Quan Tran

qmt1@rice.edu • https://github.com/QuanTran255 • quan-tran.com

EDUCATION

Rice University – B.S. and Master of Computer Science (5th-Year Program) **Houston, TX**

Aug 2022 – Expected May 2027 (GPA: 3.86/4.00)

Relevant Courses: Computer Vision, Quantum Computing, Software Methodology, Practical Machine Learning, Concurrent Program Design, Compiler Construction, Intro to Computer Systems, Tools and Models for Data Science, Probability and Statistics, Intro to Program Design, Reasoning about Algorithms, Linear Algebra, Fund. Computer Engineering

WORK EXPERIENCES

Rice University Computer Vision Lab - Research Intern – May 2025 – Present

- Built a full-stack annotation tool with JavaScript front end and Python backend (Flask, OpenCV) to evaluate rigid object tracking performance
- Enabled dynamic point and line selection, real-time track visualization, and seamless integration with models such as CoTracker3 and PIPs++
- Designed backend architecture for loading video data, managing tracker configuration, and multi-model support
- Benchmarked state-of-the-art trackers through both visual analysis and quantitative metrics to identify edge cases and performance bottlenecks
- Currently working on fine-tuning and training models for test-time optimization to improve rigid shape tracking on unseen data

HP/Poly - Software Engineer Intern – May 2024 – August 2024

- Developed an end-to-end Python framework using Zoom Rest API for automated testing, enabling Zoom Rooms configurations before and after testing to validate and certify software for Zoom
- Integrated the framework with the current Jenkins pipelines for minimal configurations when setting up and creating new test cases
- Collected and benchmarked Quality of Service data to evaluate live video performance in real-time test environments
- Refactored legacy codebase to double data retrieval speed and improve API efficiency

VNG Cloud - Software Engineer Intern – June 2023 – August 2023

- Built a scalable ML-based anomaly detection system to monitor cloud server traffic for millions of users, improving incident response time by 30%
- Evaluated time-series models (Holt-Winters, Prophet, ARIMA) and built an online-learning system that improved detection performance by 10% over baseline on seasonal traffic data
- Integrated a relational database to log and trigger user alerts on server traffic anomalies, with average delivery latency under 3 seconds
- Collaborated with notification team to deploy the alert system within the existing monitoring infrastructure

PROJECTS

Text-Based Game Editor and Engine – January 2025 – May 2025

- Led the UI/UX development team and coordinated design discussions to ensure seamless integration with backend systems
- Managed the database team by using CosmosDB and Redis to support scalable, real-time data storage
- Simulated 500 concurrent users during load testing to identify bottlenecks in CosmosDB and Redis-backed real-time operations
- Collaborated cross-team to integrate backend infrastructure with frontend editor features and conducted end-to-end testing

Messaging App – October 2024 – December 2024

- Built a real-time messaging framework with a custom NoSQL backend in Go, optimized for speed and scalability
- Designed intuitive user interfaces with the focus on accessibility using HTML, CSS, and Typescript frameworks
- Integrated database storage for users, message history, and conversation metadata, optimizing query efficiency to support fast message retrieval

OwlDB - Custom NoSQL Database – August 2024 – October 2024

- Developed an extensible NoSQL database with RESTful APIs and high-concurrency support, simulating real-world backend frameworks
- Designed and implemented robust APIs in Go to support seamless CRUD operations, integrating with frontend systems via structured endpoints
- Leveraged Go's concurrency features to high-volume requests efficiently
- Achieved 92% code coverage through extensive unit and integration testing, improving reliability

H.O.L.L.Y. – December 2023 – August 2024

- Created a productivity device that transforms any flat surfaces into interactive screens using computer vision
- Implemented lens correction algorithms for accurate hand gestures tracking
- Designed an intuitive user interface with projection mapping to navigate through apps using Kivy library
- Integrated large language models (LLMs), speech recognition, and text-to-speech technologies for a personalized assistant

SKILLS

Programming Languages: Python, Go, Java, C++, C, C#, TypeScript, JavaScript, Bash, R, Assembly, HTML/CSS

Frameworks & Libraries: React, TensorFlow, PyTorch, Pandas, NumPy

Cloud & DevOps: AWS, Azure, Docker, Jenkins, Git, CI/CD, Postman

Databases & Storage: NoSQL (MongoDB, Redis, CosmosDB), Relational (MySQL), GraphQL

Software Architecture & Systems: Object-Oriented Programming, Concurrent Programming, RESTful APIs, Model-View-Controller (MVC), Event-Driven Architecture, Client-Server Model

Data Science & Machine Learning: Regression Models, Time-Series Forecasting (Holt-Winters, Prophet, ARIMA), Anomaly Detection, NLP, Deep Learning, Regularization, Cross-Validation, RNNs, CNNs, Spark, Hadoop