Matthew Hoffman

matthewhoffman.dev

Education

- **University of Texas at Austin** Austin, TX
  B.S. Computer Science; GPA: 3.75/4.00 Aug 2015–May 2020

Experience

- **Protopia AI** Austin, TX
  Research Engineer May 2022–Present
  - **Real-time Face Recognition Pipeline**: Deployed a real-time face recognition pipeline and optimized its throughput from 6 FPS to 60+ FPS. Parallelized the pipeline using Python multiprocessing and shared memory; implemented face-tracking and five-crop algorithms in NumPy, vectorized KNN using PyTorch, vectorized array operations using the OpenCV CUDA module, and optimized array management & operations; wrote a script that reused the pipeline to generate training & validation datasets to handle data drift.
  - **PyTorch Training Library**: Rearchitected the entire internal PyTorch training library used to train all models at the company. Removed dead code, restructured the module hierarchy, consolidated class design, and simplified the training control flow. Transitioned the configuration model to pydantic for improved data modeling and automatic schema generation & validation. Enabled unique features and guarantees necessary to do correct training for research.
  - **Dynamic Activation Routing**: Implemented dynamic rerouting of neural network activations in PyTorch using a wrapper class & forward hooks and enabled its use in parallel training.
  - **Engineering Excellence**: Lead initiatives to improve and standardize engineering practices, including adherence to the Google Python Style Guide and Google-style docstrings, the usage of linting, type hinting, code coverage reports, and updates to the build-specification & CI pipelines.
- **Amazon.com, Inc.** Austin, TX
  Software Development Engineer Jun 2020–May 2022
  - **Accounting Configuration**: Delivered the accounting requirements for multiple large-scale, cross-region accounting projects as a solo developer; responsible for the clarification of requirements, design, implementation, testing, verification, communication of timelines with stakeholders, and post-launch monitoring.
  - **Workflow Orchestrator**: Designed an accounting workflow orchestrator using AWS Step Functions, Lambda, SQS, SNS, & DynamoDB.
  - **Personal Project—VS Code Language Server Extension**: Created a Language Server extension in TypeScript to improve the developer experience when writing accounting configuration in Amazon Accounting's custom language. Implemented syntax diagnostics, redefinition error detection, keyword definition hovers, dynamic go-to definition, code-completion, and format on-save.
  - **Personal Project—Automated CLI Generator**: Created a Python module to automatically create command-line parsers by statically analyzing the entry point function and its documentation. This module simplified the developer experience and served as a central place to register and share commonly used scripts between team members.
  - **Personal Project—Web Scraping & Requests Automation**: Created a Python package to scrape and automate requests to Amazon-internal websites. Used this package to create a script to automate an event failure process saving 1 developer-day per week. This package also saved multiple days per project when used to collect and validate transaction examples during end-to-end testing.
- **Blue Cross and Blue Shield of IL, MT, NM, OK & TX** Richardson, TX
  Data Science Intern Jun 2019–Sep 2019
  - **Data Visualization**: Created a Python script to generate interactive choropleth maps. Combined multiple publicly available data sources to visualize insurance coverage by zip code and identify candidate locations for BCBS' Care Van immunization program.
  - **Text Mining**: Used regular expressions and NLP models to analyze case management notes to identify member outcomes for use in program evaluations.

Projects

- **Ray Tracer**: Implemented core functionality and additional features of a ray tracer in C++; homogeneous coordinates; ray-triangle intersection; interpolation of surface normals; reflection, refraction, and attenuation of light; shadow attenuation; anti-aliasing; texture mapping; cube mapping; bounding volume hierarchy (acceleration structure).
- **Monopole Cell Tower Detection**: Trained an object-detection neural network in PyTorch using the YOLO (you only look once) architecture to detect monopole cell towers from satellite images (precision = 0.1750, recall = 0.8696); advised by Dr. Sanjay Shakkottai and Dr. Constantine Caramanis.

Programming

- **Languages**: Python, C, C++, Java, TypeScript **Computing**: PyTorch, NumPy, OpenCV
- **Data Analysis**: pandas, SQL, scikit-learn, matplotlib **Cloud**: AWS