

Kamal Mehta

Dallas–Fort Worth, TX | kamalmehta.ai@gmail.com | linkedin.com/in/kamal-m-3ba627b4 | github.com/kamalkimahek14 | <https://kamal-mehta.com>

Machine Learning Engineer | Perception Engineer

SUMMARY

Machine learning and perception engineer, 5+ years across industry and graduate research. Shipped deep reinforcement learning to 500+ autonomous freight locomotives as onboard models about 100 times smaller than the model they replaced, under 10 ms per decision on CPU (12% fuel saving, EPA-certified). Built camera and LiDAR perception for an unmanned vehicle (YOLOv3, Kalman fusion), physics simulation with a sim-to-real correction, and AWS data pipelines. Worked directly with an Australian railroad client, from data validation to fleet deployment.

SKILLS

Methods: Machine Learning (ML), Deep Learning, Deep Reinforcement Learning (RL), Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Reward Design and Reward Shaping, Supervised Learning, Transfer Learning and Fine-Tuning, Feature Engineering, Hyperparameter Tuning, Model Evaluation (blind held-out, LLM-as-judge), Model Monitoring

Computer Vision: Computer Vision (CV), Object Detection (YOLOv3), Convolutional Neural Networks (CNNs), Sensor Fusion, Kalman Filter and State Estimation, LiDAR and Stereo Camera Calibration

Generative AI: Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), AI Agents, Llama 3.1, Cloudflare Workers AI, Agentic Software Engineering (AI coding agents, GitHub Copilot)

Languages and Frameworks: Python, PyTorch, TensorFlow, scikit-learn, pandas, NumPy, SQL, C++, Cython, MATLAB

Cloud and MLOps: Amazon Web Services (AWS): S3, Lambda, QuickSight, Event-Driven Serverless Pipelines, Docker on AWS, CI/CD (AWS CodePipeline, GitHub Actions), Git, Linux, OpenSearch

Data: Extract, Transform, Load (ETL) Pipelines, Time-Series Data, GIS, Data Validation, Multiprocessing, Dashboards

Simulation and Robotics: Physics-Based Simulation, Digital Twins, System Identification, Robot Operating System (ROS), Gazebo, Path Planning, Edge Deployment (locomotive onboard computers)

EXPERIENCE

Machine Learning Engineer | Progress Rail, A Caterpillar Company | Jun 2022 – Jun 2026

- Shipped Deep Q-Network (DQN) driving models to the onboard gateway computers of all 500+ locomotives, under 10 ms per decision, CPU-only: each a 15 to 20 MB file, about 100 times smaller than the 2 GB supervised tree ensemble they replaced.
- Moved freight-train driving to deep reinforcement learning in PyTorch: DQN policies (MLPs) that make each throttle and brake decision from the track ahead and the train state, trained in simulation, with rewards from the features that made the supervised model generalize.
- Shaped the reward toward a rule-based reference driver on ~2k synthetic trips; my own rule-based refinement layer raised saving in high-opportunity spots and fixed unsafe behavior.
- Built a slack predictor (a small fully connected network, trained offline) that runs on board as a learned safety filter, blocking any action predicted to cause a serious slack event; in a paired lab scenario check it blocked both over-limit events.
- Reached a 12% fuel saving vs manual driving on trips from Aug 2024 to Dec 2025, and 3 to 4% better trip time; the software is EPA-certified (2025), and the earlier supervised pilot, a separate cohort, saved about 5%.
- Set the fuel targets and measured the result with the EPA methodology: like-for-like groups by territory, train type and power setup, ton-mile weighted, with the sim-to-real gap fed back as a fuel correction factor.
- Engineered look-ahead terrain features so one model serves each train type: 4 to 6 RL models replaced per-territory models as territories grew from 2 to ~6–7 (supervised pilot) to 100+ (RL). Local RL training cut AWS training cost ~80%.
- Cut normalized RMSE of fuel and running time on held-out field trips (random 80/20 split) from 0.25 to 0.15 by calibrating the physics-based train simulator with a bounded grid search, shipped as an event-driven AWS S3 and Lambda pipeline feeding the C++ and Cython physics engine; extended its Python bindings to fit the engine and simulator changes.
- Investigated 100+ client-reported field issues under a 24-hour response SLA, from locomotive to cloud, and added a regression test for each.
- Wrote an AWS Lambda reliability pipeline and a replacement fleet-health crawler as fleet availability rose from about 90% (Jan 2024) to about 99% (Jun 2026).

