Aerial Detection for Object Placement

Machine Process

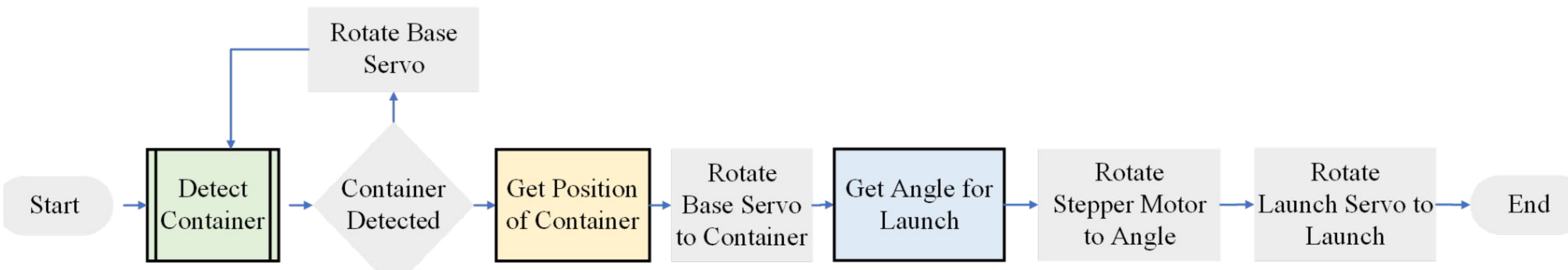


Fig. 2. Flowchart of machine process

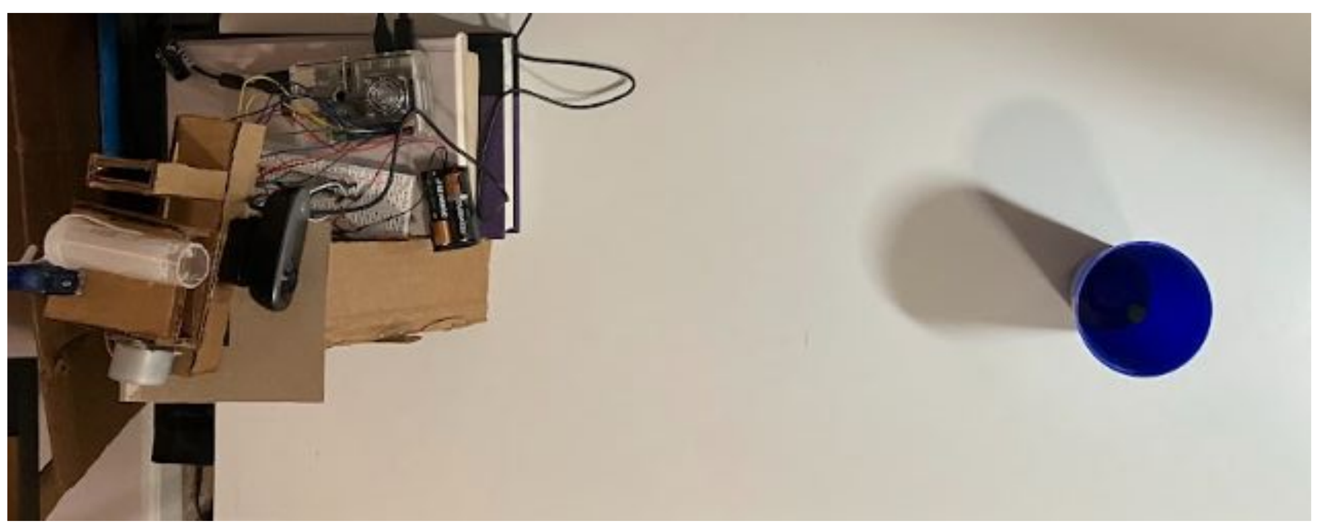


Fig. 4. Image of trial

Design

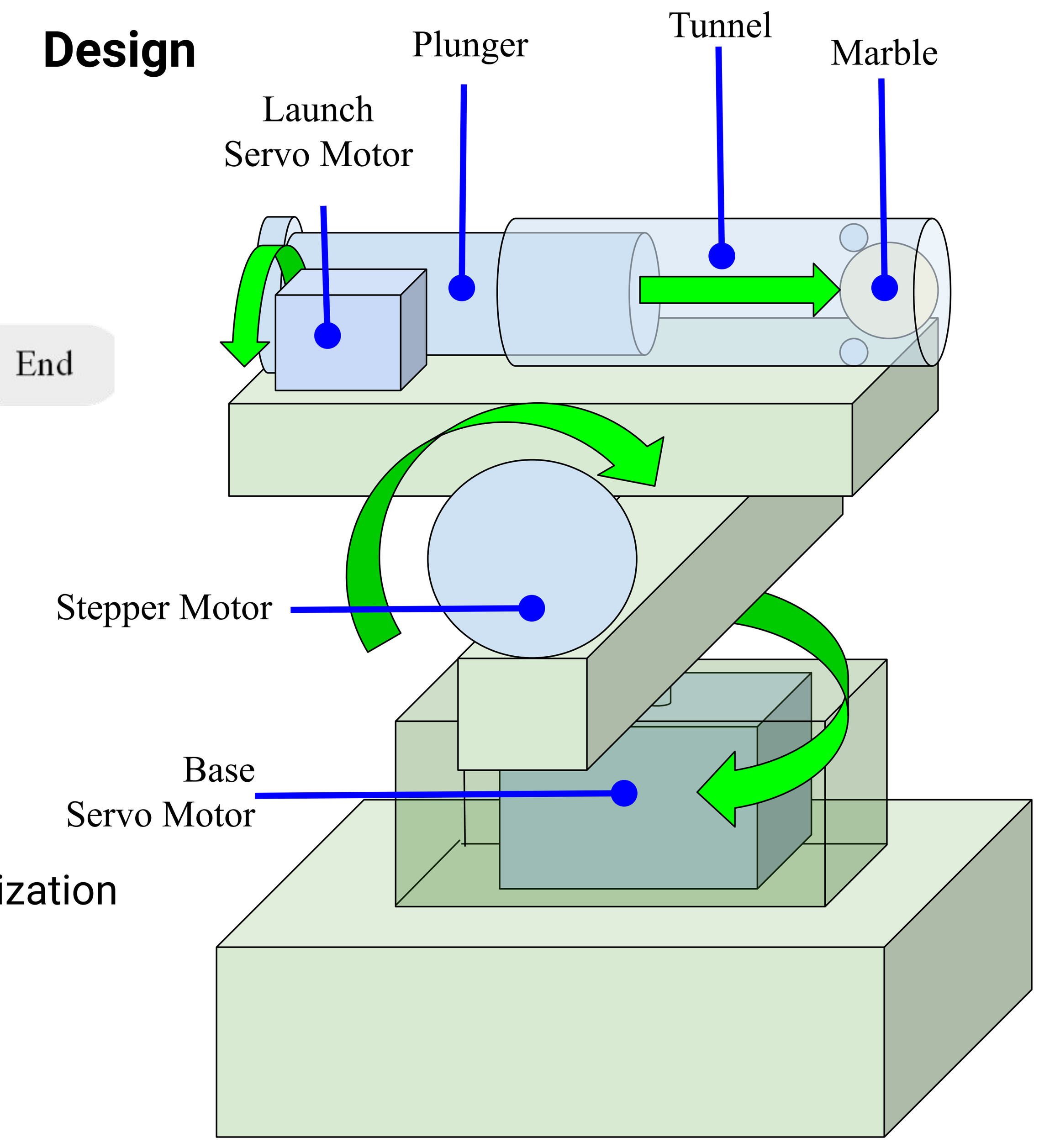


Fig. 3. Diagram of Machine

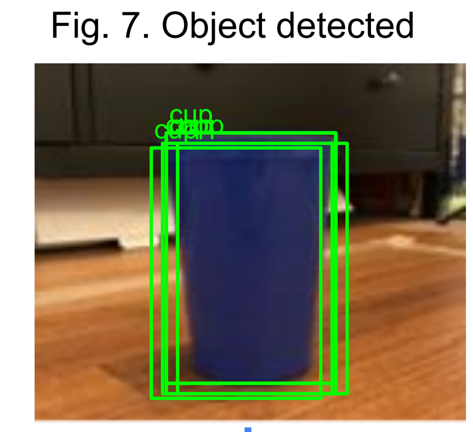
1. Detecting Container

You Only Look Once (YOLO Algorithm)

- Pre-trained machine learning model
- Object detection, classification, localization
- Processes entire image at once
  - Divides image into cells (Fig. 6)
  - Filters image through layers
  - Suppress redundant results (Fig. 8)
- Return results
  - Object classification
  - Width
  - Center coordinate

