[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

PROCESSING **KERNEL** [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)

A kernel is an Earth Engine parameter object that represents a “neighborhood,” a set of pixels that lie within a specified vicinity of any given pixel and whose values

can be used to characterize that pixel. Neighborhoods can be processed by using operations of the types listed below, which vary according to the nature of that processing. Each operation name is linked to a separate page describing that operation.

**CREATING** KERNELS

TO **MEASURE GRADIENTS**  [ee.Kernel.roberts](#roberts) [ee.Kernel.prewitt](#prewitt) [ee.Kernel.sobel](#sobel)

[ee.Kernel.compass](#compass) [ee.Kernel.kirsch](#kirsch)

TO **MEASURE INFLECTION** [ee.Kernel.laplacian4](#laplacian4) [ee.Kernel.laplacian8](#laplacian8)

TO **SMOOTH TRANSITIONS** [ee.Kernel.euclidean](#euclidean) [ee.Kernel.gaussian](#gaussian)

[ee.Kernel.manhattan](#gaussian) [ee.Kernel.chebyshev](#chebyshev)

OF **PARTICULAR SHAPES** [ee.Kernel.circle](#circle) [ee.Kernel.octagon](#octagon) [ee.Kernel.square](#square)

[ee.Kernel.diamond](#diamond) [ee.Kernel.cross](#cross) [ee.Kernel.plus](#plus)

[ee.Kernel.fixed](#fixed)

**TRANSFORMING** KERNELS [kernel.rotate](#rotate) [kernel.add](#add)

**DOCUMENTING** KERNELS kernel.getInfo ee.Algorithms.Describe(reducer)

kernel.toString reducer.serialize

**PRESENTING** KERNELS

IN **PRINT** print(kernel) console.log(kernel)

alert(kernel) confirm(kernel)

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE GRADIENTS**

ee.Kernel.roberts creates a new 3x3-pixel Roberts kernel with optionally specified settings. This kernel measures the first derivative

(i.e. the rate of change) in the value of a specified image within the immediate vicinity of each pixel by

- measuring the difference between the value of that pixel’s upper left neighbor minus that of its lower right neighbor,

- measuring the difference between the value of that pixel’s upper right neighbor minus that of its lower left neighbor, and

- computing the vector sum of those two directed quantities.

The first two of these three steps are respectively equivalent to convolution with the weights +1 0 0 and 0 0 + 1 0 0 0 0 0 0

0 0 -1 -1 0 0

newKernel = ee.Kernel.roberts( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

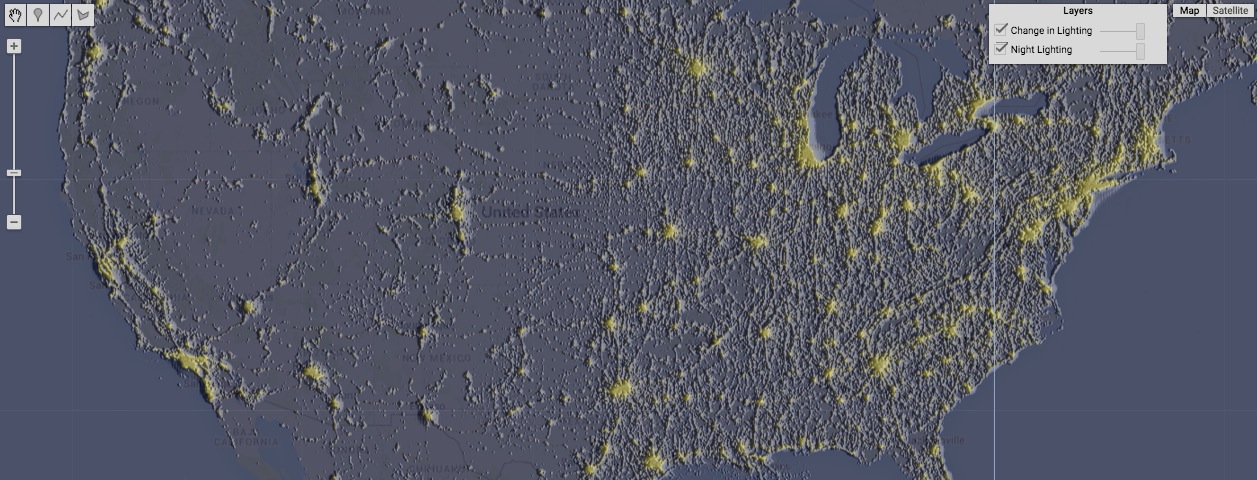
var TheKERNEL = ee.Kernel.roberts( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-1, max:1, opacity:0.5}, 'Change in Lighting');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE GRADIENTS**

ee.Kernel.prewitt creates a new 3x3-pixel Prewitt kernel with optionally specified settings. This kernel measures the first derivative

(i.e. the rate of change) in the value of a specified image within the immediate vicinity of each pixel by

- measuring the difference between the average of that pixel’s three lower neighbors minus its three upper neighbors,

- measuring the difference between the average of that pixel’s three right neighbors minus its left lower neighbors, and

- computing the vector sum of those two directed quantities.

