

Image Sequencer

User Manual ver 0.1

May 26, 2019



Table of Contents:

Introduction:	4
Specifications:	4
Input Image Format options:	4
Output Options:	4
Color Scheme – RGBA:	4
Image coordinates:	4
Processing Environment:	5
Image Modules:	5
Add-qr-module:	6
Average-module:	7
Blend-module:	8
Blur-module:	9
Brightness-module:	10
Channel-module:	11
Color Temperature:	12
Color-bar-module:	13
Colormap-module:	14
Contrast-module:	16
Convolution-module:	17
Crop-module:	18
DecodeQr-module:	20
Detect Edges:	20
Dither-module:	21
Draw-rectangle-module:	22
Dynamic-module:	23
Exposure:	25
FisheyeGI-module:	25
Flipimage-module:	26
Gamma-correction-module:	27
Gradient-module:	27
Grid Overlay:	28

Import-image-module:	30
Invert-module:	30
NDVI-module:	31
NDVI-colormap-module:	31
Overlay-module:	32
Paint-bucket-module:	33
Resize-module:	34
Replacecolor-module:	37
Rotate-module:	38
Saturation-module:	39
Text-overlay:	41
Threshold:	41
Tint:	42
White Balance:	43
Applications:	45
NDVI (NIR and Red) with colorbar:	45

Introduction:

This manual provides instruction for the web browser applications of [Image Sequencer](#). The user manual consists of specifications, a description of the modules and an applications section which describes how the modules, when combined, can be used for different types of image analysis.

Specifications:

Input Image Format options:

Jpg (freezes and locks up)

Jpeg

png

Gif

canvas

Max Image Size : 1Mb?

(processing time vs file size)

Bit depth ??

Color and black/white

Maximum length and width size?

Output Options:

PNG

GIF of sequences

Color Scheme – RGBA:

Color is represented as a 'tuple' in (R,G,B) format. Each color channel has a value between 0 and 255.

black is: (0,0,0),

white is (255,255,255)..

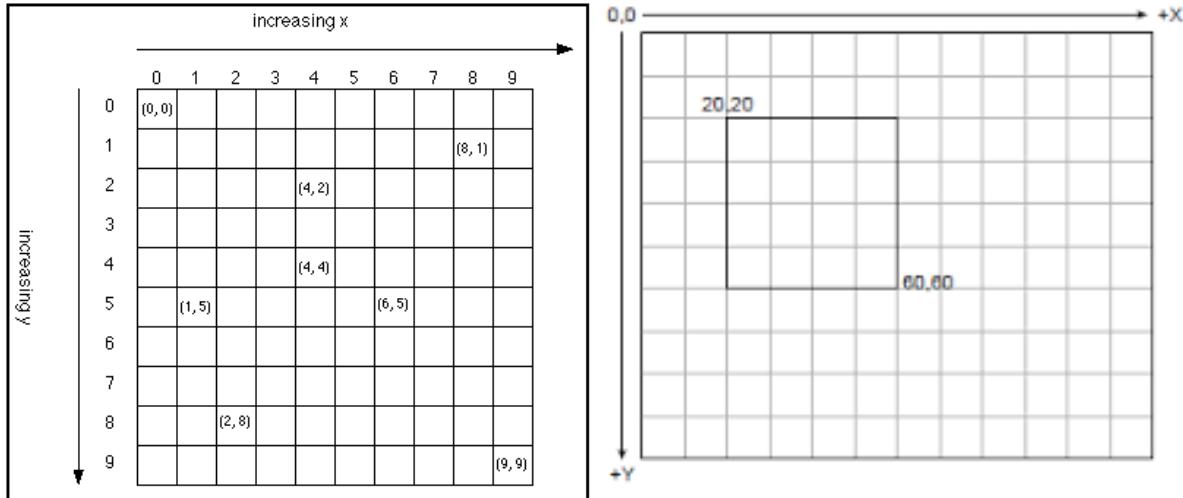
red is (255,0,0).

blue is (0,0,255).

‘A’ represents transparency with values 0 to 255.

Image coordinates:

Coordinates in (X,Y), which is (0,0) is located at the top left of the image



Processing Environment:

(What is the processing environment ???)

(processing time vs file size)

Image Modules:

Image Sequencer uses modules that perform a variety of step-by-step image processing tasks. Figure x describes the general workflow.

This section describes each module and a general implementation example. Most examples used Figure X as the input reference image.

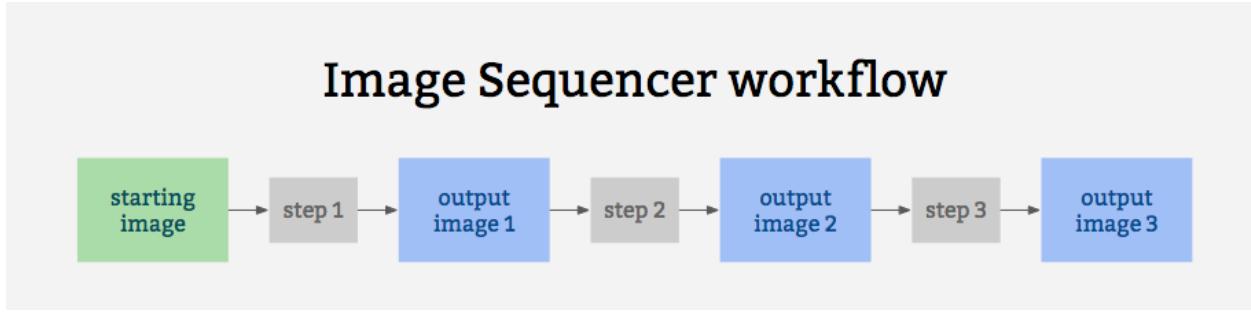


Figure Y Reference Image:

load-image ↗

This initial step loads and displays the original image without any modifications.



Additional information on the modules and other operating modes can be found at the Image Sequencer github site. <https://github.com/publiclab/image-sequencer/tree/main/docs> <https://github.com/publiclab/image-sequencer/blob/main/README.md>

Add-qr-module:

This module adds a QR code to the image that corresponds to an input string.

Usage:

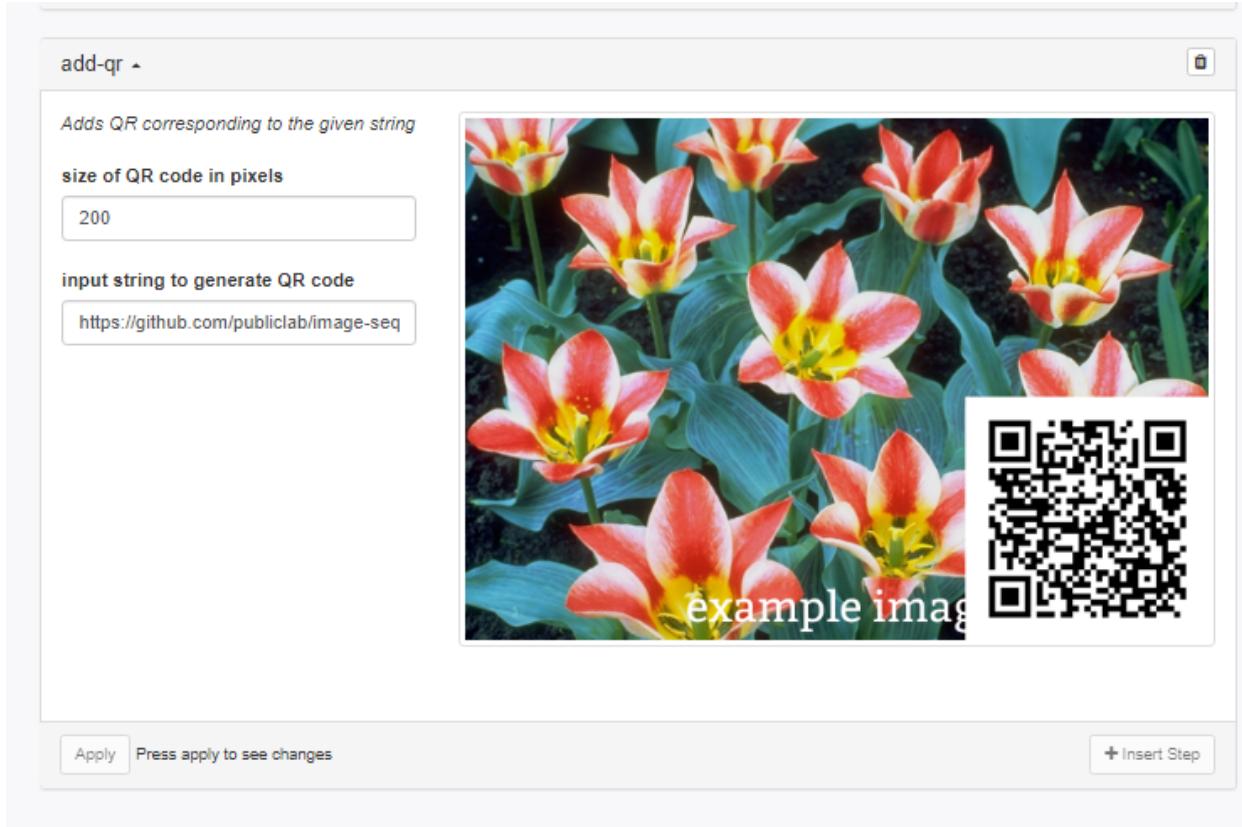
```

sequencer.loadImage('PATH')
  .addSteps('add-qr',options)
  .run()
  
```

where options is an object with the following properties:

- size : size of QR code in pixels (default 200)

- qrCodeString : input string to generate QR code



Average-module:

This module is used for averaging all the pixels of the image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('average',options)
    .run()
```



Blend-module:

This module is used for blending two images . *Blend two chosen image steps with the given function. Defaults to using the red channel from image 1 and the green and blue and alpha channels of image 2.*

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('blend',options)
    .run()
```

where options is an object with the following properties:

- offset: step of image with which current image is to be blended(Two steps back is -2, three steps back is -3 etc; default -2)
- func: function used to blend two images (default : function(r1, g1, b1, a1, r2, g2, b2, a2) { return [r1, g2, b2, a2] })

For this example it is necessary to import a second image.

load-image

This initial step loads and displays the original image without any modifications.



import-image

Select or drag in an image to overlay.
Choose File

Import a new image and replace the original with it. Future versions may enable a blend mode. Specify an image by URL or by file selector.

URL of Image to Import
./images/monarch.png

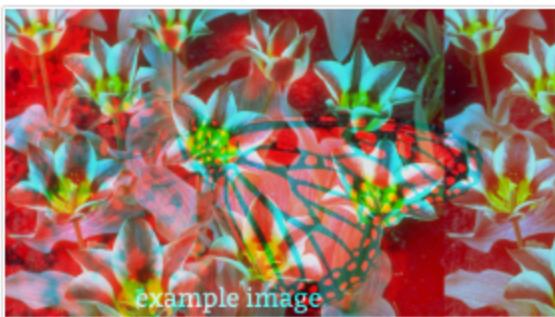


blend

Blend two chosen Image steps with the given function. Default to using the red channel from Image 1 and the green and blue and alpha channels of Image 2. Easier to use interfaces coming soon!

