PET FEEDING MODULE

This project shows a small demonstration of automatic pet feeding using ultrasonic sensors. A signal is sent to turn on the webcam if an object is detected within the specified range. Then tested to see whether it's a cat or dog, and then a cat or dog food is provided based on what is detected.

Things used in this project

1. Hardware

- WIZNET W5100S-EVB-pico
- Ultrasonic Sensors
- Stepper motors
- Jumper wires
- Breadboard

2. Software

- Arduino IDE
- PyCharm

Problem Statement:

Pets make excellent companions for humans. It helps to relieve tension from a busy lifestyle and acts as a deterrent to robbers. Nowadays, pets are regarded as family members. As a result, the pet's overall health must be considered, and a balanced diet must be prioritized. Depending on the pet's size, different amounts of food and nutrients will be required. According to the poll, the obesity problem in LI pets is caused by the pet owner's busy schedule. Furthermore, pets cannot obtain food independently and must rely on the pet owner to feed them. As a result, the Automated Pet Feeder was designed and is now being utilized to handle the problems of feeding the pet.

An automated pet feeder is a product that can be programmed to replace manual feeding and can be set to a specific feeding amount and time. Pet feeder innovation is used to overcome the pet owner's forgetfulness about feeding their pet and avoid extra spending charges by leaving them at a pet hotel.

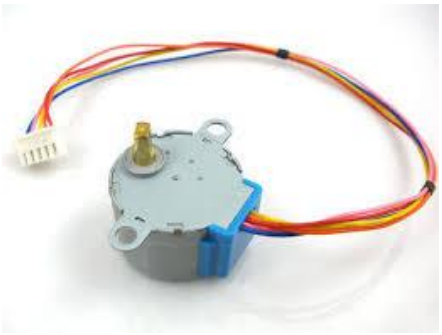
Components:

Ultrasonic sensor:

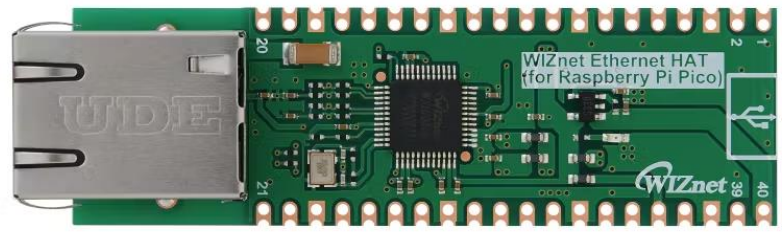
An ultrasonic sensor is an electronic device that uses ultrasonic sound waves to detect the distance between a target item and converts the reflected sound into an electrical signal. Ultrasonic waves travel quicker than audible sound waves (i.e., the sound that humans can hear). The transmitter (which generates sound using piezoelectric crystals) and the receiver are the two primary components of

ultrasonic sensors (which encounters the sound after it has traveled to and from the target).
The formula for this calculation is D = ½ T x C (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second).

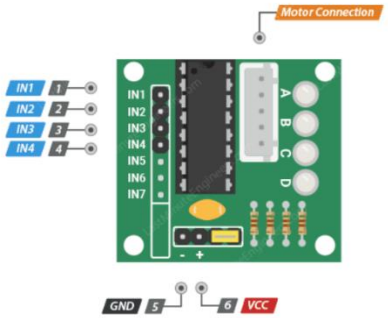
Stepper motor:
A brushless DC electric motor that divides a full rotation into several equal steps is referred to as a stepper motor or a step motor. As long as the motor is appropriately scaled for the application in terms of torque and speed, the position of the motor can be instructed to move and hold at one of these steps without any position sensor for feedback (an open-loop controller).



W5100S-EVB-Pico:
The W5100S-EVB-Pico is a microcontroller evaluation board that uses the Raspberry Pi RP2040 microprocessor chip and the W5100S complete hardwired TCP/IP controller chip. Because the W5100S-EVB-Pico performs the same function as the Raspberry Pi Pico platform and includes the W5100S, the Ethernet function is essentially incorporated.



ULN2003 Stepper motor driver:



- The motor is driven by the IN1–IN4 pins. Connect them to a pico board
- GND stands for "common ground pin."
- The motor is powered by the VDD pin. Attach it to a 5V external power source.
- Motor Connector The motor is plugged into this. The connector only fits one way because it is keyed.