Fig. 5. Original Image

Fig. 7. Object detected



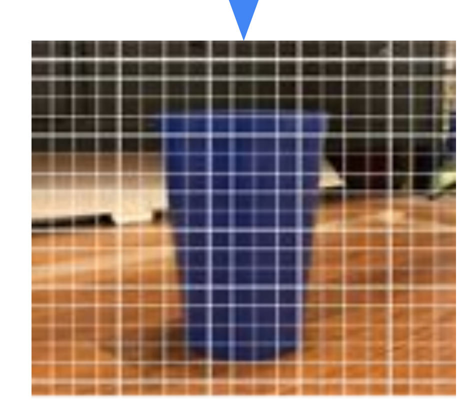


Fig. 6. Image divided into cells

Fig. 8. Result with suppressed redundancies

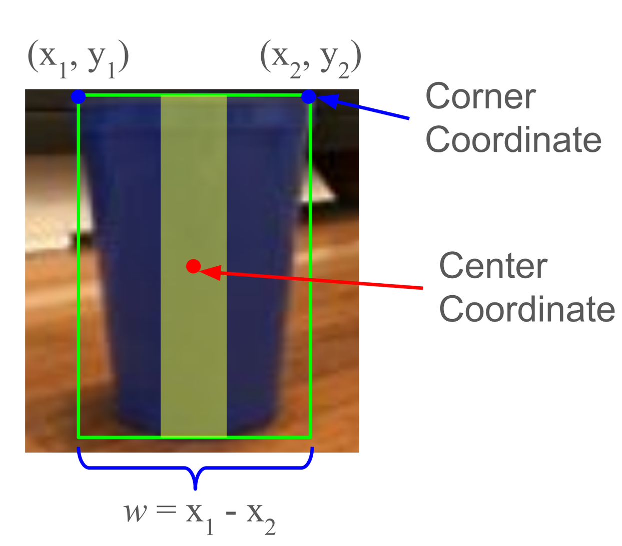


Fig. 9. Annotated image of reference points

2. Getting Position of Container

Location of Container Relative to Machine

- Compare container center coordinate to center range of image
- Rotate base servo until center coordinate in center range

Distance of Container Relative to Machine

- Given width of container in image from detection results
- Determine relationship between width(px) and distance (cm)
  - Place container at distances 25-125 cm, increment by 10 cm
  - Measure width(px) of container 100 times at each distance
    - Ex. Widths(px) at 30 cm: [475, 477, 476, 476, 476, …]
  - Form dataset of average widths and distances
  - Generate equation from dataset
    - Approximate Equation: $d = 9.32w^2 - 1.8w + 9.36$

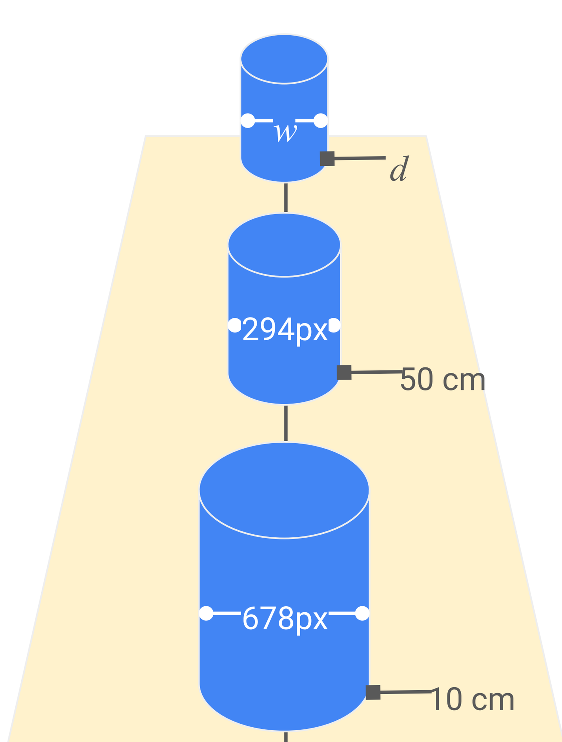


Fig. 10. Model of Width-Distance Relationship

Partial Code for Machine Calculations

```
# Data
w = [...]
d = [...]
eq = numpy.polyfit(w, d, 2)

# Conversion for Distance
distance = eq[0]*width**2 + eq[1]*width + eq[2]

# Calculation for Angle
angle = math.asin(9.8*distance/math.pow(5.781,2))/2
```

