Linkedin

Karthik Raj Ramachandrappa

📞 (631) 428 4257

✉ karthik4293@gmail.com   ⓘ **Karthik4293.me**   🏠 Stony Brook, NY11790

ACADEMIC QUALIFICATIONS

Stony Brook University,   **New York, NY**   December 2018
Master of Science in Computer Engineering,
- *Courses:* Computer Architecture, Operating Systems, Parallel and Heterogeneous Computing, Machine Learning;
- *Teaching Assistant* for ECE 358 Computer Vision

University Visveswaraya College of Engineering,   **Bangalore, India**   June 2010
Bachelor of Engineering

WORK EXPERIENCE

Graduate Research Assistant   **Stony Brook University**, NY   March 2018
- Research interests: GPU Systems and Architectures, applications of high performance computing to aid Artificial Intelligence.
- Advisor: Prof. Yuefan Deng (Department of Applied Mathematics and statistics).

Network Engineer   **Ericsson,** India   August 2017
- Responsibilities: integration and troubleshooting of nodes(DTN) in the Radio Access Network (3G/4G LTE).
- *Technologies used:* Mo-Shell Scripting, Python.
- Received multiple client appreciations (PAT awards) for developing parser tools which helped analyze the error patterns in the network configuration files and accelerated the productivity of the team up to 200 man-hours.

PROJECTS: **GITHUB** ◯

Gesture Recognition System using Deep Convolutional Neural Networks (C++, CUDA, Python):   **(Ongoing)**
- The project aims to implement a Gesture recognition system, intended for UAV and search and rescue bots. Implementing a cooperative model architecture to maximize the performance and sensitivity of the system.

Design and Modelling of SPU, A Dual-issue Multimedia Processor based on CELL Architecture: **Spring 2018**
- The Synergistic Processor Unit (SPU) is the integral part of CELL and is designed to accelerate the media and streaming workloads.
- Implemented a SPU-lite multimedia processor in System Verilog. Behavioral design of fundamental modules such as Instruction Line Buffer(ILB), Instruction Decoder, dual pipelined ALU, Data forwarding using forwarding Macro and Local Store needs to be implemented.
- Handling structural hazards and data hazards (RAW, WAR and WAW) & implementing the static branch prediction.
- Dense matrix multiplications and image reconstructions in cross domains were performed to scale the performance of the processor.

Simulation of CPU Scheduling Algorithms (C++, Linux):   **Summer 2018**
- Implemented Non-Preemptive Priority (NP), Preemptive Priority (PP), Shortest Job First (SJF), Shortest-Remaining-Time-First (SRTF) scheduling algorithms which determine the instruction flow in the execution pipeline.

Parallelized N-body Simulations using CUDA: **Summer 2018**
- Simulated interaction of N particles in a 2-dimensional space, described by the gravitational dynamics on multiple platforms. A simulation time of 2.779s was recorded on CUDA platform (GTX 1050) at a body count of 30720 over an iteration count of 20. This was roughly 6 times faster than in OpenMP.

Hardware Generation Tool for a Configurable Neural Network (C++, Neural Networks):   **Fall 2017**
- A hardware generation tool was implemented in C++ to generate the hardware description script (in System Verilog) for three layered neural networks. The design was capable of generating the hardware script for varying port selection and parallelism. The generated scripts were compiled and synthesized on Design Compiler library.

System Programming (C++, Linux, Multithreading):   **Spring 2018**
- Implemented a multi-threaded caching server using a producer-consumer and readers-writers locking pattern to handle concurrency issues.
- Developed shared-memory pages for processes to communicate through memory. Implemented kernel threads and built spinlocks and mutexes to synchronize access among them.

TECHNICAL EXPERTISE

Programming languages	C, C++ (C++ 11, STL, Multithreading), Python(Numpy, Matplotlib, Tensorflow), Java
Fields of interest	High Performance Computing, Computer Vision, Machine learning, Deep learning
Tools and libraries	CUDA, OpenMP, MPI, OpenCV, MATLAB, Git Version Control, Linux, GPFS