1. Title: Applying Neural Networks to Minesweeper for Mine Detection
   Students: Moez Bajwa (ID: 101196537) and Steven Kong (ID: 101189675)
   Course Code: COMP3106A
   Semester: Fall 2024

2. Minesweeper is a popular game where to win, players need to reveal spaces on a grid while avoiding spaces with mines. To be successful, players need to apply various problem-solving techniques and strategies. Utilizing probabilistic decision making and reasoning is crucial to avoid game ending mines. Applying these sorts of methods to win the game becomes more challenging as you increment the difficulty. As the difficulty increases, so does the grid size and the number of mines on the board, making Minesweeper a challenging problem to tackle. Our solution would aim to solve minesweeper boards to a decent degree of consistency.

3. We aim to take a supervised learning approach to solve this problem. Our main objective is to create an agent which can win a game of minesweeper more than the average human player, which is approximately 20% of the time on an expert-level board (30x16 board size and 99 mines).

4. Our implementation of the minesweeper agent would be a supervised binary classification task, implementing a convolutional neural network (CNN) to find possible mine positions on a game board. CNNs are widely known and used commonly in image tasks due to their ability to capture and recognize spatial relationships, something crucial for finding patterns in matrix shaped inputs. In this context, providing our model the game grid at any state, it will be able to provide the probability of each space on the board being a mine or clear.

   Apart from this, we will be implementing a minesweeper board generation module. This module will likely end up using a variation of DFS to explore nodes to uncover based off an initial starting node on the grid, clearing out nodes which have no adjacent mines. This is a smaller part of the project, part of data preprocessing/generation, however worth mentioning as DFS is something we did cover in earlier portions of the course.

5. The data required for this project is rather simple. We will need different boards of minesweeper in various states; partially covered, fully uncovered, etc. Each of these boards will be labeled with a grid of the same size, where spaces with 1's are the mines on that grid, and spaces with 0 are the empty spaces. There is no dataset available for this sort of data, however minesweeper is a game which is rather easy to develop states for. We plan to create a large dataset in the range of 5000 to 20,000 different minesweeper boards. This will be done via development of a minesweeper board generation module which will create the dataset which the model will be trained on.

6. For analyzing our results, our metric that we will use is win percentage. Depending on the size of the board, an ideal win percentage varies. With larger boards, this win percentage will be lower as there are more mines and it is more likely to have to guess what move to

make next, rather than being able to know if a square is or isn't a mine. For the sake of analyzing the results, we will use what most Minesweeper websites consider to be the expert difficulty. This would be a 30-by-16 board with 99 mines. After researching what the expected win percentage of a random board of this size is, a player should be able to have a win percentage of around 20%. As such, we will consider it to be a success if our agent is able to win at least 20% of games over a large sample size.

7. As for novelty of our project, using AI to solve Minesweeper boards is not a novel idea in and of itself. However, most of these projects are done using reinforcement learning. Our approach will be due to using supervised learning, where we will utilize convolutional neural networks to solve the boards. The neural network will calculate the probability that each square is a mine. If a square is guaranteed to be a mine, it will have a probability of 1 and will be flagged and if a square is guaranteed to be safe, it will have a probability of 0 and the square will be revealed. If no uncovered square is guaranteed to be safe, the square that is least likely to be a mine will be revealed.

8. Weekly Schedule:
   Week 1 (Nov 3-Nov 9)- Generating the boards
   Week 2 (Nov 10-Nov 16)- implementing the logic of the game (clearing board and flagging mines)
   Week 3 (Nov 17-Nov 23)- implementing neural network to solve the board
   Week 4 (Nov 24-Nov 30)- finish implementing neural network to solve the board and testing the agent by running it on many randomly generated boards
   Week 5 (Dec 1-Dec 6)- writing report

9. We are available to present at any time between 9am-5pm on December 6th or 1pm-2pm on December 4th.

10. Our project will not require any GPU resources through the School of Computer Science