Use Case

With our system, it is possible to play card games over the internet with physical cards.

Use Case Requirements

- Play with physical cards
- Plays the games Go Fish, Euchre, and Rummy
- Multiplayer support up to 5 players per game
- Be able to input any card for game logic
- Ability to have concurrent games

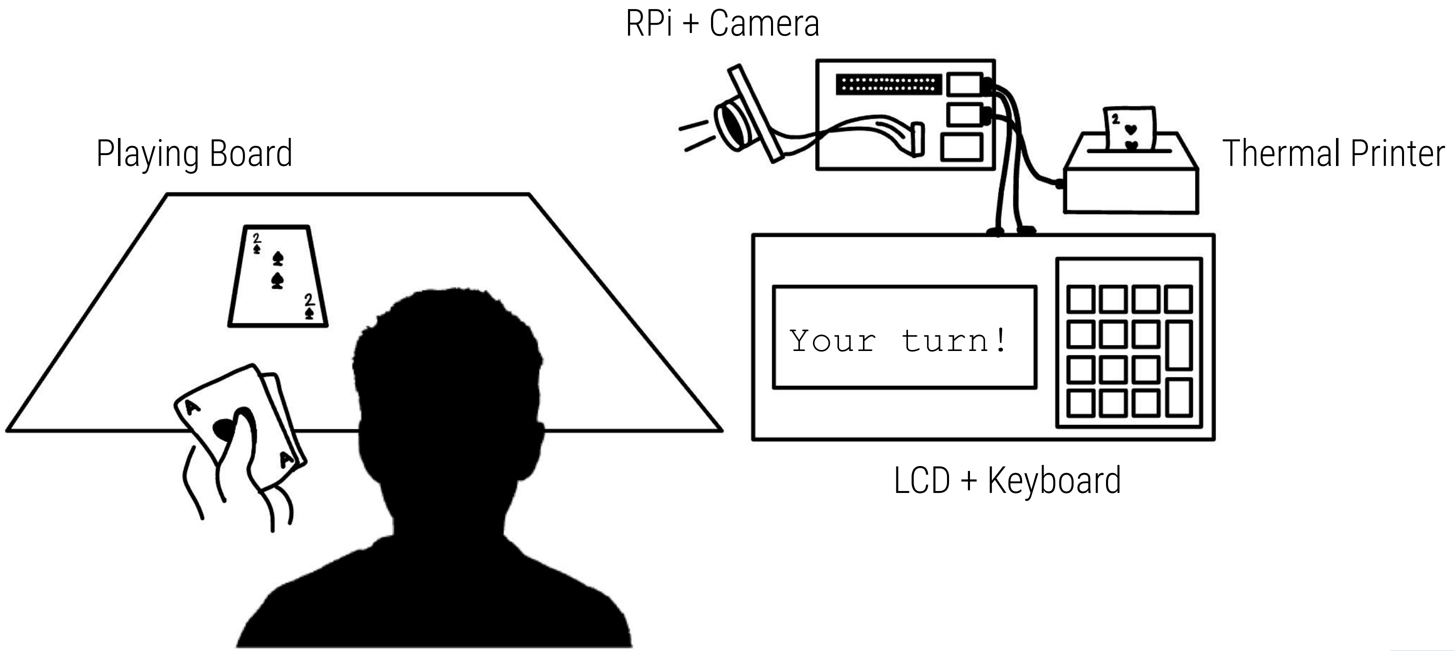
Quantitative Design Requirements

- A 18” x 24” playing/vision area
- Playing/vision area updates are done at least once per second
- When dealing cards are emitted at least once every 2 seconds
- The full physical device is smaller than a shoebox (14 in x 10 in x 5 in) and lighter than 10lbs

