Md Al Amin

nsuer.alamin@gmail.com +880 164 104-9756 Dhaka, Bangladesh Md Al Amin
Md Al Amin

Education

BSc in Computer Science and Engineering, 01/2018 – present
North South University-CGPA-3.14(86%) Dhaka, Bangladesh

Skills

Programming Languages
Python , C, C++ , JavaScript

Web & Backend Frameworks:
- Django, FastAPI, HTML, CSS, Bootstrap, MySQL

Development & Collaboration Tools:
- Git, GitHub, Jupyter Notebook/Lab, Google Colab

Deployment & Prototyping:
- **Streamlit** for rapid AI application development.

Soft & Analytical Skills:
- Problem-solving skills and critical thinking,

AI & Machine Learning:

- ***Libraries:*** PyTorch, TensorFlow, scikit-learn, Pandas, NumPy, Matplotlib, Seaborn
- ***LLM & Model Tools:*** Familiar with **Langchain**, Langsmith, OpenAI APIs (GPT), Groq API, Ollama model, HuggingFace(Model and Datasets)
- **Techniques:** Retrieval-Augmented-Generation **(RAG)**, LoRA/QLoRA fine-tuning, prompt engineering.

Projects

1. AI-Based Adaptive Algebra Tutor [07/2024-present]
Description: Developed an AI-powered adaptive tutoring system that provides **personalized** algebra learning experiences. The system leverages **Microsoft's Phi-3 Mini 4k Instruct** model, fine-tuned with custom datasets, and is built using **Streamlit** for an interactive UI.
Key Features:
- **Interactive AI Tutoring:** Chat-based algebra learning with step-by-step solutions.
- **Adaptive Learning (Planned):** Personalized difficulty adjustments based on user performance.
- **Fine-tuned model:** 1287+ algebra problems with multiple student interaction variations.
- **Realistic Student Simulations:** Model trained on diverse student responses and misunderstandings.
- **User-Friendly Interface:** Built using **Streamlit** for an engaging learning experience.
- **Future Enhancements:** Memory tracking, advanced reasoning, gamification, and cloud deployment.

Technologies Used:
- **Pre-train Model:** Microsoft Phi-3 Mini 4k Instruct (Huggingfacec Hub)
- **Fine-Tuning:** LoRA (PEFT), Hugging Face PEFT Library.
- **Frameworks & Libraries:** PyTorch, Transformers, Streamlit, PEFT
- **(Future Enhancements):** LangChain, LangGraph, Retrieval-Augmented Generation (RAG)

Fine-Tuned Model on Hugging Face: alam1n/phi3-mini-algebra-tutor-v4

2. LangChain - Chat with Search
Description: Developed a chatbot application using **Streamlit** and LangChain that enables users to interact with a search-powered AI. The chatbot integrates with multiple external APIs (Arxiv, Wikipedia, DuckDuckGo) to fetch and provide relevant information in real time.
Key Features:
- **Web Search Integration:** Retrieves information from Arxiv, Wikipedia, and DuckDuckGo.
- **Interactive Chat Interface:** Built using Streamlit's st.chat_message.
- **Streamlit Callback Handler:** Provides real-time agent insights and actions.

Technologies Used:
- **Frameworks & Libraries:** Streamlit, LangChain
- **APIs, Tools & Model:** Arxiv API, Wikipedia API, DuckDuckGo API, **Groq API** (Llama3-8b-8192)

Live Website: https://search-engine-llm-hve88ggypuwbukbyzi4kq7.streamlit.app/

3. AI-Powered(RAG) Math Equation Solver with Document Embedding [07/2014-12/2024]
Description: Developed an AI-powered equation solver that leverages LLMs and vector embeddings to provide step-by-step mathematical solutions. The system integrates FAISS-based document retrieval, Hugging Face embeddings, and Groq's Gemma 2-9b model for accurate and context-aware responses.

Key Features:
- **AI-Powered Math Solver:** Uses **LLM and embeddings** to provide **step-by-step explanations** for math equations.
- **Retrieval-Augmented Generation (RAG):** Finds relevant **context from uploaded PDFs** to generate precise answers.
- **FAISS-Based Document Search:** Efficient **vector-based retrieval** of mathematical references.
- **Hugging Face Embeddings:** Converts text data into vector representations for **improved retrieval accuracy**.
- **Interactive UI:** Built with **Streamlit** for an easy-to-use interface.

Technologies Used:
- **Large Language Model (LLM):** Gemma 2-9b (via Groq API)
- **Vector Database:** FAISS (Facebook AI Similarity Search)
- **Embeddings Model:** Hugging Face (all-MiniLM-L6-v2)
- **Frameworks & Libraries:** LangChain, Streamlit
- **Document Processing:** PDF Loader (PyPDF)
- **Retrieval-Augmented Generation (RAG):** Context-based equation solving

4. Fine-Grained Feature Imitation for Efficient Object Detection Using Knowledge Distillation [07/2014-12/2014]
Description: Developed a lightweight **object detection model** using **Knowledge Distillation (KD)** to optimize performance on **resource-constrained devices**. The model applies a **fine-grained feature imitation** technique to enhance the accuracy of a smaller student model while significantly reducing parameters compared to state-of-the-art (SOTA) models.

Key Features:
- **Lightweight Object Detection:** Developed a **YOLOv5-based student model** with **1.78M parameters**, reducing size by **71%** while improving accuracy.
- **Fine-Grained Feature Imitation:** Focused on **region-specific knowledge transfer** to enhance object localization.
- **Superior Performance:** Achieved **mAP@50 of 0.707** and **mAP@[50:95] of 0.435**, outperforming the teacher model.
- **Efficient Deployment:** Enabled real-time execution on **low-end devices** without requiring specialized hardware.
- **Generalization Across Datasets:** Evaluated model on diverse datasets, achieving strong performance in **medical imaging, road safety, and agriculture domains**.

Technologies Used:
- **Deep Learning Frameworks:** PyTorch, YOLOv5
- **Machine Learning Techniques:** Knowledge Distillation (KD), Fine-Grained Feature Imitation
- **Optimization Methods:** Combined Imitation Loss Function (KL Divergence + MSE)
- **Datasets:** Pascal VOC, BCCD, Lemon Disease, Incorrect-Mask-2
- **Evaluation Metrics:** mAP@50, mAP@[50:95], Precision-Recall

5. Ecommerce Web Application [01/2024-06/2014]
Description: Developed a full-stack web application for an ecommerce platform where customers can browse and order products online.
Key Features:
- Product Browsing and Ordering: Implemented a user-friendly interface for customers to view and purchase products.
- Payment Integration: Integrated **SSLcommerz** payment gateway for secure online transactions.
- Admin Panel: Utilised Django's built-in admin panel for managing products, orders, and user accounts.

Technologies Used:
- Backend: **Django framework** for server-side logic and SQLite for database management.
- Frontend: HTML, CSS, and Bootstrap for responsive design and user interface.
- Future Enhancements: migrating the SQLite database to MySQL.