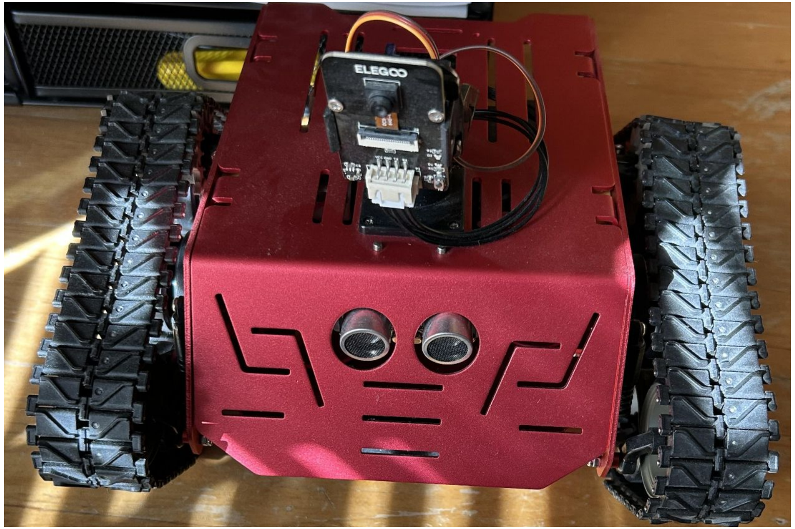


FISH LIVE! (w/ Little Guy)

By: Ben Annicelli, Ben Blechman, Daniel Sanguino

Table of contents

01 Problem Overview

What is Fish Live?

02 Fish Data

What is a fish?

03 Fish Model Evaluation

How well did our model work?

04 Interactive Demonstration

How you can see what little guy sees?

05 Key Challenges and Future direction

Where did we struggle? What’s next?

Problem Overview

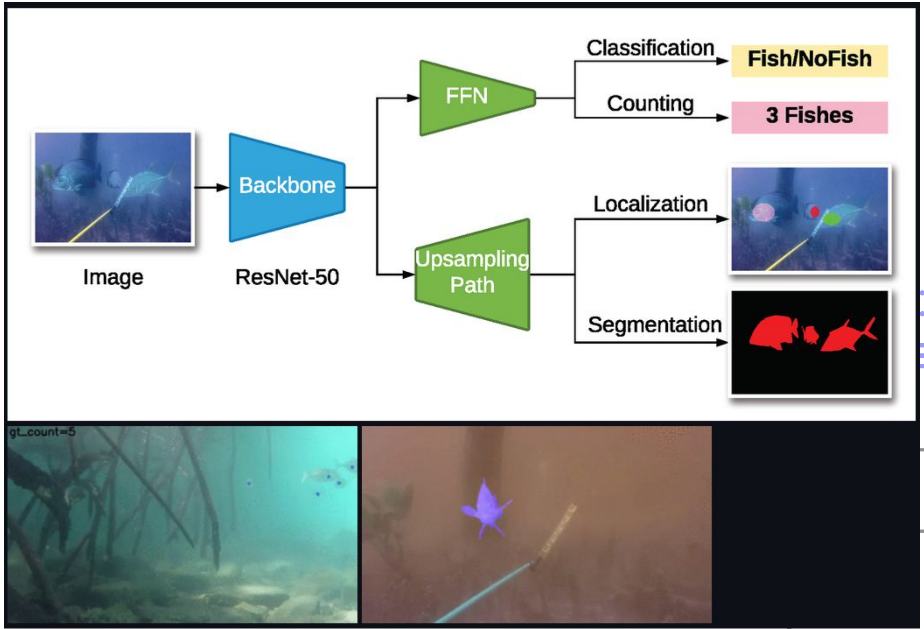
- Fish population monitoring is critical for conservation, research, and aquaculture
- Traditional models are manual, expensive, and disruptive to aquatic life.
- Our project offers a lightweight, real-time fish detection system using a Tiny-YOLO model on a mobile robot
- Enables autonomous fish detection and tracking in natural or controlled environments

Fish Data

Described as “A Realistic Fish-Habitat Dataset to Evaluate Algorithms for Underwater Visual Analysis” by the creators.

