UCSB CE Capstone 23-24

PROJECT P.E.T.E

PROCEDURE EXECUTION TRACKING ENGINE

Spencer Tang
Sophie Guan
Frank Yao
Anoushka Sawant
Aaron Sin

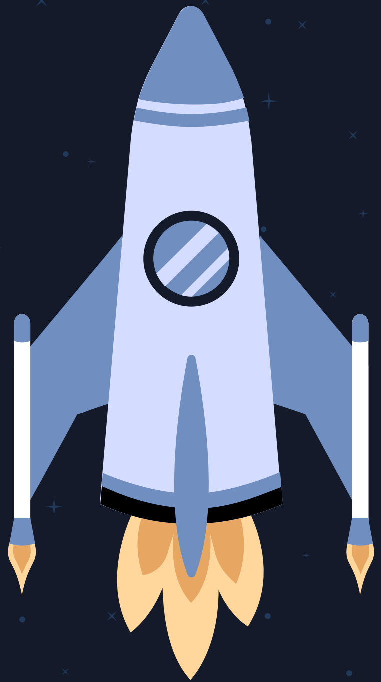


TABLE OF CONTENTS

01.A Problem Summary

Purpose

Automated assistance for users as they perform a procedure

Requirements

1. Track completion of each step in the procedure
2. Evaluate correctness of performance
3. Identify deviations

01.B Application & Significance

NASA

Replacing ground assistance with computer assistance

Medical

Assisting with surgeries

Education

Help with learning & repeatedly practicing a procedure

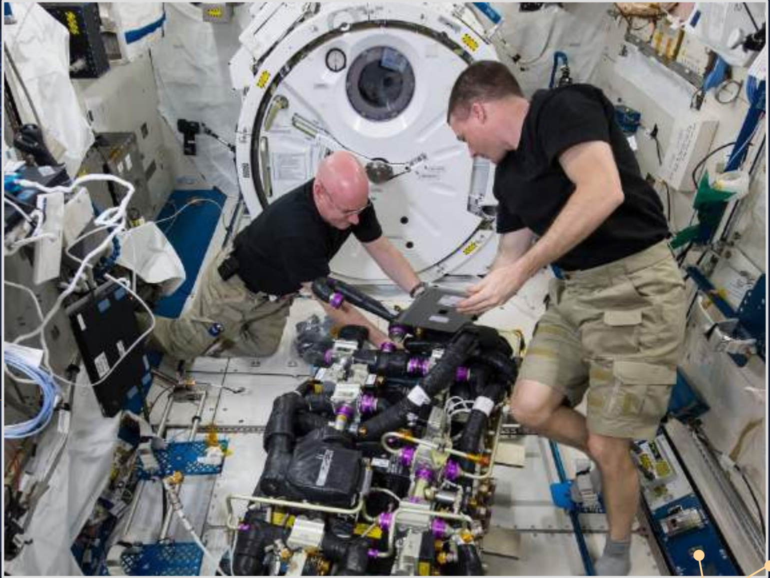


Figure 1.1. Astronauts in the ISS performing a maintenance task

Project Goal (Proof of Concept)

Automated assistance as users install the bottom bracket of a bike

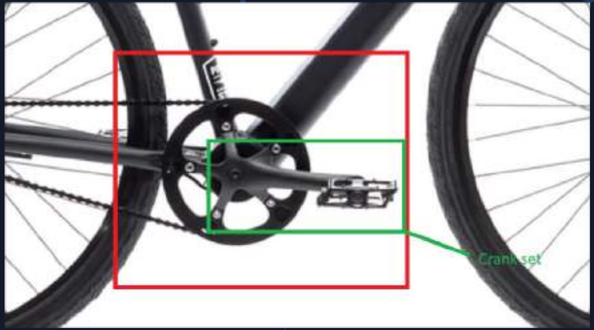


Figure 1.2. Bottom bracket

This procedure is analogous to a typical space maintenance task:

