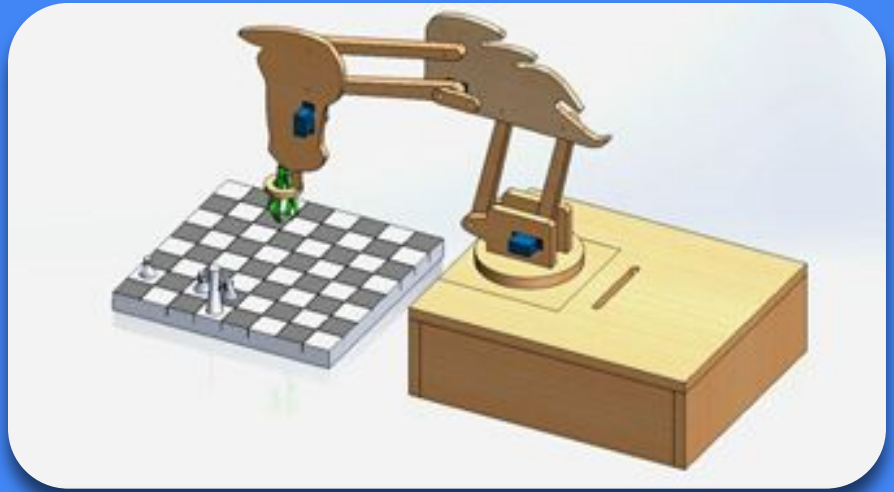


WCRA-AI

Wireless Chess Robotic Arm using artificial intelligence.

Team Members

Ahmed Sabry Ali Lilah

Alyaa Mosaad Ahmed Sherif

Moataz Mohammed Ibrahim Ali

Mohamed Ashraf Hassanen Ali

Mohamed Moataz Moustafa Hassan

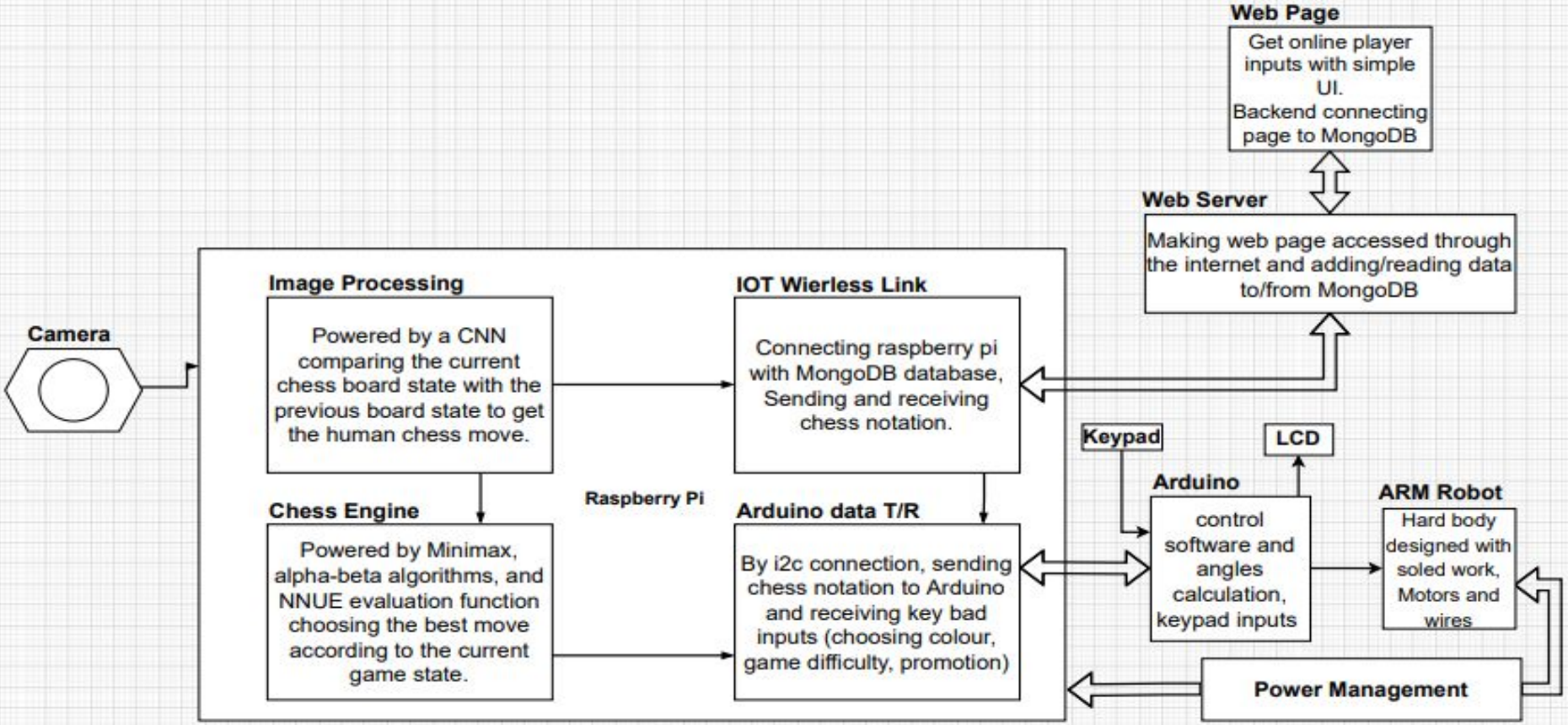
Mostafa Hisham Abd Elgawad

Supervisor

Dr.Mohammed
El-Sayed Ali Hammad

Introduction

Project block diagram



Chess Engine

How our WCRA play chess ?

Chess is a zero sum, Turn-based, and complete information game.

A zero-sum game is one in which points awarded to one player come at the expense of the other player, therefore maintaining an overall balance between players.

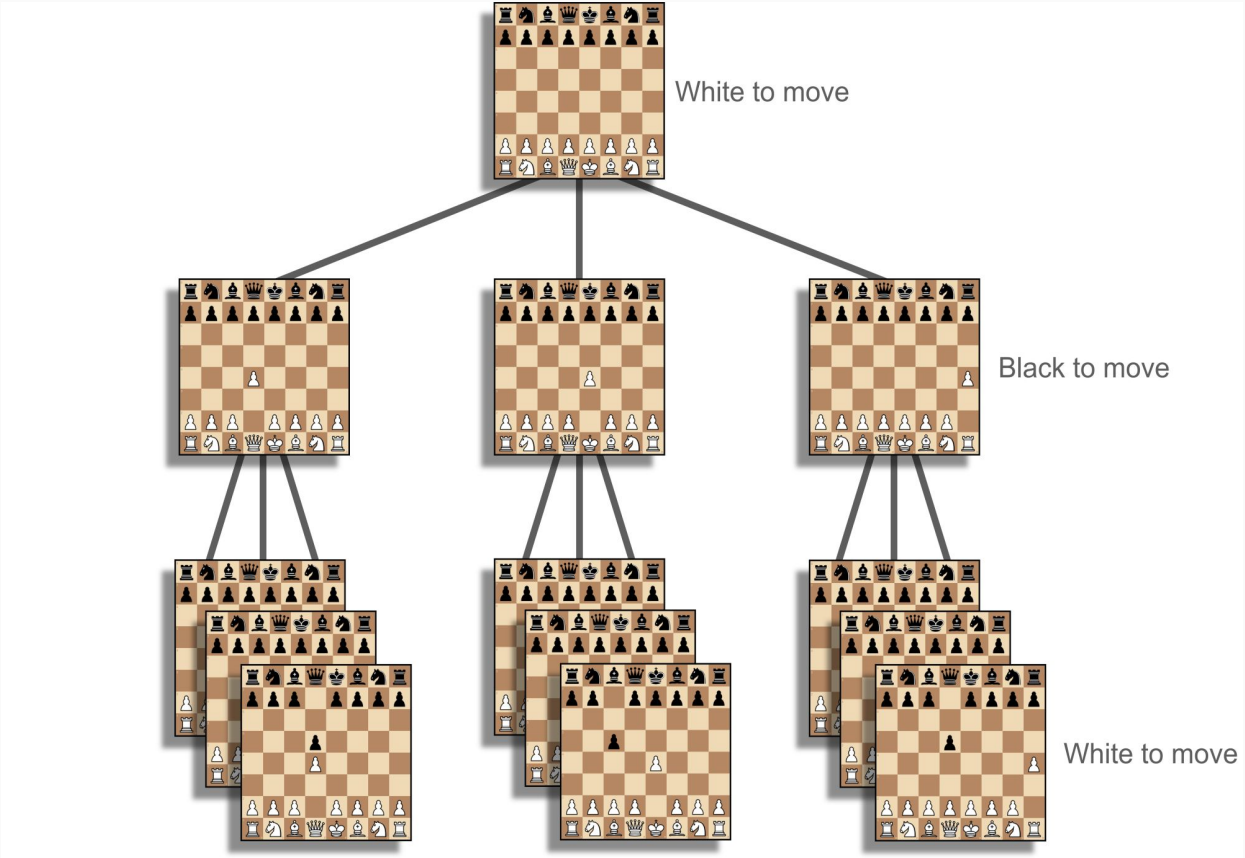
Tree-based decision making approach can be used to solve turn-based games.

Can we solve chess by brute force ?

Predicting the outcome of all possible player moves starting from the current state of the game.

Then choose the move that yield the best result.

This algorithm produces only one move.



Can we solve chess by brute force ?

Chess is not in the category of solved games.

Consider a game with only 40 pairs of moves.

So we have 10^40 different games, if there is a very very powerful computer that can process a game in 1ns, producing one move would take 10^40ns, that’s about 300*10^21 years.

a million-core supercomputer would take 300*10^15 years.

How Long is 300*10^15 years ?

Age of the Earth:	4,500,000,000 years
Age of the universe:	13,800,000,000 years
Producing one move:	300,000,000,000,000,000 years

IBM was aware of this back in 1983

Her is the treek

- Chess can’t be solved by brute force.
- We need to stop at a certain depth.
- Only terminal states produces reliable utility values.
- Estimations of these values will be enough to get a good move.
- We need something to judge the current chess position.
- We need an evaluation function.

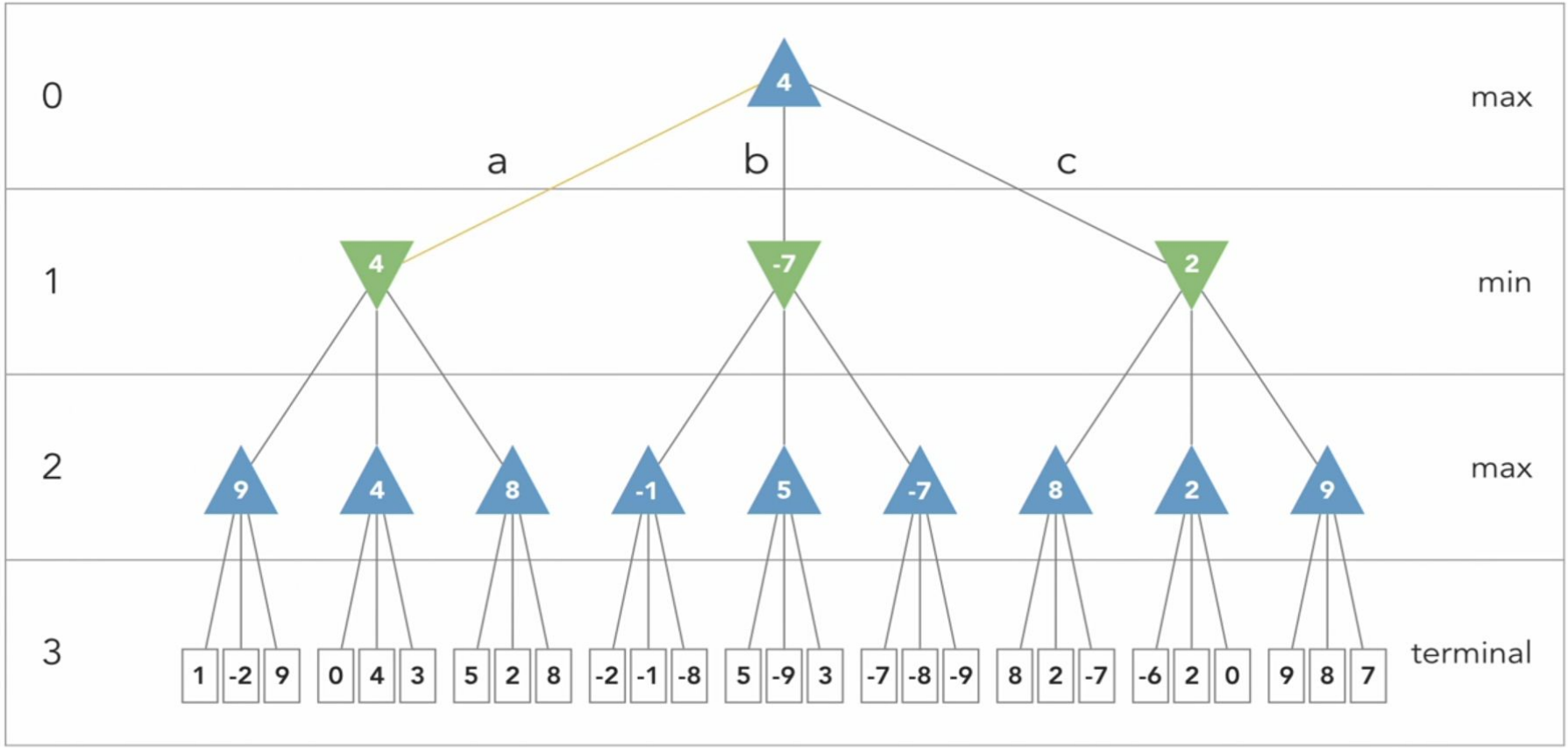
Evaluation function as a black box

$$f\left(\right.$$  $$\left.\right) = 5$$

Minimax Algorithm

- Minimax is a decision rule used in artificial intelligence, decision theory, game theory, and statistics.
- It’s a tree-based , depth-first search(almost a brute force).
- Minimize the opponent’s maximum payoff.
- Minimizes your own maximum loss.
- Simulates an opponent who wants to beat our intelligent agent.