Choose which image to blend the current image with. Two steps back is -2, three steps back is -3 etc.
-2

Function to use to blend the two images.
function(r1, g1, b1, a1, r2, g2, b2, a2) {



Blur-module:

This module is used for applying a Gaussian blur effect.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('blur',options)
    .run()
```

where options is an object with the following property:

- blur : Intensity of Gaussian blur (0 to 5; default 2)

blur ▾

Applies a Gaussian blur given by the intensity value

Amount of gaussian blur(Less blur gives more detail, typically 0-5)

2.5

example image

Apply Press apply to see changes + Insert Step

Brightness-module:

This module is used for changing the brightness of the image.

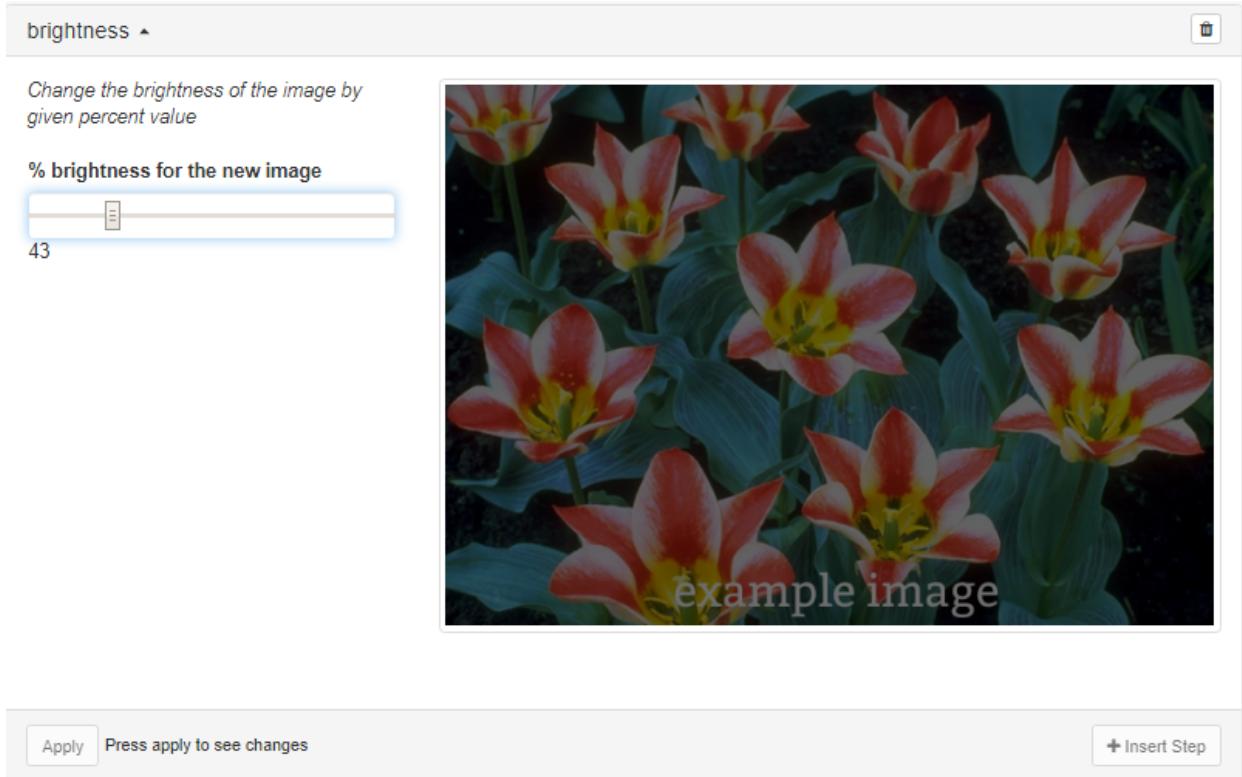
Usage:

```
sequencer.loadImage('PATH')
    .addSteps('brightness',options)
    .run()
```

where options is an object with the following property:

- brightness : brightness of the image in percentage (0 to 100; default 100)

note: sliders goes to 200? Not 100



Channel-module:

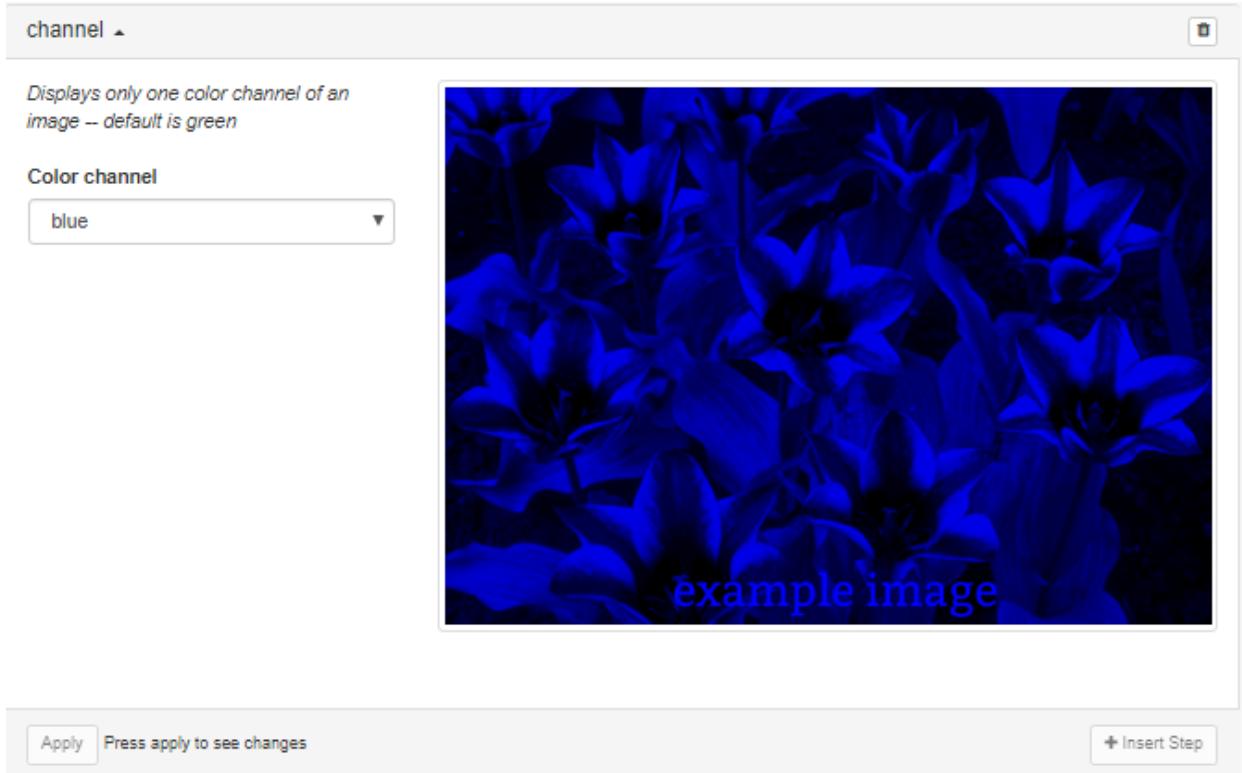
This module is used for forming a grayscale image by applying one of the three primary colors.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('channel',options)
    .run()
```

where options is an object with the following property:

- channel : color of the channel (red, green, blue; default green)



Color Temperature:

This changes the color temperature of the image.

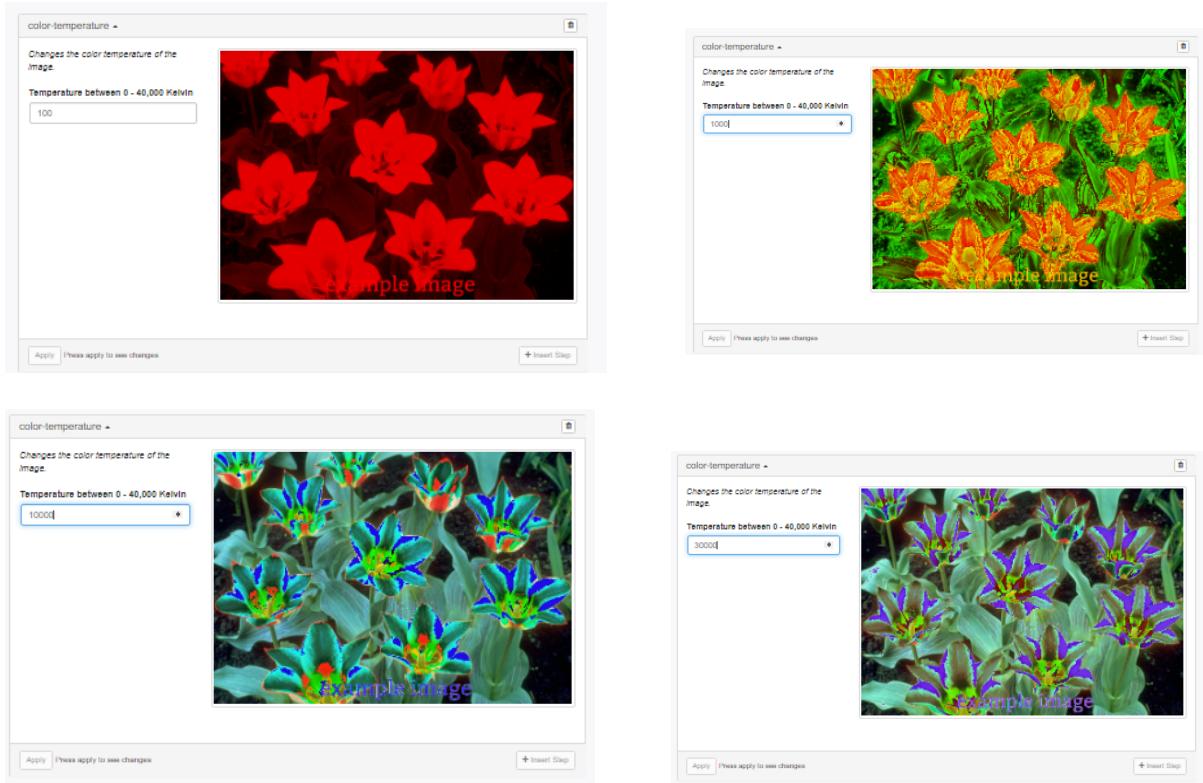
Usage:

```
sequencer.loadImage('PATH')
    .addSteps('color-temperature',options)
    .run()
```

where options is an object with the following property:

- temperature : temperature between 0 - 40,000 kelvin (default 6000)

Color temperature AT 100, 1000, 10,000 and 30000.



Color-bar-module:

This module is used for displaying an image with a colorbar.

