

AIADS Configuration File

Version 5.0.10

The aiads software requires a configuration to specify how to generate anomalies. The configuration specifies the candidate image, the historical images, and all parameters for anomaly generation. The configuration file is JSON and at the top level, all parameters are contained in an object named “parameters”.

Image Specification

The configuration JSON contains one structured type, the `image_spec`, which defines an image and its acquisition time. The structure of the `image_spec` is:

```
{  
  "path": "/path/to/image.tiff",  
  "stamp": "2021-07-16T09:00:00+0000"  
}
```

The stamp must be in exactly this format. (YYYY-MM-DDThh:mm:ss+offset), that is, an ISO-8601 date.

In the “parameters” object, the candidate image is the tag “candidate” with an `image_spec` specifying the candidate image. Likewise, the “history” tag is an array of `image_specs` specifying the Historical Image set. The order of this array is not significant – it is sorted based on the stamps of the historical `image_specs`.

There are several other parameters which change the way images are interpreted.

“pixel_bits” - (positive integer) The number of bits of an (integral) image data type. The Pixel values are interpreted as $\frac{value}{2^{bits}-1}$ thus, if bits == 1 then value is unchanged. Certain metrics work better when the values are unchanged.

“logistic” - (boolean) This imposes a logistic curve on the pixel values **instead** of applying pixel_bits. The scale of the pixel is always 0.0 – 1.0. This should be applied to reduce the effects of the darkest and lightest objects in the image, i.e. glint and dark shadows, which can affect the variation generated by variation based metrics. This parameter is applied to all bands of all images. For each band, the Logistic Equation is $\frac{L}{1.0 + \exp(-K - (band - X0))}$ where L is 1.0 (representing the maximum value of the result), X0 is the mean of the band for the entire image, and $K = \frac{2.1965}{3.0 * \sigma}$ of the band. Note that most non-transferring indices are not valid when logistic is true, so it should be used only for transferring indices (i.e. red, green, blue, panchromatic, etc.).

“bands” - (array of string) – list of bands in the image. Band names should be used as the requirements for an index. i.e. The “red” index reifies the band named “red”. Whichever band is named “red” will be exported as the index “red”.

“indices” - (array of string) List of the indices of the image. Indices are computed based on a band or bands called out in “bands”

Output Specification

The output of AIADS is the Anomaly Image. Several parameters guide the generation of the anomaly image.

“color_output” - (boolean) If true, then the output is a scaled RGB 8 bit image, which has been scaled. i.e. is suitable for use by human analysts. If false, then the image is an unscaled mono Floating Point image, where the pixel value is the base anomaly score generated by the metric. Threshold is applied to mask low-value pixels. This method is suitable for using the images for further automated analysis, or for the use of viewing tools.

“output_directory” - (string) The directory to output the anomaly image.

“autoScale” - (boolean) If true, scale each tile so that anomalies appear at a “reasonable” scale.

Anomaly Generation

“tile_size” - (integer) The size in pixels of one side of a (square) tile. Larger values process faster, smaller values use less memory. In metrics, methods, and indices which process the

tile when instantiated, only this size tile or smaller will be analyzed by the algorithm, thus, different results are possible with different tile sizes when using algorithms which base their operation on the contents of the tile. Examples are the center method and the Gabor index.

“interactive” - (boolean) Display a progress line which estimates the time to completion of the anomaly image.

“singleThread” - (boolean) Process tiles with a single thread. When true, all tiles will be processed linearly, and in order. This greatly limits memory usage, and can be used when memory overflows are occurring. When false, the system will use several threads for processing the tiles.

“sub_tile_size: - (integer) The size of a side of the subtile. A subtile is created when an algorithm must divide the tiles into smaller tiles. When this occurs, the sub_tile_size specifies that size. An algorithm divides the tile into subtiles if it needs to examine the area adjacent to the pixel it is analyzing. This is currently used in version 5.0.10, but not in previous versions.

“method” - (string) The name of the method algorithm to use to compare pixels.

“metric” - (string) The name of the metric algorithm to use to compute anomalies.

“threshold” - (float) The minimum anomaly to display. Anomalies less than this value are transparent. The range of this value is 0.0 - 1.0 where 0.0 displays all anomaly values and 1.0 displays only anomalies of value 1.0

“scale” - (float) The factor to multiply anomalies by in order to get a visible result. Most AIADS metrics are designed to ordinarily work at a scale of 3.0

Parameter Name	Type	Default	Description
candidate	image_spec	None	The image to analyze
history	Array of image_spec	None	The Historical Images
pixel_bits	integer	None	Number of image bits
logistic	Boolean	False	Use a logistic curve which is scaled to the data instead of pixel_bits
bands	Array of String	None	The band names

Indices	Array of String	None	The index names
color_output	Boolean	None	Use color output image
output_directory	String	None	Path for output
autoScale	Boolean	True	Scale output dynamically
tile_size	Integer	1000	The size of a tile in Pixels
Interactive	Boolean	False	Display progress
singleThread	Boolean	False	Do not process tiles in parallel
sub_tile_size	Integer	10	Dimensions of any subtile of Tiles
method	String	None	Algorithm to compare pixels
metric	String	None	Algorithm to determine anomalies
threshold	Float	None	Minimum anomaly to display
scale	Float	None	Factor to multiply anomalies by.

Parameters without a default must be specified, or execution will terminate with an error.