The first two of these three steps are respectively equivalent to convolution with the weights -1 0 +1 and -1 -1 -1 -1 0 +1 0 0 0

-1 0 +1 +1 +1 +1

newKernel = ee.Kernel.prewitt( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

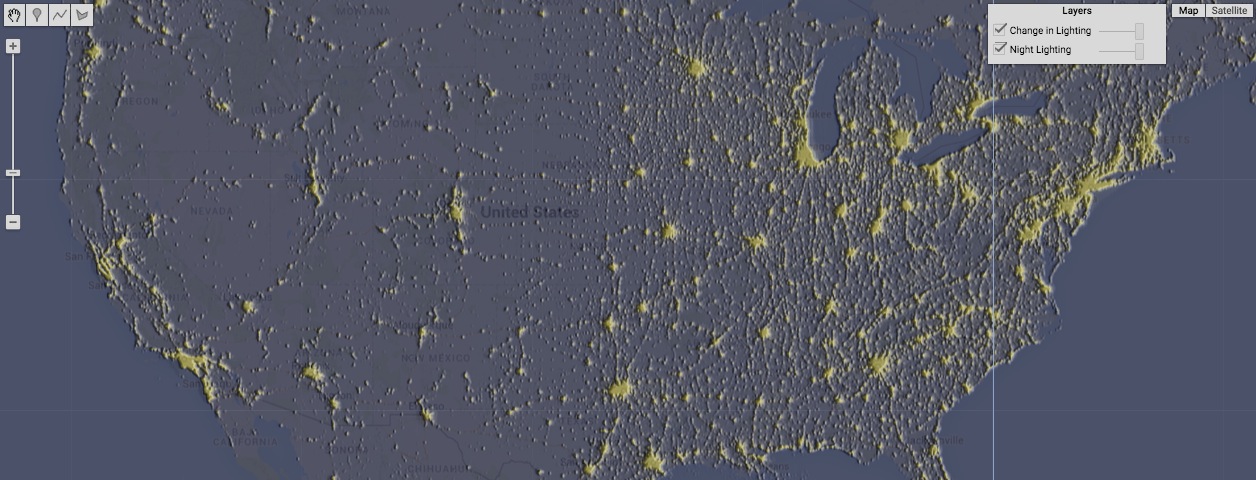
var TheKERNEL = ee.Kernel.prewitt( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-30, max:30, opacity:0.5}, 'Change in Lighting');

****

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE GRADIENTS**

ee.Kernel.sobel creates a new 3x3-pixel Sobel kernel with optionally specified settings. This kernel measures the first derivative

(i.e. the rate of change) in the value of a specified image within the immediate vicinity of each pixel by

- measuring the difference between the weighted average of that pixel’s three lower neighbors minus that of its three upper neighbors,

- measuring the difference between the weighted average of that pixel’s three right neighbors minus that of its left lower neighbors, and

- computing the vector sum of those two directed quantities.

The first two of these three steps are respectively equivalent to convolution with the weights -1 -2 -1 and -1 0 +1 0 0 0 -2 0 +2

+1 +2 +1 -1 0 +1

newKernel = ee.Kernel.sobel( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

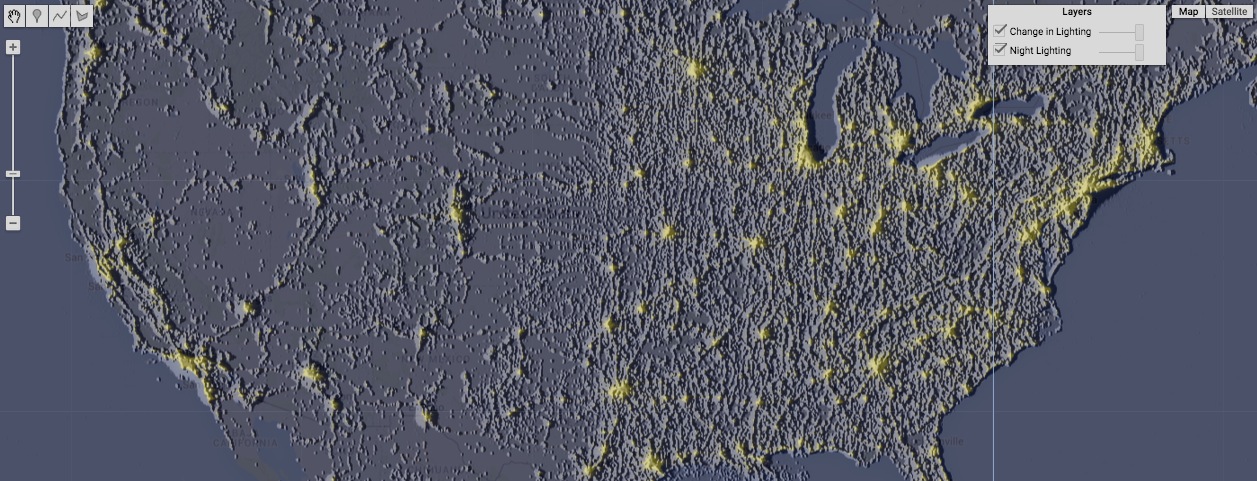
var TheKERNEL = ee.Kernel.sobel( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-1, max:1, opacity:0.5}, 'Change in Lighting');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE GRADIENTS**

ee.Kernel.compass creates a new 3x3-pixel Prewitts’ Compass kernel with optionally specified settings. This kernel measures the first derivative

(i.e. the rate of change) in the value of a specified image within the immediate vicinity of each pixel by computing the maximum of

- the difference between the average of that pixel’s five upper neighbors minus that of itself (x2) and its other three neighbors,

- the difference between the average of that pixel’s five upper right neighbors minus that of itself (x2) and its other three neighbors,

- the difference between the average of that pixel’s five right neighbors minus that of itself (x2) and its other three neighbors,

- the difference between the average of that pixel’s five lower right neighbors minus that of itself (x2) and its other three neighbors,

- the difference between the average of that pixel’s five lower neighbors minus that of itself (x2) and its other three neighbors,

- the difference between the average of that pixel’s five lower left neighbors minus that of itself (x2) and its other three neighbors,

- the difference between the average of that pixel’s five left neighbors minus that of itself (x2) and its other three neighbors,, and

- the difference between the average of that pixel’s five upper left neighbors minus that of itself (x2) and its other three neighbors.

