Wenqing Zong

(+44) 07713 918152
wenqing.zong98@gmail.com
zongwenqing.com
Github: WenqingZong

Education

2021 - 2022 **Imperial College London**,
MSc.Advanced Computing.
Final Grade 70+

2018 - 2021 **The University of Manchester**,
BSc.Artificial Intelligence.
Top 10% Graduate

2017 - 2018 **INTO Manchester**,
Foundation Year.
Top 10% Graduate with A*A*A* in Maths, Further Maths and Physics

Experience

Sep 2023 - Present **Software Engineer - Full Time**, *Emotech*, London, UK, Rust + Python.
- Served as a key engineer on the ASR (Automatic Speech Recognition) team.
- Responsible for backend API development, transforming AI models trained by the research team into market-ready products, employing innovative libraries and algorithms to minimize API response times.
- Responsible for developing testing and visualization tools to monitor model performance.
- Responsible for Azure SaaS integration to market our products.
- Contributed to company's open source Silero VAD (Voice Activity Detection) library.
- The product earned high praise from both colleagues and clients.

Jan 2023 - Aug 2023 **Software Engineer - Full Time**, *Codethink*, Manchester, UK, C.
- Developed and contributed to Free and Open-Source Software to automate quality assurance testing.
- Quality Assurance Deamon: Provides remote interaction with a device in place of having to physically interact with it. It's a remote control for test rigs. [Link to Blog Post]
- Testing in a Box: Integrates GitLab server/runner, OpenQA webUI/worker, and Q.A.D. into one box, making it an all-in-one solution for fully automated hardware testing. On-going project.

Jun 2021 - Aug 2021 **Machine Learning Engineer - Internship**, *AgCIM Research Centre*, Guangzhou, China, Python.
- Utilized Pytorch to develop an image-based rural area hazard detection system with core functionalities such as object segmentation and road category classification.
- Improved the accuracy of the road width calculation module in City Information Modeling (CIM) by incorporating the MegaDepth network.

Jun 2022 - Aug 2022 **Network Support Engineer - Part Time**, *Sobey*, London.
- Monitored the status of over 200 server clusters and ensure the proper functioning of the database.
- Regularly performs maintenance on PCs for non-technical colleagues.

Skills

Languages Proficient in Python, Rust, Java, C, familiar with JavaScript and C++

Frameworks Tokio, Ndarray, PyTorch, PyTest, Flask, Spring, OpenGL, JUnit

Utilities WebAssembly, Linux, Docker, Ansible, Git, Markdown, LaTeX, CI/CD, Nginx, AWS, Azure

Communication English(fluent), Chinese(mother language)

Prizes

Oct 2024 Internet Computer's 5000 USDC prize, awarded at Encode London 2024 Hackathon

Projects

Oct 2024 - Present
Python Debug Library, [PyPI], Python.
- Easily inspects values of variables and expressions in a human-readable way.
- Supports primitive types, user-defined classes, nested objects and recursive objects.
- Can be used as a drop-in replacement of python's print() function.

Aug 2024 - Present
Silero Voice Activity Detection Library, [GitHub], Rust.
- Pure Rust implementation of Silero VAD model and algorithm. Support running on any hardware.
- Easy-to-use batch and streaming interface, plus all the utilities you'd expect to see in an audio project.
- In progress: Async interface.

Jun 2023 - Jul 2023
Rust Octree Library, [Github], Rust.
- A highly optimised Octree implementation.
- Capable of tracking dynamic objects in the environment.
- Easy to integrate into existing codebase.

Mar 2023 - May 2023
Brainf*ck Interpreter in Rust, [Github], Rust.
- Developed a highly optimized interpreter for the Brainf*ck language using Rust.
- Implemented a modern and user-friendly command-line interface.
- Included extensive debugging messages for static checking and runtime errors.
- Achieved high test coverage and fully documented the project.

May 2022 - Sep 2022
Unsupervised Domain Adaptation on Medical Images, *Dr. Matthew Williams, Imperial College London*, [Github], PyTorch.
- Devised a novel method for addressing the domain shift problem, enabling a model trained on one dataset to adapt and fit to another dataset without significant loss in performance.
- The proposed novel method offers two key benefits:
  1. Source-Free: Model adaptation does not require the source dataset, which enhances cross-institutional collaboration efficiency and addresses data privacy.
  2. Supports Various Network Backbones: The novel method is compatible with all neural network architectures, without any special requirements.
- Demonstrated the efficacy of the proposed method on BraTS2021 dataset, achieving comparable performance with the state-of-the-art approach.

Jan 2022 - Mar 2022
Robot Learning and Control in Maze Environment, *Self-motivated*, PyTorch.
- Implemented several algorithms to teach a robot how to solve a maze.
- Traditional algorithm: Cross Entropy Method. Continuously adjusted the covariance matrix to make the action distribution approach the known optimal solution.
- Machine Learning: Trained a model to learn the non-linear environment and later used in Model Predictive Control algorithm.
- Behavioural Cloning. Trained a model to mimic how humans navigate in the maze. Implemented the DAgger algorithm to improve the model's performance while reducing the amount of data needed.

Oct 2020 - Apr 2021
Procedural Terrain Generation for Video Game Development, *Dr Ke Chen, The University of Manchester*, PyTorch, C#, Unity.
- Utilized Perlin noise to procedurally generate terrains for modern RPG games and simulated hydraulic erosion process to enhance playability.
- Employed Spatial GAN model to generate realistic terrain for a flight simulation game.
- Completed as a First Class Final Year project for my undergraduate degree.

Oct 2020 - Dec 2020
MCTS Board Game AI, *Team*, [Github], Java.
- Collaborated with a team of four to develop an AI bot to play a board game, Kalah.
- Implemented a bot based on Monte Carlo Tree Search with some improvements such as Early Payout Termination and MCTS-Minimax hybrid.
- Our bot beats 37 bots submitted by other teams (51 in total) in a tournament.