

Kate Xiao

607-262-1417 | [xx257.github.io](https://github.com/xx257) | xx257@cornell.edu

Education

Cornell University, M.Eng Electrical and Computer Engineering, GPA 3.6 – **2017-2018**

Relevant Courses: Machine Learning (Python), OO Programming & Data Structure (Java), Algorithms, Computer Systems (C++), Database (SQL), OS, Computer Vision, Embedded OS (Unix)

Northwest University (China), BS Electrical Engineering, GPA 3.6 – **2013-2017**

Industry Experience

SOFTWARE TEAM MEMBER, WAVE-IOT; SAN FRANCISCO, CA – WINTER 2018-PRESENT

Working on the next generation IoT platform. Developed full-stack features such as showing real-time updating temperature graphs from remote sensors using **Go**, **GraphQL**, **React/JavaScript**. Also designed new sensor message structures in **Protobuf**, as well as modified the **SQL** schema to reflect the new proto changes. Finally, helped with setting up **docker-compose** script for easy deployment on AWS.

SOFTWARE ENGINEERING INTERN, CADENCE DESIGN SYSTEMS; SAN JOSE, CA – SUMMER 2018

Improved runtime memory footprint of the **Tempus** Timing Analysis Tool by dynamically packing long integers into narrower bit-width formats. Implemented the **data compression algorithm** in **Python**, and later converted to **C++** for performance. Also wrote Python/Bash scripts to automate the data pre-processing step in order to generate analytical graphs.

Projects

SIMULATED SELF-DRIVING SYSTEM [RASPBERRY PI] [C, PYTHON, SCIKIT, OPENCV]

An autonomous RC-controlled car that drives freely in a mocked traffic environment with curvy roads and traffic signs. The system uses **Artificial Neural Networks** to analyze the road condition in the forward-facing camera footage. It also detects traffic signs in the footage with **OpenCV**. It also samples forward distance using an ultrasonic sensor. By fusing multiple sources of information together, the RC car **achieved pseudo-autonomy** and was able to reach its destination while obeying the traffic rules.

HANDWRITING RECOGNITION SYSTEM [C++, PYTHON, TENSORFLOW]

A software system that was capable of recognizing MNIST handwriting images with certain noise added. The system was first implemented from scratch in C++ using Brute-Force K-Nearest Neighbors method. Then a better **TensorFlow CNN** version of it was written in Python and it reached 97.3% accuracy.

TREE RING COUNTER [PYTHON, OPENCV]

An image filter that processes images of tree rings and gives prediction about the age of the tree. The filter was developed around an algorithm that counts the number of rings on that tree using **OpenCV** and preprocessed tree ring images.