1. Complex interactions
2. Requires specialized tools in specific order
3. Complicated without prior experience

Step 1 – Install Spindle

Step 2 – Install Bottom Bracket

Step 3 – Tighten with Double-Flats Wrench

Step 4 – Install Crank Arm

Step 5 – Install Bolt

Step 6 – Install Pedal

Step 7 – Tighten Pedal with Pedal Wrench

01.C Procedure: Bottom Bracket Installation

Figure 1.3. Installing spindle into the bike axle

Step 1 – Install Spindle

Step 2 – Install Bottom Bracket

Step 3 – Tighten with Double-Flats Wrench

Step 4 – Install Crank Arm

Step 5 – Install Bolt

Step 6 – Install Pedal

Step 7 – Tighten Pedal with Pedal Wrench

01.C Procedure: Bottom Bracket Installation

Figure 1.4. Installing bottom bracket

01.C Procedure: Bottom Bracket Installation

Step 1 – Install Spindle

Step 2 – Install Bottom Bracket

Step 3 – Tighten with Double-Flats Wrench

Step 4 – Install Crank Arm

Step 5 – Install Bolt

Step 6 – Install Pedal

Step 7 – Tighten Pedal with Pedal Wrench

Figure 1.5. Tightening bottom bracket with a double-flats wrench

Step 1 – Install Spindle

Step 2 – Install Bottom Bracket

Step 3 – Tighten with Double-Flats Wrench

Step 4 – Install Crank Arm

Step 5 – Install Bolt

Step 6 – Install Pedal

Step 7 – Tighten Pedal with Pedal Wrench

01.C Procedure: Bottom Bracket Installation

Figure 1.6. Installing crank arm onto junction

01.C Procedure: Bottom Bracket Installation

- Step 1 – Install Spindle
- Step 2 – Install Bottom Bracket
- Step 3 – Tighten with Double-Flats Wrench
- Step 4 – Install Crank Arm
- Step 5 – Install Bolt
- Step 6 – Install Pedal
- Step 7 – Tighten Pedal with Pedal Wrench

Figure 1.7. Secure the crank arm with a bolt

01.C Procedure: Bottom Bracket Installation

- Step 1 – Install Spindle
- Step 2 – Install Bottom Bracket
- Step 3 – Tighten with Double-Flats Wrench
- Step 4 – Install Crank Arm
- Step 5 – Install Bolt
- **Step 6 – Install Pedal**
- Step 7 – Tighten Pedal with Pedal Wrench

Figure 1.8. Installing pedal onto other end of crank arm

Step 1 – Install Spindle

Step 2 – Install Bottom Bracket

Step 3 – Tighten with Double-Flats Wrench

Step 4 – Install Crank Arm

Step 5 – Install Bolt

Step 6 – Install Pedal

Step 7 – Tighten Pedal with Pedal Wrench

01.C Procedure: Bottom Bracket Installation

Figure 1.9. Tighten the bolt on pedal with pedal wrench

02.A Solution Summary

Overview

1. User places **P.E.T.E.** board in vicinity (overhead)
2. Overhead camera and sensor feeds data to board
3. User performs steps
4. For each step, the system will detect, validate, and display if it is correctly done

Detect → Validate → Display

Detection

1. Computer Vision (CV)

Purpose: identify objects of interest

```
bounding_box =
['867', '729', '1136', '1080',
'0.976039', '4.0']
```

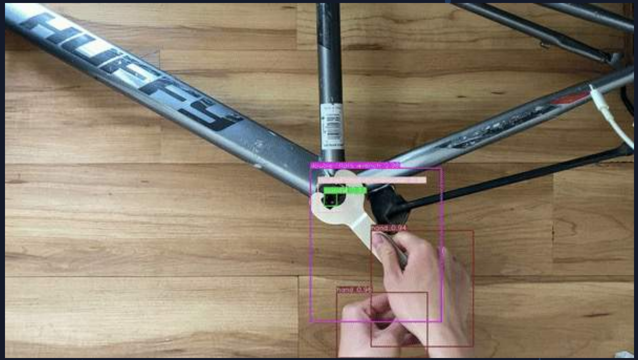


Figure 2.1. Computer vision output on a sample procedure run

Task: Real-Time Object Recognition

Model: YOLOv7-Tiny (You Only Look Once)

