Solar Jet Hunters Video Tool v1.0.0

Kekoa Lasko (lasko062@umn.edu)

May 2023

1 Usage

The video tool runs through the python script `make_movie.py`, located in the Solar Zooniverse Processor repository.

While the script contains its own documentation and provides a `print_help()` method that runs when the input arguments cannot be accepted, instructions for the use of the video tool are included here as well.

1.1 Running the Script

Once you have installed the Solar Zooniverse Processor, the `make_movie.py` script will be included in the `SolarJets/` directory.

To run the script, first navigate to the `SolarJets/` directory

```
$ cd /path/to/SolarJets/
```

and run the script using

```
$ python make_movie.py -i <input> <other parameters>
```

1.2 Input Parameters

Required Parameters

-i <input>

The only required input for the video tool. `<input>` must be a string containing a year, month, or day formatted as one of `YYYY-MM-DD`, `YYYY-MM`, or `YYYY`.

Depending on the input given, the video tool will work with all data it can find from the start to the end of the year, month, or day provided.

Modifiable Parameters

All of the modifiable parameters have default values that the Solar Jet Hunters team use to create the video subjects uploaded on Zooniverse. If these default values should ever need to be changed, this can be done using the parameters below when running the script.

-a <frames per subject>

Modifies the number of frames (or images) included in each subject. Must be an integer.

Default Value: 50 frames per subject

-b <overlapping frames per subject>

Modifies the number of frames at the end of subject i that will be included in subject $i+1$. Note that the final subject in a HEK event may have a greater number of overlapping frames due to the length of the event. Must be an integer.

Default Value: 5 frames of overlap

-c <cadence>

Modifies the interval of time between individual fits and image files. Greater cadence will result is longer intervals between images. Cadence is measured in seconds and must be an integer multiple of 12.

Default Value: 24 seconds

-d <dpi>

Modifies the dpi (quality) of the images produced. Higher dpi gives better quality. Must be an integer.

Default Value: 150 dpi

-f <frames per second>

Modifies the number of frames per second in the videos produced. Must be an integer.

Default Value: 10 frames per second

-s <max video size>

Modifies the max size of the videos produced in bytes. Videos will take up to this amount of space. Must be an integer.

Default Value: 1024000 Bytes = 1000 Kilobytes

Toggled Parameters

None of the toggled parameters have a following argument. They are all boolean parameters that are toggled on or off by calling them when running the script.

-m

Toggle this parameter to skip creating fits and image files.

-n

Toggle this parameter to disable creating the metadata associated with any subjects that are made.

-o

Toggle this parameter to allow the script to overwrite any files it tries to produce that already exist.

-v

Toggle this parameter to skip creating fits files.

2 How it Works

By default, the `make_movie.py` script runs through four main steps: creating the fits (`.fts`) files, creating the image files, creating the video subjects, and creating the subjects' metadata file.

2.1 Making Fits Files

To create the fits files, the script uses the Solar Zooniverse Processor. It references the HEK database to look for activity flagged as jets within the time provided as an input. For each event, cutouts are retrieved at the given cadence (default: 24s), and saved as a .fts file on the local machine.

The default location for the fits files is `files/fits/`.

2.2 Making Image Files

To create the image files, there must already exist fits files from within the given input time interval inside of the `files/fits/` directory. Each of these fits cutouts are used to generate an image using the Solar Zooniverse Processor and the provided dpi (default: 150).

The default location for the image files is `files/generated/pngs/`.

2.3 Making Video Subjects

To create the video subjects, there must already exist images with their associated fits files inside of the `files/generated/pngs/` directory. In addition, video creation uses ffmpeg, which must be installed on the machine and available to use through the command line.

For each HEK event, subjects will be created from X images, Y of which will be shared with the next subject in the HEK event. There are some exceptions to the above. One being the last subject in a HEK event, which will still have X images, but may include more than Y overlapping images. The last subject

in any HEK event will always have the last frame match with the last image in the HEK event. The other exception occurs when there are less than X images available in the HEK event. In this case, only one subject will be created for the event using all of the images available. By default, videos run at 10 frames per second, and the values of X and Y are 50 and 5, corresponding to 20 and 2 minutes, respectively.

