Wetherington, Evan



SeeOtter User Guide

A guide into the usage of SeeOtter created for users

Table of Contents

[About 3](#_Toc118458583)

[System Requirements 3](#_Toc118458584)

[User Interface 3](#_Toc118458585)

[Survey Manager Screen (Home Page) 3](#_Toc118458586)

[Otter Checker 9000 Screen 4](#_Toc118458588)

[Settings Screen 6](#_Toc118458589)

[Page Navigation 6](#_Toc118458591)

[Files and File Structure 6](#_Toc118458592)

[Setup 7](#_Toc118458593)

[How to Use 8](#_Toc118458594)

[Open Survey 8](#_Toc118458595)

[Save Survey 8](#_Toc118458596)

[Create New Survey 9](#_Toc118458597)

[Create Survey from Waldo Data 9](#_Toc118458598)

[Run Processing (Generate Predictions) 9](#_Toc118458599)

[Generate Results 10](#_Toc118458600)

[Actions (Commands) 10](#_Toc118458601)

[Transects 11](#_Toc118458602)

[Inclinometer Data 13](#_Toc118458603)

[Camera System 14](#_Toc118458604)

[Clone Filtered Survey 15](#_Toc118458605)

[Handling Ambiguous VAlidations 16](#_Toc118458606)

[Image Validation (OtterChecker9000) 17](#_Toc118458607)

[Image/Prediction Navigation 17](#_Toc118458608)

[Validation 18](#_Toc118458609)

[Drawing Annotations 18](#_Toc118458610)

[Filters 19](#_Toc118458611)

[Settings/Config 19](#_Toc118458612)

[Other 20](#_Toc118458613)

[GPU Acceleration 20](#_Toc118458614)

[Scaling Issues 20](#_Toc118458615)

[Controls 21](#_Toc118458616)

See Otter User Guide

# About

SeeOtter is an application designed to process and validate arial photography. Utilizing the YOLOv5 detection model, it will predict the location of objects within an image, calculate the locations of that object, and allow the user to validate the predictions.

## System Requirements

* Windows PC
* > 8GB ram recommended
* Nvidia GPU recommended

## User Interface

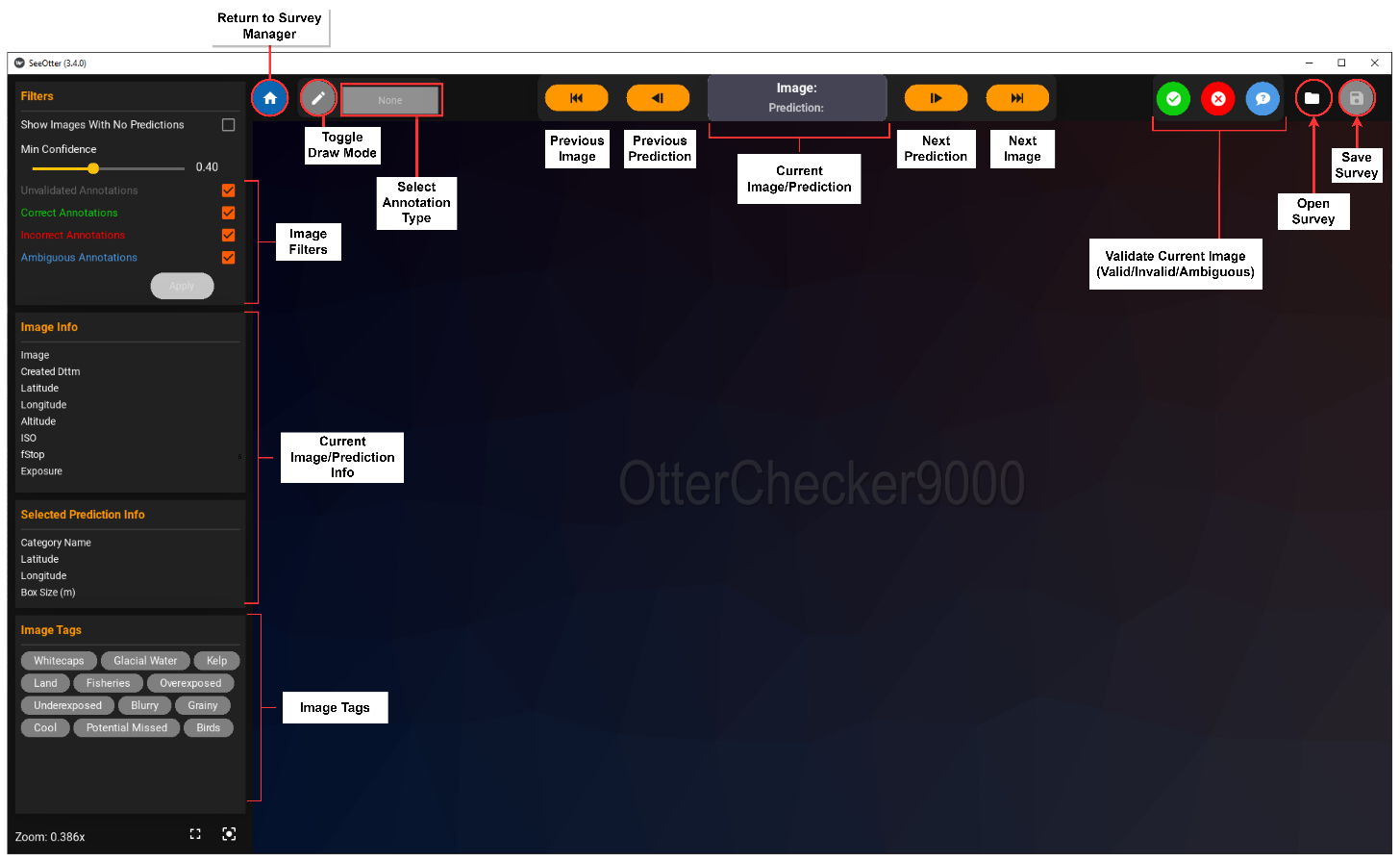
See Otter is divided into 3 screens:

* Survey Manager (Home Page)
* OtterChecker9000
* Settings

### Survey Manager Screen (Home Page)

### 

### Otter Checker 9000 Screen

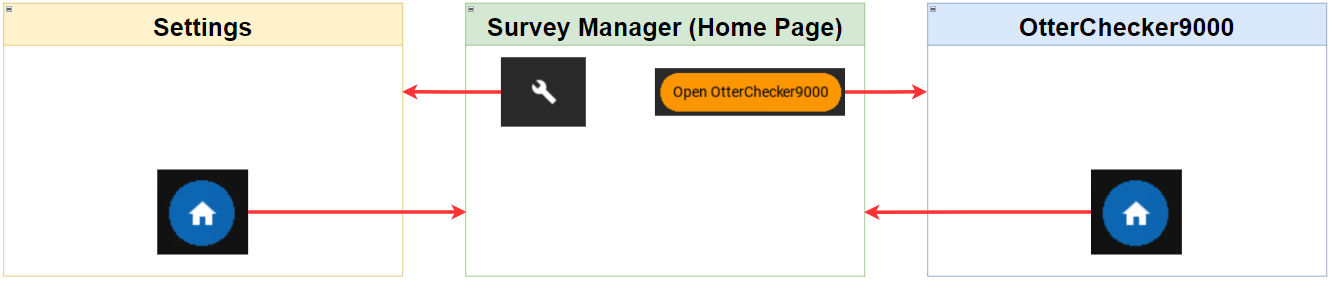


### Settings Screen

### Graphical user interface Description automatically generated with low confidence

### Page Navigation

Below are the buttons/icons used to navigate between the screens.