- Took an Australian railroad client from its first 50 to 100 trips to fleet-wide deployment with an ETL pipeline for ~300-car iron-ore trains with ECP brakes, confirming data standards with the client.
- Fused two telemetry sources into per-second records with GIS track profiles on AWS S3 and Lambda; its trip analysis fed model updates that reached about 8 to 10% fuel saving in that pilot.
- Implemented an electronically controlled pneumatic (ECP) air brake model in the C++ physics engine from research papers, which enabled the first deployment to a railroad in Australia.
- Parallelized train consist generation from OpenSearch with Python multiprocessing: about 2,500 consists in 4 to 5 hours instead of about 1 week, helping grow from 55 to 120 approved territories in a year.
- Presented fuel and reliability analyses, and their method, to directors and the SVP of Advanced Technology. Mentored two interns, whose work is still in use.

Technology Developer and Researcher | Artifex Technologies, Inc. | Dec 2021 – May 2022

- Built the point-cloud and IMU pipeline that turns handheld scans into floor plans; density-based clustering separates walls from open space as the user walks. About 95 to 98% wall-position accuracy against architects' plans (within 5 cm), a plan in about 2 minutes; piloted by about 30 architects (about 100 plans).

Graduate Research Assistant | Aerospace Systems Laboratory, The University of Texas at Arlington | Jan 2020 – May 2021

- Fine-tuned the YOLOv3 deep learning object detector in MATLAB (SqueezeNet backbone) with transfer learning on 10,989 labeled images, plus augmentation. Tuned hyperparameters (bias-variance); test mAP@0.5 of 96.25% and mean IoU 0.82 (4 classes, 220 test images, a random 2% split of the same recordings), 59 ms average detection time per image.
- Calibrated a stereo camera and LiDAR (intrinsic and extrinsic) and fused object positions through a linear Kalman filter (constant-acceleration model); in simulation, fusion gave each object its class and position at 0.42 m error (camera alone 0.97 m, LiDAR alone 0.29 m). Ran detection and position estimation live on an Intel NUC with a GTX 1050 GPU, stereo IMX179 cameras and a RealSense L515 LiDAR at 1.44 s a frame for the full pipeline (about 8 s on CPU); cut training time about 50% through GPU and CPU memory optimization.
- MS thesis: object classification, detection and state estimation with YOLOv3 and stereo camera and LiDAR fusion (UT Arlington, 2021; in the university repository).

PROJECTS

Disha, a career profile reviewer in Python

- Scores a resume, LinkedIn, GitHub and portfolio separately against live postings for one target role, with strict role matching, skill gaps from those postings, an ATS recruiter view and inline-diff rewrites.
- Writes tailored DOCX resumes and letters within page and word limits, a designed PDF report and a course-style roadmap; local-first, PII scan and 617 tests in CI; refined over panel review rounds R19 to R24.

Private local-first AI board with blind held-out evaluation loops (current work)

- Specified a local-first AI board (local LLM agents, a voice assistant) built by AI coding agents; its loops fix on a dev set and score fresh held-out sets: voice-assistant pass rate 55.3% to 71.2% (3-round means, 44 new prompts each round), resume parse 85.6% to 100% across 5 parsers.

Double DQN eco-driving (public: github.com/kamalkimahek14/dqn-train-energy)

- Public learning project on a simulated train, PyTorch and TensorFlow: 31.5% less fuel than a speed-holding baseline on 50 test tracks, on time on 99% (2 seeds); a slack predictor cut slack events 93% (synthetic data).

Digital twin chat on kamal-mehta.com

- Built a generative AI chat with RAG over write-ups (Llama 3.1 8B, Cloudflare Workers AI); 100/100 regression checks.

EDUCATION

Master of Science in Aerospace Engineering | The University of Texas at Arlington | Aug 2018 – Jul 2021 | GPA 3.5/4.0

Bachelor of Engineering in Aeronautical Engineering | Gujarat Technological University | Aug 2013 – Jun 2017 | CGPA 8.28/10

CERTIFICATIONS

Graduate Certificate in Unmanned Vehicle Systems, UT Arlington, 2020; Neural Networks and Deep Learning, DeepLearning.AI, 2020; Supervised Machine Learning and Advanced Learning Algorithms, DeepLearning.AI, 2026; Self-Driving Cars: Introduction, and State Estimation and Localization, U of Toronto, 2020