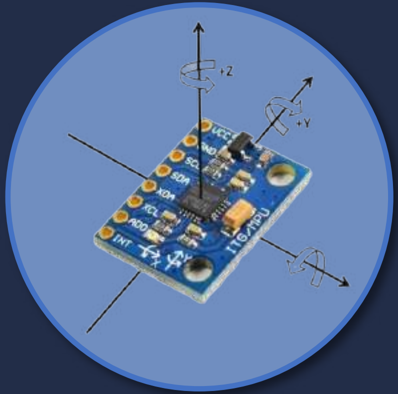
- **YOLOv7-Tiny:** 6M parameters
- **YOLOv7**: 37M parameters
- **PRO**: Faster inference, less resource-intensive
- **CON:** Decreased accuracy

Training: Image augmentation, transfer learning/fine-tuning.

Detection

MPU-6050 Sensor

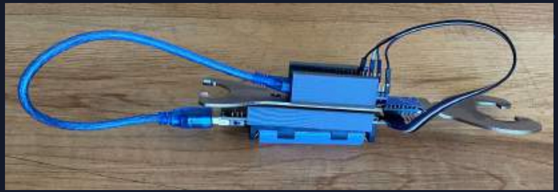
- 6-axis Accelerometer/Gyroscope
- GY-521 breakout board
- Accelerometer -> X, Y, Z
- Gyroscope -> Pitch, Yaw, Roll
- Temperature



2. Sensors

Purpose: identify rotation, temperature, and relative positions

```
["Accel X:  0.04310
       Y: -0.32801
       Z:  8.82742",
 "Angle X: -0.07260
       Y: -0.01221
       Z: -0.00588",
 "Temp:  27.07118C"]
```



Validation

Main Step

Step 3 – Tighten with Double-Flats Wrench

Substeps / "Step Criteria"

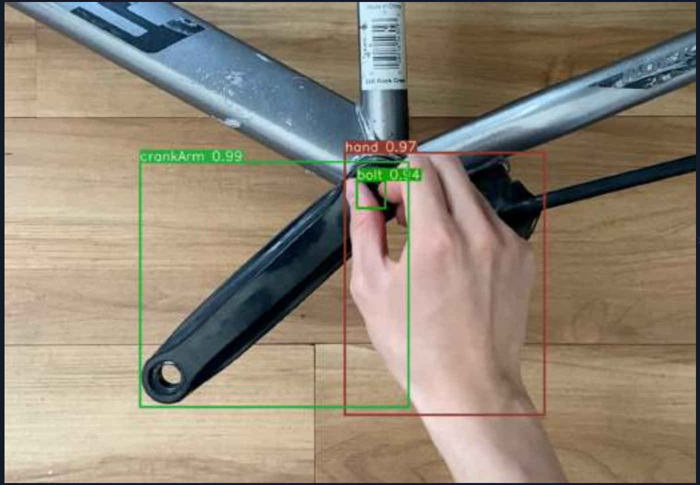
- Double-Flats in use
- Double-Flats is used on Bottom Bracket
- Achieves min. 3 rotations with Double-Flats

02.B Software Breakdown

Validation

1. Verify interaction between two objects

Ex: is **user** installing the **bolt** to the right place? Is the right **bolt** being used?



Validation logic:

1. **Check if interested objects exist in frame**
2. **Check Threshold**:
- Intersection over union between bounding boxes
- Euclidian Distance
- Duration of frames

Validation

2. Verify tightening/rotation using Tool

Ex: is **user** using the **pedal wrench** to tighten the **pedal**?