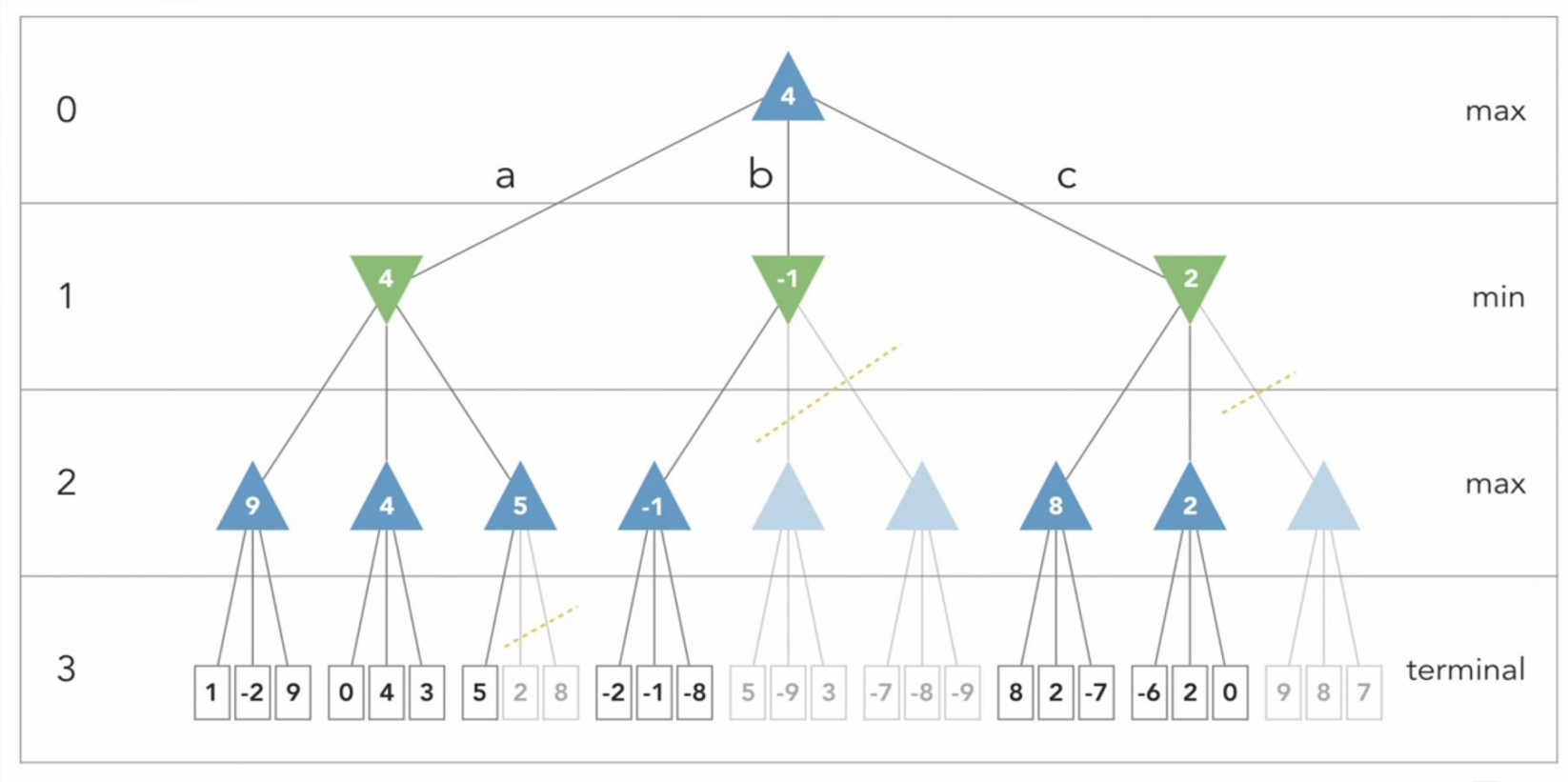
Minimax Algorithm



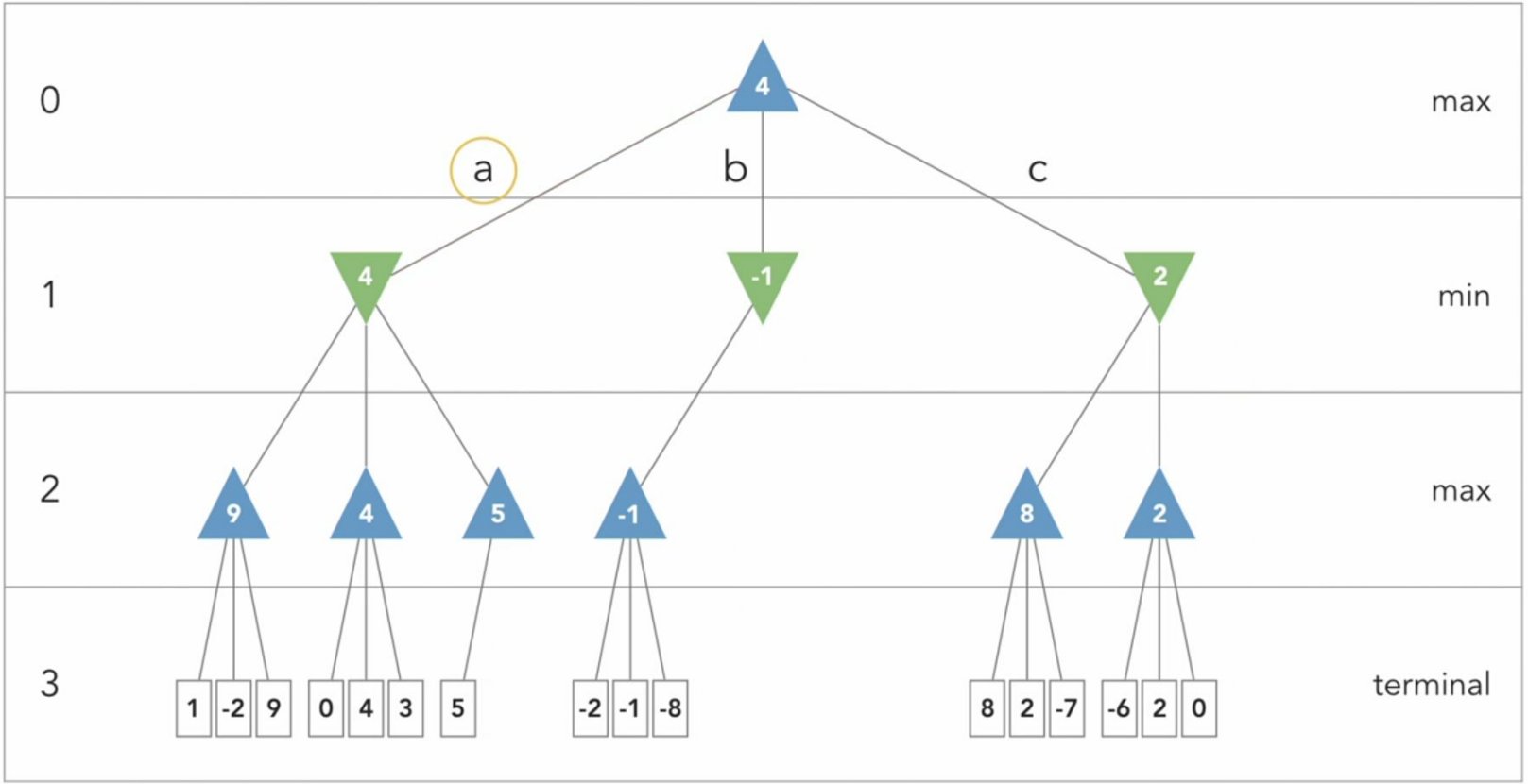
Alpha-beta Pruning

- An optimization technique for the minimax algorithm.
- Some options can't possibly be a maximum or minimum.
- These options can be safely left unchecked.
- These options are sub-trees, which may be pruned off.

Alpha-Beta Pruning



Alpha-Beta Pruning



Move Ordering and complexity

number of leaves with depth n and b = 40

depth n	b^n	$b^{\lceil n/2 \rceil} + b^{\lfloor n/2 \rfloor} - 1$
0	1	1
1	40	40
2	1,600	79
3	64,000	1,639
4	2,560,000	3,199
5	102,400,000	65,569
6	4,096,000,000	127,999
7	163,840,000,000	2,623,999
8	6,553,600,000,000	5,119,999

b: tree branching factor.

n: maximum depth.

Time complexity T(b,m) = $O(b^n)$,

space complexity $S(b,n) = O(n)$

for minimax Using alpha-beta pruning will reduce the time complexity to be $O(b^{n/2})$.

Evaluation Function

$$f\left(\right) = 5$$

Takes a chess board as input and gives a score value.

+ve value white is winning ,
-ve value black is winning , 0 mens drow.

The quality of the chess engine relies on the quality of this evaluation function.

hand-crafted evaluation Function Approach

Assign each piece to a value representing its power.

hand-crafted evaluation Function Approach

```
[ -3.0, -4.0, -4.0, -5.0, -5.0, -4.0, -4.0, -3.0],
[ -3.0, -4.0, -4.0, -5.0, -5.0, -4.0, -4.0, -3.0],
[ -3.0, -4.0, -4.0, -5.0, -5.0, -4.0, -4.0, -3.0],
[ -3.0, -4.0, -4.0, -5.0, -5.0, -4.0, -4.0, -3.0],
[ -2.0, -3.0, -3.0, -4.0, -4.0, -3.0, -3.0, -2.0],
[ -1.0, -2.0, -2.0, -2.0, -2.0, -2.0, -2.0, -1.0],
[  2.0,  2.0,  0.0,  0.0,  0.0,  0.0,  2.0,  2.0 ],
[  2.0,  3.0,  1.0,  0.0,  0.0,  1.0,  3.0,  2.0 ]
```

```
[ -2.0, -1.0, -1.0, -0.5, -0.5, -1.0, -1.0, -2.0],
[ -1.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -1.0],
[ -1.0,  0.0,  0.5,  0.5,  0.5,  0.5,  0.0, -1.0],
[ -0.5,  0.0,  0.5,  0.5,  0.5,  0.5,  0.0, -0.5],
[  0.0,  0.0,  0.5,  0.5,  0.5,  0.5,  0.0, -0.5],
[ -1.0,  0.5,  0.5,  0.5,  0.5,  0.5,  0.0, -1.0],
[ -1.0,  0.0,  0.5,  0.0,  0.0,  0.0,  0.0, -1.0],
[ -2.0, -1.0, -1.0, -0.5, -0.5, -1.0, -1.0, -2.0]
```

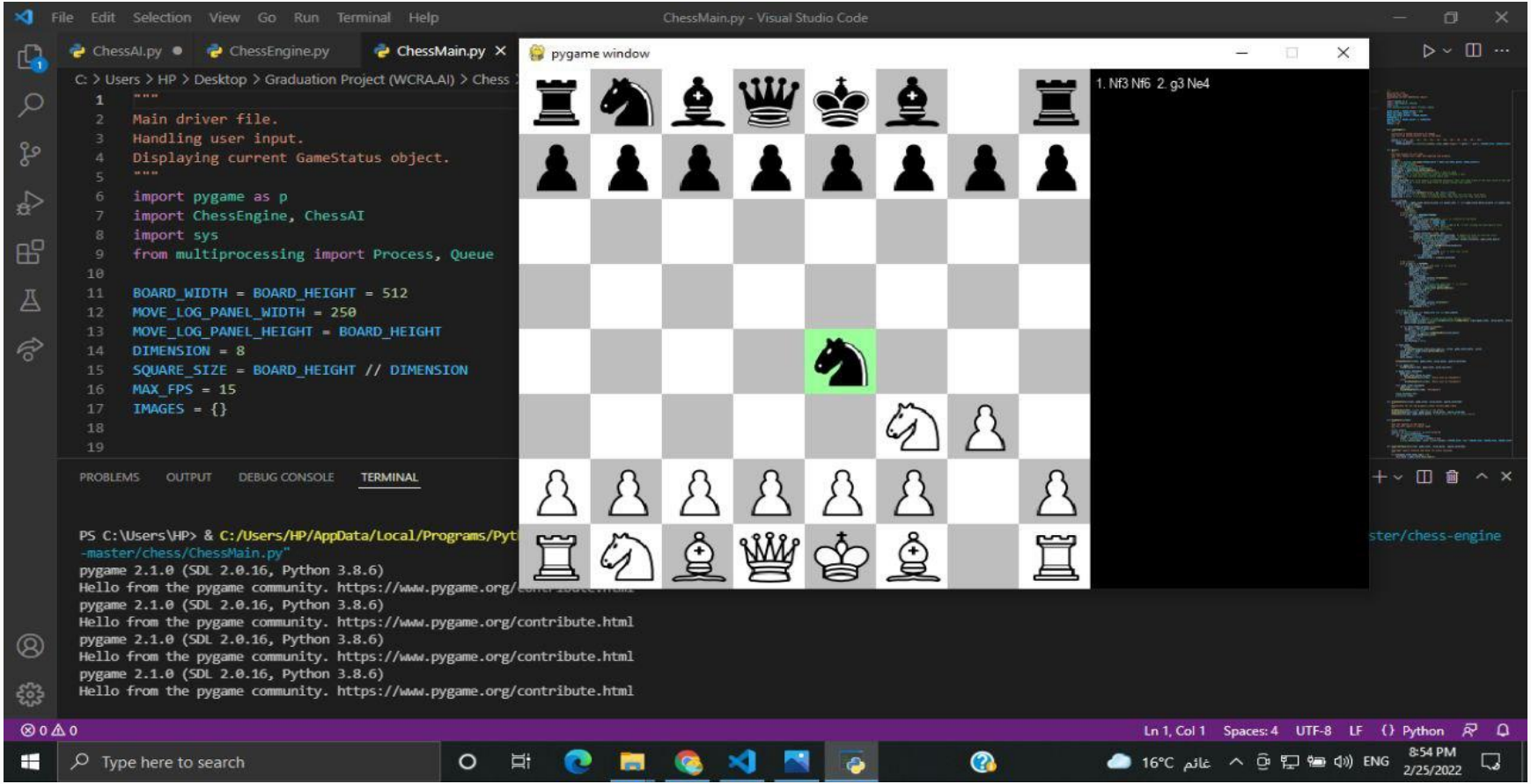
```
[  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0],
[  0.5,  1.0,  1.0,  1.0,  1.0,  1.0,  1.0,  0.5],
[ -0.5,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -0.5],
[ -0.5,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -0.5],
[ -0.5,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -0.5],
[ -0.5,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -0.5],
[ -0.5,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -0.5],
[  0.0,   0.0, 0.0,  0.5,  0.5,  0.0,  0.0,  0.0]
```

```
[ -2.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -2.0],
[ -1.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, -1.0],
[ -1.0,  0.0,  0.5,  1.0,  1.0,  0.5,  0.0, -1.0],
[ -1.0,  0.5,  0.5,  1.0,  1.0,  0.5,  0.5, -1.0],
[ -1.0,  0.0,  1.0,  1.0,  1.0,  1.0,  0.0, -1.0],
[ -1.0,  1.0,  1.0,  1.0,  1.0,  1.0,  1.0, -1.0],
[ -1.0,  0.5,  0.0,  0.0,  0.0,  0.0,  0.5, -1.0],
[ -2.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -2.0]
```

```
[-5.0, -4.0, -3.0, -3.0, -3.0, -3.0, -4.0, -5.0],
[-4.0, -2.0,  0.0,  0.0,  0.0,  0.0, -2.0, -4.0],
[-3.0,  0.0,  1.0,  1.5,  1.5,  1.0,  0.0, -3.0],
[-3.0,  0.5,  1.5,  2.0,  2.0,  1.5,  0.5, -3.0],
[-3.0,  0.0,  1.5,  2.0,  2.0,  1.5,  0.0, -3.0],
[-3.0,  0.5,  1.0,  1.5,  1.5,  1.0,  0.5, -3.0],
[-4.0, -2.0,  0.0,  0.5,  0.5,  0.0, -2.0, -4.0],
[-5.0, -4.0, -3.0, -3.0, -3.0, -3.0, -4.0, -5.0]
```

```
[0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0],
[5.0,  5.0,  5.0,  5.0,  5.0,  5.0,  5.0,  5.0],
[1.0,  1.0,  2.0,  3.0,  3.0,  2.0,  1.0,  1.0],
[0.5,  0.5,  1.0,  2.5,  2.5,  1.0,  0.5,  0.5],
[0.0,  0.0,  0.0,  2.0,  2.0,  0.0,  0.0,  0.0],
[0.5, -0.5, -1.0,  0.0,  0.0, -1.0, -0.5,  0.5],
[0.5,  1.0, 1.0,  -2.0, -2.0,  1.0,  1.0,  0.5],
[0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0]
```

Results


File Edit Selection View Go Run Terminal Help
ChessMain.py - Visual Studio Code
ChessAI.py
ChessEngine.py
ChessMain.py
C: > Users > HP > Desktop > Graduation Project (WCRA.AI) > Chess
"""
Main driver file.
Handling user input.
Displaying current GameStatus object.
"""
import pygame as p
import ChessEngine, ChessAI
import sys
from multiprocessing import Process, Queue
BOARD_WIDTH = BOARD_HEIGHT = 512
MOVE_LOG_PANEL_WIDTH = 250
MOVE_LOG_PANEL_HEIGHT = BOARD_HEIGHT
DIMENSION = 8
SQUARE_SIZE = BOARD_HEIGHT // DIMENSION
MAX_FPS = 15
IMAGES = {}
pygame window
1. Nf3 Nf6 2. g3 Ne4
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Programs/Pyt
-master/chess/ChessMain.py"
pygame 2.1.0 (SDL 2.0.16, Python 3.8.6)
Hello from the pygame community. https://www.pygame.org/
pygame 2.1.0 (SDL 2.0.16, Python 3.8.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
pygame 2.1.0 (SDL 2.0.16, Python 3.8.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
pygame 2.1.0 (SDL 2.0.16, Python 3.8.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
ster/chess-engine
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python
Type here to search
16°C
ENG
8:54 PM
2/25/2022

How to improve it ?

Pawn-structures

Center-control

King-safety

Piece Development

All of these can be programed and many more, but requires domain knowledge.
Is there a better approach not require studying chess instead of engineering.

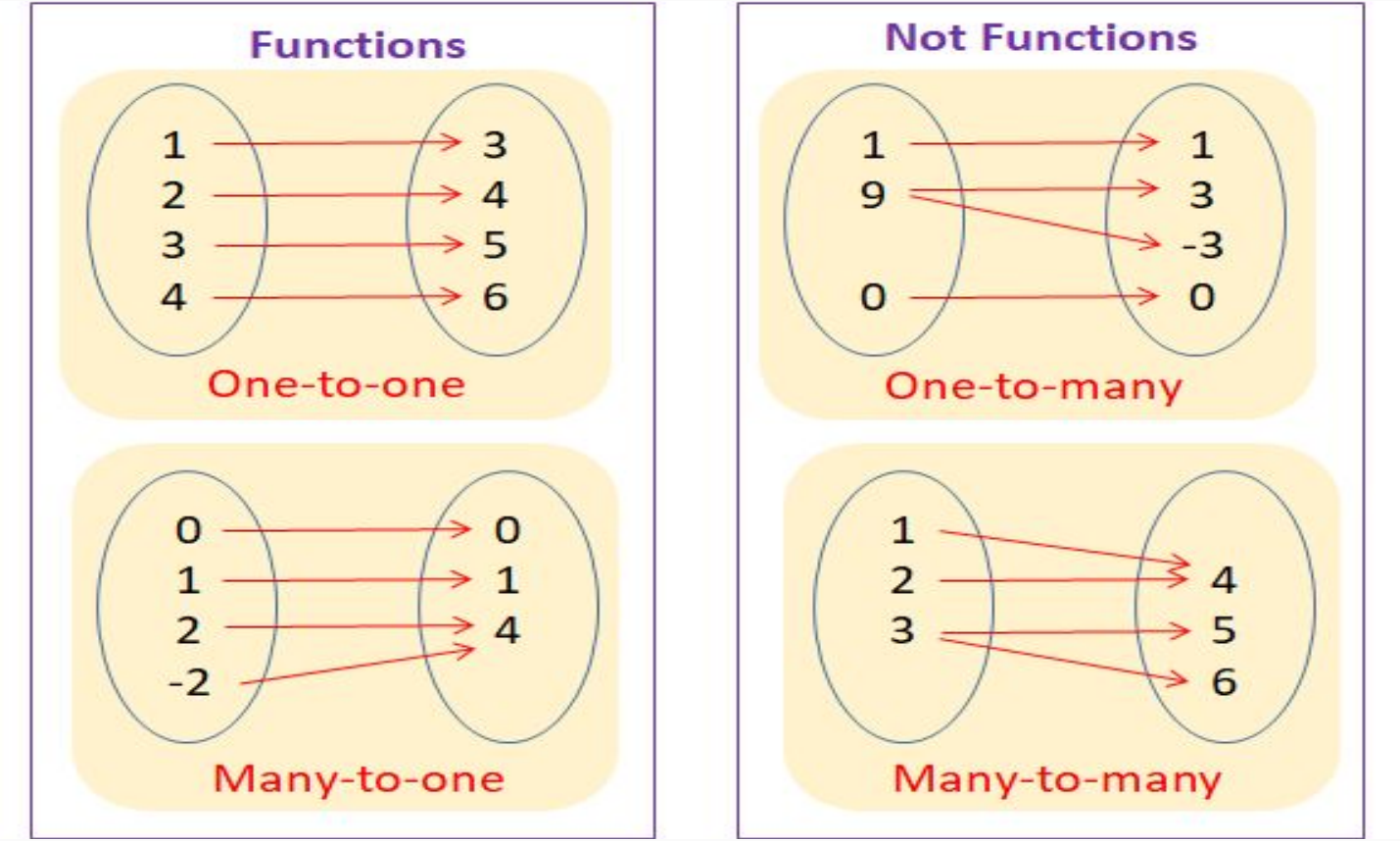
ML-Based Evaluation Function Approach

Machine learning is about extracting knowledge from data.

