Tensor Constructors

The following functions construct Tensors like Gaussian or Laplacian kernels, or images like Lenna and Fabio.

[res] image.lena()

Returns the classic `Lenna.jpg` image as a `3 x 512 x 512` Tensor.

[res] image.fabio()

Returns the `fabio.jpg` image as a `257 x 271` Tensor.

[res] image.gaussian([size, sigma, amplitude, normalize, [...]])

Returns a 2D Gaussian
kernel of size `height x width`. When used as a Gaussian smoothing operator in a 2D convolution, this kernel is used to `blur` images and remove detail and noise (ref.: Gaussian Smoothing).
Optional arguments `[...]` expand to
`width`, `height`, `sigma_horz`, `sigma_vert`, `mean_horz`, `mean_vert` and `tensor`.

The default value of `height` and `width` is `size`, where the latter has a default value of 3. The amplitude of the Gaussian (its maximum value) is `amplitude`. The default is 1.
When `normalize=true`, the kernel is normalized to have a sum of 1.
This overrides the `amplitude` argument. The default is `false`.
The default value of the horizontal and vertical standard deviation `sigma_horz` and `sigma_vert` of the Gaussian kernel is `sigma`, where the latter has a default value of 0.25. The default values for the corresponding means `mean_horz` and `mean_vert` are 0.5. Both the standard deviations and means are relative to kernels of unit width and height where the top-left corner is the origin. In other works, a mean of 0.5 is the center of the kernel size, while a standard deviation of 0.25 is a quarter

of it. When `tensor` is provided (a 2D Tensor), the `height`, `width` and `size` are ignored. It is used to store the returned gaussian kernel.

Note that arguments can also be specified as key-value arguments (in a table).

[res] image.gaussian1D([size, sigma, amplitude, normalize, mean, tensor])

Returns a 1D Gaussian kernel of size `size`, mean `mean` and standard deviation `sigma`.
Respectively, these arguments have default values of 3, 0.25 and 0.5.
The amplitude of the Gaussian (its maximum value) is `amplitude`. The default is 1.
When `normalize=true`, the kernel is normalized to have a sum of 1.
This overrides the `amplitude` argument. The default is `false`. Both the standard deviation and mean are relative to a kernel of unit size.
In other works, a mean of 0.5 is the center of the kernel size,
while a standard deviation of 0.25 is a quarter of it.
When `tensor` is provided (a 1D Tensor), the `size` is ignored.
It is used to store the returned gaussian kernel.

Note that arguments can also be specified as key-value arguments (in a table).

[res] image.laplacian([size, sigma, amplitude, normalize, [...]])

Returns a 2D Laplacian
kernel of size `height x width`.
When used in a 2D convolution, the Laplacian of an image highlights
regions of rapid intensity change and is therefore often used for edge detection
(ref.: Laplacian/Laplacian of Gaussian).
Optional arguments `[...]` expand to
`width`, `height`, `sigma_horz`, `sigma_vert`, `mean_horz`, `mean_vert`.

The default value of `height` and `width` is `size`, where the latter
has a default value of 3. The amplitude of the Laplacian (its maximum value)
is `amplitude`. The default is 1.
When `normalize=true`, the kernel is normalized to have a sum of 1.
This overrides the `amplitude` argument. The default is `false`.

The default value of the horizontal and vertical standard deviation `sigma_horz` and `sigma_vert` of the Laplacian kernel is `sigma`, where the latter has a default value of 0.25. The default values for the corresponding means `mean_horz` and `mean_vert` are 0.5. Both the standard deviations and means are relative to kernels of unit width and height where the top-left corner is the origin. In other works, a mean of 0.5 is the center of the kernel size, while a standard deviation of 0.25 is a quarter of it.

[res] image.colormap(nColor)

Creates an optimally-spaced RGB color mapping of `nColor` colors.
Note that the mapping is obtained by generating the colors around the HSV wheel, varying the Hue component.
The returned `res` Tensor has size `nColor x 3`.

[res] image.jetColormap(nColor)

Creates a jet (blue to red) RGB color mapping of `nColor` colors.
The returned `res` Tensor has size `nColor x 3`.