### Files and File Structure

#### Application Files

The SeeOtter folder (the folder that contains the main application) contains the main files to run the program. As well as a number of other user files that can be viewed/modified:

* SeeOtter.exe (Main Application)
* SeeOtterUserGuide.docx
* See\_otter\_config.json (processing/runtime settings)
* Otter\_checker\_config.json (visual/behaviour settings)

#### Project Files (Surveys)

A SeeOtter project (also referred to as a “survey”) contains a number of files and folders that are made upon creation of project.

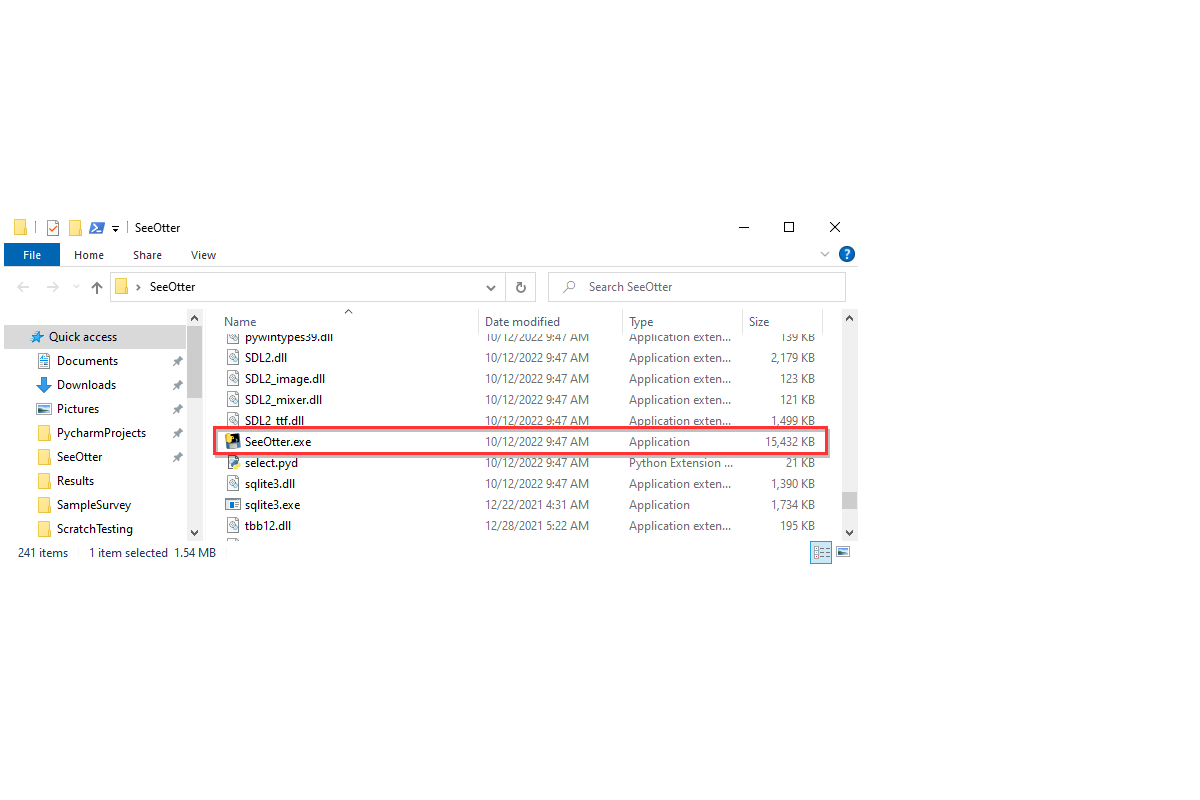
Graphical user interface, text, application

Description automatically generated

* savefile.json – Main project file containing the metadata about the project (file paths, predictions, validations, etc…)
* Camera\_system.json – Defines the current camera system
* Image\_path\_mapping.csv – If created from Waldo data, contains a mapping between the original and current image paths
* Annotations – Output folder for generated annotations
* Backup – Program backups
* Images – Images
* InclinometerData – Output files from inclinometer placed here will be read by SeeOtter when opening project
* Results – Output folder for results (csv files and maps)
* Transects – Kml transect files placed here will be read by SeeOtter when opening project
* WaldoFiles – Archive of original waldo files

# Setup

* Download SeeOtter and copy folder to computer
* Run “SeeOtter.exe” located in the SeeOtter folder



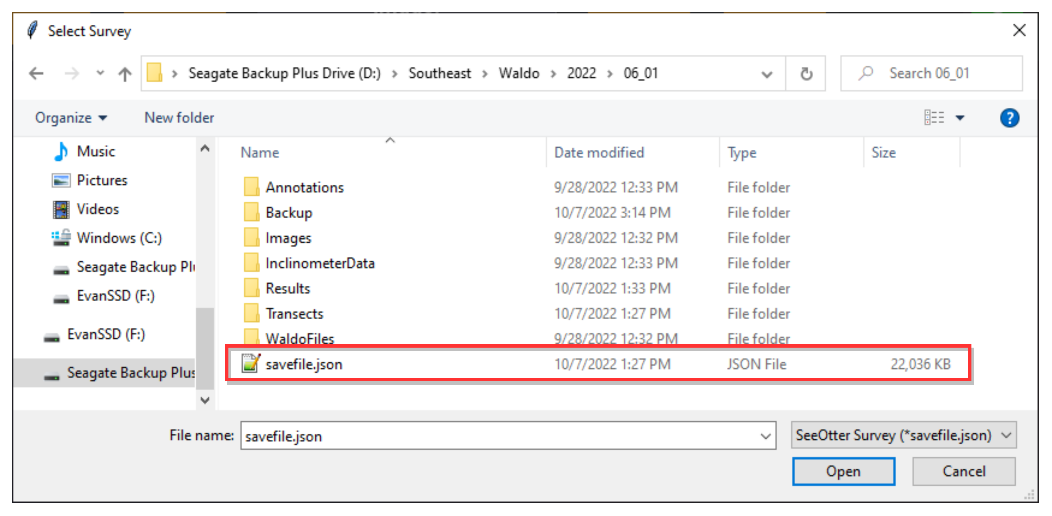
# How to Use

The typical workflow is as followed:

* [Create project](#_Create_New_Survey)
* [Run Processing](#_Run_Processing_(Generate)
* [Validate Images](#_Image_Validation_(OtterChecker9000))
* [Generate Results](#_Generate_Results)

### Open Survey

* Click the folder icon in the top right of the window
* Navigate to the folder your survey is located at
* Select “savefile.json”



