Edison Ling

Toronto, Canada · **Portfolio** · LinkedIn · FFmpeg Account

Education

University of Toronto **Expected Grad: June, 2028**
Bachelor of Applied Science in Computer Engineering + PEY (Co-op)
Relevant Coursework: Object Oriented Programming, Data Structures & Algorithms, Deep Learning, Computer Architecture, Operating Systems, Digital Logic, Project Management

Skills

Languages: C++, C, Python, RISC-V Assembly, Javascript, SQL, HTML/CSS, PHP, MATLAB, Verilog
Frameworks and Libraries: PyTorch, NumPy, scikit-learn, FFmpeg, Advanced Media Framework, DX12
Tools: Git/Github, VS Remote Debugger, Jira, VQAnalyzer, Perforce, Intel Quartus Prime, WinDBG

Experience

Software Developer Intern Toronto, Canada
Advanced Micro Devices (AMD) - Video Pipeline Team May 2025 - August 2026

- Won **AMD Spotlight Award** for implementing and upstreaming full hardware-accelerated D3D12 video pipeline support to FFmpeg 8.1, the open-source industry standard powering video infrastructure for platforms handling **over 1 billion daily active users** (YouTube, Netflix, Meta, etc.)
- Architected and deployed a self-initiated **InnerSource AI-agent automation toolkit** for autonomous driver build-test-analyze workflows, scaling cross-team adoption to **100+ engineers**
- Root-caused a decade-old **C-ABI vtable bug** in AMF breaking property API calls for public C consumers; diagnosed compiler-dependent vtable ordering defects and drove header fixes to mainline
- Shipped **10+** AMF encoder features, updates, and bug fixes directly to production codebases, while authoring comprehensive public-facing technical API documentation used by external developers
- Resolved driver conformance failures and restored failing **Windows HLK tests** by fixing multi-resolution support for the DX12 video-encode driver
- Led performance & quality triage, delivering **VMAF benchmarks** to attribute reported regressions to hardware limits or intended behavior, resolving end-user tickets and preventing unnecessary code reverts

Full-Stack Software Engineering Intern Toronto, Canada
Lingic Technologies Inc. May 2024 - August 2024

- Utilized **WAMP stack** to architect and code a multi-company website and database to record donors/transactions, **print CRA Charitable Receipts** and generate detailed reports based on a CPA client's needs, resulting in a simpler and more affordable tool for small Charities and Non-Profits
- Migrated an existing Windows SCM (Supply Chain Management) application from Visual C++ 6 (MS SQL), to Visual C++ 2022 through researching, debugging, and navigating deprecated API functions and libraries

Projects

Frame to Game *Python, PyTorch, scikit-learn, NumPy, FFmpeg*

- Designed and trained a **CNN from scratch** with four convolutional blocks and a flattened classification head that preserves HUD layout to identify six games or other content from a single frame
- Built a data pipeline to label gameplay clips, extract and clean frames, and split recordings by person and session; processed **683 clips into 11,055 cleaned frames**
- Evaluated generalization using leave-one-player-out validation across **14 players**, achieving **85.1% accuracy and 0.725 macro-F1** on 9,879 frames, compared with 39.7% accuracy and 0.319 macro-F1 for a colour-histogram baseline

[WIP] GPU Accelerated Screen Capture Utility *C++, IPC, FFmpeg, Git/Github, Qt, QML*

- Designing the C++ application backend **orchestrating capture and encode** through custom libobs wrapper and implementing playback, filesystem storage, and optional second-pass compression
- Building a capture engine wrapping libobs that **owns screen capture, replay buffer, and hardware encode**, and exposes them to the application over a localhost websocket for cross-platform use
- Planning a **Qt/QML frontend**, dynamically linked against the LGPL build exposing encode settings, hotkey bindings, video playback, and video clip library & management to users
- Future plans to: inject in-game overlay through graphics API, prioritize OS-native transport instead of websockets, in-application video editing, detect in-game events for auto-clipping via API hooks