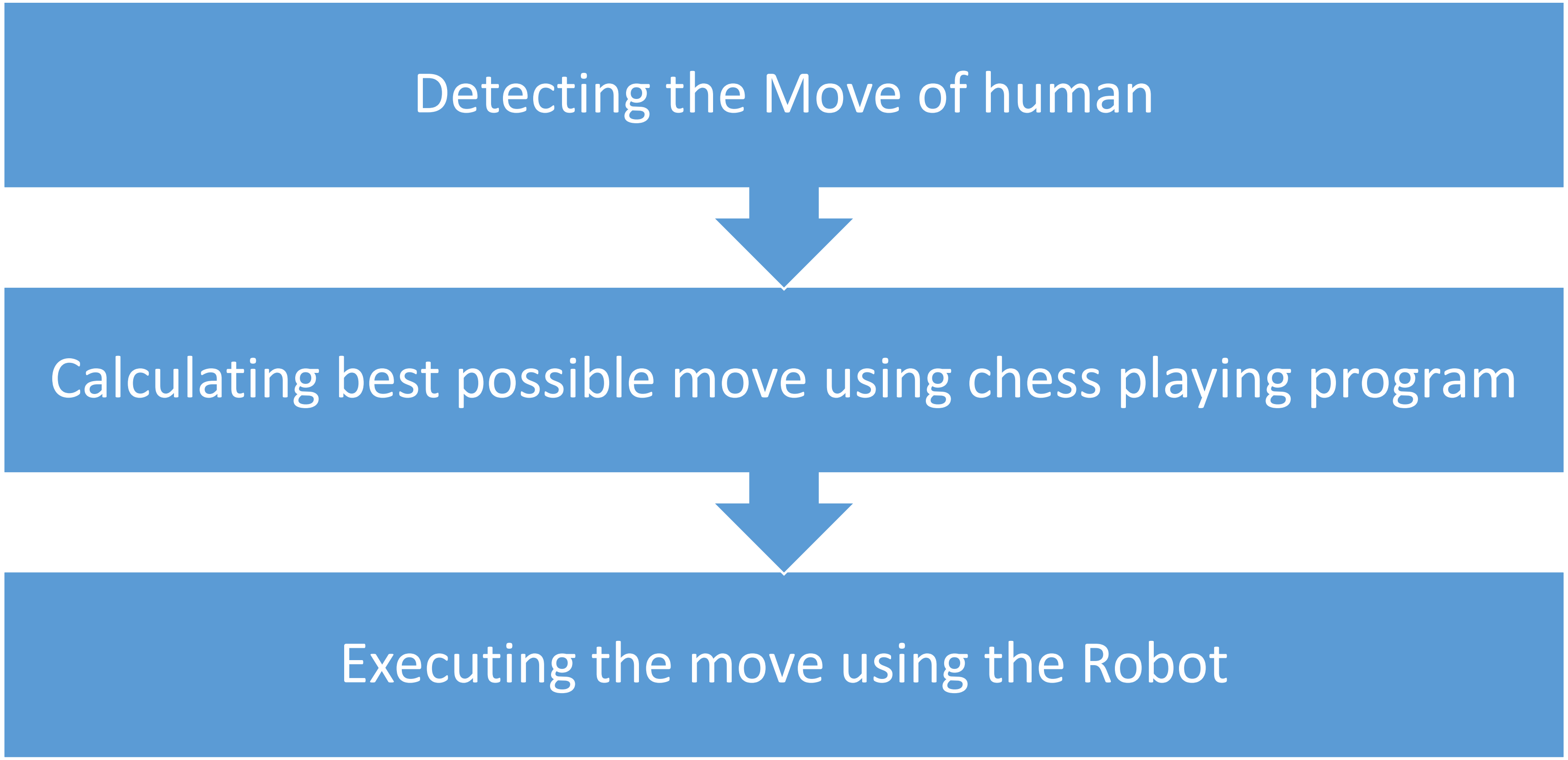
Chess Playing Robot using Lego Mindstorms

