**SeaScout: An Underwater Organism Detector**

Student Contributors: Srihari Krishnaswamy & Vivian Wang

**Problem Description**

We built our model and the surrounding scripts to fulfill [MATE's 2023 Computer Coding Challenge](https://files.materovcompetition.org/2023/2023-OER-MATE-ROV-Computer-Coding-Challenge_FINAL.pdf).). This tasked us with identifying marine organisms in seafloor footage collected by Remotely Operated Vehicles.

The competition was divided into two levels:

* Level 1: Classify and place bounding boxes around organisms in the NOAA ship Okeanos Explorer remotely operated clips provided.
* Level 2: Identify the organisms in (a provided) broad morphological category and draw a bounding box for each instance.

**Data**

The classes that our project identifies are summarized below:

|  |  |
| --- | --- |
| **Class** | **Description** |
| Annelids | Segmented Worms |
| Arthropods | Crustaceans (shrimp, crabs, copepods, etc.), pycnogonids (sea spiders) |
| Cnidarians | Sea jellies, corals, anemones, siphonophores |
| Echinoderms | Sea stars, brittle stars, basket stars, urchins, sea cucumbers, sea lilies, sand dollars |
| Mollusca | Cephalopods (squid, octopi, cuttlefish), gastropods (sea snails and slugs), bivalves, aplacophorans (worm-like mollusks) |
| Porifera | Sponges, glass sponges |
| Vertebrates: Fishes | Cartilaginous, bony, and jawless fishes |
| Other Invertebrates | Includes tunicates (sea squirts and larvaceans), ctenophores, many worm phyla |
| Unidentified Biology | Unidentified Biology |

We based our dataset on last year's deepsea-detector project's dataset. We ended up adding a lot of images since at first, our model would label almost everything as fish. This was because of the large concentration of the species in the previous dataset. We added images from the World Register of Marine Species (WoRMS) and had to relabel some of their annotations using Robolfow. Our model was overfitting for a while during our training iterations, and would still label almost everything as fish or arthropods, but we were able to solve this issue by tweaking some training parameters, as discussed in the next section.

**Object Detection Model**

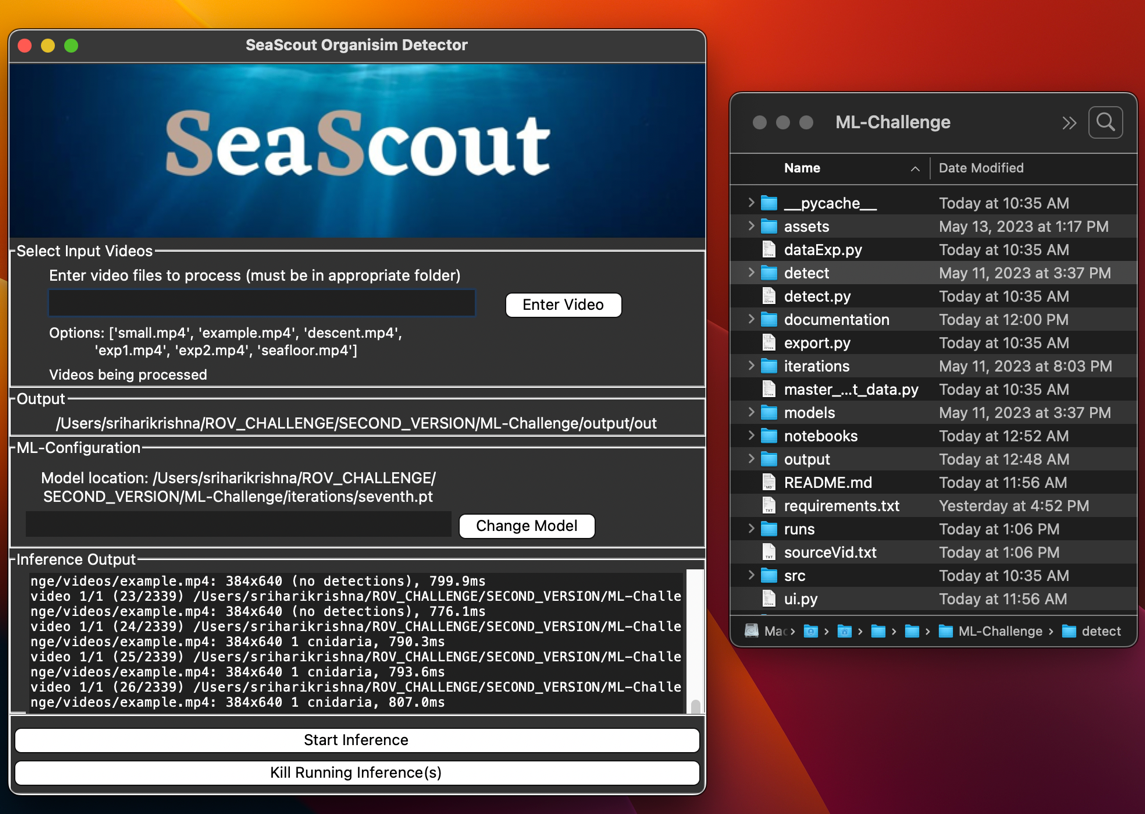
Our project uses a Yolov5 object detection model due to its popularity in the CV field and accuracy. We decided to train a pre-trained yolov5 model from FathomNet, specifically the MBARI Monterey Bay Benthic YOLOv5x model. This was because of the relatively small size of our dataset compared to those of other similar objectives, so we wanted to leverage the fact that the weights of the base model would already be tuned to detect underwater organisms.