Dataset Info

- A 40,000 image dataset from 20 underwater habitats in Australia
- Videos in full HD (1920 x 1080 pixels) resolution
- Total video frames: 39,766 images
- Has 3 different label types: classification (fish or no fish), segmentation (boundary information) and point-level (specific region rather than entire image classification)



Plan for Data Processing

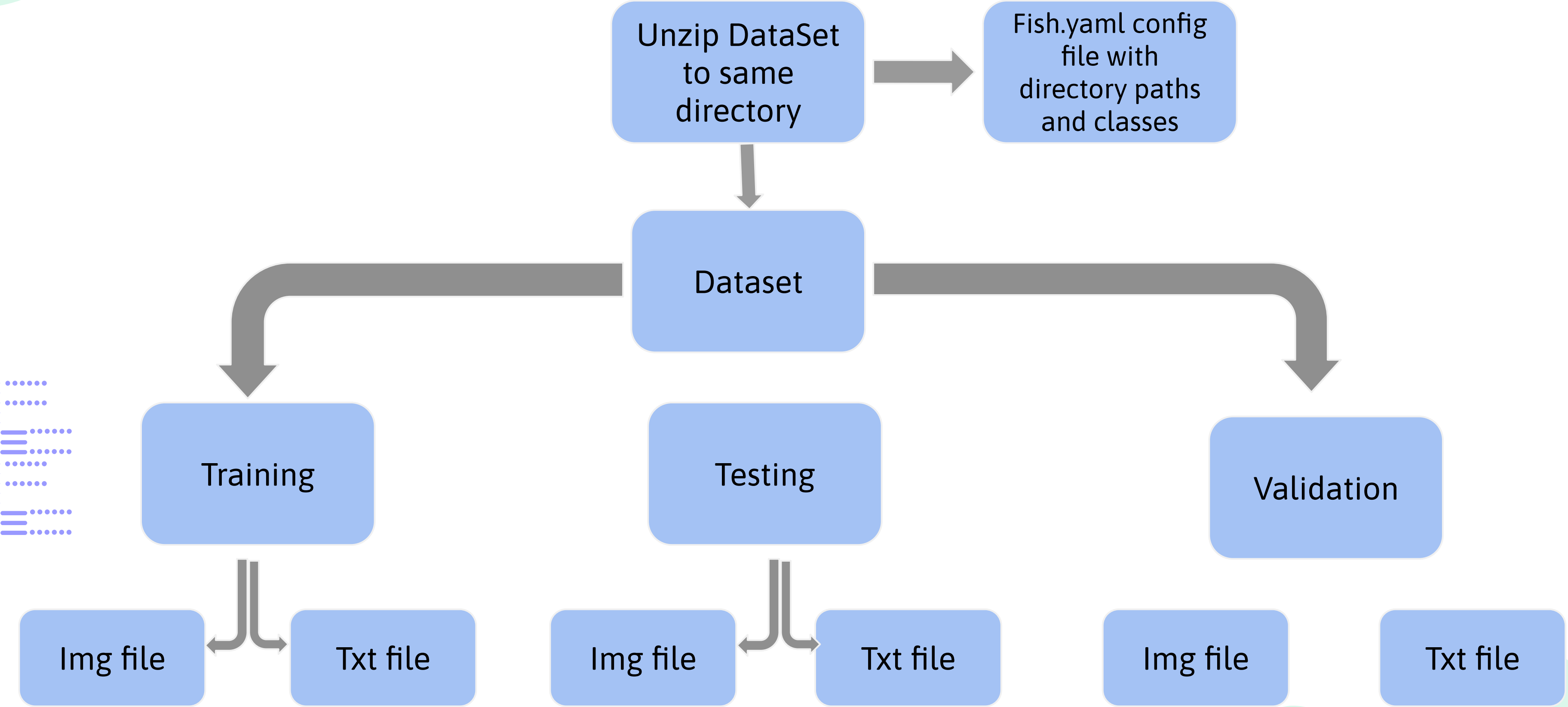
Image Statistics

- **Resolution Analysis:**
  Analyze the distribution of image resolutions to decide if resizing or standardization is necessary.
- **Color Channel Analysis:**
  Evaluate the color channels (e.g., are images in RGB, or do they require color correction due to underwater lighting conditions?).
- **Quality Checks:**
  Identify any blurry, low-quality, or duplicate images that might need to be excluded or pre processed further.

Data Preprocessing

- **Resizing:**
  Standardize image dimensions if your model requires a consistent input size.
- **Normalization:**
  Normalize pixel values (e.g., scaling pixel intensities to the [0, 1] range) to improve model training.
- **Data Augmentation:**
  Since underwater images can be variable, consider augmentations such as:
  - Horizontal and vertical flips
  - Rotations and translations
  - Brightness and contrast adjustments
  - Scaling and cropping
    These augmentations can help improve model robustness and address any imbalance in the dataset.
- **Handling Imbalances:**
  If you notice class imbalances, you might need techniques such as oversampling, undersampling, or applying class weighting during training.

