Sai Kopparthi

https://sai.cs.ucdavis.edu
kopparthisai0014@gmail.com | 201.887.5931 | Open for Relocation.

EDUCATION

UC DAVIS
MS IN COMPUTER SCIENCE
Sept-2018 to Jan-2021 (Expected)
Cum. GPA: 3.814 / 4.0

UC DAVIS
BS IN COMPUTER SCIENCE
Sept-2017 to June-2018
Cum. GPA: 3.66 / 4.0

AMRITA UNIVERSITY
BS IN COMPUTER SCIENCE
May-2014 to May-2018
Cum. GPA: 9.02 / 10.0
Dean's List for 4 semesters

LINKS

Github:// **sai-kopparthi**
LinkedIn:// **sai-kopparthi**

SKILLS

PROGRAMMING
C++ • Java • C • Javascript

SCRIPTING LANGUAGES
Bash•Python •Matlab•R

ML AND AI PACKAGES
TensorFlow•Pytorch•sklearn
Pandas•SciPy • Keras

WEB DEVELOPMENT
HTML • CSS • MySQL•Express.js
Angular.js • Spring boot • Hibernate
Docker•Docker Swarm• Ngnix

BIGDATA TOOLS
Apache Hive •Apache cassandra •Pig
Hadoop •MangoDB •Spark

COURSEWORK

Deep Learning
Computer Vision
Computer Graphics
Empirical Software Engineering
Statistical Computing(Big Data)
Design and Analysis of Algorithms
5G: Cellular Networks NR principles
Visual Recognition
Distributed Databases
(Research Asst)

TEACHING ASSISTANT

Artificial Intelligence - Spring 2020
Computer Graphics - Winter 2020
Analysis of Algorithm - Spring 2019
Object Oriented Prog - Winter 2019
Software Engineering - Fall 2019

INDUSTRY EXPERIENCE

CEPHEID | SOFTWARE ENGINEERING INTERN
June - Sep 2020 | Sunnyvale, CA

- Helped roughly 350 Cepheid Clinical research Scientists in their efforts of COVID-19 testing by adding new REST end points to the Assay Bundle SPA .
- Utilized OAuth2.0 with OpenID Connect to add Single Sign-On feature and maintain active directory of all recent login info of the users.
- Reduced user lookup time from 5 seconds to less than a second.
- Employed Agile Software development lifecycle model to complete the tasks.
- Developed well organized and robust code following design patterns in Java using Spring boot, MySQL, Hibernate, Oauth2.0, Angular.js.

STRATOVAN | SOFTWARE ENGINEERING INTERN
Jan 2018- Apr 2018 | Davis, CA

- Implemented Forward back projection for 3D volume visualization on Human skull dataset and further extracted 2D slice images using C++ and openGL.
- Successfully implemented most common filters, such as Ram-Lak, Cosine ,Image noise class, including Gaussian, Poisson, and Salt-pepper noise in C++.
- Compared the output of existing Matlab code (CT scan) in the market with my C++ implementation and reported a low MSE value of 0.02 .

RESEARCH

DISTRIBUTED SYSTEMS | GRADUATE RESEARCH ASSISTANT

- Decreased latency time for replicating data across 10+ global data warehouses by 50% by designing Geo-PBFT, a multi-leader PBFT protocol.
- Implemented a Fault-Tolerant Distributed Key-Value Store with throughput up to 5×10^5 ops/ sec using asynchronous and concurrent programming in C++.

BIG DATA TECHNOLOGIES | UNDERGRAD RESEARCH ASSIST

- Established a 23 node Hadoop cluster to count the number of unique string matches in a 2TB nucleotides combinations data stored on HDFS.
- Reduced search time from 14hours to 3hours by writing map-reduce task.

PROJECT EXPERIENCE

COMPUTER GRAPHICS

- Implemented basic Raytracer, 3D transformations on objects using OpenGL API's in C++.

EMPIRICAL SOFTWARE ENGINEERING

- Scroll across 1000 NPM packages on Github to Calculate triviality, techinical lag and transitive dependency and their correlation with each other using Python,PostgreSQL ,R and Docker.

COMPUTER VISION : IMAGE ANALYSIS

- Developed algorithms for image re-sizing, finding the center of circle with fixed radius and creation of visual vocabulary, retrieving images with similar patch using Matlab.

DEEP LEARNING : VISUAL RECOGNITION, NLP

- Implemented CNN using Keras to identify ethnicity/race of a person achieved 84.5% accuracy with over 23,000 face images from the UTKFace dataset .
- Restructured comments in the form of structured comments similar to java doc comments which is made up of block tags like @param, @return.

PARALLEL PROCESSING AND COMPUTATION

- Developed Gradient Descend and logistic Regression algorithms and optimize their gradient function so that it will work on sparse data as well.
- Developed multiprocessing code which allow multiple process to take part in performing computation on the algorithm.