The first two of these eight computations are respectively equivalent to convolution with the weights +1 +1 +1 and +1 +1 +1

+1 -2 +1 -1 -2 +1

-1 -1 -1 -1 -1 +1

newKernel = ee.Kernel.compass( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

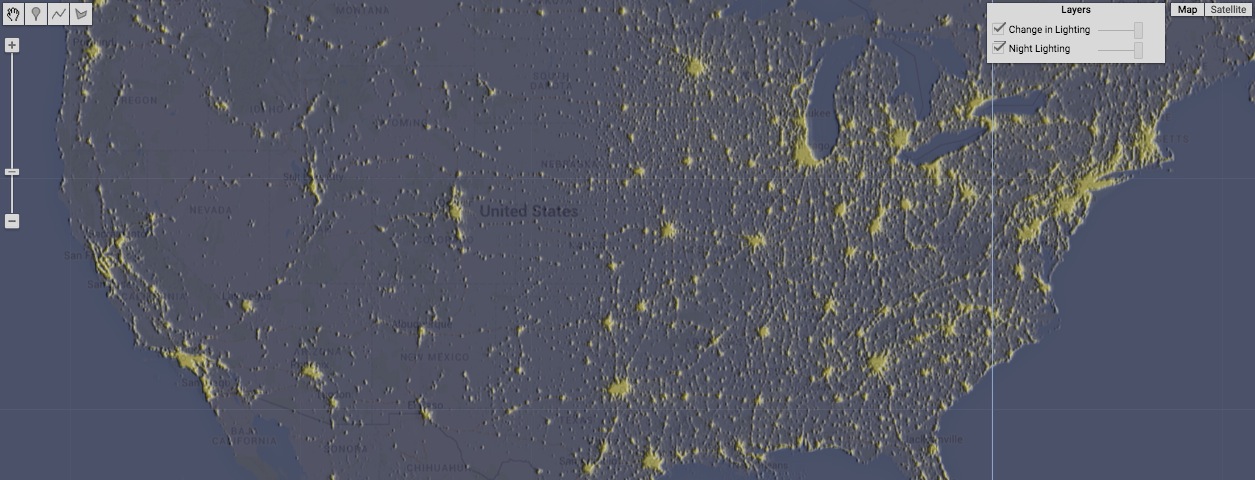
var TheKERNEL = ee.Kernel.compass( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-50, max:50, opacity:0.5}, 'Change in Lighting')



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE GRADIENTS**

ee.Kernel.kirsch creates a new 3x3-pixel Kirsch kernel with optionally specified settings. This kernel measures the first derivative

(i.e. the rate of change) in the value of a specified image within the immediate vicinity of each pixel by computing the maximum of

- the difference between the average of that pixel’s three upper neighbors minus that of its other five neighbors,

- the difference between the average of that pixel’s three upper right neighbors minus that of its other five neighbors,

- the difference between the average of that pixel’s three right neighbors minus that of its other five neighbors,

- the difference between the average of that pixel’s three lower right neighbors minus that of its other five neighbors,

- the difference between the average of that pixel’s three lower neighbors minus that of its other five neighbors,

- the difference between the average of that pixel’s three lower left neighbors minus that of its other five neighbors,

- the difference between the average of that pixel’s three left neighbors minus that of its other five neighbors, and

- the difference between the average of that pixel’s three upper left neighbors minus that of its other five neighbors.

The first two of these eight computations are respectively equivalent to convolution with the weights +5 +5 +5 and -3 +5 +5

-3 0 -3 -3 0 +5

-3 -3 -3 -3 -3 -3

newKernel = ee.Kernel.kirsch( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