### Save Survey

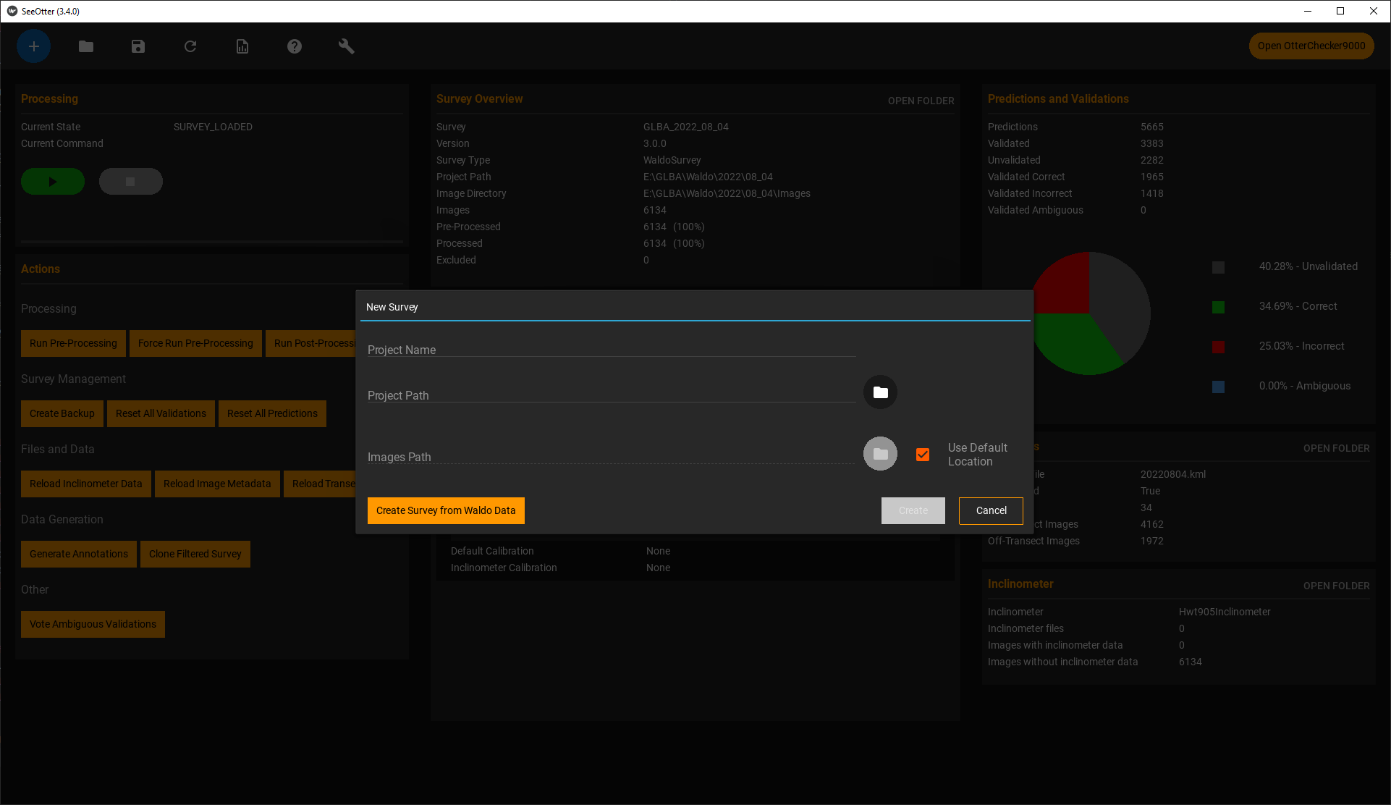
* Press save icon in top right to save project
* A light grey circle will appear around the icon when there are unsaved changes

### Create New Survey

A screenshot of a computer

Description automatically generated with medium confidenceCreate a new survey by clicking the blue plus button in the top left corner of the Survey Manager screen. You will be prompted with a “New Survey” window.

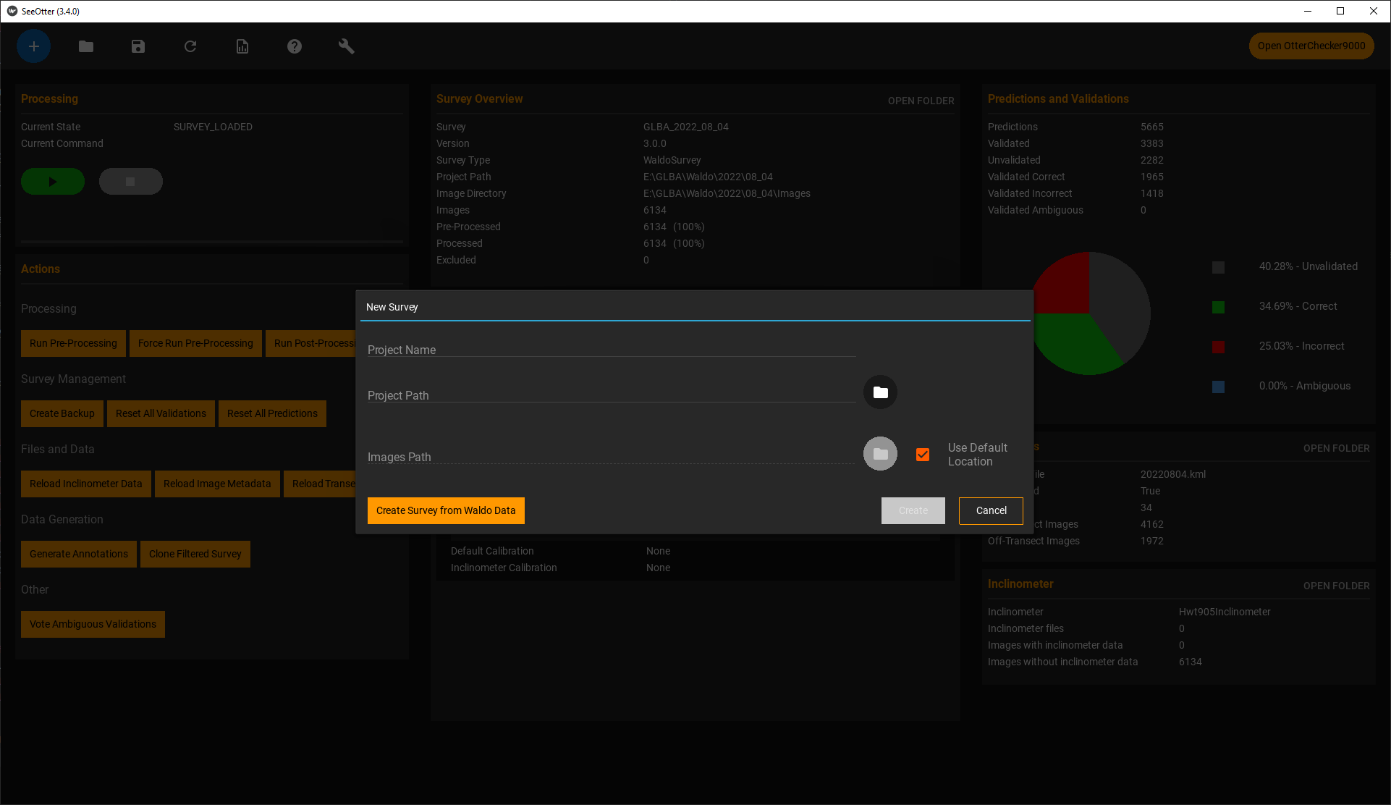
Fill in the required fields:

* Survey Name: Name of survey
* Project Path: Folder that survey will be created in
  + \*Must be an existing folder\*
* Images Path: Folder containing images
  + Defaults to [ProjectPath/Images]
  + Will be created if it doesn’t exist
  + Can be set to different location by unchecking “Use Default Location”
    - *Note: Using non-default location can cause file path issues and is not recommended.*