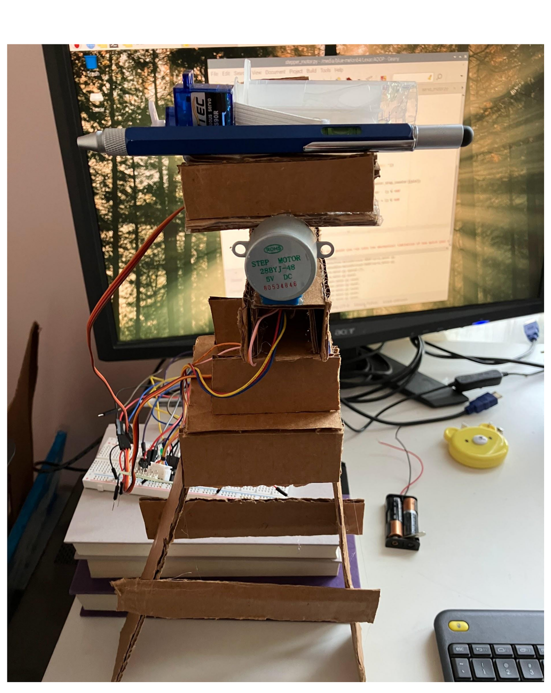


Fig. 13. Side view of machine

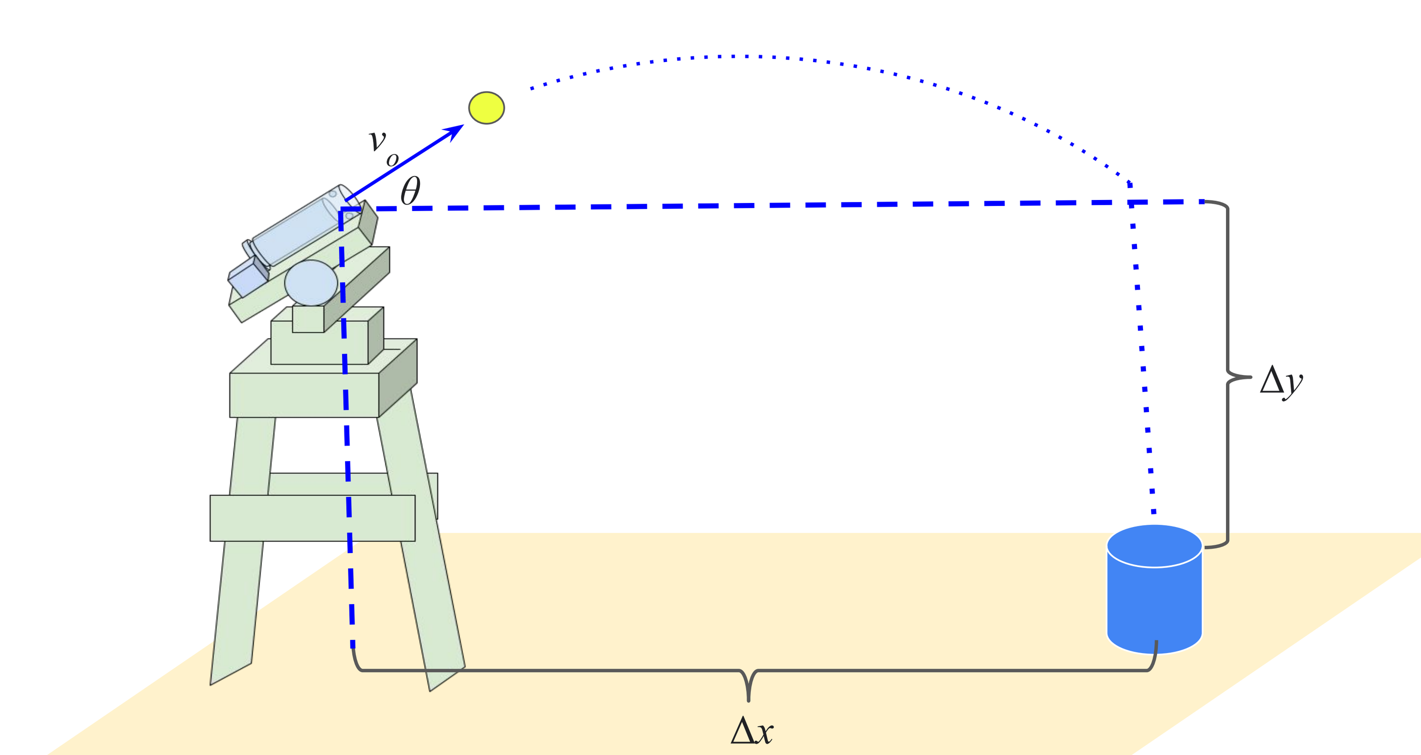


Fig. 12. Diagram of projectile

3. Getting Angle for Launch

Getting Initial Velocity (v_o) from Launcher

- Marble launched 11 times from fixed 35° angle
- Distance marble traveled recorded each launch
- Calculate initial velocity
  - Average Initial Velocity (N) = 5.783

Launch #	Distance (m)	V_o (N)
1	0.91	5.850
2	0.88	6.127
3	0.90	6.235
4	0.93	5.976
5	0.90	6.102
6	0.87	4.965
7	0.85	6.071
8	0.84	6.151
9	0.89	5.515
10	0.89	5.596
11	0.87	5.030

Table 1. Launch trials for distance and velocity

Solve for Initial Velocity (v_o)
Given distance (Δx) and angle (θ)

$$\Delta x = v_x t + at^2$$