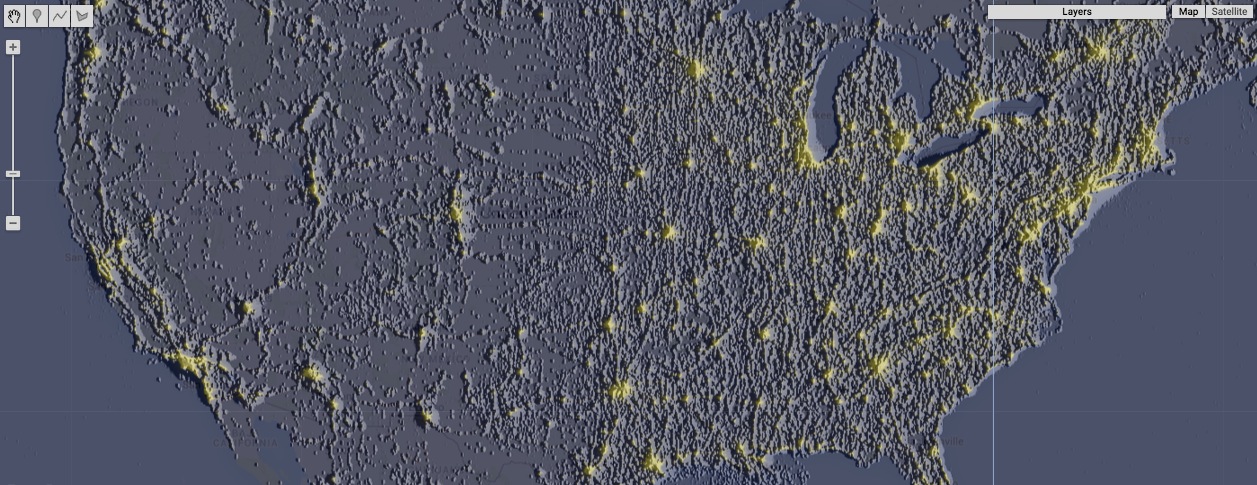
var TheKERNEL = ee.Kernel.kirsch( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-1, max:1, opacity:0.5}, 'Change in Lighting');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE INFLECTION**

ee.Kernel.laplacian4 creates a new 3x3-pixel Laplacian four-neighbor kernel with optionally specified settings. This kernel measures the second derivative

(i.e. the rate of change in the rate of change) in the value of a specified image within the immediate vicinity of each pixel by measuring

the difference between that pixel’s own value minus the average of its four laterally adjacent neighbors. This is equivalent

to convolution with the following weights

0 -1 0 -1 4 -1 0 -1 0

newKernel = ee.Kernel.laplacian4( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

var TheKERNEL = ee.Kernel.laplacian4( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-1, max:1, opacity:0.5}, 'Change in Lighting');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **MEASURE INFLECTION**

ee.Kernel.laplacian8 creates a new 3x3-pixel Laplacian eight-neighbor kernel with optionally specified settings. This kernel measures the second derivative

(i.e. the rate of change in the rate of change) in the value of a specified image within the immediate vicinity of each pixel by measuring

the difference between that pixel’s own value minus the average of its eight adjacent neighbors. This is equivalent to convolution with

the following weights -1 -1 -1

-1 8 -1

-1 -1 -1

newKernel = ee.Kernel.laplacian8( *magnitude, normalize?* )

A specified amount by which all input value are to be multiplied, given as a Float. Default 1.0

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new

kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

var TheKERNEL = ee.Kernel.laplacian8( );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-1, max:1, opacity:0.5}, 'Change in Lighting');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **SMOOTH TRANSITIONS**

ee.Kernel.euclidean creates a new Euclidean distance-weighted kernel of specified (square) dimensions and optionally settings. This kernel multiplies

(and then sums the resulting products of) the input values of all pixels by weights that decrease with direct distance from its center.

newKernel = ee.Kernel.euclidean( dimension*, dimensionUnits, normalize?, multiplier*  )

The specified kernel width or height, given as a number of specified **dimensionUnits**.

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

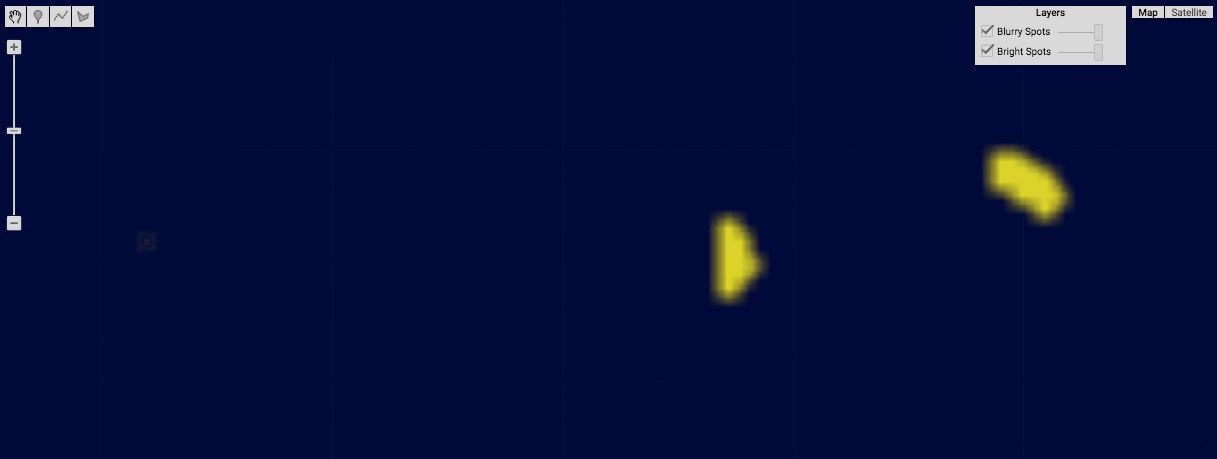
var TheKERNEL = ee.Kernel.euclidean( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **SMOOTH TRANSITIONS**

ee.Kernel.gaussian creates a new Gaussian distribution-weighted kernel of specified (radial) dimension and optionally settings. This kernel multiplies

(and then sums the resulting products of) the input values of all pixels by weights that decrease with a Gaussian function of distance

from its center.

newKernel = ee.Kernel.gaussian( dimension*, standardDeviation, units, normalize?, multiplier*  )

The specified kernel radius, given as a

floating-pointnumber of specified **units**.