Validation logic:

CV:

- Check if objects of interest exists
- Check for continuous, rotational coordinate changes

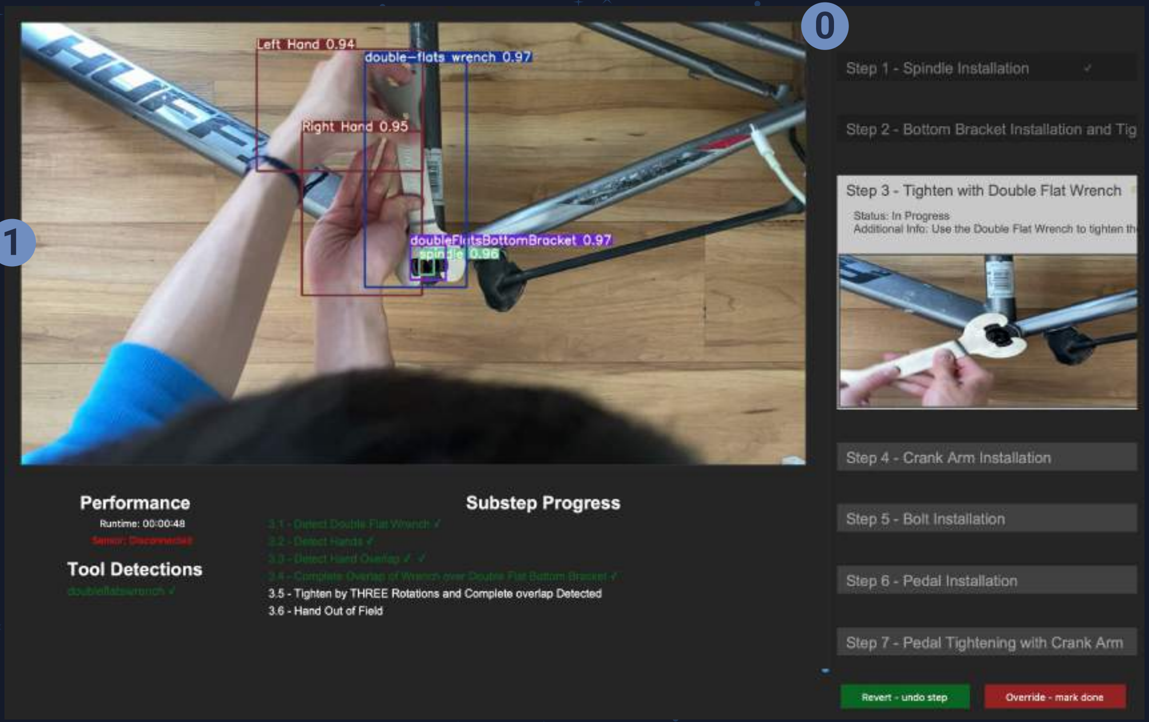
Sensor:

- Angular displacement % 360
- Exponential Weighted Average (EWMA) filter to smooth signal
- Instantaneous angular velocity

02.B Software Breakdown

Display

Graphical User Interface



0 Step List

1	Remove Chain	OK
1	Remove Chain	OV
2	Grab pedal wrench	IP
3	Slot the wrench between the connection of the crank arm and pedal	

