A Novel Approach to Underwater Mobility

Materials

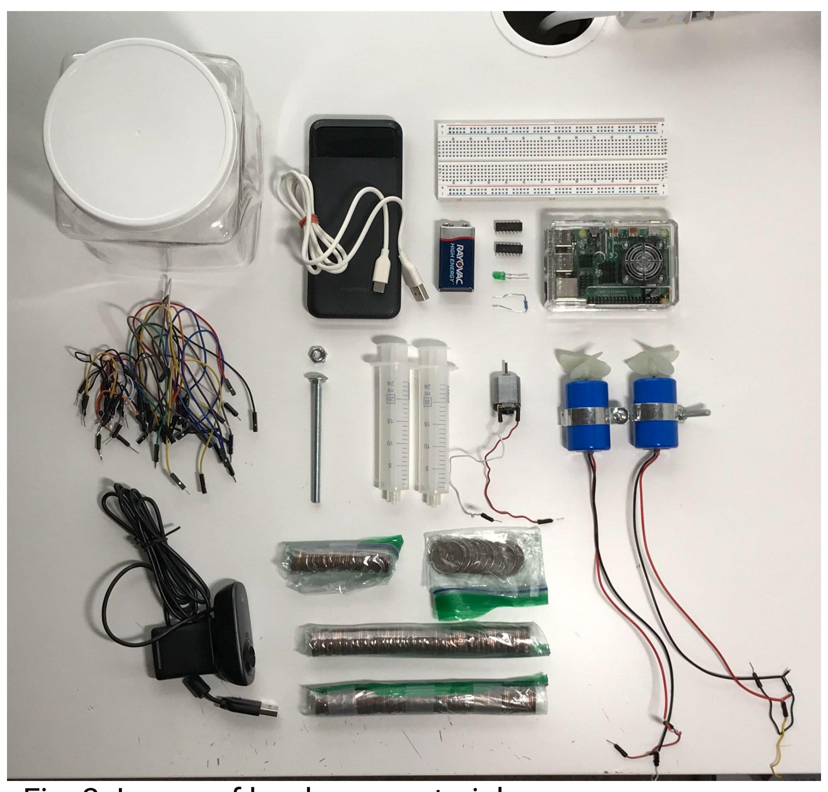


Fig. 2. Image of hardware materials

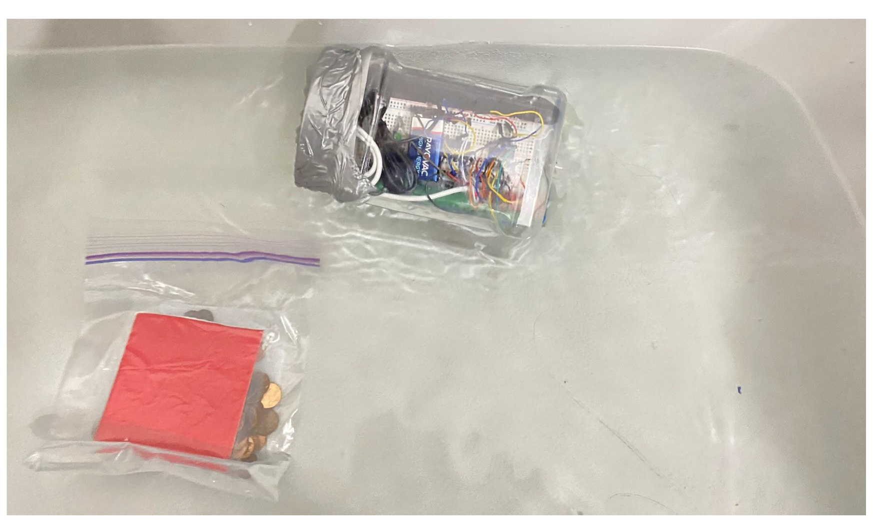
Hardware

- Raspberry Pi 4
- Breadboard
- Motor Drivers (L293D)
- DC Motors
- Plastic Container
- Syringes
- Nut and Bolt
- LED Light
- 9v Battery
- Power Bank
- Webcam
- Coins