Raghav Gupta

Nikhil Aggarwal

Our Algorithm



Detecting the Move of human

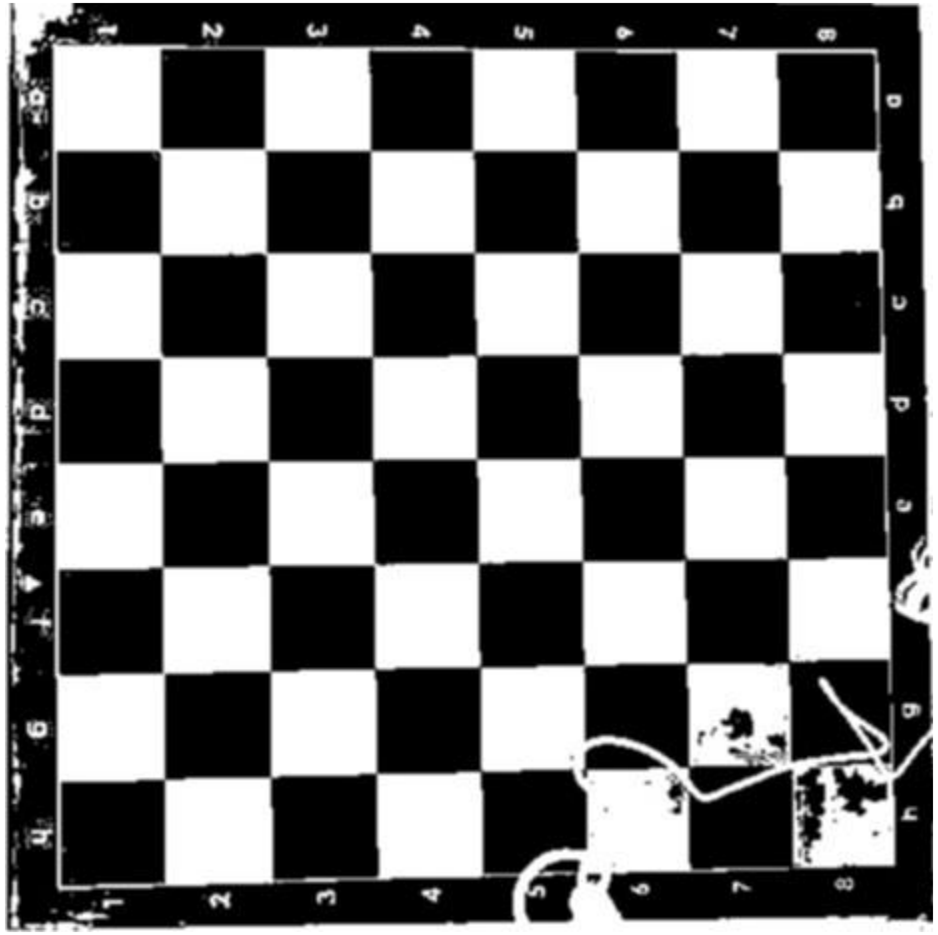
Calculating best possible move using chess playing program

Executing the move using the Robot

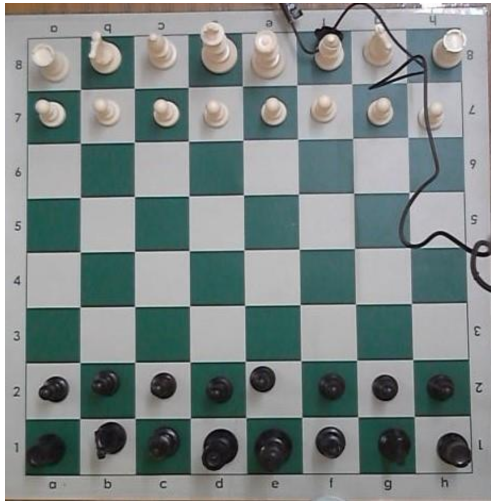
Detecting the Move

1. The Chessboard

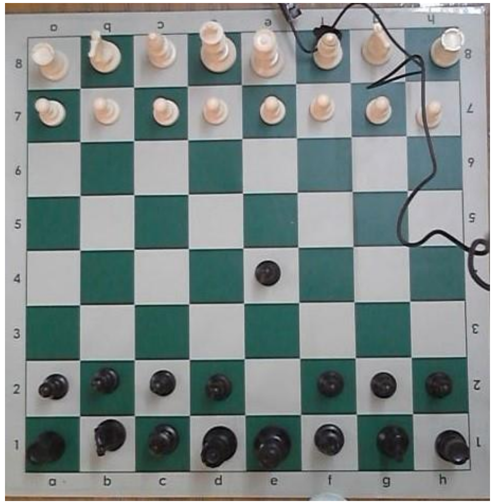
- Will create binary images of the chess board to fix the coordinates of the centres of the squares.
- Detect “blobs” – store centroid.



