

Ethan Huang

408-324-6947 | eyh002@ucsd.edu | [linkedin.com/in/ethanhuangpy](https://www.linkedin.com/in/ethanhuangpy) | github.com/hyatch

EDUCATION

University of California: San Diego

September 2025 – Expected June 2028

B.S Artificial Intelligence, Minor in Mathematics (GPA: 3.91)

La Jolla, CA

Relevant Coursework: Linear Algebra (MATH 18), Intro to Probability (MATH 180A), Algorithms (CSE 21)

EXPERIENCE

Undergraduate Research Assistant

February 2026 - Present

F2 Skeletal Lab

La Jolla, CA

- Designed a custom U-Net architecture in PyTorch to automate segmenting bones from adipose tissue in multiple myeloma patients
- Built a data processing pipeline to handle thousands of high-resolution volumetric images for self-supervised learning
- Leveraged Dragonfly software to visualize 3D skeletal structures and validate model segmentation accuracy.

Front of House

Seasonal

Paris Baguette

San Jose, CA

- Managed high-volume customer flow during peak hours, resolving inquiries regarding product ingredients and allergens

PROJECTS

LPPLS Algorithm Trader | *Python, QuantConnect*

October 2025 - February 2026

- Researched an algorithm that relies on approximating the **Log-Periodic Power Law Singularity** to trade volatile assets
- Optimized runtime to a 2 minute backtest (60x speedup) in Python with production-level code
- Published a paper to QuantConnect Strategies under mentorship of Triton Quantitative Trading

Interview Preparation | *Node.js, React*

October 2025

- Collaborated with 3 other builders in a Parallel Agent Hackathon to deliver a product that could curate relevant interview questions provided a job title and company name
- Developed using the Parallel Agent Search API to collect a cohesive prompt to Claude

Multimodal Inference AI | *Python, PyTorch*

October 2025 – Present

- Engineered a Vision-Language Model (VLM) from scratch by coupling a **SigLIP** encoder with a **Gemma** decoder
- Optimized autoregressive generation by implementing **KV Caching** and **Grouped Query Attention (GQA)** mechanisms
- Performed manual weight surgery to map and load pre-trained parameters from the official **PaliGemma** checkpoint

nanoGPT | *Python*

August 2025 – October 2025

- Rebuilt GPT-2 model (124M parameters) in PyTorch, improving training efficiency with vectorized operations
- Trained on the Tiny Shakespeare dataset in accordance with Karpathy's Zero to Hero series

TECHNICAL SKILLS

Languages: Python, SQL, Java, Node.js

Frameworks/Libraries: Pytorch, TensorFlow, Streamlit, transformers, FastAPI

Developer Tools: Git, VS Code, AWS, Cursor

Data/Visualization: NumPy, pandas, Matplotlib