Python Libraries

- OpenCV
- NumPy

Note: Program written in Python 3.7



Design

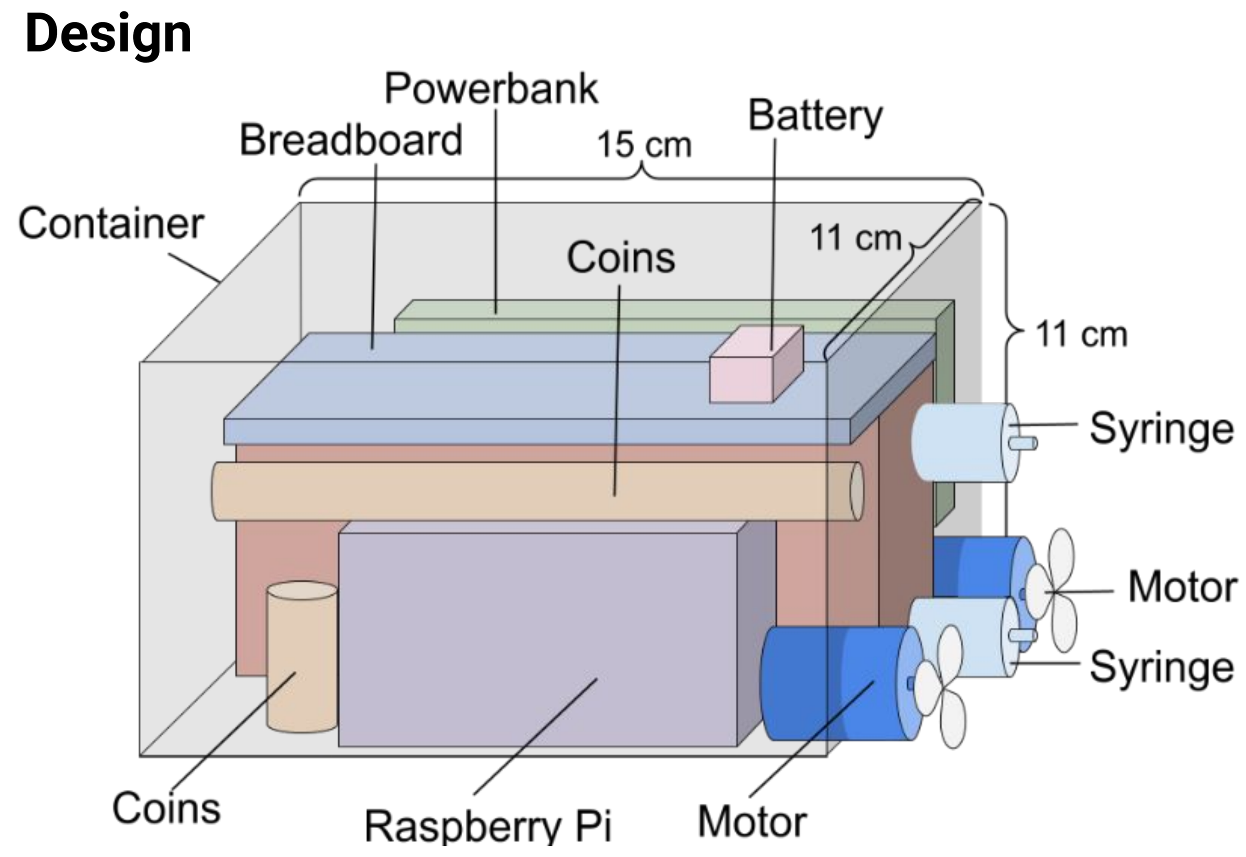


Fig. 3. Diagram of machine design

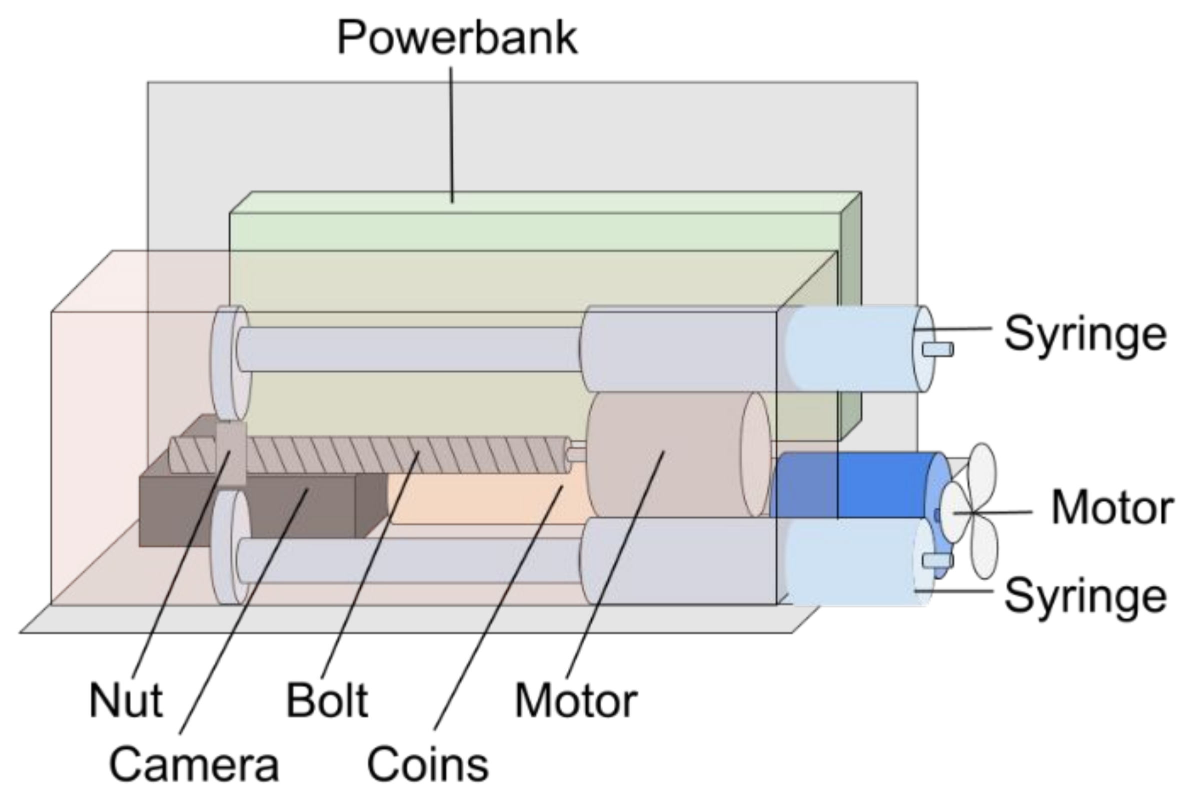
