Avinash Shanker

https://github.com/avinash273
avinash.shanker@mavs.uta.edu
https://www.linkedin.com/in/avishanker/

Skills

Languages: Python, Java, JavaScript, SQL, Shell Scripting, C++
Web Technologies: HTML5, CSS, PHP, Android, iOS, Spring, NodeJS, ReactJS, Django, PostgreSQL, Apache Spark, Hadoop
Libraries and Tools: Keras, TensorFlow, Google Colab, OpenCV, Open MPI, JUnit, AWS, GitHub, Jenkins, Docker, Splunk

Education

Master of Science in Computer Science **Aug 2018 – May 2020**
University of Texas Arlington GPA: 3.50/4.0
Dual Specialization: Artificial Intelligence *(Neural Network, Computer Vision),* Database Systems *(Hadoop, Spark, AWS)*

Experience

Software Engineer, TESCO *(Java, JavaScript, Spring, MongoDB, PL-SQL, ProC)* **Jul 2015 – Jul 2018**

- Developed RESTful Webservices using Java8 EE, Spring Framework, Tomcat and MongoDB as an agile team. Created queries on JSON files for generating real-time ordering status on a ReactJS UI with response time of 50 milliseconds
- Co-Developed and tested web APIs, for migrating product and ordering data from live legacy Oracle Retail Merchandising System (RMS) to AWS cloud in 5 phases. Decreased the error rate in Stock on Hand (SOH) data by 5% by creating order auto-reprocessing batch for manual replenishment orders and then synchronizing it with a API call
- Designed a multithreaded algorithm for performance tuning, existing auto-ranging functionality in RMS using ProC and PL-SQL. Created n+1 threads for n warehouses, it assists in ranging over 500,000 products to newly setup warehouses. Reduced job runtime of 6hrs by 40% to 2.4hrs, bottleneck was in data flow to Store line system
- Developed android application for creating customized product offers as notifications to users when they are in the vicinity of a Tesco Store in Hackathon'17. Won 2nd place for my implementation with minimal dependence on GPS

Projects

Parallel Sieve of Eratosthenes for Prime Generation *(Open MPI, C)* **Mar 2020 – Apr 2020**

- Implemented Sieve algorithm for prime number generation as a parallel program on Frontera(5th) Supercomputer.
- Generated primes under 10^9 using 32 parallel processors in 0.074908 sec, improved the performance of algorithm by making greedy choice in assigning the initial prime number for each processor to mark multiples and counting array indexes to obtain primes. The original algorithm took 0.085918 sec for 10^9 primes making it 12.7% faster

Google Inception Convolution Neural Network *(Python, TensorFlow, Keras, OpenCV)* **Dec 2019 – Jan 2020**

- Designed 3 layer densely connected CNN for categorizing images to 10 categories training my model on CIFAR10 dataset, optimized quality using Hebbian network, visualized graphs on Tensorboard for loss functions. This deep learning network generated Mean Average Precision (mAP) of 38.021%, 3% better previous model

Deep Privacy Face De-Identification Using GAN *(Python, TensorFlow, Keras, OpenCV)* **Nov 2019 – Jan 2020**

- Ensure anonymity for faces in public images, used Generative Adversarial Network (GAN) and Autoencoders to generate a highly realistic anonymous face mask. Used Kalman filter for smoothing boundaries of frame and MTCNN face detection algorithm to reconstruct superimposed face on Google Colab decreasing reconstruction loss to 30.02%

Analysis on Drug Consumption Dataset *(Python, Scikit, Pandas, NumPy, Seaborn)* **Jan 2019 – Mar 2019**

- Performed K-Means and Hierarchical Agglomerative clustering algorithm from scratch to train, predict & contrast performance of both models using confusion matrix (precision & recall) and classify drugs used by different age groups
- Visualized the data with 92.3% accuracy in K-means and 79% in agglomerative clustering. Improved the K-means accuracy by 4.2% isolating elements that are very far above the cluster's variance threshold value

Honors and Awards

- Awarded biannual Tesco Star Performer twice in a span of 3 years among a team of 22 members for creating a UI for order reject re-processing and performance tuning of auto-ranging batch within 4 days for going live in Poland stores.
- Successfully organized annual university coding completion HackUTA'19 which had 400 participants. Conducted Coding Basics in Python and tools workshop for encouraging participants with minimal coding knowledge.