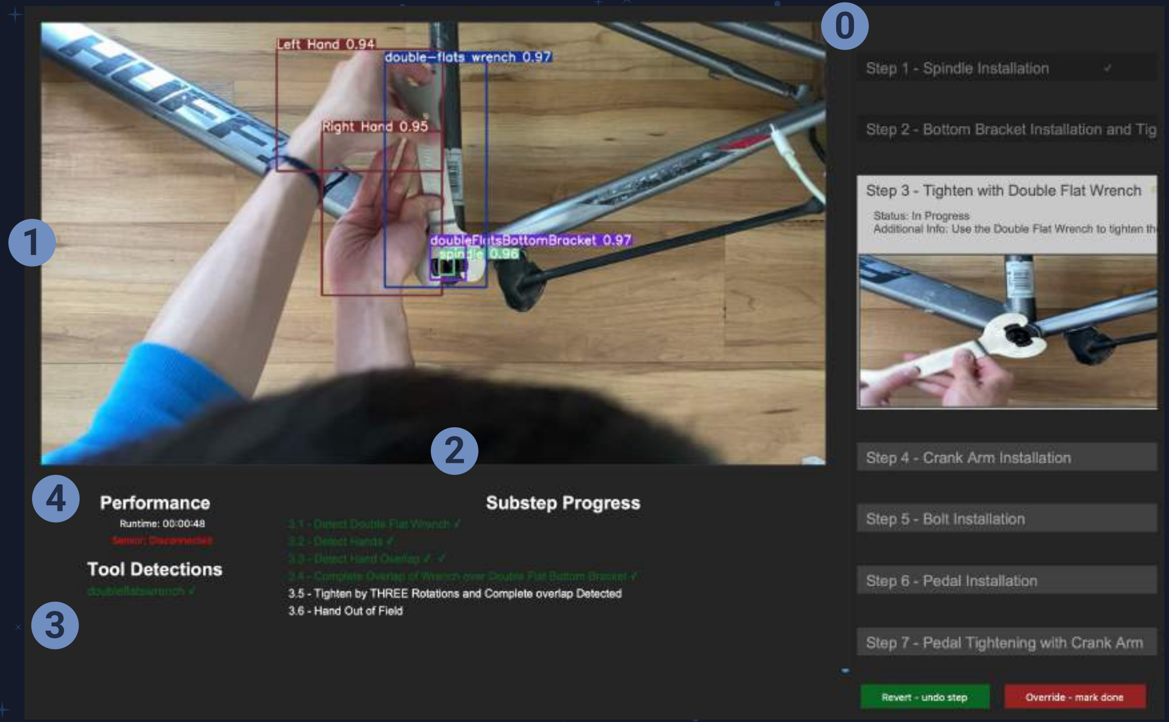
Revert Override

1 Video Stream

Live video with bounding boxes

Display

Graphical User Interface



2 **Substep List**

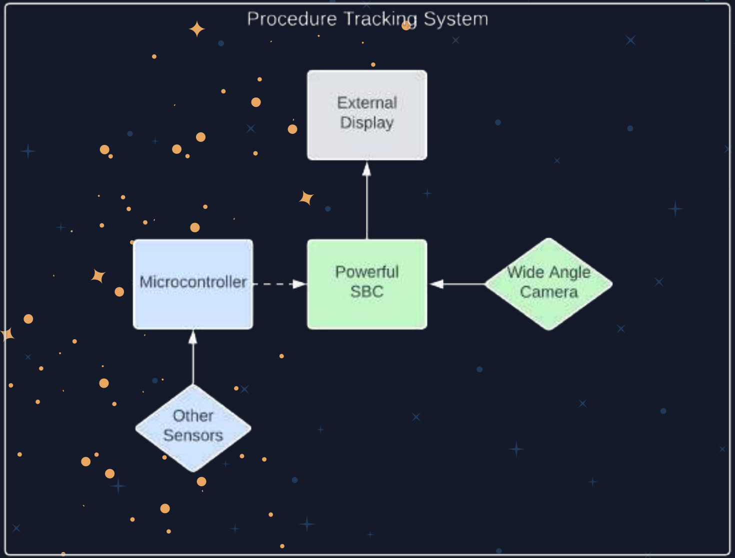
Substep criteria & progress

3 **Tool Detections**

Detected tools

4 **System Stats**

Runtime, sensor connectivity



02.C
Hardware Block Diagram

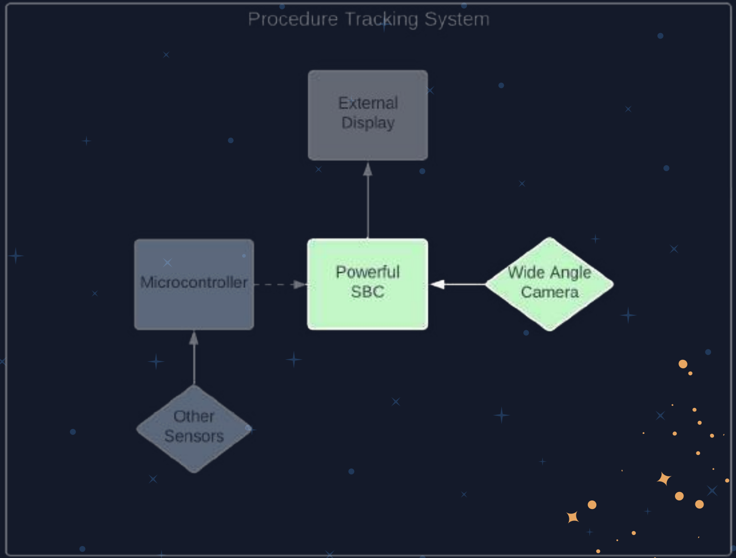
02.C Hardware Block Diagram

NVIDIA Jetson Orin Nano

Powerful Single Board Computer (SBC) with the processing power to run our image recognition model

