# MIKE TRAN

✉ maitran.mt02@gmail.com · 🔗 linkedin.com/in/acorn27 · ⚙ github.com/Acorn27

## Education

**University of Texas at Arlington**                                                                 **Arlington, TX**
*M.S. in Computer Science*                                                          *Jan 2025 – Dec 2026 (Expected)*
- GPA: 4.0
- Relevant Coursework: Computer Vision, Distributed Systems, Advanced Topics in Computer Architecture

**University of Texas at Arlington**                                                                 **Arlington, TX**
*B.S. in Computer Science (Graduated)*                                                            *Aug 2022 – Dec 2024*
- GPA: 4.0
- Honors: Summa Cum Laude, Dean's List (All Semesters)

## Experience

**USDA-ARS**                                                                          **May 2025 – August 2025**
*Research Apprentice, Computer Vision*                                                               *Arlington, TX*
- Developed a computer vision pipeline to automatically detect corn ears, count individual kernels, and cluster kernels into rows.
- Reduced manual labeling time by 90% by building a preprocessing pipeline: initial detection via contour analysis, separation of touching kernels using convexity defects.
- Modified and fine-tuned a pre-trained Faster R-CNN model (ResNet-50 backbone), achieving 99% precision and 97% recall on images with up to 800 densely packed corn kernels.
- Implemented a custom DBSCAN-based algorithm to cluster kernels into spatial rows for structural analysis.

**University of Texas, Arlington - IDIR Lab**                                                 **May 2024 – Present**
*Graduate Research Assistant*                                                                        *Arlington, TX*
- Develop a full-stack dashboard for RDF database using Streamlit, Plotly, Cytoscape.js, and Mapbox to provide interactive graph visualizations.
- Containerized the complete application infrastructure with Docker for streamlined deployment and scalability.
- Integrated Large Language Models (LLMs) using LangChain to enable natural language querying of the graph database.
- Implemented a Retrieval-Augmented Generation (RAG) pipeline enhanced with vector search capabilities using MongoDB Atlas Vector Search.
- Co-authored a paper on Web engineering and LLMs, accepted at ICWE 2025.

**Nagarro Inc.**                                                                          **May 2023 – Aug 2023**
*Customer Research Intern*                                                                                 *Remote*
- Analyzed public brand sentiment across multiple social media channels using Brandwatch, compiling key insights for strategic reports.
- Designed and deployed real-time, interactive dashboards to present sentiment data and campaign perfor- mance to clients.
- Crafted stakeholder-facing reports using a storytelling approach to effectively communicate key insights

## Projects

SOCK-KG Dashboard | *Python, Streamlit, LangChain, Neo4j, Ollama, GraphDB, React JS, MongoDB, Docker*    **May 2024**
- Developed and deployed a full-stack web application for exploring and interacting with RDF graph databases.
- Integrated dynamic visualizations using Plotly, Cytoscape.js, and Mapbox GL to render real-time data from Neo4j and GraphDB.
- Containerized the entire architecture with Docker to streamline deployment, scaling, and system monitoring.

Pet Adoption System | *CockroachDB, gRPC, Python, Go, Docker*                                         **Oct 2024**
- A full-stack pet adoption system with front-end and back-end communication using gRPC.
- Designed a fault-tolerant database backend using CockroachDB with a 5-node cluster, allowing users to upload and store pet information (names, breed, pictures, etc.) through a Go-based web interface.
- Containerized the front end, back end, and database using Docker and orchestrated them with Docker Compose for simplified deployment.

Raft Consensus System | *Flask, Python, Docker, Go, HTML/CSS, gRPC*                                   **Aug 2024**
- Designed and implemented a Raft-based consensus protocol with five nodes in a distributed environment.
- Built a monitoring dashboard with Flask to track node communication, logs, and system states.
- Optimized inter-node communication using Python's asyncio for non-blocking operations, enhancing efficiency.