If we have enough data we can train a neural network to evaluate chess positions.

Mapping inputs to outputs and changing whites of Neurons.

What is function?



NNUE

Using NN to evaluate chess positions is relatively new after Yu Nasu invented the efficiently updatable neural network (NNUE) in 2018. On 6 August 2020, NNUE was for the first time ported to a chess engine, Stockfish 12 As of 2022, all of the top-rated classical chess engines, have an NNUE implementation to remain competitive.

NNUE runs efficiently on central processing units without a requirement for a graphics processing unit (GPU). Compared to deep neural network-based evaluations requiring dedicated GPUs, NNUE avoids idle times during the substantial data transfer operations between GPU and CPU required before and after each evaluation.

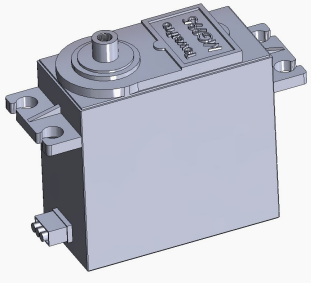
Mechanical Design

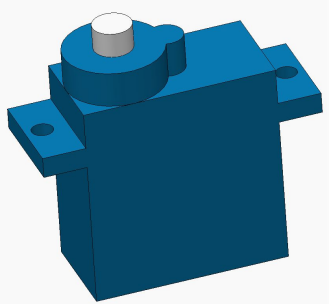
Mechanical Design

The program used for the design is Solidworks, a program provided by the French company Dassault Systems for 3D design.

Getting ready for the drawing

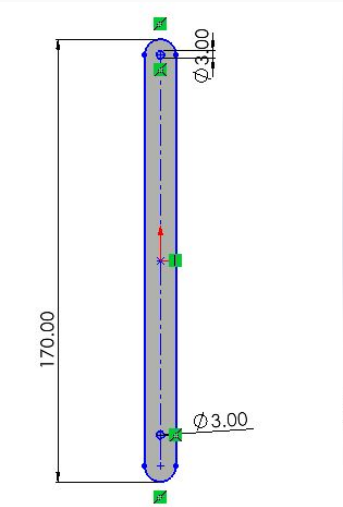
Before starting the drawing, the dimensions of the chess board were defined, the distance of the chess board from the structure of the robotic arm and the dimensions and sizes of pieces such as motors, ball bearings, screws and nuts.

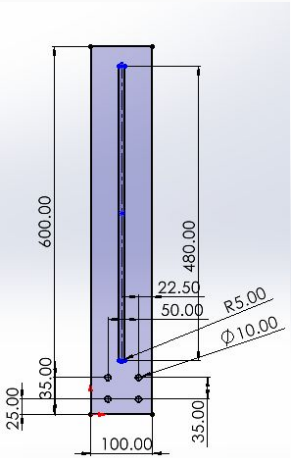


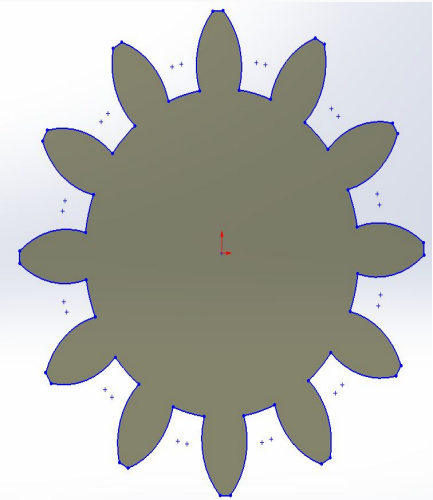


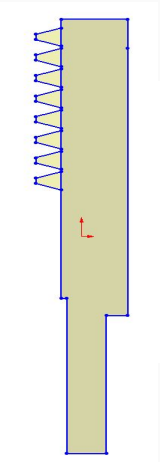
2D-Drawing

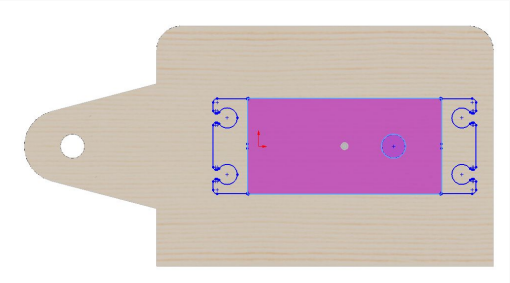
Solidworks provides the ability to draw in 2D as well.

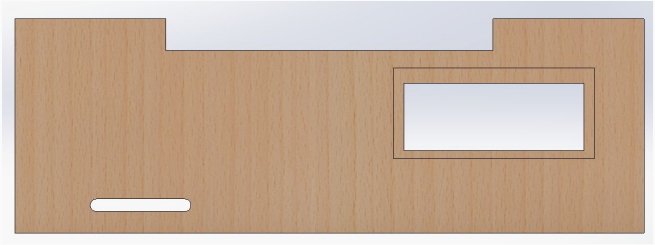


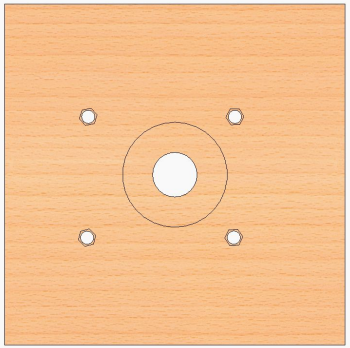


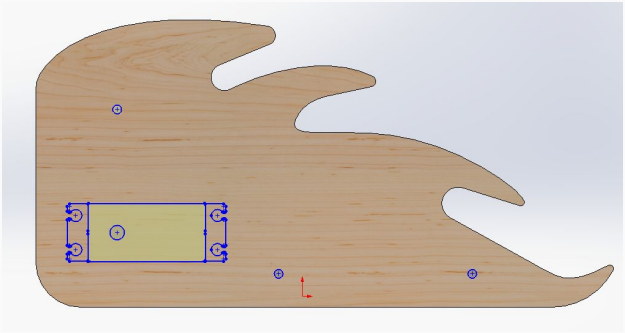


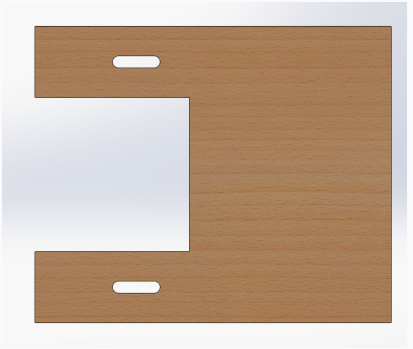










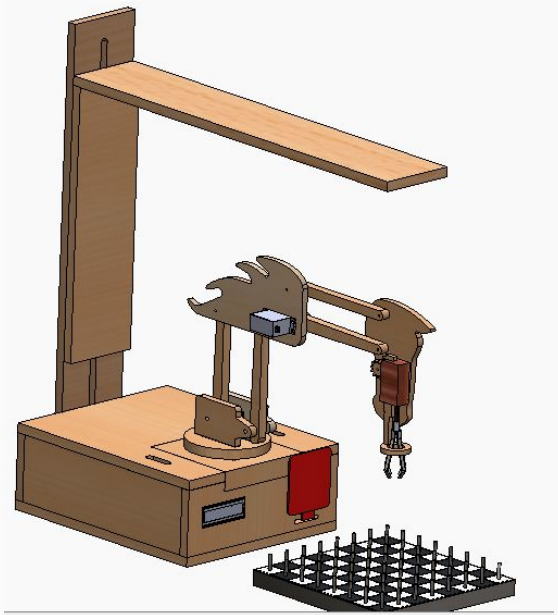


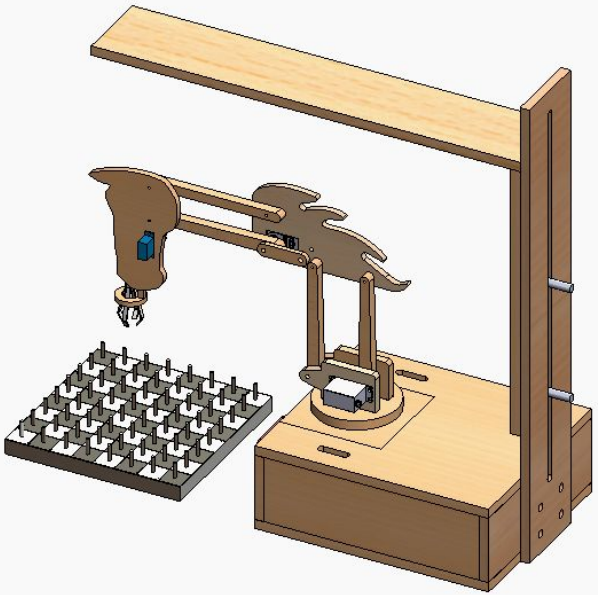
3D-Drawing

After completing the 3D drawing, we made an extrude of the drawings to convert them to a 3D drawing.

Then the pieces were assembled after taking the three-dimensional shape to form the final assembly of the robotic arm.

Complete Assembly Of The Arm





Assembly feature in SolidWorks

The assembly allows us to test the actual movement of the robotic arm to a large degree, approximating the real working environment. It also enabled us to verify the values of the angles resulting from the equations of Inverse kinematics, which give the angles for the motors to stop at the specified square.After making sure that the drawing of each piece was correct, the drawings were taken to the CNC wood cutting shop to match each drawing.

Manufacturing Process

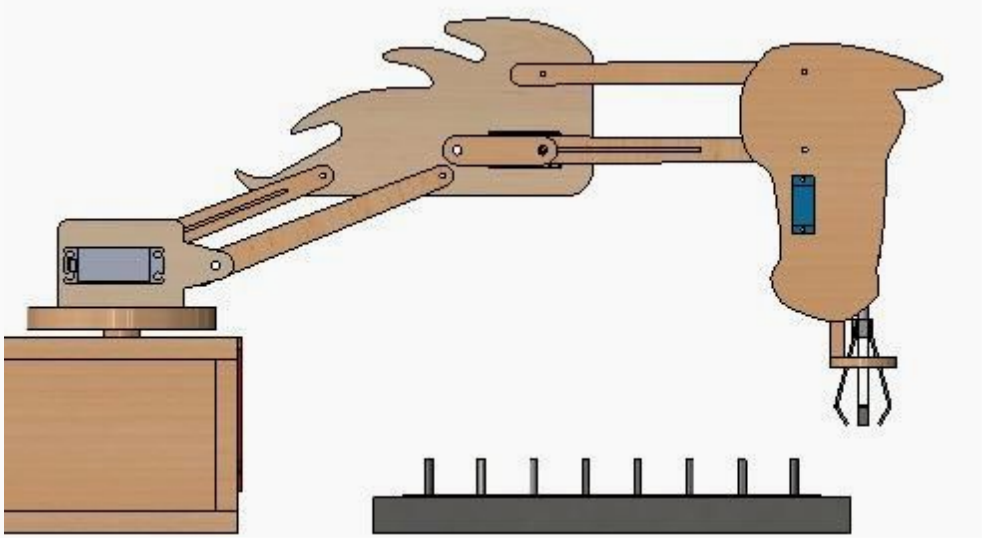
After making sure that the drawing of each piece was correct, the drawings were taken to the CNC wood cutting shop to cut wood according to each drawing.

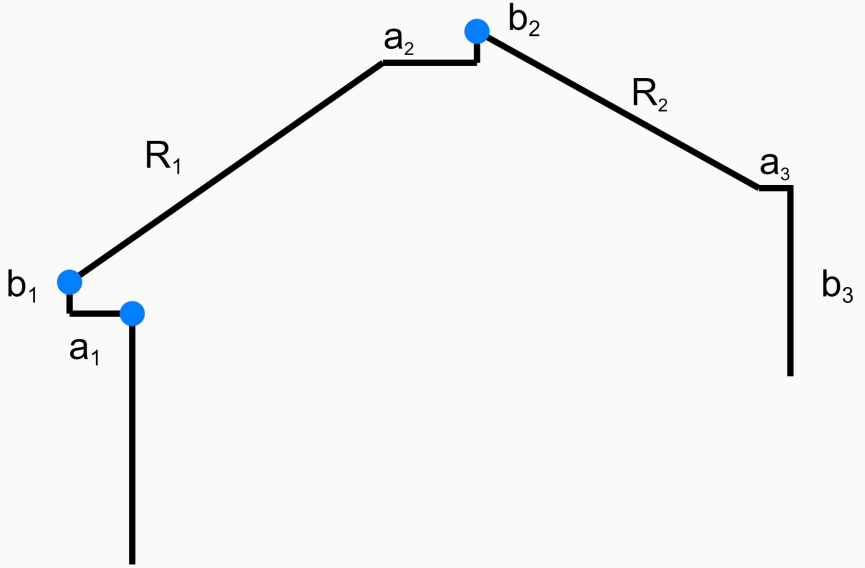
Assembly Process

After the completion of cutting the wood, the motors were installed in each part designated for them. The box containing the electronics was assembled, and a ball bearing was installed in its place to carry the robotic arm and allow it to move freely.Then the electronic parts and connections were assembled inside the box, and the camera holder was assembled and installed on the box, and the camera was installed on it.

Inverse Kinematics

Robot Model Equation

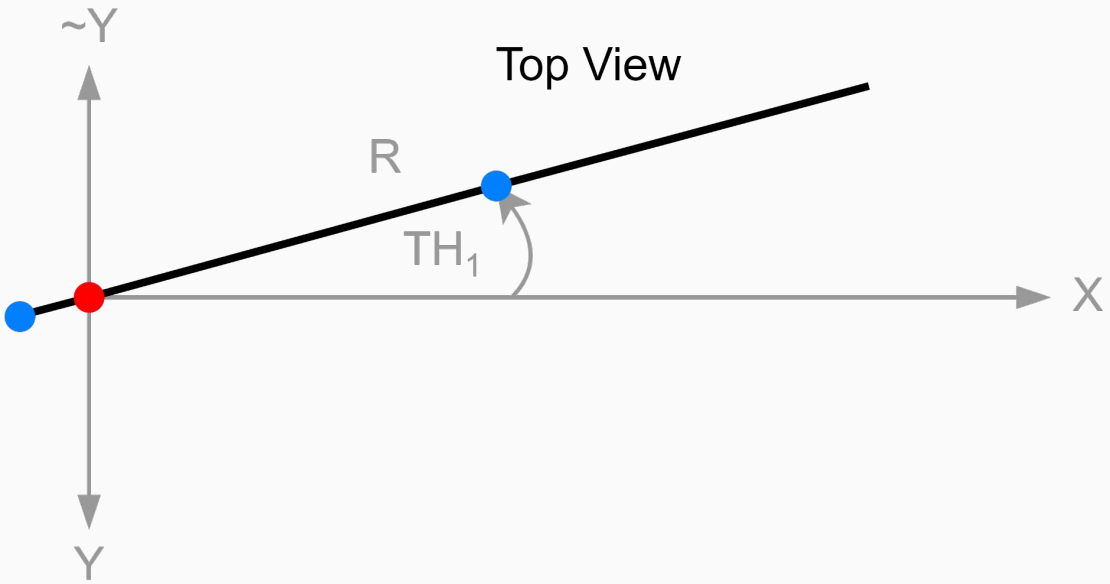




Robot Model Equation

$R = \text{sqrt}(X^2+Y^2)$

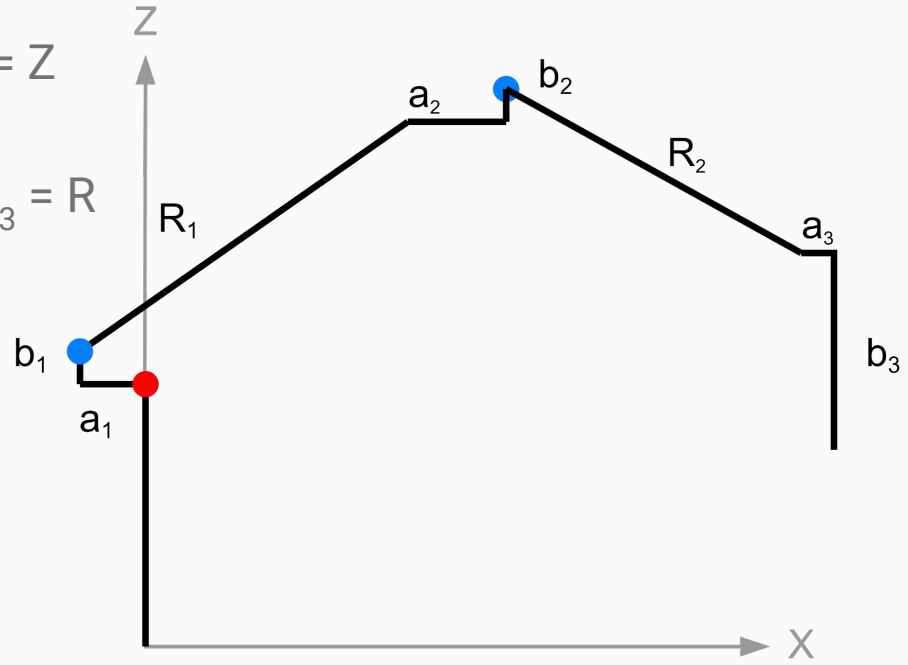
$Th_1 = \text{atan}(Y/X)$



Robot Model Equation

$$a_1 + (r_1 * \sin(th_2)) + a_2 + (r_2 * \cos(th_3)) - a_3 = Z$$

$$(-b_1) + (r_1 * \cos(th_2)) + b_2 + (r_2 * \sin(th_3)) + b_3 = R$$



Problems and Proposed Solutions

- Equation are more fitted for forward Kinematic.
- Equations are not linear to solve for inverse kinematics.

