How to: SpotiPy!

1. Clicker Script

set all your parameters (starts at ~line 130 and ends at line 163)

- **myemail**
- **mydir** (right now should already have the correct dir, feel free to change)
- **NOAA** (sunspot number)
- **latitude** (see f.e. https://www.solarmonitor.org/)
- **wavelength** (just make sure the name is consistent here)
- **No_LD_Series** (default = 'hmi.Ic_noLimbDark_720s')
- **LimbDark_Series** (default = 'hmi.Ic_720s')
- **start_date** (in format YYYY-MM-DD and HH:MM:SS ; see here also https://www.solarmonitor.org/ to guess the first moment the sunspot is visible on the sun)
- **days** (default = 11 days, one visible solar rotation)
- **dt** (cadence/frequency of images, mostly 6h or 12h)

If you use my SDO data, which is already downloaded, make sure to set: **FITS_downloaded** to **True**, otherwise it will download the first image from the series from SDO and make a new directory (if SDO download works today, you an also try this!)

➔ Run script, click sunspot-center (note that the image will appear flipped compared to the Solar Monitor)
➔ Output should be a txt file with the coordinates first in pixels then arcsec in the folder of the NOAA

2. Spotipy_LimbDarkening.py

BEFORE running, set the parameters (starts at line 677, ending in line 741)

(You will have multiple options to run this, so depending on where your data comes from, what output you need/want to save and so on you need to set a few Boolean cases)

These parameters can and should be the same as in the clicker script (if you used it before):

- myemail
- mydir
- NOAA

- latitude
- wavelength
- No_LD_Series = 'hmi.Ic_noLimbDark_720s'
- LimbDark_Series = 'hmi.Ic_720s'

If you need to need to download the sunspot data (set **download_needed** = **True** in line 737, see further below) you can now specify how they should be downloaded:

i) Specify the **start_time** (in format YYYY-MM-DD and HH:MM:SS) as the first appearance of the sunspot on the limb and say how many **days** should be observed (default = 11) and the frequency/step between two images via **dt** (in hours)
(you can use https://www.solarmonitor.org/ or my list of sunspots for the time of first sighting)

ii) if you dont want tp specify the cadance and rather use a list (array) of time stamps (to have more variety closer to the limb or to work with telescopes that only get data once per day) and input it into **time_stamps** (or import a list there via np.load or similar)
Important: if you want to use time stamps you must set **use_time_stamps** = True

Now we need to give the rough coordinates of the first sunspot position (so just as it enters the visible solar disk). These coordinates will then be used by the python module 'sunpy' later on, which will rotate them along side the solar rotation for the chosen latitude. Solar Rotation is not rigid, but instead we see differential rotation, meaning the solar surface rotates faster at the equator than at the poles.
Unfortunately, sunspots also show a slightly different rotation rate than the 'quiet sun' around them, meaning even if we follow the sunspot with the sunpy-coordinates, we will sometimes move faster or slower than the sunspot, making it move out of frame. This needs to be avoided.
Unfortunately, this ends up being mostly a matter of trial and error.

i) If you used the clicker for the coordinates before set:
clicker_coordinates = True

The script will then read out the coordinates from the txt created by the clicker script.
If you run the 'Spotipy_LimbDarkening.py' script and the sunspot moves out of

frame, you can try to correct it by setting an offset to start coordinates by editing:

- **x_start_offset** = 0 (it usually helps to get closer to the limb. By setting this to -10 e.g. you move a bit closer to the limb)
- **y_start_offset** = 0 (depending on which hemisphere you are you can move closer to the equator (=faster rotation))

ii) If you didn't want to use the clicker, set **clicker_coordinates** = **False** and instead manually enter the start coordinates:

- **x_start_position**
- **y_start_position**

you may need to also try a few times if the sunspot moves out of frame

if there are still troubles with keeping the sunspot in frame edit the size of the window that's following the sunspot: **frame_size** (default = 400)

Set threshold for masks:
penumbra_threshold = 175 (default)
umbra_threshold = 83 (default)

lastly, set:

- **download_needed** (If data was already downloaded set to **False**)
- **wantCheck** (set to TRUE if you want to see images of each mask/sunspot)
- **wantSave** (set to TRUE if you want to save those images of masks as pngs in the output folder)