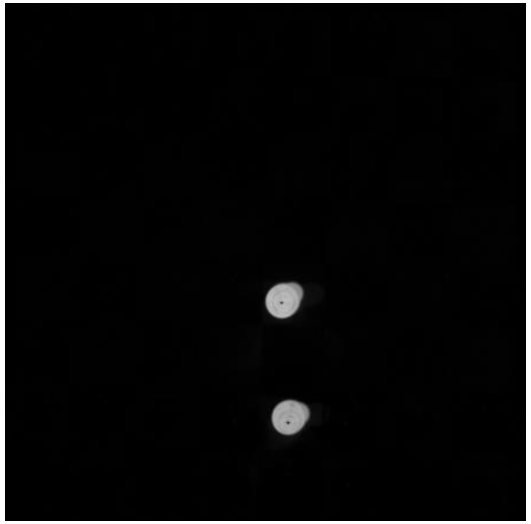
Detecting the Move



Initial Position



Final Position



Difference

Calculating the Move

- We are using a gnu chess playing algorithm ()-
- It has 2 modes-
    - Easy- Does not think when opponent is moving
    - Hard- Thinks when opponent is moving
- We can also specify the depth of thinking.
- The program takes an input from the user and calculates its move and prints the final configuration of the chessboard.

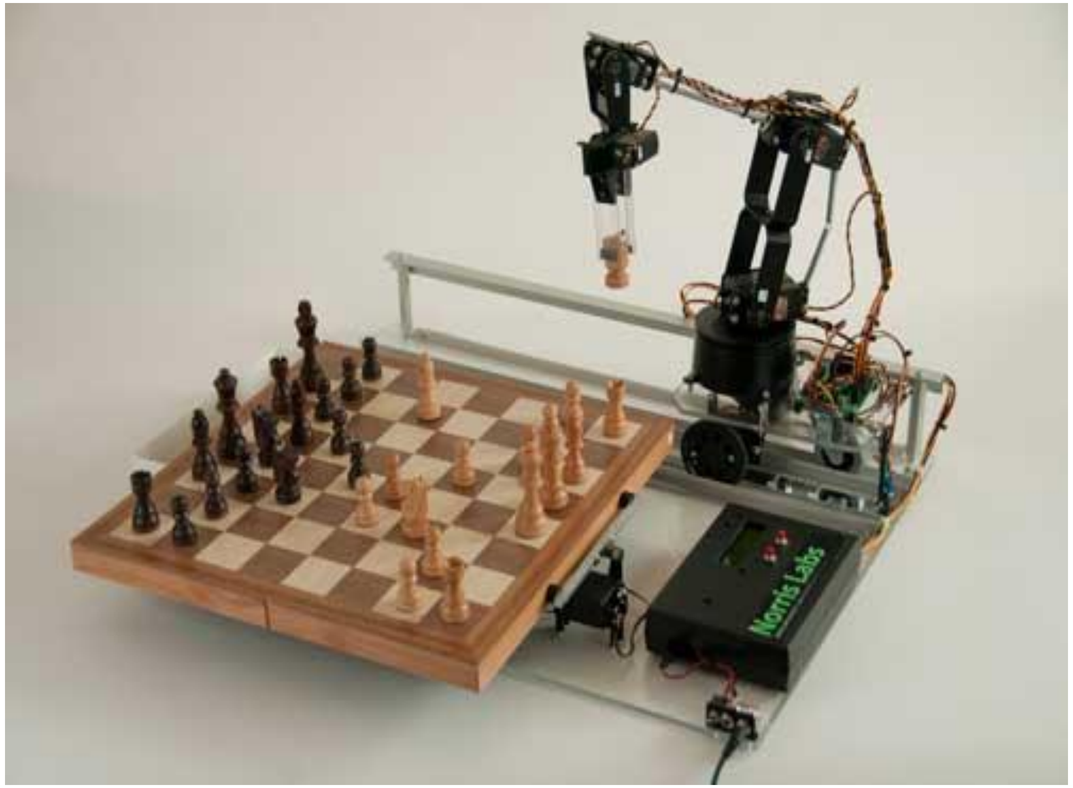
Changes In Algorithm

- The program takes an input from the Step 1 of our algorithm (Detecting the human move) and gives back the best move for the robot.
- In the last step before printing the final configuration of the board, we will return the move of the bot in the form of a string which we can use to command the robot to execute.

Executing the Move

Design of the Bot:

- The bot will have 5 degrees of freedom-
    1. Along the rails
    2. 3 hinges
    3. one for the gripper



Executing the Move

- The gripper would be marked with a different colour.
- As we know the final coordinates, we will first move the arm along the rail to reach the desired x coordinate.
- Then, using the other 3 hinges we will reach the desired Y-coordinate and pick up the piece using the gripper.
- Similarly move to the desired position and place the piece there.
- In case the move kills a piece of the opponent, we will first remove that piece and place it outside the board and then execute the above commands.