- Try inverse kinematics the analytical approach.
- Try a smarter solution give what we know about our system.

Vanilla Symbolic Solver Approach

Inverse Kinematics Unconstrained Equations

th2 = -2*atan((544000000*Z - 584000000*R + (38990073*(584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073) - (557640000*R(584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407)) / (2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073) + (544880000 *Z(584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000 * Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*Z2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073) + (2000000000*R2(584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)) + (2000000000*R2(584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073) + (2000000000*R2(584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073) + 76095120)/(- 2000000000*R2 + 13640000*R - 2000000000*Z^2 + 39120000*Z + 42417447))

th3 = 2*atan((584000000*R - 557640000*R*(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 544880000*Z(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 38990073(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) + 2000000000*R2(((2000000000*R^2 - 557640000*R + 2000000000*Z2 - 39120000*Z + 38861593)(- 2000000000*R^2 + 557640000*R - 2000000000*Z2 + 39120000*Z + 119986407))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073)2)(1/2) - 81415440))/(2000000000*R2 - 557640000*R + 2000000000*Z2 + 544880000*Z + 38990073))

7948

Total number of Characters in the equations

Systematic Constraints

Givens

- Motors range of movement is $0° \rightarrow 180°$
- Range of motion of interest is known
- Solution must be Real
- Solution must be positive

Constrained Symbolic Solver Approach

Inverse Kinematics Constrained Equations

Th_1 = atan(Y/X)

th2 = -2*atan((544000000*Z + (- 400000000000000000*R^4 + 2230560000000000000*R^3 - 800000000000000000*R^2*Z^2 + 1564800000000000000*R^2*Z - 1487127416000000000*R^2 + 2230560000000000000*R*Z^2 - 4362975360000000000*R*Z - 45238441278960000*R - 400000000000000000*Z^4 + 1564800000000000000*Z^3 + 1607192536000000000*Z^2 - 3173602723680000*Z + 4662862914366351)^(1/2) - 5320320)/(- 2000000000*R^2 + 13640000*R - 2000000000*Z^2 + 39120000*Z + 42417447))

th3 = 2*atan((584000000*R + ((2000000000*R^2 - 557640000*R + 2000000000*Z^2 - 39120000*Z + 38861593)*(- 2000000000*R^2 + 557640000*R - 2000000000*Z^2 + 39120000*Z + 119986407))^(1/2) - 81415440)/(2000000000*R^2 - 557640000*R + 2000000000*Z^2 + 544880000*Z + 38990073))

729

Total number of Characters in the equations

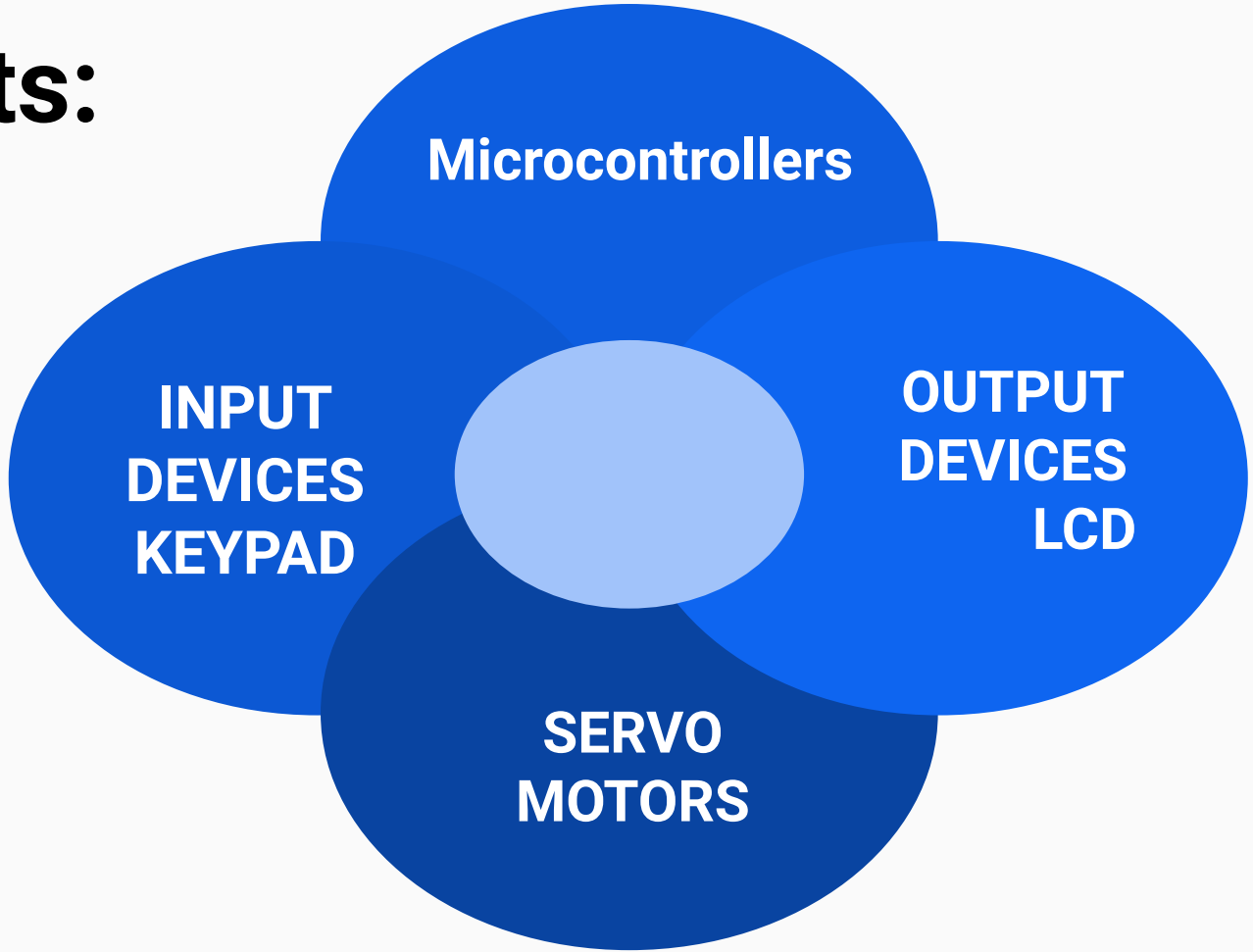
Inverse kinematic

Problem Solved

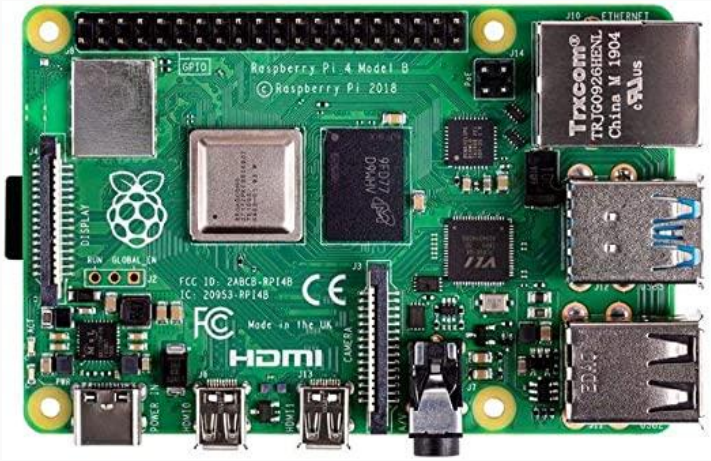
Embedded System Architecture

Components

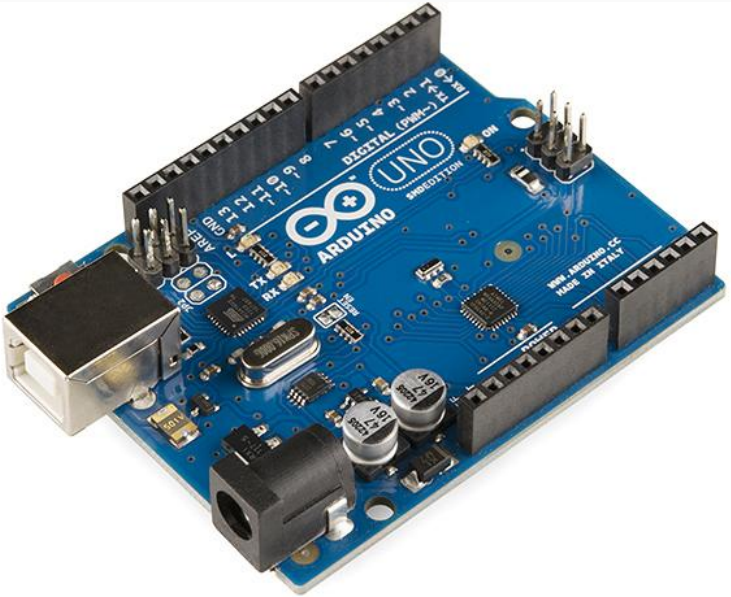
Components:


Microcontrollers
INPUT
DEVICES
KEYPAD
OUTPUT
DEVICES
LCD
SERVO
MOTORS

Raspberry pi 4 Model B



Arduino Uno



Why Arduino UNO?

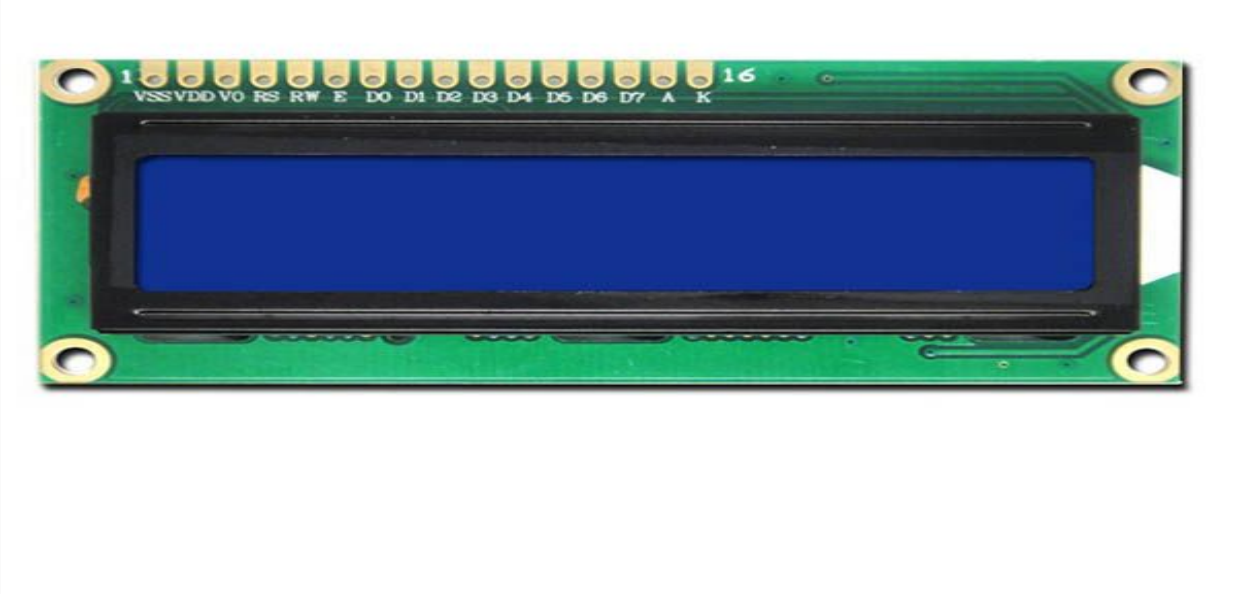
- Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board .
- Arduino Uno is programmed using the Arduino Software Integrated Development Environment (IDE).
- The board contains the **USB** interface support that enables the board to act as a serial device and provide the functionality to connect the board to other interfaces.
- Arduino Uno board has only one I2C module.
- The board contains PWM pins.

Why Raspberry pi 4 model B ?

- Huge processing power in a compact board.
- Many interfaces (multiple USB , I2C).
- Supports Python (making it easy to build applications).