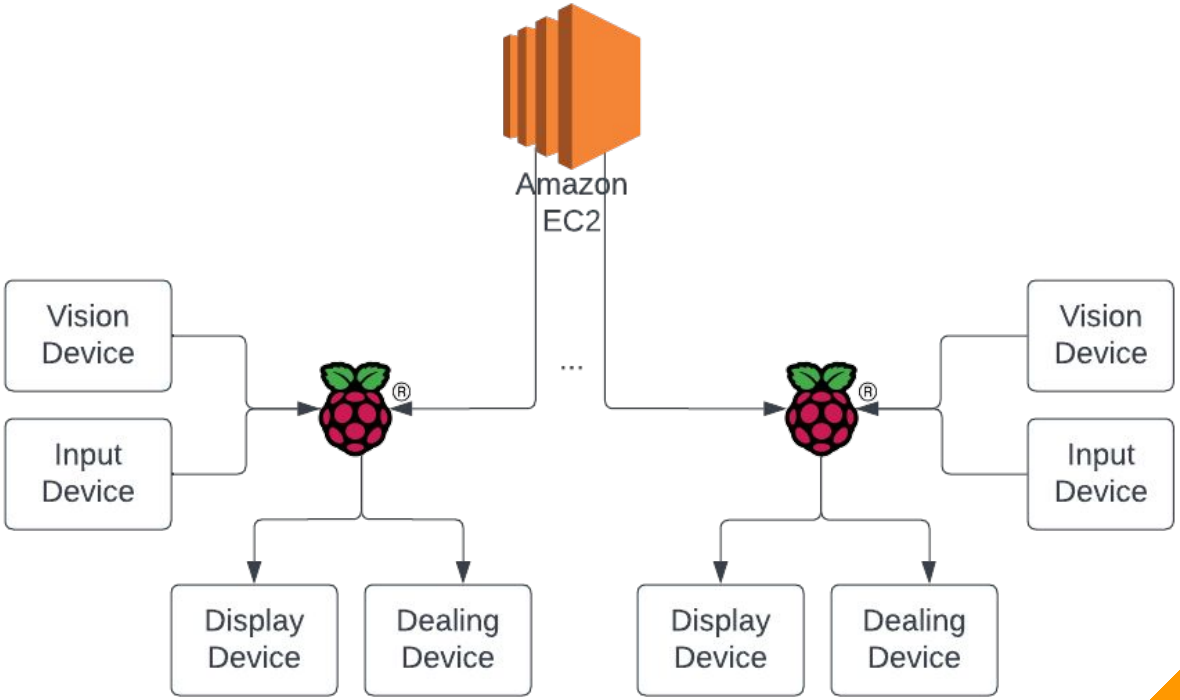
Solution Approach

- Vision Device
  - Hardware: Raspberry Pi Camera Module
  - Software: Open CV, Tensorflow, YOLO algorithm
- Input Device
  - 10 key keyboard
- Output Device
  - 40x4 character LCD screen
  - Thermal Receipt Printer

