

Yuan Huang

yuan.huang.yhh@gmail.com | 213-519-2954 | linkedin.com/in/yuan-huang-945995203

Education

University of Southern California | Master's Degree

Los Angeles, CA | 01.2021 - 12.2022

Electrical and Computer Engineering (Machine Learning and Data Science)

East China University of Science and Technology | Bachelor's Degree

Shanghai, China | 09.2016 - 07.2020

Electrical Engineering (Computer Information Science)

Work experience

Corteva Agriscience | Software Engineer

Des Moines, IA | 08.2024 – Now

Backend engineer building distributed systems and real-time data pipelines in production. Work spans edge computing on NVIDIA Jetson/K3s, event-driven microservices, and automated geospatial data ingestion across cloud and on-prem environments.

- Real-Time Sensor Data Acquisition Platform – Argus

- Contributed to migration from an inheritance-heavy Factory-pattern architecture to **containerized microservices**, standardizing **gRPC** contracts across 10+ sensor services running on **NVIDIA Jetson** edge devices.
- Built a **Kubernetes**-based Sensor Manager using the Python **K8s** API to orchestrate ephemeral sensor pod lifecycles with event-driven provisioning, health polling, and parallel startup workflows.
- Implemented distributed state coordination using **MongoDB** and **Redis** for session persistence, transient caching, and multi-device orchestration.
- Diagnosed and fixed a **GStreamer** frame-drop bug at EOS that broke GPS-to-video frame alignment in recorded sensor data, restoring reliable post-run geospatial reconstruction.
- Replaced **Redis pub/sub** with **NATS-based** telemetry distribution to avoid injecting Redis credentials into ephemeral sensor pods and simplify secure edge-device communication.
- Introduced **Redis Streams** as a durable command buffer after direct **ZMQ** publishing caused command drops and sensor deadlocks under concurrent sensor workloads.
- Developed integration tests covering **gRPC** contracts, sensor lifecycle flows, and edge cases including multi-sensor concurrency and connection failure handling.

- Full-Stack Image Processing Pipeline – FSIP

Sole engineer owning end-to-end delivery, including pipeline orchestration, desktop app, and cloud infrastructure shipping to operators across the world.

- Implemented **Argo Workflows orchestration** across 11+ containerized microservices for LiDAR, multispectral, and photogrammetry processing; built Python resolver mapping 10+ analysis types to parameterized YAML templates with runtime GPU scheduling, processing 100+ flight jobs per week.
- Maintained core management API (**Flask, PostgreSQL**) handling flight lifecycle, execution tracking, and result retrieval (TIF, shapefiles, logs); cursor-based pagination and Flask-Alembic migrations to **AWS RDS**.
- Shipped desktop ingestion app (**Flask + PyInstaller**) to 100+ field operators with Azure B2C OAuth2 and EXIF/shapefile processing; **S3** manifest upload triggers in-cluster cron job invoking downstream flight APIs via **Lambda/API Gateway**.
- Provisioned and maintained cloud infrastructure with **Terraform** and **GitLab CI/CD**, deploying across 4 environments (DEV/QA/PROD/COM) to both **EKS** and on-prem Kubernetes clusters.

- Enterprise AI Data Integration Pipeline

Automated geospatial ingestion pipeline handling up to 200 drone survey flights per day (20 GB per flight), replacing a previously manual process.

- Architected **Databricks** pipeline ingesting orthomosaic GeoTIFF imagery from DroneDeploy via **GraphQL** API with concurrent thread pool downloading and streaming ZIP extraction.
- Built ArcGIS FeatureServer shapefile upload endpoint with coordinate reprojection and batched uploads, enabling automated geospatial layer sync for downstream ML inference pipelines.
- Designed 3-step idempotent pipeline with cursor-based pagination resumption and exponential backoff, ensuring reliability under API rate limits and multi-hour export windows.

- Autonomous AI Developer Tooling – ContextBot

- Built an **MCP (Model Context Protocol) server** in **FastAPI** that auto-injects project-specific rules and prior session history into Claude Code CLI and GitHub Copilot, so each session starts with full context.
- Designed a memory layer with cross-project semantic rules and append-only audit logs, keeping AI context in sync with the local repo whenever rule or history files change.
- Exposed tools bridging the MCP server to the local dev environment, giving the assistant grounded context for code generation and debugging instead of generic responses.

Beaconfire Solution | Software Developer

Plainsboro, NJ | 01.2024 - 07.2024

- Built internal collaboration platform with **Flask/React** microservices architecture, PostgreSQL via **SQLAlchemy** on **AWS RDS**, and async email notification system via **RabbitMQ**.
- Containerized and deployed services using **Docker** and **AWS ECS** with **JWT**-based authentication.

University of Southern California | Research Assistant

Los Angeles, CA | 04.2022 - 01.2024

- Built a retrieval-augmented generation (**RAG**) pipeline over long-context QA data — implementing document chunking, embedding generation, and cosine similarity-based passage retrieval to ground **LLM reasoning** in relevant context.
- Improved multi-choice QA accuracy by **20%** on a long-context benchmark by redesigning transformer architecture with parallel encoding branches per answer choice, reducing cross-choice interference.
- Developed synthetic data augmentation strategies to improve **multi-modal LLM** robustness against out-of-distribution instructions in **Vision-Language Navigation** tasks; applied **Scaling Law**-guided experiments to inform model architecture decisions.

Skills

- **Languages:** Python, C/C++, JavaScript/TypeScript, SQL, Shell
- **Backend:** FastAPI, Flask, Django, gRPC/Protobuf, REST APIs, GraphQL
- **AI/ML:** PyTorch, TensorFlow, RAG pipelines, embedding models, Transformer fine-tuning
- **Data & Messaging:** PostgreSQL, MongoDB, Redis, NATS, ZeroMQ, RabbitMQ, Kafka
- **Infrastructure:** Kubernetes/K3s, Docker, AWS (Lambda, S3, EC2, EKS, ECR, RDS, API Gateway), Terraform, GitLab CI/CD
- **Orchestration:** Argo Workflows, systemd, Databricks
- **Real-Time & Edge:** GStreamer, gRPC streaming, NVIDIA Jetson