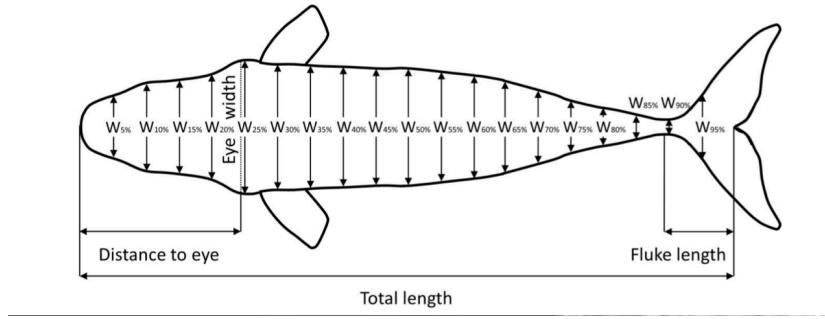


# ***Save the Whales Project***

## VGG Annotation Task Manual

### Introduction:

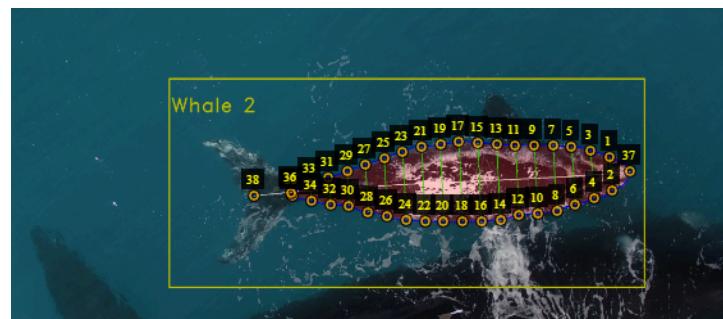
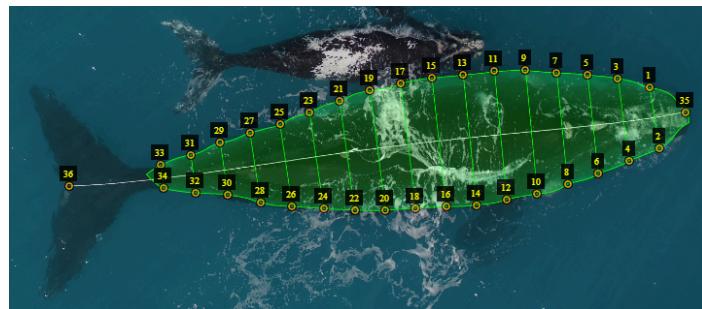
In this project, we try to calculate the body conditions of whales using aerial images. This is done by calculating the widths of the whale at 5% intervals along the whale's body axis.



Previously, these measurements on drone images were done by hand using an R script. In this project, we have automated this process. We have trained a Machine Learning model that can make these measurements. Even though the model does a good job in general, there are some cases where a user might want to go in and make slight changes to the measurements made by the model to make it more accurate. We do this using the [VGG Image Annotator \(VIA\)](#).

## ML Model Measurement Examples:

Next, we have a few examples where the ML does a good job at making the measurements:



For the image measurements using the AI, you will sometimes see the mask (green mask showing the outline of the whale in the first image and a similar red mask in the second image) not be perfect. The axis (marked in white) might also be off sometimes.

