

GALDA JAYACHANDRA

New York | +1 (716) 789-1234

jgalda@buffalo.edu | linkedin.com/in/jayachandra | github.com/Jayachandragg

Professional Summary

Data Science professional with 1+ years of experience across **fintech and research**, specializing in **machine learning, computer vision, and data engineering**. Master's student in **Artificial Intelligence** with hands-on experience building **end-to-end AI pipelines** and **scalable data workflows**. Strong background in **Python, SQL**, deep learning frameworks, and **cloud-based tools**, with experience communicating technical findings through **reports, dashboards, and presentations**.

Education

State University of New York at Buffalo

Master of Science in Engineering Science, Artificial Intelligence

Buffalo, NY

Jan 2025 – Present

Coursework: Data Intensive Computing, Machine Learning, Computer Vision, Reinforcement Learning.

Technical Skills

Programming: C, C++, Python, SQL, Java, HTML, CSS, JavaScript, R, MATLAB

Cloud/Databases: AWS (ECS, Lambda, S3), GCP, Databricks, MySQL, SQL Server, PostgreSQL, MongoDB

Tools/Frameworks: Pandas, NumPy, SciPy, Tableau, Power BI, PyTorch, TensorFlow, OpenCV

Experience

University at Buffalo - Graduate Research Assistant

Buffalo, NY. Aug 2025 – Present

- Designed and implemented an **end-to-end multimodal AI pipeline (StrategyGen)** to analyze adult-child interaction strategies from educational videos, enabling scalable automation of previously manual research workflows.
- Built **Python-based ETL workflows** to convert unstructured **ELAN (.eaf)** annotations into **structured JSON** and generate **topic-level timestamps** using rule-based temporal aggregation.
- Integrated **LLM-based semantic validation** and a **Vision-Language Model (Qwen2.5-VL)** to verify topic alignment and classify visual context, improving reliability and reproducibility of extracted insights.

Digitap.AI - Data Science Intern

Bangalore, India. Jan 2024 – Aug 2024

- Collaborated with computer vision engineers to build a **real-time face detection system** using **YOLOv8 and OpenCV**, achieving **96.3% accuracy** with reduced inference time.
- Designed and optimized a **3D face alignment algorithm** using **NumPy matrix operations**, improving **verification success rates by 21%** and communicating trade-offs to stakeholders.
- Deployed and monitored **CNN-based inference pipelines** on **AWS EC2**, including a lightweight **SqueezeNet model (98% accuracy)** and a **YOLO-based blur detection pipeline**, ensuring **scalability**.

IIT Roorkee - Research Intern

Roorkee, India. Sep 2023 – Dec 2023

- Curated and annotated a 53-class plant disease dataset and benchmarked **CNN and Transformer architectures** using **PyTorch**.
- Implemented a **hybrid ensemble model** combining **DenseNet embeddings with XGBoost** to address **class imbalance** and optimize performance metrics.
- Analyzed results and presented comparative findings through **technical reports** and research discussions to guide **model selection**.

National University of Singapore (NUS) - Deep Learning Intern

Singapore. Jun 2022 – Jul 2022

- Trained and optimized a **VGG16-based fruit freshness classification model**, achieving **91.7% accuracy** using **normalization, regularization, and early stopping**.
- Ran systematic experiments and tracked **hyperparameters** and model performance in structured **Excel sheets**.
- Applied **Grad-CAM visualizations** to interpret model predictions and documented **explainability insights** through concise technical reports.

Projects

End-to-End E-Commerce Analytics Platform. Architected a **PostgreSQL** database with **normalized 3NF schemas** for 100k+ orders. Built robust **ETL pipelines** using **Python and SQL** for data cleansing. Developed interactive **Streamlit dashboards** and containerized the full stack using **Docker and Docker Compose** for reproducible deployment.

AI-Generated Image Detection. Trained a **beta-VAE (Variational Autoencoder)** on mixed real/synthetic images to learn **discriminative latent representations** using **PyTorch**. Applied **Logistic Regression** on **latent embeddings** to create an **interpretable classification boundary** for Generative AI detection.