

The goal of this short reference is to show the Python equivalents to the morphology functions of Matlab. As far as the author of this document knows, there is not a single package that implements all morphology functions of Matlab. For this reason, three packages are presented, each offering a wide range of morphology functions for image processing. To include these three packages, the following lines must be inserted into your code:

```
import scipy.ndimage.morphology
import cv2
import skimage.morphology
```

1 Structuring Elements

Hint: All introduced packages work with normal arrays. This means a structuring element can simply be created in numpy or other packages.

Matlab	scipy	OpenCV	scikit-image	Description
<code>strel(nhood)</code>	-	-	-	Creates a flat structuring element with specified neighborhood.
<code>strel('diamond', r)</code>	-	-	<code>diamond(r)</code>	Creates a diamond-shaped structuring element with radius r .
<code>strel('disk', r, n)</code>	-	<code>getStructuringElement(cv2.MORPH_ELLIPSE, (m,n))</code>	<code>disk(r)</code>	Creates a disk-shaped structuring element with radius r and n line structuring elements to approximate the disk shape.
<code>strel('rectangle', [m,n])</code>	-	<code>getStructuringElement(cv2.MORPH_RECT, (m,n))</code>	<code>rectangle(m, n)</code>	Creates a rectangular structuring element of size $m \times n$.
<code>strel('square', w)</code>	-	-	<code>square(w)</code>	Creates a square structuring element with width w .
-	-	<code>getStructuringElement(cv2.MORPH_CROSS, (m,n))</code>	-	Creates a cross-shaped structuring element with size $m \times n$.
<code>strel('cube', w)</code>	-	-	<code>cube(w)</code>	Creates a 3-D cubic structuring element with width w .
<code>strel('cuboid', [m,n,p])</code>	-	-	-	creates a 3-D cuboidal structuring element of size $m \times n \times p$.
<code>strel('sphere', r)</code>	-	-	<code>ball(r)</code>	Creates a 3-D spherical structuring element with radius r .

2 Dilation and Erosion

Matlab	scipy	OpenCV	scikit-image	Description
<code>imdilate(img, se)</code>	<code>binary_dilation(img, se)</code> <code>gray_dilation(img, se)</code>	<code>dilate(img, se)</code>	<code>dilation(img, se)</code>	Dilates image <code>img</code> with structuring element <code>se</code>
<code>imerode(img, se)</code>	<code>binary_erosion(img, se)</code> <code>gray_erosion(img, se)</code>	<code>erode(img, se)</code>	<code>erosion(img, se)</code>	Erodes image <code>img</code> with structuring element <code>se</code>
<code>imopen(img, se)</code>	<code>binary_opening(img, se)</code> <code>gray_opening(img, se)</code>	<code>morphologyEx(img, cv2.MORPH_OPEN, se)</code>	<code>opening(img, se)</code>	Perform morphological opening
<code>imclose(img, se)</code>	<code>binary_closing(img, se)</code> <code>gray_closing(img, se)</code>	<code>morphologyEx(img, cv2.MORPH_CLOSE, se)</code>	<code>closing(img, se)</code>	Perform morphological closing
<code>bwskel(img)</code>	-	-	<code>skeletonize(img)</code>	Erodes all objects to centerlines without changing the essential structure
<code>bwperim(img)</code>	-	-	-	Find perimeter of an binary image
<code>bwhitmiss(img, se1, se2)</code>	<code>binary_hit_or_miss(img, se1, se2)</code>	<code>morphologyEx(img, cv2.MORPH_HITMISS, se)</code>	-	Binary hit-miss transformation in a binary image whose neighborhoods match the shape of <code>se1</code> and do not match the shape of <code>se2</code>
<code>imtophat(img, se)</code>	<code>white_tophat(img, se)</code>	<code>morphologyEx(img, cv2.MORPH_TOPHAT, se)</code>	<code>white_tophat(img, se)</code>	Top-hat filtering
<code>imbothat(img, se)</code>	<code>black_tophat(img, se)</code>	<code>morphologyEx(img, cv2.MORPH_BLACKHAT, se)</code>	<code>black_tophat(img, se)</code>	Bottom-hat filtering

3 Morphological Reconstruction

Matlab	scipy	OpenCV	scikit-image	Description
<code>imreconstruct(mrk, msk)</code>	-	-	<code>reconstruction(mrk, msk)</code>	Morphological reconstruction
<code>imregionalmax(img)</code>	-	-	<code>local_maxima(img)</code>	Regional maxima
<code>imregionalmin(img)</code>	-	-	<code>local_minima(img)</code>	Regional minima
<code>imhmax(img, H)</code>	-	-	<code>h_maxima(img, H)</code>	Suppresses all maxima in the intensity image <code>img</code> whose height is less than <code>H</code>
<code>imhmin(img, H)</code>	-	-	<code>h_minima(img, H)</code>	Suppresses all minima in the grayscale image <code>img</code> whose depth is less than <code>H</code>
<code>imimposemin(img, bw)</code>	-	-	-	Modifies the grayscale mask image <code>img</code> using morphological reconstruction so it only has regional minima wherever binary marker image <code>bw</code> is nonzero.
<code>imfill(img, ...)</code>	<code>fill_holes(img, se)</code>	<code>floodFill(img, se)</code>	<code>flood_fill(img, ...)</code>	Fill image regions and holes.