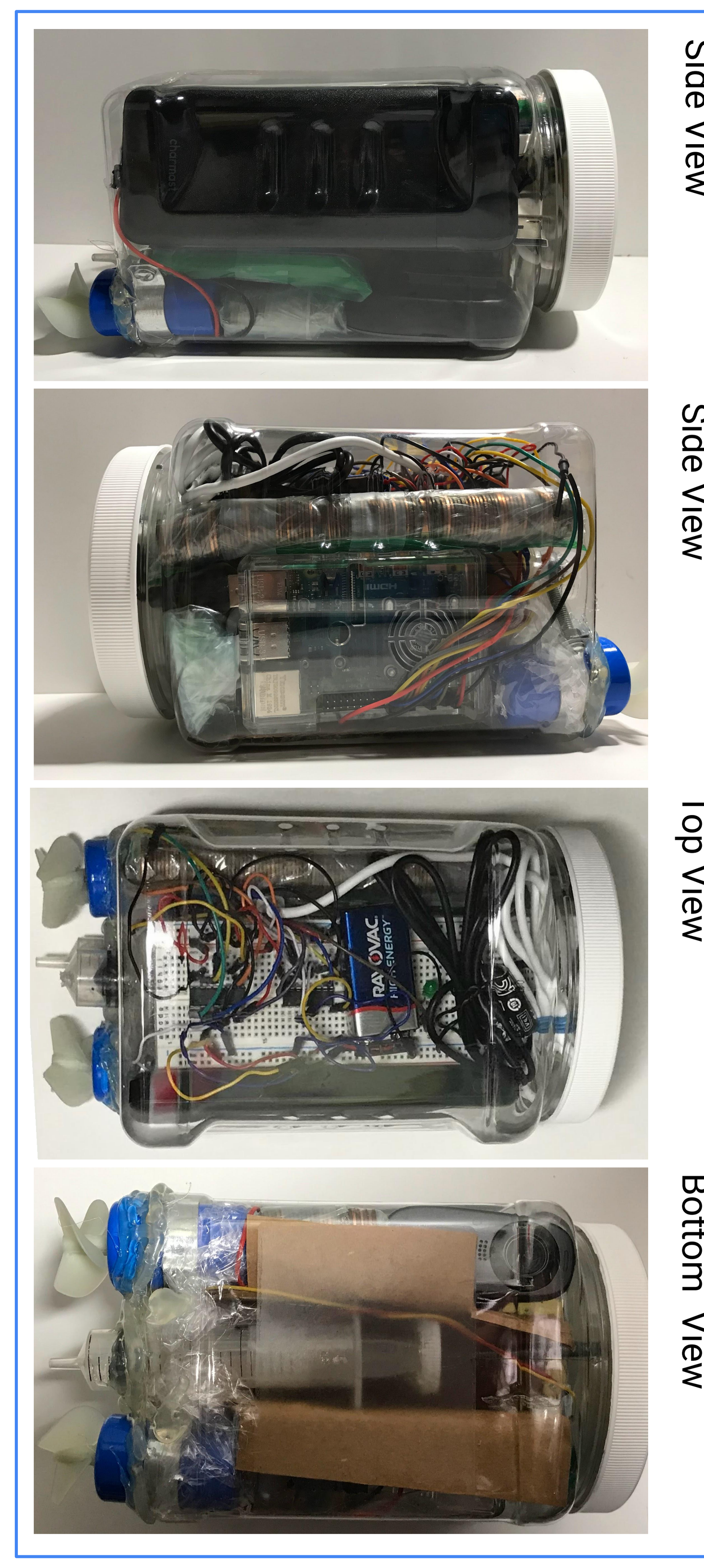
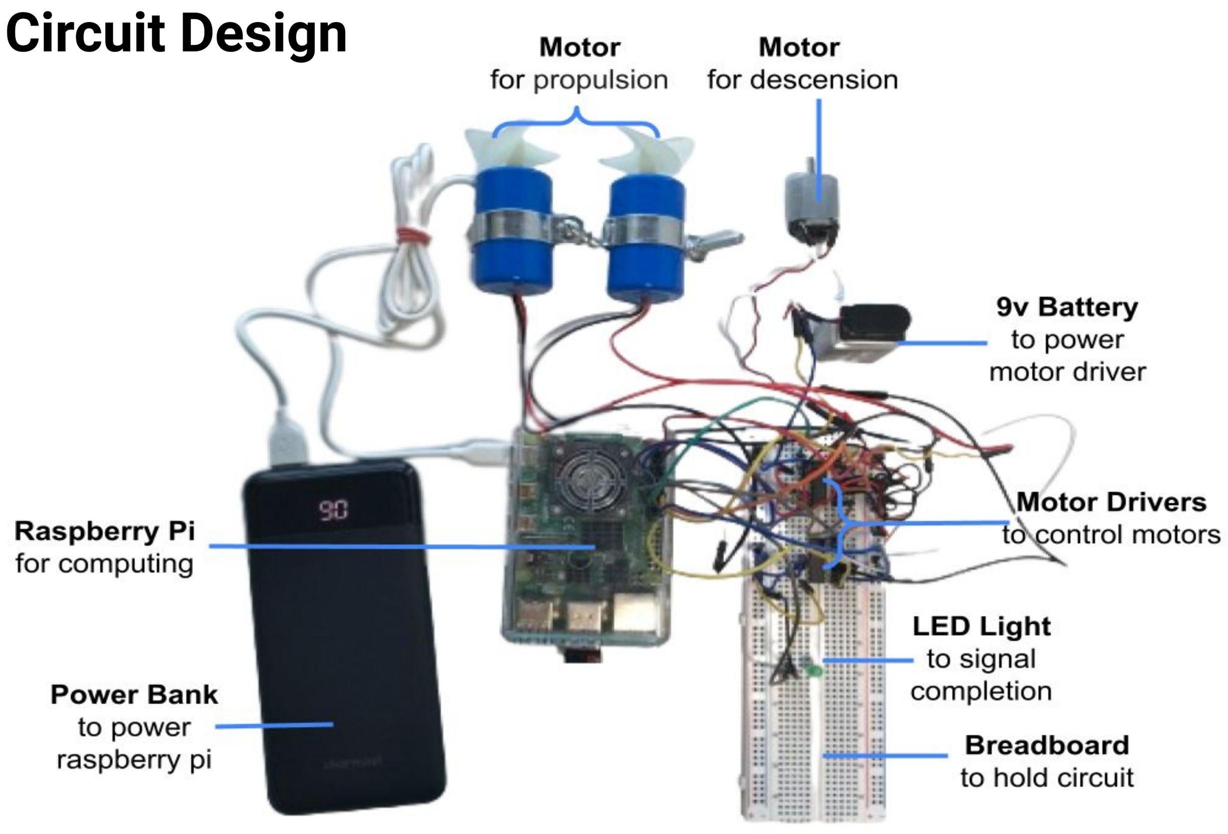


