Brady McAtee

bradymcatee12@gmail.com | bradymcatee.com | github.com/bradymcatee | linkedin.com/in/brady-mcatee

Education

Portland State University, BS in Computer Science, Minor in Physics — Sept 2023 – March 2026

- GPA: 3.9/4.0
- **Coursework:** Data Structures and Algorithms, Operating Systems, Computer Graphics, Computer Vision

Experience

Software Engineer Intern, MSEI Biotronik – Lake Oswego, OR — June 2025 – Present

- Built simulation models for multiple hardware components, enabling R&D engineers to test new device designs entirely in software
- Ported a legacy .NET Framework WPF library to .NET 8, ensuring compatibility across modern and legacy applications
- Refactored a graphical hardware simulation tool by separating core logic from display layers for cleaner architecture and easier maintenance
- Worked in an Agile environment using Git for version control and collaboration across the software and hardware teams

Projects

recipeBox — github.com/bradymcatee/recipeBox

- Built a full-stack web app for restaurants to manage recipes and ingredients with full CRUD support
- Designed RESTful APIs and normalized PostgreSQL schemas to keep recipe and ingredient data consistent
- Implemented responsive React UI with intuitive forms for adding, editing, and searching recipes
- Focused on clean architecture and efficient data handling for a smooth, real-world user experience
- Deployed app on AWS EC2 for restaurant staff use

Tennis Shot Tracker — github.com/bradymcatee/servetracker

- Trained a ball tracking CNN model based on the TrackNet architecture using PyTorch
- Used computer vision techniques and homography estimation to map 2D video coordinates to real-world court dimensions
- Measured tennis serve velocity from iPhone video recordings by combining frame analysis and geometric calibration

Interactive Photo Refocusing App — github.com/bradymcatee/refocusing

- Built full-stack app using Python, PyTorch, and Flask for DSLR-style depth-of-field simulation from single images
- Implemented Multi-Scale CNN (87.9M parameters) achieving 79% accuracy for monocular depth estimation
- Created responsive web interface with real-time click-to-focus interaction and configurable aperture effect
- Deployed production-ready solution with Docker and cloud deployment options

Plant Disease Classifier — huggingface.co/spaces/diseases

- Created PyTorch/ResNet model achieving 99% accuracy on 50,000+ plant images
- Used data augmentation to improve model performance across plant species
- Deployed model on Hugging Face Spaces for real-time disease diagnosis

Technologies

Languages: C++, C, C#, Python, Java, JavaScript, SQL

Technologies: .NET, Node.js, Express, PostgreSQL, Git, PyTorch, Jest, PyTest, NUnit, Linux, Vim