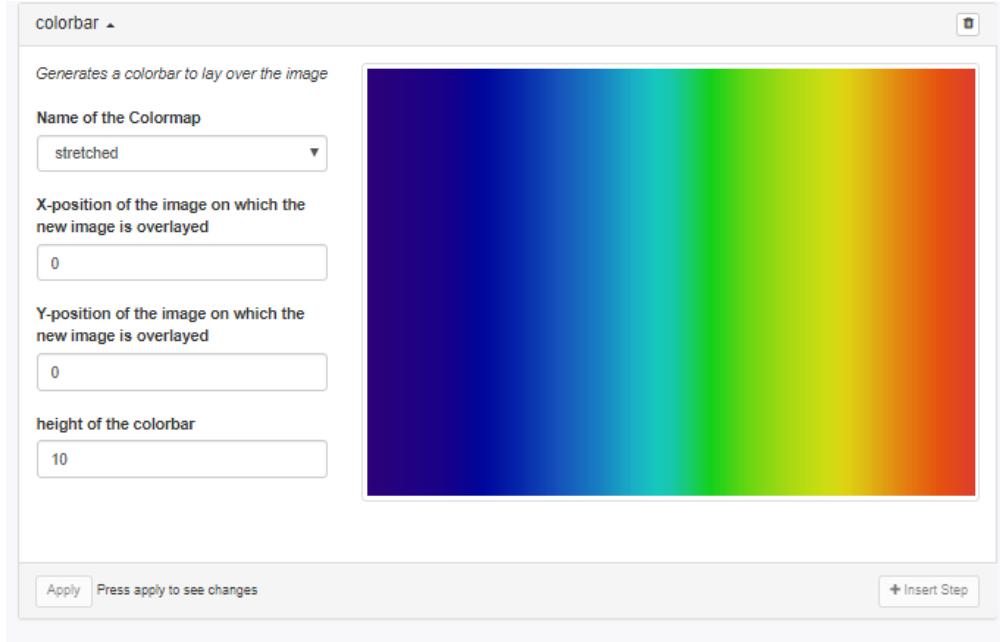
Usage:

```
sequencer.loadImage('PATH')
    .addSteps('colorbar',options)
    .run()
```

where options is an object with the following properties:

- colormap : Name of the Colormap(default, greyscale, stretched, fastie, brntogrn, blutoredjet, colors16; default: default)
- x : X-position of the image on which the new image is overlaid (default 0)
- y : Y-position of the image on which the new image is overlaid (default 0)
- h : height of resulting cropped image (default : 50% of input image width)

Could not get colorbar to work



Colormap-module:

This module is used for mapping brightness values (average of red, green & blue) to a given color lookup table, made up of a set of one more color gradients.

Usage:

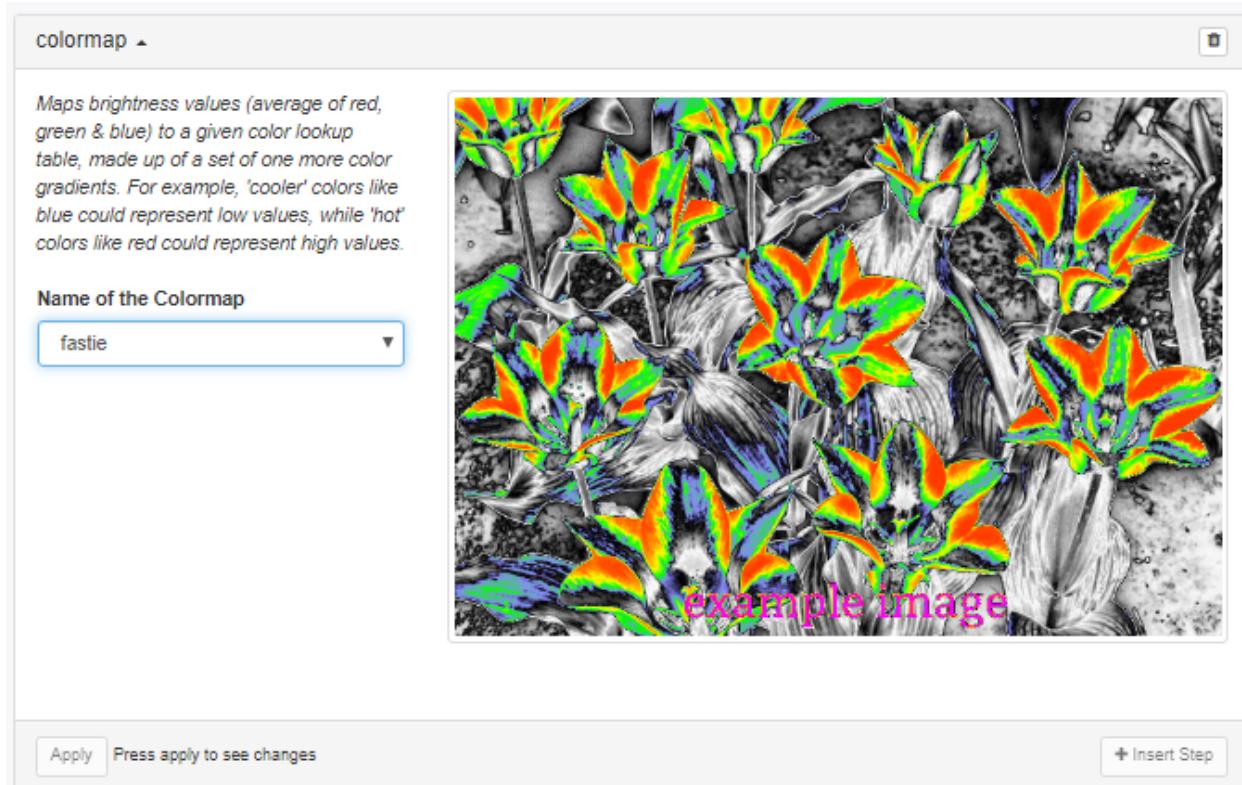
```
sequencer.loadImage('PATH')
    .addSteps('colormap',options)
    .run()
```

where options is an object with the following property:

- colormap : Name of the Colormap (greyscale, stretched, fastie, brntogrn, blutoredjet, colors16)

Image Sequencer colormaps with applications - Jan 2019

Colormap	Application	Reference
default/blutored 	Produces an all color heatmap with values below zero colored blue or green. Plants should be green to orange. Based on a continuous look up table from ImageJ called Physics.	https://publiclab.org/wiki/ndvi-gradients
greyscale 	General Purpose. Applicable when no color display is available.	
bluwhtgrngis 	General purpose extended green colormap. Obtained from the open source geospatial wiki.	https://grasswiki.osgeo.org/wiki/File:Colorable_ndvi.png
stretched 	A variant of the physics Image J colormap. Distinguishes NDVI values below zero by making them blue, but retains the gradient so details of the non-plant scene can be discerned. The magenta for values ~0.9 and higher allows troubleshooting of high values. Plants should be greenish yellow to red.	https://publiclab.org/wiki/ndvi-gradients
fastie 	<i>Custom NDVI colormap suitable for (some) color blind users. Left half of the colormap has three gradients between black and white designed to recognize objects and textures that are not foliage. Narrow band of violet between 0.1 and 0.2 which could represent very low photosynthetic activity, but might also be noise or error. Primary gradient of photosynthetic activity (from 0.2 to 0.9) represents a heat map from green to yellow to red. Highest values (> 0.9) are colored magenta. Foliage generally does not have NDVI values this high, so this color represents non-plants.</i>	https://publiclab.org/notes/cfastie/08-26-2014/new-ndvi-colormap
brwtogrn 	Colormap obtained from NASA website to show plants under stress (drought).	https://earthobservatory.nasa.gov/images/50328/drought-in-texas
blutoredjet 	Similar to matplotlib and opencv 'Jet' colormap. Useful for comparing image processing routines from other image processing/analysis applications.	https://publiclab.org/wiki/ndvi-gradients
colors16 	A discrete class look up table from ImageJ. Produces a heatmap with the highest values (which should not be plants) colored dull pink or white. Good for troubleshooting. Plants should be yellow to red.	https://publiclab.org/wiki/ndvi-gradients



Contrast-module:

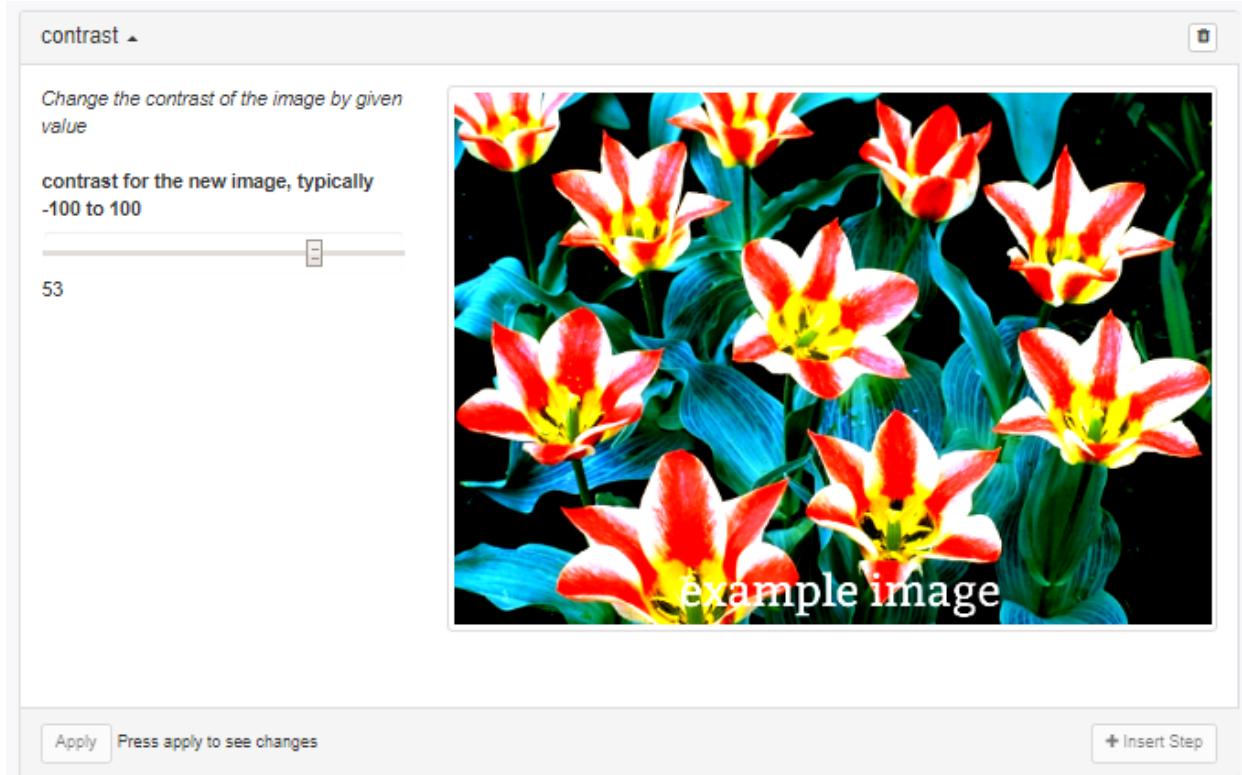
This module is used for changing the contrast of the image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('contrast',options)
    .run()
```

where options is an object with the following property:

- contrast : contrast for the given image (-100 to 100; default : 70)



Convolution-module:

This module is used for performing image-convolution.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('convolution',options)
    .run()
```

where options is an object with the following properties:

- constantFactor : a constant factor, multiplies all the kernel values by that factor (default : 1/9)
- kernelValues : nine space separated numbers representing the kernel values in left to right and top to bottom format(default : 1 1 1 1 1 1 1 1 1)

convolution ▾

Image Convolution using a given 3x3 kernel matrix [Read more](#)

a constant factor, multiplies all the kernel values by that factor

nine space separated numbers representing the kernel values in left to right and top to bottom format.