Fig. 4. Diagram of machine internal design



Circuit Design



Descension Mechanism

- Density must be greater than density of water to sink
  - $D = m/v$
  - Density of Water ≈ 1.00 g/ml
- Original Density of Machine < 1.00 g/ml
- Increase mass of machine to increase density
  - Mass Added = 40 ml
  - New Density > 1.00 g/ml
- Increase mass by intaking water through syringes
  - Nut and bolt mechanism translate rotational motion from motor to linear motion for syringe plungers

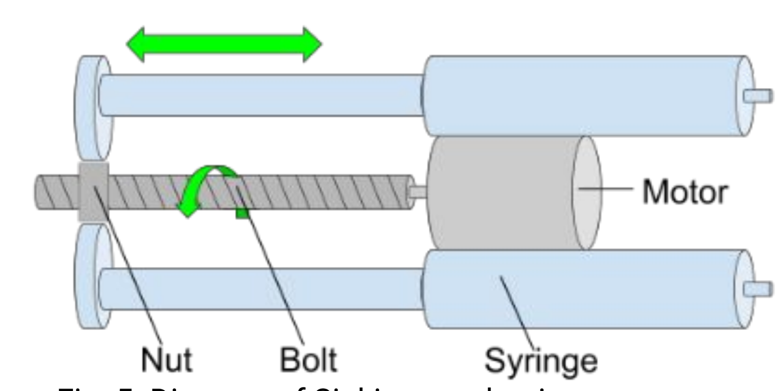


Fig. 5. Diagram of Sinking mechanism

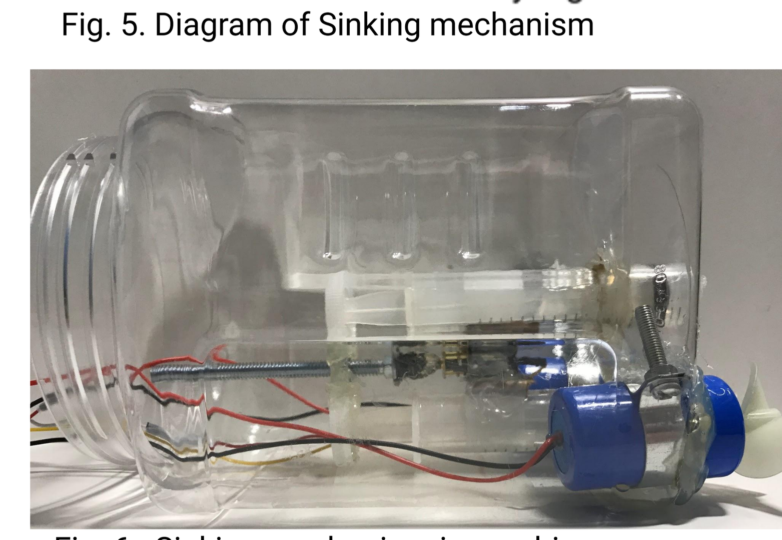
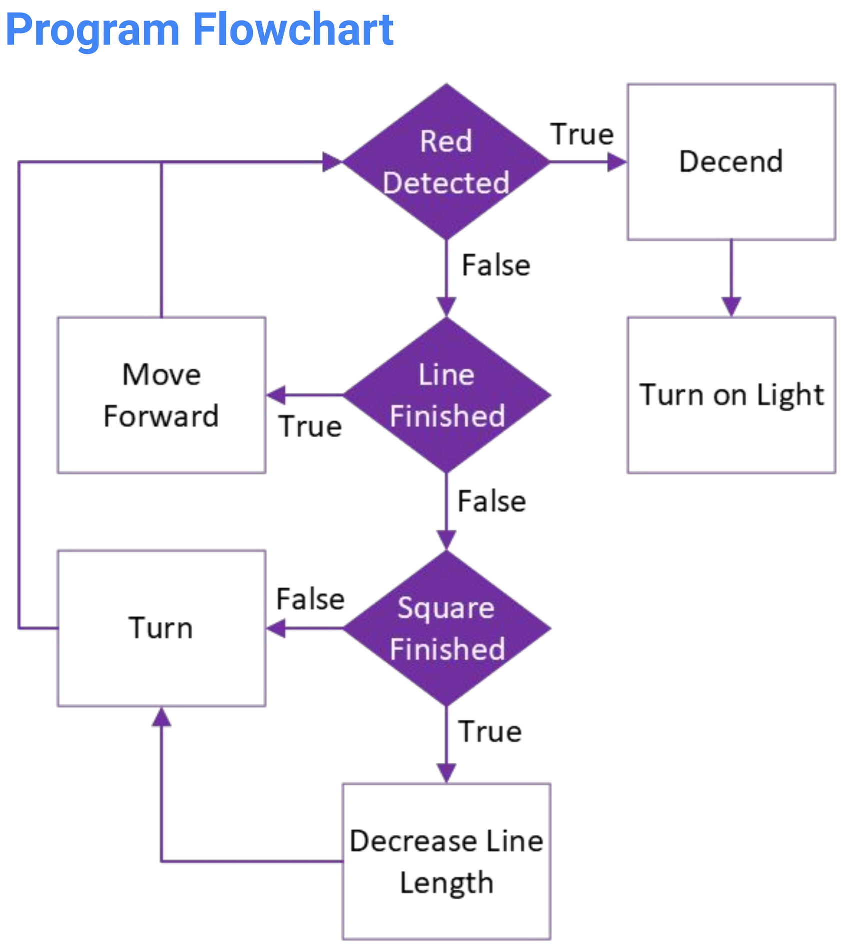


