Smart Fishing Rod

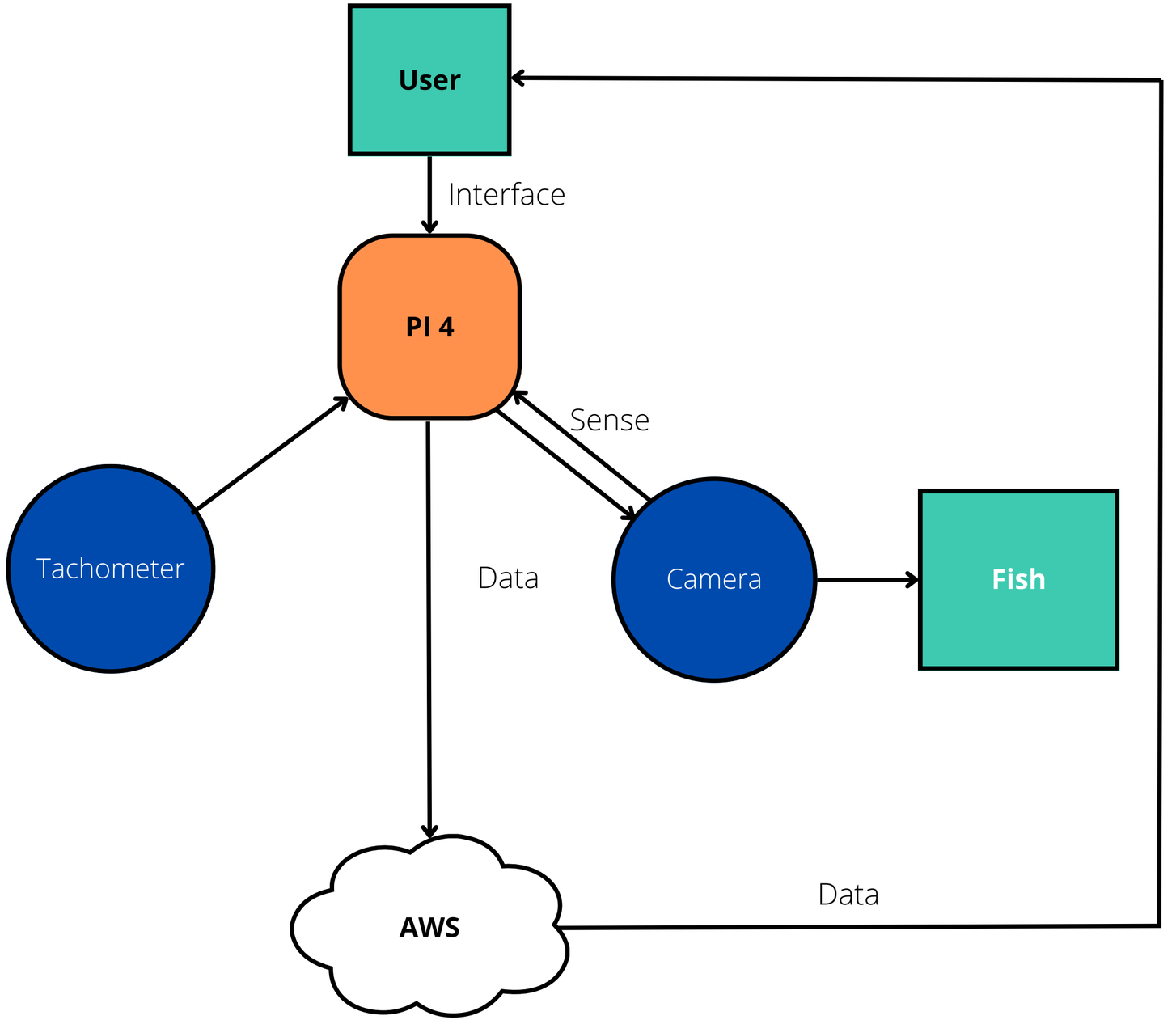
By: Raphael Jayr, Brent Zimmerman, Muhammad Zaid, Trankley Mahler

Introduction

What does it do?

Our cutting-edge fishing rod is designed with innovative features to elevate the angler's experience. Integrated into the rod is a tachometer that precisely measures the speed at which the fishing line departs which allows us to calculate the casting speed, identify bites, and monitor the amount of line being taken by the fish. The rod is also equipped with a camera that livestreams to the cloud. Utilizing a TinyML object identification algorithm, we are able to identify the species of fish, currently achieving a commendable 81% accuracy in distinguishing between a variety of fish.