E-con81 Camera

High fidelity, wide angle camera used to monitor the overall view of the workstation



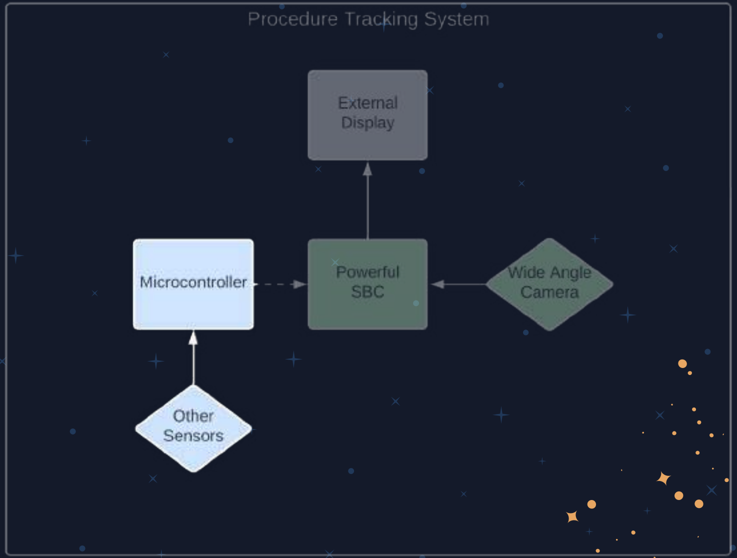
02.C Hardware Block Diagram

ESP 32s

Microcontroller to interface with additional sensors. Equipped with Bluetooth and wireless capabilities to allow for wireless sensor usage

GY-521 MPU-6050

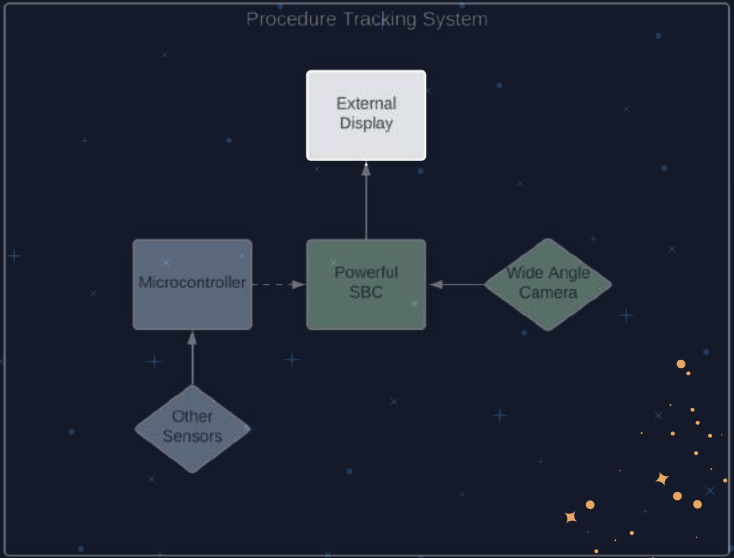
6-axis accelerometer gyroscope sensor module used to track movement of tools where camera data is not enough



