**Phenocam processing – Python codes**

SITES Spectral is a network of research stations that collect spectral data from Swedish ecosystems. Inside the network, the data is collected in a similar way and later it is harmonized, so the data can be comparable among sites and along the time.

One of the sensors in the network are phenocameras. These are regular cameras (RGB) that collect data continuously, at a particular frequency. Generally, in SITES Spectral, they collect one photo each 30 minutes, between 9 a.m. and 3 p.m.

The purpose of phenocameras is to describe phenology of ecosystem, or their life cycle. The photos can be used to simply observe changes in vegetation along the year. However, the spectral information can also be used to quantify the ecosystem photosynthesis, by means of spectral indices.

In SITES Spectral, we are using 2 spectral indices: GCC (green chromatic coordinate) and RCC (red chromatic coordinate). They are calculated as:

GCC = DNgreen / (DNred + DNgreen + DNblue) (Toomey et al., 2015)

RCC = DNred / (DNred + DNgreen + DNblue) (Richardson et al., 2009)

High values of GCC and RCC depict green vegetation, active vegetation, or photosynthesis. Low values, the contrary.

In SITES Spectral, four levels of processing have been defined:

L0 – Raw data. Photos as they come from the phenocamera

L1 – We remove images have low quality: sun glare, rain droplets, fog or shadows, or represent low-light or blurry conditions.

L2 – We create RGB, GCC and RCC daily composites, by averaging all images of one day, between 10 a.m. and 2 p.m., solar noon. The composite is a raster file with 3 bands for the RGB, and one band for GCC and RCC.

L3 – GCC and RCC time series from all existing data. It is in a CSV format. Each record of the database is a daily average for all pixels within a defined Region of Interest (ROI).

A ROI is an area, within a phenocamera photo, that comprehends a vegetated area or any target of interest. We can have a single ROI that covers most of the image (except the sky, for instance) or several ROI within the same photo. For example, in experimental croplands, there can be more than one experimental plot (crop) in a photo. Then, we define one ROI per crop plot.

The processing is done in Python, using the following codes:

1. imgRename.py

First step is to homogenize names of photos from the phenocameras, to simplify the work of the codes later. Each station has a different way of naming photos. If we want to run the codes on different camera datasets, it would require calling the variables each time in a different way, to recognize the files.

By renaming all photos at the beginning of the process, we make sure all photos from different stations/cameras will match the variable naming criteria of the following codes.

1. removeDark.py

Some phenocameras are connected 24 hours, so we have some night scenes that we don’t want. Also, sometimes the light conditions are too dark (too cloudy, rainy, etc.), and also those images we do not want. We remove those images from the dataset and only keep photos that have good light conditions.

1. timeFilter.py

For phenology purposes, only images between 10 am and 2 pm (solar noon) are valid. Beyond that time frame, the light is so scattered that the relation between reflected light and plant productivity is not linear. For that reason, we keep only photos taken between those times.

1. dailyAVG\_L2.py

All images from the same day will be averaged, pixel by pixel, to create a composite. This is done per band, individually; in other words, all red bands for one day will be averaged to create a new composite raster, and so on for blue and green.

1. snowThreshold.py

Snow is an important meteorological event that eventually condition the phenology of the ecosystem under study. However, the spectral signature of snow is very different to the one of vegetation. We want to keep the information about snow, but also tag the record so the user knows that the low spectral signature is due to presence of snow, and not low greenness.

With this code, we identify which images present snow and tag them. This tag is a column in the time series CSV in level 3 (L3).

1. 1day\_summary\_L3\_ST.py

This code calculates the time series for RGB, GCC and RCC, for a particular phenocamera and ROI. Only one ROI can be processed at a time and one year of data, each time. The result is a CSV file.

If more than one ROI has been defined for a phenocamera, we need to run this code as many times as number of ROIs. Each time, we will obtain a CSV file.

In the same fashion, if more than one year of data is created, we will have to run the code many times, and each time we will have a new CSV file.

1. mergeAnnualCSV.py

This codes merges all data for several years in a single one, by appending data of successive years at the end of the CSV of the first year (new rows at the end of the file). Only data is appended. The headers of the appended files are removed, so only one header remains at the final file.

1. mergeROIcsv.py

If more than one ROI has been defined for a phenocamera, we will have several CSV files. This code merges all files for a single camera in one. RGB, GCC and RCC values for each ROI will be added to the CSV file of the first ROI (ROI1).

Original ROI CSV files share two columns, one for Timestamp and one for Day of Year (DOY). Only the fist ROI1 file keeps those columns. These two columns are removed from appended ROIs (ROI2, and so on).

REFERENCES:

Toomey, Michael, Mark A. Friedl, Steve Frolking, Koen Hufkens, Stephen Klosterman, Oliver Sonnentag, Dennis D. Baldocchi et al. "Greenness indices from digital cameras predict the timing and seasonal dynamics of canopy-scale photosynthesis." Ecological Applications 25, no. 1 (2015): 99-115.

Richardson, Andrew D., Bobby H. Braswell, David Y. Hollinger, Julian P. Jenkins, and Scott V. Ollinger. "Near-surface remote sensing of spatial and temporal variation in canopy phenology." Ecological Applications 19, no. 6 (2009): 1417-1428.