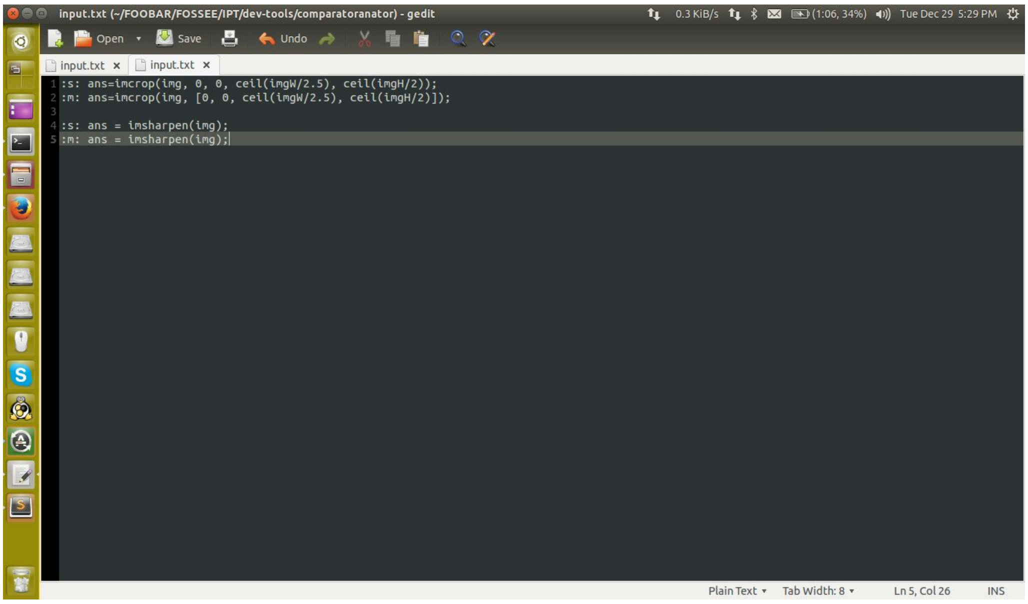
First of all, a 3d matrix in matlab and a list in scilab will not match. Thus, for the tool to work, your functions must return a matrix for an image. So make the appropriate wrappers before you start testing. The tool to make wrappers is in dev-tools/migrator

Second, extract the file sample_images.zip to get the images to test on. You can add more images to test on by simply putting more image files in the color or greyscale folder.

Input.txt

The input file to the comparison tool will be 'input.txt', whose format is as follows.
Each comparison has a Scilab function, preceded by ':s:' and it's Matlab equivalent, preceded by ':m:', as shown below.



```
:s: ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
:m: ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);

:s: ans = imsharpen(img);
:m: ans = imsharpen(img);
```

Since the functions are tested for multiple images, input.txt will use the variables img, imgH (image height) and imgW (image width) to generate different combinations of commands.
[Matlab and Scilab have different rounding rules. So please use 'ceil()' or ‘floor()’ explicitly while dividing on imgH or imgW].

After manually writing input.txt,

Running the comparison(s)

Now, open a terminal session inside dev-tools/comparinatoranator and type 'make'.
Give 'input.txt' as the input file.

The program will ask if you want to test for coloured images as well. Some image processing functions only make sense for greyscale images, so answer with “No” for those, otherwise “Yes”.

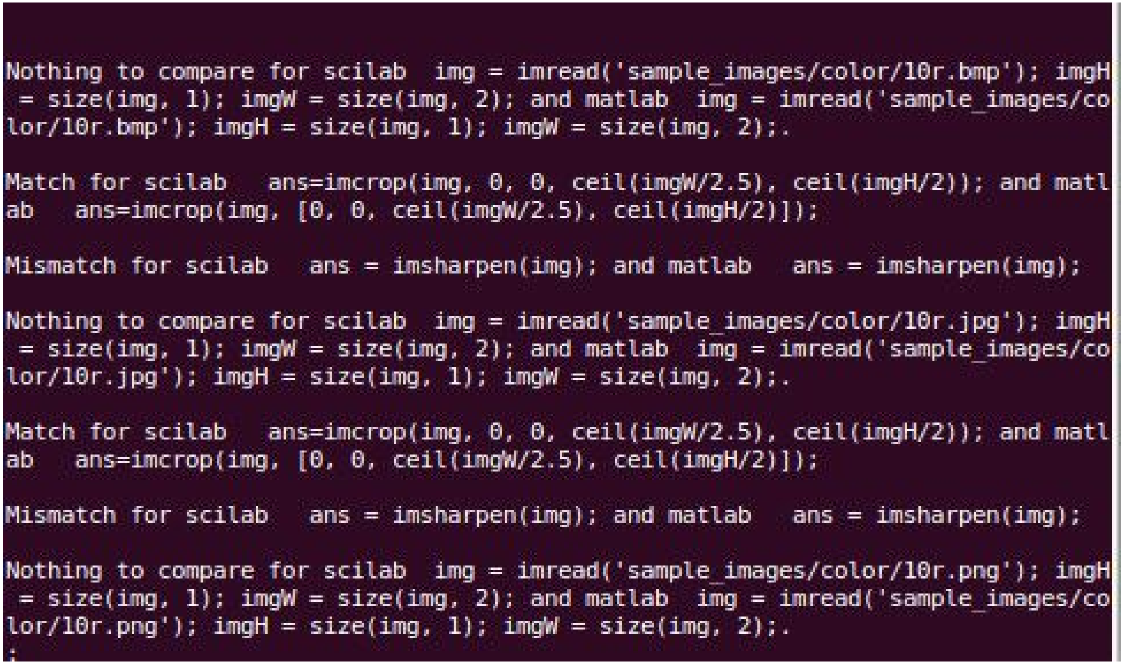
The program will create a file called “intermediate.txt” . Later, each command in intermediate.txt will be run on matlab and scilab, and the answers compared.

