|  |  |
| --- | --- |
| Matlab input | Python input |
| mask(:,:,1) =  1 1 1 1  1 1 1 1  1 1 1 1  1 1 1 1  1 1 1 1  mask(:,:,2) =  1 1 0 1  1 1 1 1  1 1 1 1  1 1 1 1  1 1 1 1  mask(:,:,3) =  1 1 1 1  1 1 1 1  1 1 0 1  0 1 1 1  0 1 1 1  mask(:,:,4) =  1 1 1 1  1 1 1 1  1 1 1 1  0 1 1 1  0 1 1 1 | ﻿morph\_voxel\_grid = mask.copy()  morph\_voxel\_grid[:,:,0]=  [[1 1 1 1]  [1 1 1 1]  [1 1 1 1]  [1 1 1 1]  [1 1 1 1]]  morph\_voxel\_grid[:,:,1]=  [[1 1 0 1]  [1 1 1 1]  [1 1 1 1]  [1 1 1 1]  [1 1 1 1]]  morph\_voxel\_grid[:,:,2]  [[1 1 1 1]  [1 1 1 1]  [1 1 0 1]  [0 1 1 1]  [0 1 1 1]]  morph\_voxel\_grid[:,:,3]  [[1 1 1 1]  [1 1 1 1]  [1 1 1 1]  [0 1 1 1]  [0 1 1 1]] |
| COMPUTATION OF DISTANCE MAP  mask = padarray(mask,[1,1,1],0);  perimeter = bwperim(mask,6); % Computing the smallest ROI edge possible. Source of difference?  perimeter = perimeter(2:end-1,2:end-1,2:end-1); % Removing the padding.  mask = mask(2:end-1,2:end-1,2:end-1); % Removing the padding  distMap = bwdist(perimeter,'cityblock') + 1; % +1 according to the definition of the IBSI | ﻿morph\_voxel\_grid = mask.copy().astype("int")  img\_dims = np.shape(ROIOnlyInt)    # COMPUTATION OF DISTANCE MAP    # Create an empty distance map (technically one could also do np.ones, and skip the last part)  dist\_map = np.zeros(img\_dims)    # Copy of roi morphology mask  roi\_eroded = copy.deepcopy(morph\_voxel\_grid)    # Distances are determined in 3D  binary\_struct = generate\_binary\_structure(rank=3, connectivity=1)    # Incrementally erode the morphological mask  while np.sum(roi\_eroded) > 0:  roi\_eroded = binary\_erosion(roi\_eroded, structure=binary\_struct)  dist\_map += roi\_eroded \* 1    # Update distance from border, as minimum distance is 1 (according to IBSI definitions)  dist\_map[morph\_voxel\_grid] += 1 |
| Output for the above Matlab code | Output for the exact code of Alex  In Python |
| distMap(:,:,1) =  1 1 1 1  1 1 1 1  1 1 1 1  1 1 1 1  1 1 1 1  distMap(:,:,2) =  1 1 2 1  1 2 1 1  1 2 1 1  1 2 2 1  1 1 1 1  distMap(:,:,3) =  1 1 1 1  1 2 1 1  1 1 2 1  2 1 1 1  2 1 1 1  distMap(:,:,4) =  1 1 1 1  1 1 1 1  1 1 1 1  2 1 1 1  2 1 1 1 | ﻿dist\_map  [[1. 1. 1. 1.]  [1. 1. 1. 1.]  [0. 0. 0. 0.]  [0. 0. 0. 0.]  [0. 0. 0. 0.]]  dist\_map  [[1. 1. 1. 1.]  [1. 2. 1. 1.]  [0. 1. 0. 0.]  [0. 1. 1. 0.]  [0. 0. 0. 0.]]  dist\_map  [[1. 1. 1. 1.]  [1. 2. 1. 1.]  [0. 0. 0. 0.]  [0. 0. 0. 0.]  [0. 0. 0. 0.]]  dist\_map  [[1. 1. 1. 1.]  [1. 1. 1. 1.]  [0. 0. 0. 0.]  [0. 0. 0. 0.]  [0. 0. 0. 0.]] |
| Not the same result! | |