List of frequent errors with pyDMS sharpening

This document will explore some of the common mistakes that could be made using the downscaling code. Some might directly give errors, some might be more insidious and not return an error but still return an abnormal/empty result. If you encounter an error that's not on this list and you think it should, please email me to quentin.dehaene@jpl.nasa.gov.

A general good practice to avoid errors is to double check the paths you had to type, that could be incorrect. Also, some user induced mistakes could be undetected and still allow the sharpening, but a good indicator is the output of the sharpening cell used. It should look like this, with relatively low values for bias and RMSD:

```
Training regressor...
Homogeneity CV threshold: 0.27
Number of training elements for is 19954 representing 79% of avaiable low-resolution data.
Sharpening...
Saved MEM
Residual analysis...
Saved MEM
Saved MEM
Saved MEM
Saved MEM
LR residual bias: -0.004309888819884992
LR residual RMSD: 0.5878601407924853
Saving output...
```

If it doesn't, it's probably that there's a mistake somewhere, and likely it is one of these:

1) **Import cell:** ModuleNotFoundError: No module named '…'
   Depending on the missing module it could be due to two things. If the module is a Python module (numpy, rasterio…) then it's most likely because you haven't installed said module in the working environment. In that case check that you are working with the appropriate conda environment and install the missing package.
   If the module is a pyDMS module, then either the structure of the pyDMS file isn't correct (cf tutorial). Or it might be than the setup.py installation has been made in another environment. So again, check that you are in the right env and run python setup.py install (in the pyDMS_master folder) (cf tutorial). I also advise to restart your VS Code and your kernel. This error should only appear in your first run so it's likely an installation problem.

2) **Path defining cell:** `SyntaxError:` (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape
   Using backslashes when you type in a file name isn't accept by Python unless to you use a raw string (ie type r before your string)

3) **Define bounding box cell:** If you make a mistake while typing the bounding box coordinates, the desired resolution or the time interval, you should receive an error. Still, I

recommend that you check the order of the bounding box, since you could have set accepted coordinates but inverted latitude and longitude and get an image at a completely different area.

4) **ECOSTRESS QC files:**

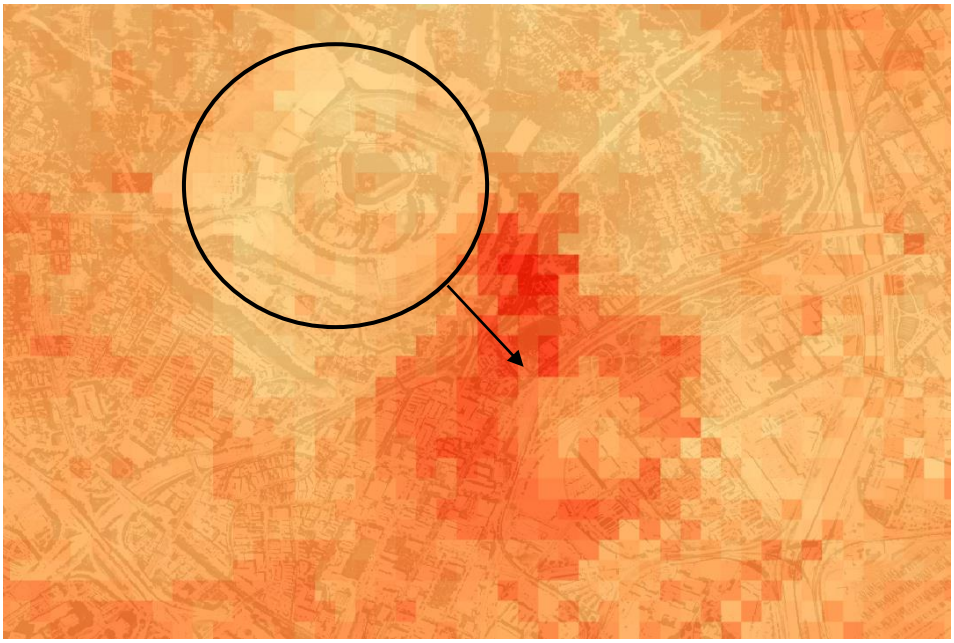
The preprocessing of the QC files should not be an issue but if the QC_dir defined earlier is not correct, or is empty or doesn't contain a QC file for every LST scene, you'll see an error if you run the sharpening with one of the first two cells (i.e. the cells where QF files are needed) you'll see RasterioIOError: No such file or directory. In that case very your QC folder and that it contains the QF files for each scene.

Another problem with QC may occur, it is really rare but also unexpected hence uneasy to fix. It's a problem due to AppEEARS, the QC file might be not aligned with the LST file. In that case if you decide to cut the extent of the final product to the extent of the ECOSTRESS LST (the 2nd cell for sharpening), you will receive: ValueError: operands could not be broadcast together.

There's no know fix for that problem, try to renew your AppEEARS request or clip your desired output to a slightly different extent so that the final extent is included in both the QF and the LST. However, the shift also means that the quality of each pixel is not correctly assessed. See this example with 1.5-pixel shift.

5) **Approximative ECOSTRESS geolocation**

Here is also an insidious error, that would affect the results of the training. The approximate geolocation of ECOSTRESS. Here's an example over Dodgers Stadium:



Here the Dodgers Stadium (which is hotter than the surroundings), is shifted by several hundred meters. This is an ECOSTRESS geolocation problem. When the sharpening will occur, the training will be biased by this inexact ground truth. Here the surroundings of the stadium are going to be mistaken for hot and the stadium area for cooler. Indeed, the correct geolocation for the algorithm is the Sentinel's and luckily it is often much better that ECOSTRESS's, but that significates that the results would tend to show the features visible with Sentinel, so it is important that these features have their actual temperature in their actual place in the ECOSTRESS scene.
This situation will not create any error in the code but can generate results that are incorrect. So, I would advise to check on a GIS software the quality of the ECOSTRESS geolocation at some point.

6) **Scaling related problems**

In this case too, the scaling cell won't return any error, so only the result will alert you if something is wrong. First possibility, you didn't scale your data, this can happen if you don't run the cell or if you thought yours were already scaled. To check the easiest is to use a GIS software and check the scale. You can also open the metadata of the file or read the product description. If you run the sharpening on a non-scaled LST here is the kid of results you would see:

```
Training regressor...
Homogeneity CV threshold: 0.22
Number of training elements for is 15669 representing 79% of avaiable low-resolution data.
Sharpening...
Saved MEM
Residual analysis...
Saved MEM
Saved MEM
Saved MEM
Saved MEM
LR residual bias: 4420.595274277293
LR residual RMSD: 4552.063373686725
Saving output...
```

Such values for bias and RMSD are completely abnormal, it is due to the 50 scale.

However, the opposite can also happen and you can scale a product with the force of habit but, if it was provided to you or if it is a multi-day aggregate or if you are working with ECOSTRESS Collection 2 tiled products, it is likely already scaled. So, if your LST results are close to 0 (in the single digits), it is not because the temperature is in °C or in °F but most likely because the data was scaled twice.

Also, if you didn't use the cell for scaling and that you lr_dir_sc is empty then you won't receive an error but simply the cell be immediately executed with nothing printed.

7) **Sharpening cell error: '**NoneType' object has not attribute 'GetProjection'

```
Training regressor...

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[39], line 55
     52     disaggregator = NeuralNetworkSharpener(**opts)
     54 print("Training regressor...")
---> 55 disaggregator.trainSharpener()
     56 print("Sharpening...")
     57 downscaledFile = disaggregator.applySharpener(highResFilename, lowResFilename)

File c:\Users\qdehaene\AppData\Local\anaconda3\envs\jvsrp_main\Lib\site-packages\pydms-1.0-py3.12.egg\pyDMS\pyDMS.py:315, in DecisionTreeSharpener.trainSharpener(self)
    311 scene_LR = gdal.Open(lowResFile)
    313 # First subset and reproject low res scene to fit with
    314 # high res scene
--> 315 subsetScene_LR = utils.reprojectSubsetLowResScene(scene_HR, scene_LR)
    316 data_LR = subsetScene_LR.GetRasterBand(1).ReadAsArray()
    317 gt_LR = subsetScene_LR.GetGeoTransform()

File c:\Users\qdehaene\AppData\Local\anaconda3\envs\jvsrp_main\Lib\site-packages\pydms-1.0-py3.12.egg\pyDMS\pyDMSUtils.py:186, in reprojectSubsetLowResScene(highResScene, lowResScene, resampleAlg)
    183 def reprojectSubsetLowResScene(highResScene, lowResScene, resampleAlg=gdal.GRA_Bilinear):
    184
    185     # Read the required metadata
--> 186     proj_HR, gt_HR, xsize_HR, ysize_HR, extent = getRasterInfo(highResScene)[0:5]
    187     proj_LR, gt_LR, xsize_LR, ysize_LR = getRasterInfo(lowResScene)[0:4]
    189     # Transform "middle pixel" and "middle pixel + 1" between LR and HR projections
    190     # to obtain LR resolution in HR projection. This method can handle different
...
---> 31     proj = r.GetProjection()
     32     gt = r.GetGeoTransform()
     33     sizeX = r.RasterXSize

AttributeError: 'NoneType' object has no attribute 'GetProjection'
```

This error is very unlikely to happen in the latest version since the potentially problematic paths are automatically defined. In case you encounter this, it is an error returned by GDAL during the process, the code is trying to identify the projection of a raster, but said raster is 'NoneType'. This means that this raster doesn't exist, in our case it is the high resolution image (HR suffix). This is simply a case where the hr_img_path is leading to a non-existing file. It won't happen unless you overwrite the path with your own, since it is defined earlier in the code. But if you did, review that path and make sure it is correct.