The specified standard deviation, given as a number of specified **units**.

The new

kernel

A specified amount

by which all input

values are to be

multiplied,given as

a Float. Default 1.0

A Boolean set to 1

(only) if the input

values of all pixels

in the kernel are to

be normalized to

yield a sum of 1.

Default: false

The units of **radiusLengtth** and **standardDeviation**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

var TheKERNEL = ee.Kernel.gaussian( 10.0, 1.0, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );

****

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **SMOOTH TRANSITIONS**

ee.Kernel.manhattan creates a new Manhattan distance-weighted kernel of specified (square) dimensions and optionally settings. This kernel multiplies

(and then sums the resulting products of) the input values of all pixels by weights that decrease with rectilinear distance from its center.

newKernel = ee.Kernel.manhattan( dimension*, dimensionUnits, normalize?, multiplier*  )

The specified kernel width or height, given as a number of specified **dimensionUnits**.

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

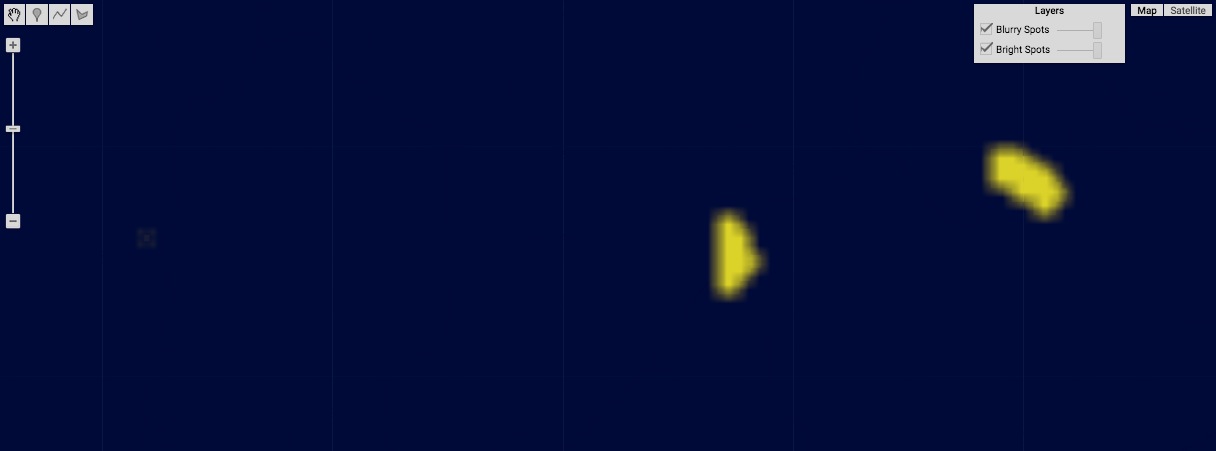
var TheKERNEL = ee.Kernel.manhattan( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) TO **SMOOTH TRANSITIONS**

ee.Kernel.chebyshev creates a new Chebyshev distance-weighted kernel of specified (square) dimensions and optionally settings. This kernel multiplies

(and then sums the resulting products of) the input values of all pixels by weights that decrease with horizontal or vertical distance

(whichever is greater) from its center.

newKernel = ee.Kernel.chebyshev ( dimension*, dimensionUnits, normalize?, multiplier*  )

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The specified kernel width or height, given as a number of specified **dimensionUnits**.

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

The new kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

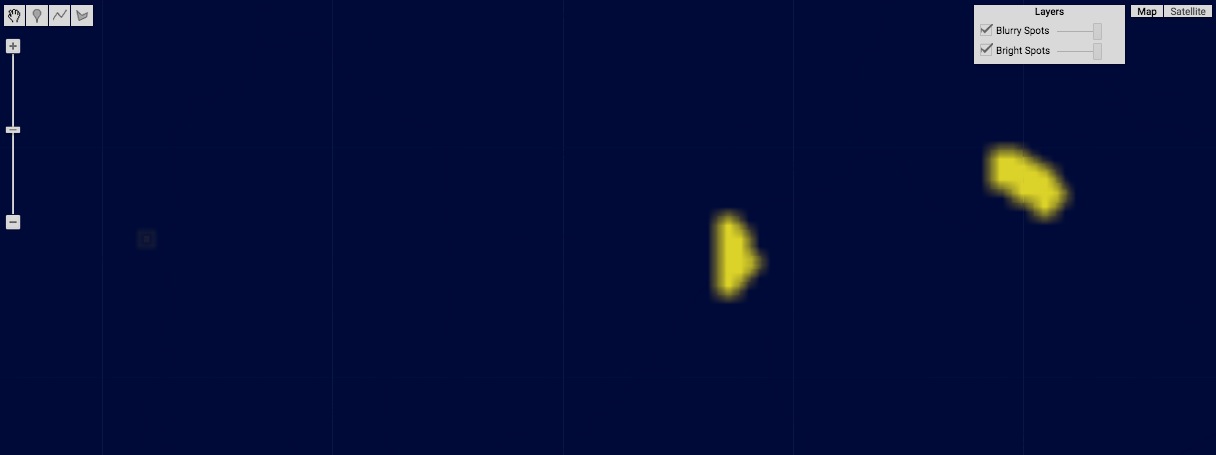
var TheKERNEL = ee.Kernel.chebyshev( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );

****

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.circle creates a new circular kernel of specified radial dimension and optional settings with pixel weights all set to 1.

newKernel = ee.Kernel.circle( dimension*, dimensionUnits, normalize?, multiplier*  )

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The specified kernel radius, given as a number of specified **dimensionUnits**.