MCU-Micro USB Breadboard 5V Power Supply Module:

5V Power Supply Module 5Pin Female Connector B type PCB Converter for MCU-Micro USB Breadboard. This breakout board is excellent for connecting USB signals to microcontroller boards that you have specifically created. It is also helpful if all you need is access to 5V USB power for your breadboard, perforated board, or Vero board prototype circuits.

Pinouts:

VCC: 5V Signal ,D+: USB D+ Power Supply, D-: USB D- Signal, ID: OTG Devices USB Identifier Signal ,Power Ground: GND

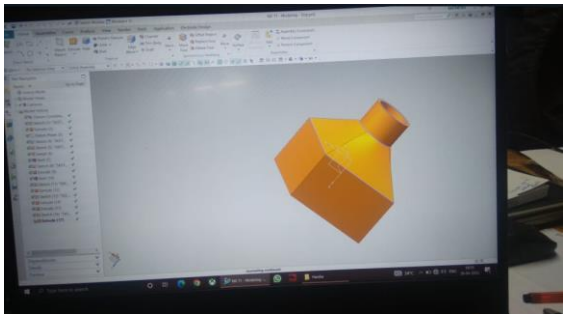
3 D Model:

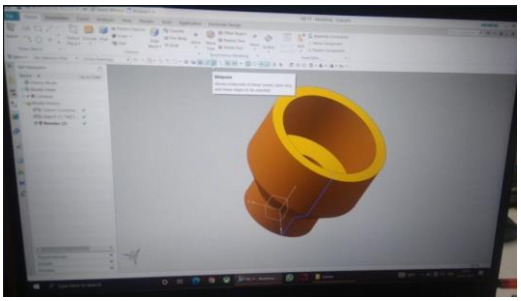
A 3D model can be used to control machinery in construction. These reproductions include the physical environment's points, lines, and surfaces. They make use of coordinate information, which shows where horizontal and vertical points are with relation to a fixed point. These spatial linkages allow you to observe the representation from different perspectives.

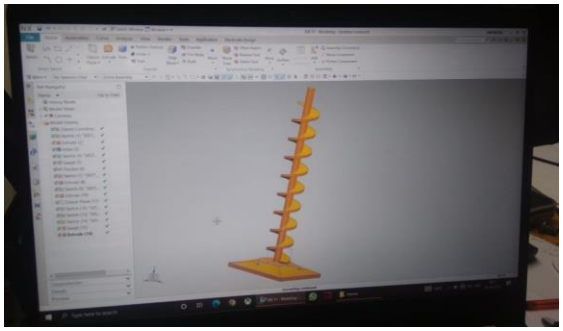
3 D printer used to print Archimedis screw.

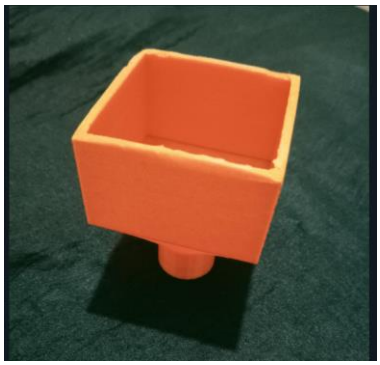
Archimedis Screw:

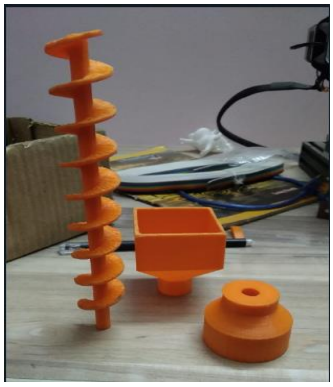
The spiral component of the Archimedes screw can be inside, but for this example, it will be placed within the hollow cylinder. The discharge area is at one end, while a low-lying food source is at the other. Simply rotating the screw will move the food. A small amount of food is scooped up while the screw rotates.



