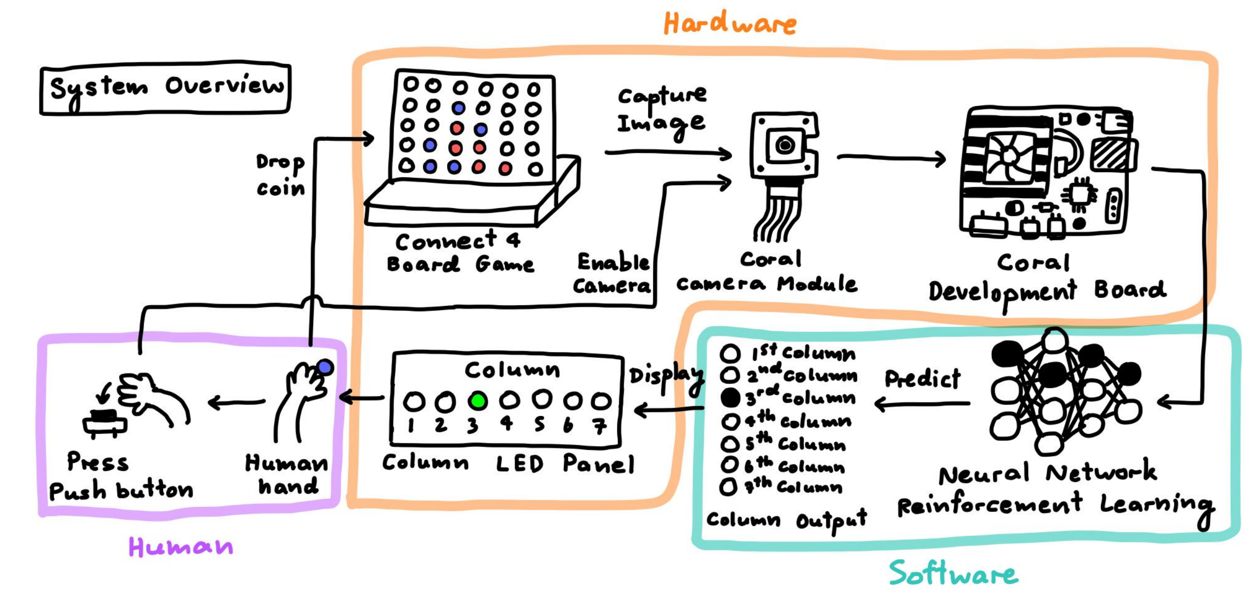


Connect4Bot

Nithid Mahattanasin 6314552737

Nut Wongwanwatana 6314552729

Patipon Petchtone 6314552745

Host : Kasetsart University

Overview

Connect4 is a two-player connection board game, in which the players choose a color and then take turns dropping colored discs into a seven-column, six-row vertically suspended grid. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs.

For this project, we try our best to produce a good Connect 4 player by using Reinforcement learning. The created player will learn from experience, likes humans do, finally, he can know what 'good' and 'bad' board positions look like.

We will embed this system into the embedded device which can work standalone. The system will get the input board image from the camera and output the column number to drop a disc.

Background Knowledge

1. **Reinforcement learning**

   Reinforcement learning is the core module of this system. It is the brain that computes and generates the output. Basically, Reinforcement Learning is the method to learn from the interaction between the agent and the environment. There are several kinds of Reinforcement Learning that can be applied in this project such as Q-Network, Policy Gradient, and AlphaZero.

2. **Image processing**

   This module is the only sensor of this system. RGB image will be sent from the camera and be processed in this module to get the board positions for our agent.

3. **Edge AI**

   As we know, image processing and neural network are required a lot of computational power. We need to embed these things into an embedded device which have limited resource. We need to find a solution to optimize our system to be able to run on an embedded device.

Goals

1. The created player can understand the Connect4 and beat normal people like us.
2. The system can work standalone in an embedded system.

Hardware design

1. **Coral Development board**

   The Coral Development board has a good performance of CPU and GPU unit. In addition, it has ML accelerator, so this is a good board to create on-device ML products.