$$v_x = v_o \cos\theta$$
$$\Delta x = v_o \cos\theta * t$$
$$v_o = \frac{\Delta x}{t\cos\theta}$$

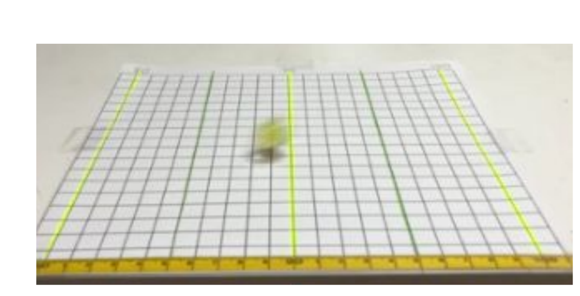


Fig11. Measuring distance of launch

Getting Angle for Launch

Solve for Launch Angle (θ)
Given distance (Δx) and initial velocity (v_o)

$$v_{max\,y} = v_{initial\,y} + gt_{max}$$
$$v_y = v_o \sin\theta$$
$$0 = v_o \sin\theta + gt_{max}$$
$$t_{max} = \frac{-v_o \sin\theta}{g}$$
$$t_{total} = \frac{-2v_o \sin\theta}{g}$$
$$\Delta x = v_o \cos\theta * t$$
$$\Delta x = \frac{-2v_o^2 \sin\theta\cos\theta}{g}$$
$$-\frac{g\Delta x}{v_o^2} = 2\sin\theta\cos\theta$$
$$-\frac{g\Delta x}{v_o^2} = \sin 2\theta$$
$$\theta = \frac{\sin^{-1}\left(-\frac{g\Delta x}{v_o^2}\right)}{2}$$