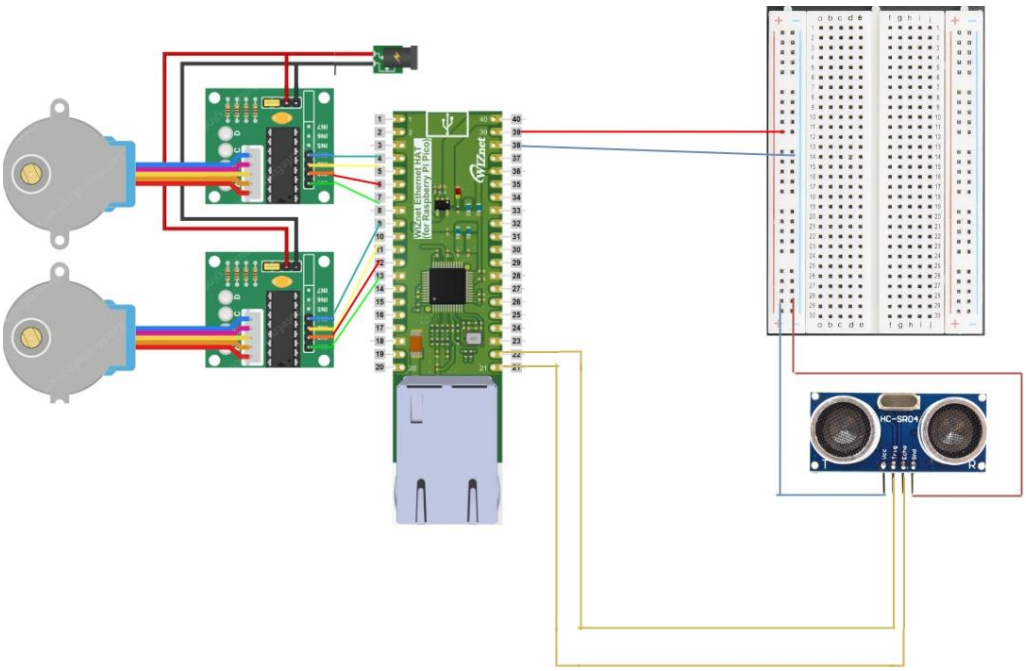








Circuit Diagram:



Explanation:

- Stepper motors and ultrasonic sensors link to VCC and GND through their corresponding VCC and GND pins to ppower supply(12V).
- The Trig pin of the ultrasonic sensor is connected to GP16 of the Wiznet Pico board.
- The Echo pin is connected to GP17 of the Wiznet Pico board.
- GP2, GP3, GP4, GP5 are connected to the IN1, IN2, IN3, IN4 of one stepper motor, respectively, and GP6, GP7, GP8, GP9 are connected to the IN1, IN2, IN3, IN4 of another stepper motor, respectively.

- ❖ If an object is seen in front of the ultrasonic sensors, a serial write is sent to the Pycharm to activate the camera to identify a cat or dog.
- ❖ Either of the stepper motor will rotate if the webcam detects a cat or dog.

Code for arduino :

```
#include <Stepper.h>
#define trigPin 15
#define echoPin 14

const int stepsPerRevolution = 2048;  // change this to fit the number of steps per revolution
// ULN2003 Motor Driver Pins
#define IN1 6
#define IN2 7
#define IN3 8
#define IN4 9

#define IN6 2
#define IN7 3
#define IN8 4
#define IN9 5

// initialize the stepper library
Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);
Stepper myStepper2(stepsPerRevolution, IN6, IN7, IN8, IN9);

void setup() {
Serial.begin (9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
// set the speed at 5 rpm
  myStepper.setSpeed(10);
   myStepper2.setSpeed(10);
  // initialize the serial port
  Serial.begin(115200);
}

String input = "";


void loop() {
```

```
long duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance < 10) {


Serial.write("detected");
//delay(150);
}



input = Serial.readString();
if(input=="Cat")
{
  bool fHasLooped = false;
  if ( fHasLooped == false)
    {
  for(int x=0;x<5;x++)
    {

  Serial.print("clockwise");
  myStepper.step(stepsPerRevolution);
  //myStepper.step(stepsPRevolution);
  }

  }
  fHasLooped = true;
}

if(input == "Dog")
{
  bool fHasLooped = false;
if ( fHasLooped == false)
    {
  for(int x=0;x<5;x++)
    {
```

```
  Serial.print("clockwise");
  myStepper2.step(-stepsPerRevolution);
 // myStepper2.step(stepsPRevolution);
  }

  }
  fHasLooped = true;
}

}
long microsecondsToInches(long microseconds) {
   return microseconds / 74 / 2;
}
```

