Shihe Wang

Tel: (434) 995-8650 / E-mail: qvw9pv@virginia.edu

EDUCATION

University of Virginia Charlottesville, VA, US
MCS in Computer Science. *Sept 2022 - May 2024 (Expected)*
GPA: 3.9/4.0
Relevant Coursework: Internet Scale Applications, PL for Web Applications, Human-Computer Interaction, Autonomous Mobile Robots, Machine Learning in Image Analysis, Software Analysis, Software Security via Program analysis, Compilers, Introduction to VLSI Design

University of Michigan - Shanghai Jiao Tong University Joint Institute Shanghai, China
B.S.E. in Electrical and Computer Engineering; Minor in Computer Science *Sept 2018 - Aug 2022*
Relevant Coursework: Data Structures and Algorithms, Algorithms, Operating Systems, Computer Organization, Artificial Intelligence, Discrete Maths, Probabilistic Methods in Eng.

SKILLS

Programming:	Python, C++, Matlab, C, SQL, Java, Scala, Go, OCaml, HTML, CSS, JavaScript, PHP
Tools/Frameworks:	TensorFlow, PyTorch, React, Springboot, MongoDB, AWS
Software/OS:	Mathematica, Multisim, PSpice, Origin, Arduino, Blender&BlenderProc, Linux, ROS, Docker

PROJECTS

Online Shopping System
This project built online shopping distribute system with Spring boot and Spring MVC, handle Prime Day Sale event up to 100K QPS traffic.
- Built a set of RESTful API for online shopping service models, serve customer requests through microservices integrated with filtering, sorting and pagination.
- Processing Online shopping CRUD operation with MyBatis, used MySQL as persistent storage.
- Implemented distributed UUID Generator service with SnowFlake, provide sequence Unique ID for distributed persistent storage.
- Designed and Implemented Prime Day Sale even, which can handle up to 100K QPS burst traffic under Jmeter stress test.
- Improved Product Search experience via Inverted Index search(ElasticSearch), speed up 90% for search experience compared with MySQL query.
- Achieve peak load shifting for Prime Day Sale event with RocketMQ, handle burst traffic requests as async jobs, reduced 95% invalid request reach to MYSQL.
- Utilized Distributed Lock with Redis and Lua scriptcompletely prevent overselling happens for Prime day sale events.

Keyfinder
The course project KeyFinder aimed to make keyboard shortcuts more accessible to computer users as well as provide users the viability to customize keyboard shortcuts.
- Designed a functioning KeyFinders prototype which will give hints regarding available shortcuts when binding key is pressed, e.g, ctrl.
- Conducted a survey to participants to evaluate the KeyFinder and received positive feedback.

Brain MRI Images Classification
This course project utilized GhostNet to classify healthy brain MRI image from the ones with Alzheimer's disease.
- Studied the architecture of GhostNet, modified the network to produce binary labelling results.
- Preprocessed the dataset, retrained the model on brain MRI image and reached an accuracy of 74% on the test dataset.

WORK EXPERIENCE

Sensetime Shanghai, China
Software Engineer Intern *Dec 2021 - Mar 2022*
Path Planning Based on Neural Network and Hybrid A* Search
This internship experimented on improving path planning task with a neural network.
- Reproduced the work of Neural A* paper, in which the result of a deep network is provided as a guidance map to A* search to improve the optimality and time.
- Implemented hybrid A* search module which is applicable to the automatic driving scene and allows for forward/ backward prorogation.
- Trained the neural network on maze occupancy map dataset, fed the result into the hybrid A* module and yielded valid non-holonomic path.

XYZ Robotics Shanghai, China
Software Engineer Intern *Dec 2020 - Mar 2021*
Bin-picking Robot Object Detection Based on Synthetic Depth Mask RCNN
This internship developed a pipeline to train a model on synthetic data for bin-picking task.
- Studied the industrial scene and camera setup to determine the synthetic dataset composition.
- Implemented a synthetic depth image generation pipeline built upon blenderProc using AWS; Utilized the multi-thread techniques and GPU to promote the data generation efficiency.
- Trained the SD MRCNN model with the generated dataset and adjusted the image generation pipeline accordingly.
- Used Docker to test the software environment; Deployed the trained model and performed stress tests on objects with diverse image information.