Apply Press apply to see changes

+ Insert Step

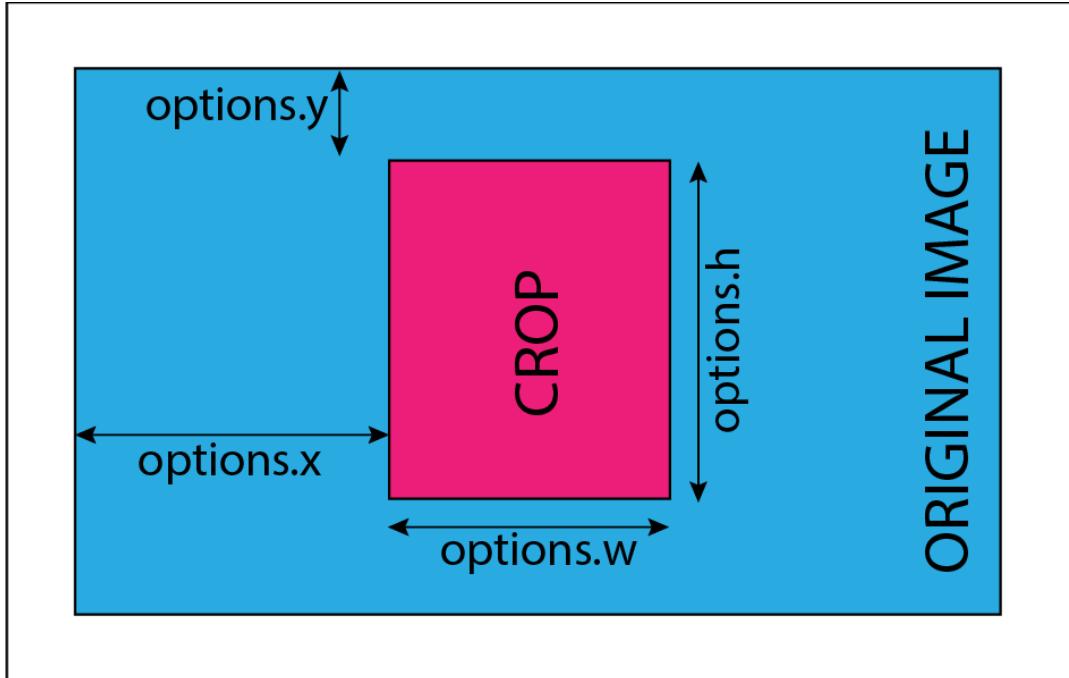
Crop-module:

This module is used to crop an image.

Usage:

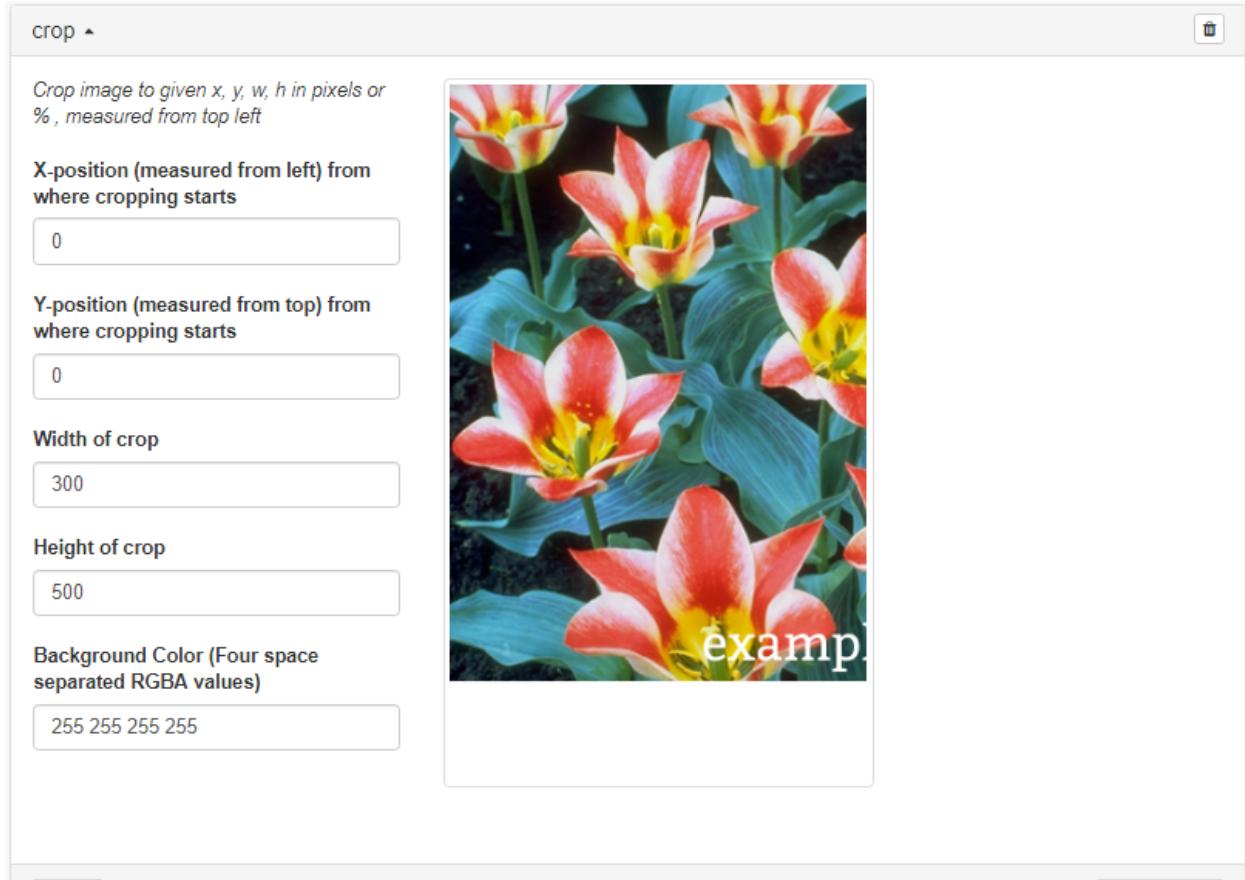
```
sequencer.loadImage('PATH')
    .addSteps('crop',options)
    .run();
```

Where options is an object having the properties x, y, w, h. This diagram defines these properties:



Defaults:

- options.x : 0
- options.y : 0
- options.w : half of image width
- options.h : half of image height



DecodeQr-module:

This module is used for decoding a QR in image (if present).

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('decode-qr',options)
    .run()
```

Detect Edges:

This module detects edges using the Canny method, which first Gaussian blurs the image to reduce noise (amount of blur configurable in settings as `options.blur`), then applies a number of steps to highlight edges, resulting in a greyscale image where the brighter the pixel, the stronger the detected edge

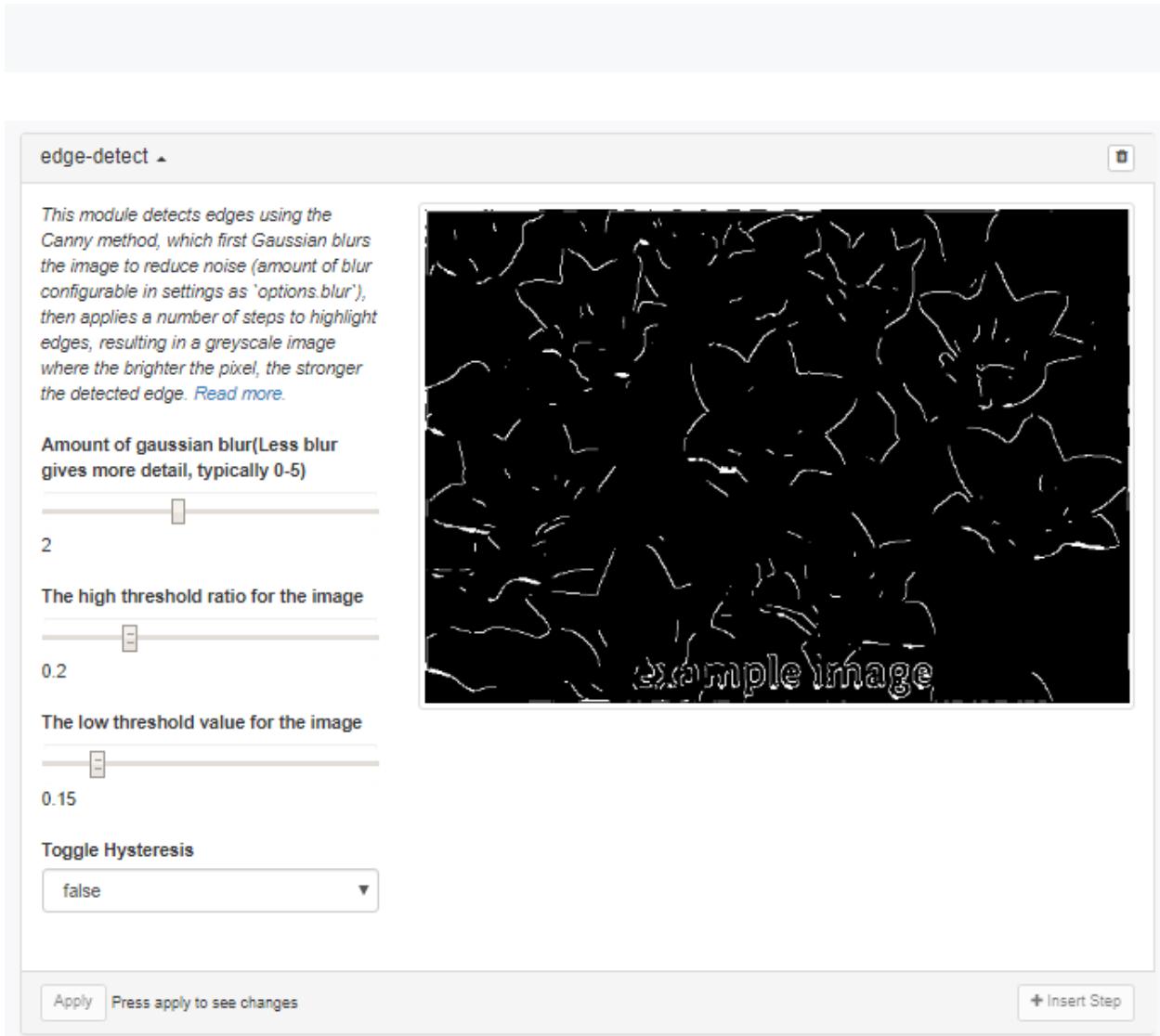
This module is used for detecting images.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('edge-detect',options)
    .run()
```

where options is an object with the following properties:

- blur : Intensity of Gaussian blur (0 to 5; default 2)
- highThresholdRatio : Upper Threshold Ratio (default : 0.2)
- lowThresholdratio : Lower Threshold Ratio (default : 0.2)



Dither-module:

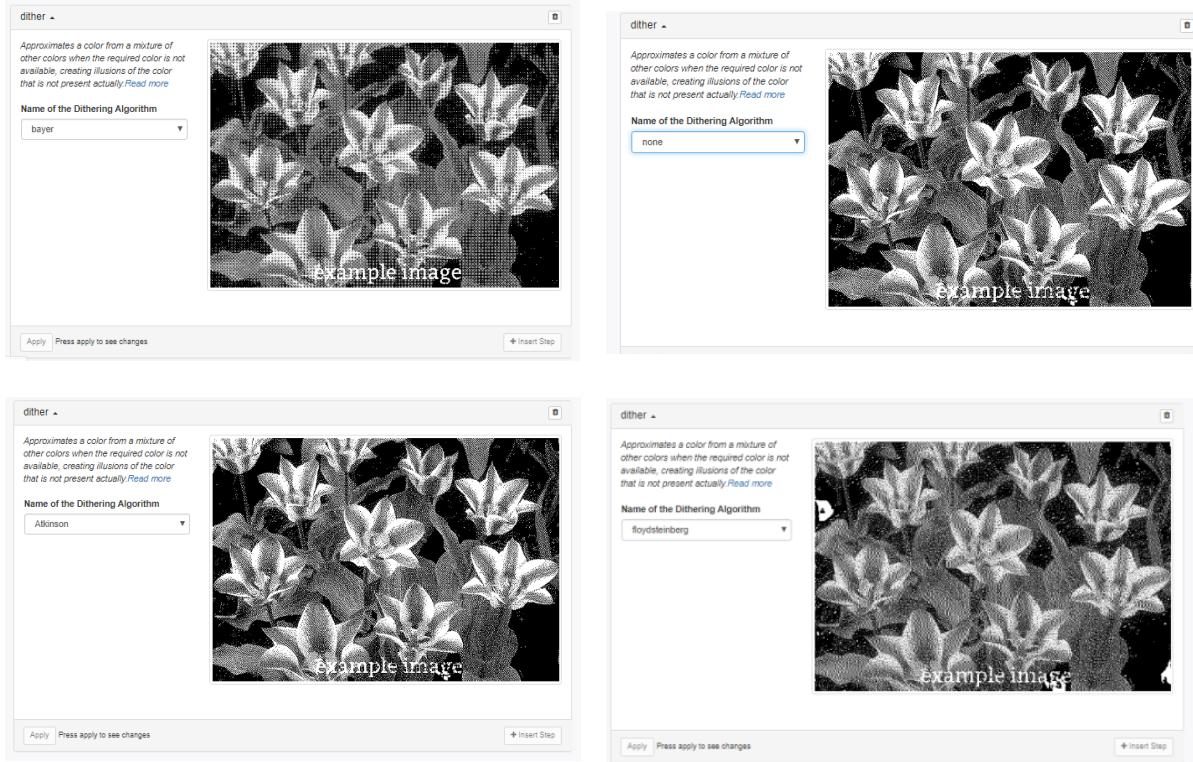
This module approximates a color from a mixture of other colors when the required color is not available, creating illusions of the color that is not present actually. (Not sure what this means)

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('dither',options)
    .run()
```

where options is an object with the following property:

- dither : Can select the name of the Dithering Algorithm(default none)



Draw-rectangle-module:

This module helps to draw a rectangle on the image with a starting and ending corner with the specified thickness and color of the border.

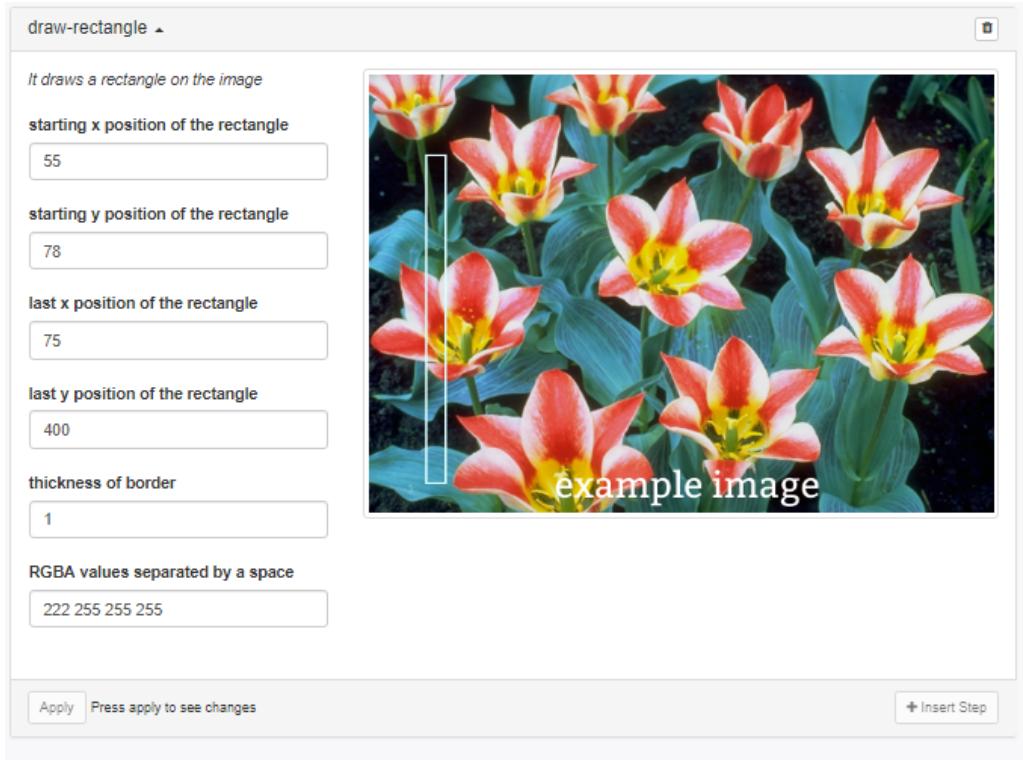
Usage:

```
sequencer.loadImage('PATH')
    .addSteps('draw-rectangle',options)
    .run()
```

where options is an object with the following properties:

- startingX : starting x position of the rectangle (default 0)
- startingY : starting y position of the rectangle (default 0)
- endX : last x position of the rectangle (default "width")
- endY : last y position of the rectangle (default "height")
- thickness : thickness of the border (default 1)

- color : RGBA values separated by a space (default "0 0 0 255")



Dynamic-module:

This module is used for producing each color channel based on the original image's color.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('dynamic',options)
    .run()
```

where options is an object with the following properties:

- red : expression for red channel (R, G, B and A as inputs; default r)
- green : expression for green channel (R, G, B and A as inputs; default g)
- blue : expression for blue channel (R, G, B and A as inputs; default b)
- monochrome: fallback for other channels if none provided (default : r+g+b/3)

dynamic ▾

A module which accepts JavaScript math expressions to produce each color channel based on the original image's color. See [Infragrammar](#).

Expression to return for red channel with R, G, B, and A inputs

Expression to return for green channel with R, G, B, and A inputs

Expression to return for blue channel with R, G, B, and A inputs

Expression to return with R, G, B, and A inputs; fallback for other channels if none provided

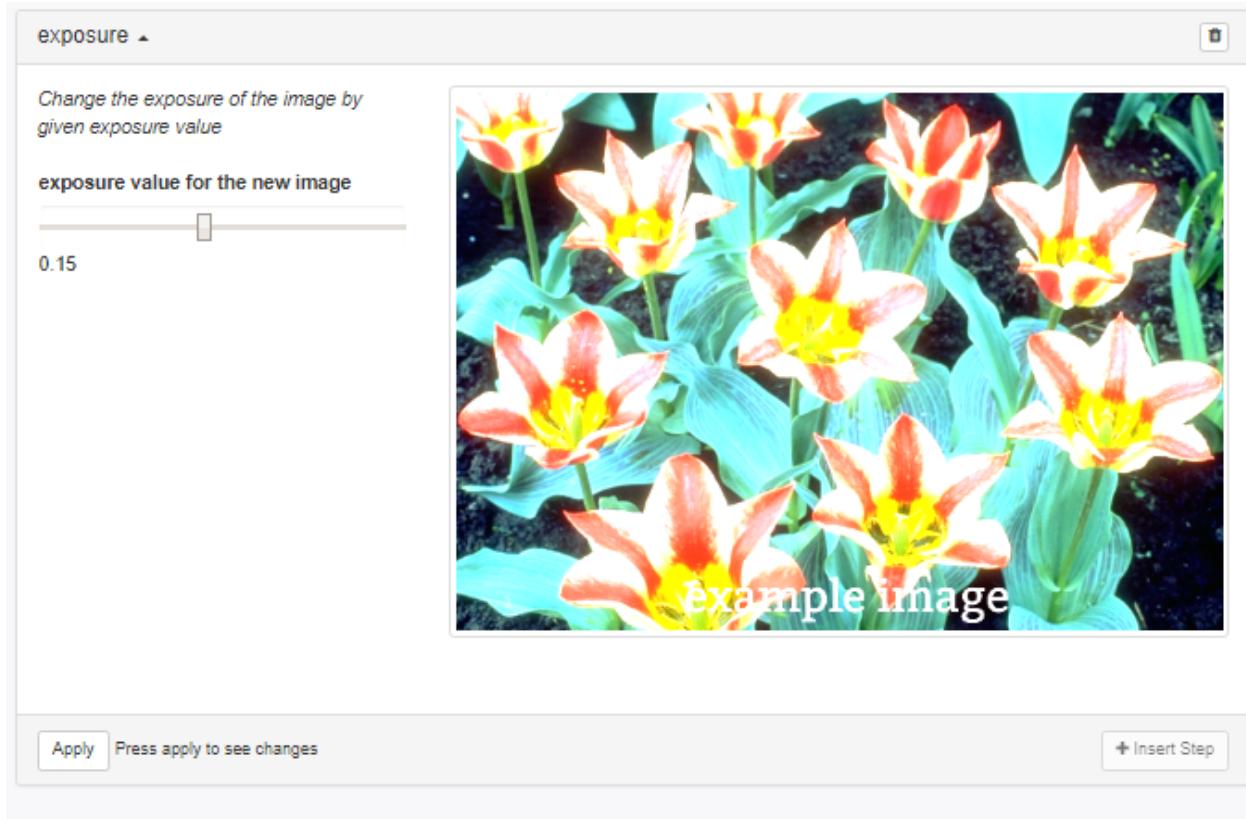
example image

Apply Press apply to see changes



Is it $r+g+b$ or $r+g+b/3$?

Exposure:



FisheyeGI-module:

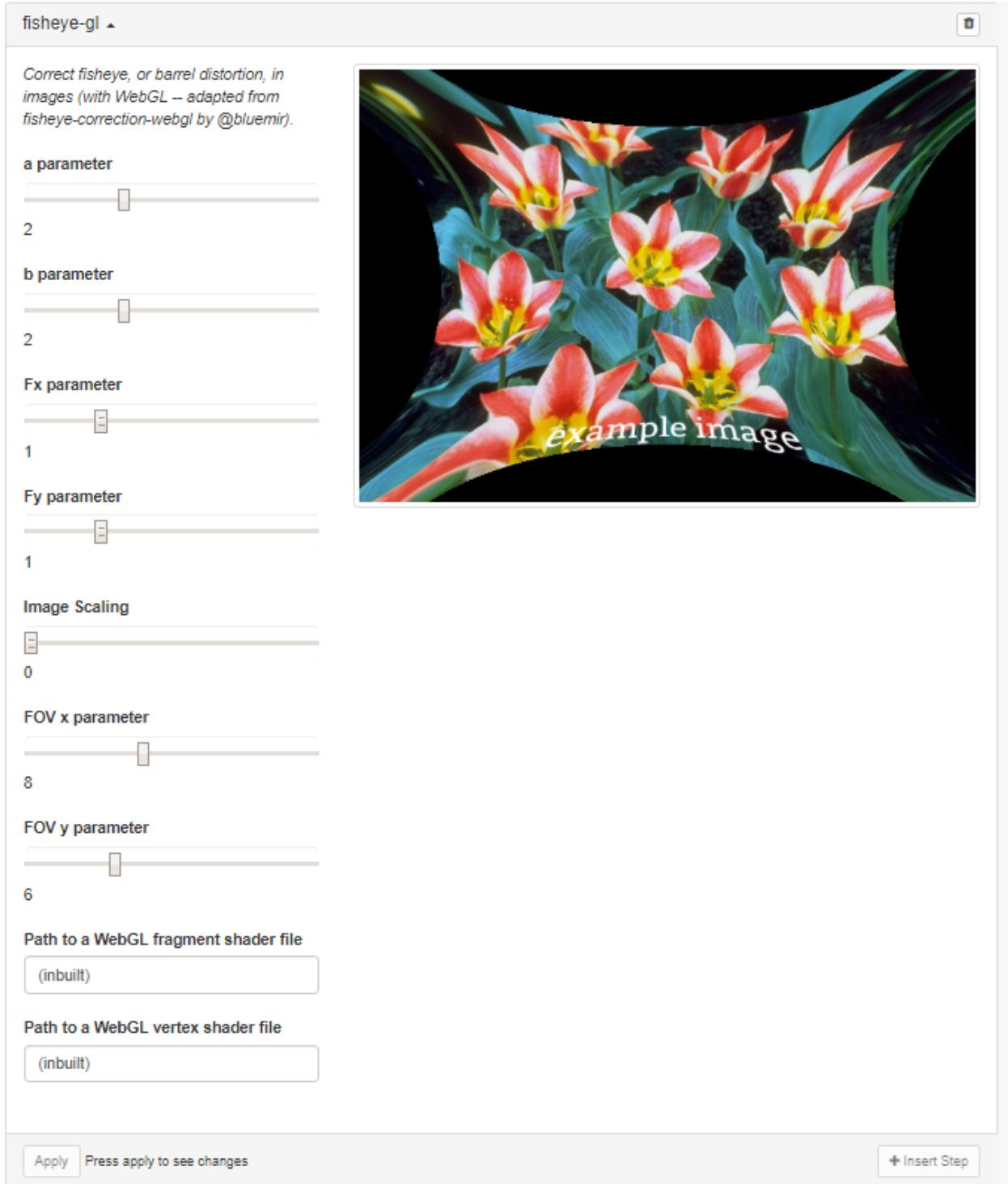
This module is used for correcting Fisheye or Lens Distortion