The most accurate model we produced (located in the iterations folder of our project) was the result of us freezing (keeping the weights of) 18 layers of the MBARI model and training the rest of the layers with our data. It was important to find a good balance of layers to freeze and unfreeze, since unfreezing all the layers could detract from accuracy since we would have abandoned the weights from the MBARI model. On the flip side, unfreezing less layers would allow our dataset to create less of an impact on the model's weights.

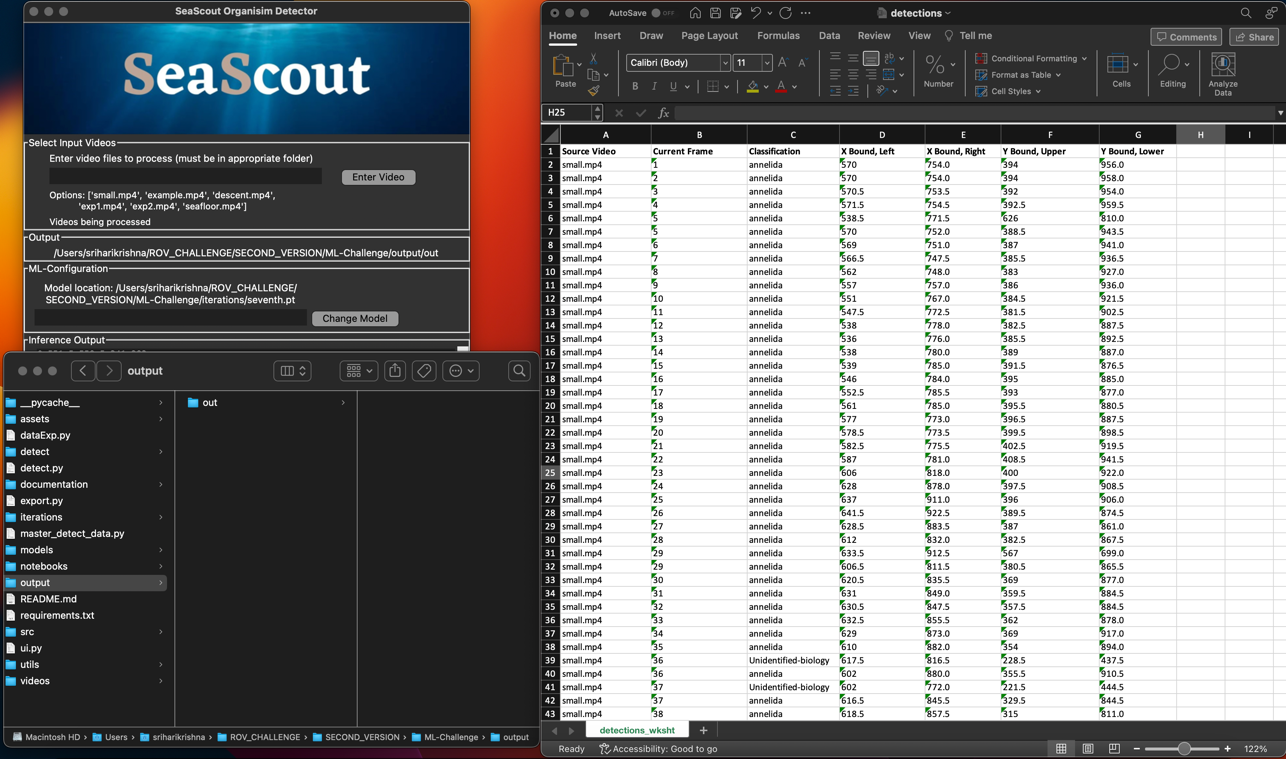
Our model was trained for 14 epochs, and all our training specs are available in our Model Training Colab Notebook. As mentioned, we faced some issues with overfitting while trying to find the optimal model with our dataset, but this was mitigated mainly by changing our number of epochs in training. We strived to create the most accurate model we could with the data we could find, but there are still some inconsistencies with the model's accuracy. Regardless, we still hold that it provides value with its detections and labels.