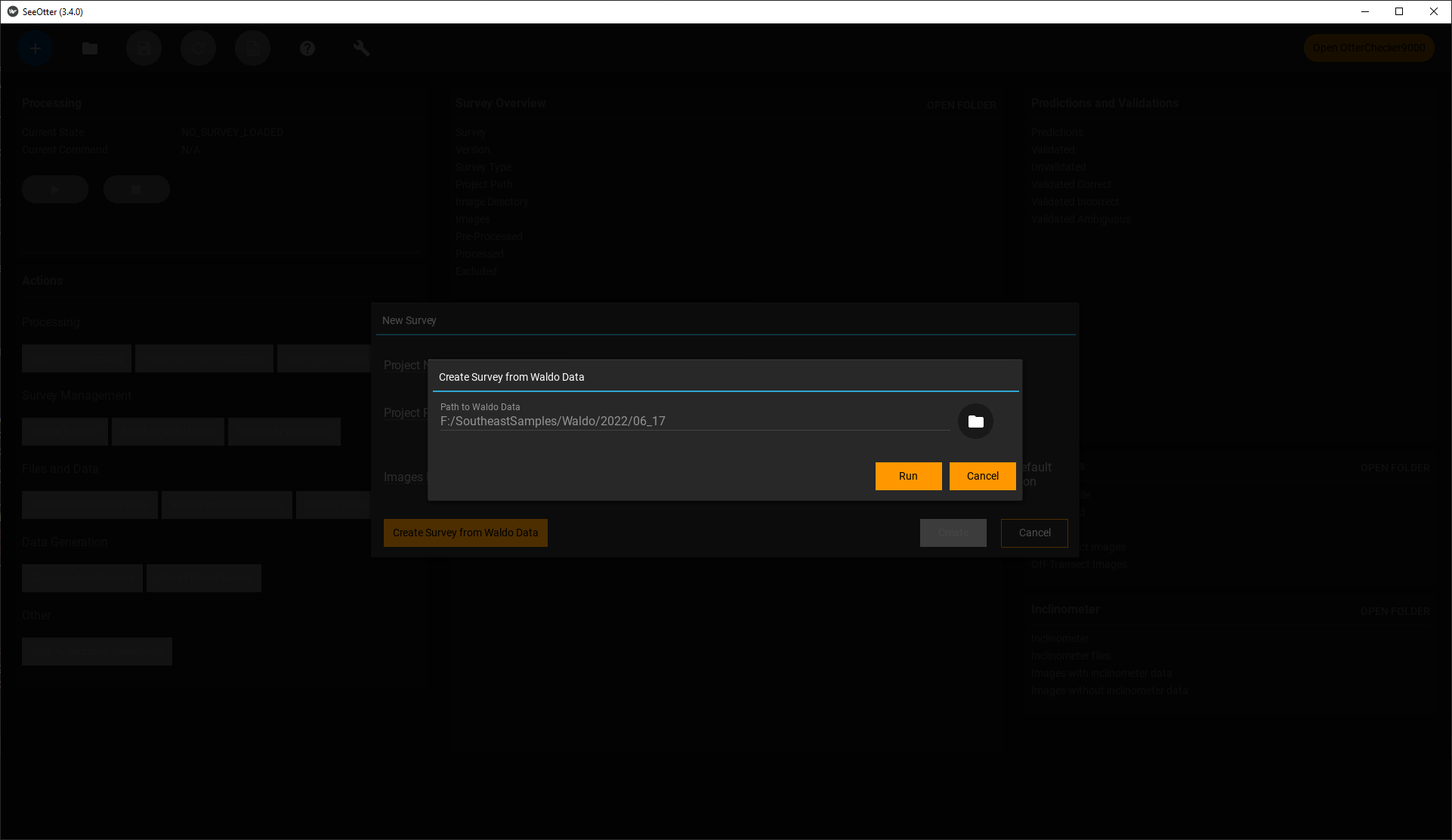
### Create Survey from Waldo Data

* Click the “New Survey” button as above, and select “Create Survey from Waldo Data”

A screenshot of a computer

Description automatically generated with medium confidence 

* Click the folder icon and select the folder containing your Waldo Data
  + Note: This will convert a single day of Waldo data and the file path must be formatted as [*DriveLetter:/Location/CameraSystem/YYYY/MM\_DD]*



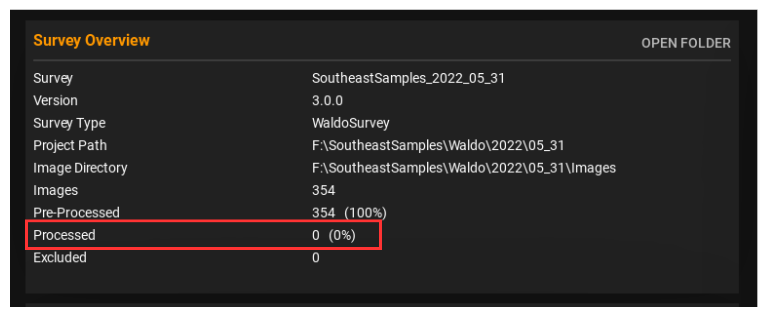
* Click “Run”
  + Note: If “Images” folder already exists, this can cause an error. As long as folder is empty, delete and try again.
* If this worked, you should have a new survey located in the given folder

### Run Processing (Generate Predictions)

A survey needs to be processed before it can be validated or data can be generated. Processing runs the following steps:

* Pre-Processing
* Processing (Generate predictions)
* Post-Processing

You can see if the current survey has been processed or not via the “Processed” field in the Survey Overview card.

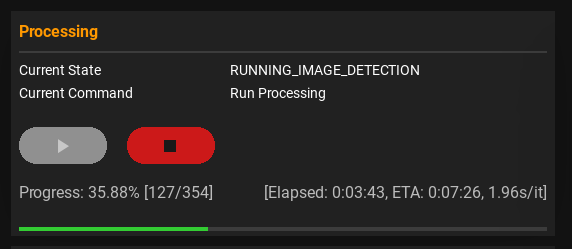


To start processing, press the green play button on the Processing card

A screenshot of a computer

Description automatically generated with medium confidence

This process will take a long time depending on how many images your survey contains. Once pre-processing completes, the “Current State” will switch to “RUNNING\_IMAGE\_DETECTION” and the progress will be displayed.



By default, the survey will be saved for every 100 images processed (this can be changed in settings). But you can also stop processing with the red stop button (this may take several seconds as it waits for the current image to finish processing) and save your project to resume later.

### Generate Results

* A screenshot of a computer

  Description automatically generated with medium confidenceClick the “Generate Results” button on the top bar
* An explorer window should open. Results are saved to the Results folder

### Actions (Commands)

Actions are located on the survey management screen. They are used to perform various commands on the current survey.

* **A screenshot of a computer

  Description automatically generated with medium confidenceRun Pre-Processing** – Steps to prepare images for image detection:
  + Calculate Bearing: Determines the direction that the plane was heading based off successive image location
  + Correct Image Orientation: If the “rotate\_image” setting for a given image’s config is set to True, the image will rotated by 180 degrees to match the actual orientation of that camera
* **Force Run Pre-Processing** – Runs pre-processing for all images, ignoring the “has\_been\_pre\_processed” flag
* **Run Post-Processing** – Run Post processing steps:
  + Calculate coordinates of predictions
  + Flags predictions located in the “temporal overlap” zone of another image
* **Create Backup** – Creates a backup of the current survey (not including images) at [SurveyPath/Backup]
  + There is currently no “restore” feature, but project can be restored by copying files from a given backup folder into the main project folder
* **Reset All Validations** – Resets all validations to “UNVALIDATED”
* **Reset All Predictions** – Removes all predictions, and their associated validation data
* **Reload Inclinometer Data** – Clears inclinometer data and attempts to reload from files located at [SurveyPath/Inclinomter]
* **Reload Image Metadata** – Reloads all metadata related to images (coordinates, altitude, datetime, etc…)
  + Note: this will not affect any other data such as predictions or validations, but post-processing will need to be re-run if the image’s GPS location has changed to recalculate predicted object coordinates.
* **Reload Transects** – Clears transect data and reloads data from transect files at [SurveyPath/Transects]
* **Generate Annotations** – Converts predictions to annotation format (default: YOLOv5) and outputs files at [SurveyPath/Annotations]
* [**Clone Filtered Survey**](#_Clone_Filtered_Survey) – Creates a clone of the survey, while filtering out images and predictions based off the selected validation types.
* [**Vote Ambiguous Validations**](#_Handling_Ambiguous_VAlidations) – Used to “vote” on ambiguous images given multiple people

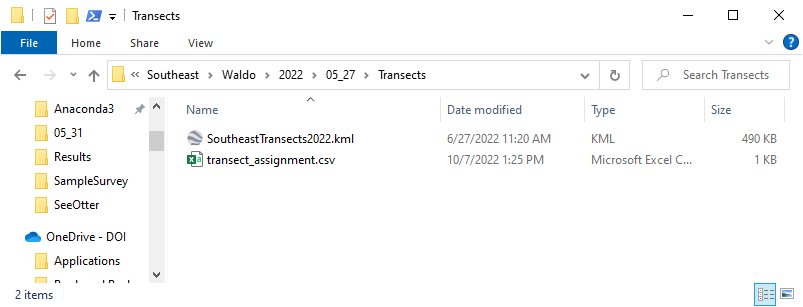
### Transects

SeeOtter can load transect files (.kml format) as well as manual transect assignments to correct for flight paths, missing transects, etc.