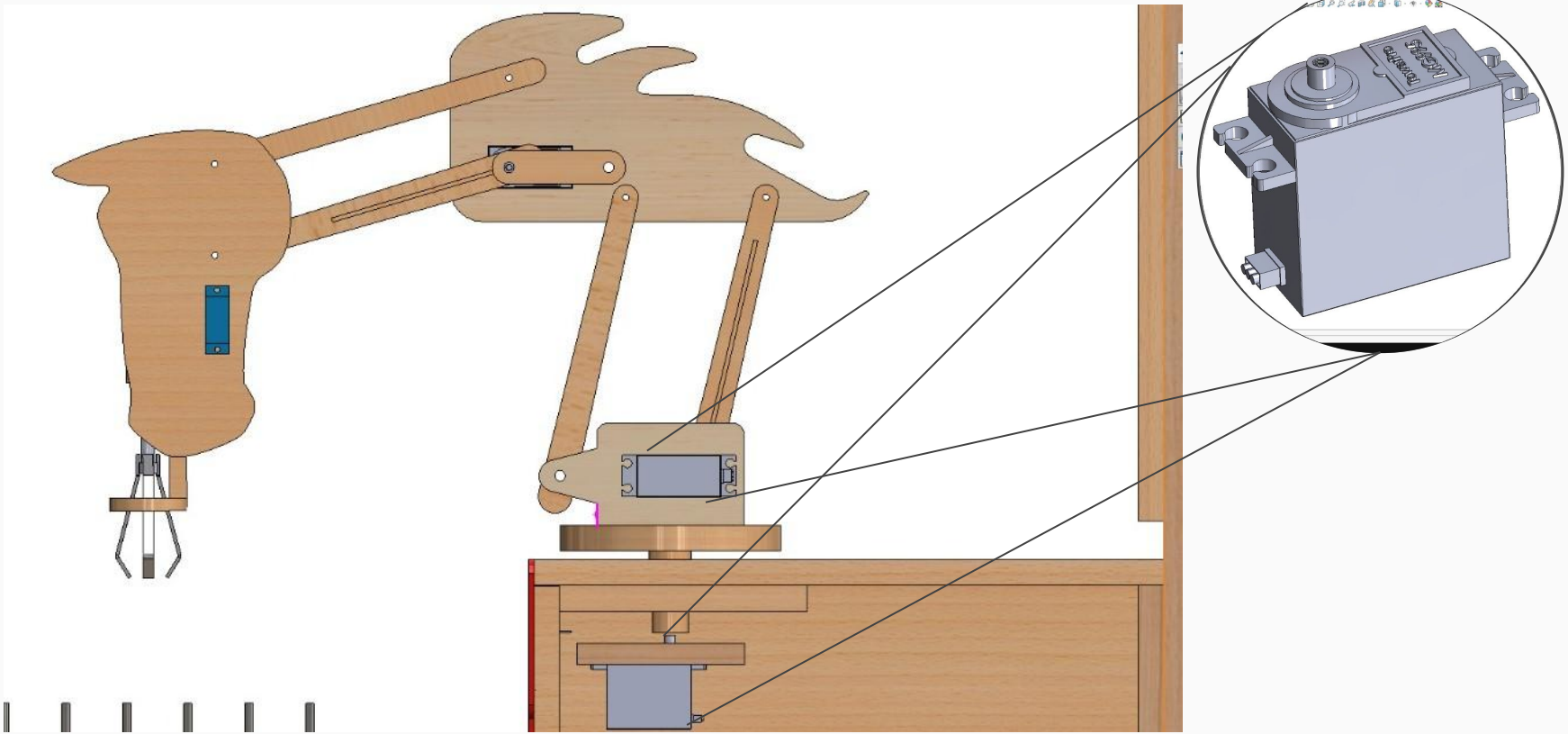
INPUT DEVICE: KEYPAD

OUTPUT DEVICE
LCD

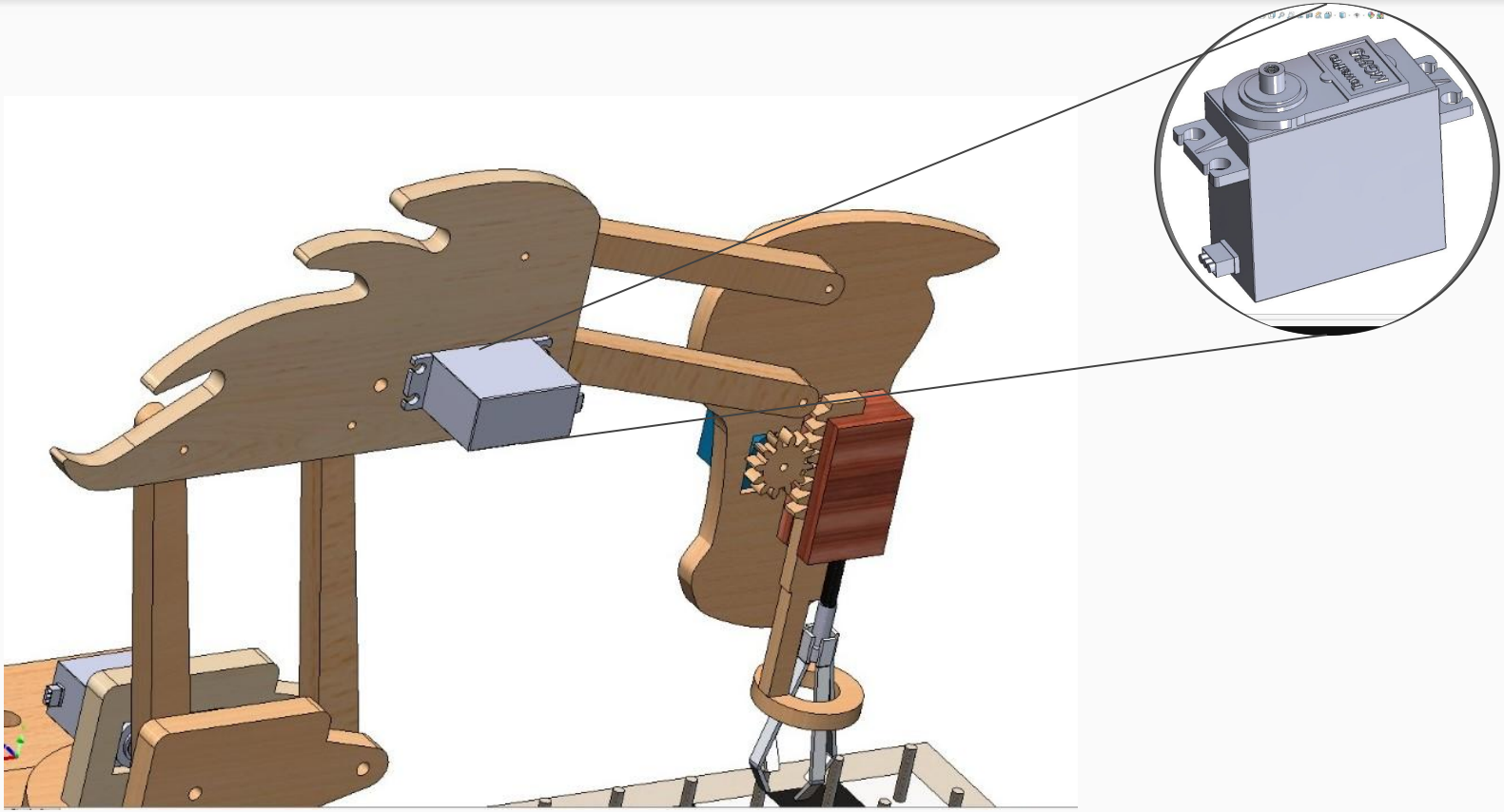


Servo Motors

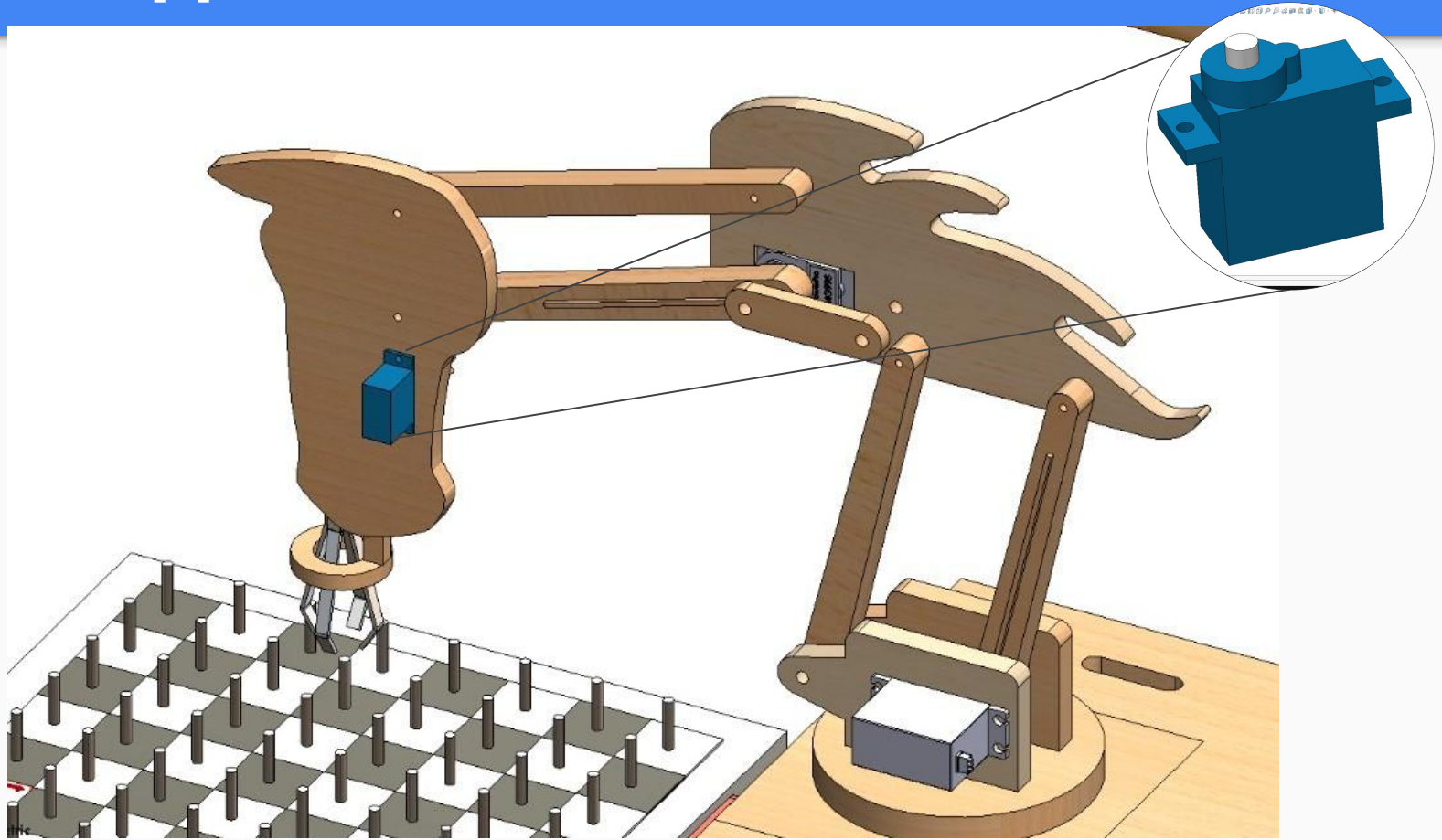
Base and Shoulder motor



Elbow motor



Gripper motor



Communication Protocols

Why to use I2C?

- Addressing
- Simplicity
- Better error handling mechanism

How did we implement I2C?

- Raspberry pi (Master)uses I2C to send chess notation to Arduino(Slave) to define the next move and move type.

Why to use USB?

- Compact Size.
- Speed.
- Reliability.
- Low power consumption

How did we implement USB?

- Arduino (Master)uses USB to send input devices data to Raspberry pi (Slave) to define the what color did user choose “white or black” and Mode “online or offline”.

Algorithm

Convert chess notation to Board index:

- Chess engine estimates “the next move” , by defining the square of initial position and the square of the final position and the move type.
- Raspberry pi send the next move arranged in an array of 5 characters using I2C protocol.
- Arduino uses “inverse kinematics ” to move pieces.

For example:

Initial square = d 1
Last square = a 4
Move type = ‘m’

Chess notation =
“d1a4m”

Move piece :

“g1f3m”

1

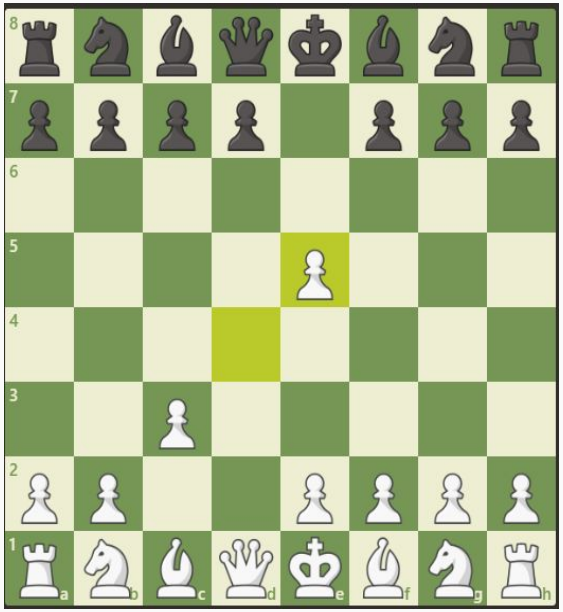
2

Elimination Algorithm:

Chess notation: “c4e5c”

‘1’

‘2’



‘3’

Image Processing

Problem Definition

Design & Training

- A need for detecting the board to collect data for ML

Runtime

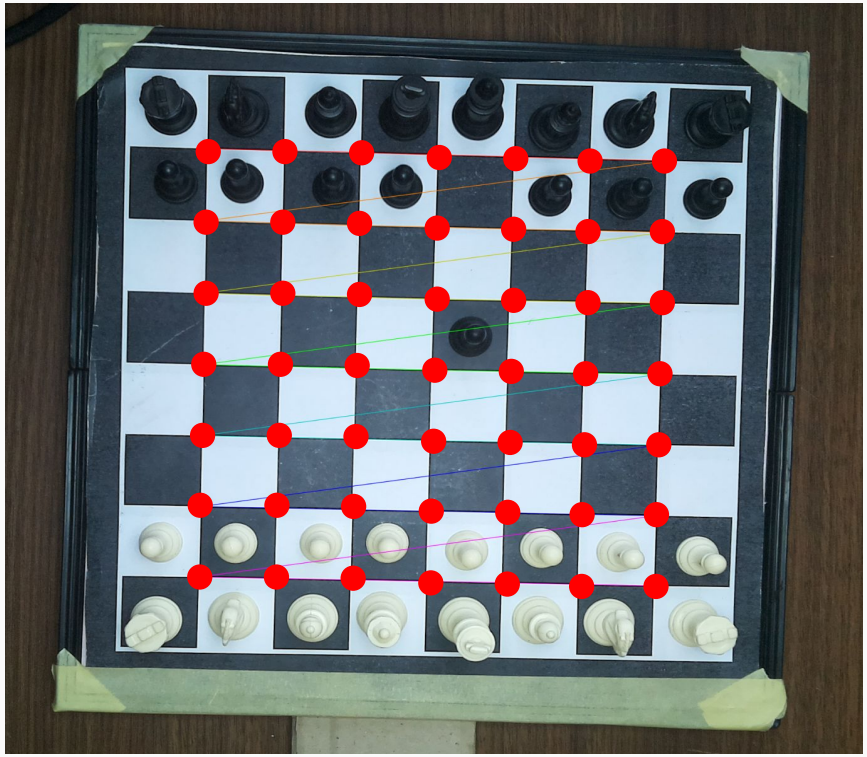
- Reading player input during the game

Capturing Images

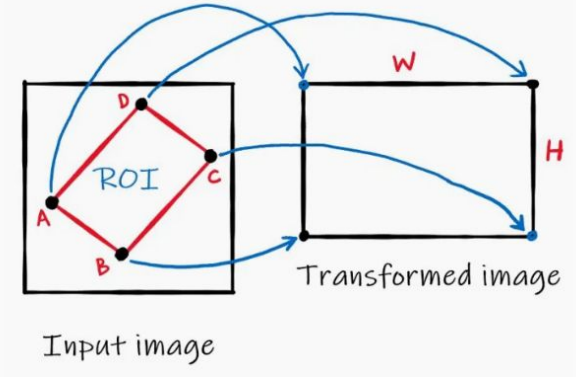
- OpenCV doesn’t use full resolution
- OpenCV only supports legacy mode

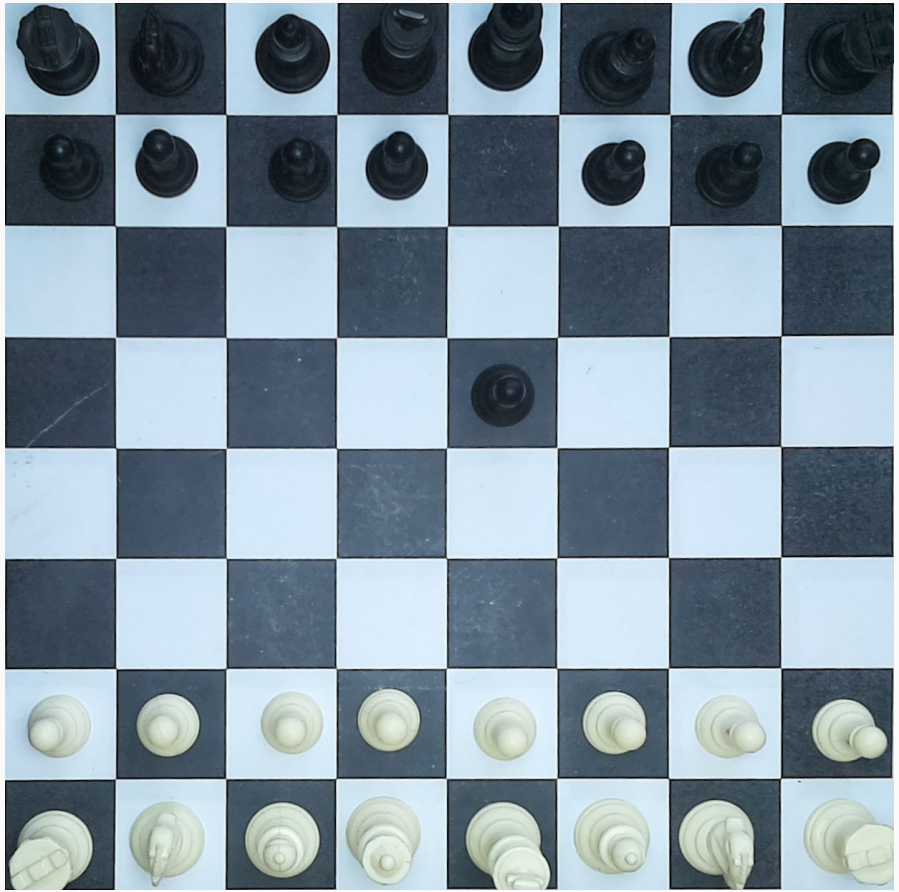
- Libcamera API
- Run shell command
- Read image using OpenCV

