

Ethan Haines

ethanhaines.github.io

863-267-2636 | ethanrhaines@gmail.com | linkedin.com/in/ethanrhaines | <https://github.com/ethanhaines>

EDUCATION

University of Florida

Bachelor of Arts in Computer Science; GPA: 3.82

Gainesville, FL

Expected May 2026

State College of Florida

Associate's in Liberal Arts; GPA: 3.6

North Port, FL

Fall 2022 – August 2023

EXPERIENCE

Software Developer Intern

May 2024 – August 2024

SEMCO

Bartow, FL

- Automated an integration service to retrieve Sage Intacct data and update a SharePoint operations log every 4 hours.
- Connected Sage Intacct and Microsoft Graph APIs, and added Azure Monitor logging to improve reliability and troubleshooting.
- Used daily by 100+ employees tracking the status of 25+ different jobs at one time.

PROJECTS

NEST (Nearest Extant Similarity Tool) | Python, PyTorch, FAISS, OpenCV

January 2026 – Present

- Engineered a computer vision similarity engine for fossil pollen analysis using DINOv3 embeddings and FAISS cosine similarity search to retrieve nearby extant grain matches.
- Collaborated with the Curator of Paleobotany and the Assistant Curator of Artificial Intelligence at the Florida Museum of Natural History to align NEST with museum research workflows.
- Designed an OpenCV slide-cropping and annotation workflow with persisted markers, species normalization, and metadata generation for downstream search.
- Ran embedding extraction with flip/rotation augmentation, batch inference, and GPU/CPU fallback to improve robustness across grain orientations.
- Processed and indexed 900+ grain images across 6+ species, then exported interactive t-SNE views and web-ready graph JSON (nodes/edges/manifest) for research demos.

Kalshi Weather Prediction Bot | Rust, Tokio, Reqwest, Kalshi API, NWS, Open-Meteo

February 2026 – May 2026

- Created an asynchronous Rust prediction-market bot for Kalshi temperature contracts using multi-model weather forecasts (GFS, ICON, ECMWF, GEM, NWS) with paper and live modes.
- Implemented risk-adjusted pricing and execution with normal-CDF probability estimation, uncertainty haircuts, spread/liquidity filters, model-agreement checks, and half-Kelly sizing.
- Integrated RSA-signed Kalshi API requests with environment-based demo/production configuration and order/trade logging.
- Added risk controls and post-trade analytics, including exposure caps, daily loss limits, circuit breaker logic, and reconciliation/reporting utilities.

ALAN (Academic LLM Analysis Network) | Python, Bittensor, LLM Evaluation

December 2025 – May 2026

- Prototyped a Bittensor subnet for academic peer review quality evaluation with a custom miner/validator protocol for structured review submissions.
- Built an OpenReview ingestion and preprocessing pipeline that normalizes reviews, generates paper metadata, and splits per-miner datasets for local subnet testing.
- Authored an LLM rubric scorer (6 weighted criteria) with JSON validation, caching, and confidence scaling for consistent review scoring.
- Orchestrated grouped-by-paper ranking and normalized reward assignment (exponential rank decay) to simulate validator scoring across miner responses.

TECHNICAL SKILLS

Languages: Python, Rust, JavaScript, SQL, C++, Java, C, HTML/CSS, Lua

Frameworks/Libraries: PyTorch, Transformers, FAISS, OpenCV, NumPy, scikit-learn, Matplotlib, React, Node.js, Express, Requests

Tools/Platforms: Git, Linux, Microsoft Azure, Postman, Jira, GNU Debugger, Bittensor, OpenReview, OpenAI API, Kalshi API

Keywords: REST APIs, Data Pipelines, Async Programming, Distributed Systems, Machine Learning, Computer Vision, LLM Evaluation, Similarity Search, Risk Management