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

var TheKERNEL = ee.Kernel.circle( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.octagon creates a new octagonal kernel of specified dimensions and optional settings with pixel weights all set to 1.

newKernel = ee.Kernel.octagon( dimension*, dimensionUnits, normalize?, multiplier*  )

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The specified kernel width and height, given as a number of specified **dimensionUnits**.

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

var TheKERNEL = ee.Kernel.octagon( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.square creates a new square kernel of specified dimensions and optional settings with pixel weights all set to 1.

newKernel = ee.Kernel.square( dimension*, dimensionUnits, normalize?, multiplier*  )

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The specified kernel width and height, given as a number of specified **dimensionUnits**.

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

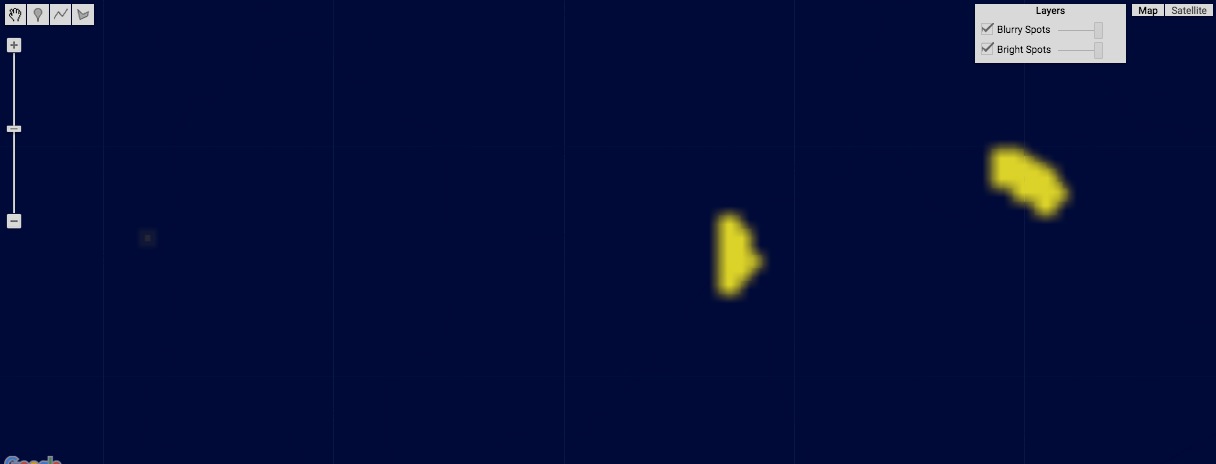
var TheKERNEL = ee.Kernel.square( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.diamond creates a new diamond-shaped kernel of specified dimensions and optional settings with pixel weights all set to 1.

newKernel = ee.Kernel.diamond( dimension*, dimensionUnits, normalize?, multiplier*  )

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The specified kernel width and height, given as a number of specified **dimensionUnits**.

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

var TheKERNEL = ee.Kernel.diamond( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.cross creates a new “X-shaped” kernel of specified dimensions and optional settings with pixel weights all set to 1.

newKernel = ee.Kernel.cross( dimension*, dimensionUnits, normalize?, multiplier*  )

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The specified kernel width and height, given as a number of specified **dimensionUnits**.

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

var TheKERNEL = ee.Kernel.cross( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.plus creates a new “+-shaped” kernel of specified dimensions and optional settings with pixel weights all set to 1.

newKernel = ee.Kernel.plus( dimension*, dimensionUnits, normalize?, multiplier*  )

A specified amount by which all input values

are to be multiplied, given as a Float. Default 1.0

The specified kernel width and height, given as a number of specified **dimensionUnits**.

The units of **dimension**, given as one of the strings 'pixels' or 'meters.' Default: 'pixels'

The new kernel

A Boolean set to 1 (only) if the input values of all pixels in the kernel are to be normalized to yield a sum of 1. Default: false

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(50);

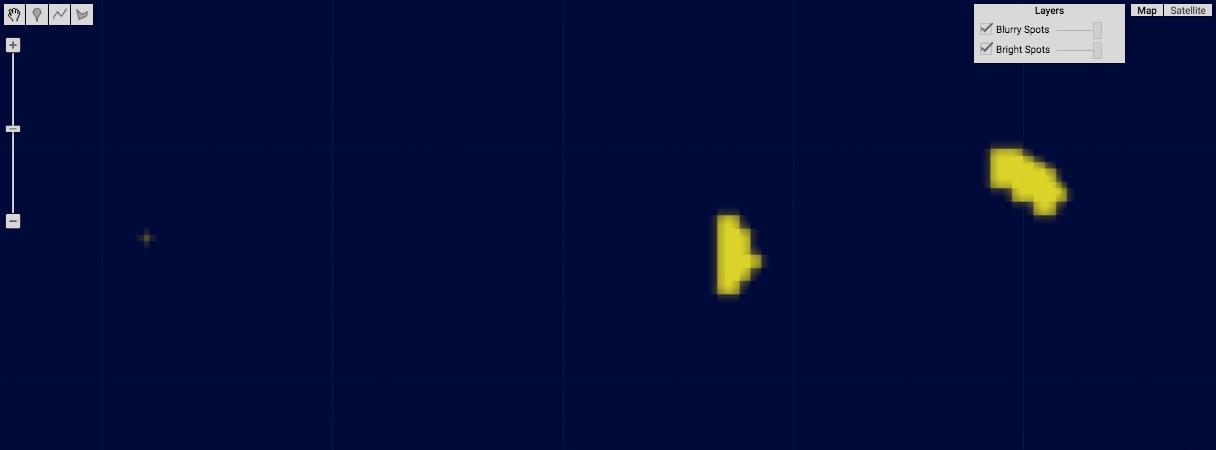
var TheKERNEL = ee.Kernel.plus( 10, 'pixels', true );