## Stage 1- Create JSON file for VGG Image Annotator:

The format of the output CSV file from our ML model is different from the input format for the VGG Image Annotator. Therefore, we need to convert the output CSV file from our ML model to the correct format for VGG. This can be done using the following command:

Unset

```
python create_json_for_VIA_from_CSV_outputs.py -i <FULL PATH TO INPUT CSV FILE>  
-o <OUTPUT JSON FILE>
```

```
## For example:
```

```
python create_json_for_VIA_from_CSV_outputs.py -i  
/outputs/output_output_2023_03_08_220616/predicted_data.csv -o  
vgg_image_anno_data.json
```

## Stage 2 - Setup:

1. Go to <https://www.robots.ox.ac.uk/~vgg/software/via/via.html>
2. Have your set of images and the JSON file ready.
3. Go to VGG Annotator Online site (will work best with Chrome)
4. Under **Project -> Load**, select the whale\_annotation\_attributes JSON file that describes the attributes to be annotated. Once loaded, you should be able to see the image names (under Project) and the Attributes on the left

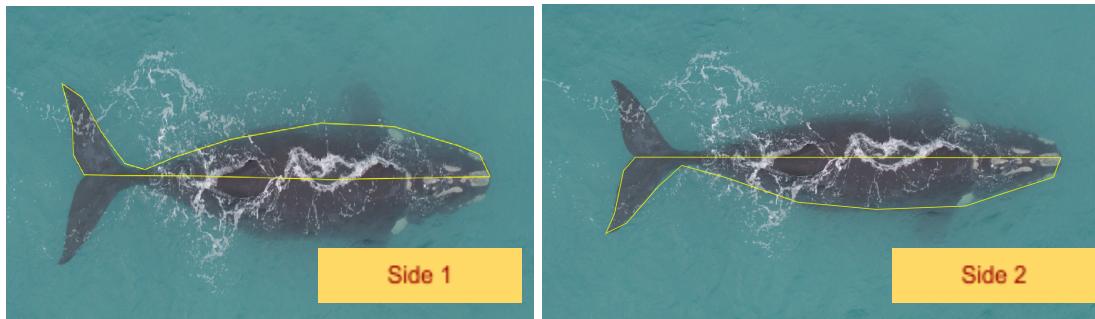
The screenshot shows the VGG Annotator interface. On the left, there's a sidebar with icons for Region Shape, Project, and Attributes. The Project section shows a list of files under 'via\_project\_10Nov2023\_12'. The Attributes section shows a list of attributes with dropdown menus for Name, Desc., and Type. Below these are tables for Region Attributes and File Attributes, each with columns for id, description, and def. The 'File Attributes' table lists several attributes for 'side1' with radio button options.

id	description	def.
side1_5		<input type="radio"/>
side1_1		<input type="radio"/>
side1_1		<input type="radio"/>
side1_2		<input type="radio"/>
side1_2		<input type="radio"/>
side1_3		<input type="radio"/>
side1_3		<input type="radio"/>
side1_4		<input type="radio"/>
side1_4		<input type="radio"/>
side1_5		<input type="radio"/>
side1_5		<input type="radio"/>

- Once the attributes are loaded, under **Project ->Add local files** select all of the set of images that you will be working on to be uploaded.

### Stage 3 - Annotation:

For ease of annotation, we will be logically splitting the whale into *2 sides*, separated by the axis.



Please make sure you are consistent with the *side* number during keypoint selection. It doesn't matter which is exactly side 1 or side 2 but the keypoints for each side need to be consistent.

### Width measurements:

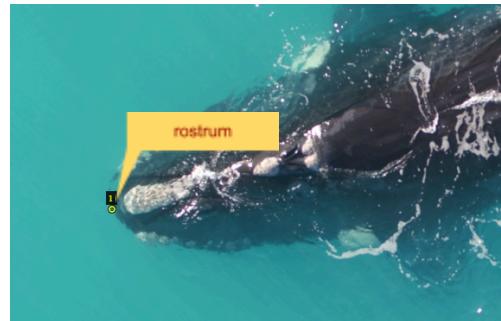
The width measurements at 5% intervals are done in the following way:

- The axis of the whale is estimated.
- Perpendicular lines are dropped at 5% intervals along the axis.
- The point where these perpendicular lines intersect with the whale's outline is marked as a point.

