

Aditya Kaleeswar

+91 8438353880 | adityakaleeswargk04@gmail.com | [LinkedIn](#) | [GitHub](#) | [Portfolio](#)

EDUCATION

Vellore Institute of Technology (VIT)

B.Tech in Computer Science — CGPA: 8.41

Maharishi Vidya Mandir

Senior Secondary, CBSE — Class XII: 96%, Class X: 94%

Vellore, India

Sep 2022 – Jul 2026

Chennai, India

Apr 2018 – Jul 2022

EXPERIENCE

Research Intern

University of Texas Rio Grande Valley (AIM Lab) — Remote

Jan 2026 – Jul 2026

ECC: [Preprint](#) / [Code](#) · PINN: [Preprint](#) / [Code](#)

- Built a CatBoost model predicting the strength and ductility of ECC, a bendable concrete, from its mix design
- Added conformal prediction for calibrated uncertainty ranges — ~88% observed coverage on 80% intervals
- Benchmarked against ET+GBR ensembles, NGBoost, LightGBM, TabPFN, and Mixture-of-Experts variants
- Built an inverse-design pipeline: screens 100k candidate mixes for the cheapest one meeting a target strength, with out-of-distribution checks so recommendations stay within data the model trusts
- Built a single image-to-verdict crack pipeline: a vision front end measures each crack's geometry, and a physics-informed neural network converts it to a structural severity verdict — within 6.5% of a finite-element reference
- Made the vision front end pluggable: any YOLO segmentation model swaps in without touching the physics backend

PRISM Research Intern

Samsung Research, Bangalore — Remote

Oct 2024 – Jun 2025

[Code](#)

- Built BQ2Viz, a multi-agent GenAI framework that turns natural-language questions into data visualizations
- Chained three agents: a profiling agent summarizes the data, a reasoning planner infers which files and charts the query needs, and a code-generation agent writes and runs the plotting code
- Achieved 84.57% accuracy on the VisEval benchmark's complex multi-table queries, ahead of the NVAGENT baseline
- Cut error rate to 11.5% with a self-correcting loop that diagnoses and repairs failed chart code

Software Engineer — Autonomous Systems

Team Vyadh (Mars Rover Team), VIT — Vellore

Apr 2024 – Sep 2025

[Rover stack](#) · [Obstacle avoidance](#)

- Built the rover's autonomous navigation stack on ROS 2 and Nav2, using frontier exploration to map unknown terrain within dynamic GPS geofences
- Implemented real-time obstacle and pit avoidance from Intel RealSense depth point clouds, with path recovery and heading correction for wheel drift
- Cut state-estimation noise by fusing IMU and wheel odometry in an Extended Kalman Filter; team placed 13th at IRC 2025 and 4th at IRDC 2025

PROJECTS

HazMap — Risk-Aware Exploration on an Autonomous Rover | ROS 2, Nav2, YOLOv8

[Preprint](#) · [Code](#)

- Designed HazMap, an exploration algorithm coupling gas sensing and vision: gas anomalies trigger close-up sweeps of suspected leaks, while visual hazard detections steer the rover away from danger
- Built a coverage planner that inspects cluttered regions densely and skims open floor, cutting waypoints by ~60%
- Localized gas sources to within 0.25 m and flagged the first hazards within 2% of the mission — validated in the GADEN gas-dispersion simulator ([sim benchmark](#)), then deployed on an NVIDIA Jetson rover running perception at ~15 FPS

Alpha Stack — Natural Language to Production Codebases | Python, C++, Tree-sitter, TUI

[Preprint](#) · [Code](#)

- Built a terminal-based engine where an orchestrator agent turns a prompt into a software blueprint and parallel worker agents generate each file with inherited context, keeping the whole codebase mutually consistent
- Wrote DGAT, a C++ dependency engine that parses imports across 12+ languages using Tree-sitter and re-analyzes only changed files — keeping the project's dependency graph valid and cycle-free during generation
- Added self-healing validation: builds and tests run in an isolated sandbox, and a diagnosis-repair agent pair fixes failures until the project converges
- Created AlphaBench, a 40-task benchmark across CUDA, Go, Rust, and TypeScript, graded by real build-and-test execution instead of code matching
- Benchmarked frontier models — GPT-5.2, Gemini 3 Pro, Claude Opus, DeepSeek V3 — showing the self-correction loop lifts project success from $\leq 45\%$ to 75%

PATENT

Force Sensor-Based Train Presence & State Detection

Indian Patent App. No. 202641007500

- Under-track force-sensor arrays and distributed ESP32 nodes running a finite-state machine passively classify a train's state — approaching, stopped, pass-through, departing — and its direction of entry, with no train-side equipment

TECHNICAL SKILLS

Languages: Python, C/C++, JavaScript (ES6+), SQL (Postgres)

GenAI & ML: PyTorch, LangGraph, LangChain, RAG, ChromaDB, CatBoost, Conformal Prediction, PINNs

Robotics & Vision: ROS 2 (Nav2, SLAM Toolbox), OpenCV, PCL, YOLOv8, SAM, NVIDIA Jetson, ESP32

Web & Data: React, Node.js, Express, MongoDB, REST APIs, pandas, NumPy, NetworkX, DuckDB

DevOps & Cloud: GCP, Docker, Git, Linux