Daris Chen

Los Angeles, CA | (562) 386-3975 | daris.chen@gmail.com | US Citizen
https://www.linkedin.com/in/darischen | https://github.com/darischen | https://darischen.com/

Education

University of California San Diego March 2026
Bachelor of Science (B.S.) in Computer Engineering *San Diego, CA*

Work Experience

Lead Full-Stack Engineer Jan 2026 – Jun 2026
Product Manager Accelerator *Remote*

- Constructed a RAG-based patent search and drafting assistant deployed end-to-end on Vercel, AWS EC2, and Supabase using Next.js, exposing FastAPI REST endpoints with Auth0 authentication to simplify intellectual property research for early-stage founders.
- Programmed a hybrid semantic search engine over 12,000+ USPTO patents combining OpenAI embeddings with PostgreSQL pgvector and HNSW indexing, achieving sub-100ms query latency for real-time prior-art discovery.
- Led a 14-member cross-functional team as lead architect and primary contributor, running Agile workflows in ClickUp and assigning work by strength and task atomicity to sustain a 100% no-blocker rate on daily standups across a six-week MVP.

Lead Full-Stack Engineer Aug 2025 – Mar 2026
Groundwork Books *San Diego, CA*

- Led an in-place migration of a production e-commerce storefront to Next.js and Vercel under live customer traffic, integrating the Square API so storefront stock reflects point-of-sale inventory without manual updates and modernizing the tech stack.
- Engineered a semantic search layer using Pinecone with integrated text embeddings, enabling topic and concept-based discovery across 4,000+ SKUs without requiring exact title matches.
- Architected a two-tier cache: a look-aside Redis reducing Square API latency by 90% (sub-200ms p99), and a client-side request coalescing hook batching 50+ concurrent inventory lookups into a 75 ms window API call with 2-minute TTL.

AI Researcher Feb 2025 – Mar 2026
UCSD Engineers for Exploration (E4E) FishSense *San Diego, CA*

- Rebuilt a published RRPN laser-detection network (ResNet-50 backbone) to a working training and inference state in Detectron2, pinning a brittle CUDA and PyTorch dependency matrix into a reproducible setup for the research group.
- Devised a dual-laser photogrammetry algorithm to extract biological metrics (fish length/mass) from 2D video feeds, achieving sub-5% error rates in pixel-to-centimeter conversion.

Projects

Career-Ops Pipeline | *Python, Node.js, Anthropic SDK, Claude Haiku, Playwright, Prompt Caching* Jan 2026 – Present

- Spearheaded three-tier conductor-worker-judge agentic pipeline for batch job applications, serializing Playwright extraction to prevent race conditions while parallel worker subagents generate outputs against a structured schema.
- Built a deterministic evaluation layer enforcing output rubrics through structural and cross-field checks with bounded retry loops (3 attempts, 5-min timeout), chosen over an LLM judge to eliminate scoring variance and per-validation token cost.
- Reduced token cost by 60.4% through Anthropic prompt caching of instructions, cv.md, and 4 resume PDFs across worker subagents, with two-phase keyword scoring auto-selecting the best resume variant on a 10-point lead.

Mini-Stockfish Chess Engine | *PyTorch, Torchscript, Cython, C++, NumPy, Parquet* Mar 2025 – May 2026

- Composed a hybrid chess engine pairing a HalfKP NNUE evaluation network with alpha-beta search featuring quiescence, null-move pruning, and late move reductions.
- Profiled the hot path and rewrote leaf node evaluation in Cython with C++ legal move generation and Zobrist hashing, cutting eval time from 1.1 ms to 0.25 ms (77.2%) to search deeper within the same per-move time budget.
- Rebuilt the training data pipeline around Parquet and disk-streamed chunking after the 329M-row dataset outgrew available RAM, cutting per-epoch time from 31 hours to 30 minutes without reducing dataset size or renting cloud compute.

HandVol | *Python, MediaPipe, GitHub Actions, NSIS, pycaw, faster-whisper* May 2026 - Jun 2026

- Engineered a hands-free Windows control system providing full mouse, media, and volume operation without physical input devices, targeting accessibility users and occupied-hands scenarios; shipped 12 public releases from a packaged installer.
- Designed a real-time gesture state machine with asymmetric debouncing (5-frame entry, 3-frame exit, 10-frame cooldown on edge-triggered actions) and EMA landmark smoothing, eliminating false triggers while holding inference within a 30fps CPU budget.
- Automated the release path in GitHub Actions: derives the next semantic version from tag history, compiles an NSIS installer, and publishes on merge to main, with serialized concurrency preventing version-tag races between rapid merges.

Technical Skills

Programming Languages: Python, JavaScript/TypeScript, C/C++, SQL, Cython, Java, Go, HTML/CSS, Verilog/SystemVerilog, ARM Assembly
AI & ML: RAG pipelines, LLM integration, prompt engineering, semantic search, vector databases, embeddings, computer vision, object detection, semantic segmentation, model evaluation, multi-agent orchestration, PyTorch, TensorFlow, CUDA, ROCm, cuDNN, Transformers, Detectron2, MediaPipe, Scikit-learn, NumPy, pandas, OpenAI, Claude, Hugging Face
Frameworks & Libraries: React, Next.js, Node.js, Express.js, FastAPI, Three.js, Svelte, Tailwind CSS, Zod, Prisma
Tools & Technologies: Git, GitHub Actions, Docker, Kubernetes, Linux, GCP, EC2, Vercel, gRPC, Playwright, Puppeteer, Jest, ESLint, Turbopack, Jupyter, Wandb, Figma, Clickup, Jira, Auth0, OpenCL, WebSpatial (Vision Pro)
Databases: PostgreSQL, pgvector, Pinecone, Upstash Redis, MongoDB, Firebase, Supabase, Parquet