# **Image Tagging**

# Sloth

## Setup

In order to tag images, we use a piece of software called Sloth. To install it from the robosub debian repository, run the following command:

```
sudo aptitude install sloth
```

Since we use Darknet, there are special configurations needed. After having cloned the vision\_dev repository, add the following line to your .bashrc, replacing the dummy path with the path to the vision dev/sloth directory so that sloth can create darknet files.

export PYTHONPATH=/path/to/vision\_dev/sloth:\$PYTHONPATH

### **Using Sloth**

To use sloth and start tagging images, you can run the following command.

```
sloth -c /path/to/vision_dev/sloth/robosub_config.py
/path/to/annotation/file
```

\*-c\* Is a flag to give the path to a configuration file. This file is provided in the vision\_dev/sloth directory. The path you provide should point to this file.

The last argument is a path to an annotation file. This is most likely named something like `labels.json`. You will need to provide this path in order to tag images.

Should you find this tedious, there is a script in the sloth directory of the vision\_dev repository which performs some of this for you. It can be used as follows:

```
./robosub sloth.sh /path/to/annotation/file
```

This removes the need to add the -c flag repeatedly. This script can also be symlinked to without issues.

## Using sloth output with Darknet

When attempting to use tagged images with darknet, the annotation file will need to be in the correct format. To convert to this format, use the following command

```
sloth convert <original file> <new_filename>.darknet
```

This will ask sloth to convert an annotation file in one format to the correct \*.darknet format.

Once the file is in this format, you will need to run the `sloth\_to\_darknet.py` script which will generate the multiple files darknet requires based upon the compact information in the \*.darknet file.

The following is an example command to generate said files.

```
./sloth_to_darknet.py -f /path/to/label/file -o /path/to/annotation/dir/
(optional)[-t training_list_filename.txt]
```

## **Sloth Keybindings**

The full list of keybindings used in sloth can be found in the robosub\_config.py file near the bottom. A shorthand list is provided below

# **Default keybindings**

Space Mark image as labeled/confirmed and go to next

Backspace
 PgDown
 PgUp
 Tab
 Shift+Tab
 Next image/frame
 Previous image/frame
 Select next annotation
 Select previous annotation

• Ctrl+f Fit current image/frame into window

• **Del** Delete selected annotations

• Esc Exit insert mode

• Shift+I Mark current image as labeled

• **Shift+c** Mark all annotations in image as confirmed

#### **Robosub Specific**

• **F5** Toggle the visibility of label names on annotation boxes

Ctrl+Shift+D<sub>Delete</sub> all annotations from the current image and mark it as unlabeled

**el** Copy all annotations from the previous image to this one

C Mark image as labeled/confirmed and go to next then copy all annotations

• **n** from the previous image.

(Equivalent to **Space** then **c**)

# **Annotations**

• r • y Red Buoy • Yellow Buoy

• g

* 5	Green buoy
• p	Start Gate Post
• m	Path Marker
• ,	Marker Drop Platform
• .	Object Pickup Platform
• h	Object Drop-off Platform
• t	Hexagon
• z	Torpedo Target
• x	Navigation Channel Post
	<b>Navigation Channel Bar</b>

Green Ruov

#### **Sloth Mouse Controls**

. .

While creating annotations, the following are useful mouse controls.

## Right Click and Drag

Resize an annotation. Resizing is based upon the quadrant of the annotation clicked on.

## • Ctrl+Left Click

Select multiple annotations at once.

# **Getting Data to Label**

We use the download\_data.sh and download\_data\_current.sh scripts from the vision\_dev repository to get data from the server. Make sure that you always do a git pull before getting new data because the scripts change frequently.

First claim one or more batches to label from the current spreadsheet of data. The current spreadsheet is located here. Note which numbers you want.

Once you have claimed some batches run the following command in vision\_dev to download and extract the data:

```
./download_data.sh bag_prefix_0 ## ##
```

For example if you want batch numbers 00 04 and 12 of the path\_marker batch you would run:

```
./download_data.sh path_marker_0 00 04 12
```

Before downloading the data it will ask you for a username and password, please ask in either the it or general channels on slack

## **Uploading Data**

Here is the upload link for .json files. Just drag and drop it.

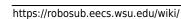
From:

https://robosub.eecs.wsu.edu/wiki/ - Palouse RoboSub Technical Documentation

Permanent link:



Last update: 2017/10/31 10:10



×