Timothy Cai

timcai.tyc@gmail.com | timcai.dev | 832-951-7889

Education

- **Texas A&M University** — College Station, TX
  Bachelor of Science in Computer Science; ***GPA: 3.82*** — *Graduating May 2024*
  - **Relevant Coursework:** Data Structures and Algorithms, Competitive Programming (**C++**), Probability, Discrete Mathematics, Analysis of Algorithms, Formal Languages and Automata, Computer Organization, Distributed Networks and Systems

Experience

- **Frogslayer** — College Station, TX
  Junior Software Developer — *June 2022 - Present*
  - Working at a software consulting firm, with projects focusing on **Angular** with **TypeScript** and **.NET Core**
  - Designed and reformatted data transfer system to reduce observable customer delay by up to **70%**
  - Responsible for tight turnaround in fast paced web development environment, comfortable with ambiguity
  - Implemented numerous features affecting everything from UI to customer interaction with **RxJS** and **React**
- **Aggie Research Program** — College Station, TX
  Researcher — *Sept. 2021 - Aug. 2022*
  - Designed multi-faceted non-deterministic model simulating society to observe effects of media
  - Implemented mentioned model in **NetLogo** and processed over 30 million data points through **R** and **Python**
  - Worked closely with graduate students and professors on a multi-disciplinary team to design statistical models
- **Senseye** — Austin, TX
  Software Engineer & Machine Learning Intern — *Jun. 2023 - Aug. 2023*
  - Designed and implemented internal **Python** library to autonomously pull data from **AWS Athena/Glue S3**
  - Intended to be lightweight and exist on the cloud as a piece of a larger machine learning pipeline using **Sagemaker**
  - Able to manipulate data through **ffmpeg** commands and can isolate keyframes within videos to probe machine learning models for accuracy

Projects

- **Found in Translation** — *Tools Used: Python, JavaScript, Slack Bolt, Flask, Node.js, Co:Here, Pinecone, Azure*
  - **Overall Winner** at the Cohere 2023 Semantic Search Hackathon
  - Intelligent Slack bot that can **semantic search** for messages across languages as well as analyze emotions in a server
  - Utilized **Flask** to train **CoHere** models on the **Google GoEmotion dataset**, with storage through **Pinecone**, and everything hosted on **Azure**
  - Developed interface with **Slack Bolt API**, and improving time efficiency by up to **50%** by designing a **multithreaded** system of back-end calls,
- **Mock Shell** — *Tools Used: C++, PHP*
  - Mock Linux shell accepting multiple commands and flags with **multithreading** capability
  - Utilized **POSIX** standard to implement and maintain low level, efficient C++ code
  - Developed and designed systems for piping, file I/O redirection, background processes, with the option of multithreading for each
- **makea.horse** — *Tools Used: Javascript, HTML, CSS*
  - Fun website displaying generated image at a certain time created for a Speed Development Hackathon
  - Written in a single Javascript file as emphasis on **monolithic** architecture and deployed through **Netlify**
- **The Galactic Algorithm** — *Tools Used: Python, tensorflow, keras, PIL*
  - Heuristic algorithm that was designed to descramble a large image of mixed quadrants, written for 2023 TAMU Datathon
  - Designed three separate heuristics which combined into the end algorithm, with a final accuracy of **97%**
  - Utilized augmented **Nearest Bodies** algorithm for object detection, and further reduced inaccuracy by matching edges
- **Personal Server** — *Tools Used: C++, Twilio, Pinecone*
  - Simple multithreaded server that reads/writes various inputs regarding personal calendar, habits, and data through **Twilio SMS**
  - Designed with **multithreading** in mind, has capacity of handling single requests up to **1.2 GB**, with data stored on **Pinecone**
- **Small Distributed Social Network** — *Tools Used: C++, gRPC, glog, cmake*
  - Small distributed implementation of a social network service utilizing Chubby lock system and server clusters, and built with **grpc**. Designed to be scalable, fault tolerant, and highly available where failures are handled transparently to users.
  - Server clusters split into master / slave, with **Chubby** running as coordinator, close to **99%** uptime.

Technical Skills

- **Languages**: *(Proficient):* Python, Java, C++, Typescript, HTML, CSS *(Familiar):* Go, Scheme, SQL, Rust
- **Technologies**: Pinecone, Pandas, React, Angular, .NET Core, Node.js, Firebase, Heroku, gRPC, glog, cmake, Azure, AWS