

EDUCATION

University of California, Irvine

M.S. in Computer Engineering, GPA: 3.40/4.00

Irvine, CA

Sept. 2022 - Present

University of California, Santa Cruz

B.S. in Computer Engineering, Minor in Computer Science, GPA: 3.31/4.00

Santa Cruz, CA

Sept. 2019 - Jun. 2022

SKILLS

Programming languages

Python, Java, C, C++, C#, SQL, HTML, JavaScript, CSS, SystemC, MIPS ASM, Verilog

Technical Tools

Linux, Git, NodeJS, React, Jupyter, PyTorch, TensorFlow, Scikit-learn, Pandas, Bash, WebGL

Software

VS Code, CCStudio, Bitvise SSH, FileZilla, MATLAB, Wireshark, QEMU, VirtualBox, LaTeX

EXPERIENCE

Winservo Electrical Technology Co., Ltd.**Xiamen, Fujian, China***Embedded Software Development Engineer Intern*

Jun. 2023 - Sept. 2023

- Implemented advanced PID control algorithms to enhance servo motor tension control, replacing outdated linear algorithms
- Utilized CCStudio for debugging, and optimizing the existing codebase, resulting in improved performance and reliability
- Collaborated with electrical and mechanical engineers to adjust the control algorithm, ensuring the change fit with the system
- Improved the real-time response of the servo motors, resulting in increased production efficiency and product quality
- Documented the development process and provided reports to assist in knowledge transfer within the development group

CE Grow Force Technology Co., Ltd.**Remote***Software Development Engineer Intern*

Jul. 2022 - Sept. 2022

- Designed a new platform facilitating documents and installation packages, aimed to improve user resource accessibility
- Implemented data uploading strategies by integrating the Alibaba cloud storage solution to guarantee easy access to resources
- Applied elastic search algorithms to allow users to search through the database for desired documentations and packages
- Developed user-friendly download pages in HTML and JavaScript offering in-depth information on downloadable content
- Enhanced page layout by incorporated a navigation bar to provide file type categorization with version selection tables

University of California, Santa Cruz**Santa Cruz, CA***Undergraduate Student Researcher*

Jul. 2021 - Jan. 2022

- Developed a Vision-RFID fusion system to achieve real-time identification of individuals' physical IDs within the camera field of view, and implemented a tracking algorithms to monitor targets and capture their movement patterns in video frames
- Implemented a C# program with Low-level Reader Protocol and Kinect to deploy RFID reader and data collection system
- Created a Python-based RNN model using a LSTM network for binary time series classification to process physical data
- Utilized PyTorch to train and assess a deep learning model with the Siamese OSNet for accurate person detection and tracking
- Performed data collection and analysis to validate the RFID tag detection and movement tracking capabilities of the model

PROJECTS

Long Marine Lab Stranding Map

- Collaborated in the implementation of a database visualization map project for geospatial data on stranded marine life cases
- Developed front-end layouts using JavaScript with React, leveraging Mapbox GL for mapping functionalities, and utilized Node.js on the backend for efficient data management and utilized Firebase and Algolia for database indexing and retrieval
- Contributed to the existing codebase by resolving bugs and optimizing database management through peer code review
- Collaborated with team members by Jira for efficient task management and GitHub for code collaboration and version control

The BreakAlert

- Developed a productivity-enhancing Google Chrome extension to empower users through a range of useful add-on tools
- Utilized HTML and JavaScript skills to architect fully functional extension pages that integrated a scheduler, calendar GUI, tab shortcut, to-do list, alarm, and website blocker, fostering heightened productivity and concentration
- Crafted a HTML landing page that showcased the extension's capabilities, and introducing users to its features and benefits
- Led a team of four developers to adopt Scrum methodologies, facilitating efficient collaboration and on-time project delivery

First-Person 3D World Exploration

- Developed a 3D scene application using Three.js, incorporating several 3D primary shapes, including cubes, spheres, and cylinders, with one textured and animated 3D object to create a visually engaging and dynamic world on website
- Implemented light sources including Ambient Light, Directional Light, and Point Light, to enhance the visual appeal
- Integrated a textured skybox using a cubemap, creating a captivating background that enhances the immersive experience
- Configured the camera with incorporated Orbit Controls to allow users to navigate the scene with mouse interactions
- Incorporated advanced features, including Fog for heightened depth perception, Render to Texture for dynamic rendering, and Shadows for enhanced realism and visual impact, successfully achieving an elevated level of visual effect