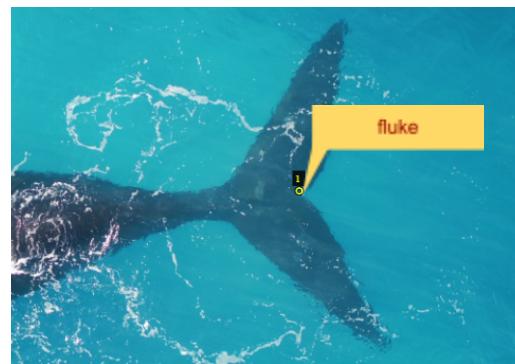
The ML model might make an error in estimating the axis or the outline of the whale. In this case, a human would need to correct this error and that is the task here.

The following points should exist on every image:

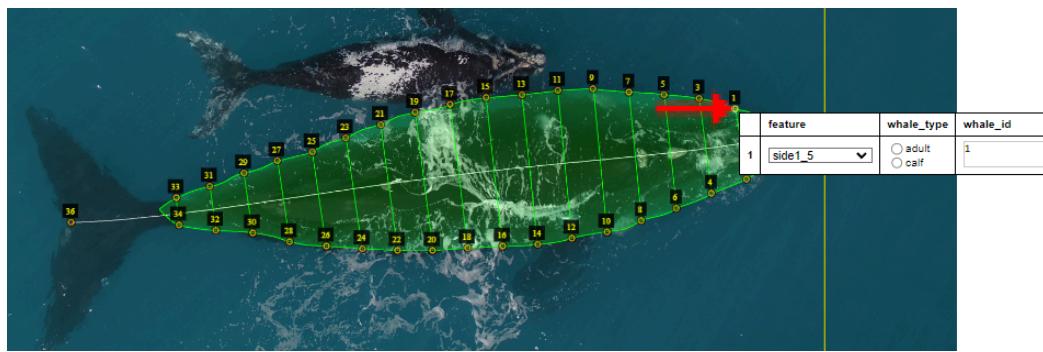
- Rostrum (tip of the mouth)

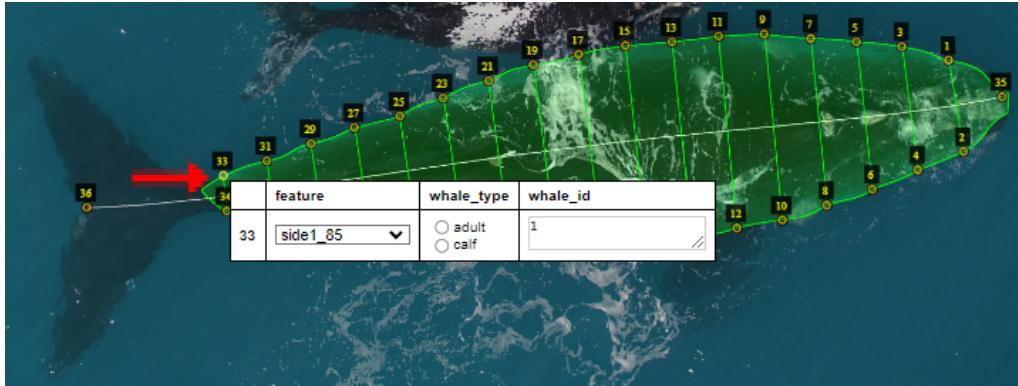


- Fluke (notch in the tail)