8) **Extents of Sentinel and ECOSTRESS not matching**

If you mistyped your bounding box or if you haven't set the path to the correct files, it can lead to having the S2 image whose extent isn't overlapping the extent of the ECOSTRESS scene to be sharpened, in that case the sharpening cell would return.

```
Training regressor...
Homogeneity CV threshold: 0.00
Sharpening...
c:\Users\qdehaene\AppData\Local\anaconda3\envs\jvsrp_main\Lib\site-packages\pydms-1.0-py3.12.egg\pyDMS\pyDMS.py:631: RuntimeWarning: Mean of empty slice
  print("LR residual bias: "+str(np.nanmean(residual_LR)))
c:\Users\qdehaene\AppData\Local\anaconda3\envs\jvsrp_main\Lib\site-packages\pydms-1.0-py3.12.egg\pyDMS\pyDMS.py:632: RuntimeWarning: Mean of empty slice
  print("LR residual RMSD: "+str(np.nanmean(residual_LR**2)**0.5))
Saved MEM
Residual analysis...
Saved MEM
Saved MEM
Saved MEM
Saved MEM
LR residual bias: nan
LR residual RMSD: nan
Saving output...
```

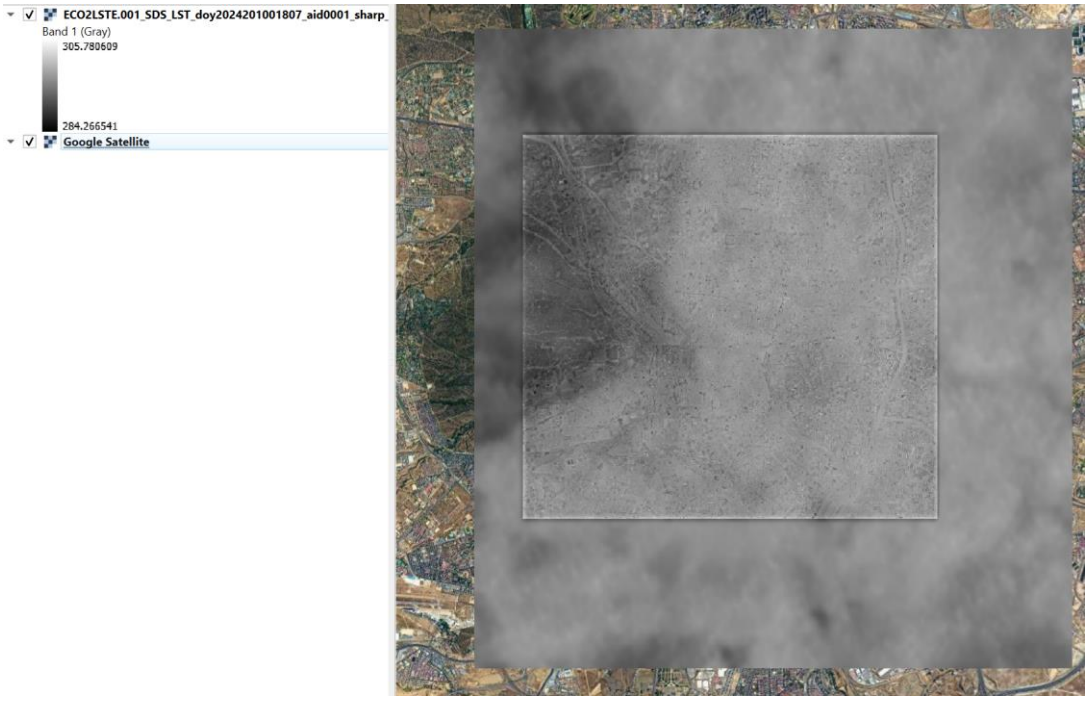
9) **Clipping the S2 image to an extent bigger than its actual extent.**

If you decide to use the two cells that allow you to cut the extent of the final product to one of your choosing, you have to make sure that said extent is fully contained in the original S2 extent. Otherwise, the end result will be incorrect.

```
Training regressor...
Homogeneity CV threshold: 0.23
Number of training elements for is 15670 representing 33% of avaiable low-resolution data.
Sharpening...
Saved MEM
Residual analysis...
Saved MEM
Saved MEM
Saved MEM
Saved MEM
LR residual bias: 5.371140298474936
LR residual RMSD: 7.206608596658802
Saving output...
```

Here again the values for RMSD and bias should alert you, this is too big to be acceptable. On QGIS, let's visualize what happened:

The S2 image (color) has a much smaller extent than the ECOSTRESS image (grayscale). If we try to force the sharpening on a bigger extent than the high resolution image (the ECOSTRESS extent per choice here), we obtain :



This is quite obviously incorrect. If you obtain this, review the extent you used to clip. Especially, the order of the coordinates, because GDAL's required order is not intuitive.