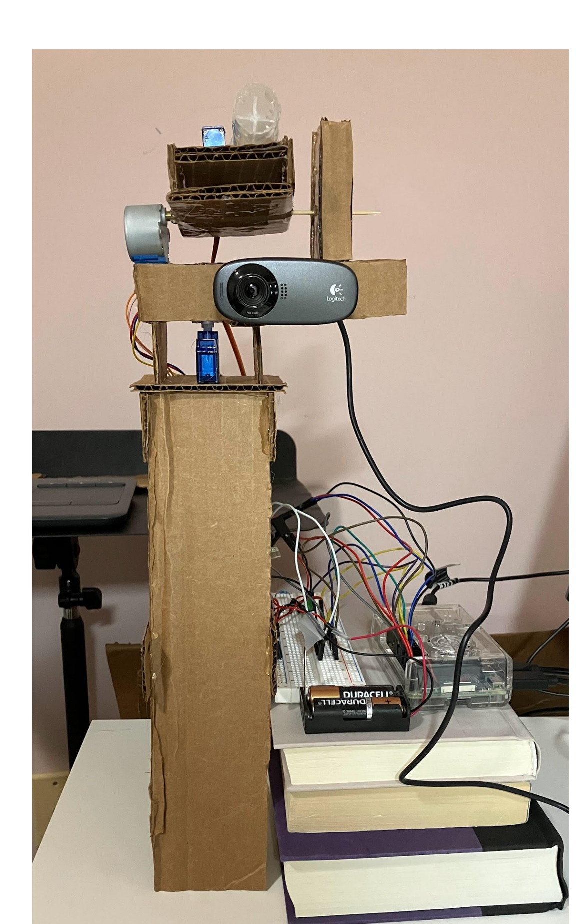


Fig. 15. Front view of machine

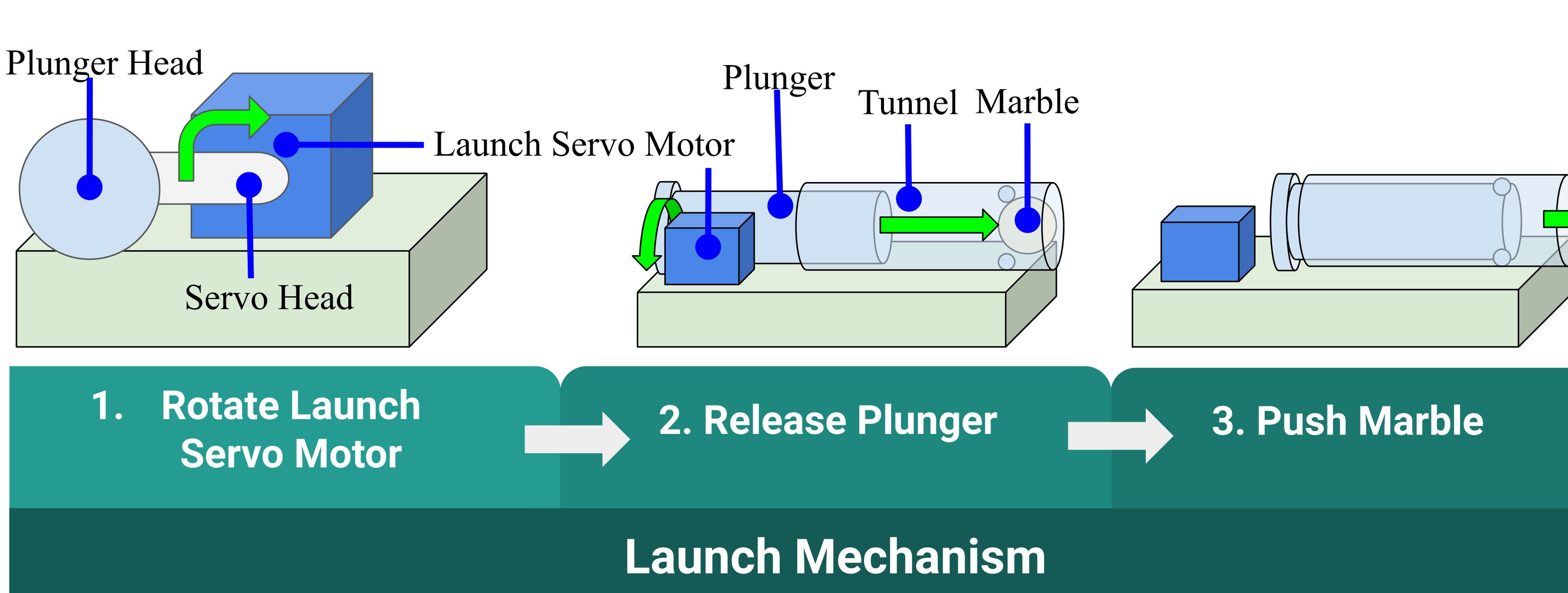


Fig. 14. Diagram of launch mechanism

Abstract

From level ground, human vision is limited to immediate surroundings. For a broader scope of view, humans require a higher vantage point. However, a greater distance sacrifices accuracy. Humans need to efficiently and accurately place objects over large target areas for various tasks such as agricultural mass-seeding, combating wildfires, or marine surface surveillance. To expand human vision for wide-range object placement, the project is a stationary machine designed to autonomously detect a container from an aerial view and complete a simple task: launch a marble into the container. The machine searches for the container with machine learning then estimates the distance from the container and finally calculates the angle to launch the marble. The machine begins centered 25 cm away from the edge of a 1.0 by 0.5m area. The machine detects a 14.2 cm tall container with a 9 cm diameter placed at a random location within the defined area. When the machine locates the container, the machine orientates to launch a 14mm diameter marble into the container. After ten trials, the machine achieved a 90% success rate of landing the marble in the container. Improvements include a more consistent launch mechanism or a more efficient detection model to increase application. Future experimentation includes a larger area or varying targets to further challenge the machine. Ultimately, features of the current machine functioned as intended and could be implemented in more extensive real-world projects.