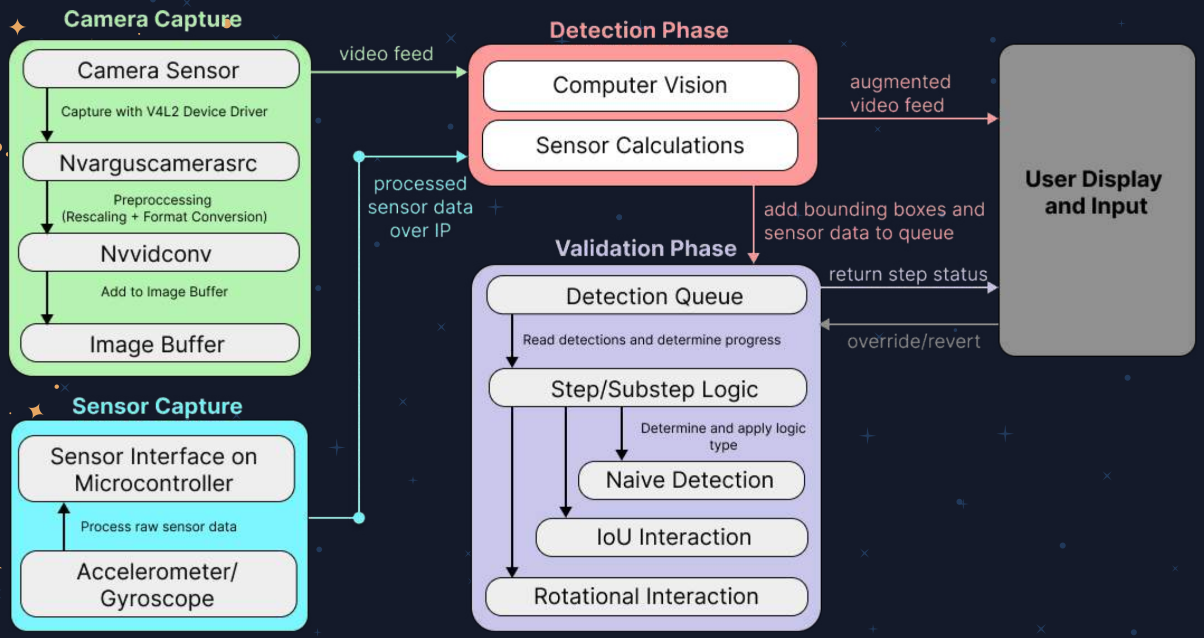
02.C Hardware Block Diagram

External Display (Monitor)

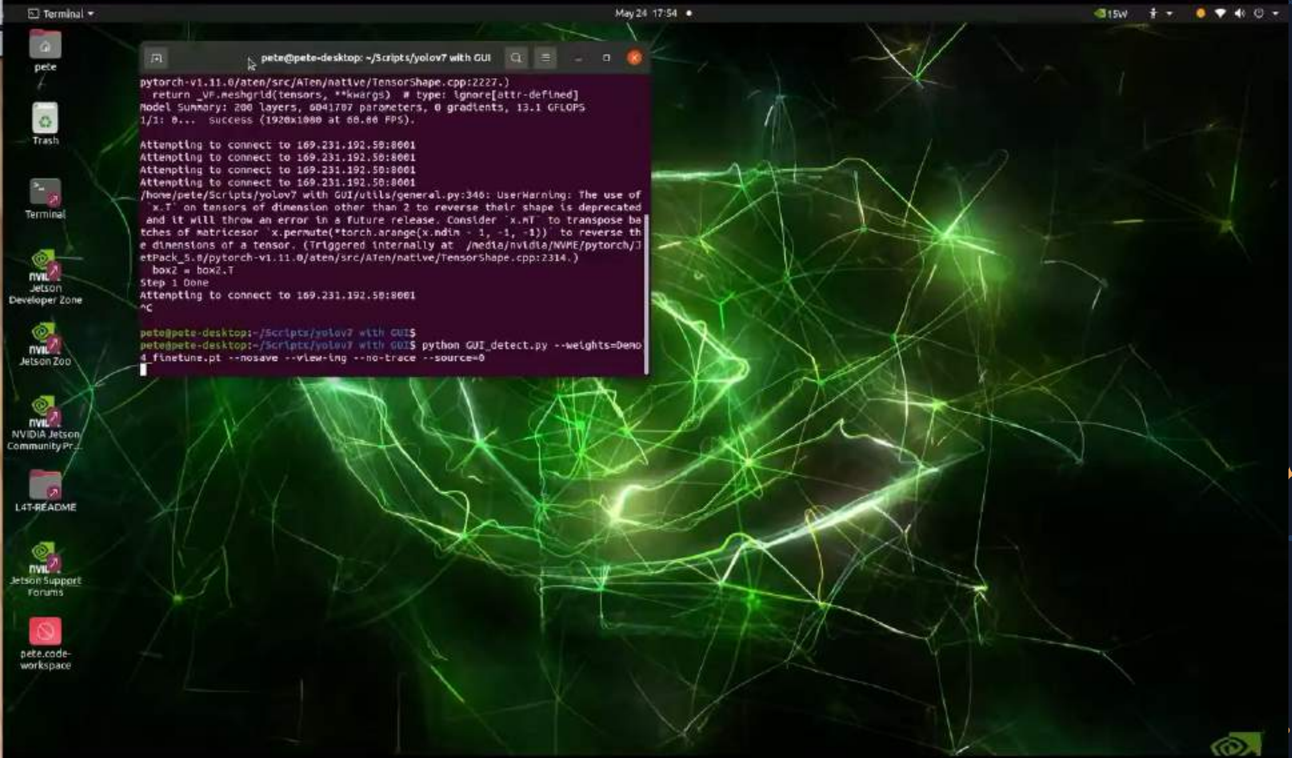
Used to notify user of current progress of the tracked procedure through a GUI to-do list



