FitsFix – some Details, Comments & Explanations

With most sensors the sensor has more columns than rows. The library astropy regards the smaller shorter axis as the axis of the rows. Further reading FITS File Handling (astropy.io.fits) — Astropy v5.2.3.dev0+g32d49b960.d20230328 How it works with squared sensors needs to be checked.
Be sure to have astropy and numpy installed.

A whole column is counted as defect when more than 25% (=Default) of the elements of a column are counted as local defects. The program compares the pixel value of a central pixel to the values of one pixel to the left and one pixel to the right in the same row. So it works on 2D = grayscale images only. Intensity values to be integers.

A pixel is treated as a local defect when the intensity **I** of the element in the central column is significantly **less** compared to **both of the two** neighboring elements **in the same row**.
Significantly means that the intensity value is less than a given threshold T (Default =50 ADU) compared to both neighboring elements.

The default values were chosen because they worked well with my images.

The program searches solely for **darker columns** in the image. So far I had no brighter columns in my images. Brighter columns can be corrected by entering their column number manually to the file defects.csv. Be careful in the reading of the column's number in an external viewer as it needs to be 0-based.(Most viewers are 0 based- so no problem).

Images from Bayer matrix sensors need to be handled different to images from monochrome sensors.

Let r be the row index and n the column index.

Local defect = (I(r ,n) +T) < I(r, n-2)) AND (I(r,n) +T) < I(r, n+2)) for Bayer (Default)

Local defect = (I(r ,n) +T) < I(r, n-1)) AND (I(r,n) +T) < I(r, n+1)) for monochrome

In case that a neighboring column that is used for comparison is also defect then the detection will most probably fail.

The result of the scan is documented in the file defects.csv located in the current working directory. In case that the program does not detect all defect columns you may edit the file defects.csv on your requirements. This is the way to correct bright columns.

Take care that the manually entered integer number does not exceed the total amount of columns for the specific sensor.

For just creating a defects.csv file run FitsFix.py with option –m s in scan only mode.

In case a column is identified as defect in e.g. 1-2 images in a series of 20 then the defect may be not visible in the result after stacking. You may then erase the column number and it's separator ; in the csv-file, save the csv-file and re-run the program.
In case the original fits file has an entry ISTRUME in its header this value will be copied to the defects.csv so the result of the scan can be mapped to a camera and its sensor.