#### Kml Transect files

Kml transect files must follow certain formats. The must contain a field marked “trans\_id”, and coordinates for each transect line.

To load transects into SeeOtter, place your .kml file into the Transects folder located at [\*PATH\_TO\_SURVEY\*/Transects]



Then open the survey in SeeOtter. If this worked, you should see relevant transect info in the Transects card.

Text

Description automatically generated

*Note: For performance reasons, SeeOtter will only reload the .kml file if it has been modified or renamed. This can also be manually performed via the Actions card.*

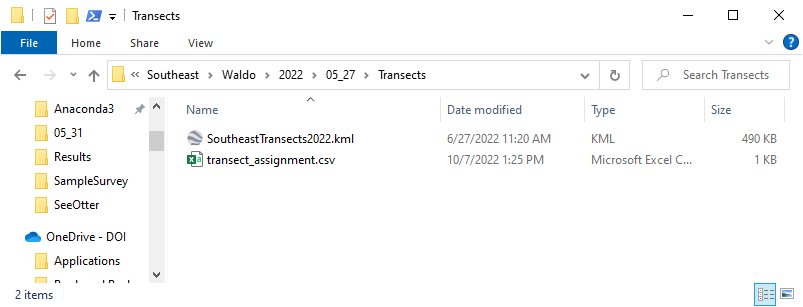
If this did not work, it is likely your .kml file does not conform to one of the valid formats. Check the console window for more details.

#### Manual transect assignment

Manual transect assignment happens after the transect file has been read and will override the transect\_id for any given images.

To manually assign transects, open the csv file at:

* [\*PATH\_TO\_SURVEY\*/Transects/transect\_assignment.csv]



And open the SurveyTransectMap in Google Earth. This file located in the results folder at:

* [\*PATH\_TO\_SURVEY\*/Results/SurveyTransectMap(YOUR\_SURVEY\_NAME).kml]
* *Note: You need to* [*generate results*](#_Generate_Results) *before this file will be created*

The transect assignment works by taking 2 images (start\_img, end\_img), and assigning a transect\_id to every image between them (inclusive).

On the SurveyTransectMap, find the first and last images you want to assign to a transect. Click on the pin to get the file names.

Text

Description automatically generated

Copy these file names into the “start\_img” and “end\_img” fields of the csv file respectively, and set the transect\_id (leaving the transect\_id field blank will mark it as “off-transect”)

*Note: start\_img has to be the one that comes first numerically or alphabetically (ie: “1\_000\_00\_016.jpg” would be start\_img and “1\_000\_00\_019.jpg” would be end\_img)*

Text, application, table

Description automatically generated

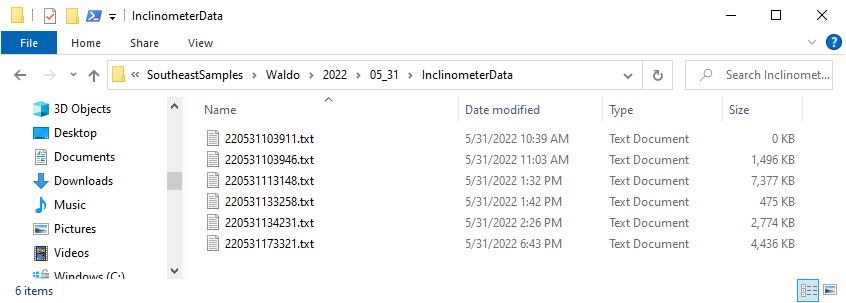
For performance reasons, if your camera system has multiple cameras (such as the Waldo camera system), only images from one of the cameras is displayed on the SurveyTransectMap. In that case, duplicate the row and rename the images to match each camera system(as seen above).

### Inclinometer Data

SeeOtter is able to accept inclinometer data to apply as a correction factor for plane orientation (only the HWT-905 inclinometer is currently supported)

To import inclinometer data, copy inclinometer data files to the folder at:

* [\*PATH\_TO\_SURVEY\*/InclinometerData]



Then open your survey in SeeOtter. The Inclinometer card should now display info about the loaded file(s).

Text

Description automatically generated with low confidence

*Note: Inclinometer data is assigned to images based off datetime. If an image has the wrong datetime, it will not be properly assigned with the correct inclinometer data record.*

### Camera System

The camera system contains the parameters for SeeOtter to model an actual camera system in order to georeference image predictions. This is meant to allow SeeOtter to be configurable with various different camera systems with varying number of camers. Info about the current camera system can be seen on the Camera System card.

A screenshot of a computer

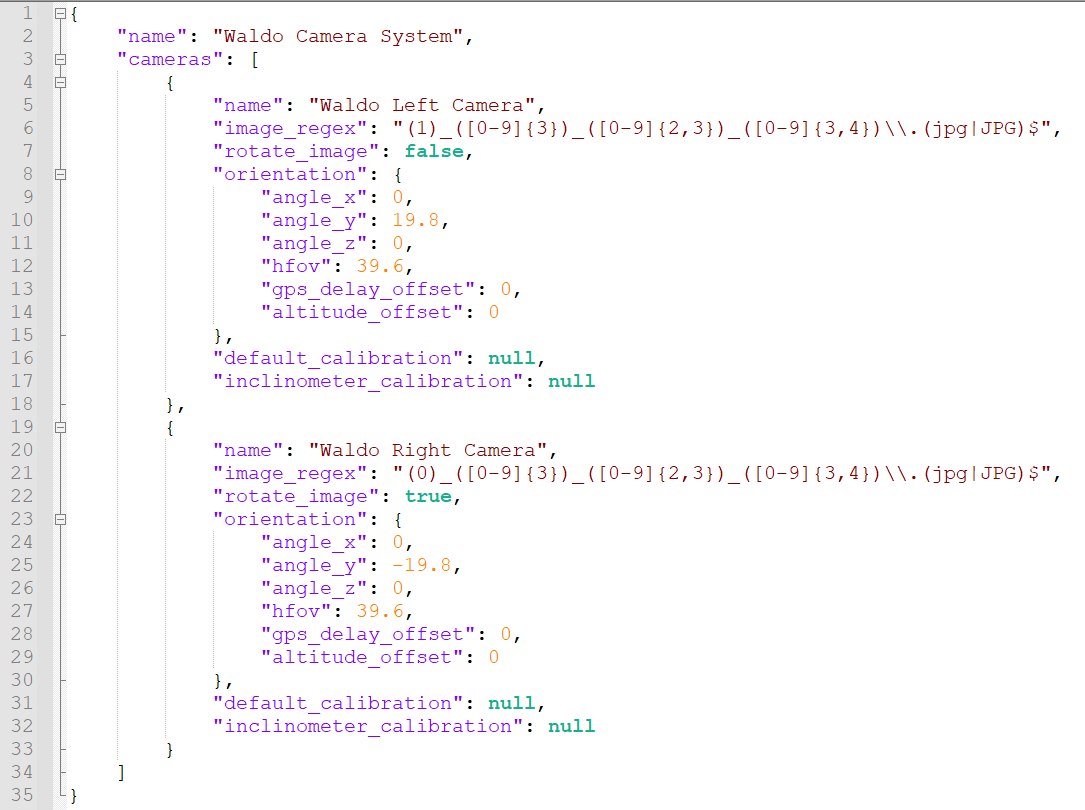
Description automatically generated with medium confidence