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('fisheye-gi',options)
    .run()
```

where options is an object with the following properties:

- a : a correction (0 to 4; default 1)
- b : b correction (0 to 4; default 1)
- Fx : x correction (0 to 4; default 1)
- Fy : y correction (0 to 4; default 1)
- scale : The ratio to which the original image is to be scaled (0 to 20; default 1.5)
- x : Field of View x (0 to 2; default 1)
- y : Field of View y (0 to 2; default 1)



Flipimage-module:

This module is used for flipping the image on the selected axis.

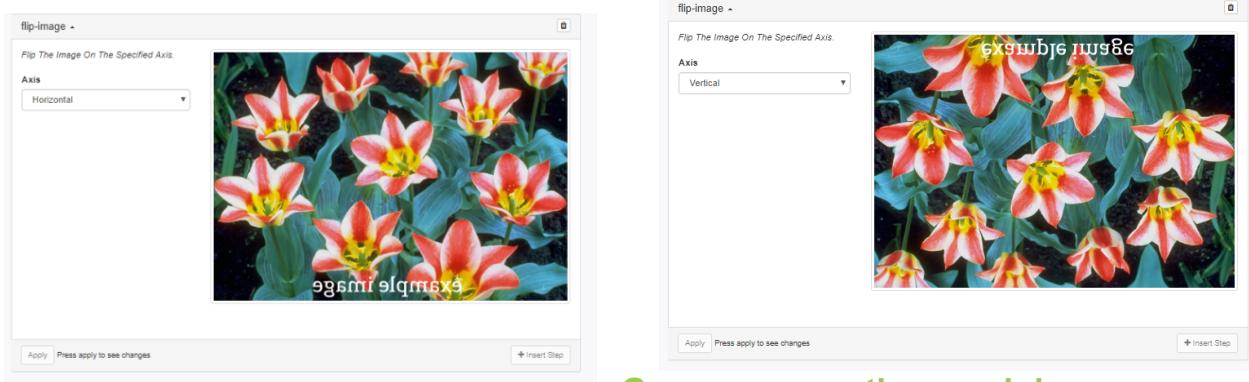
Usage:

```
sequencer.loadImage('PATH')
    .addSteps('flip-image',options)
```

```
.run()
```

where options is an object with the following properties:

- Axis : select the required axis (default : vertical)



Gamma-correction-module:

This module is used for applying gamma correction.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('gamma-correction',options)
    .run()
```

where options is an object with the following property:

- adjustment : Inverse of actual gamma factor (default 0.2)

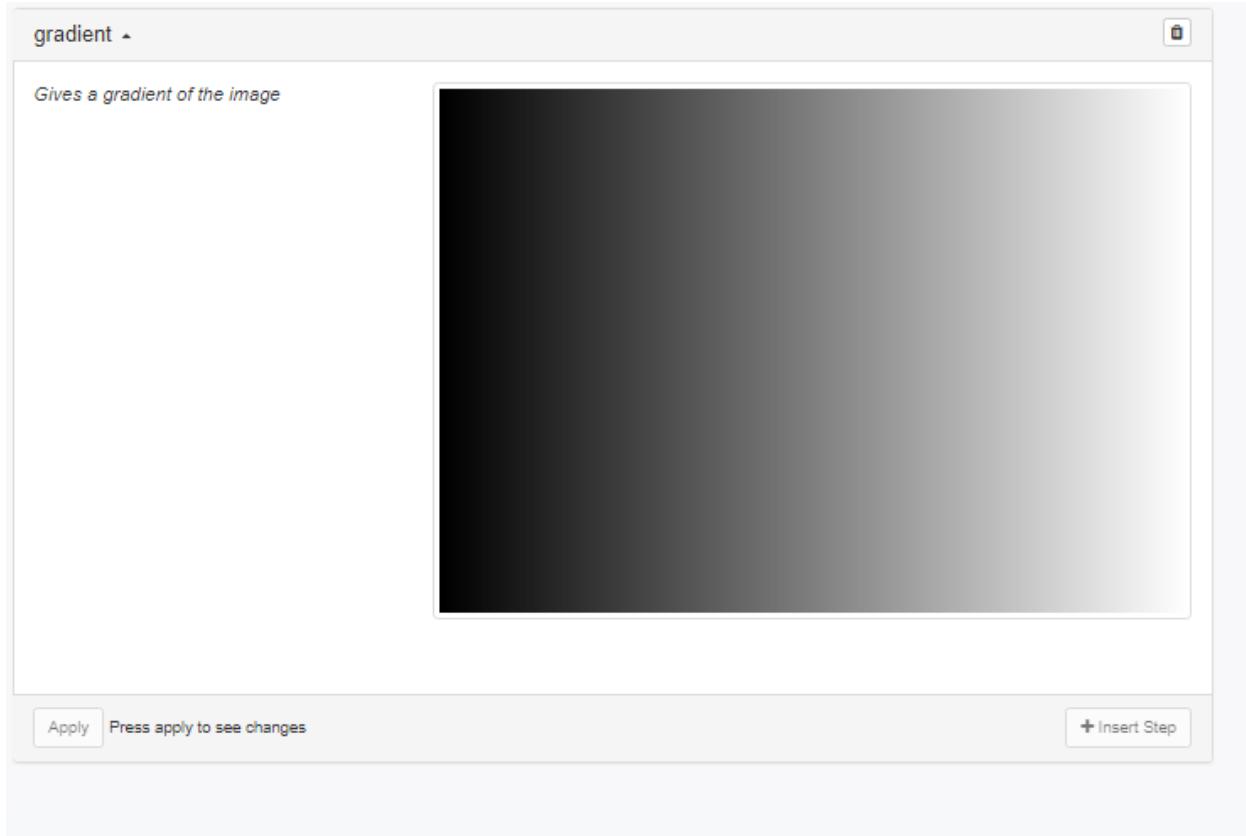
(slider does not work)

Gradient-module:

This module is used for finding gradient of the image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('gradient',options)
    .run()
```



Grid Overlay:

This adds the grid over an image.

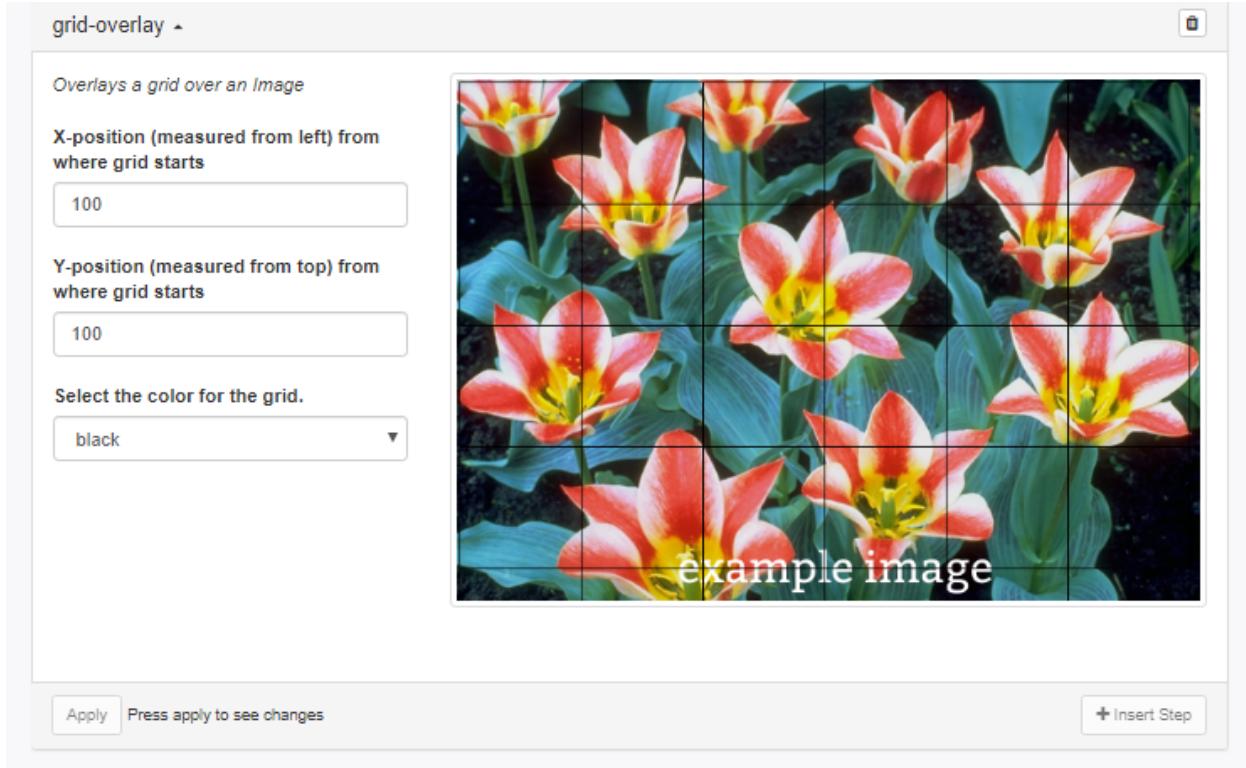
Usage:

```
sequencer.loadImage('PATH')
    .addSteps('grid-overlay',options)
    .run()
```

where options is an object with the following property:

- options.x : The value at which the grid line should start in x-axis.
- options.y : The value at which the grid line should start in y-axis.
- color : Color for the grid on the image.

(only works once)



Histogram-module:

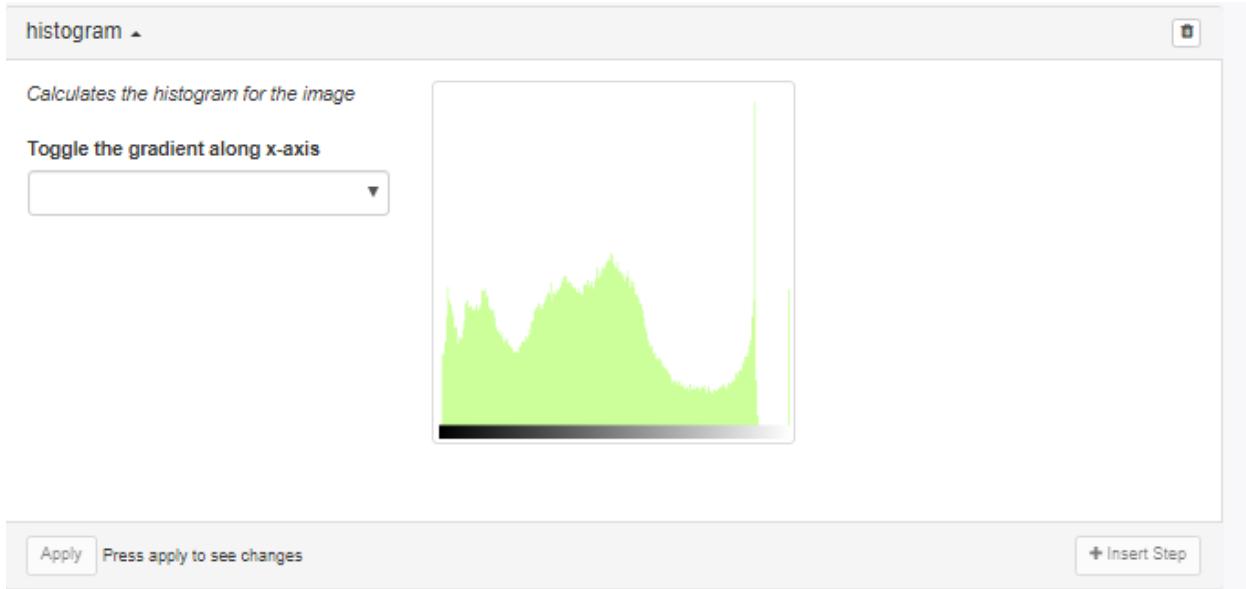
This module is used for calculating histogram of the image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('histogram',options)
    .run()
```

where options is an object with the following property:

- gradient : boolean value used to toggle gradient along x-axis (true or false; default true)



Import-image-module:

This module is used for importing a new image and replacing the original with it.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('import-image',options)
    .run()
```

where options is an object with the following property:

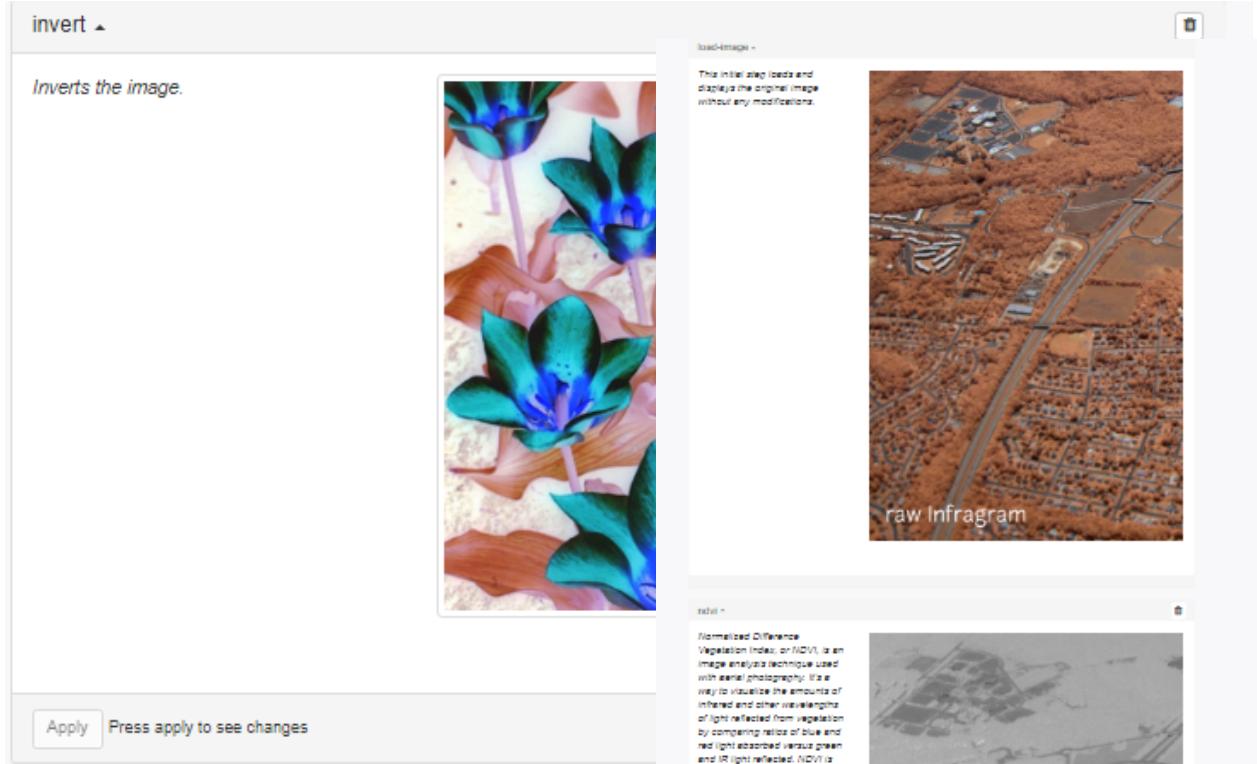
- url : url of the new image (local image url or data url; default : "./images/monarch.png")

Invert-module:

This module is used for inverting the image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('invert',options)
    .run()
```



NDVI-module:

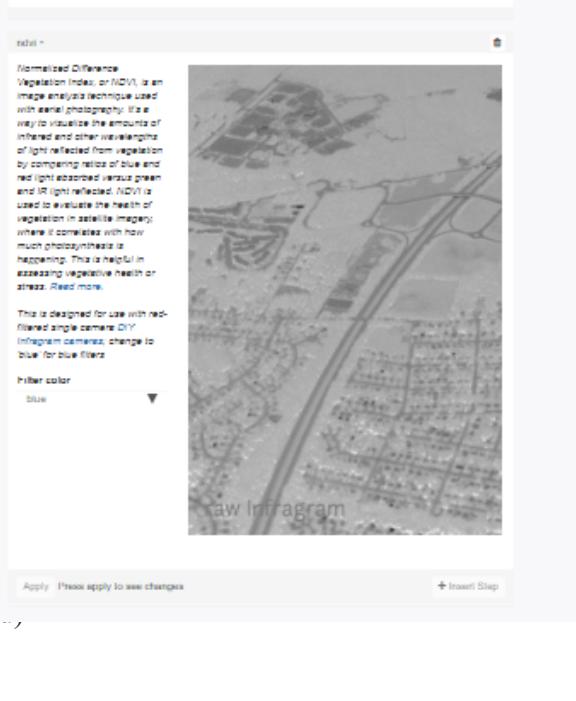
This module is used for applying ndvi technique to

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('ndvi',options)
    .run()
```

where options is an object with the following properties:

- filter : filter for NDVI (blue or red; default blue)



NDVI-colormap-module:

This module is used for demonstrating ndvi and colormap properties consecutively.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('ndvi-colormap',options)
    .run()
```



Overlay-module:

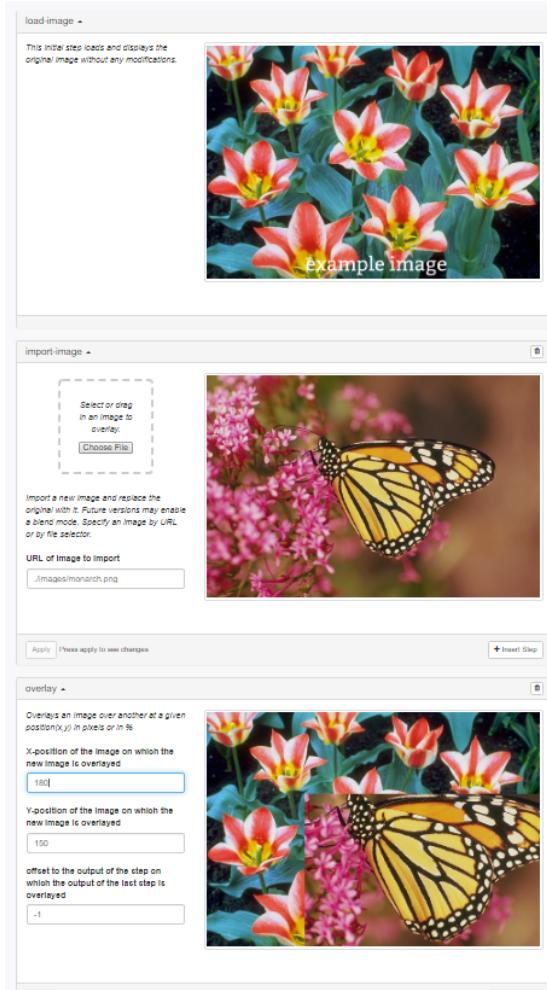
This module is used for overlaying an Image over another .

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('overlay',options)
    .run()
```

where options is an object with the following properties:

- x : X-position of the image on which the new image is overlaid (default 0)
- y : Y-position of the image on which the new image is overlaid (default 0)
- offset : offset to the step on which the output of the last step is overlaid (default -2)



Paint-bucket-module:

This module fills any polygonal shape with the specified color in pixels.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('paint-bucket',options)
    .run()
```

where options is an object with the following property:

- startingX : it is the value of the starting x-coordinate (default 10)
- startingY : it is the value of the starting y-coordinate (default 10)
- fillColor : they are four spaced separated numbers representing the RGBA values of fill-color (default "100 100 100 255")
- tolerance : it is the % tolerance (default 10)

(does this work, all images seem the same)

Resize-module:

This module is used for resizing an image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('resize',options)
    .run()
```

where options is an object with the following property:

- resize : Percentage value of resize (default 125%)

This initial step loads and displays the original image without any modifications.



resize ▾

Resize image by given percentage value

Percentage value of the resize

750|



Apply Press apply to see changes

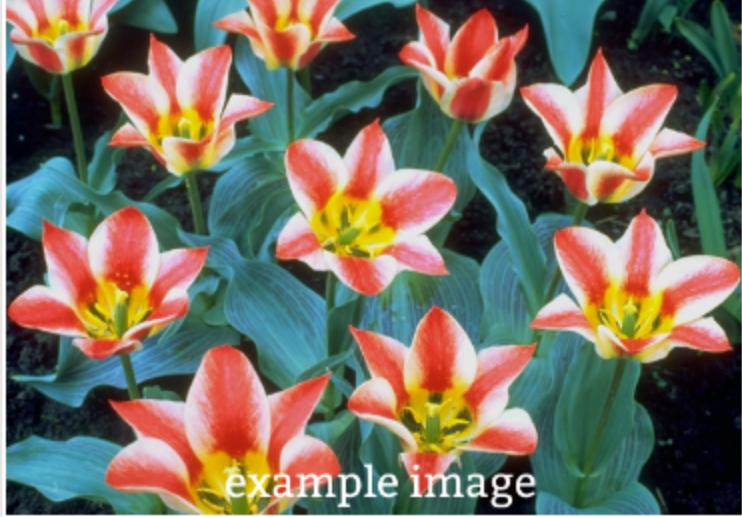
+ Insert Step

(is this off by 100?)

Resize-canvas:

load-image *

This initial step loads and displays the original image without any modifications.



example image

canvas-resize *

This module resizes the canvas and overlays the output of the previous step at given location

Final width of the canvas

1000

Final height of the canvas

1000

X-cord of the top left corner of the image on the canvas

500

Y-cord of the top left corner of the image on the canvas

500



example image

Replacecolor-module:

This module is used in replacing the color with grey or any desired color.

Usage

