Timothy Medewase

medewaset@gmail.com | linkedin.com/in/timothy-medewase | github.com/TimothyMedewase | timothy-medewase.com

Education

University of Montevallo Montevallo, AL

Honors Bachelor of Science in Computer Science, Minor in Mathematics *Aug. 2021 - May 2025*

Relevant Coursework: Calculus I, II, III, Discrete Mathematics, Complex Analysis, Numerical Analysis, Introduction to Computer Science, Computer Programming I (Python), Object Oriented Programming, Data Structures and Algorithms, Database Management Systems, Operating Systems, Computer Networks, Programming Languages, Software Engineering

Technical Skills

Languages: Java, TypeScript, JavaScript, SQL, Python, C++, HTML/CSS

Frameworks & Libraries: React, Node.js, Flask, Express.js, Spring Boot, Material-UI, TailwindCSS, BootStrap, Next.js, Pandas, NumPy, Matplotlib, Scikit-learn, SciPy, Tensorflow, PyTorch, LangChain, FastAPI

Tools & Platforms: Git, GitHub, VS Code, LaTeX, SQLite, PostgreSQL, Jupyter Notebook, Webpack, Figma, Google Cloud Platform (GCP), AWS, Azure

Concepts & Practices: Machine Learning, Deep Learning, Model Training & Evaluation, Data Preprocessing, Feature Engineering, Hyperparameter Tuning, Model Deployment (Batch & Realtime), Model Interpretability, Cross-Validation, Experimentation, Software Testing, Version Control, Agile Development, Internal Documentation, Continuous Learning

Experience

Software Development Intern May 2023 – July 2023

Solera, Inc. *Westlake, TX*

- Designed and delivered a full-stack analytics platform using **React** and **d3.js**, building modular, component-based data visualizations that improved user engagement by **30%** and reduced feature development time by **40%** through reusable UI logic
- Engineered high-performance backend data workflows with **Microsoft SQL Server**, optimizing stored procedures and query execution to reduce data retrieval latency by **50%** while ensuring consistency across services packaged in **Docker** containers
- Integrated RESTful API endpoints using **Axios** and automated CI/CD pipelines with **Bitbucket**, streamlining deployment, reducing integration bugs by **40%**, and increasing deployment frequency to support real-time, data-driven insights for stakeholders

Projects

Transformer Implementation | *Python, PyTorch, Deep Learning* | **Link** May 2025

- Implemented the full **Transformer** architecture from the ground up in PyTorch with 6-layer encoder-decoder stacks, **8 attention heads**, and **512-dimensional** embeddings following Vaswani et al.
- Developed modular components including multi-head attention, residual connections, and sinusoidal positional embeddings, enabling **configurable depth, width, and head count** via a base/big setup
- Optimized parallelized attention computations and tensor reshaping, improving training throughput by **45%** and enabling autoregressive decoding with $<$ **1 ms/token** inference latency

ExplainAI | *Next.js, FastAPI, FAISS, LangChain, OpenAI API* | **Link** April 2025

- Built a **retrieval augmented generation (RAG)** web app, improving information retrieval by **40%** over traditional methods with **Next.js** frontend and **Python-based FastAPI** backend
- Implemented secure file handling for **5 concurrent uploads** and integrated **Facebook AI Similarity Search (FAISS)** with **OpenAI API** to enable fast semantic search, cutting query latency by **60%**
- Developed modular **React** components and deployed APIs on **Heroku** with **99% uptime**, supporting a scalable user experience and handling **50%** more user traffic without performance loss

MNIST Digit Classification CNN | *Python, PyTorch, Computer Vision* | **Link** Aug 2025

- Built a high-accuracy **Convolutional Neural Network (CNN)** in PyTorch, achieving **99.41%** on the MNIST handwritten digit classification task
- Designed and trained a **3.3M-parameter CNN**, optimizing performance with data normalization and early stopping techniques to prevent overfitting and achieve peak validation accuracy within 14 epochs
- Interpreted the model's learning process and classification accuracy by visualizing its foundational learned filters and comparing predictions against actual test images

Housing Predictions Project | *Python, Scikit-learn, SciPy, NumPy, Matplotlib, Pandas* | **Link** November 2024

- Built a predictive system for California housing prices using a dataset of **20,000+ samples**, applying data preprocessing and feature engineering to improve data quality
- Optimized a **Random Forest regressor** with hyperparameter tuning, increasing model R^2 score from **0.75 to 0.85**, resulting in highly accurate price predictions