Finding Board



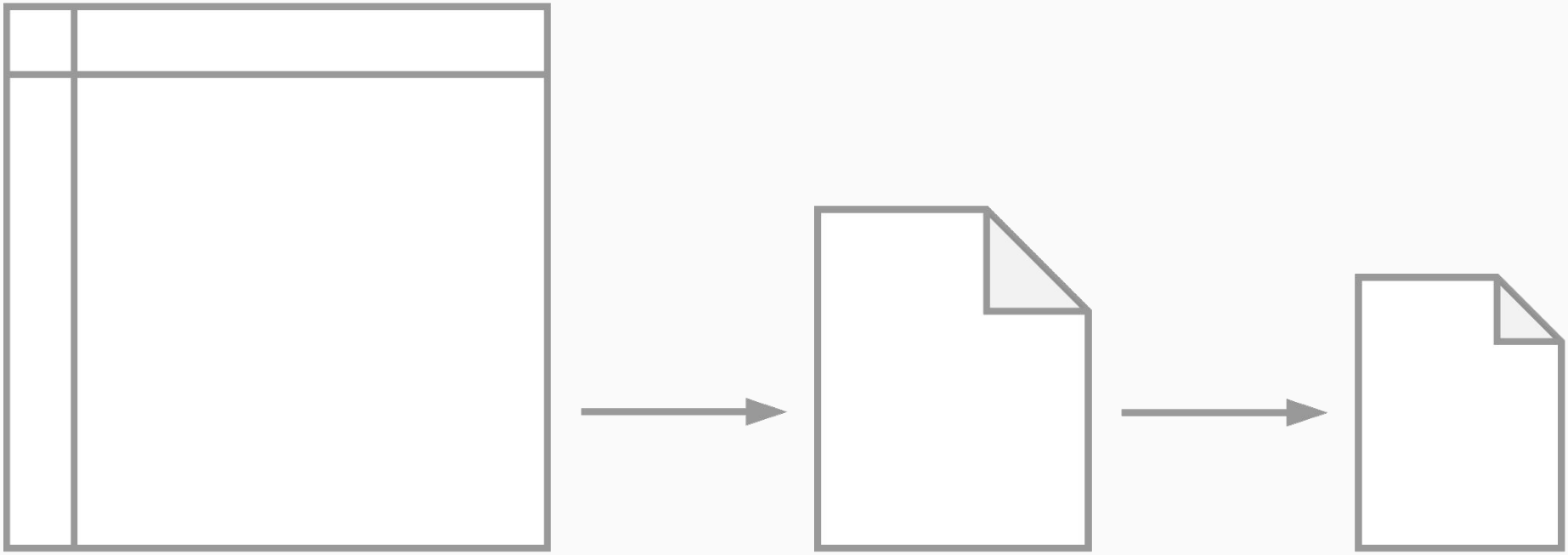
Perspective Warp Transformation





Tile Splitting

Numpy Compression



Full numpy matrix of images

10 GB

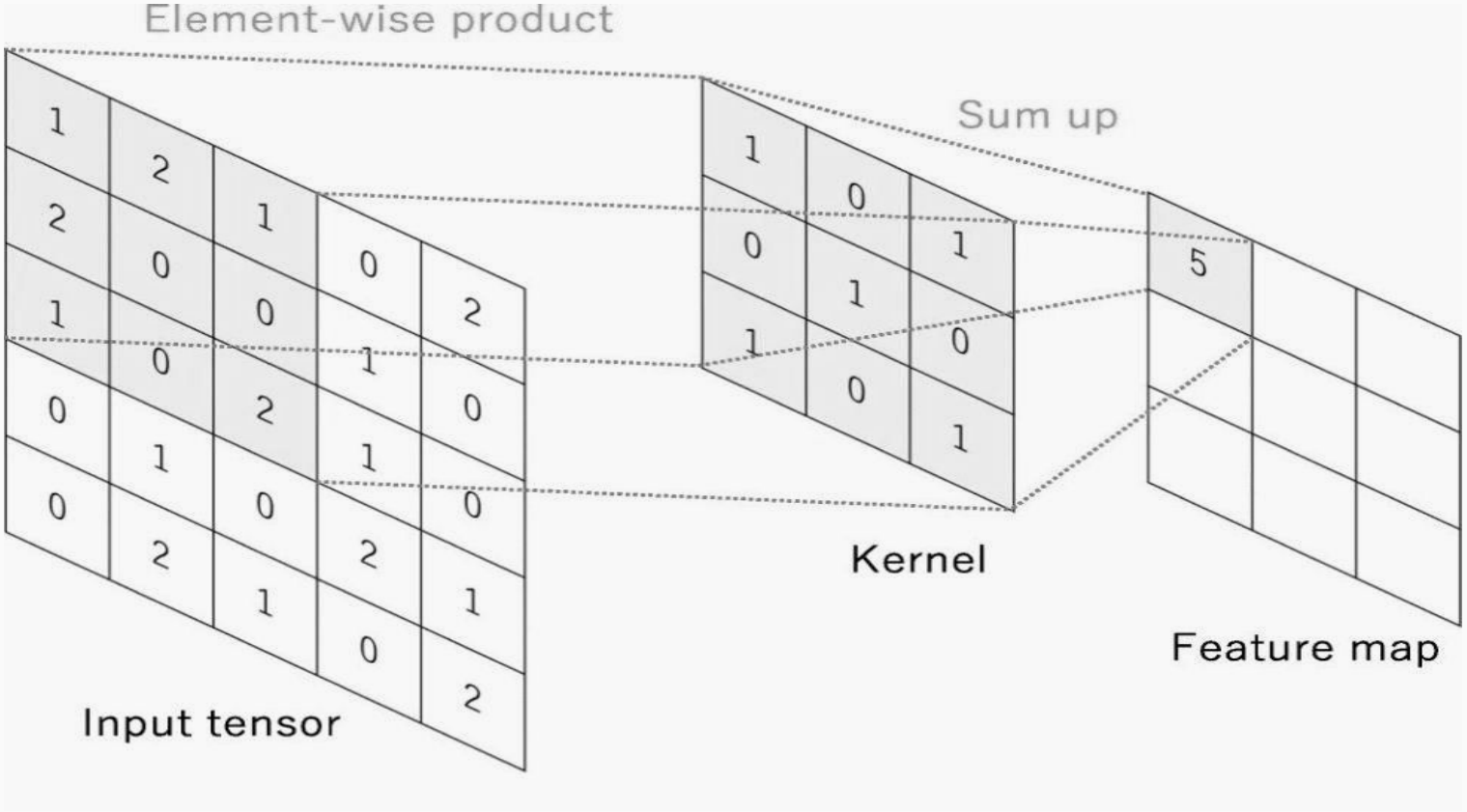
615 MB

Collected Data

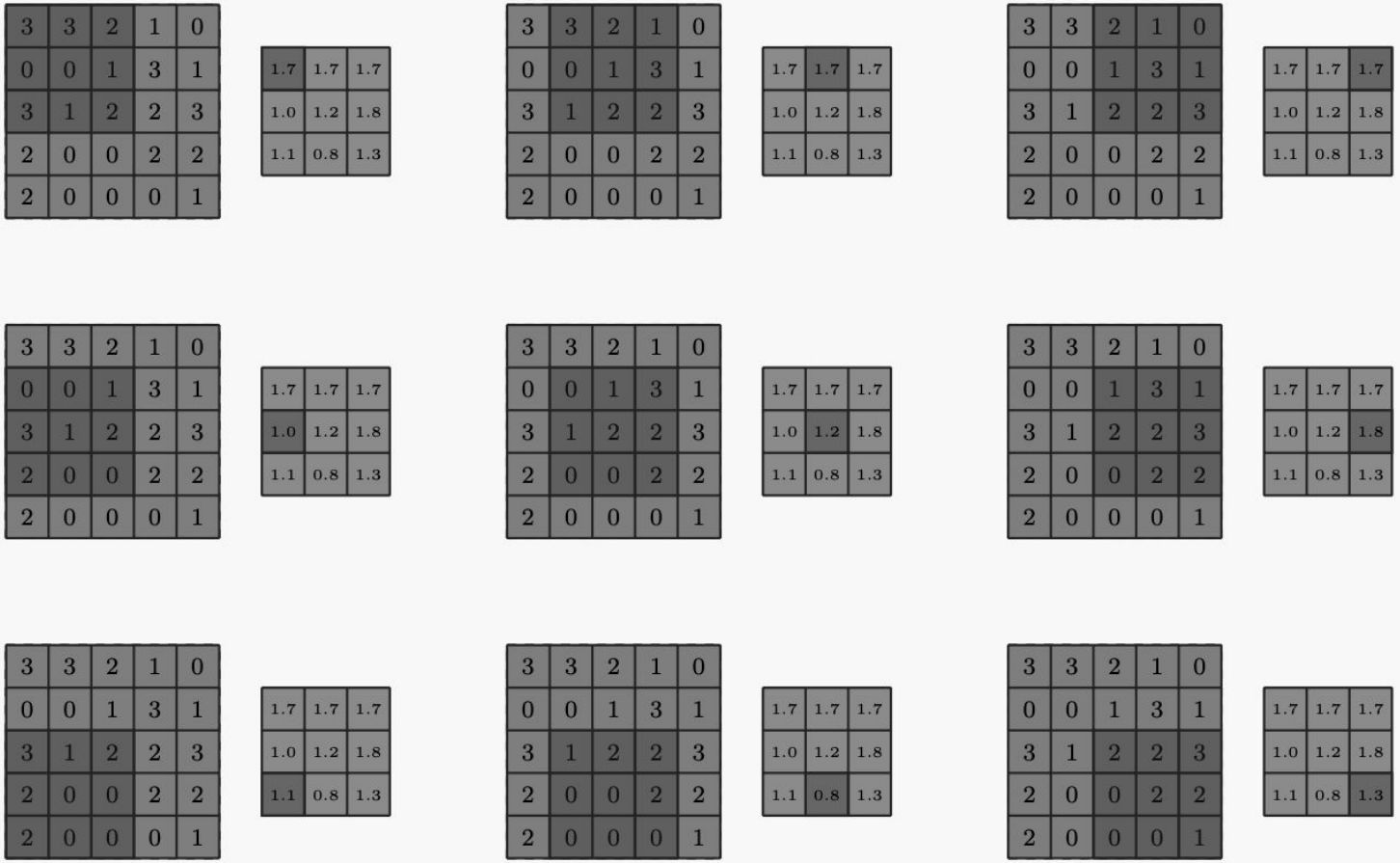
https://www.kaggle.com/datasets/mohamedmoataz99/wcra-ai

CNN Chess Piece Recognition

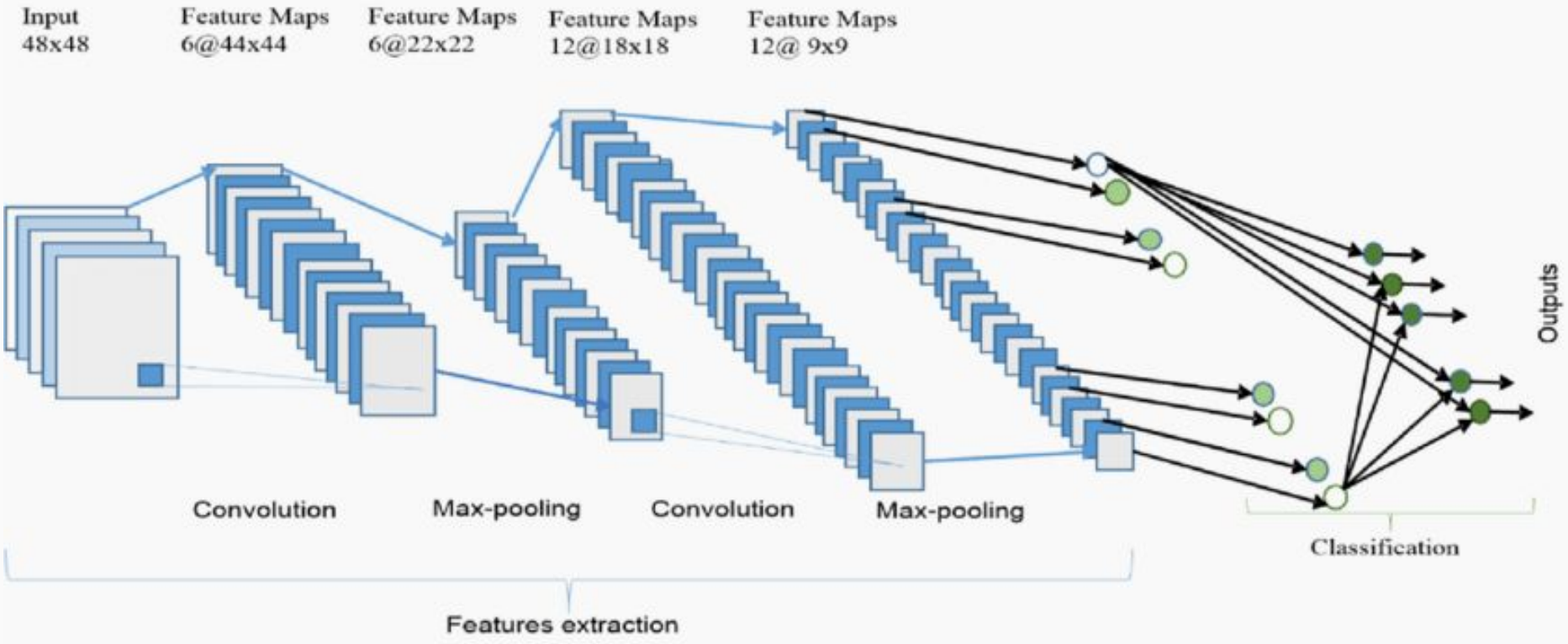
Convolution



Convolution



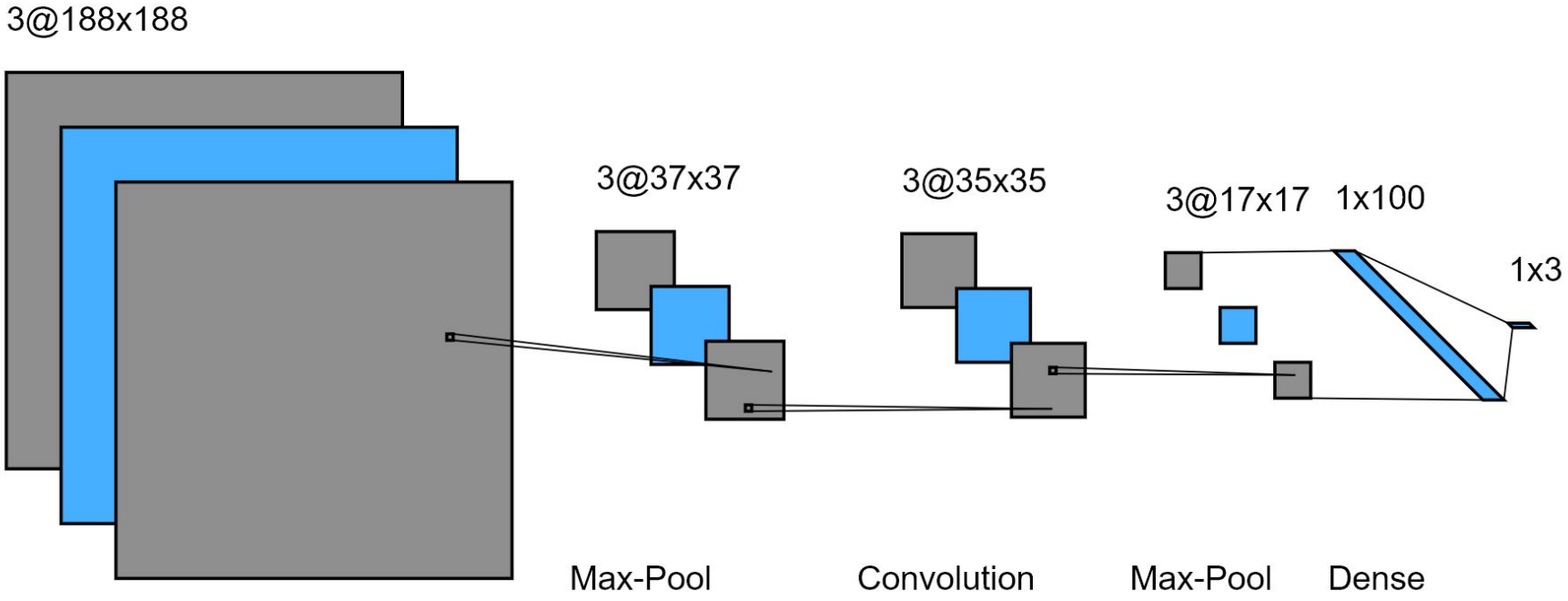
Convolutional Neural Networks



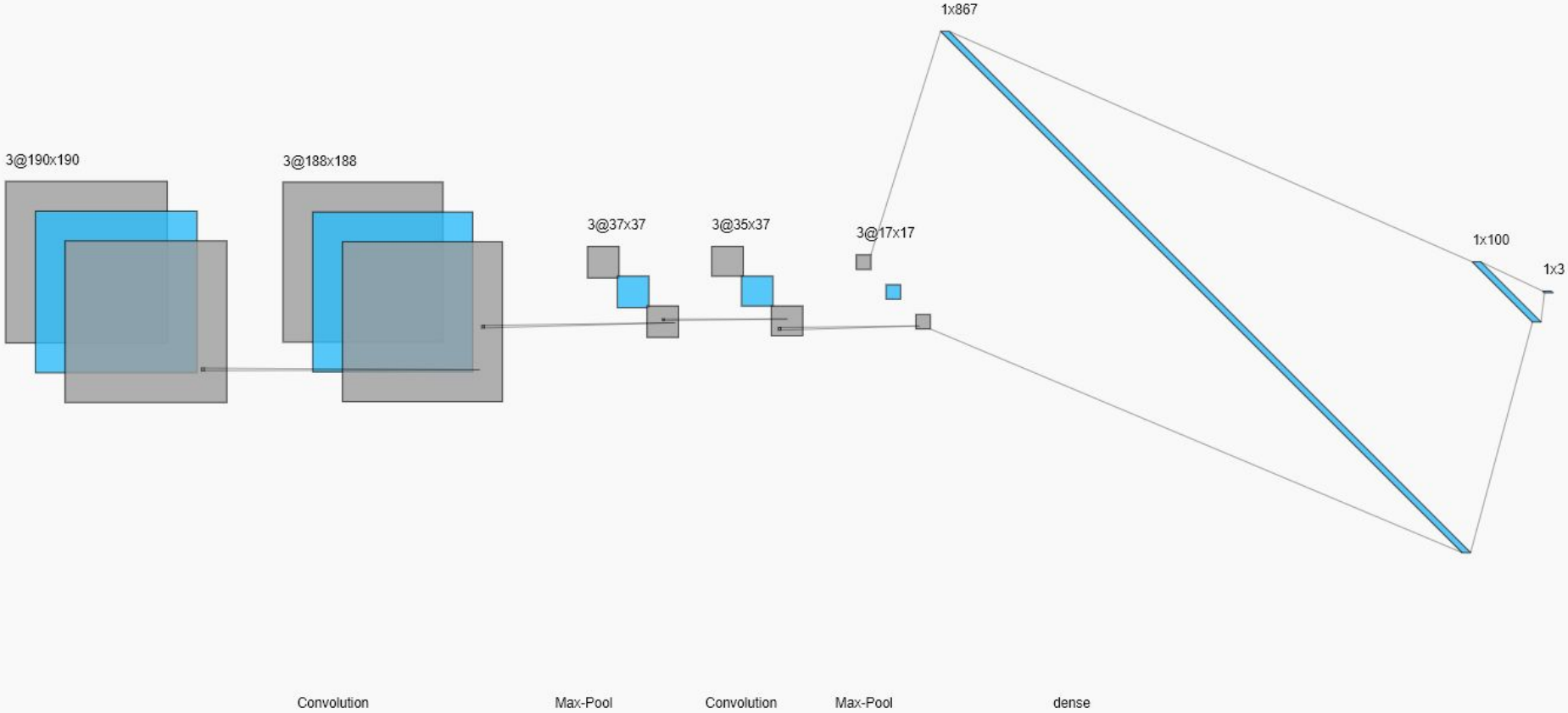
Motivation For CNN

- Takes advantage of fast matrix multiplication found in Gpus and Processing units
- Very suited for image-based pattern recognition
- embed a feature extractor in the neural network
- Reduces the need for complex classifier ANN