Sometimes, the script will take a couple of tries to generate a video for a single subject. This occurs when the video exceeds the maximum size of 1000kB. To lower the video size, new videos will be created with a higher constant rate factor (CRF), slightly lowering the quality. This value is increased by 1 from the default of 25 until the video is smaller than 1000kB. The maximum video size can be altered by the parameter **-s <size in bytes>**, but Zooniverse will not except subjects larger than 1000kB.

The default location for the video subject files is `files/generated/mp4s/`.

2.4 Creating Metadata

After the videos have been created, for each video subject that the script creates, it will also create a line in the metadata file `meta.csv`. Details of the created metadata can be found in section 3.

The default location for the metadata files is `files/exports/`.

3 Subject Metadata

Each created subject is represented by one line in the created `meta.csv` file. Each line contains the following data in the order shown below.

#file_name

The name of the video file as saved to the local disk.

#fits_names

A list of the names of the fits files used to create the subject. By default, the list will contain 50 file names.

#frames_per_sub

The number of frames included in this subject.

#event_db_id

The id of the HEK event as saved in the Solar Zooniverse Processor database.

#sol_standard

The sol standard of the HEK event. Includes the starting date and time of the event.

#visual_type

The type of media being uploaded - always the string "video".

#framerate

The number of frames per second of the subject.

#cadence

The cadence used when generating the video. Note that this value could be incorrect if the video subjects are created from images with a different cadence than the script is currently using.

#start$_time$

The start time of the subject. Stored as a string in the form: **YYYY-MM-DD HH:MM:SS.SSSSSS**.

#end$_time$

The end time of the subject. Stored as a string in the form: **YYYY-MM-DD HH:MM:SS.SSSSSS**.

#im_ll_x

The x pixel value of the lower left corner of the AIA image in the video, normalized by **#width**.

#im_ll_y

The y pixel value of the lower left corner of the AIA image in the video, normalized by **#height**.

#im_ur_x

The x pixel value of the upper right corner of the AIA image in the video, normalized by **#width**.

#im_ur_y

The y pixel value of the upper right corner of the AIA image in the video, normalized by **#height**.

#width

The width of the video resolution in pixels.

#height

The height of the video resolution in pixels.

#naxis1

Used for creating SunPy maps. Defined as: *Pixels along axis 1.*

#naxis2

Used for creating SunPy maps. Defined as: *Pixels along axis 2.*

#cunit1

Used for creating SunPy maps. Defined as: *Units of the coordinate increments along naxis1 e.g. arcsec.*

#cunit2

Used for creating SunPy maps. Defined as: *Units of the coordinate increments along naxis2 e.g. arcsec.*

#crval1

Used for creating SunPy maps. Defined as: *Coordinate value at reference point on naxis1.*

#crval2

Used for creating SunPy maps. Defined as: *Coordinate value at reference point on naxis2.*

#cdelt1

Used for creating SunPy maps. Defined as: *Spatial scale of pixels for naxis1, i.e. coordinate increment at reference point.*

#cdelt2

Used for creating SunPy maps. Defined as: *Spatial scale of pixels for naxis2, i.e. coordinate increment at reference point.*

#crpix1

Used for creating SunPy maps. Defined as: *Pixel coordinate at reference point naxis1.*

#crpix2

Used for creating SunPy maps. Defined as: *Pixel coordinate at reference point naxis2.*

#crota2

Used for creating SunPy maps. Defined as: *Rotation of the horizontal and vertical axes in degrees.*

#stddev_crpix1

The standard deviation of the values of **crpix1** in the .fts files used to generate the images for this subject. Ideally, should be 0.

#stddev_crpix2

The standard deviation of the values of **crpix2** in the .fts files used to generate the images for this subject. Ideally, should be 0.

#stddev_crota2

The standard deviation of the values of **crota2** in the .fts files used to generate the images for this subject. Ideally, should be a number on the order of $\leq 10^{-5}$.