OLIVIA **MARKHAM**

Website | LinkedIn | GitHub | ogmarkha@uwaterloo.ca | 416-677-7166

SKILLS

Languages: C++, Python, C, Java, JavaScript, HTML, CSS, SQL
Frameworks: TensorFlow, OpenCV, Matplotlib, NumPy, Pandas, Pygame, Keras
Tools: Git/GitHub, GitLab, PyCharm, Visual Studio, Jupyter Notebook, Altium Designer

TECHNICAL EXPERIENCE

Waterloo Aerial Robotics Group (WARG) — Nov. 2022 - Present
Computer Vision Team Member

- Achieved an 80% accuracy rate using a **convolutional neural network** to train on the **CIFAR10** dataset using Python, TensorFlow and Matplot
- Researching and developing a **semantic 3D map** of an urban environment with a **3D LiDAR** and a camera for robot navigation with **autonomous landing research and development** team

FIRST Robotics Competition Team 8884 — Oct. 2021 - July 2022
Founder and President

- Incorporated ball acquisition and firing mechanisms into the robot design, as well as implemented a functional gripping system, landing the Rookie Inspiration Award and **Top Rookie Seed Award**
- Optimized the process flow between sub-teams by enforcing consistent scheduling and delegation of tasks, playing a key role in driving the success of the now **highly competitive** robotics program
- Executed a highly effective campaign that resulted in the acquisition of **$36,000** in funding, yielding a marked improvement in our team's performance.

PROJECTS

Toyota Innovation Challenge Hackathon — Oct. 2022

- Accurately tracked the real-time position of a model car and provided trigger signals to the vehicle in C++ using **OpenCV**
- Implemented a **live position tracking** system using a lit depth visualizer and averaged frames to mitigate camera warping and achieve **95% accurate** bounding boxes

Self-driving Car — Dec. 2022 - Present

- Developed a self-driving car and **neural network simulation** using **JavaScript** with no external libraries, including the implementation of driving mechanics, environment definition, sensor simulation, collision detection, and control using a neural network
- **Artificial Intelligence model** achieved high accuracy in simulated traffic scenarios

Personal Portfolio Website — Dec. 2022 - Present

- Designed and constructed a responsive website with interactive elements using HTML, CSS, and JavaScript, enhancing user experience and engagement

Minesweeper AI — Nov. 2022 - Dec. 2022

- Crafted a highly precise version of the classic Minesweeper game using Pygame, **artificial intelligence**, machine learning techniques, resulting in an AI with a **99% win rate** on an 8x8 grid

EDUCATION

University of Waterloo - BASc in Mechatronics Engineering - GPA 3.74 — Sep. 2022 - May 2027

- Relevant Courses: Algorithms and Data Structures, Digital Computation, Circuits, Linear Algebra

EdX — Oct.2022 - Present
CS50 Introduction to Computer Science, and CS50's Introduction to Artificial Intelligence with Python

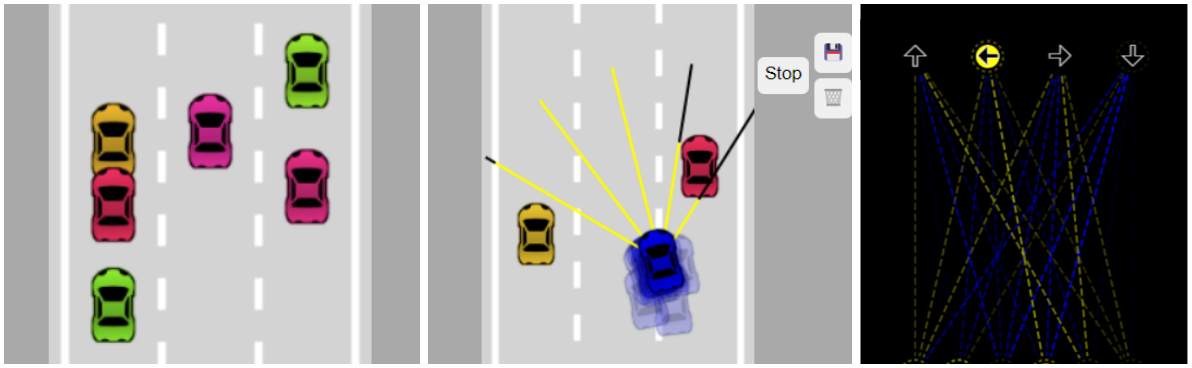
- Projects involving neural networks, algorithms, data structures, **software engineering**, and AI
- Languages and frameworks including: Python, C, SQL, HTML, CSS, JavaScript, TensorFlow, PyTorch, and SciKit-Learn

AWARDS

- FIRST Canada Canadian Women in STEM Scholarship (2022)
- **1st place** team in the Velocity Speed Hack Event Environment Division (2022)
- **1st place** team in the Toyota Innovation Challenge Hackathon for C++ (2022)
- **2nd place team in Canada** in the Act In Space Hackathon (2022)