#### Camera System Parameters/schema

* Name: Name of camera system
* Cameras: List of cameras contained in camera system
  + Name: Name of camera
  + Image\_regex: Regex pattern used to identify which images belong to a given camera
  + Rotate\_image: Will image be rotated by 180 degrees during pre-processing
  + Orientation: Camera orientation
    - *Note: an x/y/z angle of 0,0,0 correlates to a camera looking straight down from the plane, where the top of the image is the direction of heading*
    - Angle\_x: Pitch
    - Angle\_y: Roll
    - Angle\_z: Yaw
    - Hfov: Horizontal field of view
    - Gps\_delay\_offset: Not used
    - Altitude\_offset: Not used
  + Default calibration: Calibration used to correct for camera orientation for images that do not contain inclinometer data
    - \*Follows same data structure as “Orientation”
  + Inclinometer calibration: Calibration used to correct for camera orientation for images that contain inclinometer data
    - \*Follows same data structure as “Orientation”

#### Editing Camera System

Open the camera system config [camera\_system.json] by pressing the “OPEN CAMERA CONFIG” button on the top right of the Camera System card.



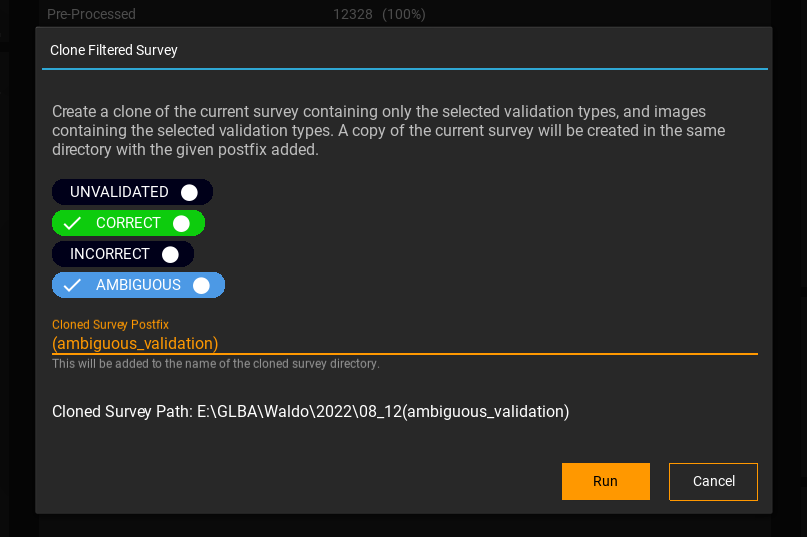
Refer to the above section for information about the definitions of each field.

### Clone Filtered Survey

Cloning a filtered survey allows you to create a copy of a survey, while filtering by validation type. Any validation types not selected, and any images without the selected validation type will be removed from the cloned survey. Making the file size of the project much smaller and easier to share.

This feature can be accessed from the “Clone filtered survey” button on the Actions card.

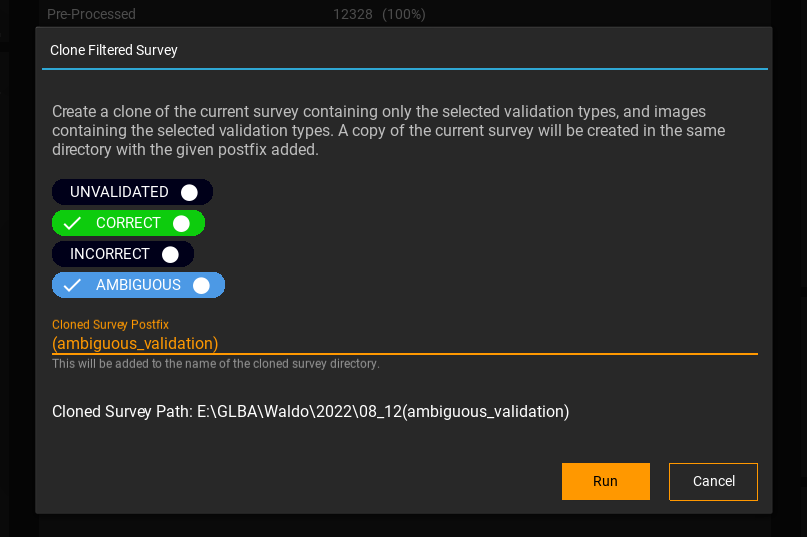




### Handling Ambiguous VAlidations

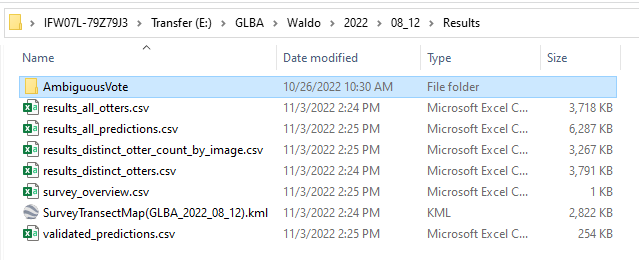
SeeOtter has the functionality to allow multiple observers to “vote” on ambiguous validations to act as a tiebreaker. To do this, you need an odd number of validators to run through the ambiguous validations. Each validator will need access to SeeOtter, and a copy of the survey.

For the sake of file size, it’s recommended to [create a filtered cloned survey](#_Clone_Filtered_Survey) containing only ambiguous validations (correct or incorrect are optional, but selecting them will add to the number of included images). This will be the project that is shared to your observers.



After each observer finishes validating the images (marking only correct or incorrect), they must generate results, then send the “validated\_predictions.csv” file back to the original project owner.

Take these csv files and place them in the folder at [\*PATH\_TO\_SURVEY\*/Results/AmbiguousVote]



*Note: There must be an odd number of observers/files*

Now, run the action “Vote Ambiguous Validations”



This will go through the results placed in the “AmbiguousVote” folder for each of the observers. Then each ambiguous validation in the survey will get replaced by “Correct” or “Incorrect” depending on the majority vote.

