Fatih Ayıbasan

+90 506 176 84 16 | fathaybasn@gmail.com | linkedin.com/in/fatih-ayibasan | github.com/fatihaybsn | fathaybasn.me

Summary

Final year Computer Engineering student focused on Edge AI, embedded computer vision and autonomous systems, particularly deploying and optimizing inference on resource-constrained hardware. Hands on project experience spans edge devices, ARM64/Linux tooling, real-time vision pipelines, backend systems and UAV integration. Only the internship remaining for graduation, seeking an internship in defense, robotics, embedded, AI with immediate availability.

Technical Skills

Edge & Embedded: CUDA, TensorRT, ONNX, NVIDIA Jetson (TX2, Xavier, Nano, Orin), Raspberry, Navio, GStreamer, MAVLink, Pixhawk, Gazebo, OpenDroneMap
AI & Computer Vision: PyTorch, Hugging Face Transformers, LLM Fine-tuning, Ollama, PEFT/LoRA, INT8 Quantization, Multi-task Learning, scikit-learn, ANFIS, OpenCV, NumPy, Pandas, Matplotlib
Backend: FastAPI, PostgreSQL, Redis, GitHub Actions (CI/CD), TCP/UDP, Pytest, OpenTelemetry, RAG
Languages: Python, C++, C#
DevOps & Tools: LangGraph, MCP, Git, CMake, Docker, Podman, QEMU, Linux

Experience

Teknofest Club President Sep 2024 – Jul 2025

ARES UAV Team

- Led 15 member ARES team through all qualification stages to TEKNOFEST 2025 Combat UAV Finalist; executed live autonomous flights on the Gökçeada Airport competition field under jury evaluation.
- Built an end-to-end, camera-fed YOLOv8 target detection pipeline on NVIDIA Jetson Orin NX — custom dataset preparation, model training, and CUDA inference via TensorRT — sustaining high FPS under active flight conditions.
- Designed the six-state mission logic (idle, search, lock-on, dive, abort, RTB) that turns live detections into MAVLink commands for the Pixhawk Cube autopilot.
- Built ARE-ION, a C# ground control station with live MAVLink telemetry, geofence visualization, and Python-based video streaming, used alongside Mission Planner during the competition.

Software Team Lead May 2023 – Jul 2024

AUVSI SUAS International UAV Competition

- Trained and deployed a YOLOv8 model on a Jetson Nano with a Raspberry Pi HQ Camera for real-time alphanumeric ground-target detection, achieving high quality and high FPS during testing.
- Integrated the Jetson Nano with a Pixhawk Cube Orange over MAVLink to autonomously hold the aircraft over a detected target and deploy cargo using a motorized fishing-line delivery system, validating the mission logic in Gazebo before flight.
- Built a post-flight pipeline using OpenDroneMap to generate orthophotos from aerial imagery and transfer them to the ground station over FTP.

Projects

Orin Stage | *JetPack, Cross-Compilation, Podman, QEMU* Jul 2026 – Sep 2026

- Built a Python CLI that creates persistent, isolated JetPack development environments for the NVIDIA Jetson Orin family on x86_64 Linux using official NVIDIA inputs, rootless Podman and QEMU.
- Designed an immutable-base/mutable-workspace architecture that keeps multiple independent JetPack environments side by side and uses the same target filesystem for ARM64 execution and x86_64 cross-compilation.
- Validated an Orin Stage workspace against a physical Jetson Orin NX by matching JetPack/L4T and NVIDIA package identity and comparing all 51 nvidia-l4t-core payload files byte-for-byte. Ran the same ARM64 binary in both environments to confirm matching runtime behavior.
- GitHub · Portfolio · Validation

Adaptive Edge-Inference Controller | *Edge AI, Camera Pipeline, CUDA/TensorRT* Dec 2025 – May 2026

- Built a closed-loop thermal controller for continuous YOLO inference on Jetson Orin NX, using real-time GPU telemetry, FOPDT prediction, fuzzy logic, and safety guards to dynamically adjust input resolution and frame-processing ratio.
- Characterized both workload controls through separate hardware experiments and fitted FOPDT thermal models ($R^2 = 0.963$ and 0.936); the measured performance–thermal trade-offs informed the controller's scaling strategy.
- Validated the controller in a 130-minute hardware run with 3,900 telemetry samples, reaching a peak GPU temperature of 81.82°C without triggering the 85°C emergency threshold.

- GitHub · Technical Article · Portfolio · Report

PathFinder-Ship | *Hybrid RAG, Intent Routing, Pydantic, ONNX Runtime, ChromaDB* Sep 2025 – Dec 2025

- Built a local-first multimodal assistant with a FastAPI orchestration layer that routes requests across INT8 ONNX Flan-T5 generation, MiniLM intent classification, hybrid RAG, browser camera actions, and YOLO object detection on CPU.
- Fine-tuned Flan-T5 Large(783M parameters) with LoRA on 100,000 sample Chat/RAG dataset, compared multiple training configurations, and achieved 0.5216 Chat F1, 0.8894 RAG F1, and 0.7938 RAG EM. Exported the selected model to INT8 ONNX for local CPU inference and published both LoRA and ONNX artifacts on Hugging Face.
- Designed the backend around a central 'PipelineOrchestrator' with typed Pydantic contracts and separate modules for intent prediction, routing, retrieval, generation, and detection. Supports local T5 and Gemini providers, ChromaDB, SQLite FTS5/BM25 retrieval, and gated web search.
- GitHub · Portfolio · Hugging Face

MRI OOD Benchmark | *PyTorch, Medical AI, OOD Generalization, scikit-learn* Jan 2026 – Feb 2026

- Benchmarked 10 standard, hybrid, and custom architectures across 13 checkpoints for brain MRI tumor/no-tumor classification using 15,000 images with source and resolution variation.
- Designed a custom Multi-Scale Attention Fusion (MSAF) + EfficientNet-B0 architecture that ranked first on the external test set, achieving 90.8% accuracy, 0.988 ROC-AUC, and 0.901 F1.
- Published all 13 checkpoints on Hugging Face with Safetensors, model cards, OOD metrics, loading instructions, and SHA-256 provenance
- GitHub · Portfolio · Hugging Face

Diagent | *FastAPI, PostgreSQL, Celery, Redis, LLM-as-Judge, Docker Compose* May 2026 – Jul 2026

- Built a self-hosted FastAPI/PostgreSQL observability backend whose Python tracer records agent runs, LLM/tool spans, and retrieval evidence, while Celery workers evaluate anomalies and RAG quality.
- Dogfooded it with PathFinderShip as the first real integration, keeping application-specific policy in the calling system.
- GitHub · Portfolio · Dogfooding

API Change Radar | *Docker, PostgreSQL, OpenTelemetry, Pytest, Pydantic* May 2026 – Jul 2026

- Built and publicly deployed a FastAPI service that compares 2 OpenAPI specifications, detects breaking endpoint and schema changes, assigns severity levels, and stores reports in PostgreSQL; packaged it with Docker and validated the upload-to-report flow through CI and end-to-end tests.
- Live · GitHub · Portfolio

Education

Atatürk University Erzurum, Türkiye
B.Sc. Computer Engineering *Aug 2022 –*

Technical Writing

"Thermal-Aware Closed-Loop Workload Control for Long-Running YOLO Inference" *Medium, May 2026*

"Semantic Caching with Redis: How to Optimize LLM Cost and Latency" *Medium, May 2026*

"Building OpenCV with CUDA Support on Jetson Devices" *Medium, Apr 2026*