Callum Hepworth

403-966-1937 | callumahepworth@gmail.com | github.com/calhep | www.callumhepworth.com

Education

University of British Columbia Vancouver, BC, Canada
Bachelor of Applied Science in Engineering Physics, course concentration in Computer Science *Sept. 2018 – Apr. 2023*

- Cumulative GPA: 3.9/4.0
- Relevant Coursework: Advanced Machine Learning, Data Structures & Algorithms, Algorithm Design & Analysis, Relational Databases, Software Construction, Digital Systems and Microcomputers, Numerical Methods, Applied Linear Algebra, Probability with Physical Applications

Skills

Languages: Python, SQL, JavaScript, Java, C, C++, HTML, CSS
Frameworks: ReactJS, VueJS, Bootstrap, Flask, ROS
Tools & Libraries: Git, mpi4py, pytest, JUnit, OpenCV, Node.js, OracleDB

Work Experience

Research Intern **July 2022 – Aug. 2022**
SLAC National Accelerator Laboratory, Stanford University *Menlo Park, CA, USA*

- Developed a **Python** tool that used lightweight **machine learning** to detect and visualize outliers in high-dimensional experimental image data obtained from the **LCLS** free electron laser facility
- Researched and implemented an algorithm to perform incremental principal component analysis (iPCA) at scale, achieving **100%** statistical accuracy against **sklearn**'s implementation of batch PCA
- Achieved a **100x** increase in runtime performance by shifting to a parallelized **MapReduce** paradigm using **MPI**, managing resulting compute cluster using **SLURM**
- Currently expanding project functionality as a SLAC affiliate researcher

Teaching Assistant **Sept. 2022 – Present**
UBC Department of Computer Science *Vancouver, BC, Canada*

- Supervised twice-weekly lab sessions of **30+ students**, providing guidance on assignments assessing the fundamentals of data structures and algorithms in **C/C++**
- Hosted a weekly office hour, providing 1:1 support to a class of **200+ students**

Software Developer Intern **Jan. 2020 – Dec. 2020**
Validus Research, Inc. *Waterloo, ON, Canada*

- Developed a novel data visualization feature in **VueJS** that retrieved requested client data through a **RESTful API** built using **Flask**, streamlining workflows for **15** analysts company-wide
- Enabled the exporting of client data to formatted excel notebooks by leveraging the **SQLAlchemy** ORM and **openpyxl**, saving $\sim$ **2** man-hours weekly for analytics team
- Developed a comprehensive test suite using **Vue Test Utils** and **Jest** to surpass **90%** code coverage, deploying resulting project to production using a **Jenkins CI/CD** pipeline
- Returned after an initial four month internship (January - April) as a remote contractor starting in June 2020

Projects

cryoEM Data Management Platform | *Docker, Python, C*

- Developed ***compSPI***: an open-source, **Python** wrapped **C** program that generates large datasets of simulated cryoEM images
- Integrated developer contributions using a **CI/CD** pipeline with **Docker** and **GitHub Actions**
- Reduced cloud storage costs by **100%** by interfacing with the OSF.io API, an open source storage solution for scientific datasets

Music Label Manager | *SQL, OracleDB, PHP, HTML, CSS*

- Developed a CRUD web application in **PHP** and **HTML** to manage the client information of a fictional record label
- Successfully mapped **100%** of stakeholder data needs to a relational database schema implemented in **SQL** and maintained using **OracleDB**
- Reduced server response times by **60%** by rewriting bloated queries and leveraging frontend caching

Autonomous License Plate Classifier | *Python, ROS, Deep Learning, Computer Vision*

- Leveraged the **ROS** framework to build a simulated robot in **Python** and **C++** designed to autonomously traverse a competition environment, avoiding pedestrians and scraping license plate data from passing vehicles
- Successfully retrieved **100%** of license plates from passing vehicles using a novel car positioning subroutine combined with a modified SIFT algorithm from **OpenCV**
- Designed a convolutional neural network using **TensorFlow** and **Keras** to identify alphanumeric license plate characters, achieving **99%** classification accuracy on test datasets