

The attributes associate with the stack class are given below

**self.width**

The width of the image.

**self.height**

The height of the image.

**self.nframes**

The number of frames in the image stack, this is the total number of frames and not the number of frames between resets.

**self.starting\_value**

The frame number corresponding to the start of the first complete image block.

**self.difference**

The number of frames in a block.

**self.number\_of\_stacks**

The number of blocks in the image stack.

The class has several useful functions which are outlined below

**get\_frame(self, frame\_number):**

This functions return the frame requested as frame\_number, the frame is returned as a NumPy array.

**find\_resets(self, start\_frame\_number, end\_frame\_number, line\_number):**

This function returns two values, the first is the first frame of the first whole block of data (self.starting\_value), the second is the length of a block (self.difference), currently this is either 100 or 500 frames. This function is redundant as the values can be access as class attributes. Start frame and end frame define the frames over which the resets are to be discovered, the start frame typically should be zero. The line\_number is arbitrary but should be between zero and the height of the image.

**load\_block(self, blocknum):**

This function returns the whole block requested by blocknum (the particular block in the series) as a 3D NumPy array of frames.

**get\_pixel\_trace(self, start\_frame\_number, end\_frame\_number, x\_pixel, y\_pixel)**

This function returns the values of a single pixel with coordinates x\_pixel and y\_pixel between the start frame and end frame. The data is returned as a 1D NumPy array.

## Example code

The example code provided (main.py) opens a stack with a defined filename and prints the class attributes of the stack. It then loads the block requested, set as the 10th by default, and from this the equivalent CMOS data is calculated for the block and displayed. The brightest pixel in the CMOS frame is selected and highlighted, this is shown in the top image. The middle image corresponds to a 21 x 21 pixel area around the brightest pixel. Once the brightest pixel has been identified a line trace of the pixel and its cardinal pixels throughout the block is plotted in the bottom figure.