```
sequencer.loadImage('PATH')
    .addSteps('replace-color',options)
    .run()
```

where options is an object with the following properties:

- replaceMethod : replaces with the desired color (default greyscale)
- replaceColor : three space separated numbers representing the RGB values of color to be filled (default "0 0 255")
- color : three space separated numbers representing the RGB values of color to be replaced (default "228 86 81")
- tolerance : it is the % tolerance (default 50)

load-image *

This initial step loads and displays the original image without any modifications.



replace-color *

Replace color with grey or your desired color

Replace Method

grayscale

three space separated numbers representing the RGB values of color to be filled

0 0 255

three space separated numbers representing the RGB values of color to be replaced

228 88 81

% tolerance

50



Apply Press apply to see changes

+ Insert Step

Rotate-module:

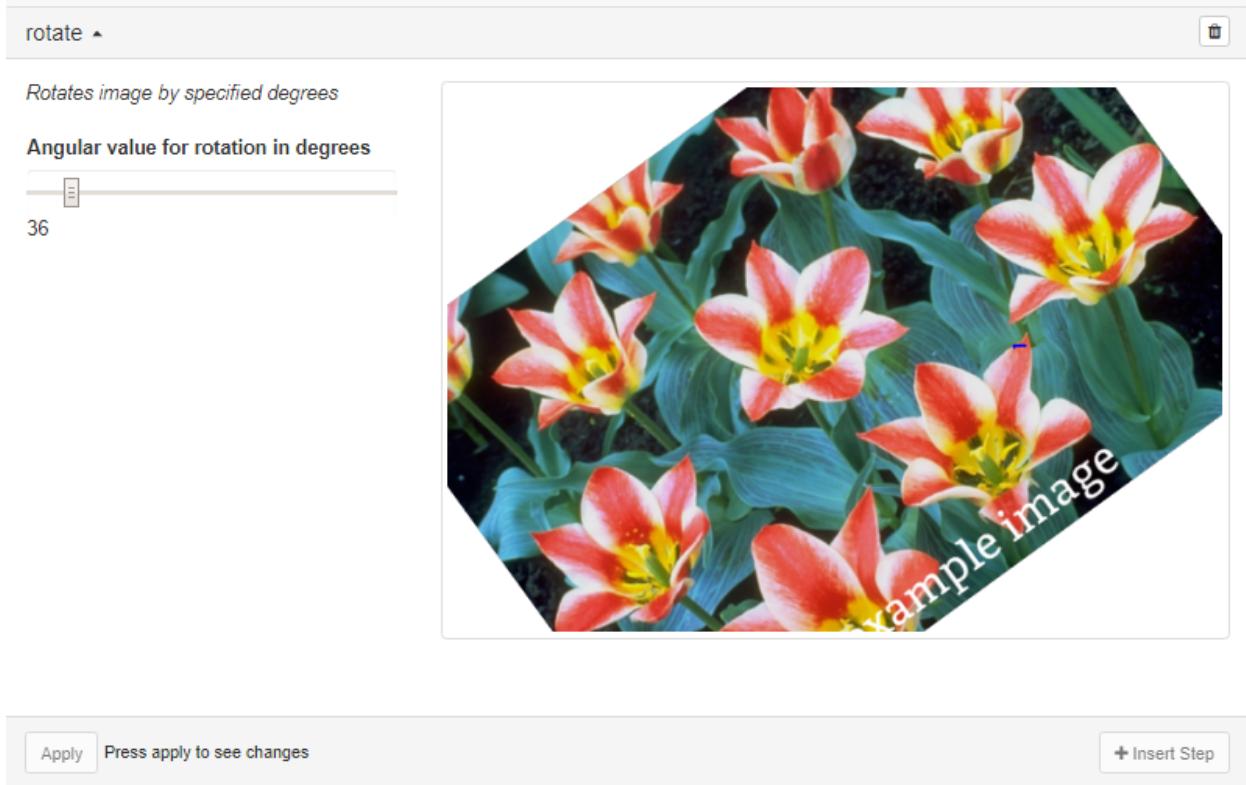
This module is used for rotating an image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('rotate',options)
    .run()
```

where options is an object with the following property:

- rotate : angular value for rotation in degrees (between 0 and 360; default 0)



Saturation-module:

This module is used for changing the saturation of the image.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('saturation',options)
    .run()
```

where options is an object with the following property:

- saturation : saturation for the new image (between 0 and 2; default 0)

load-image ▾

This initial step loads and displays the original image without any modifications.

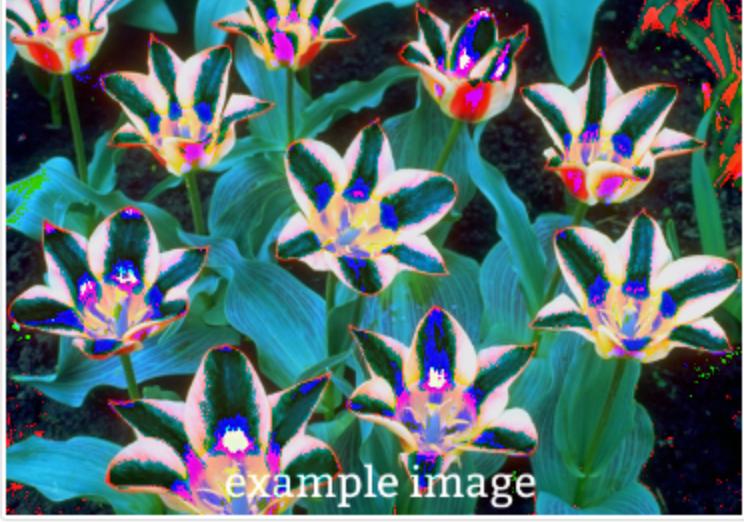


example image

saturation ▾

Change the saturation of the image by given value, from 0-1, with 1 being 100% saturated.

saturation for the new image between 0 and 2, 0 being black and white and 2 being highly saturated



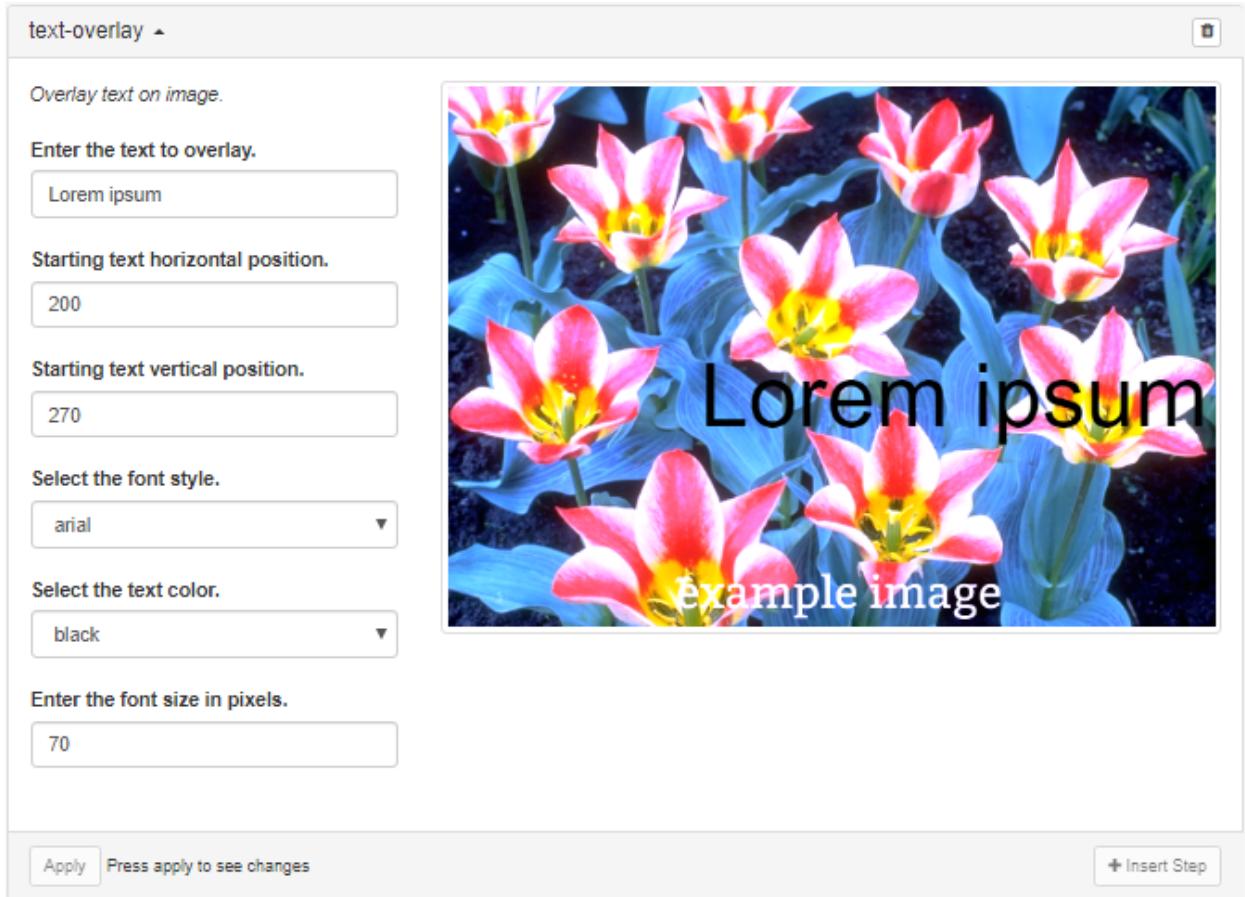
1.6

example image

Apply Choose sample to use elsewhere

Import Slides

Text-overlay:



Threshold:

Thresholding is used to create binary images.

Usage:

```
sequencer.loadImage('PATH')
    .addSteps('threshold',options)
    .run()
```

where options is an object with the following property:

- threshold : it allows to select the types of thresholding (default "manual thresholding")
- input : it allows to select the range of thresholding value (default 120)

load-image

This initial step loads and displays the original image without any modifications.



example image

threshold

Thresholding is used to create binary images

Type of Thresholding

Automatic Thresholding

Threshold Value

100



example image

Apply Press apply to see changes + Insert Step

Tint:

It adds color tint to an image.

Usage:

```
sequencer.loadImage('PATH')
```

```
.addSteps('tint',options)
.run()
```

where options is an object with the following property:

- color : RGB values separated by a space (default "0 0 255")
- factor : amount of tint (default 0.5)

load-image
▼

This initial step loads and displays the original image without any modifications.



example image

tint
▼

Add color tint to an image

RGB values separated by a space

Amount of tint



example image

Apply
Press apply to see changes
+ Insert Step

White Balance:

white-balance ▾

Render neutral colours correctly based on the whitest pixel in the image.

Red component of the whitest pixel

Green component of the whitest pixel

Blue component of the whitest pixel



example image

Apply Press apply to see changes

+ Insert Step

Applications:

This section demonstrates how Image Sequencer modules can be applied for different applications.

NDVI (NIR and Red) with colorbar:

