Huan Xu

6086987552 | hxu296@wisc.edu
Madison, WI
hxu296.github.io/home | https://github.com/hxu296

Education

University of Wisconsin-Madison Dec 2022
Computer Sciences, Mathematics, Statistics B.S. Madison, WI

- Cumulative GPA : 3.94 / 4.0
- Accomplished Coursework: Algorithms, Operating Systems, Computer Networks, Database Management Systems, Computer Graphics, Artificial Intelligence, Regression Analysis, Probability Theory, Discrete Mathematics, Linear Algebra, Calculus, etc.

Research Experience

Synthetic Training Jan 2021 - Present
Research Assistant Supervised by Prof. Yin Li Madison, WI

- Spearheaded the development of a graphic rendering pipeline that infers human mesh from 3D human joints and shape, and samples dense depth maps with respect to customized camera views by using **OpenGL**, **Pytorch**, and **Scipy**. This pipeline serves as the synthetic data generator for 2 pose estimation research projects.
- Deployed, optimized, and benchmarked a 3D human pose lifting network on Jeston Nano and achieved SOTA accuracy in real-time using **TensorRT**.
- Used both **Tensorflow** and Pytorch to conduct 2 ablation experiments with topics of skeleton-based action recognition and per-joint uncertainty estimation for a conference paper targeting ECCV 2022.

Human Pose Estimation Sep 2020 - Jan 2021
Research Volunteer

- Set up 6 human pose estimation methods on Unix-based OS with **Docker** and **Anaconda**, analyzed the Procrustes-Aligned Mean Per Joint Position Error from 13460 common human poses with **Pandas**, achieved a 33.67% PA-MPJPE reduction by integrating estimations from model-based and model-free SOTAs with **Numpy**, and visualized the result with **Matplotlib**.
- Wrote research proposal, conducted proof of concept, and presented preliminary results of integrating model-based and model-free methods for human pose estimation to an assistant professor from CS department's Computer Vision research group.
- Joined the research group and made this project a continued effort with the collaboration of a CS Ph.D. student under the supervision of Prof. Yin Li.

Work Experience

Teradata Jun 2021 - Aug 2021
Data Engineer Intern Beijing, China

- Worked in the Bank of China team and was responsible for both improving the ETL routine of the Data Warehouse using **Perl** and **Bash script** and performing custom data extraction tasks using **Teradata-SQL** and **DSQL** for **fraud detection**.
- Communicated with fraud detection analyst, analyzed their demand from a data perspective, and performed downstream data extraction tasks 3 times a week.
- Led the communication, demand analysis, and SQL development for a complex data extraction task requiring joining more than 20 tables across 2 databases. Wrote optimized, peer-reviewed SQL and communicated with the operation team to ship the data in time.

Computer Science Learning Center, University of Wisconsin-Madison Jan 2021 - Jun 2021
Tutor Madison, WI

- Hosted one-on-one virtual tutoring sessions for 3 hours each week to address students' questions about Machine Organization, Data Structure, and Object-Oriented Programming.
- Showed students how to use the GDB as well as GUI Debuggers to get more information about a run-time error.
- Introduced students to automated formatting tools that oblige Google's Java and C/C++ Style Guide.

Project Experience

Hongyuan Displays Sep 2021

- Developed a WeChat mini-program that allows users to search, browse, and share office supply products from Ningbo Hongyuan Electronic Technology Co. using **JavaScript**, **React.js**, **Bootstrap**, **Django,** and **PostgreSQL.**
- Paired the mini-program with a background management website where administrators can create, update, and delete products constructed with **TypeScript**, **Node.js**, **Redis**, **MikroORM**, and **MongoDB.**
- Deployed the mini-program on the WeChat official server and the background management website on a remote CentOS server using **Apache** and **WSGI**.

Stock Drop Notifier Dec 2020

- Built a customizable stock notifier for Newegg and BestBuy with a Telegram command-line interface that allows users to personalize search filters, fire up notifiers, and receive notifications all through 1 Telegram bot account by using **Kafka**, **Scrapy**, **Selenium**, **BeautifulSoup**, **Regex**, **Requests**, and the **Python-Telegram-Bot API**.
- Deployed the bot to a Raspberry Pi 4 and purchased 3 Nvidia 30 series GPUs in 1 month with its assistance.

Personal Information

- **Tech Stack:** React.js, Vue.js, Bootstrap, Ajax, JWT, Redis, Django, Flask, Node.js, PostgreSQL, MongoDB, Kafka, Spark
- **Programming Languages:** Python3, TypeScript, Javascript, Java, C/C++
- **Awards:** Dean's List 2019-2021