Fig. 6. Sinking mechanism in machine

Implemented Process

Program Flowchart



Program Pseudocode

```
Setup output pins
Define variables for speed, lengths,
square side, line progress, found status
While no red found
    Check for red
    If no red
        If line progress < length
            Both motors on
            Increase line progress
        Else
            If square side < 4
                One motor on
                Increase square side
            Else
                One motor on
                Set square side to 0
                Decrease line length
            Set line progress to 0
Descension motor on
Light on
```

Detection Script

- To detect and locate target
- Image of sent from camera periodically
- Image then converted into Hue Saturation Value (HSV) using OpenCV
  - Each color has a range of HSV values
  - Range for red defined as NumPy array
- Mask applied with OpenCV
  - Filters out pixels with HSV values out of red range
- Program iterates through masked image for red pixels
  - If red pixels found, program is signals dissension
  - Else machine continues searching

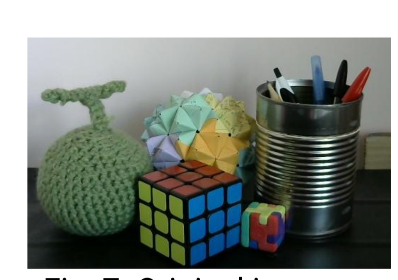


Fig. 7. Original image

Fig. 8. HSV image

Fig. 9. Masked image

Partial Program

```
# Define HSV values
redMin = np.array([0, 70, 50])
redMax = np.array([10, 255, 255])

# Set condition
found = False
while not found:

# Read image from camera
_,img = cam.read()

# Convert image to HSV
hsv = cv2.cvtColor(img,
cv2.COLOR_BGR2HSV)

# Filter for red in image
mask = cv2.inRange(hsv, redMin, redMax)

# Search for red in image
for i in range(0, len(mask), 50):
    for n in range(0, len(maskasi]), 50):
        if mask[i][n] == 255:
            found = True
```

Search Method

- Search entire area in shortest amount of time
  - Minimum turns, maximum straight line movement
- Always choose next fastest move
  - Starting from close-right corner

1. Straight line along border
2. Single 90 degree turn
3. Repeat for three sides
4. Decrease length for fourth side
5. Repeat in concentric rectangles

- Until area covered or target found

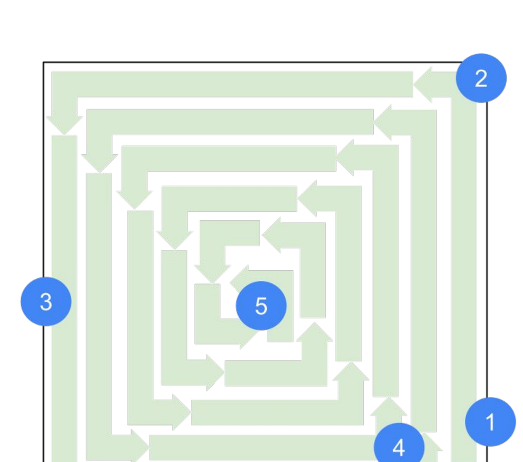


Fig. 10. Search method diagram

Rational

- **Problem**: Aquatic environments challenge human mobility
    - Lack of oxygen
    - Extreme pressures and temperatures
    - Wet nature of water
    - Physical human limitations
  - 80% of the ocean unexplored, 70% of Earth's surface is covered by water (NOAA)
- **Solution**: Submersible vehicles for underwater mobility
  - Applications of underwater vehicles
    - Marine life research
    - Underwater mechanical repairs
    - Faster aquatic rescue
    - Automated ocean cleanup
  - Reduce risk to human life with machines
  - Increase efficiency underwater with automated alternative
  - Expand underwater opportunities for humans

