+1 (781) 518 1168
minh_d.nguyen@tufts.edu
minhducnguyen.com

Minh D. Nguyen

Education

Tufts University
Computer Science, *GPA 3.94* *2015-2019*

Skills

- Languages: Python, C, C++, JavaScript, HTML, CSS, Matlab
- Frameworks: jQuery, Node.js, Bootstrap, Twisted, Qt, PyQt, TensorFlow
- OS: Linux, macOS, Windows

Experience

EditShare, LLC **Watertown, MA**
Software Engineering Intern, *Summer 2017*
- Implemented a new storage solution based on a Linux file system that allows administrators to manage access permissions of other users on individual directories stored on the file system.
- Used Twisted Perspective Broker framework to implement a server API that can handle Remote Procedure Call to modify Access Control List (ACL) permissions on directories stored on the file system.
- Used Qt and PyQt to implement a cross-platform Graphical User Interface application that allows users to utilize the permission management server API.

Tufts University Human - Robot Interaction Laboratory **Medford, MA**
Research Intern, *Fall 2016 - Spring 2017*
- Implemented the Python version using TensorFlow for the neural field model for real time speech perception project.
- Participated in discussions on how to improve the prediction accuracy of the model.

Hanoi University of Science High School for Gifted Students (HUS - HSGS) **Hanoi, Vietnam**
Computer Science Olympiad Team Trainer, *Summer 2016*
- Prepared the HUS HSGS Computer Science Olympiad Team, one of the best in Vietnam, for national and international competitions such as the International Olympiad in Informatics.
- Taught how mathematics concepts and theorems such as the Hall's marriage theorem or the invariant principle can be applied in computer science.
- Introduced the students to various advanced data structures, such as Binary Indexed Tree and Interval Tree, and their applications.

Projects

Neural Field for Speech Recognition
Python program,
- Given the implementation of the neural field model in C++, re-implemented the model in Python using TensorFlow for better scalability and efficiency.
- Doubled the maximum size of the sound data set processable by the model by using sparse tensors.
- Modularized the implementation of the neural network by utilizing tensor slicing and joining, allowing the model to support any given number of neural field layers.

Gomoku AI
Python and C++ program,
- Built two different Artificial Inteligence (AI) programs that can play the game Gomoku.
- Implemented the first AI using a move score calculation rule based on a defensive strategy. The AI followed roughly 80% of predicted moves.
- Implemented the second AI using the Minimax Algorithm to determine the best possible move under the assumption that the opponent also played an optimal game.
- Researched to see how a third AI can be implemented using deep learning.

Trailer Nailer
Web application,
- Implemented a web game that allows players to test their movie recognition ability by watching trailers.
- Used MongoDB to store information of roughly 200 movies from 5 different genres, and used Node.js to implement a web server to respond to movie information queries.
- Used YouTube API to embed in-game movie trailer videos.
- Used Facebook API to allow players to log in to Facebook and introduce the game and send challenges to friends.

Image Compressor and Decompressor
C program,
- Used quantization of pixels and bit-level manipulations to compress each 2x2 pixel block into a 32-bit word.
- Reduced (on average) 66.7% in image size. Decompression restored (on average) 98.2% of the original image.
- Used C, written in Linux environment.