Rationale

- **Problem**: Limitations of Human Vision for Wide-Range Areas
    - Restricted to immediate surrounding
    - Require an higher vantage point
    - Lower accuracy at greater distance
- **Solution**: Elevated Machine for Autonomous Object Detection
    - Applications of Wide-Range Object Detection
        - Mass seeding in agriculture
        - Identifying and combating wildfires
        - Maritime surveillance
        - Long-distance package delivery
    - Reduce risk to human life with machines
    - Increase efficiency with automated alternative
    - Expand aerial opportunities for humans

Engineering Goal

Create a machine with machine learning algorithms to autonomously detect a 14.2 cm tall container 9 cm in diameter located within a 1.0 by 0.5 m area and launch a 14 mm diameter marble into the container from a stationary position 0.25 m from the center of an edge of the defined area.

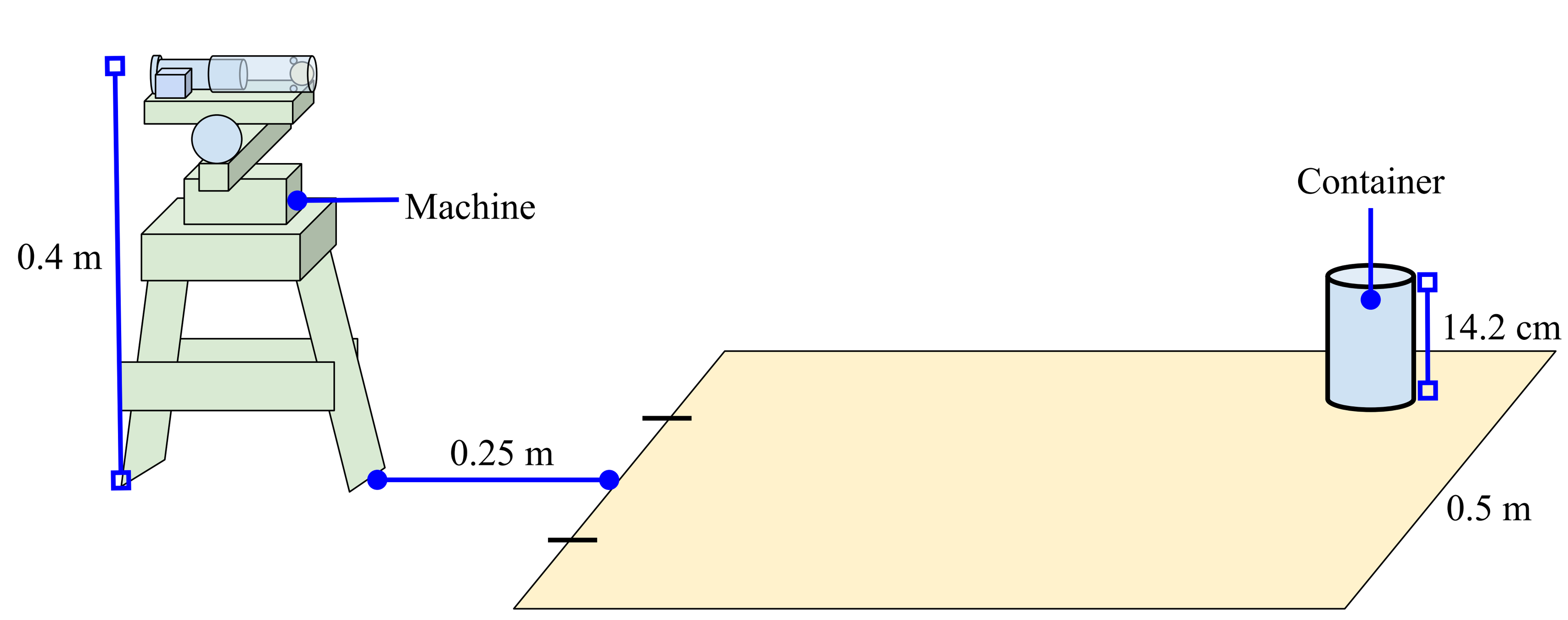


Fig. 1. Diagram of experiment setup

Procedure

1. Place container at random location within 1.0 by 0.5 m. area
2. Place machine 0.25m away from center edge of area
3. Turn on machine, run program, and allow machine to move
4. When marble is launched, record if marble is in container (true/false)
5. Repeat steps 1 through 4 ten times

