Arnav Jain

arnavjainpo@gmail.com | linkedin.com/in/arnavjainpro | arnavj.me | 972-983-8163

Education

University of Texas at Dallas — Dallas, TX
Bachelor of Science in Computer Science, Minor in Entrepreneurship — *May 2027*

Skills

Languages: Python, Java, JavaScript/TypeScript, SQL, C, C++, HTML/CSS, Bash, Rust, SwiftUI
Frameworks & Libraries: React, Node.js, Spring Boot, Django, Bootstrap, Pandas, TensorFlow, Tailwind, PyTorch
Tools & Platforms: Git, GitHub, Docker, JIRA, OpenAPI, CI/CD, VS Code, Figma, Agile, OWASP ZAP, Selenium
Cloud & DB: AWS (Lambda, ECS, EC2, S3, DynamoDB, SQS, CloudWatch, Terraform), Firebase, Supabase
AI: OpenAI, LangChain, Streamlit, RAG, Transformers (MTSIT, GRU, LSTM), NumPy, Hugging Face, DeepSpeed

Experience

The University of Texas at Dallas — Oct. 2024 – Present
Web Specialist @ UT Dallas Department of Strategic Marketing & Communications — *Dallas, TX*

- Conducted A/B tests on site hierarchy, microcopy, and CTA placement, reducing bounce rate from **40%** to **28%**
- Automated data collection for faculty directories and event listings by deploying Python web scrapers with Selenium, cutting manual effort by **90%**
- Integrated dynamic alt-text tagging for **100+** images across UTD Engineering sites, improving WCAG 2.1 accessibility compliance for screen reader users

UT Dallas Association for Computer Machinery — Sep. 2024 – Dec. 2024
Undergraduate Researcher — Pneumonia Segmentation in Chest X-Rays — *Dallas, TX*

- Fine-tuned **Meta's** (SAM) on labeled outlier regions, boosting mean IoU by **15%** on edge cases
- Applied CLAHE and RAPMED to **5K+** chest X-rays, improving pneumonia segmentation precision by **22%**
- Preprocessed and normalized **5K+** DICOM images with NumPy and OpenCV, reducing training latency by **35%**

Texas A&M Department of Computer Science — Jan. 2024 – May 2024
Undergraduate Researcher — Dr. Anthony Medellin, Hyperspectral Algorithms for 3D Mapping — *College Station, TX*

- Trained unsupervised autoencoders and DinoV2 models for 3D object tracking on **10K** hyperspectral image stacks
- Increased object boundary detection accuracy by **20%** using TPR and TNR evaluation on segmented outputs
- Optimized ViT pipeline to handle **200** spectral bands, improving multi-layer object recognition speed by **30%**

Projects

OneFace Dashboard | *AWS (Lambda, DynamoDB, CloudWatch, S3, SQS, Rekognition), Node.js* — Jun. 2025

- Built serverless facial recognition on AWS, processing **100+** scans at **99%** accuracy for real-time analysis
- Integrated React UI with AWS Lambda and S3 to enable seamless photo uploads and identity matching, reducing image processing latency to under **2 seconds** during live testing
- Designed a fault-tolerant recognition workflow processing over **1,000** facial scans by integrating SQS with event-driven Lambdas and DynamoDB Streams, guaranting delivery and consistent state across all user events

Voyager Magazine | *React, Tailwind CSS, WordPress PostgreSQL, CI/CD* — Feb. 2025

- Converted legacy HTML to modular React components with scoped CSS, reducing DOM reflow by **28%**
- Partnered with design teams to build responsive UI across breakpoints, achieving **100%** Lighthouse accessibility
- Developed interactive frontend components for features like Dean's Message, increasing engagement by **40%**

Aetheria iOS App | *SwiftUI, Firebase, Python, TensorFlow, FastAPI, AWS Lambda, Terraform* — Nov. 2024

- Simulated IoT inputs via Rust-based CLI and ingested through local WebSocket API to test model responsiveness
- Engineered sustainability scoring models in Python using scikit-learn (Linear Regression), with **90%** R^2 accuracy
- Provisioned AWS Lambda and API Gateway via Terraform, enabling **500ms** inference latency in local testing

QuillQuest CNN | *Python, NumPy, Pandas, SciPy, scikit-learn, Matplotlib, Jupyter Notebook* — Jul. 2024

- Built a feedforward neural network from scratch in Python using NumPy to classify handwritten digits across 10 output classes, achieving **95%** test accuracy
- Implemented one-hot encoding for labels and vectorized all operations across **784×m** input matrices, improving training time by **3x** over baseline loops
- Designed and visualized the gradient descent learning loop, updating weights and biases via partial derivatives of cost wrt parameters using matrix calculus