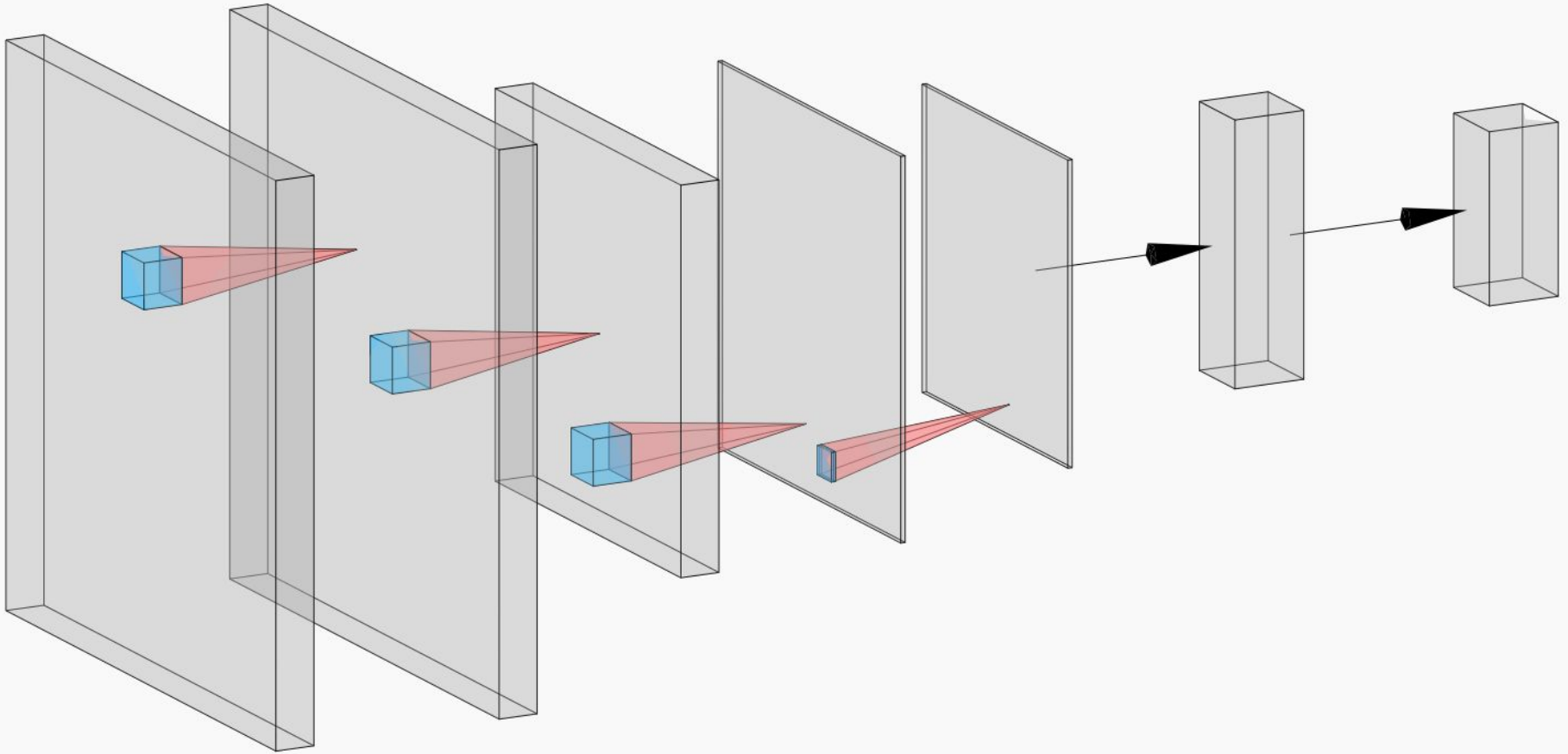
CNN Model Architecture



CNN Model Architecture



CNN Model Architecture



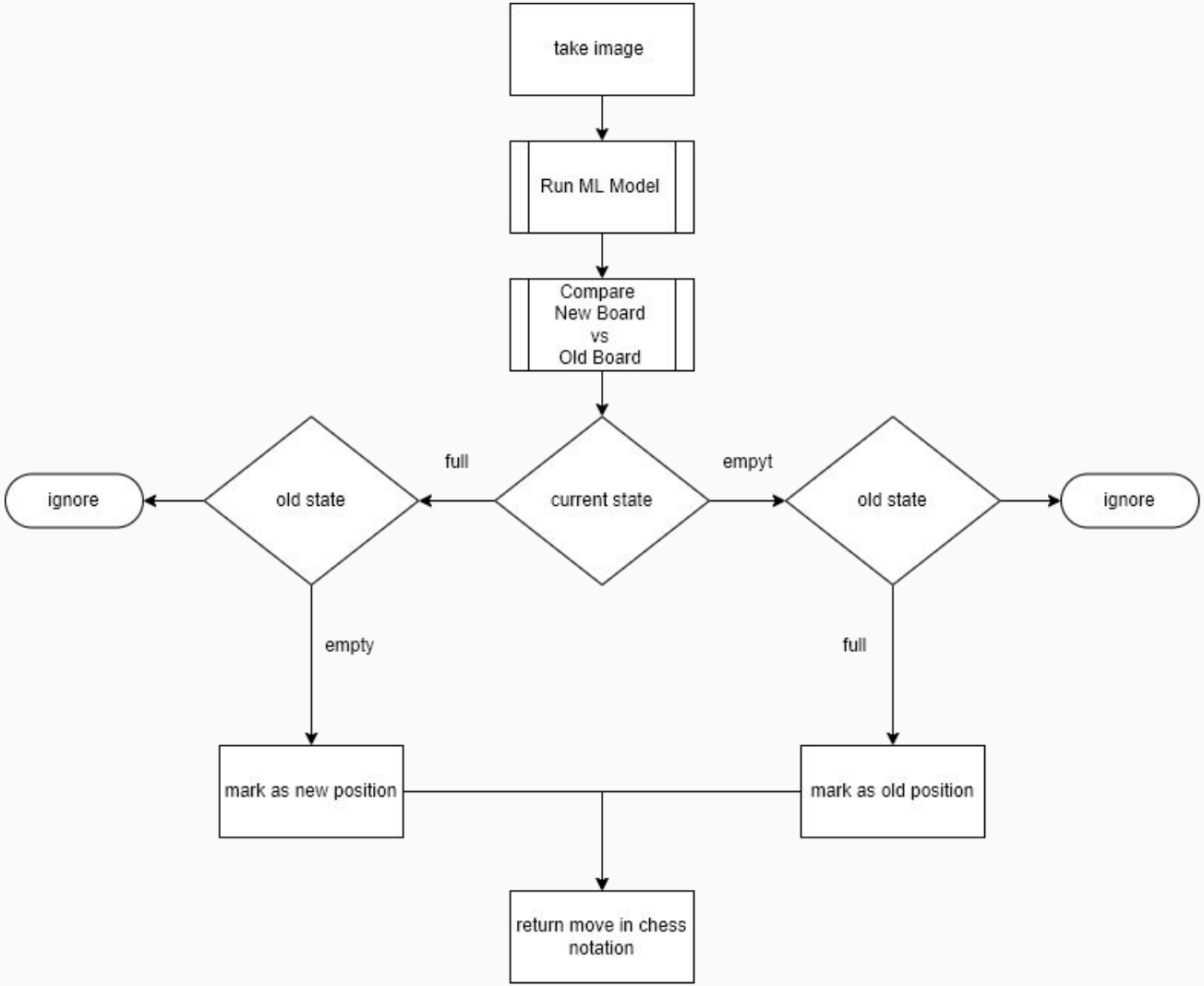
Model Training

- 20K training Image
- 75% - 25% training to testing data split
- 150 Batches
- 100 Epochs
- 5 Hrs Training Time
- 87K Parameters

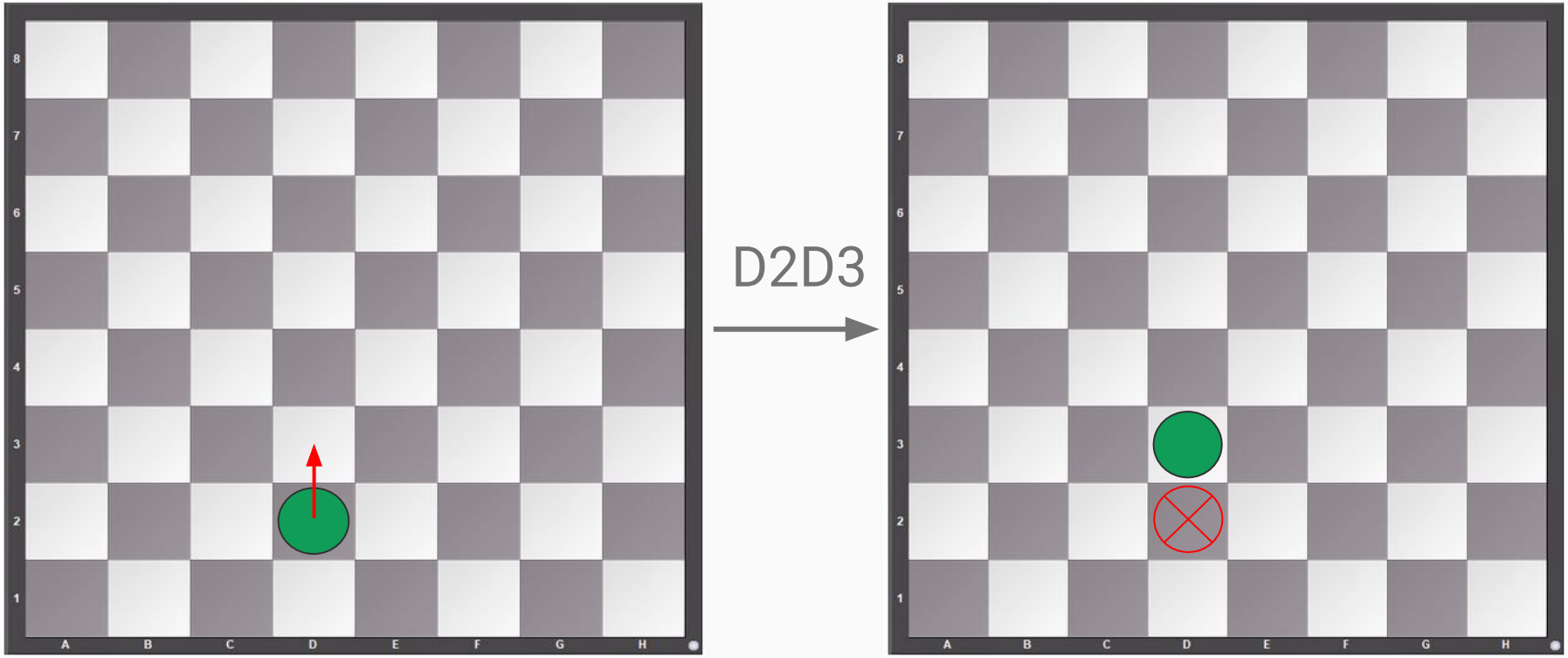
99.14%

CNN Piece Recognition Accuracy

Move Detection


take image
Run ML Model
Compare
New Board
vs
Old Board
current state
full
empyt
old state
ignore
old state
ignore
empty
full
mark as new position
mark as old position
return move in chess
notation

Move Detection

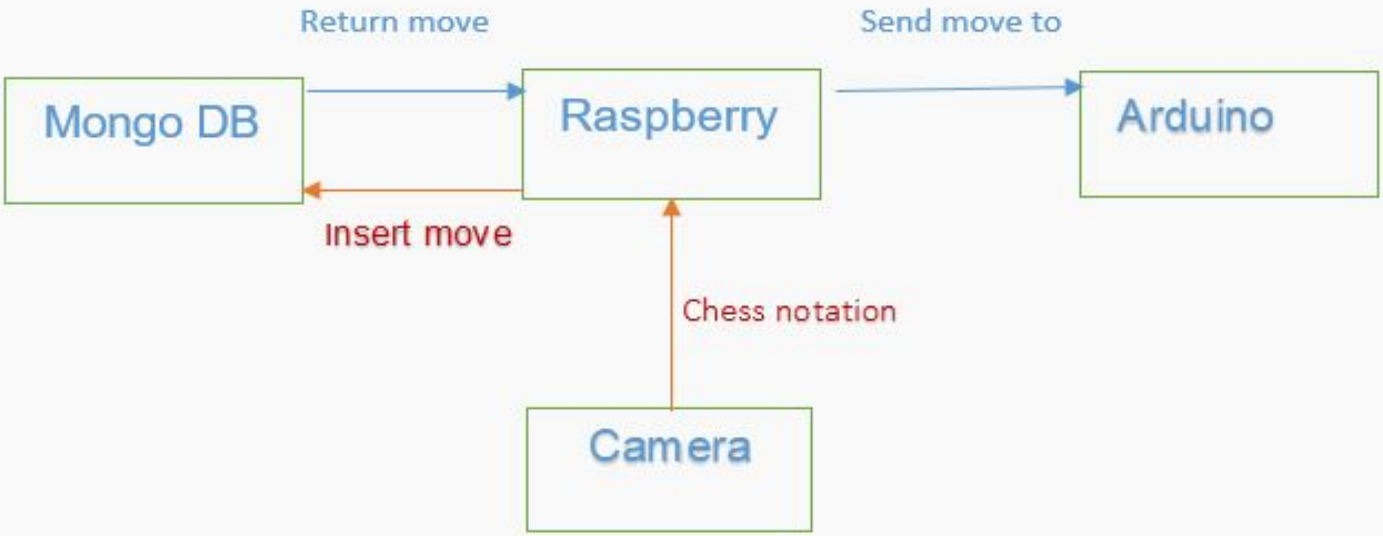


IoT

Problem definition

- Need to transfer human move from Raspberry pi that come from camera to database and return online move to Raspberry to allow the robotic arm to make the move in the board .
- How to connect with the Web application ?

Problem flowchart



Proposed solution

-Using IOT

The Internet of Things (IoT) is a scenario in which objects or people are provided with single identifiers and the capability to automatically transfer data more to a network without requiring human-to-human or human-to-computer communication.

- Raspberry pi and Web application connection

We use python language with pymongo library to connect to MongoDB database

Communication structure

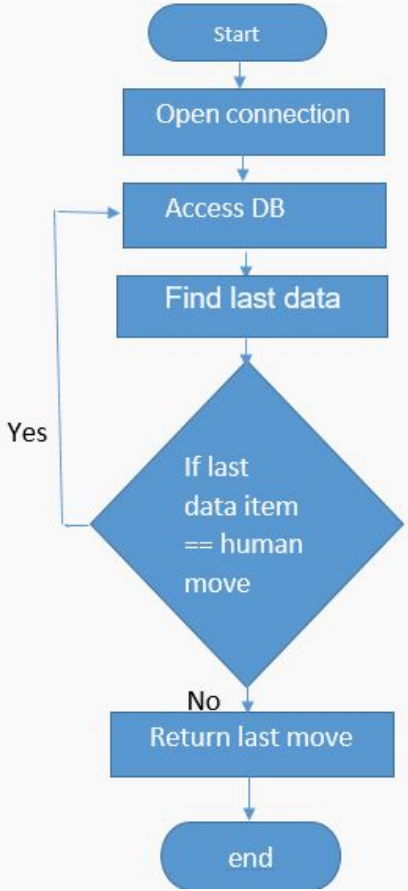
- Open connection with MongoDB.
- Access DB.
- Find and return all document from DB.
- Check the last move.
- Insert data in DB.

