Education

Master of Science,
Computer Science;
UNC-Chapel Hill;
Jan. 2022-Dec. 2022

Bachelor of Science,
Computer Science (Primary),
Statistics(Secondary);
UNC-Chapel Hill;
Aug 2018-Dec.2021

GRE:330

Dean's List: All semesters

Major GPA: 3.91

CGPA: 3.87

Clubs: Carolina Analytics and Data Science

Technical Skills

Languages: Java, Python, C, R, JavaScript, CSS, HTML, SQL, Octave, MATLAB, C, Prolog;

Frame Works: MongoDB, Pytorch, TensorFlow, React.js;

OS: Windows, Linux;

Other tools: Git, GitHub, Tableau, AMPL;

Course Work

Operating Systems, Computer Security, Machine Learning, Distributed Systems, Algorithms and Analysis, Web Programming, Data Structures, Internet Services and Protocols.

Vikram Thirumalai

My LinkedIn My GitHub vikram14@live.unc.edu +1 (201) 554 8285

MS Student | UNC Chapel Hill Aug. 2018 – Dec. 2022

Work Experience

Amazon Music, Software Development Engineer Intern *May. – Aug. 2022*

Resume Playlist Listening Experience

- Introduced a feature that resumes a customer's playlist listening experience on Alexa and Amazon Music when "Alexa, Play Music/Playlist" is requested.
- Used **AWS Lambda**, **SQS** and other internal services to set up the infrastructure to generate and store resume points for a customer.
- Improved UPL (user perceived latency) by ~15-20 ms when resuming customer experience by making asynchronous/parallel **RPC** calls to relevant services using the **CompletableFuture API** in **Java**.

Happiest Minds Technologies, Machine Learning Intern *May – Aug. 2021*

CCTV surveillance automation and Content Based Image Retrieval

- Created a use-case for **action recognition** models in automating CCTV surveillance. Developed a model using **3D Convolutional Neural Networks** (3D CNN) on **Pytorch**. Achieved 90% accuracy in detecting anomalous behavior from videos. Minimized the processing time per video to 1.5 seconds on a CPU.
- Built a **Content Based Image Retrieval System** to recommend and recover images of similar e-commerce products. System utilized a **Convolutional Autoencoder** backbone for feature extraction coded on Python. Used **ANNOY** to build an index and retrieve images with an 81% top-5-acccuracy. Reduced retrieval time per image by 12.5 % on a single-core CPU.

UNC Department of Computer Science, Teaching Assistant *Jan. - May 2022.*

- Maintained and added front-end and back-end features to a **clojure** codebase for a class management website that allows students to take programming tests and assignments online and receive feedback immediately.
- Created **API** endpoints in clojure to enable different levels of user privilege and used **POSTGRESQL** to manage the data required by the application.
- Collaborated with the Professor to grade, evaluate and assist with creation of homework assignments.

Research Experience

UNC Department of Computer Science, Research Assistant *Oct. 2019.- Present*

Source Code Similarity Detection

- Implemented two **software similarity detection algorithms** in **python** using the concepts of **Normalized Compression Distance** and **Winnowing** to assist in identifying Intellectual property theft in universities and the corporate world. Achieved performance similar to services provided by GradeScope and helped verify 6 possible cases of plagiarism for a COMP course.
- Collaborated with a team of researchers to build an **educational machine learning software** that aids students with assignments by recommending potential solutions to errors gathered from similar past data. Reduced amount manual intervention required by enabling faster debugging of programs.

Personal Projects (My GitHub)

Melanoma Detection *May 2020*

- Conducted a survey on the performance of **CNNs** and **SVMs** on the task of **detecting malignant melanoma** from images of moles. Developed the candidate models using **Pytorch** and **Sklearn**. The CNN out-performed the SVM and achieved an accuracy of 86.52%.

Online Multiplayer Chess *Jun. 2020*

- Built a website that enables users to create an account, view their game history and play chess with friends online. Used **React** and **Semantic UI** on the frontend and **Express, MySQL** to develop the **REST API** and database. Facilitated online play with the help of **socket.io**.

Distributed Map Reducer to Find Mutual Friends on Facebook *Feb. 2021*

- Used **Java's RMI** (Remote Method Invocation) package and **Threads** to write a distributed software that leverages the **Map Reduce**, **Model View Controller**, and **Factory** design patterns to find mutual friends among a group of Facebook users.