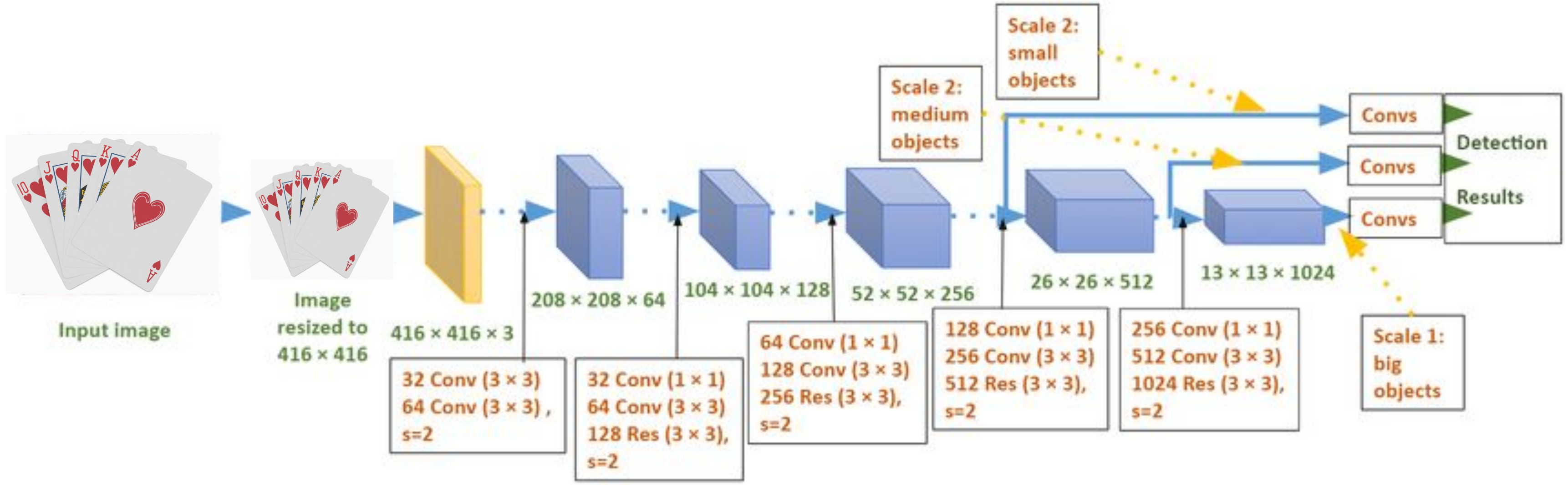
Mock up


RPi + Camera
Playing Board
Thermal Printer
Your turn!
LCD + Keyboard

System Block Diagram

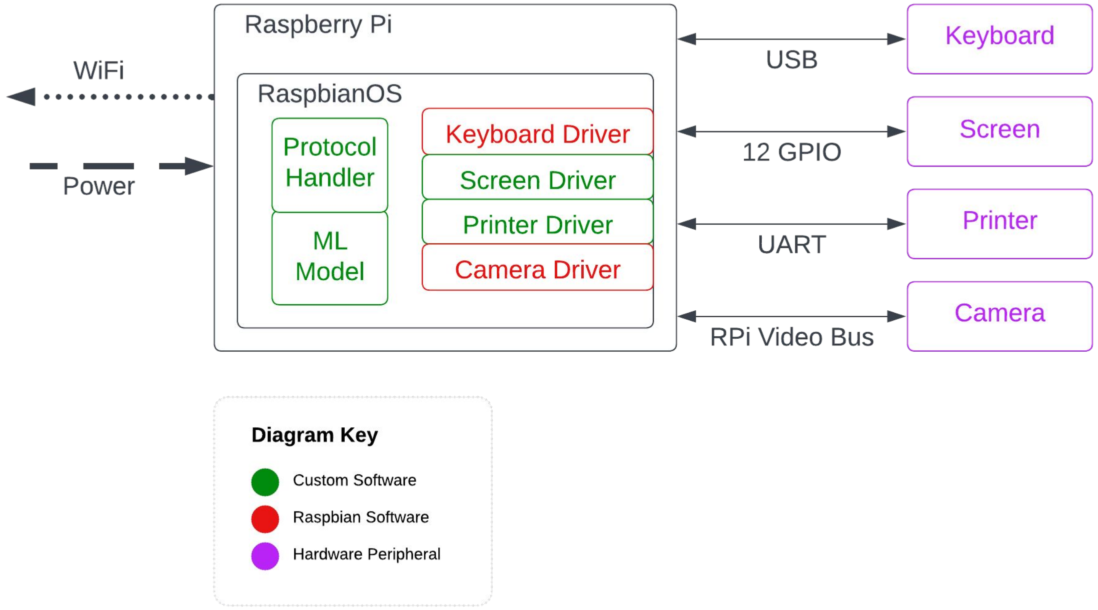


YOLO Architecture



Input image
Image resized to 416 × 416
416 × 416 × 3
208 × 208 × 64
104 × 104 × 128
52 × 52 × 256
26 × 26 × 512
13 × 13 × 1024
32 Conv (3 × 3) 64 Conv (3 × 3) , s=2
32 Conv (1 × 1) 64 Conv (3 × 3) 128 Res (3 × 3), s=2
64 Conv (1 × 1) 128 Conv (3 × 3) 256 Res (3 × 3), s=2
128 Conv (1 × 1) 256 Conv (3 × 3) 512 Res (3 × 3), s=2
256 Conv (1 × 1) 512 Conv (3 × 3) 1024 Res (3 × 3), s=2
Scale 2: medium objects
Scale 2: small objects
Scale 1: big objects
Convs
Convs
Convs
Detection Results

Device Block Diagram



Implementation Plan

Device	**Purpose**	**Implementation Actions/Software**
Thermal Printer	Card dealing device	Custom driver for their TTL interface using RPi’s TX/RX pins
Raspberry Pi Camera Module	CV/scan cards	Picamera2 library, YOLO for object detection, implemented in TensorFlow
LCD Screen	Game state display	Custom driver for their custom protocol using GPIO pins
Keyboard	Bets/card requests	RaspbianOS keyboard driver

