Xianda (Bryce) Xu

412-418-1784 xiandax@cs.cmu.edu linkedin.com/in/brycexu/ www.xiandaxu.info

Education

Carnegie Mellon University | School of Computer Science **Pittsburgh, PA**
Master of Science in Artificial Intelligence and Innovation *May 2023*

University of Toronto **Toronto, Canada**
Master of Engineering in Computer Engineering (remote, 2 semesters), GPA: 4.0/4.0 *Jan. 2021 - Aug. 2021*

University of Electronic Science and Technology of China **Chengdu, China**
Bachelor of Engineering in Computer Science, GPA: 3.99/4.0 *May 2020*

Skills

Programming Languages: Python, Java, SQL, C/C++, Objective-C, HTML, CSS, JavaScript, MATLAB, Shell

Frameworks: PyTorch, Tensorflow, Scikit-Learn, AWS, Django, Flask, NodeJS, Bootstrap, Vue

Databases: MySQL, Oracle, PostgreSQL, Redis

Work Experience

Ericsson **Oct. 2020 – May 2021**
Machine Learning Engineer Intern *Nanjing, China*

- Collaborated with a team of 5 to develop a smart trouble-shooting system that could offer repair recommendations for the radio products with an accuracy of **75%** and was estimated to save over **$10 million** per year for the company.
- Trained a **YOLO-v4** model using **Tensorflow** to detect missing components or defects in the PCBs and created an **AdaBoost** framework to learn from global repairing data stored in the **Oracle** database.
- Built an interactive website based on **Django** and **Vue** that facilitated receiving feedback from onsite operators to continuously revise the system; received group recognition for its interface design.

Mitacs Globalink **May 2019 – Sep. 2019**
Research Assistant Intern *Montreal, Canada*

- Studied binarized neural networks without pooling or fixed filter size by combining techniques of multiple resolutions and differentiable architecture search, improving more than **3%** on CIFAR-10 compared with manual architecture design.
- Devised a computing kernel for network binarization using **C++** and **CUDA** that supported fast Xnor-Bitcount operations and could be wrapped to be deployed on PyTorch; accelerated inference of the binarized model by about **3 times** on GPU and about **4.5 times** on CPU.

Selected Projects

Cloud Computing (Python, HTML, JavaScript, SQL) | *University of Toronto* **Spring 2021**

- Completed an interactive website that supported online face mask detection; utilized **Flask** as the back-end and **Bootstrap** as the front-end; deployed the website on **AWS EC2** with data stored in **AWS S3**.
- Designed a social media using **Django** and **Vue** where people can post cartoon blogs and socialize; deployed it on **AWS Lambda**; leveraged **AWS DynomaDB** to store user information; created a model to predict the long-term cost.

Fundamental Computer System Designs (C) | *Carnegie Mellon University* **Summer 2020**

- Implemented a **cache simulator** following LRU replacement policy and write-back & write-allocate regulation.
- Completed a **dynamic memory allocator** (malloc, free, realloc, calloc) and enhanced its performance in both time and space by applying a segregated free list and a better fit algorithm.
- Finished a **tiny linux shell** that could deal with process control, I/O redirection and signal handling.
- Built a **proxy simulator** that supported web caching and handling of multiple concurrent requests.

Website Development (Java, JavaScript, SQL) | *University of Electronic Science and Technology of China* **Spring 2019**

- Designed a social media called IdeaShare using **Java** based on the framework with Spring, SpringMVC and MyBatis.
- Utilized **Bootstrap** as the front-end and **MySQL** as the database; deployed the website on the Ali server.

Mobile Development (Objective-C, C) | *University of Electronic Science and Technology of China* **Spring 2018**

- Led a team of 4 to devise a smart controller that helped people control their favorite functions throughout their homes; won Excellent Award in National Innovation Competition for college students.
- Designed a user-interactive iOS application utilizing **Objective-C** and configured communication between the **BLE** device and the software in the Peripheral & Central modes.