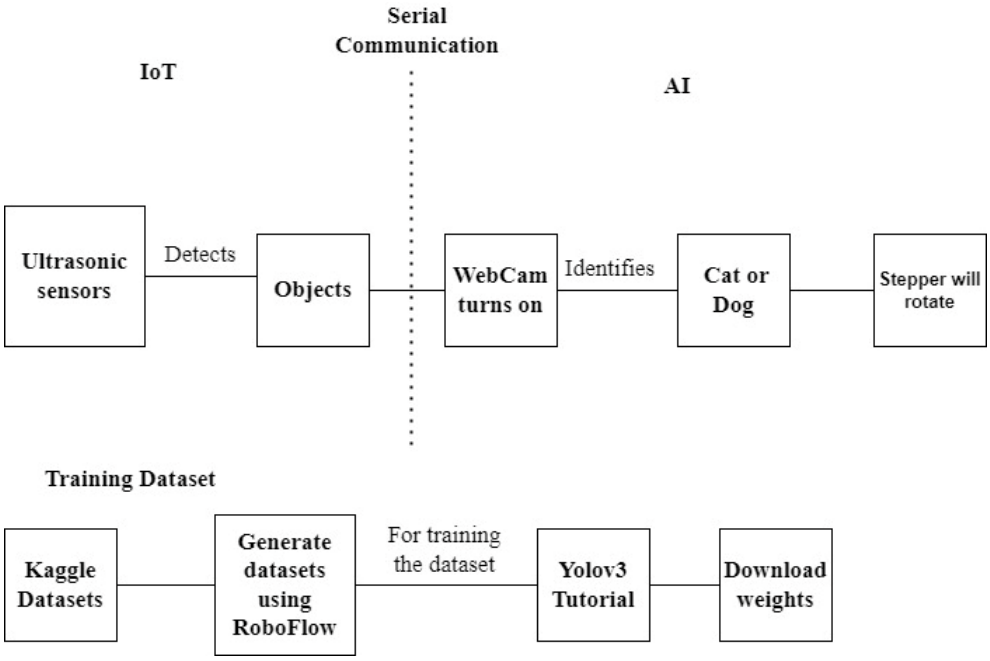
Code Explanation:

- Stepper.h: We can operate unipolar or bipolar stepper motors with this library. We will need a stepper motor and the necessary hardware to control it to utilize it.
- Pin 1 (Vcc): This pin gives the sensor a +5V power supply.
- Pin2 (Trigger): This input pin is used to transmit ultrasonic waves and initiate measurements by keeping it high for 10us.Connected to GP15
- Pin3 (Echo): This output pin goes high for a predetermined amount of time, which corresponds to how long it takes for the wave to travel back to the sensor.Connected to GP14
- Pin4 (Ground): This GND pin is used to attach to the system's GND.
- The distance is then calculated based on the amount of time needed to wait for the sound to be reflected back.
- This formula can be used to determine the precise distance needed to be measured from your sensor:

  Distance = 1/2 T times C
  (T = Time, C = Sound Wave Speed)
  distance = (duration/2) / 29.1
- A serial write is made to the PyCharm "detected" if the distance is less than 10 cm, which activates the webcam for additional detection.
- Either stepper motor will rotate based on the input data acquired(Cat or Dog) through camera detection.
- Here, only a small quantity of food needs to flow, thus I've used a for loop where, if the condition is false, the stepper motor's rotation will halt immediately.

Integrating IoT with AI:

AI-driven IoT produces intelligent technologies that simulate intelligent behaviour and support decision-making with little to no human involvement.



Here, an object is initially located using ultrasonic sensors, and after that, a camera turns to determine if the thing is a cat or a dog, and depending on the determination, either cat food or dog food is fed.