**UI**

The UI for SeaScout was written with Tkinter. The UI for our project allows the user to enter multiple videos to process and logs the results for the detections for each video on one spreadsheet. After a sequence of videos are processed, the processed videos with bounding boxes are available in the latest folder in the output folder, along with the spreadsheet. The user can cancel video processing at any time, but if this is done, no spreadsheet or videos with bounding boxes will be generated. It is worth noting that the same detection and logging process can be run via the Terminal/CLI, and that the UI is simply a wrapper for this.

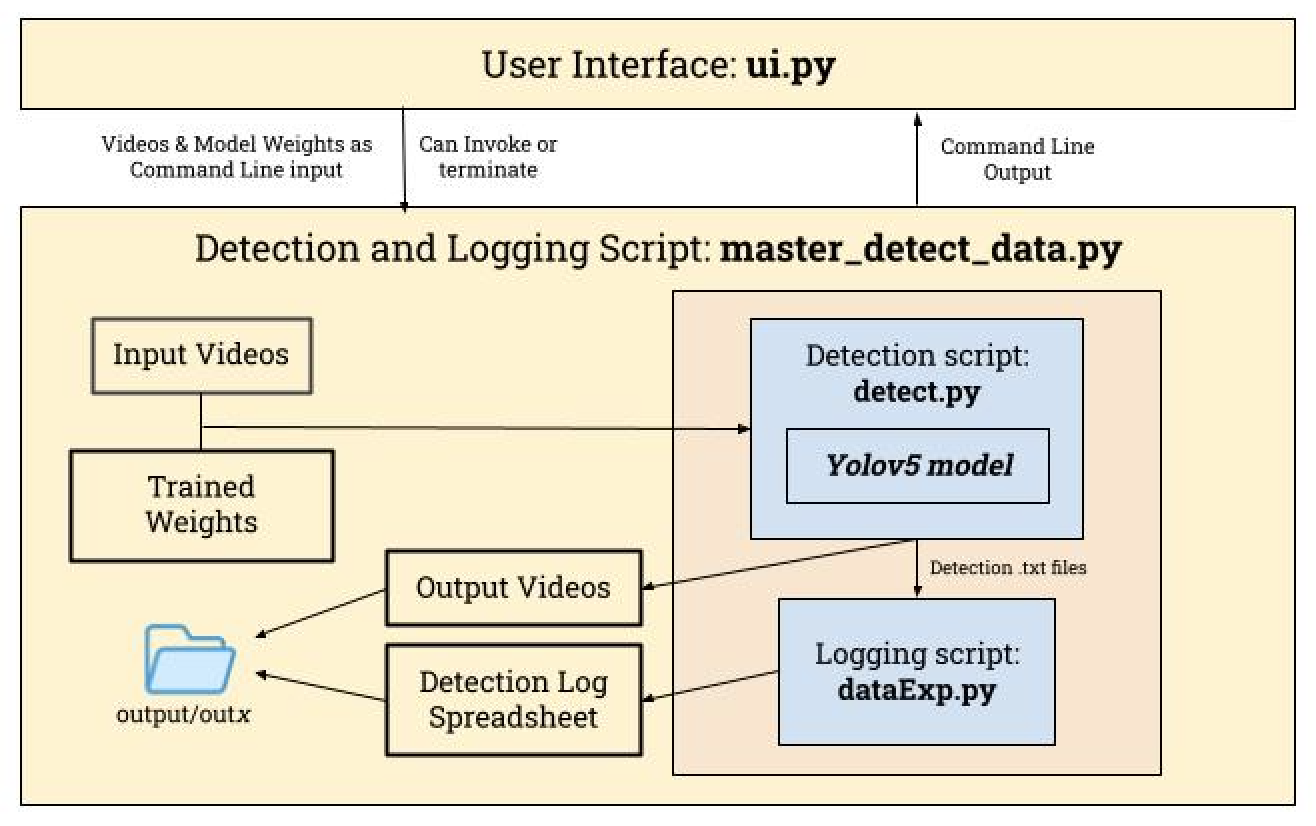


After the inference finishes, the detections are outputted to an excel spreadsheet located in the latest folder inside the output folder. The processed video files will also be available in the same folder:



**Architecture and Code Descriptions**

Below is a figure depicting the base architecture of our program:



**ui.py:**

**master\_detect\_data.py:**

**detect.py:**

**dataExp.py:**

**Results**

Our model performs well with detecting organisms, but generally struggles in classifying them correctly. In the future, we would try to collect even more data to improve the dataset even further to mitigate this issue. Below is a video of the model's annotations of the seafloor video:

Here are the training loss graphics for our model:

**Limitations and Extensions**

One reason for the model's inaccuracy was the sheer amount of diversity present in each class. Multiple organisms that look very different were part of the same categories, something that would directly detract from model accuracy.

Of course, with more time, we would expand our dataset even further, however it is worth noting that annotating images by and finding correctly annotated images is still difficult.

**Acknowledgements and Resources**

We would like to thank MATE and NOAA Ocean Exploration for hosting the ML Challenge. We would also like to thank FathomNet and NOAA Ocean Exploration for providing data. Additionally, we would also like to thank Peyton Lee and the team from last year's UWROV deepsea-detector project for advice, providing their dataset for us to build on. In this year's project, we used a similar structure to them for documentation and the notebook to train our model, as well as the code for showing standard output to the UI.

Additionally, we want to acknowledge our use of Ultralytics Yolov5. The detection.py script is sourced from the Yolov5 repository and the other python files in the project's subfolders are from the repository as well (these are dependencies of detect.py).

**Video Demo and Explanation**

**Citations**

Jocher, Glenn. "yolov5." GitHub, 2023, https://github.com/ultralytics/yolov5

ShrimpCryptid. "deepsea-detector." GitHub, https://github.com/ShrimpCryptid/deepsea-detector

"The Main Differences Between Arthropods and Cnidarians." Biobubble Pets, https://biobubblepets.com/the-main-differences-between-arthropods-and-cnidarians