Materials

Hardware

- Raspberry Pi 4
- Servo Motors (SG90, FS90R)
- Stepper Motor (28BYJ-48)
- Breadboard
- Wires/Resistors/AA Batteries
- Webcam (Logitech C310)
- Hollow Cylinder/Elastic

Python Libraries/Packages

- OpenCV
- NumPy
- YOLOv3 - 320

Note: Program written in Python 3.7

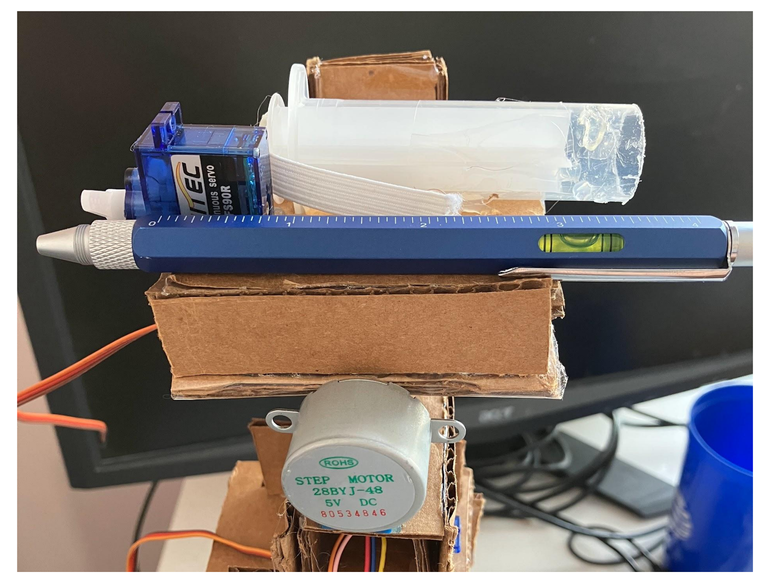


Fig. 16. Side-view of launch mechanism

Fig. 17. Slant view of machine

Program Pseudocode

```
Import files and packages
Initialize hardware
While True:
   While detector returns False
      If servo not at final position then
         Rotate base servo
      Else return “no container found” and break
   While detector returns center coord outside center range
      Rotate base servo
   Calculate distance from container
   Calculate angle needed for launch
   Rotate stepper to angle
   Rotate launch servo to launch
Clean hardware
```

Results

The machine achieved a 90% success rate, failing 1 out of 10 trials.

Trial Number	1	2	3	4	5	6	7	8	9	10
Contact with Target	True	True	True	True	False	True	True	True	True	True

Table 2. Table of results

Conclusion

Overall success

- Single failed trial due to inconsistent launch mechanism - marble fell out before launch

Further Investigation

- Larger testing area
- Varying targets
    - Multiple or moving targets
- Alternative input for search

Limitations

- Inconsistent launch mechanism
- Slow turning mechanism
- High time consumption

Possible improvements

- Full rotational capability
- More data collected for initial velocity determination
- Alternative detection model

References

Brownlee, Jason. "4 Distance Measures for Machine Learning." *Machine Learning Mastery*, Mar. 2020, machinelearningmastery.com/distance-measures-for-machine-learning/. Accessed 5 Nov. 2023.

“Servo Motor SG-90.” *Components 101*, 18 Sept. 2017, https://components101.com/motors/servo-motor-basics-pinout-datasheet. Accessed 8 Nov. 2023.

"Raspberry Pi Documentation." *Raspberry Pi*, www.raspberrypi.com/documentation/computers/os.html. Accessed 1 Nov. 2023.

Raspberry Pi Pin Layout. *Raspberry Pi*, www.raspberrypi.com/documentation/computers/os.html. Accessed 1 Nov. 2023.

Saha, Sumit. "A Comprehensive Guide to Convolutional Neural Networks - the ELI5 Way." *Towards Data Science*, 15 Dec. 2018, towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed 5 Nov. 2023.