SUVI Trainer User Guide

Version / Date	Revision
1.0 / Aug. 2, 2018	Initial draft
1.1 / Aug. 16, 2018	Added description of sliders
1.2 / Aug. 22, 2018	Added description of configuration file

SUMMARY

The SUVI trainer annotation tool allows users to label solar features that are used to generate training datasets. These datasets are inputs into various machine learning algorithms developed by the STP group within NCEI.

INSTALLATION

The `suvitrainer` package requires Python 3 and its reference page can be found at:

https://pypi.org/project/suvitrainer/

Running `pip install suvitrainer` will install the program. Additional instructions and comments are documented on the reference page.

SETUP

1. Download the example configuration file at: https://github.com/jmbhughes/suvi-trainer/blob/master/config_example.json
2. Rename the configuration file to `config.json` and store it in your home directory.
3. Under the "`train`" category, edit the following in `config.json`:
    a. "`name`": name (string) to associate with the trainings
    b. "`upload`": boolean indicating whether to upload the training to server
    c. "`upload_password`": password (string) for server (contact hughes.jmb@gmail.com for the password)

    Note that any of the default settings in `config.json` may be edited to the user's preferences, but the above parameters are required to be modified by each user.
4. Ensure the following is in your user `.bashrc` file (only for `goesr-dev` users):

```
# NOTE: we have both Python2 and Python3 installed. We are still
# on Python2 by default, so add to path so that py2 is default.
# (ie reading path left to right, py2 install found first)
# added by Anaconda3 installer
export PATH="/nfs/stp_goesr/users/pyfunc/anaconda3/bin:$PATH"
# added by Anaconda2 installer
export PATH="/nfs/stp_goesr/users/pyfunc/anaconda2/bin:$PATH"
```

CONFIGURATION

The user can customize many different aspects of the trainer through the configuration file (config.json) they pass with the --config flag. This JSON file currently must include all the specified parameters (with the exception of the median_kernel settings in which case no median filtering will be assumed). An example configuration setup with notes is:

Configuration file example	**Explanation**

{
"Train": {
"name":"marcushughes",
"upload":true,
"upload_password" : "PASSWORD",
	the name of the expert labeling
	If true will upload to Marcus's ftp storage
	To upload a password is required, ask hughes.jmb@gmail.com for it.

"suvi_url":"https://data.ngdc.noaa.gov/platforms/solar-space-observing-satellites/goes/goes16/l1b/",	path to the website that serves the SUVI images and currently is used if l1b products are requested
"suvi_path":"/home/marcus/grive/suvi_training/images/",	local file path to the SUVI files and currently is used if composite products are requested. This can be changed to a local partial or complete copy of the file structure seen in the /nfs/spades_suvi_prod/archive/GOES-16/
"products": ["suvi-l1b-fe094","suvi-l1b-fe131", "suvi-l1b-fe171", "suvi-l1b-fe195", "suvi-l1b-fe284", "suvi-l1b-he304", "halpha"] },	which products to load and display in the GUI ● suvi-l1b-SSXXX where SS is the element and XXX is the wavelength in angstroms of l1b channel, e.g. suvi-l1b-fe131 ● suvi-l2-ciXXX where XXX is the wavelength in angstroms of the composite channel, e.g. suvi-l2-ci131 ● halpa will fetch GONG halpha images ● aia-XXX where XXX is the wavelength of AIA requested in angstroms.
"classes":{ "unlabeled":0, "outer_space":1, "bright_region":3, "filament":4,	The solar themes to label and what number to save them as in the FITS image

"prominence":5, "coronal_hole":6, "quiet_sun":7, "limb":8, "flare":9 },	
"data": { "median_kernel":0 },	If 0, no median filtering is applied, otherwise the number specifies how many pixels wide the window is

"display":{
"colors":{
"unlabeled":"white",
"outer_space": "#191970",
"bright_region": "yellow",
"filament": "#FF3E96",
"prominence": "#FF69B4",
"coronal_hole": "#0085b2",
"quiet_sun": "#006400",
"limb": "#8FBC8F",
"flare": "#ffdb99"
},
	Every theme defined in *classes* must be specified with a color to display it