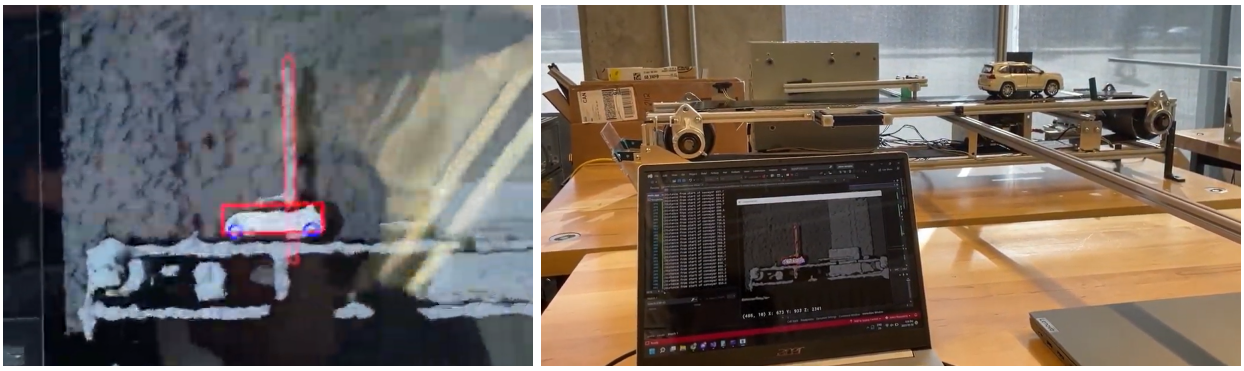
PORTFOLIO

Self-driving Car Simulation



1. Developed a neural network using JavaScript with no external libraries to control the movement of a self-driving car in a simulation, along with a visualization of the network
2. The neural network was trained to make decisions about the car's movement based on sensor data and the position of the car on the road.
3. Matrix operations were used to perform forward propagation and backpropagation, increasing accuracy of the model
4. Utilized browser storage by saving the best performing neural network to the browser's local storage using the 'localStorage' API, to be retrieved and used in future runs of the simulation
5. Devised traffic using collision avoidance techniques to ensure properly generated cars
6. Code and demonstration here

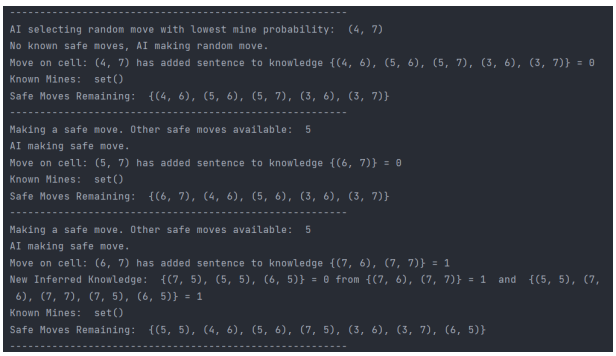
OpenCV Live-Position Tracking Program



1. Competed in the Toyota Innovation Challenge Hackathon to program a solution for accurate position tracking due to communication delays and mechanical issues. Developed a successful program using OpenCV and C++ in under 12 hours with a team of four.
2. Implemented a live position tracking system using a lit depth visualizer, resulting in fully accurate bounding boxes around key systems such as the wheels, car, and post.
3. Averaged frames to mitigate camera warping and achieve full accuracy in location detection. When the front wheel passes a certain point(the post bounded in red) the camera is triggered and an image is taken and saved to the computer. During this period the program is constantly updating and outputting the car's distance from the start of the conveyor belt.
4. Code and demonstration here

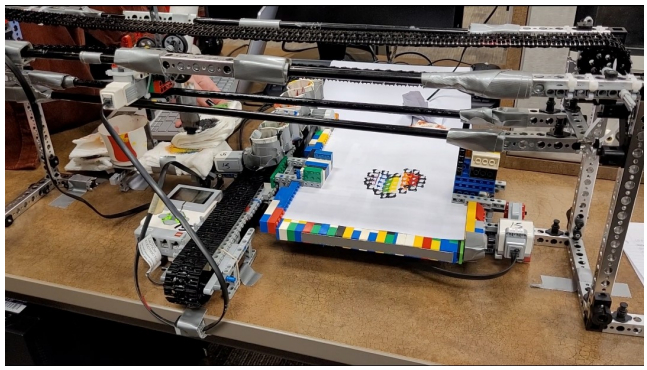
PORTFOLIO

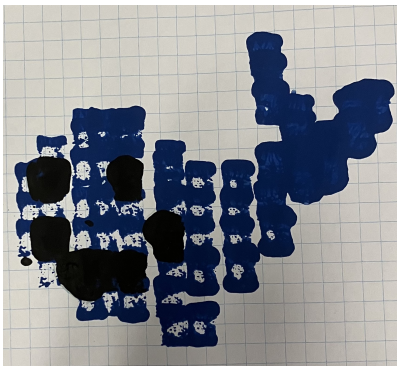
Minesweeper AI



1. Developed an AI agent for Minesweeper using rule-based logic, probability, and logical deduction in Python and Pygame
2. Inducted design that prioritized cells with a higher probability of being a mine based on inference strategies as shown above. Calculated best cost decision when no safe moves were available.
3. Kept track of the number of mines that were flagged, and the number of mines that were left to be flagged, in order to make more accurate decisions. New inference knowledge constantly updated the set of known mines, thus computing possible safe moves.
4. Code here, demonstration here

Pixel Painter Robot





1. Designed mechanisms capable of allowing the brush to move to all points on the page, and to accurately switch paint colors (and dip into water and a paper towel to remove paint). Collaborated with three peers to create the pixel painter robot as our first term final project.
2. Built a C++ program capable of inputting user-generated pixel art images. Programmed(in RobotC) functions capable of moving the brush and paper along four axes and ensured the robot was able to accurately paint a variety of images in multiple colors.
3. Created and blueprinted flowcharts for major procedures and necessary functions, using motor encoders and sensors to allow for the verification of accurate placement of all initialized parts.
4. Full report here