Alan Mu

Email: xm24@illinois.edu
Mobile: +1-2173053796

Education

- **University of Illinois Urbana Champaign** — Urbana, IL
  B.S. in Computer Science & Geography & GIS; GPA: 4.0/4.0 — *Aug. 2022 – May. 2026*
  - **Coursework**: Discrete Structure, Data Structure, Computer Architecture, Algorithms & Models of Computation, Computational Photography, Machine Learning, Computer Systems, Database Systems, Programming Languages and Compilers, Probability and Statistics, Spatial Analysis, Geographic Information Systems.

Programming Skills

Languages: C++, C, Python, Java, R, SQL, JavaScript, HTML/CSS
Technologies: Linux, Django, React, MySQL, SQLite, Docker, REST APIs, MongoDB, NodeJS, Spring Boot, Redis

Experience

- **ZTE Corporation** — Nanjing, China
  Software Engineer Intern — *May 2024 - August 2024*
  - **Mock Test for Fault Diagnosis API**: Defined **Mock API endpoints**, and implemented asynchronous functions to handle API calls and responses. Created **callback functions** to validate request payloads and return predefined JSON responses. Created **Text2APIAgent** and **ExecApiAgent** to execute the fault diagnosis and validate mock API interactions.
  - **SelectFunction in Text2APIAgent Class**: Engineered an advanced API filtering mechanism within the Text2APIAgent class, utilizing user input to dynamically select relevant APIs. Invoked **Large Language Model** with a prepared prompt using a predefined template. Verified and validated responses from the LLM. Parsed and stored the selected API name and parameters for further processing.
  - **Globalization and Workflow Optimization**: Expanded API accessibility by adding English descriptions to JSON data across the API suite, ensuring seamless international usage. Refined the fault diagnosis workflow by embedding prompt placements into the automation pipeline, reducing latency and improving overall process efficiency.
- **UIUC PURE Research** — Urbana, IL
  Data Science Research Assistant — *Aug 2023 - Jan 2024*
  - **Project Design and Data Preprocessing**: Developed a robust data analysis framework using **R**, identifying key dietary components (vitamins, minerals, fatty acids, sweeteners, and carbohydrates) that correlate with anxiety levels. Applied **data cleaning** and **feature engineering techniques** to ensure data quality and relevance for analysis.
  - **PCA Analysis**: Conducted **PCA Analysis**, creating **correlation matrices** and **biplot visualizations** with **ggplot2** to reveal correlation and cluster patterns.
  - **Multilinear Analysis and ANOVA Analysis**: Conducted **multilinear analysis** and **ANOVA**. Applied **Tukey-Kramer Test**, **Kruskal-Wallis Test** to evaluate statistical significance and p-values, drawing conclusions about group differences.

Projects

- **Image Quilting Algorithm Implementation**:
  - **Overlapping Patches**: Synthesized textures by randomly sampling square patches from a given sample. Selected starting patch randomly. Iteratively sampled and filled overlapping patches based on the **sum of squared differences(SSD)** of the overlapping regions of the existing and sampled patch.
  - **Seam Finding Algorithm**: Incorporated seam finding to remove edge artifacts from the overlapping patches by finding the **min-cost path** from the left to right side of the patch.
- **EasyShare**:
  - **Backend Development**: Developed a robust file and message sharing web application using **Node.js** and **Express.js**, handling real-time requests with **asynchronous functions** and scalable **RESTful API endpoints**.
  - **Frontend Integration**: Designed a responsive user interface with **HTML5**, **CSS3**, and **JavaScript**, incorporating **Bootstrap** for layout and styling. Implementing client-side file selection, validation, and messaging features.
  - **File Handling**: Utilized **Multer** for file uploads with custom storage. Integrated **Fetch API** for seamless client-server communication.
  - **Cloud Deployment**: Deployed the application using **AWS Elastic Beanstalk**, automating deployment pipelines with **AWS CLI** and **Docker** for containerization. Configured environment variables for secure key management and dynamic port allocation.
- **Memorandum App**:
  - **Login Page, Registration Page, and Navigation Bar**: Used **React** to implement a login and registration page. Designed and integrated a responsive navigation bar with **React Router**. Utilized React components for efficient rendering and state management.
  - **Task Management System**: Developed a task management feature using **SQL** for data storage and retrieval, supporting task prioritization through a well-structured data schema. Managed task states and CRUD operations in **React**, optimizing performance through **React's Context API** for global state management.