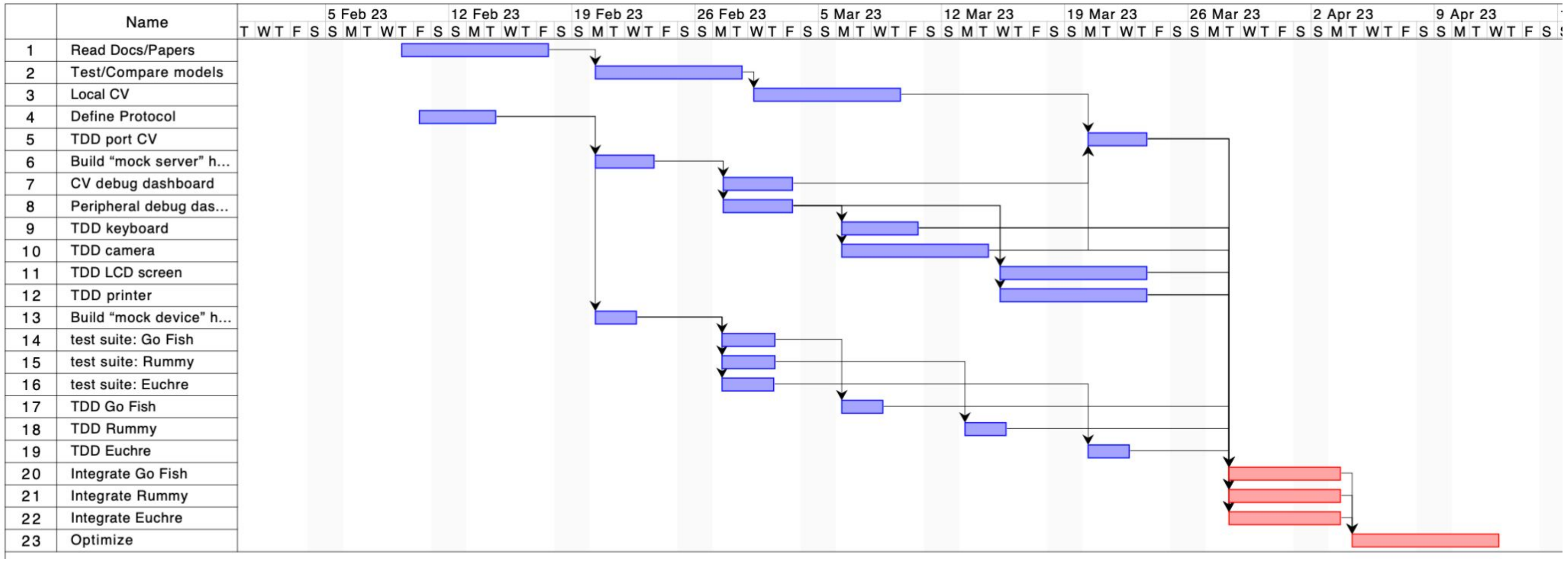
Unit Testing, Verification, Validation

1. *Thermal printer:* Be able to print 3.25” x 2.25” cards with corresponding suit and number in a maximum of 1.5 seconds
2. *Camera/Computer Vision:* Properly identifies card(s) in <35 ms
3. *Small keyboard:* Inputs are properly received and buffered in <10 ms.
4. *LCD Screen:* Displays text, then special characters like suits in <1 ms
5. *Implementing game logic for different games:* Go fish, Euchre, Rummy
6. *EC2/Networking:* Concurrency and logic tests.

Integration Testing, Verification, Validation

1. *Software device-level supervisor:* Services interrupt from peripherals in a timely manner without dropping any signals.
2. *Keyboard/Screen Coupling:* Keypresses appear on screen within our latency targets.
3. *Server/device Network Protocol*: The device supervisor is able to send game state update messages to the server in a timely manner, and the server can reconstruct a matching local game state. The reverse is also true, the server can send commands to the device, which are serviced in a timely manner.

Project Management



	Name
1	Read Docs/Papers
2	Test/Compare models
3	Local CV
4	Define Protocol
5	TDD port CV
6	Build “mock server” h...
7	CV debug dashboard
8	Peripheral debug das...
9	TDD keyboard
10	TDD camera
11	TDD LCD screen
12	TDD printer
13	Build “mock device” h...
14	test suite: Go Fish
15	test suite: Rummy
16	test suite: Euchre
17	TDD Go Fish
18	TDD Rummy
19	TDD Euchre
20	Integrate Go Fish
21	Integrate Rummy
22	Integrate Euchre
23	Optimize

Division of Labor:

ML Track: Rachel
Hardware Track: Mason & Miya
Software Track: Mason & Miya (& Rachel)