Jasher Grunau

Fork Union, VA | P: +1 (434) 825-6769 | email: jasherg7@vt.edu | website: https://jasherg7.github.io/

OBJECTIVE

Software Engineer who is versatile, but prefers data science, robotics, firmware design, machine learning, communication systems, and/or image processing. Can do Web Development, but prefer to not make it a career focus.

EDUCATION

B.S. IN COMPUTER ENGINEERING   Blacksburg, VA 24061
Virginia Polytechnic Institute and State University   August 2019

SKILLS

Programming Languages: Python, C, C++, Javascript, Java, Matlab, Bash, Verilog, MIPS assembly
Markup Languages: HTML, JSON, XML, LaTeX
Libraries: Tensorflow, Anaconda Suite, Matplotlib, OpenCV, Qt, ProcessingJS
Databases: MySQL, MongoDB
Operating Systems: Cloud computers (usually Ubuntu), FreeRTOS, Unix based systems

WORK EXPERIENCE

AALTA   Seattle, WA (remote)
Software Engineer   Oct 2021 – Present
Work in Artificial Intelligence, Machine Learning, Web Development, Digital Image Processing, Predictive Analytics, Text Mining, etc

ROBOGRINDER (International Robot team for *RoboMaster*)   Blacksburg, VA
Software Engineer Team Member   2017 – 2019
Work in Embedded Systems, Computer Vision, Artificial Intelligence, Robotics, etc

BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING   Blacksburg, VA
Electrical Engineer, part time   2015 – 2016
Work in Signal Processing, Embedded Systems, Communications, etc

PROJECTS

MARITIME AERIAL OBJECT DETECTION AI   Project for Aalta
Proof of concept developed for an open project for the Navy. It was a computer vision project to detect ships in raw video aerial footage

- Tried a simple Neural Network and then a deep neural network but the machine was not effectively learning object detection
- Constructed a Convolutional Neural Network, and performance still didn't improve
- Mapped the convolutions and poolings visually, and saw the objects were too small to be recognized in the noise of the sea
- Re-cropped dataset and immediately fixed performance and achieved 89% accuracy with the Cross-Validation Set
- Implemented sliding-window object detection
- Parsed, formatted, and labeled raw video footage into a dataset
- Built a ResNet50 Convolutional Neural Network architecture because it measurably performs well with satellite imagery
- Froze and transferred training weights from a model trained off of the COCO 2017 dataset
- Reached a Loss less than 0.002 with very original maritime aerial footage then concluded the Proof of Concept

IONIC LIQUID ANALYSIS AI Project for Aalta

We were contracted by a university's chemistry department to assist with a machine learning research project. Had a couple extra interns hired to work with.

- Extracted image data manually and algorithmically from Chemistry PDF Textbook
- Cleaned and organized the collected data
- Transformed the image data with various image processing techniques to make the model run more accurately and quickly
- Implemented a Regression model in a Multi-layer Neural network to get a numerical output
- Expanded the image dataset by using image augmentation
- Later implementations of CNNs

SAM.GOV NATURAL LANGUAGE PROCESSING AI Project for Aalta

First Project I worked on for Aalta. The project ended after we discovered we terminally lacked legal access to specific data to make our intelligence work. I worked under Sr. Data Scientist and SME until its conclusion. I collected, cleaned, and prepared data to flow well into a NLP model of choice.

- Programmed a cloud computer to continuously extract data from Sam.gov with a multithreaded python script
- Customized the program to continuously gather 100M+ data points from an API as stably and quickly as possible
- Designed my own tokenizer (honestly that was because I didn't know better)
- Web-scraped some pages with Python script and web crawler
- Parsed and cleaned data into a MongoDB database

UNORTHODOX SEQUENTIAL MODEL AI Personal Project

Original algorithmic development of an AI model that takes its functional roots from engineering systems controllers I learned back in college, rather than the field of mathematical regression models. It works very differently than other models, and I don't see anyone else taking the same approach as I am with this experiment. For better or worse, I want to see how my concept will develop and perform compared to mainstream approaches

- Automated data collection from cloud computers
- Developed personal server and coordinated it with the cloud computer
- Implemented SQL database management system
- Drafted and tested different system mathematical models to work in discrete systems
- Experimented, tested, and refined algorithmic performance on low level
- Compare performance visually with mainstream sequential models (in progress)

BLOCKCHAIN SMART CONTRACTS FOR ITEM INVENTORY Project for Aalta

Proof of Concept to draft an inventory control system on a niche permissioned blockchain network with unique smart contracts and token system.

- Develop ERC-20 token system logic deployed in a simulated Ethereum blockchain system with permissioned accessibility for extra security
- Make token design as the drafted permission settings to verify users into the network for added security
- Deploy nodes using docker images and manage the clusters using Kubernetes
- Design network and software architecture of a private blockchain implementing the Hyperledger Fabric Framework

COMPUTER VISION FOR ROBOT NAVIGATION AND COMBAT RoboGrinder

Developed an image processing software in an International robot team called RoboGrinder primarily focused on a Chinese competition called Robomasters hosted by DJI in Shenzhen, China. I worked on the software subteam, where we focused on Computer Vision. Specifically we focused on image processing for our robots' guns and turrets to automatically target enemies without the use of a pilot.