var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

Map.setCenter( -83.6369, 34.5394, 11 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Bright Spots' );

Map.addLayer( NewIMAGE, {min:0, max:1, opacity:0.9, palette:'000044,ffff22'}, 'Blurry Spots' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) OF **PARTICULAR SHAPES**

ee.Kernel.fixed creates a new rectangular kernel of specified horizontal width, vertical height, weights, and optional settings.

In kernels with even numbers of columns and/or rows, the central pixel will be just to the left and/or above the its central point.

newKernel = ee.Kernel.fixed( kernelWidth, kernelHeight, pixelWeights*, centerX, centerY, normalize?* )

The horizontal and vertical offsets of the kernel’s center,

given as a Float in pixel widths. Default: half of the specified kernel width or height

The specified kernel width, given in pixels

The specified

weights, given as

an array of as many Floats as there are

pixels in the kernel.

A Boolean set to 1 (only)

if the input values of all

pixels in the kernel are to

be normalized to yield a

sum of 1. Default: false

The specified kernel width, given in pixels

The new kernel

var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

var TheKERNEL = ee.Kernel.fixed( 3,3,[ [ -1,0,0 ],

[ 0,0,0 ],

[ 0,0,1 ] ] );

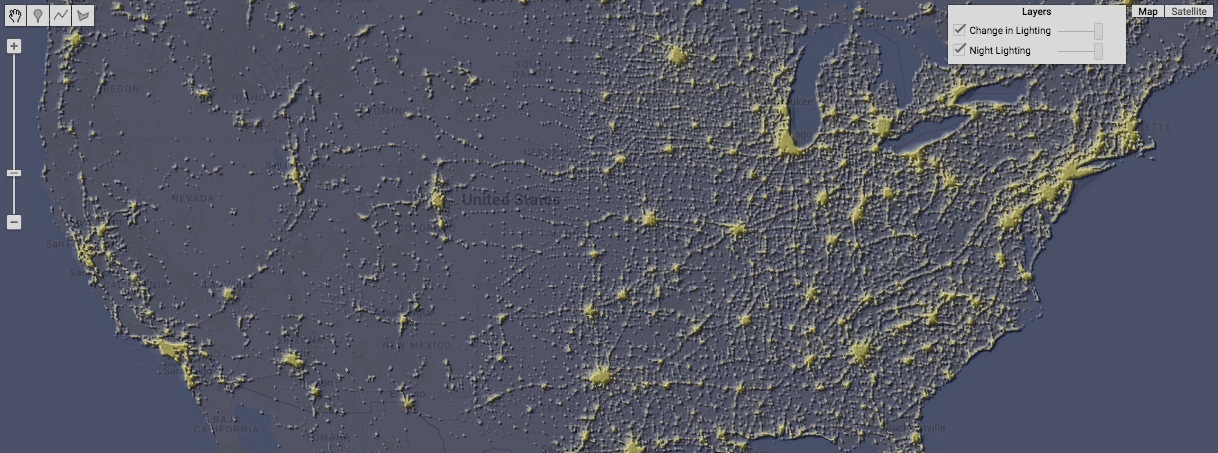
var NewIMAGE = OldIMAGE.convolve( TheKERNEL );

print( NewIMAGE.getInfo() );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:0, max:63, opacity:0.7, palette:'000044,ffff22'}, 'Night Lighting');

Map.addLayer( NewIMAGE, {min:-10, max:10, opacity:0.5}, 'Change in Lighting');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

kernel.rotate creates a new kernel by rotating a specified kernel by 90 degrees a specified number of times.

newKernel = oldKernel.rotate( quarterTurns )

The specified kernel

The specified number of clockwise 90-degree rotations, given as a negative number for counter-clockwise rotations.

The new kernel

var OriginalIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean();

var OldKERNEL = ee.Kernel.sobel( ); var OldIMAGE = OriginalIMAGE.convolve( OldKERNEL );

var NewKERNEL = OldKERNEL.rotate( 1 ); var NewIMAGE = OriginalIMAGE.convolve( NewKERNEL );

Map.setCenter( -96.592, 38.273, 5 );

Map.addLayer( OldIMAGE, {min:-1, max:1 }, 'Before Rotation');

Map.addLayer( NewIMAGE, {min:-1, max:1 }, 'After Rotation');



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

kernel.add creates a new kernel by combining the effects of two specified kernels after aligning their centers.

newKernel = oldKernel.add( oldKernel2*, normalize?* )

