Michael Elrod

Clemson, SC | michaelelrod.dev@gmail.com | +1 803 230 8694 | michaelelrod.dev

linkedin.com/in/michaelselrod | github.com/Michael-Elrod-dev

Technologies

Languages: Python, C/C++, Java, SQL, JavaScript, TypeScript

Software: PyTorch, AWS, Gymnasium, Docker, Postman

Education

Clemson University, BS in Computer Science — Aug 2020 – May 2024

- Coursework: Software Development, Cloud Architecture, Artificial Intelligence, Machine Learning

Clemson University, MS in Computer Science — Aug 2023 – May 2025

- Coursework: Deep Learning, Data Mining, AI-Receptive Software, Computer Vision, ML Optimization

Experience

Machine Learning Engineer - Intern, MIT Lincoln Laboratory – Boston, MA — May 2024 – Aug 2024

- Researched and developed a solution using a graph neural network for multi-agent path planning in unknown environments to increase autonomous collaboration in UAVs using Python & PyTorch
- Worked with the Lincoln Laboratory Super-computing Center (LLSC) to run simulations using parallel processing to decouple agent experience during model training

Software Engineer, Independence County Contracting, LLC – Batesville, AR — Aug 2024 – current

- Designed and developed a full-stack project management web application hosted on AWS for contractors to efficiently organize and distribute project plans
- Application design was created in Figma with a focus on a clear, minimalist features
- API was developed with TypeScript and hosted on AWS API Gateway
- Front-end of the app was developed with React & Tailwind and is stored in S3
- The cloud architecture was developed in AWS using DynamoDB, API Gateway, S3, EC2 and Lambda

AI Research Assistant, Clemson University – Clemson, SC — Aug 2023 – May 2024

- Researched and implemented the conversion of a traditional mathematical solution for drone swarm plant pollination to a deep reinforcement learning approach using Python, PyTorch & a deep q-network to increase model performance in dynamic environments
- Worked with other student researchers to develop the architecture in a collaborative lab environment

Software Engineer - Intern, Naval Information Warfare Center – Clemson, SC — Jan 2023 – Dec 2023

- Collaborated with the Blue Ridge Innovation and Entrepreneurship Foundation to develop a STEM-focused educational mobile app to provide an accessible learning platform for underprivileged students at The Dream Center in Easley, SC as a part of the NIWC STEM Outreach Program
- Developed the API using TypeScript and tested it with Postman, utilizing Docker for local hosting
- Contributed to the mobile app mock-up design using Figma in a collaborative team environment
- Implemented the application's front-end with Flutter, integrated API endpoints, and deployed the solution on AWS

Software Engineer - Intern, BlueCross BlueShield SC – Columbia, SC — May 2023 – Aug 2023

- Engineered an autonomous solution to identify and remove unused objects from the department's database, resulting in a 12% reduction in storage usage using Python & Selenium
- Contributed to the team by assisting in developing and maintaining new features for client contact centers using Java, Python, and proprietary software

Projects

Deep Reinforcement Learning github link

- Practiced implementing various Deep Q-Networks (DQN) using Python & PyTorch to solve single and multi-agent environments through Open AI's Gymnasium & MiniGrid
- Tested DQN capabilities in rapid image classification by applying developed solutions to various quick time events in video games
- Research how Graph Neural Networks can be used to share aggregated information with other agents in collaborative and adversarial environments
- Tools Used: Python, PyTorch, Gymnasium, MiniGrid, Super-Computing

Machine Learning github link

- Implemented and compared multiple regression models (linear, quadratic, cubic, fourth-degree) for data prediction
- Developed K-means clustering algorithm for unsupervised learning and data segmentation
- Created logistic regression and SVM classifiers, demonstrating proficiency in supervised learning techniques
- Utilized cross-validation, feature scaling, and polynomial feature generation for robust model evaluation and performance
- Implemented gradient descent optimization for logistic regression training
- Tools Used: Python, Scikit-learn, Matplotlib

Computer Vision github link

- Implemented Canny edge detection algorithm with Hough transform for line detection
- Created facial detection system using HOG features and the sliding window technique
- Built a panorama image stitching pipeline with feature matching and RANSAC
- Implemented optical flow algorithms for feature tracking and object detection
- Tools Used: Python, NumPy, scikit-image, Matplotlib, SciPy

Cloud Architecture github link

- Foundational Cloud Practitioner Certified
- Studied and practiced the use case and implementations of several Amazon Web Services through their student learning platform to learn how to develop and host scalable applications using a serverless architecture
- Tools Used: API Gateway, Lambda, DynamoDB, EC2, S3, Elastic Load Balancing, CloudWatch

SQL Database Management github link

- Designed and implemented normalized SQL databases, applying 1NF, 2NF, and 3NF principles to optimize data integrity and efficiency for data focused applications
- Developed a Java-based application simulating a multi-user customer service environment to rigorously test the database and demonstrate real-world application of database principles
- Tools Used: MySQL, Microsoft SQL, Java, Typescript

Interactive Software github link

- Developed two separate interactive game engines using both C++ & Python leveraging object-oriented programming principles and design patterns to optimize performance in high-speed processing environments
- Tools Used: C++, SDL2, Lua, Python, Pygame

Software Architecture github link

- Implemented Model-View-Controller to study the benefits of separating concerns in large, scalable applications
- Utilized the Observer pattern to optimize game engine software by reducing the number of instructions per cycle
- Applied Decorator techniques to add additional functionality to existing software packages
- Tools Used: Java, Python