Code for AI:

```
import serial
import cv2
import numpy as np
import time
def main():
    print("Pet Detected!")
    net = cv2.dnn.readNet('yolov3.weights', 'yolov3.cfg')
    classes = []
    with open('coco.names.txt', 'r') as f:
        classes = f.read().splitlines()
    cap = cv2.VideoCapture(0)
    while True:
        initial_time = time.time()
```

```
        final_val = "No pet Reconginzed!! Turn off camera"
        pet = "OFF"
        while True:

            final_time = time.time()
            time_diff = final_time - initial_time
            _, img = cap.read()
            height, width, _ = img.shape
            blob = cv2.dnn.blobFromImage(img, 1 / 255, (416, 416), (0, 0, 0),
swapRB=True, crop=False)
            net.setInput(blob)
            output_layers_names = net.getUnconnectedOutLayersNames()
            layerOutputs = net.forward(output_layers_names)
            val = "No"
            boxes = []
            confidences = []
            class_ids = []
            for output in layerOutputs:
                for detection in output:
                    scores = detection[5:]
                    class_id = np.argmax(scores)
                    confidence = scores[class_id]
                    if confidence > 0.8:
                        center_x = int(detection[0] * width)
                        center_y = int(detection[0] * height)
                        w = int(detection[2] * width)
                        h = int(detection[3] * height)
                        x = int(center_x - w / 2)
                        y = int(center_y - h / 2)
                        boxes.append([x, y, w, h])
                        confidences.append((float(confidence)))
                        class_ids.append(class_id)
                        if 15 in class_ids:
                            val = "Yes"
                            final_val = "Pet Recognized !!"
                            pet = "Cat"
                        elif 16 in class_ids:
                            val = "Yes"
                            final_val = "Pet Recognized !!"
                            pet = "Dog"
                    print("Pet Recongnized : ", val)
            indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
            font = cv2.FONT_HERSHEY_PLAIN
```

```
        colors = np.random.uniform(0, 255, size=(len(boxes), 3))
        for i in indexes:
            if class_ids[i] == 15:
                x, y, w, h = boxes[i]
                label = str(classes[class_ids[i]])
                confidence = str(round(confidences[i], 2))
                color = colors[i]
                cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
                cv2.putText(img, label + " " + confidence, (x, y + 20), font, 2, (255,
255, 255), 2)
            elif class_ids[i] == 16:
                x, y, w, h = boxes[i]
                label = str(classes[class_ids[i]])
                confidence = str(round(confidences[i], 2))
                color = colors[i]
                cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
                cv2.putText(img, label + " " + confidence, (x, y + 20), font, 2, (255,
255, 255), 2)
        # show the detected output in a window
        cv2.imshow('Pet Detection', img)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        def arduino_write(x):
            arduino.write(bytes(x, 'utf-8'))
            data = arduino.readline()
            return data
        # For printing details based on time interval
        if time_diff > 10:
            arduino_write(str(pet))
            print("\nPet recognized: ", final_val)
            print(pet)
            time.sleep(10)
            if pet == "OFF":
                cap.release()
                cv2.destroyAllWindows()
                return None
            break

# execution
while True:
    arduino = serial.Serial(port="COM12", baudrate="115200", timeout=.1)
    line = arduino.readline()
```

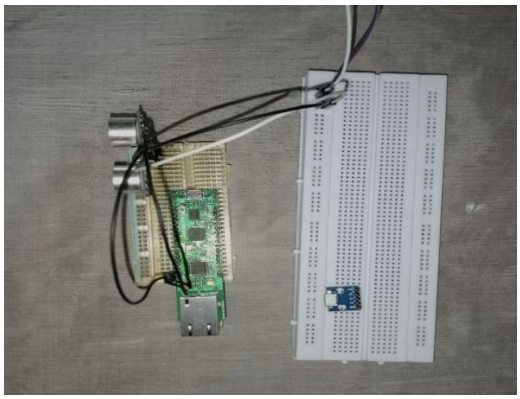
```
response = line.decode()
print("pet", response)
if response == 'detected':
    main()
arduino.close()
```

Code Explanation:

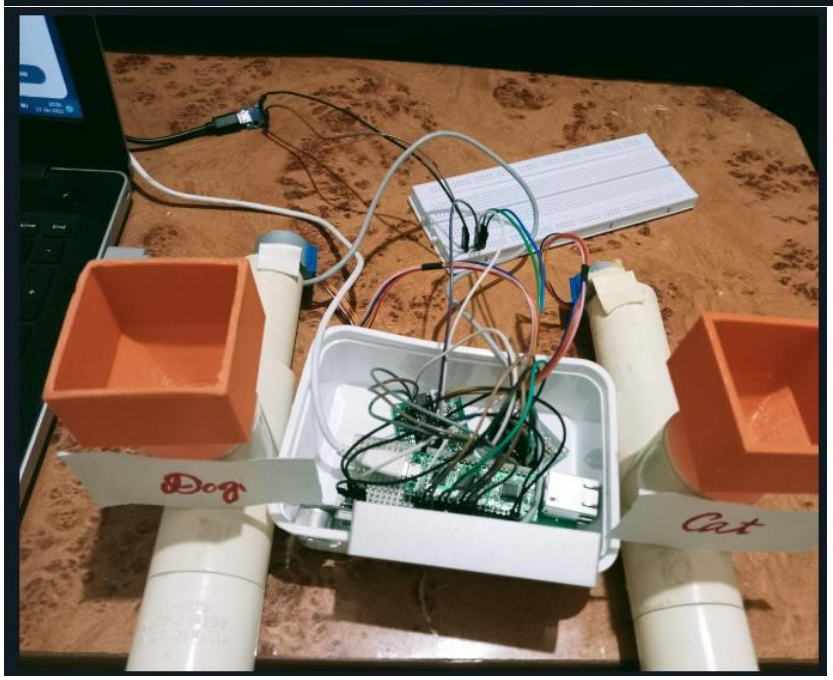
- Using several various devices, PySerial is a library that supports serial connections (RS-232).
- Any object that is detected within the range is first checked. If it is, it will read the input, store it in a line, decode it, and then compare it with the response "detected." If the results are the same, it will call the function start1(), which includes starting the webcam to look for cats or dogs.
- Since our primary objective is to detect cats and dogs, classes consist of a group of ids.
- I have given a condition for cat or dog, and if anything else is observed, it won't display a label, or nothing won't be identified.
- Binary data is written to the serial port, and str is used to convey a string as a series of bytes.
- Here, it covers accuracy with a confidence level greater than 8.
- If nothing is found, the camera will turn off, and the identical procedure will be repeated.

Images:

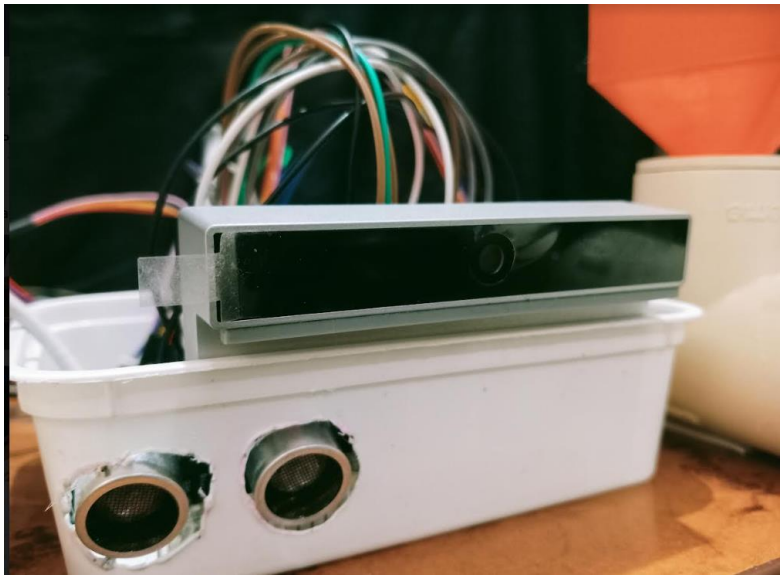
Ultrasonic sensor connection

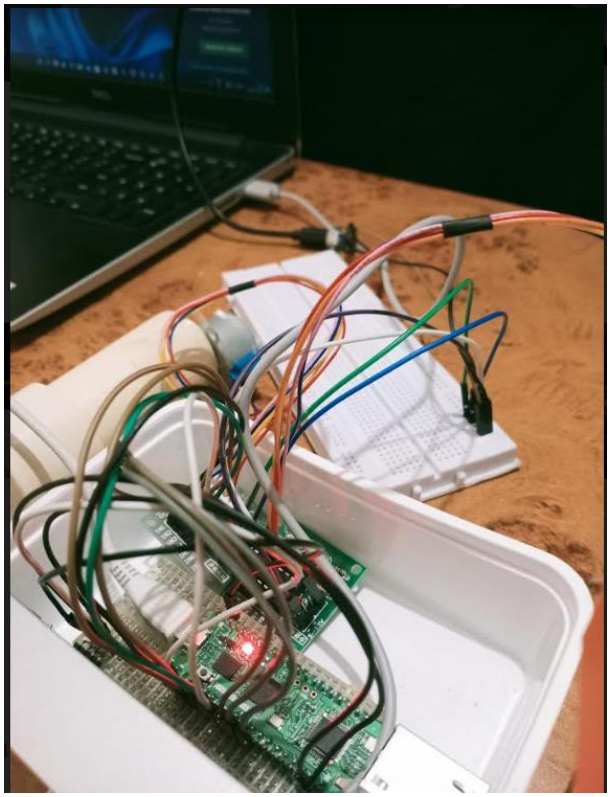


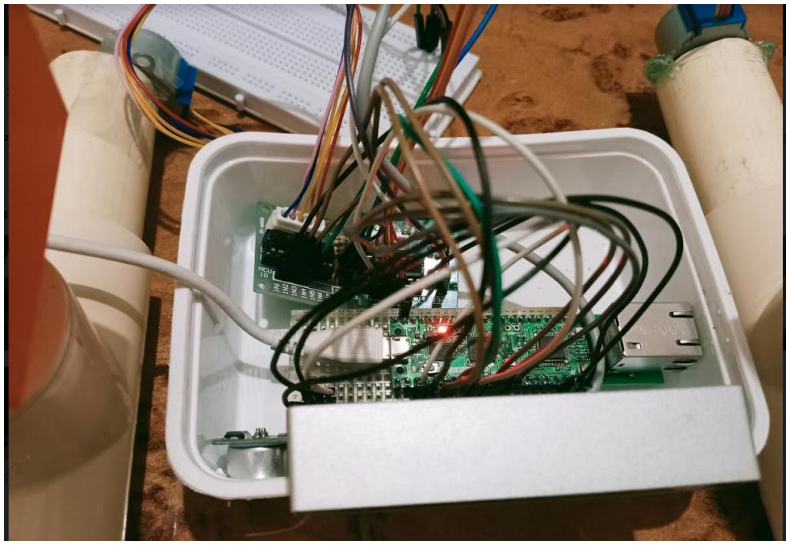
Set up of dispenser:



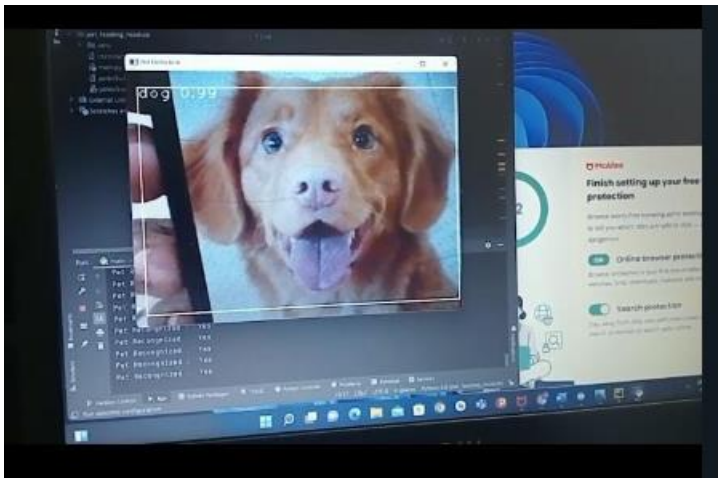
Ultrasonic sensor with camera set up:



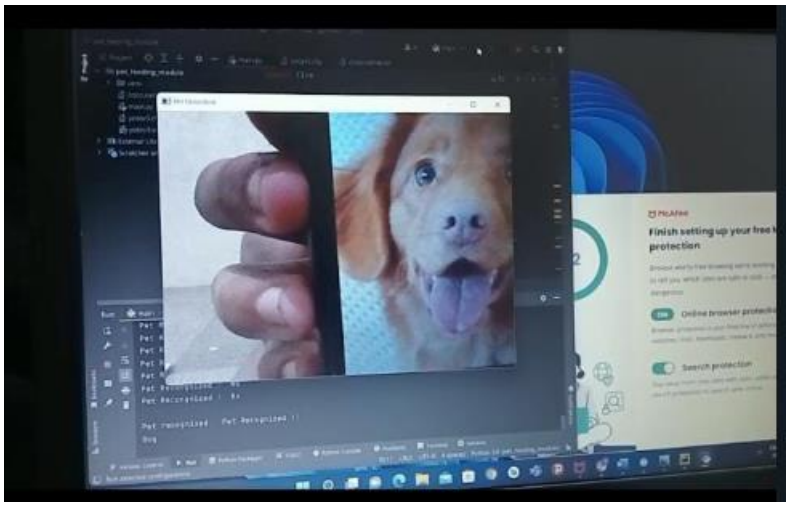




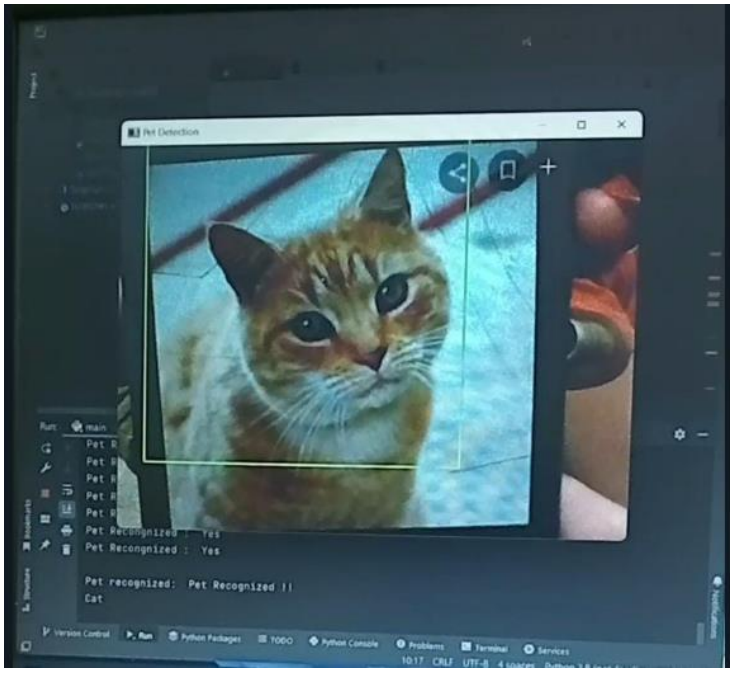
Detecting dog:



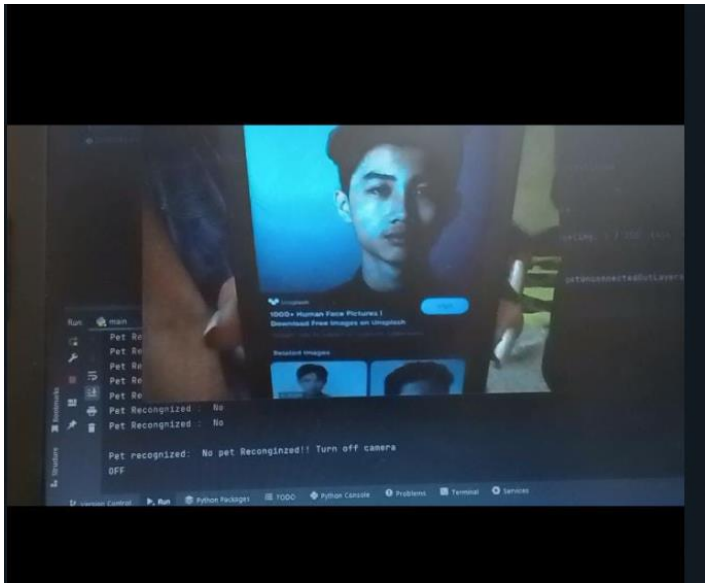
Recognizing Dog and sending it to arduino(control for stepper motor):



Recognizing cat:



When other than cat and dog is detected:



- The webcam will eventually switch off if a cat or dog cannot be found.
- If a cat is identified, cat food is dispensed. If the dog is identified, dog food is dispensed.
- This is how the IoT and AI-powered pet feeding module functions.

References:

3D model:

Archimedis screw:

https://www.youtube.com/watch?v=XTSRnj3XXqw

Code:

https://docs.arduino.cc/learn/electronics/stepper-motors

https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6

https://colab.research.google.com/github/ultralytics/yolov3/blob/master/tutorial.ipynb

https://www.arduino.cc/

Users can use any existing food dispensers such as below IoT project.

https://www.hackster.io/russo08/pet-feeder-with-3d-printed-parts-1f46f2

https://www.instructables.com/Automatic-Arduino-Pet-Feeder-3D-Printed-With-Stepp/