The first of the two

specified kernels

A Boolean set to 1 (only) if the input values of all pixels in the

kernel are to be normalized to yield a sum of 1. Default: false

The first of the two

specified kernels

The new kernel

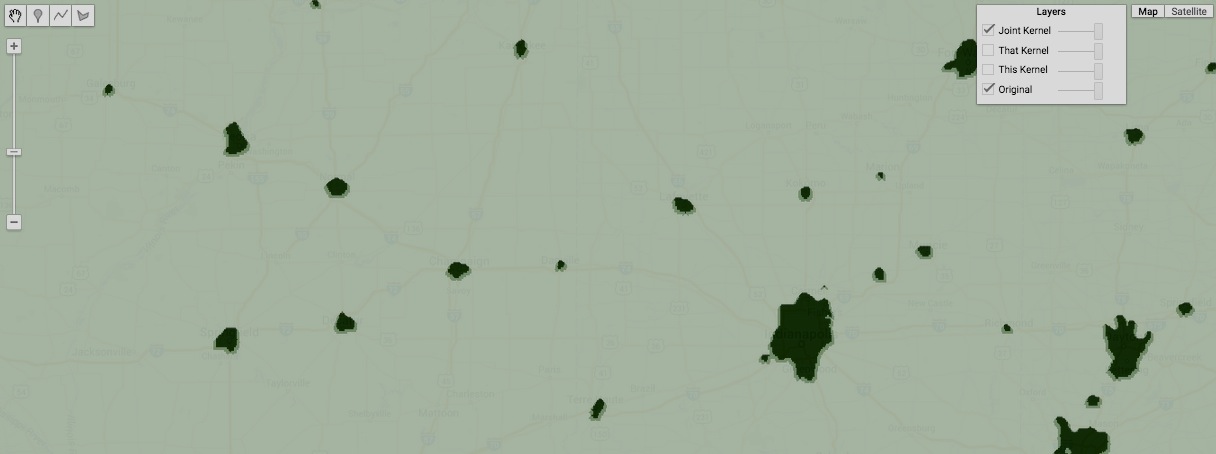
var OldIMAGE = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME\_LIGHTS').select('stable\_lights').mean().gte(60);

var ThisKERNEL = ee.Kernel.fixed( 7,7,[[ 0, 0, 0, 0, 0, 0, 1 ],

[ 0, 0, 0, 0, 0, 1, 0 ],

[ 0, 0, 0, 0, 1, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ],

 [ 0, 0, 0, 0, 0, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ]] );

var ThatKERNEL = ee.Kernel.fixed( 7,7,[[ 1, 0, 0, 0, 0, 0, 0 ],

[ 0, 1, 0, 0, 0, 0, 0 ],

[ 0, 0, 1, 0, 0, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ],

[ 0, 0, 0, 0, 0, 0, 0 ]] );

var JointKERNEL = ThisKERNEL.add( ThatKERNEL );

var ThisIMAGE = OldIMAGE.convolve( ThisKERNEL );

var ThatIMAGE = OldIMAGE.convolve( ThatKERNEL );

var JointIMAGE = OldIMAGE.convolve( JointKERNEL );

Map.setCenter(-87.3303, 40.2586, 8 );

Map.addLayer( OldIMAGE, {min:0, max:1, opacity:0.9, palette:'ffffff,000000'}, 'Original');

Map.addLayer( ThisIMAGE, {min:-1, max:1, opacity:0.5, palette:'ffffff,550000'}, 'This Kernel');

Map.addLayer( ThatIMAGE, {min:-1, max:1, opacity:0.5, palette:'ffffff,000055'}, 'That Kernel');

Map.addLayer( JointIMAGE, {min:-1, max:1, opacity:0.5, palette:'ffffff,005500'}, 'Joint Kernel');

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DOCUMENTING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

ee.Algorithms.Describe and kernel.getInfo

each creates a JSON-compatible text object representing a specified kernel.

newObject = ee.Algorithms.Describe( oldKernel )

and oldKernel.getInfo( )

The specified kernel

The new object

var TheKERNEL = ee.Kernel.laplacian8( );

print( 'From print:', TheKERNEL );

print( 'From ee.Algorithms.Describe( ):', ee.Algorithms.Describe( TheKERNEL ) );

print( 'From getInfo( ):', TheKERNEL.getInfo( ) );

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DOCUMENTING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

kernel.toString and .serialize each creates a new string presenting information on a specified kernel.

newString = oldKernel.toString ( )

and oldKernel.serialize( )

The specified date

The new string

var TheKERNEL = ee.Kernel.laplacian8( );

print( 'From print:', TheKERNEL );

print( 'From toString( ):', TheKERNEL.toString( ) );

print( 'From serialize( ):', TheKERNEL.serialize( ) );

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**PRESENTING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) IN **PRINT**

print ( kernel ) and console.log ( kernel ) present JSON-formatted text renditions of a specified kernel in the console.

print( oldKernel ) or console.log( oldKernel )

The specified kernel

var TheKERNEL = ee.Kernel.laplacian8( );

print( 'From print:', TheKERNEL );

console.log( 'From console.log:', TheKERNEL );

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**PRESENTING** [KERNEL](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) IN **PRINT**

alert ( kernel ) and confirm ( kernel ) present JSON-formatted text renditions of a specified

kernel in a pop-up message box.

alert( oldKernel ) or confirm( oldKernel )

The specified kernel

var TheKERNEL = ee.Kernel.laplacian8( );

alert( TheKERNEL );

confirm( TheKERNEL );