   Specification

   CPU : Quad Cortex-A53, Cortex-M4F
   GPU : Integrated GC7000 Lite Graphics
   RAM : 1GB LPDDR4
   Flash memory : 8GB eMMC

2. **Coral camera module**

   A 5-megapixel camera module that's compatible with Coral boards. Connects through the MIPI-CSI interface, and provides an easy way to bring images to the system.

   Specification

   Resolution : 5-megapixel OmniVision sensor
   F-number : F=2.2 ±5%
   Focal length : 2.5mm
   Focusing range : 10cm ~ ∞
   Output : Dual lane MIPI output interface

3. **Push-button switch and LED panel**

   The panel of LED will be used as the output to show the decision of the AI player while the push-button switch is used to tell that the user has already done their turn. Therefore, This panel is composed of 7 3mm LEDs (7 column decision) and 1 push button switch. For the connection, this panel uses 8 GPIO ports (7 ports for LED and 1 port for button) which have voltage around 3.3V and can provide enough power to turn on a LED.

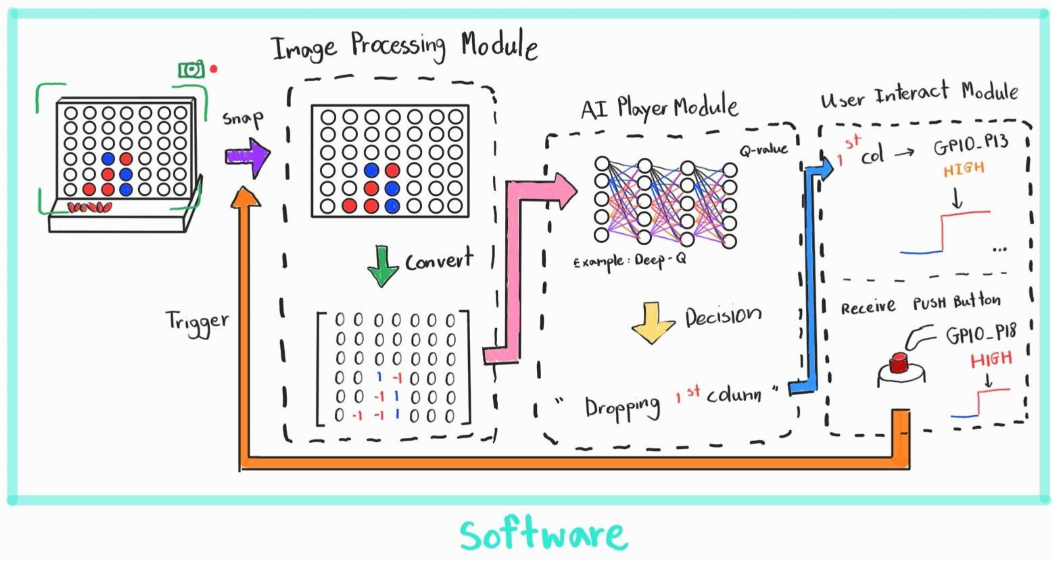
4. **Power supply**

   Use a USB Type-C power port (5V DC) to supply the Coral Development board, GPIO, and Coral camera module without another power supply separately, because 3mm LEDs are low power consumption and not operate all at the same time.

5. **Connect4 Board**

   Use 6x7 common Connect4 board, 2 colors of a token, 21 tokens in each color.

Software design



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Hardware Control and Input/output

1. **User Interact module**

   Purpose: This module will **interact with the user (human player)** such as transforming the **target column** from AI Player to **LED signal** (So, we will know which column is chosen by AI Player) and receive **a push-button signal** when the user would like to end their turn.

   Main Tools: GPIO from python-periphery lib.

2. **Image processing Module**

   Purpose: This module will be used to do **image processing tasks** to convert from the **picture of the connect4 board** (Environment) to the **current state of the board** (state). This information is used as an input for the AI Player module.

   Main Tools: OPENCV.

Artificial Intelligence

1. **AI Player module**

   Purpose: After the end of the user's turn, This module will receive the state of the board as input and select **which column will be the next one**. The core of this module is using an agent which is trained by **reinforcement learning methods**. So, the agent can learn and know which column will be the best one at that state.

   Main Tools: Tensorflow and Tensorflow Lite.