*Note: Doing this will alter the original survey, but a backup will automatically be created*

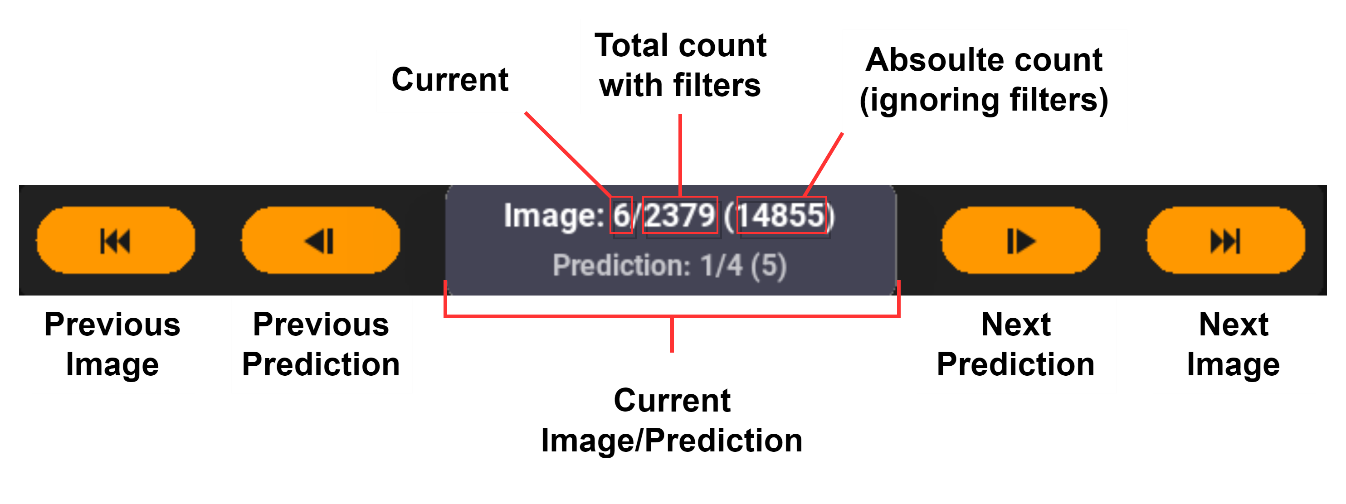
## Image Validation (OtterChecker9000)

A screenshot of a computer

Description automatically generated with medium confidenceImage validation is done in OtterChecker9000. Navigate to that screen using the button in the top right of the survey manager screen.

### Image/Prediction Navigation

* Use the arrows at the top to navigate between images and predictions
* Alternately, you can use the arrow keys
  + Up/Down -> Next/Previous Image
  + Left/Right -> Next/Previous Prediction



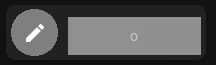
* The currently selected prediction is marked by a yellow tab above the annotation box
* Clicking an annotation will also select it as the current prediction

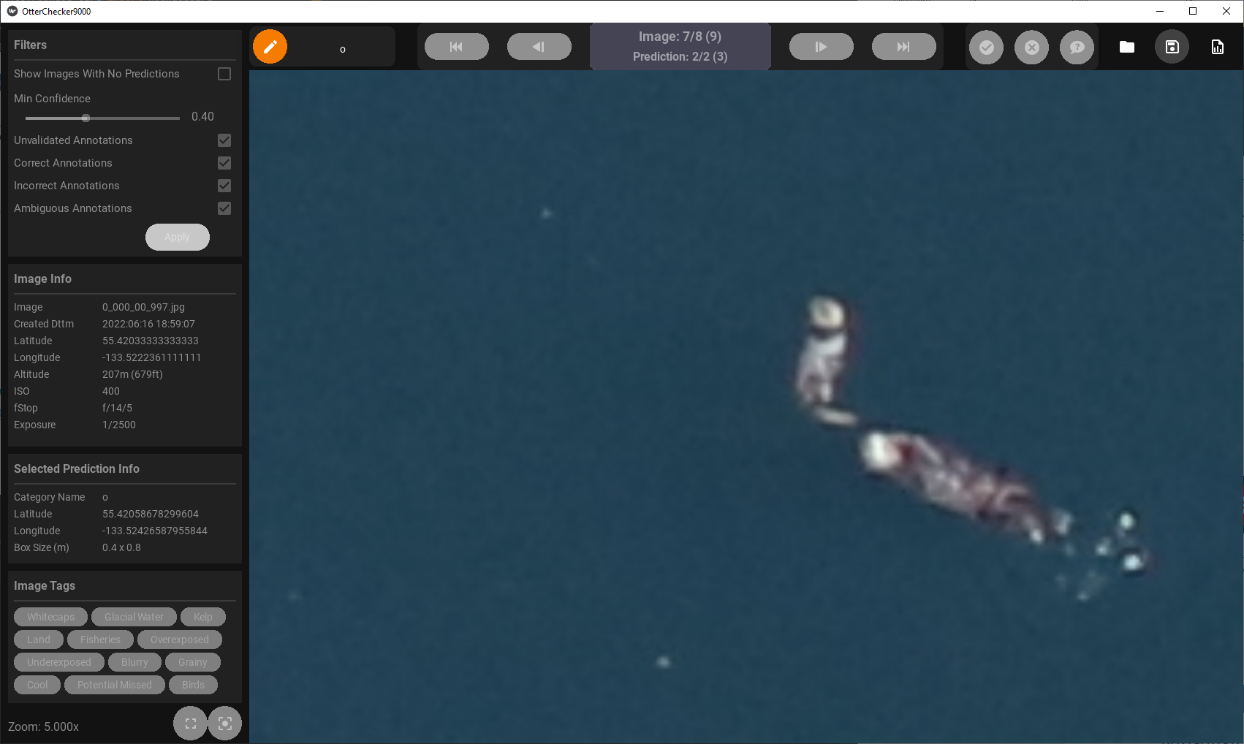


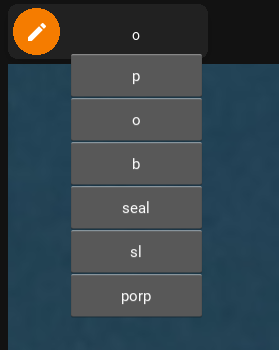
### Validation

* Use the red/green/blue buttons at the top right (or Q/W/E keys) to validate the currently selected prediction
  + Correct (green) – The marked prediction is correct
  + Incorrect (red) – The marked prediction is incorrect
  + Ambiguous (blue) – It is unclear whether the marked prediction is correct or not

### Drawing Annotations

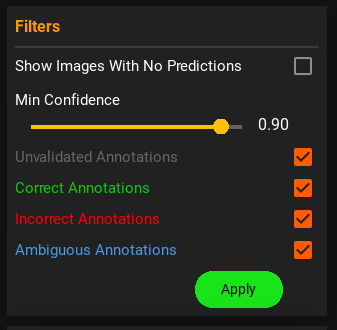
* Click the pencil icon in the top left or press the ‘D’ key to toggle annotation draw mode
* You can tell draw mode is active because the pencil icon will turn orange, and many elements will be greyed out/disabled



* Open the annotation categories by clicking the dropdown box directly to the right of the pencil icon and selecting a category
* The default categories are:
  + P = Pup
  + O = Otter
  + B = Bird
  + Sl = Sea Lion
  + Porp = Porpoise
* Click and drag to draw new annotation



### Filters

* Image filters show/hide predictions depending on settings
  + Show Images With No Predictions: If selected, shows all images regardless of whether they contain predictions
  + Min Confidence: Hides predictions with lower confidence than set by the slider
  + Unvalidated/Correct/Incorrect/Ambiguous checkboxes: Toggle visibility of predictions with the corresponding validation state
* After changing filters, press “Apply” to apply changes

### A picture containing icon Description automatically generatedGridlines

* Gridlines can be toggled on/off via the grid icon in the bottom left corner of OtterChecker
* Number of rows/columns can be edited in settings

A picture containing shape

Description automatically generated

## Settings/Config

There are 2 config files:

* Otter\_checker\_config.json
* See\_otter\_config.json

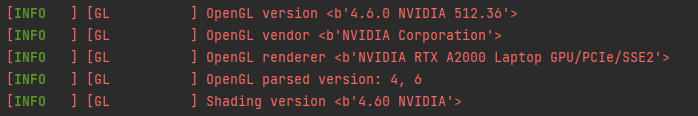
These files are located in the application folder and contain settings related to the programs behavior and appearance. They can be edited from the settings page, or directly via a text editor.

*Note: Putting invalid values into the config may result in crashes, or failure for application to start. If this happens, you can delete the config files, and new ones with default values will be created the next time SeeOtter starts.*

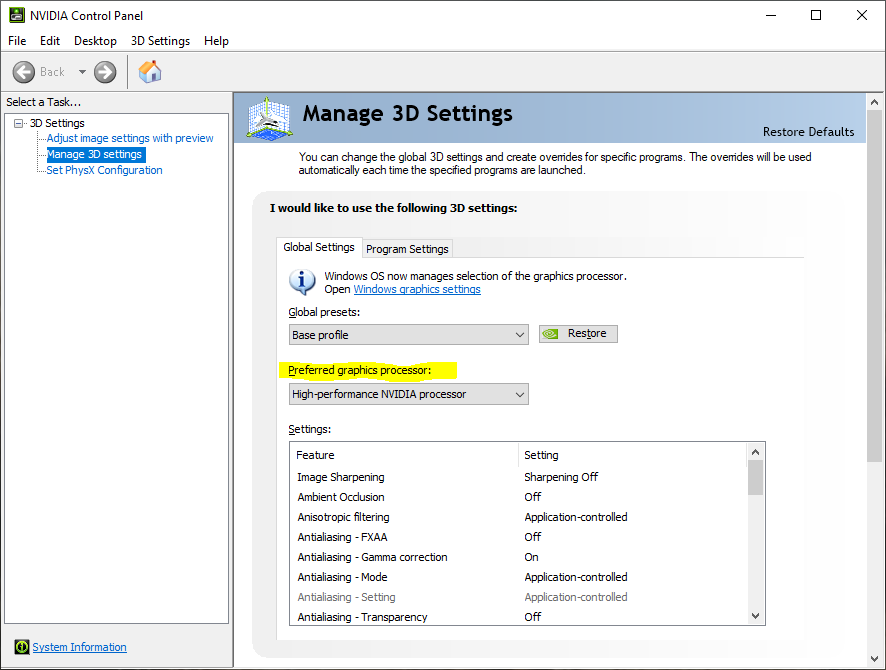
# Other

### GPU Acceleration

OtterChecker uses Kivy as a GUI frontend and can utilize a discrete graphics card to improve performance/responsiveness. You can see what device is being used on the console during startup.

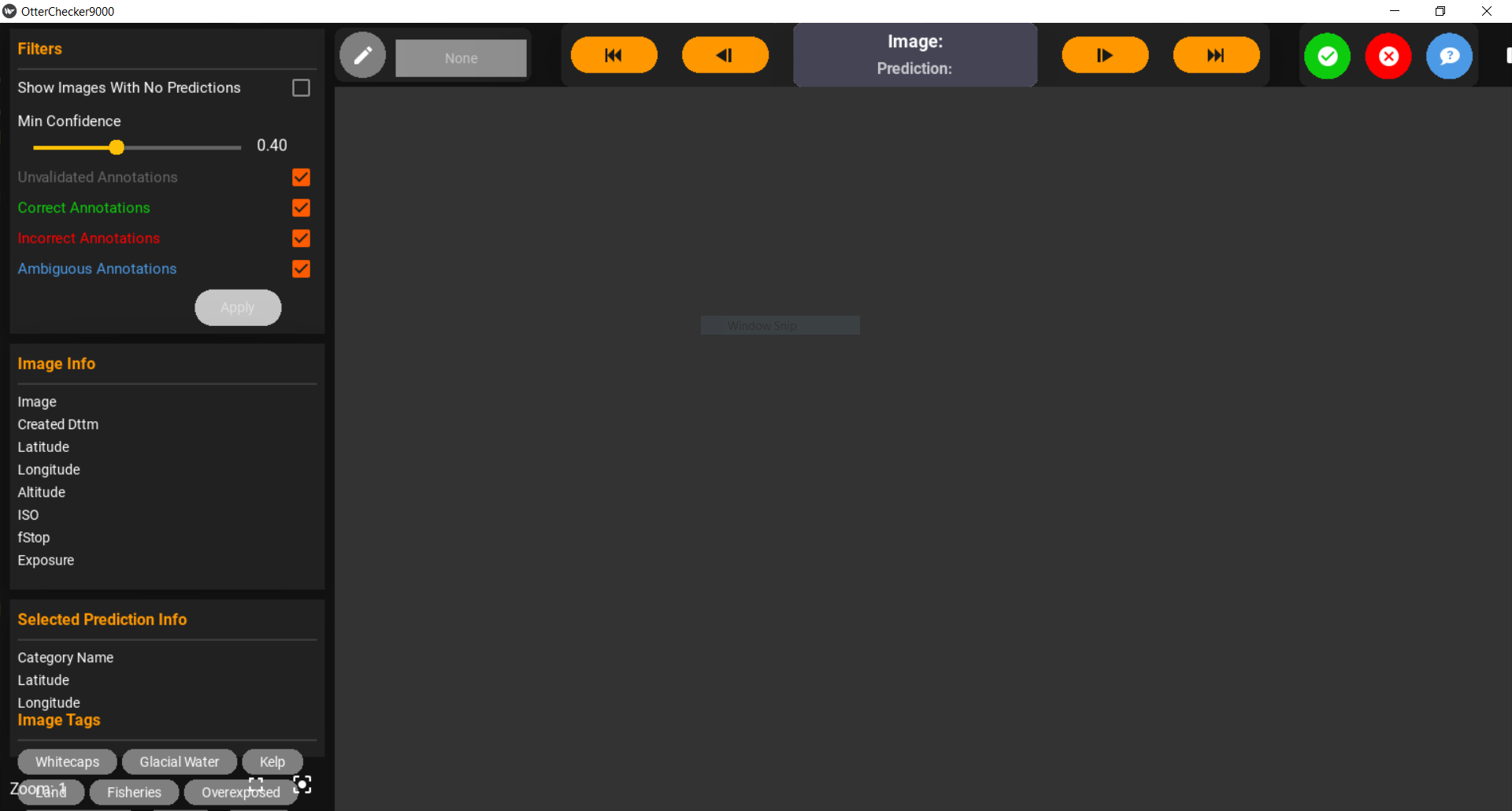


If your discrete GPU is not being listed here, you may be able to enable it in settings (mine previously defaulted to Intel integrated graphics from the CPU).

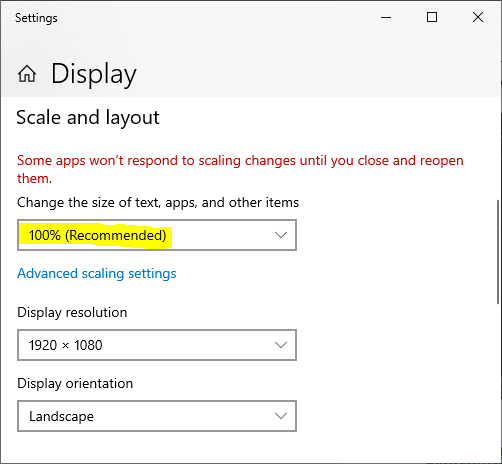


### Scaling Issues

Some users may have scaling issues where the right part of the screen is cut off as seen below.



This can be resolved by lowering your display scale (Right click desktop -> Display Settings)



# Controls

#### Navigation

|  |  |
| --- | --- |
| Next Image | Up Arrow |
| Previous Image | Down Arrow |
| Next Prediction | Right Arrow |
| Previous Prediction | Left Arrow |
| Pan to Selected Prediction | Space |
| Hide Annotations | Shift (Hold) |

#### Annotations

|  |  |
| --- | --- |
| Toggle Annotation Draw Mode | D |

#### Validation

|  |  |
| --- | --- |
| Mark Selected Prediction Correct | Q |
| Mark Selected Prediction Incorrect | W |
| Mark Selected Prediction Ambiguous | E |