- Design the image processing feed of our robots to auto-detect and target our enemy robots with our turret gun.
- Use the OpenCV library to apply various image filtering to our feed in our Embedded Environment
- Implement a Flood-Fill algorithm for object detection of that filtered image feed
- Train our turrets to target special locations in a separate competition mini-game (hosted in Quebec). Trained with Tensorflow library and Machine Learning techniques
- Work on team with a massive language barrier
- Document trips and handle technical writing to sponsors

MICROCONTROLLER INTERFACE FOR ROVER CONTROLS   Capstone Project & RoboGrinder

Capstone project for programming, design, and microcontroller interfacing for 3 different rovers that incorporates engineering standards. As part of an engineering design team, I needed to design, implement, and debug multi-threaded software that operates under real-time constraints on PIC32 embedded computer systems.

- Implement a multithreaded environment using Harmony with FreeRTOS as the system OS. Made a software system using multiple tasks, FICO queues from FreeRTOS, and ISRs as the skeleton of the system.
- Create a debug system within the hardware using the GPIO outputs to send timed codes to a Logic analyzer that works within a multi-threaded environment, while having a separate system that outputs debug codes as UART messages
- Implement Unit Testing
- Use UART, I2C, SPI, and CAN to communicate with all the various modules integrated with the PIC32 board i.e. robotic arms, sensors, LEDs, motors, etc.
- Use interrupt handlers integrated with hardware timers to read ADC units

P.I.D. STABILITY CONTROLLER   Capstone Project

In the feedback loop between my written software and the motor system in my rover robot, there are inconsistencies with response and a significant delay in the feedback. Correcting error in the environment in an efficient manner can be really difficult without leading to system instability. This problem was solved with a formal systems engineering method by employing the Proportional Integral Derivative controller.

- Tune my rover's motor systems to consistently reach its coordinates with no more than +/- 1.0 cm error, every 30.0 cm of travel.
- Develop an embedded system on a PIC32 board with C programming and FreeRTOS
- Design low-level TCP communication to communicate with other Team members' PICs and my designed server system.
- Change P, I, and D constants in the embedded system through server communication for efficient debugging and testing.
- Send motor error real time to the server, and plot the error over time to see how to monitor the stability of the system.
- Use the stability plots to tune the PID control system via the PID constants on the server.

REVERSE PROXY SERVER FOR ROVERS   Capstone Project

A server system must be able to connect with 5 clients and communicate via a customized network protocol in order to read and write information in a database all on a Raspberry Pi.

- Design network communication protocol unique to our embedded systems
- Create Reverse proxy server to establish connection with clients, multi thread communications, parse network packets, and process information
- Implement Flask server and communicate with the reverse proxy server via HTTP requests
- Make a MongoDB API, and use it as a database within the Flask Server

2D-GAME ENGINE   Senior Design Project

We must design a full functioning game using only Processing.JS's simple graphic library within the frameworks of HTML, CSS, and Javascript. To have a functioning game, I needed to build and design a game engine. Its design was meticulous and difficult. It's performance is atrocious because our class's development platform doesn't allow us to store custom sprites in cache or in memory. Meaning each sprite is manually being regenerated many times a second. Just see it on my website:
https://jasherg7.github.io/

DRONE COMMUNICATIONS IN EMBEDDED SYSTEMS   Bradley Dep of ECE

The efficient deployment of multiple unmanned aerial vehicles (UAVs) with directional antennas acting as wireless base stations that provide coverage for ground users.

- Implement Software Defined Radio (SDR) integration with microcontrollers on UAV for research department
- Redesign 802.11ac communication protocol onto SDR with very low level signal processing design
- Implement Dr. Saad's EE theoretical communication protocols onto the SDR
- Integrate Unix OS with the SDR allows the research team to automate the SDR's behavior
- Mobile device testing integration with SDR protocols to extend wifi coverage as a massive network of hotspots

MIPS ASSEMBLY SIMULATOR   Class Design Project

Large individual project to simulate the MIPS Assembly compiler and its code execution.
- Code in C++, while using Git and CMAKE for the project management
- Create robust unit testing for 100% code coverage
- Write language parser for MIPS assembly language
- Use object-oriented design for command executions previously parsed
- Manually allocate memory in C++ and fix any measured memory leakage
- Design a UI using the Qt graphics library for an easier access for clients
- Multi-thread environment and measure increased program execution speed

EMBEDDED SYSTEM VIDEO GAME DESIGN   Class Design Project

Class Design project where we programmed a microcontroller with a self designed video game on a system state machine to interact with various hardware modules with a wide variety of communication protocols.
- Design LED behavior for debugging notifications
- Integrate a MAXSONAR sensor via UART into system state machine
- SPI and Analog to digital Converter for Joystick integrated with hardware timers
- Interrupts are used to access hardware timers and A2D converters
- I2C communication to access accelerometer
- Programmed and designed a videogame to be played on the OLED with as a UART output for each frame and with customized pixel graphics

ACHIEVEMENTS

3RD PRIZE for team RoboGrinder	**ICRA 2019 International A.I. Challenge**
2ND PRIZE for team RoboGrinder	**Robomaster 2018 Final Tournament**
1ST PRIZE for team RoboGrinder	**Robomaster 2018 International Regional Competition**
3RD PRIZE for team RoboGrinder	**Robomaster 2017**
EAGLE SCOUT	