"default":{ "red": "suvi-l1b-fe171", "green": "suvi-l1b-fe195", "blue": "suvi-l1b-fe284", "single":"halpha", "red_power": 0.26, "green_power": 0.29, "blue_power": 0.26, "single_power": 2.0, "header":"suvi-l1b-fe195", "lasso_color": "red", "lasso_width": 2 },	These are default preferences of the user Red chooses which product will be used as the red portion of the three-color image, similar for blue and green. Single indicates what the default single color image channel is. The power option sets the default power for that color. Header is the reference header used when outputting the thematic map FITS file. Lassow color and width specify what color the selection tool is and what point size it is.
"ranges":{ "single_color_power_min":0.25, "single_color_power_max":2.5, "single_color_power_resolution":0.01, "multi_color_power_min": 0.01,	Power_min and power_max set the limits of the power sliders. Resolution sets how much “power” is changed by when moving the slider one step.

"multi_color_power_max":1.0,
 	
"multi_color_power_resolution":0.01,
 	
"single_color_vmin":15,
"single_color_vmax":85,
"single_color_vresolution":0.1,
 	
"multi_color_vmin":15,
"multi_color_vmax":85,
"multi_color_vresolution":0.1
},
 	
"font": {
"bold": "Helvetica 18 bold"
}
}
}
	The lower percentile slider is has range [0, vmin] and the upper percentile slider has range [vmax, 100] with steps set by vresolution.
	
	Font is actually not used anywhere.

USAGE

Type `run_suvitrainer.py` at the command line and the following GUI will appear:

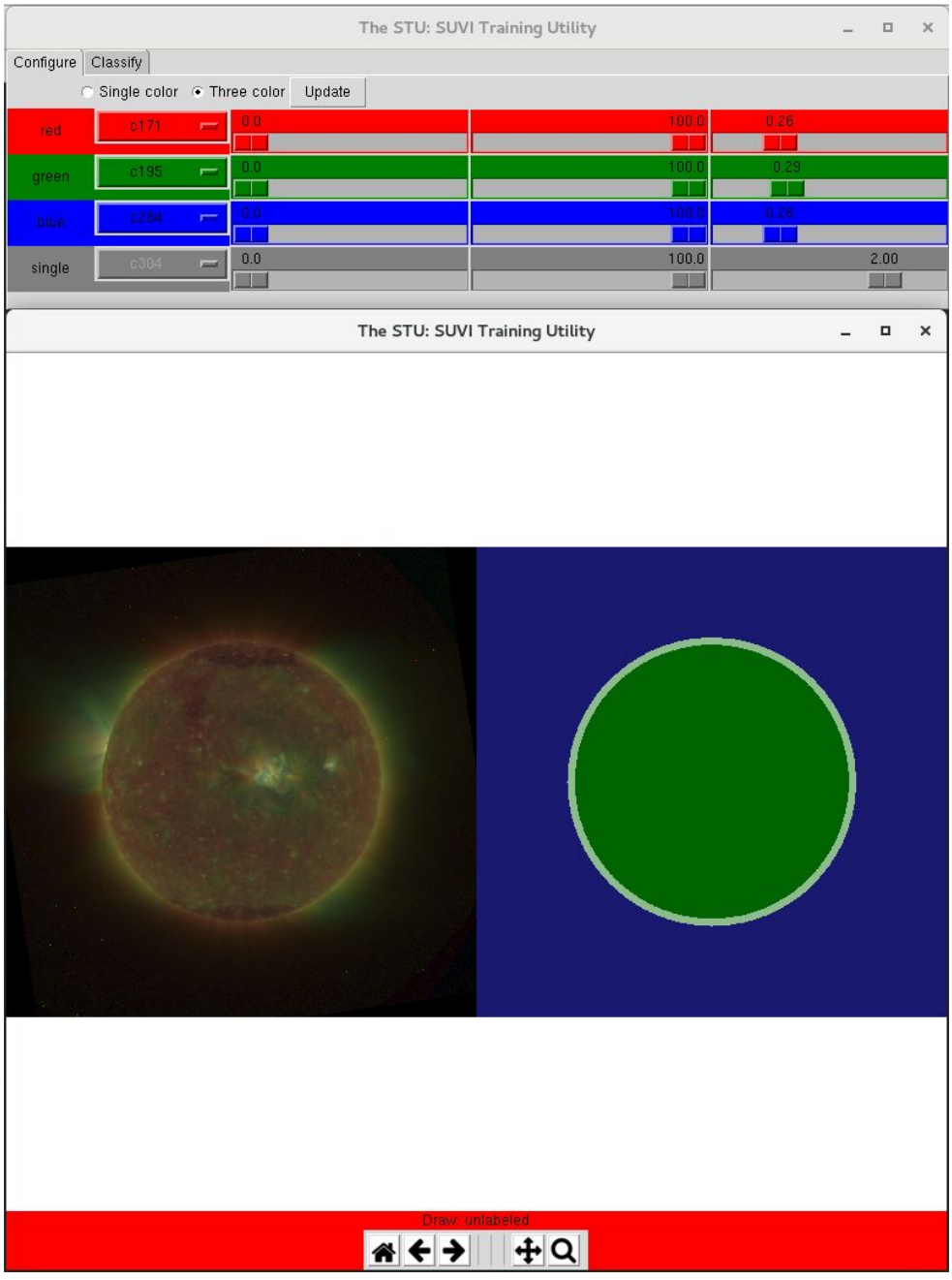

Figure 1. `suvitrainer` GUI annotation tool (the left image is the SUVI image to classify and the right image is the corresponding user labelled map).

The program can also be called with the following options:

- `--config`: Pathname to a specific configuration file (instead of the one in the home directory)
- `--date`: Specific date to annotate (most date formats are accepted but an example date format is `2018-Jun-11T12:50:06`)

`Configure` Tab

The colorbar ranges of the SUVI image (left image in Fig. 1) can be adjusted using the slide scales and toggling between a single color or three color image. The channels for the colors can also be selected. Make sure to click the `Update` button after each adjustment to reflect those options onto the shown image.

The leftmost slide scale is the lower percentile displayed; the middle is the rightmost percentile displayed; and the rightmost is the power the data is raised to. For example, if the user is looking at a single channel at 195 angstroms with the sliders set left to right as 5, 98, and 0.25 the user would be viewing the image raised to the 0.25 power (this makes it easier to see the full dynamic range) with only values between 5-98 percent of the full data range shown. Any pixel with intensity less than 5% will show with intensity of 5% and any pixel with intensity greater than 98% will show with intensity of 98%. This allows the user to clip off large values that make it hard to see the image, e.g. in very bright flares when viewed in composites.

`Classify` Tab

To annotate the solar image, select a label to classify and draw on the SUVI image with the mouse. The resulting annotation will be shown on the adjacent map (right image in Fig. 1). An example annotation is shown in Fig. 2.

The user can use the window buttons at the bottom of the training utility to pan and zoom in on the images to focus on regions of the Sun. These buttons can be toggled on and off.

The labeled regions can be overplotted onto the solar image by clicking on a labeled region. This can help the user assess how well he/she drew the region.