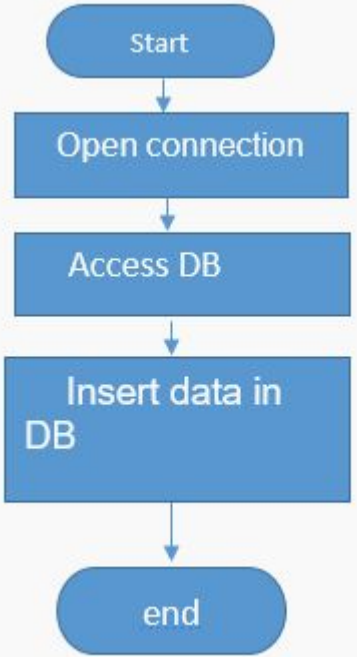
Open connection with MongoDB.

To connect with MongoDB use the URL and password of the user

(mongodb+srv://arm:123321@moves.yxjpv.mongodb.net/?retryWrites=true&w=majority)

Inserting and Retrieving Last Move





Main improvement

Determine the type of move

- Kill
- Move
- Promotion
- Casel

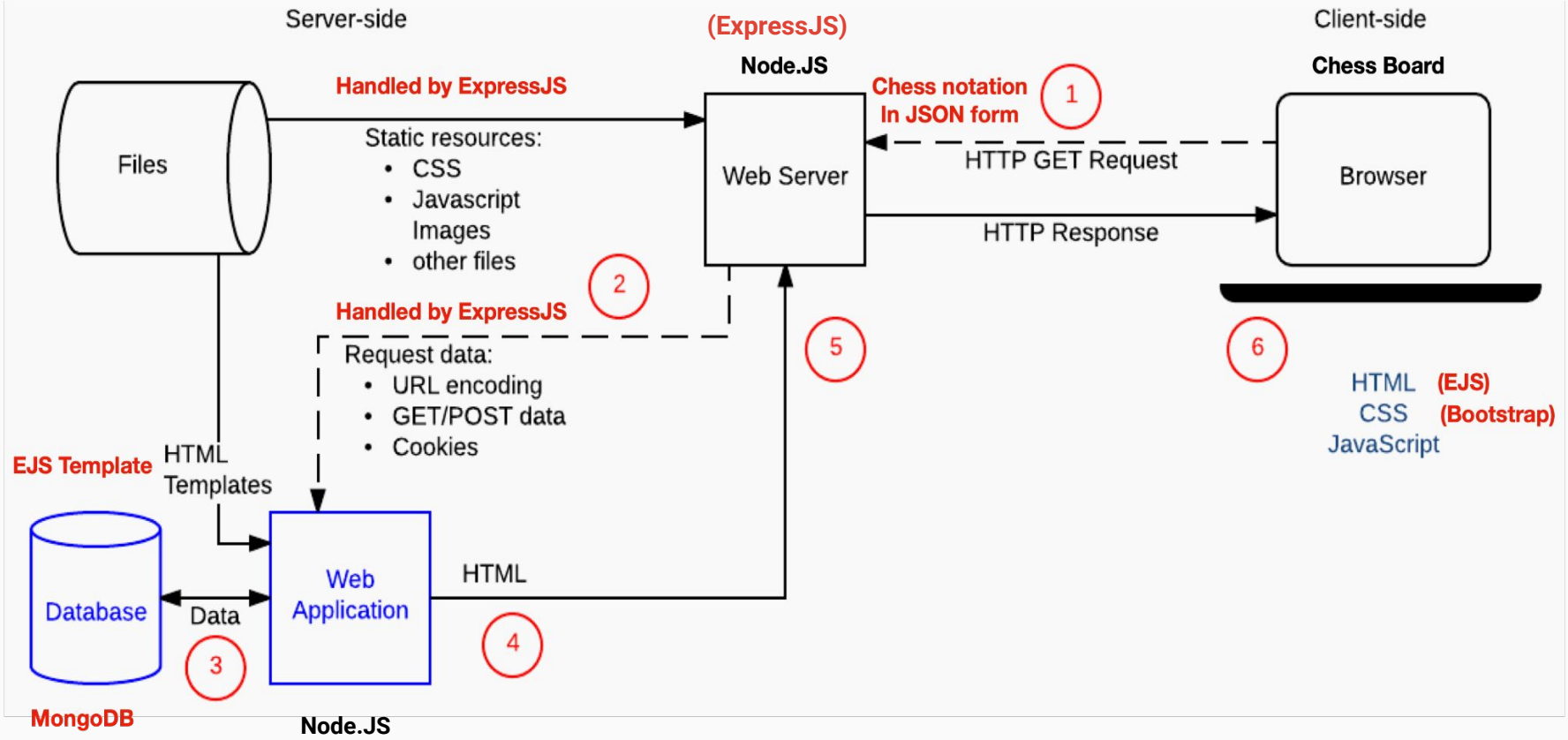
Future improvement

- Store lot of matches in DB to be used for chess engine to improve (tarin) evaluation function
- Save picture that take by camera to improve CNN

;b

The Website

Overview


Server-side
(ExpressJS)
Node.JS
Client-side
Chess Board
Handled by ExpressJS
Static resources:
• CSS
• Javascript
Images
• other files
Files
Web Server
Chess notation
In JSON form
1
HTTP GET Request
Browser
HTTP Response
2
Handled by ExpressJS
Request data:
• URL encoding
• GET/POST data
• Cookies
5
6
HTML (EJS)
CSS (Bootstrap)
JavaScript
EJS Template
HTML
Templates
Database
Data
Web
Application
HTML
3
4
MongoDB
Node.JS

The UI

- Used Figma for designing
- why figma ?

Then design :

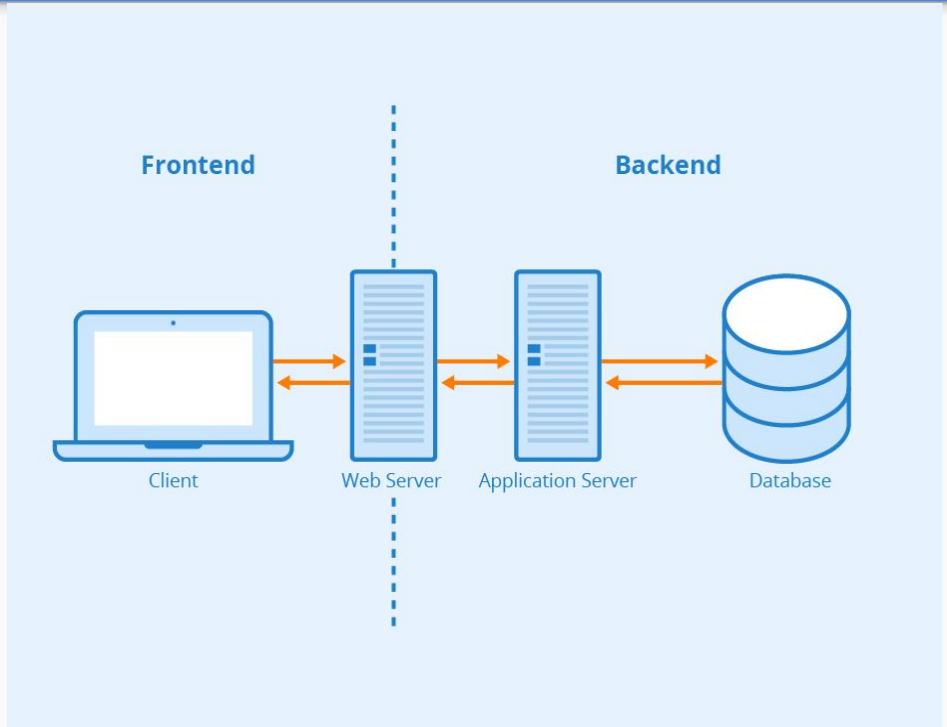
- Playboard
- Get button
- Rules of play

JS structure

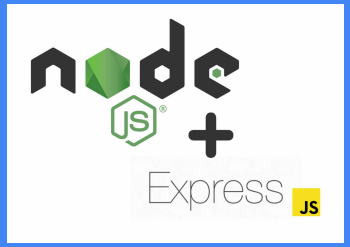
- Creating Chessboard
- Drag and Drop
- Get possibilities
- Show possibilities UI
- Move, Eat Pieces
- Castling and Switching
- Is checkmate
- Update UI
- Set Timer
- Match History

```
const chessboardParent = document.getElementById("chessboard"); //point to the playboard div

// Chess Game
class Chess {…
}

// Chess Board
class Board {…
}

// Chess Piece
class Piece {…
}

// Chess Square
class Square {…
}

// Player
class Player {…
}



const Game = new Chess(); // game

Game.init(function () { // intializing the game
    const X = [
    { piece: "bp4", from: "e7", to: "e5" },
];

    this.start();
    this.insertToMatchHistory("./assets/javascript/json/matchhistory.json");
    for(;this.info.turn == this.data.players[1];){
        console.log('hi');}
    for(;this.info.turn == this.data.players[0];){
        console.log('hi');
    }

}); // initialize
```

BackEnd


Frontend
Backend
Client
Web Server
Application Server
Database

Node.js & Expressjs



Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications.

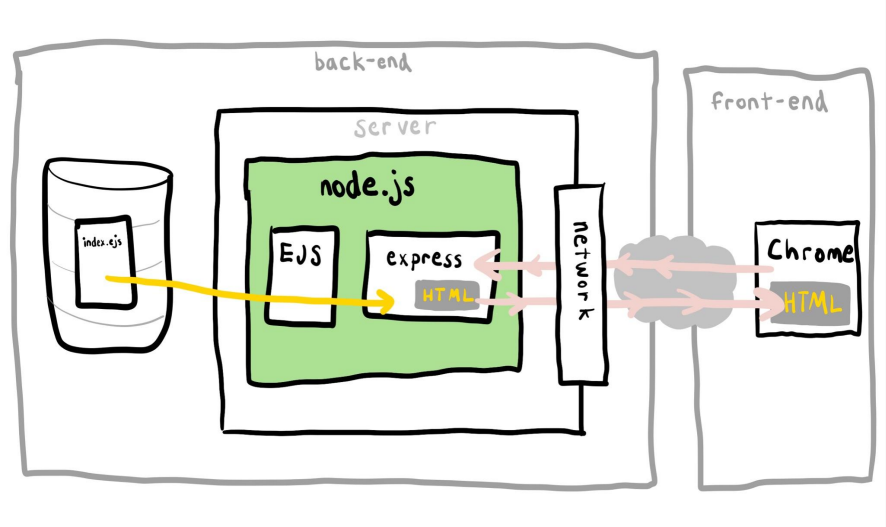
Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

```
"dependencies": {
   "body-parser": "^1.20.0",// to handle HTTP requests body
   "ejs": "^3.1.7", // for faster HTML loading
   "express": "^4.18.1",//fastest framework framework
   "lodash": "^4.17.21",//Gived extra functions to deal with arrays
   "mongoose": "^6.3.3",// helps to handle MongoDB functions and Schema
   "node-fetch": "^3.2.4"// for making a local domain for testing
 }
```

ExpressJS

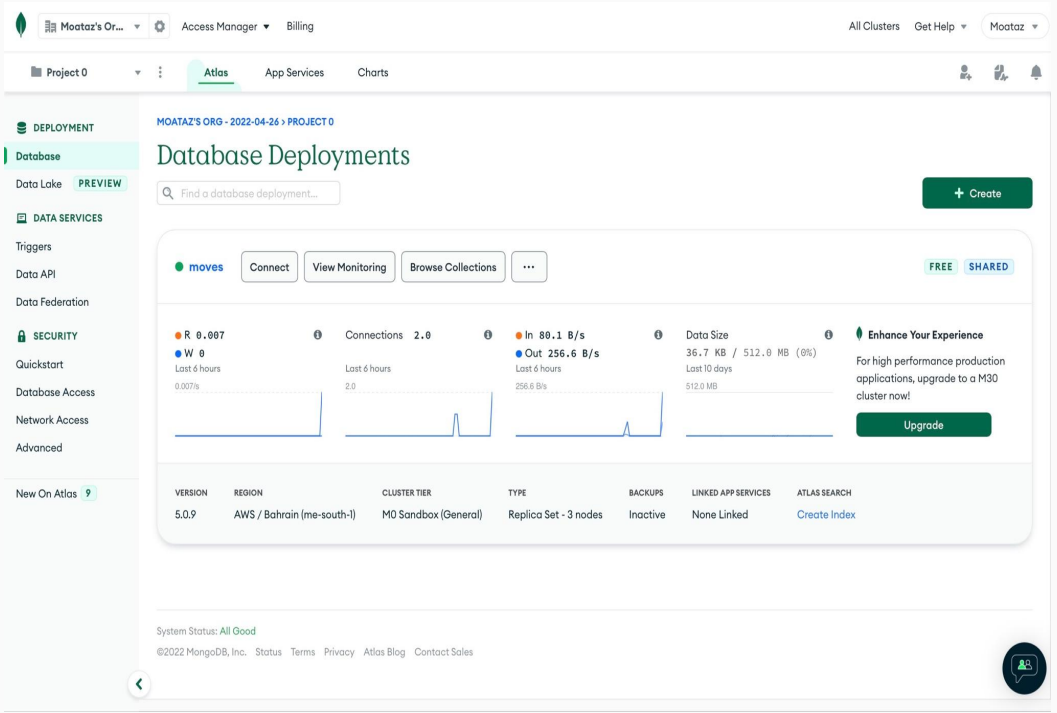
- Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application.

- It's a layer built on the top of the Node js that helps

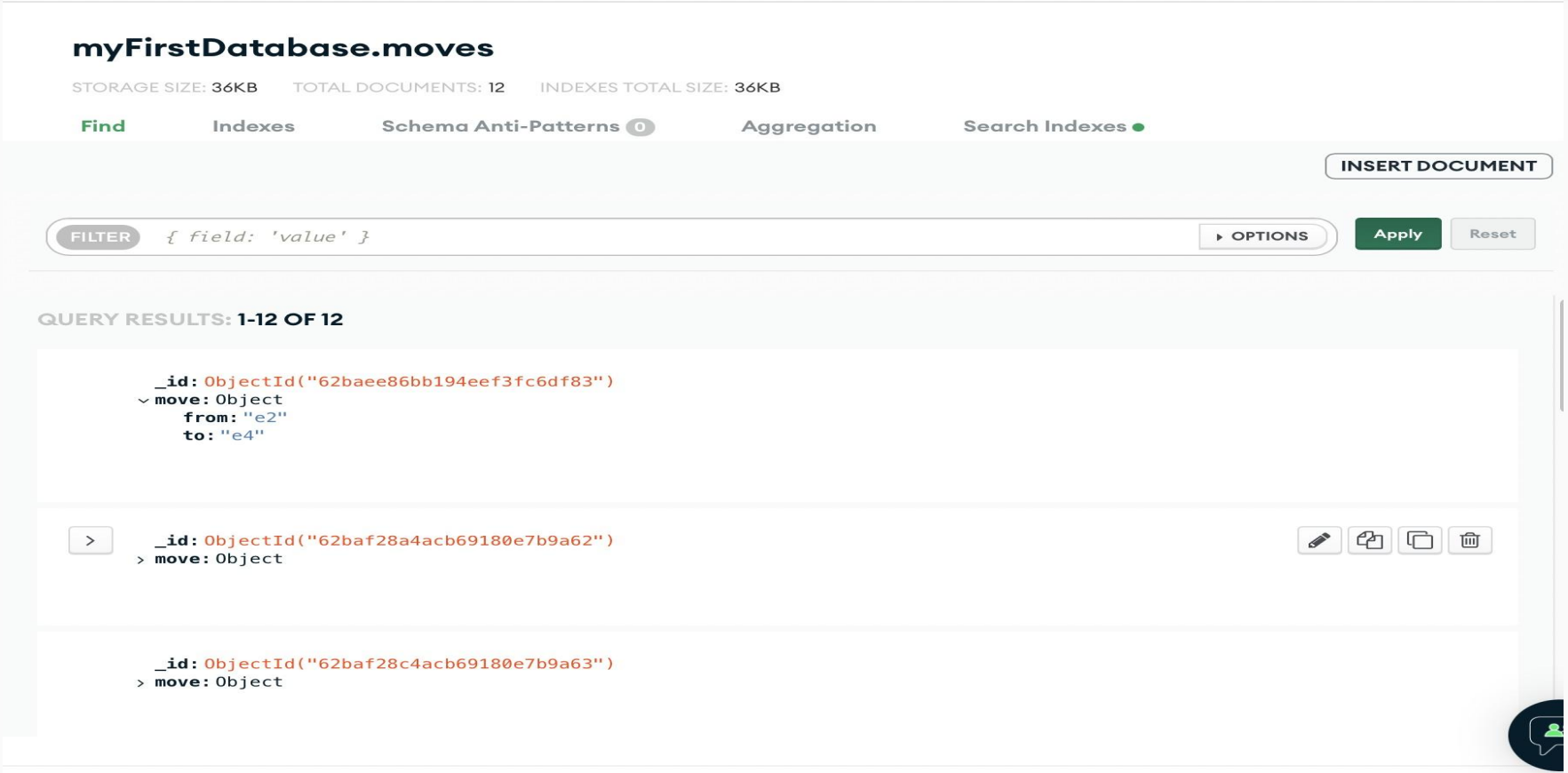


MongoDB

is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.



MongoDB schema



Dependencies

```
const moveSchema = new Schema ({ // create a new Schema
    move : {
        piece: { /// the piece
            type: String, // type of data
            required: true // it's requiered to  send the data
        },
        from: { // from where the piece moved from
            type: String,
            required: true
        },
        to: { // to where the piece moved
            type: String,
            required: true
        }
    }
```