Fish Data Flow



Unzip DataSet to same directory

Fish.yaml config file with directory paths and classes

Dataset

Training

Testing

Validation

Img file

Txt file

Img file

Txt file

Img file

Txt file

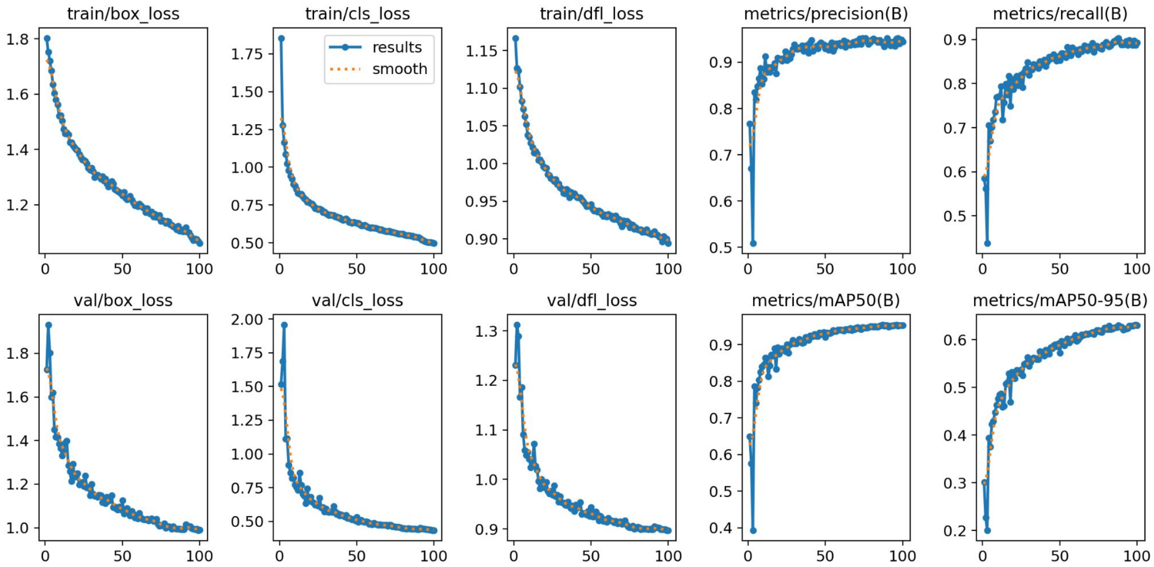
Fish Model Training Process

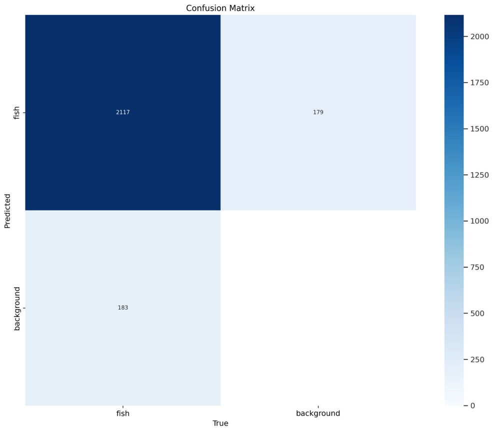
- Used the lightweight **YOLOv11n** model for efficient deployment on Raspberry Pi
- Trained on the **Deep-Fish** dataset using **YOLOv11-compatible format** (images + YOLO labels)
- Used **Weights & Biases (W&B)** for real-time logging of training loss, precision, recall, and mAP

Hyperparameters:

- Image size: 416×416
- Epochs: 100
- Batch size: 16

Model Evaluation Results



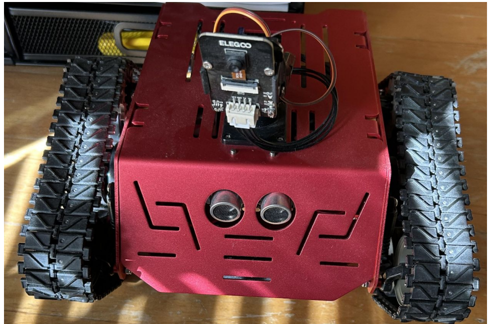


Demo – Little Guy

Little Guy is our stand in for a autonomous underwater vehicle (AUV) since a demo using a AUV would be difficult.

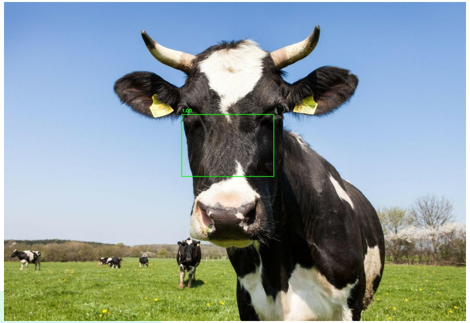
<u>Little Guy Specs:</u>

- Little guy powered by a ELEGOO rechargeable battery pack
- Little Guy moves using 2 DC-Motors attached to treds
- We a Logitech web camera to take in data
- Will use a Raspberry Pi pico for DC-Motor Control
- Raspberry Pi 4B powered by 5V portable Charger



Key Challenges

- Misclassifications of fish during testing
- Difficulties with IP connections for streaming
- Difficulties with creating accurate custom YOLO model



Future Direction

- Apply model on an actual AUV
- Find a better way to determine what fish to track
- Classifying and tracking specific species of fish
- Streaming to hosted website or make an application that comes with Little Guy

Thanks!