02.D Software Flow Diagram



02.E DEMO


Terminal
May 24 17:54
15W
pete
Trash
Terminal
Jetson Developer Zone
Jetson Zoo
NVIDIA Jetson Community Pr...
L4T-README
Jetson Support Forums
pete.code-workspace
pete@pete-desktop: ~/Scripts/yolov7 with GUI
pytorch-v1.11.0/aten/src/ATen/native/TensorShape.cpp:2227.)
return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 200 layers, 6041707 parameters, 0 gradients, 13.1 GFLOPS
1/1: 0... success (1920x1080 at 60.00 FPS).
Attempting to connect to 169.231.192.50:8001
Attempting to connect to 169.231.192.50:8001
Attempting to connect to 169.231.192.50:8001
Attempting to connect to 169.231.192.50:8001
/home/pete/Scripts/yolov7 with GUI/utils/general.py:346: UserWarning: The use of `x.T` on tensors of dimension other than 2 to reverse their shape is deprecated and it will throw an error in a future release. Consider `x.mT` to transpose batches of matricesor `x.permute(*torch.arange(x.ndim - 1, -1, -1))` to reverse the dimensions of a tensor. (Triggered internally at /media/nvidia/NVME/pytorch/JetPack_5.0/pytorch-v1.11.0/aten/src/ATen/native/TensorShape.cpp:2314.)
box2 = box2.T
Step 1 Done
Attempting to connect to 169.231.192.50:8001
^C
pete@pete-desktop:~/Scripts/yolov7 with GUI$
pete@pete-desktop:~/Scripts/yolov7 with GUI$ python GUI_detect.py --weights=Demo4_finetune.pt --nosave --view-img --no-trace --source=0

03.A

Key Challenges

Logic

- Robustly identifying ‘interaction’ from bounding box intersection
- Matching camera data with sensor data

Sensor

- Stable connection to WiFi
- Reliable accurate reading

CV

- General performance
- Performance with obstruction

Board

- Stable connection to sensor
- Image processing and general performance
- Installing drivers and building libraries from source

03.B
The P.E.T.E Team

Anoushka

- Validation logic & integration with GUI
- Data Labeling

Aaron

- CV model training
- Validation logic & integration with GUI
- Data labeling

Frank

- SBC and camera interfacing
- Data streaming and detection pipeline
- Validation logic

Sophie

- CV model training & improvements
- GUI design
- Validation logic (CV/Sensor)
- Data labeling

Spencer

- Sensor interfacing
- IoT tools
- Data collection & labeling

PROJECT

P.E.T.E

Thank you for listening!

& thank you to

Our mentors
Jessica Marquez
John Karasinski

Professor
Yogananda Isukapalli

TAs
Alex Lai
Eric Hsieh
Brian Li

and Viewers Like You