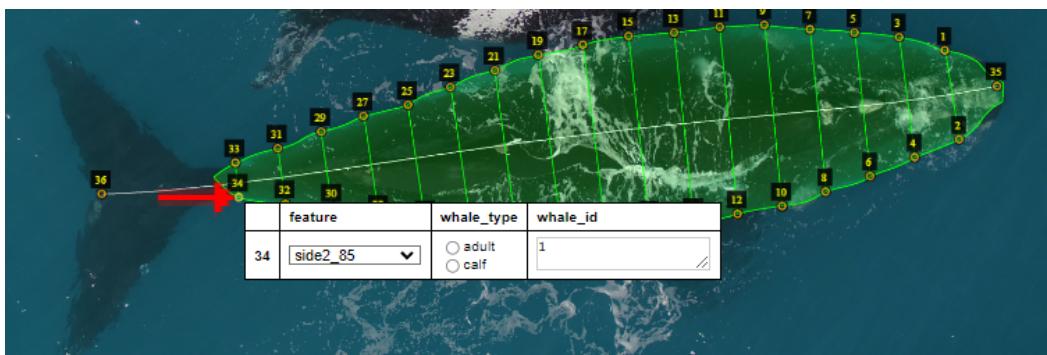
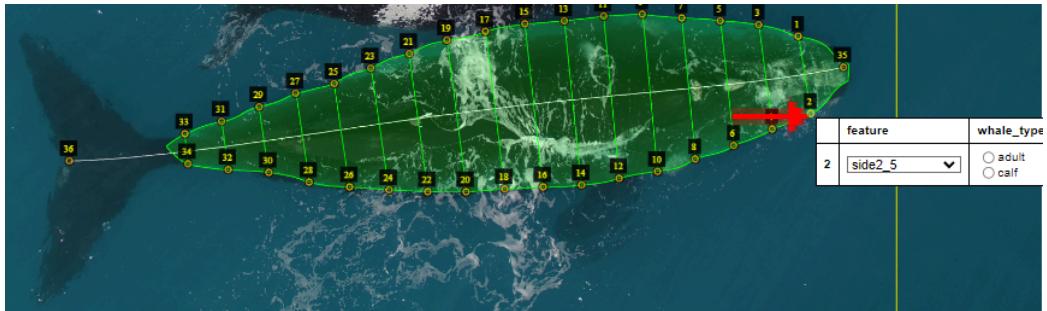


- Points on side 1: Points named side1\_5, side1\_10, side1\_15, ..., side1\_95 marked on side 1 starting from near the head (side1\_5) and going to the tail (side1\_95).





- Points on side 2: Points named side2\_5, side2\_10, side2\_15, ..., side2\_95 marked on side 2 from near the head (side2\_5) and going to the tail (side2\_95)



- From the **Region Shape Tab** on the left select the point as the mode of annotation
- You may zoom in via the zoom in button on the upper border or with the specified keyboard shortcut to focus on the area to place point annotation. And to see each point marked with their corresponding attribute, it is helpful to Open up the **Toggle Annotation Editor** (seen under the **Attributes Tab**)

3. All whales need to have points marked with the following:
  - a. Rostrum
  - b. Fluke
  - c. Side1\_5, side1\_10, ..., side1\_95 (18 points in total for side1)
  - d. Side2\_5, side2\_10, ..., side2\_95 (18 points in total for side2)
4. You can perform the following operations:
  - a. **Do nothing** if you think the point is correctly placed.
  - b. **Add** a point that is missing.
  - c. **Move** a point that is incorrectly placed. Alternatively, you can remove the old point and add a new point with the same label as well. To remove a point, select the specific point and click on "X" in the top bar.
5. You can mark the following points as well (optional):
  - a. Dorsal Fin (dorsal\_fin\_start, dorsal\_fin\_end)
  - b. Ends of the fluke (side1\_fluke, side2\_fluke)
6. After all the images have been checked, download the corrected annotations using **Project -> Save**.

#### Stage 4: Convert outputs from the VGG Image Annotator to original CSV format

The saved file from the VGG Image Annotator needs to be converted to the original CSV format of our model outputs. This can be done using the following command:

```
Unset
python create_csv_outputs_from_VIA_json.py -i <FULL PATH TO INPUT JSON FILE> -o
<OUTPUT CSV FILE>

## For example:

python create_json_for_VIA_from_CSV_outputs.py -i vgg_project.json -o
data_with_corrected_outputs.csv
```