Peiyao Tao

+1 (425) 624-5652 | taopeiyao@outlook.com
Bellevue, WA

EDUCATION

Northeastern University *09/2023 - Present (expected Dec 2025)*
Master of Science in Computer Science (In Progress) GPA**:** 3.83/4.0

- **Highlighted Coursework:** Programming Design Paradigm, Algorithms, Computer Vision, Software Engineering

Virginia Tech *09/2018 - 05/2023*
Bachelor of Science in Computer Science Major GPA: 3.23/4.0
Minor in Human-Computer Interaction; Minor in Mathematics Overall GPA: 3.19/4.0

- **Highlighted Coursework**: Machine Learning, Computer Systems, Data and Algorithm Analysis, Numerical Analysis, Applied Combinatorics, Data Structures and Algorithms, Human-Computer Interaction, GUI programming and Graphics

SKILLS SUMMARY

- **Programming Language:** C, C++, Java, C#, Python
- **Programming Tools:** Eclipse, IntelliJ, Visual Studio Code, Unity, Android Studio, VirtualBox

WORK EXPERIENCE

AI Developer Intern *05/2024-8/2024*
Fenghua Railway Informatization Co., Ltd. - Zhengzhou, Henan, China

- Developed an object detection model that recognizes objects (person, bush, new track) around railways.
- Created an image dataset with over 3000 labeled images based on provided video samples.
- Trained on the image dataset using a pre-trained yolov5 network. The final model reaches **95%** accuracy on the test set and runs efficiently on live video streams.

PROJECT

Comparative Analysis of Deblurring Models (*Python*) *09/2025-09/2025*
Main developer | Advisor: Dr. Bruce Maxwell, Professor of Computer Vision

- Developed and compared a CNN-based U-Net against a Transformer-based Swin-Unet for high-fidelity image deblurring.
- Engineered a hybrid Swin-Unet with multi-scale skip connections and enhanced the U-Net decoder with Efficient Channel Attention (ECA) principles to achieving state-of-the-art, artifact-free results.
- Implemented an advanced training pipeline using perceptual loss (VGG), a learning rate scheduler, and early stopping to maximize performance.

Calligraphy Style Recognition (*Python*) *09/2024-12/2024*
Main developer | Advisor: Dr. Bruce Maxwell, Professor of Computer Vision

- Developed a machine learning model to classify 20 distinct styles of Chinese Calligraphy based on EfficientNet50.
- The model achieved an average accuracy of **97%** after applying stratified sampling and attention mechanism.

Discord Study Group Chatbot (*Java*) *09/2024-12/2024*
Lead developer of team of four | Advisor: Prof. Alexander Lash, Professor of Software Engineering

- Design the chatbot from scratch to make it able to let students create and join study groups.
- Store user information in database using MongoDB.
- Develop the feature that sends reminder notifications to users before scheduled meeting times and allow users to customize their own reminder lead time.

Chatroom Server and Client Application (*Java*) *09/2023-12/2023*
Main developer | Advisor: Dr. Brian Cross, Professor of Programming Design Paradigm

- Design a chatroom application, both the server side and the client side. It allows users to send direct or broadcast messages. The server side can hold maximum of 10 clients at a time. Each client will be handled by an individual thread.
- The client side also contains a random grammar generator that can generate random sentences from provided grammar JSON file. User can also choose to generate and send a random sentence to another user using the random grammar generator.

Fitex Farmers Market (*C# with Unity*) *02/2023-05/2023*
Developed with a team of four | Advisor: Dr. Donald Scott McCrickard, Professor of Human-Computer Interaction Capstone

- Create a game that collect player's daily step and use step counts to grow crops. Player can complete daily quests and weekly quests to get experience and money and can also form a group with friends and complete group challenges together.
- Design a quest and group challenge generator, which generates quests based on the player's level and unlocked crop types.

Fork-Join Threadpool (*C*) *08/2022-12/2022*
Developed with a partner | Advisor: Dr. Godmar Back, Professor of Computer Systems

- Implement a fork-join thread pool that handles multi-threading. The program will spawn different number of threads based on user needs and use work-stealing approach to increase performance. The program only uses one lock for preventing data race. It locks before one thread is accessing one shared data and unlocks when the thread is done working.