MOKSH TREHAN

COMPUTER VISION | MACHINE LEARNING | SYSTEMS
Waterloo, ON | (647) 787-7866 | Mokshtrehan1@gmail.com | github.com/MokshTrehan

SUMMARY

MASc researcher and AI engineer building production-oriented perception and explainable agentic systems. With TensorOne, developing a standalone GPS-denied camera–IMU VIO/SLAM module on Raspberry Pi 4 that uses under 1 GB of RAM across CPU VIO, map matching, and an 8-bit 60M-parameter self-supervised model. Experience also spans temporal vision research, latency-sensitive AOSP software, and distributed systems.

EDUCATION

Master of Applied Science (MASc), Systems Design Engineering **Jan. 2026 – Present**
University of Waterloo, Waterloo, ON | Research focus: Machine Learning
Thesis focus: resource-efficient VIO/SLAM and online sensor recalibration for aerial robotics

Bachelor of Science, Computer Science **Sep. 2020 – May 2025**
Western University, London, ON

TECHNICAL SKILLS

ML / Computer Vision: PyTorch, CUDA, V-JEPA 2.1, multimodal self-supervised learning, on-device online learning, 8-bit/Q8 quantization, CNNs, Transformers, ConvNeXt, Mamba, visual SLAM, VIO, map matching, camera–IMU sensor fusion, GPS-denied navigation
Languages / Systems: Python, C/C++, Java, Go, JavaScript/TypeScript, Linux, Raspberry Pi 4 deployment, Raspberry Pi 5 training, AOSP/Android, Binder/IPC, multithreading, performance profiling
Apple / Graphics: Xcode, Objective-C, Swift, Metal, Metal Shading Language (MSL), OpenGL/GLSL
Agentic / Cloud: LLM agents, multi-agent orchestration, explainable AI (XAI), agent observability, tool-call tracing, source provenance, RAG, source verification, evaluation, REST APIs, AWS, Docker, Kubernetes, Jenkins/CI/CD, Git

EXPERIENCE

Graduate Researcher — Production VIO/SLAM **Jan. 2026 – Present**
University of Waterloo, SYDE | Industry collaboration with TensorOne Inc. | Waterloo, ON
- Develop a standalone GPS-denied VIO/SLAM module with TensorOne, fusing camera and IMU signals for production deployment on resource-constrained UAV hardware.
- Integrated map matching to correct accumulated altitude/scale drift, grounding the state estimate against mapped references to realign the SLAM scale factor during operation.
- Enabled end-to-end Raspberry Pi 4 execution by redesigning wasteful circular call paths and removing components redundant to the GPS-denied navigation pipeline; run the VIO estimator on CPU.
- Applied 8-bit (Q8) quantization to the trained 60M-parameter self-supervised model; the complete Pi 4 runtime—including CPU VIO, the model, and map matching—uses under 1 GB of RAM.
- Implemented online learning and training on Raspberry Pi 5 for a multimodal recalibration model combining V-JEPA 2.1 visual representations with IMU readings, using the Pi 5's additional memory while retaining Pi 4 as the deployment target.

AI Engineer **Oct. 2025 – Present**
TTEC | Remote
- Built an end-to-end research agent that verifies sources and automatically generates presentation and report drafts, reducing the sales team's RFP response cycle from ~1 month to ~1 week.
- Architected the workflow across agent orchestration, tool/API calls, retrieval, source validation, and document generation, owning implementation through testing, deployment, and evaluation.
- Built explainability and auditability into agentic systems through source provenance, citation validation, structured tool-call traces, and inspectable intermediate outputs, allowing users and reviewers to trace generated claims back to supporting evidence.

Research Assistant — Visual SLAM **Oct. – Dec. 2025**
Vision and Image Processing (VIP) Research Group, University of Waterloo | Prof. John Zelek
- Implemented and evaluated SLAM systems for autonomous visual mapping research, establishing the technical foundation for subsequent MASc work in visual-inertial state estimation.

EXPERIENCE — CONTINUED

Computer Science Researcher — Temporal Computer Vision (USRI) **May – Aug. 2025**

Western University | London, ON

- Designed a temporal object-recognition model for rapid serial visual presentation using a ConvNeXt backbone, Mamba state-space recurrence, sensory adaptation, and Transformer attention.
- Built a reproducible PyTorch/CUDA training and evaluation pipeline across 13–100 ms dwell times and measured Top-1/Top-5 accuracy as a function of presentation duration.
- Ran ablations across recurrence, adaptation, and attention; recurrence plus sensory adaptation produced the strongest results on ultra-rapid sequences.
- Evaluated cognitive alignment and dual-target behavior across power-law and exponential adaptation variants; maintained IEEE-style reports and led weekly reviews.

Software Engineer Intern **Sep. 2023 – Aug. 2024**

General Motors | Markham, ON

- Owned Java/AOSP infotainment features from design and implementation through unit/integration testing and release.
- Reduced interaction latency by ~100 ms through Android profiling, tracing/logging, thread-scheduling changes, and UI event-loop optimization; the result was validated by QA on test benches and in vehicle trials.
- Debugged AOSP–GM integration across services, lifecycle, and Binder/IPC and hardened error paths/timeouts, contributing to a ~30% reduction in reported issues within the team's scope.
- Built Jenkins CI pipelines with parallelization, caching, and quality gates, reducing test cycles by ~60% while increasing automated coverage.
- Developed thread-safe background services and state-machine-driven flows; validated hardware-adjacent behavior on emulators and bench devices under connectivity, power, and lifecycle transitions.
- Built mocks and shims for hardware interfaces and partnered with QA on soak/load testing, diagnosing failures from logs and telemetry.
- Authored design notes, runbooks, and release documentation and contributed to PR/design reviews focused on testability, maintainability, and safe rollout.

Software Engineer Intern **May – Aug. 2022**

Genpact | Toronto, ON

- Designed distributed index, storage, and query components and a GUI for accessing data in a distributed-computing environment.
- Improved runtimes through algorithm optimization, dynamic programming, and rewritten unit tests; developed AWS Data Lake/Lambda APIs to streamline dataflow and reduce input errors.

SELECTED SYSTEMS & GRAPHICS PROJECTS

Cycle-Accurate Game Boy Emulator **C++, Linux**

- Modeled CPU, RAM, bus, MMU, DMA, timers/interrupts, and LCD/PPU subsystems with explicit fetch–decode–execute and rendering state machines.
- Implemented the instruction set, memory map, interrupt behavior, and PPU mode transitions for deterministic cycle-level execution.
- Added frame stepping, breakpoints, structured diagnostics, and targeted unit tests; validated correctness against test ROMs.

Link's Room — Real-Time 3D Renderer **OpenGL/GLSL, Metal**

- Recreated Link's room as an interactive 3D scene with camera controls, dynamic lighting, Gerstner-wave effects, and tessellation shaders.
- Optimized draw calls and buffer updates for stable frame time and built a debug HUD for live shader-parameter tuning.
- Later ported the renderer to Metal in Xcode, adapting rendering abstractions and translating shaders from GLSL to Metal Shading Language while resolving differences across the API translation layer.

Apple Platform & Game Development **Objective-C, Swift, Metal, Xcode**

- Built an Objective-C news application in Xcode, establishing early hands-on experience with Apple application development and platform tooling.
- Created multiple platformer games in Swift for game jams, implementing complete playable experiences under short development timelines.
- Built small real-time 3D games directly with Metal, gaining practical experience with Apple's native graphics API and GPU rendering pipeline.