Engineering Goal

Create a machine with machine learning algorithms capable of underwater movement to detect a red 10 by 10 centimeter target placed within a 120 by 56 centimeter square at the bottom of a body of water, the machine will then move toward the target until the machine makes contact with the target, then the machine will stop moving and signal an attached LED light.

Fig. 1. Image of machine

Abstract

Aquatic environments currently pose a challenge to the human race. Physical limits restrict human mobility underwater, leaving the field vastly unexplored with considerable room for advancement. A submersible device capable of traversing aquatic environments would expand the human ability underwater. Based on the fundamental concept of moving underwater, the project is an underwater mobility unit designed to navigate an aquatic environment and complete a simple task: detect and land on a target. The underwater mobility unit utilizes a novel maneuvering system for sinking and machine learning algorithms for searching. The machine begins at the surface of a 120 by 56 centimeter area of water. The unit will move about the area to search for the red 10 by 10 centimeter square target in a random location at the bottom of the area. Essentially, the objective of the machine is to continue searching until the target is detected, then land and turn on an LED light to signal completion. After ten trials, the unit achieved a 70% success rate of landing on the target, with three failures caused by water disturbance. Improvements to the container and turning mechanism would make the unit more stable in water, circumventing issues of water disturbance. Future experimentation could include a larger area with varying targets to further challenge underwater movement. Features of the current machine functioned as intended and could be implemented in more extensive projects with real-world applications.

Procedure

1. Place target at random location at bottom of 120 by 56 cm. area of water
2. Place machine in close-right corner of area at the surface of the water
3. Turn on machine, run program, and allow machine to move
4. When LED light is on, record machine made contact with target (true/false)
5. Repeat steps 1 through 4 ten times

