KIRAN RAGHAVENDRA

Irvine, CA 92612

+1 714-971-7414 ● *kraghav1@uci.edu* ● *in/kiranraghavendra789/* ● *KiranRaghavendra1248*

EDUCATION

University of California, Irvine **Expected Dec. 2024**
Masters in Computer Science (GPA : 4/4) *Irvine, CA*
Courses: Intro to AI, Operating Systems, Transaction Processing & Distributed Data Management, Advanced Data Structures

JSS Science and Technology University **Graduated July 2022**
Bachelor of Engineering in Computer Science (GPA : 9.42/10) *Mysore, India*
Courses: Data Structures, Analysis and Design of Algorithms, Object Oriented Programming in Java, Database Systems, Software Engineering

TECHNICAL SKILLS

Languages & Frameworks : Python, C, C++, Javascript, Dart, SQL, FastAPI, Node.js, Express.js, Flask, jQuery, Flutter
Tools & Tech Stack: Pytorch, Tensorflow, Keras, Langchain, Huggingface, CUDA, ONNX, Pandas, Numpy, Matplotlib, OpenCV, PIL, NLTK

EXPERIENCE

ESRI **June 2024 - Present**
Software Engineer Intern, Software Products | Python, Finetuning LLMs, LLM Evaluation & Bias, Huggingface, Langchain, RAG

- Enhanced search algorithm using **semantic search** feature and pushed to production by converting model to **ONNX**
- Implemented semantic search by evaluating finetuned **sentence embedding** models **all-mini-LM-V6-0.2** and **paraphrase-mini-LM-V6-0.2** on geospatial dataset
- Created LLM Assistant feature using **Mistral-7b** for **intent classification** and workflow automation using **Few Shot Learning** for Information Retrieval and developed framework for **topic based evaluation** of semantic search using **Mistral 7B LLM as Judge**
- Proposed method to remove bias and unwanted association from semantic search model manually without finetuning

Western Digital **Jan. 2022 - Aug. 2023**
Embedded Software Engineer, OptiNAND | Python, C, Express, HuggingFace, Node, Express, Git, Agile, Jira, SDLC

- Spearheaded the development of features for **read error handling**, **data caching**, and **thermal data persistence** for NAND Device
- Developed methodologies to identify and isolate **performance bottlenecks** in multi-module interactive system resulting in a **30% reduction** in write-read verification latency. Proposed log summarization tool using LLMs for summarizing regression cycle logs
- Developed summarization tool using **Node, Express, HuggingFace** and created API routes for remote calls resulting in **40%** decrease in bug burndown time. Debugged and fixed production bugs and mentored 10+ interns advocating clean coding practices

Google Developer Student Clubs **Sept. 2020 - July 2022**
Machine Learning Intern | Python, Pytorch, Attention Mechanism, Flask

- Spearheaded development of chatbot using **Encoder-Decoder transformer** and integrated with college website using **Flask** API calls to the chatbot. Mentored 100+ students as the lead instructor for Deep Learning Bootcamp attended by over 200+ students

MACHINE LEARNING PROJECTS

Abstractive Text Summarization - Code *Pytorch, Transformers, Huggingface, LSTM, GloVe, BERT*

- Implemented **Encoder-Decoder** Transformer **from scratch** with Multi-Head Attention, Masked Attention, Encoder-Decoder Cross Attention and Positional Encoding. Evaluated and compared performance of **transformer** against **encoder-decoder Bi-LSTM** architecture
- Compared context-independent **GloVe embeddings** and context-dependent **BERT embeddings** as token embeddings for Encoder-Decoder Bi-LSTM, and trained on **Cross Entropy Loss** with **Adam Optimizer & Cosine Annealing** on BBC News dataset

Multi Class Image Semantic Segmentation - Code *Pytorch, Image Segmentation, Quantization*

- Performed **Multi Class Semantic Segmentation** on Human Face Dataset(Lapa Dataset)
- Implemented **UNet** and **Depth Residual Seperable UNet** Architecture **from scratch** from research paper with depth seperable convolution layers and skip connections b/w model layers. Created hybrid loss function of **BCE Loss** and **Dice Coefficient Loss.**
- Evaluated & compared performance for multi class semantic segmentation and implemented **dynamic quantization**, reducing model size by 30% and inference latency by 5%.

Image Captioning - Code *Pytorch, Resnet101, Transformers, Quantization*

- Used Resnet101 Deep CNN model as **Image Encoder** and GPT-like Decoder only **Transformer** with **Masked Attention** and **Cross Attention** built **from scratch** for image captioning
- **Trained** custom **encoder-decoder architecture** on MS COCO dataset using Adam Optimizer and Cosine Annealing
- Performed dynamic **quantization** of nn.Linear, nn.Conv2d, nn.ReLU layers in architecture to reduce model latency and size

Face Detection and Recognition using One Shot Learning - Code *Pytorch, InceptionResnetV1, Siamese Network*

- Implemented and **trained InceptionResnetV1** on custom dataset to generate face embeddings for **multi-class classification** on Pytorch framework, attaining 96% test accuracy on custom dataset
- Utilized MTCNN for face detection and **One-Shot Learning** for training of model to generate embeddings for face images
- Evaluated and compared accuracy against **Siamese Network** trained by reducing **contrastive loss** to generate feature vectors for face data