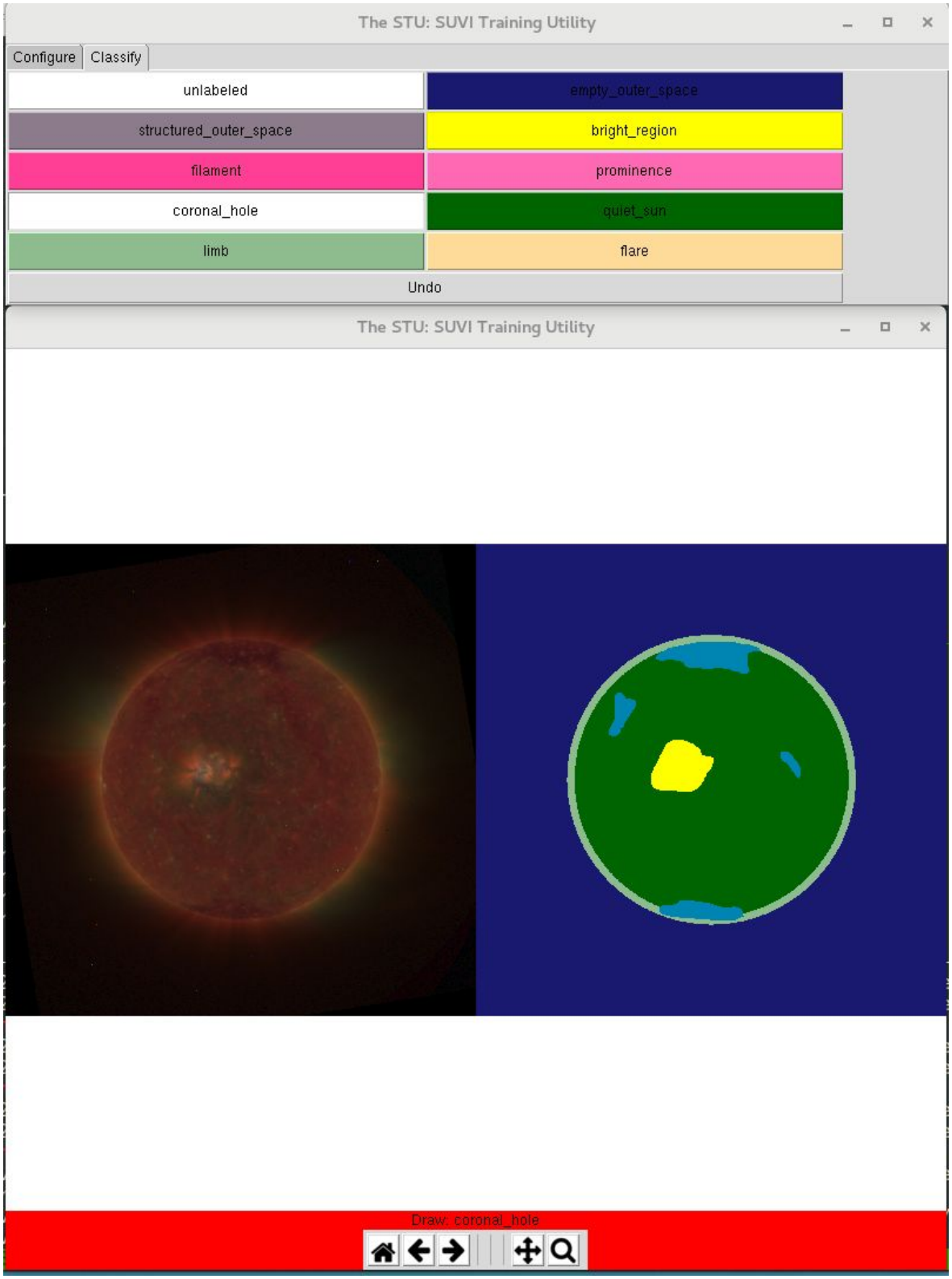

Figure 2. Example annotation using the `suvitrainer` GUI.

Keyboard Shortcuts

Table 1 summarizes keyboard shortcuts that are available for the utility.

Table 1. Keyboard shortcuts for `suvitrainer`

Shortcut	Action
u	Undo last annotation
c	Clear all contours on solar image

Saving Trainings

Upon exiting the utility, the program will ask the user whether to save the training or not. If the user decides to save the training, then the annotations (saved as a .fits file) are either saved to the user's current working directory or uploaded to a server, depending on the user's specification for the `upload` option in the configuration file.

`goesr-dev` USERS

The `suvitrainer` utility has been installed on goesr-dev for any user to use. However, the user will still have to follow the Setup instructions in this manual, as each user will have their own unique configuration file. Use the following steps to start and use suvitrainer on goesr-dev:

1. Type `ssh -X goesr-dev` in a terminal window.
2. Follow instructions in Setup section.
3. Type `run_suvitrainer.py`