jesuit Honors Society 01/2025-Present
Member of Competitive Programming Club 09/2024-Present
Member of Association for Computing Machinery at SCU 02/2024-Present
Santa Clara University Club Swim Team 09/2023-Present
Swim Across America Charity Swimmer 05/2022-Present
Flight Teen Volunteer at Hiller Aviation Museum 05/2022-09/2023

RELEVANT SKILLS

Languages: Python, C, C++, HTML, JavaScript, CSS, Java, Verilog, Assembly
Applications: Excel, Visual Studio Code, AutoCAD, TinkerCAD, Arduino