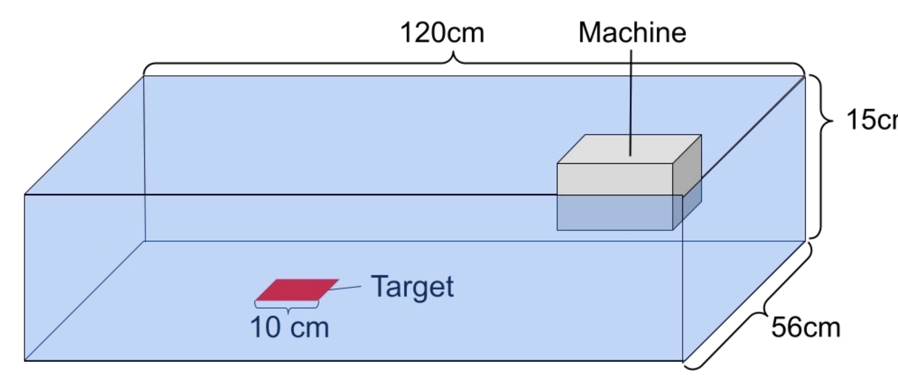


Fig. 11. Diagram of experiment setup

Fig. 12. Image of experiment setup

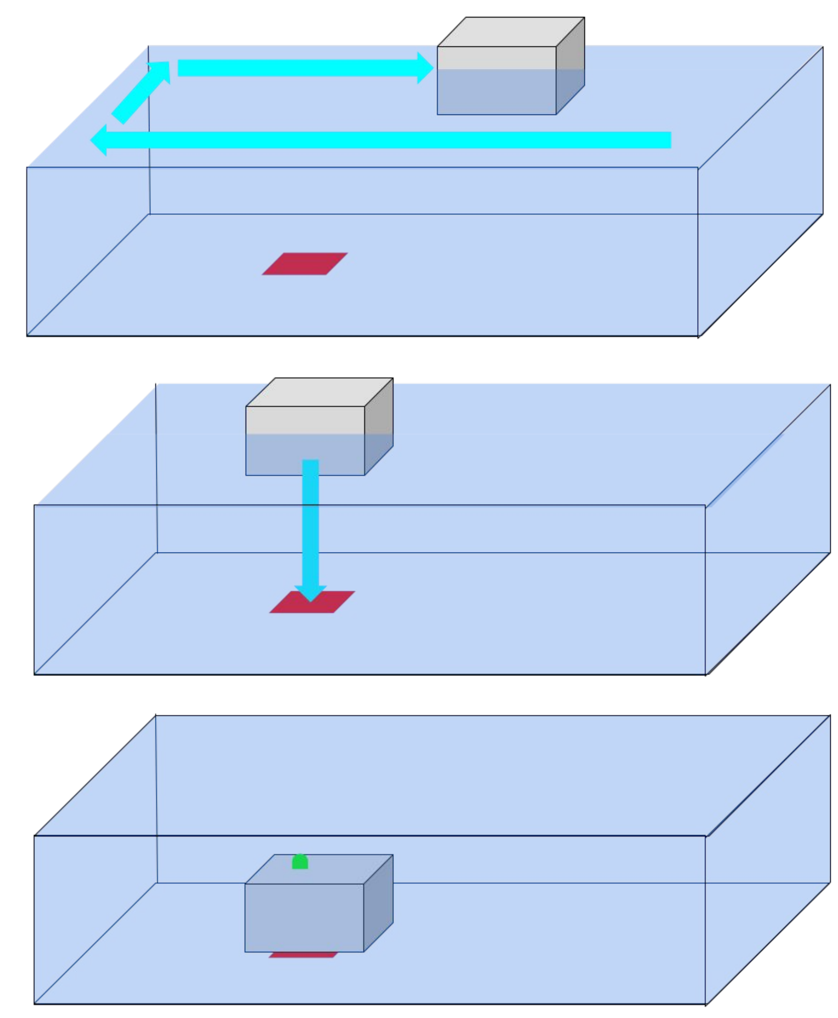


Fig. 13. Image of machine searching

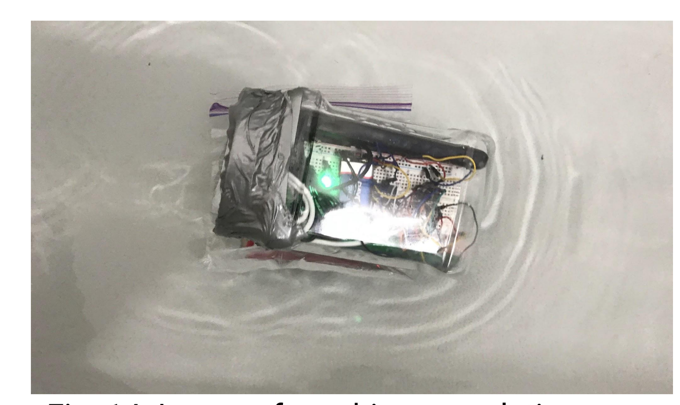


Fig. 14. Image of machine completion

Results

The machine achieved a 70% success rate, failing 3 out of 10 trials.

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

Table 1. Boolean results of 10 trials

Conclusion

Overall success

- Failures due to water disturbance

Further Investigation

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

Limitations

- Slow turning mechanism
- Time consuming process
- Not very stable container

Possible improvements

- Stronger motors for turning and sinking
- Wider scope of camera
- Container for balance and stability

References

Baylakoğlu, İlknur, et. al. "The Detrimental Effects of Water on Electronic Devices." *Advances in Electrical Engineering, Electronics and Energy*, vol. 1, 2021, https://doi.org/10.1016/j.prime.2021.100016. Accessed 23 Feb. 2023

Harris, C.R., et al. "Array Programming with NumPy." *Nature*. https://doi.org/10.1038/s41586-020-2649-2.

"L293D/ Motor Driver IC." *Components 101*, 8 Oct. 2017, https://components101.com/ics/l293d-pinout-features-datasheet. Accessed 23 Feb. 2023.

NOAA. "How Much of the Ocean Have We Explored?" *National Ocean Service*, 20 Jan. 2023, https://oceanservice.noaa.gov/facts/exploration.html. Accessed 23 Feb. 2023.

OpenCV. *Open Source Computer Vision Library*, 2015.

"Raspberry Pi Documentation." *Raspberry Pi*, www.raspberrypi.com/documentation/computers/os.html. Accessed 16 Nov. 2022.

Raspberry Pi Pin Layout. *Raspberry Pi*, www.raspberrypi.com/documentation/computers/os.html. Accessed 16 Nov. 2022.

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 16 Nov. 2022.