KARTHEEK MAKKENA

Indian Institute of Technology, Goa

Undergraduate, **Mathematics and Computing**
Address: Sattenapalli, Guntur, AP
E-mail: **makkena.kartheek.21033@iitgoa.ac.in**
Mobile: **+91 8919699339**
LinkedIn: **www.linkedin.com/in/makkena-kartheek**
GitHub: **github.com/kartheekmakkena**

Education

BTech, Mathematics and Computing, Indian Institute of Technology Goa	CGPA: **7.74/10**	2021 – 2025
Class 12, APBIE, Sri Chaitanya Junior College, Gudavalli, Vijayawada	Aggregate: **95.4%**	2019 – 2021

Experience

Software Engineer at Borqs Technologies (july 2025 - present)

- Assisted in implementing basic Android kernel security patches and SELinux rule updates.
- Supported vulnerability testing and debugging of kernel modules to improve device security.

AI/ML Research Intern at Hanyaa Auto Technologies (Jan 2025 – jun 2025)

- **Focused on LLMs and NLP**: Implemented multiple open-source LLMs for animated story script generation, explored fine-tuned models on Hugging Face, and also got familiarity with **NLP concepts, Transformer architecture, Multi-RAG, and LLM fine-tuning**.
- **Optimized structured story generation**: Designed prompt templates using **Pydantic with LangChain**, used **LangChain's structured output with Gemini APIs** to generate consistent scripts, enhancing **input quality for video and audio generation models**.
- **Developed a multi-stage script generation UI**: Built a Streamlit-based interface with **customizable duration options**, narration/style selectors, scene summary editors, and a detailed shot-by-shot refinement tool. Currently working on the deployment of multiple models

Projects

ChatBot with Multi-RAG Mar 2025 – Apr 2025

- Designed a chatbot using the concept of multi-RAG to answer user questions with **memory and conversation history**.
- Enabled support for inputs like **PDFs**, **images**, **YouTube video links**, direct **video files**, and **website URLs**.
- Used **Gemini API** for response generation, **Gemma 3** for image/video summarization, and **GoogleGenerativeAIEmbeddings** for retrieval.
- Built with **LangChain** and **LangGraph** for memory management, and deployed using **Streamlit**.

Real-Time Facial Emotion Recognition (Dec 2024 – Jan 2025)

- **Built a real-time emotion recognition system** using **3 CNN models**—LeNet5, VGG-16, and an enhanced **ResNet-18 with Depth-wise Separable Convolutions and SENet**.
- **Implemented face detection and classification** for **7 emotions** (angry, disgust, fear, happy, neutral, sad, surprise) using **PyTorch and OpenCV** to ensure real-time, efficient performance.
- **Boosted model accuracy by 36%**, improving from **61% (AlexNet)** to **83% (ResNet-18 with enhancements)** code Code

Image Captioning with Vision Transformer and GPT-2 (feb 2025 – mar 2025)

- Developed an **end-to-end image captioning model** by integrating a **Vision Transformer (ViT)** with a **GPT-2** language model in **PyTorch**, creating a hybrid architecture for sophisticated image-to-text generation.
- Implemented **memory-efficient training techniques** using **Parameter-Efficient Fine-Tuning (PEFT)** and **4-bit QLoRA**, significantly reducing computational requirements and enabling fine-tuning on consumer-grade GPUs.
- Engineered a **complete project pipeline** including data preprocessing with the **Flickr30k dataset**, a training script with periodic evaluation, and a **standalone inference script** to generate captions from novel images via URL.

Built a GPT-style Transformer Language Model from Scratch (MAY 2025)

- **Built and trained** a **decoder-only Transformer model** (like GPT) using **PyTorch** with around **121 million parameters**, focused on tasks like sentence continuation and grammar correction. code Code
- Used the **BookCorpus** dataset with **11+ million text chunks** and trained the model for **20 epochs** on an **NVIDIA RTX 4090 GPU**, reducing training loss from **4.2 to 1.1** and generating high-quality English sentences.
- Adjusted key hyperparameters like `d_model = 512`, `num_layers = 6`, `context window = 256`, and `batch_size = 32` to ensure efficient training within GPU memory limits.
- Created a **modular training setup** with support for **checkpoint saving**, **learning rate scheduling**, and **batch-wise evaluation**, making the system easy to debug, resume, and scale.

Skills

Programming Skills:	Python, C++, C, SQL
Libraries and Frameworks:	NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow, PyTorch, OpenCV, Hugging Face Transformers, LangChain, LangGraph
Tools and Platforms:	Git, GitHub, Linux, Jupyter Notebook, VS Code, Ollama
Relevant Coursework:	Data Structures & Algorithms, Machine Learning, Deep Learning, Linear Algebra, Probability & Statistics, Numerical Analysis