Block Diagram



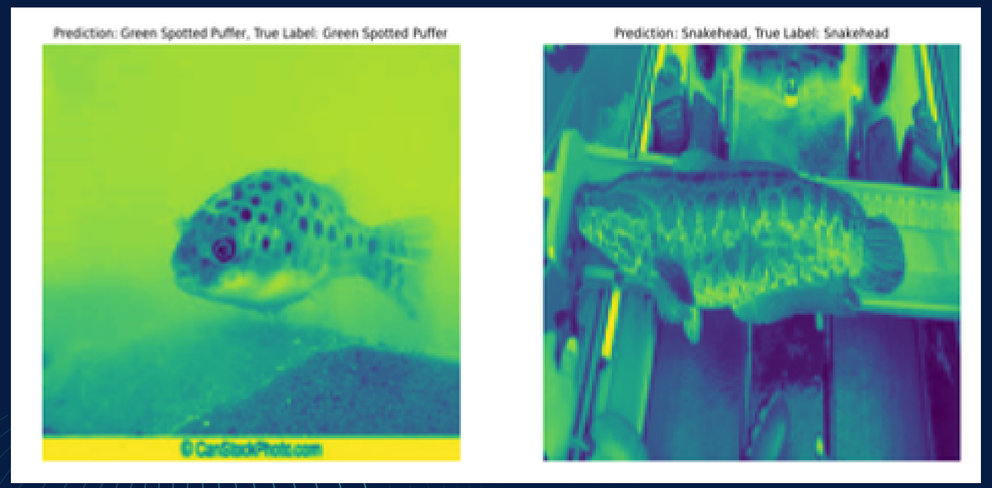
Website and Tachometer

Website shows the tach information and fish specie information

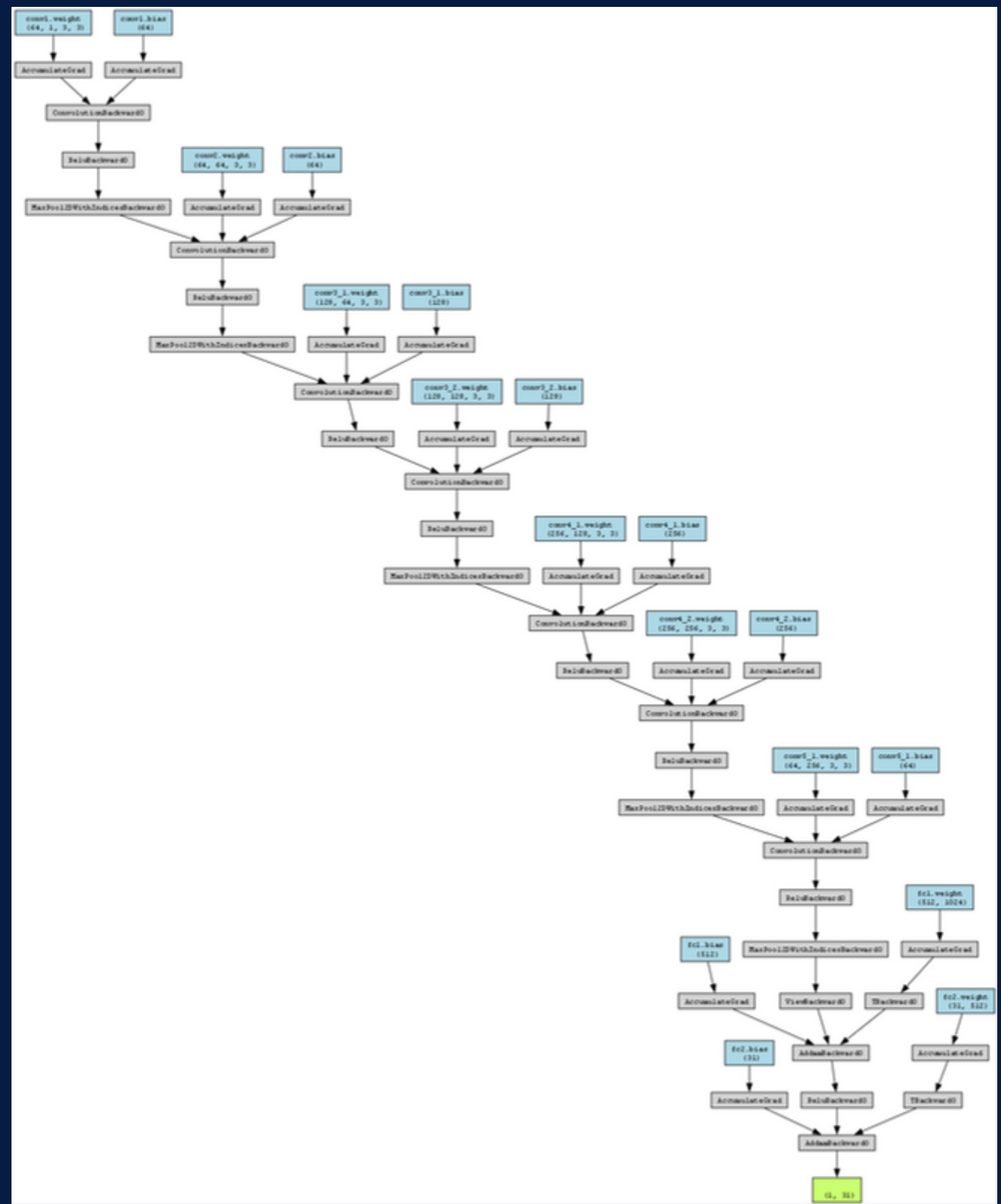
```
Initialize GPIO settings
Initialize sensor_pin, timeOn, timeOff, hole1, rad, circ, dist,
Set up GPIO pin as input

Define SensorThread class:
    Initialize run method:
        Loop:
            Read sensor state
            If sensor_state is LOW:
                If hole1 is 0:
                    If not updatedTime:
                        Set timeOn to current time
                        Set updatedTime to True
                Else if hole1 is 1:
                    Set timeOff to current time
                    Calculate time difference as tempVar
                    Calculate speed as dist / tempVar
                    Print speed
                    Set updatedTime to False
                    Set hole1 to 0
            Else if sensor_state is HIGH:
                Set hole1 to 1
```