Bitcoin Price Prediction

- Developed an autoregressive RNN model for Bitcoin price prediction using Python by utilized historical Bitcoin price data
- Imported the data using Pandas and processed it with Numpy to a chronological distribution training and test sets
- Explored and tested data using various RNN architectures, including LSTM and GRU, both unidirectional and bidirectional
- Evaluated models using MSE to plotted training, validation loss, and select the best-performing model based on evaluation
- Tuned model for accurate predictions and illustrated results by comparing predicted prices to actual prices through plots

Gesture Recognition and Classification

- Built a custom CNN model for recognizing ASL hand gestures images corresponding to the English alphabet
- Resized and preprocessed images to 200x200 pixels, adjusted pixel values, and meticulously categorized target classes
- Adjusted the convolutional and dense layers, ReLU activation, batch normalization, and dropout to mitigate overfitting
- Trained the model using Categorical Cross-Entropy loss, Adam optimizer, and implemented learning rate schedules
- Performed hyperparameter tuning with a primary focus on optimizing learning rate, architecture depth, and dropout rate, followed by model selection based on validation performance and subsequent test accuracy computation

Visualization Comparance of GAE and DANMF

- Visualized the community detection in networks and created runnable projects to enhance understanding of GAE and DANMF
- Used Python to read and process datasets like CiteSeer, Cora, and PubMed to validate and transform them for model utilization
- Utilized TensorFlow to implement GAE models and Scikit-learn to implement DANMF models, extensively experimenting with various encoder sizes and layer configurations to evaluate the model structure on performance across multiple datasets
- Generated multiple tables to assess model robustness by introducing noise to the data and analyze the difference of performance
- Re-implemented the original Python codes on Jupyter for the platform, enhanced functionality through modifications to package sources, dataset sources, and relevant codes, resulting in a streamlined user experience with one-click execution capability

C2S Proxy Server

- Designed a proxy server, enhancing its functionality to manage HTTP requests from clients through seamless conversion into HTTPS requests and implemented an access control list-based request filtering system to optimize network performance
- Implemented input validation for the port number and configuration file paths, enhancing the server's robustness and reliability
- Set up and managed multiple threads to handle concurrent client connections, ensuring efficient operation and responsiveness
- Designed a resilient signal handler for command handling and site list updates, while also developing a secure data retrieval and transmission function using SSL encryption for client-server communication to ensure data integrity
- Created error handling functions to provide error messages in response to unexpected issues, improving user experience

Thread Scheduling and Priority Management

- Implemented a priority-based ready queue system for thread scheduling and prioritization in a multi-threaded environment
- Developed a priority donation mechanism to effectively address situations in which high-priority threads required access to resources held by lower-priority threads, ensuring seamless priority transfers and the prevention of resource contention issues
- Designed and integrated advanced priority management functionalities, encompassing multi-tier, nested, and linked priority contributions, ensuring meticulous documentation of donation data to optimize thread scheduling efficiency
- Enhanced thread system responsiveness and efficiency by prioritizing high-priority threads for improved task execution

System Calls and Process Management

- Designed and implemented a comprehensive system call handler, featuring an array of essential functions for system calls, including create, open, read, filesize, wait, close, and exec, aimed at improving the system's overall functionality and efficiency
- Developed a robust stack management system to intelligently parse runtime parameters and efficiently store them, guaranteeing precise handling of critical base addresses, string lengths, word alignment, and various other essential parameters
- Developed a stack-based approach to manage the storage of parameters in order to ensure the execution of system calls
- Resolved potential parent-child process order issues by implementing a thread-sync mechanism, preventing test case failures
- Implemented data structures and lists for the system to track file openings, manage file descriptors, and support file operations

Canny Edge Detector

- Developed an efficient SLDL model with a focus on utilizing the Canny Edge Detector, SystemC compiler integration, and precise parameter management, while assessing the computational load for DUT components as a percentage of the total load
- Used the gprof tool for SystemC to identify the computational complexity and calculate the relative loads of components
- Implemented fixed-point arithmetic optimizations for a specific component, converting floating-point variables to integers and adjusting timing based on the trade-off between execution speed and image quality assessed with ImageDiff tool
- Conducted performance testing on Raspberry Pi 3 and 4 to find the improvement and adjusted model timing accordingly

LANGUAGES

Chinese - Native, **English** - Proficient, **Japanese** - Basic