Andrew Wang

Student ID: 21010686 | 647-916-9309 | a387wang@uwaterloo.ca GitHub: https://github.com/AndrewWang12345/

Languages: Python, Java, XML/HTML, Javascript/Typescript, C#, C/C++, SQL/NoSQL/MySQL/PostgreSQL, CSS
Tools: GitHub/GitLab, Pycharm, NetBeans, Eclipse, Codeblocks, Visual Studio Code(VSCode), Matlab, Jupyter Notebook
Frameworks/libraries: NumPy, Cython, scikit-learn, SciPy subprocess, matplotlib, pandas, joblib, Selenium, MongoDB, React, Express, NodeJS
Other Qualification: Excellent communication skills, knowledge of **CI/CD** and **dev utilities**, knowledge of **UX/UI design** to achieve optimal user experience, hands-on experience in **statistical analysis**, fluent in spoken and written **Mandarin**

WORK EXPERIENCE

Intermediate/Senior Python Programmer, H2O Geomatics, Waterloo, ON Sept 2023 - Dec 2023
Designed, implemented and tested **Big Data analytics and machine learning/AI** software algorithms and processing chains (workflows) from large global satellite datasets using **NumPy, Cython, SKlearn, SciPy, subprocess, matplotlib, pandas, joblib**.

- Collected additional data to improve **training datasets**
- Analyzed, researched, and proposed technical solutions for improving performance of code, implemented and resulted in up to 50x processing speed improvement for **large data sets** using **Cythonization**, **Numpy**, **Parallel Processing**, **Numba, and Python code optimization**.
- Utilized **Modular Object Oriented Python Coding** to simplify large problems
- Implemented **memory maps** to improve file IO
- Designed and implemented an algorithm to merge raster images based on **uncertainty** values of each pixel as part of the **machine learning/AI tool** chain.
- Designed and implemented **Graphical User-Interface (GUI)** for user friendly data file selection and operations.
- Drew **design diagrams** and wrote **technical documents**
- Employed **Agile** development methodologies in my work and **GitHub** for **version control**

Python Programmer, Meteorological Service of Canada, Remote Jan 2023 - Apr 2023
Accomplished the following tasks for **numerical modeling systems** handling large data sets

- Implemented **data mining algorithms** and preformed **time series analysis** in **Python** to extract erroneous temperature measurements periodically collected by weather stations across North America over the span of 5 years
- Coded **Python/Linux AWK script** to iterate through all files and detect and remove all non-Canadian stations by analyzing the coordinate location of the station
- Implemented a **Logistic Regression** algorithm to predict the temperature at each weather station and **K-nearest Neighbors Classification** algorithm to label them as “erroneous” or “proper” based on its deviation from the predicted value
- Designed test cases and organized test data with expected results in different folders so that they can be reused when code changes
- Drew **design diagrams,** wrote **technical documents,** and used **proper Python coding style**

PROJECTS/EXPERIENCE

Chat App

- The user can register an account, and text any other use on the app in real time
- Built completely responsive webpage structured and styled using **HTML/CSS** with **ReactJS**
- Stored Chat History and login information via **MongoDB**, **encrypting passwords** for **security**
- Used **Socket.IO** to enable real-time, bi-directional communication between web clients and servers.
- Created a working contact form to allow users to report misdemeanor to me via email
- Pushed files to **Github** and deployed application to **Vercel**

Automatic Chat Bot

- Takes a line seed text as input and generates a piece of synthetic, continuation text based on a set of training data
- Implemented the **Byte Pair Encoding** (BPE) algorithm to **tokenize** input text data and training data
- Implemented a **Generative Pre-trained Transformer(GPT)** model using **Pytorch,** encompassing multiple layers of self-attention mechanisms, positional encodings, and position-wise feedforward networks
- Previous inputs and generation is automatically fed back into the GPT as context
- Outputted the model to a **.pth** file for **reusability**
- Currently working towards implementing reinforcement learning from human feedback and integrating the app to a ReactJS webpage

OCR Sudoku Solver:

- Accepts an image of a Sudoku grid, determines the positions of each number, and outputs the completed puzzle
- Drew **design diagrams** using **Microsoft Visio** detailing the general programming flowchart for my program
- Used **OpenCV** to convert the image to gray-scale, then detect all contours on the page and then isolate the largest one(the Sudoku box)
- Once all vertical and horizontal gridlines have been detected, their interception points are found and used to divide the board into boxes
- Implemented the **K-Nearest Neighbors machine learning algorithm** to compare each Sudoku box to the hand drawn digits from the **MNIST dataset** and find the closest matches
- Implemented a **recursive backtracking algorithm** to iterate through all possible values for each grid of the Sudoku

Tensorflow Facial Recognition:

- Mass-collected images from Google via **Selenium** to perform **image analysis** on
- Created annotations using **LabelMe**
- Applied image augmentation on images and labels
- Built a **Deep Learning Model** using **Functional API**
- Defined a Custom Loss Function & Optimizer and trained a **Neural Network**

Face Shape and Expression Detector:

- Scans the webcam feed for the presence of a human face, then determines the shape and expression of the face
- Mass-collected images of various face shapes with various expressions from Google via **Selenium**
- Reduced each face down to a series of **Landmark coordinates** using **Mediapipe** and created a **training dataset**
- Divided dataset into **train** and **test** using SKLearn
- **Normalized** coordinates for better accuracy
- Tested and visualized various **machine learning algorithms** such as **regression**, **classification**, and **clustering** to determine the most effective algorithm to identify patterns and relationships in the data.
- Trained the machine learning models using **SKLearn** and saved it as a **joblib** file

EDUCATION: University of Waterloo

- Candidate for B.A.Sc. in Honors Computer Engineering, Co-op