CNN - Multiclass Classification

Accuracy of the Model was around 81%

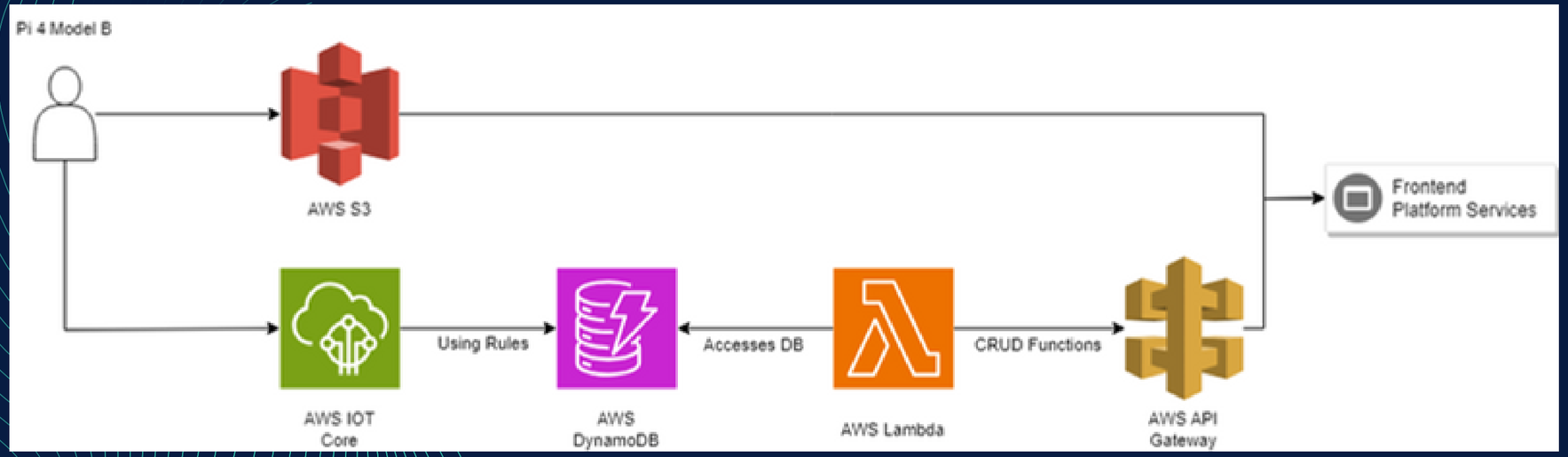


The Architecture of the CNN

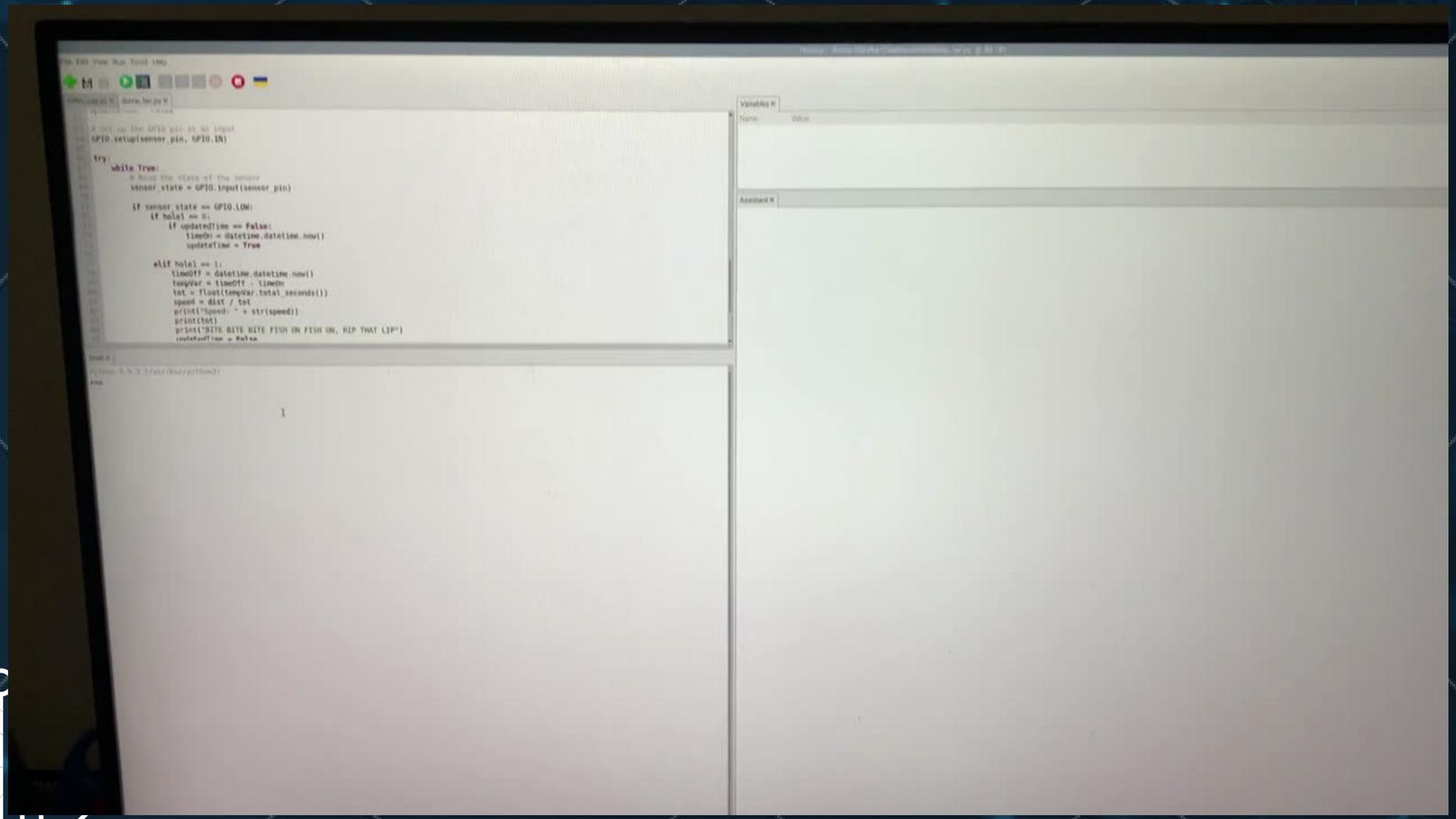


AWS

- AWS IOT Core (MQQT) – Main Communication Protocol
- AWS DynamoDB – Stored the Msgs to IOT Core
- AWS Lambda Functions – Hosted the functions for API GET calls
- AWS API Gateway – API Proxy
- AWS S3 – Video Storage



Video Demo


GPIO.setup(sensor_pin, GPIO.IN)
try:
while True:
sensor_state = GPIO.input(sensor_pin)
if sensor_state == GPIO.LOW:
if holel == 0:
if updatedTime == False:
timeOn = datetime.datetime.now()
updatedTime = True
elif holel == 1:
timeOff = datetime.datetime.now()
tempVar = timeOff - timeOn
tot = float(tempVar.total_seconds())
speed = dist / tot
print("Speed: " + str(speed))
print(tot)
print("BITE BITE BITE FISH ON FISH ON, RIP THAT LIP")
Variables
Assistant
1

Future Improvement Ideas

Measurement Approximation

Utilizing similar technology to that which is in iPhones, we would be able to take a still from the camera and estimate the length of the fish, that partnered with species identification would allow us to let the user know if the fish is of legal size to keep.

Rod Tension Sensor

A sensor on the rod itself that reads the bend of the rod and calculates the torque applied. Using this data we can estimate the size of the fish before it is caught.

Depth Reader

A seperate IoT device that communicates with the Smart Rod that is attached to the line. the device is cast out and provides depth readings back to the user to let them know if the area they are casting their line has sufficient depth.

Lessons Learned

Python

Entering this project, the majority of the group lacked prior experience with Python. At this phase in the project, we are now confident in our proficiency with the language.

AWS

The AWS setup initially proved challenging despite tutorials. Through community forum research, we successfully resolved issues, ensuring proper AWS functionality and data upload from our Raspberry Pi to the cloud.

TinyML

None of the group members had prior experience with machine learning at the project's outset. Despite the initial challenges, we achieved an impressive 81% accuracy on the Object Identification algorithm running on the Raspberry Pi.


That's all Folks!