Santhosh Sankar

sankar.s@northeastern.edu | linkedin.com/in/santhosh-sankar | 857-277-9864
santhosh-sankar.github.io/portfolio.github.io | github.com/Santhosh-Sankar

EDUCATION

Master of Science in Robotics May 2023
Northeastern University, Boston, MA **CGPA:** 3.838
Coursework: Deep Learning, Reinforcement Learning and Sequential Decision Making, Pattern Recognition and Computer Vision, Mobile Robotics, Robot Sensing and Navigation, Robot Science and Systems

Bachelor of Engineering in Mechanical Engineering May 2020
Anna University, Chennai, India

TECHNICAL SKILLS

Programming and Query Languages: C++, Python, MATLAB, SQL
Machine Learning and Parallel Programming: TensorFlow, PyTorch, Amazon SageMaker, CUDA
Software Libraries: OpenCV, PCL, NumPy, SciPy, Matplotlib, pandas, scikit-learn
Software Tools and Operating Systems: AWS, ROS, Git, Docker, Nsight Systems, Linux, Windows

PROJECTS

Machine Translation with Transformer using AWS Oct 2023 - Nov 2023

- Built a Transformer-based sequence-to-sequence model to translate German sentences to English using **Python** and **PyTorch**, with a custom module to compute multi-head attention.
- Trained, deployed, and assessed the Transformer model with the Multi30K dataset stored in **Amazon S3** using **Amazon SageMaker** and achieved a high BLEU score of **78%** on the test dataset.

Chatbot using sequence-to-sequence Transformer July 2023 - August 2023

- Designed an end-to-end chatbot based on sequence-to-sequence Transformer with **Python** and **TensorFlow** and implemented a data preprocessing pipeline to extract and process the input data for training.
- Trained the model with Kaggle TPUs on Cornell movie dialog corpus with over 200,000 conversations and improved the BLEU score by **3%** with a word tokenizer after significant hyperparameter tuning.

Parallel image processing with CUDA May 2023 - Jun 2023

- Developed a baseline code with **OpenCV** in **C++** for processing input images and frames from the live video feed with Gaussian, Sobel, magnitude, quantization, and cartoonization filters.
- Parallelized the sequential baseline code with **CUDA** and profiled it with **Nsight Systems**, optimizing memory access and accelerating performance by at least **75%**, relative to the baseline.

Siamese networks with attention for large-scale landmark retrieval Mar 2023 - Apr 2023

- Devised a data preprocessing pipeline in **Python** that yielded new landmark pairs and similarity scores from 1.6 million images in the Google Landmark Dataset (GLDv2) while training to prevent overfitting.
- Designed and trained Siamese networks in **TensorFlow** on GPU cluster, pairing ResNet-101 with spatial, channel, CBAM, and SE attention modules as subnetworks to retrieve images similar to a given image.
- Achieved **9.46%** higher mean Average Precision (mAP) than the baseline classifier with the SE module and improved the mAP further by **4.4%** with an ensemble of spatial, SE, and CBAM attention modules.

Pedestrian tracker and counter using YOLOv3 and DeepSORT Oct 2022 - Dec 2022

- Constructed YOLOv3 with **Python** and **TensorFlow** utilizing the Darknet-53 architecture to perform pedestrian detection on video frames and incorporated weights pre-trained on the COCO dataset.
- Integrated the YOLOv3 object detector with the DeepSORT algorithm for pedestrian tracking and counting and visualized the pedestrian paths over the most recent 50 frames with a **Python** script.
- Evaluated DeepSORT with YOLOv3 and Faster RCNN on MOT16 benchmark, with YOLOv3 attaining **6%** higher tracking accuracy (MOTA) and **3%** higher tracking precision (MOTP) over Faster RCNN.

Performance comparison of RL algorithms in Super Mario Bros Oct 2022 - Dec 2022

- Developed and trained DQN, DDQN, and PPO deep reinforcement learning agents to complete one level of Super Mario Bros using **Python** and **PyTorch** and assessed the performance over 1000 episodes.
- Achieved **1%** and **30%** higher success rates, **2.4%** and **22.7%** higher average returns, and **4.1%** and **34.9%** lower mean steps per episode with DDQN and PPO agents respectively, over the DQN agent.