Specify whether you want to build the toolbox or not. If you already built it before running the script, no need to build again.

Things should be automatic at this point.

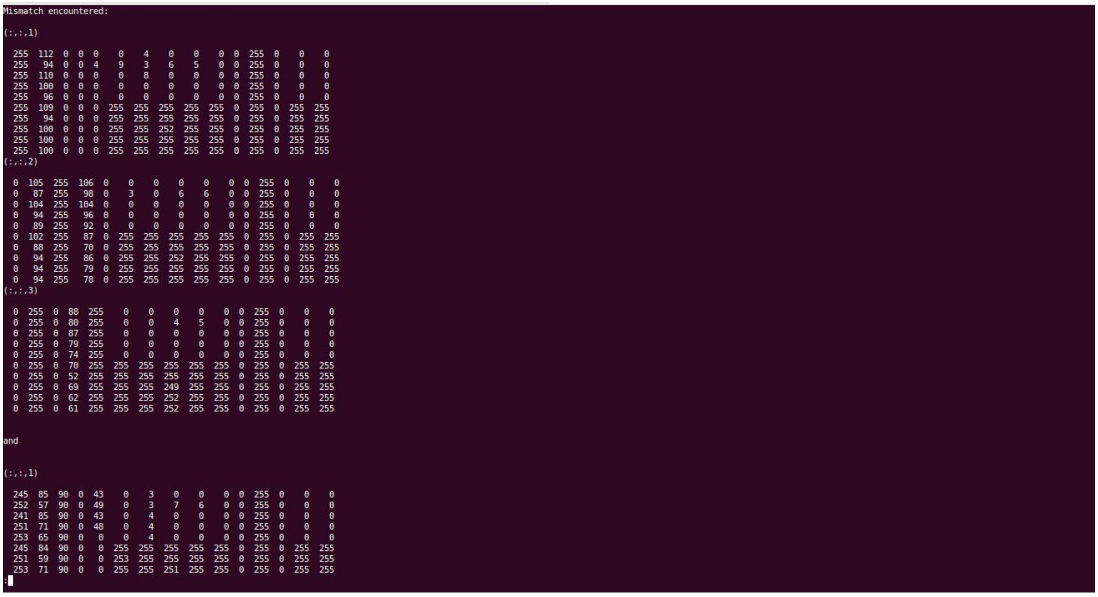
> If the console is stuck on 'sending goScilab', open another terminal session the same directory, and run:
> “scilab-cli -f sciScript.sce &> scilogs.txt”.

After the Matlab script stops running, all the comparison results are logged in 'logs.txt', as shown below.



```
Nothing to compare for scilab  img = imread('sample_images/color/10r.bmp'); imgH
 = size(img, 1); imgW = size(img, 2); and matlab  img = imread('sample_images/co
lor/10r.bmp'); imgH = size(img, 1); imgW = size(img, 2);.

Match for scilab   ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2)); and matl
ab   ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);

Mismatch for scilab   ans = imsharpen(img); and matlab   ans = imsharpen(img);

Nothing to compare for scilab  img = imread('sample_images/color/10r.jpg'); imgH
 = size(img, 1); imgW = size(img, 2); and matlab  img = imread('sample_images/co
lor/10r.jpg'); imgH = size(img, 1); imgW = size(img, 2);.

Match for scilab   ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2)); and matl
ab   ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);

Mismatch for scilab   ans = imsharpen(img); and matlab   ans = imsharpen(img);

Nothing to compare for scilab  img = imread('sample_images/color/10r.png'); imgH
 = size(img, 1); imgW = size(img, 2); and matlab  img = imread('sample_images/co
lor/10r.png'); imgH = size(img, 1); imgW = size(img, 2);.
;
```

Whenever a mismatch is encountered, the results are logged in 'sciLogs.txt'.



```
Mismatch encountered:

(:,:,1)

  255  112  0  0  0    0    4    0    0    0  0  255  0    0    0
  255   94  0  0  4    9    3    6    5    0  0  255  0    0    0
  255  110  0  0  0    0    8    0    0    0  0  255  0    0    0
  255  100  0  0  0    0    0    0    0    0  0  255  0    0    0
  255   96  0  0  0    0    0    0    0    0  0  255  0    0    0
  255  109  0  0  0  255  255  255  255  255  0  255  0  255  255
  255   94  0  0  0  255  255  255  255  255  0  255  0  255  255
  255  100  0  0  0  255  255  252  255  255  0  255  0  255  255
  255  100  0  0  0  255  255  255  255  255  0  255  0  255  255
  255  100  0  0  0  255  255  255  255  255  0  255  0  255  255
(:,:,2)

  0  105  255  106  0    0    0    0    0    0  0  255  0    0    0
  0   87  255   98  0    3    0    6    6    0  0  255  0    0    0
  0  104  255  104  0    0    0    0    0    0  0  255  0    0    0
  0   94  255   96  0    0    0    0    0    0  0  255  0    0    0
  0   89  255   92  0    0    0    0    0    0  0  255  0    0    0
  0  102  255   87  0  255  255  255  255  255  0  255  0  255  255
  0   88  255   70  0  255  255  255  255  255  0  255  0  255  255
  0   94  255   86  0  255  255  252  255  255  0  255  0  255  255
  0   94  255   79  0  255  255  255  255  255  0  255  0  255  255
  0   94  255   78  0  255  255  255  255  255  0  255  0  255  255
(:,:,3)

  0  255  0  88  255    0    0    0    0    0  0  255  0    0    0
  0  255  0  80  255    0    0    4    5    0  0  255  0    0    0
  0  255  0  87  255    0    0    0    0    0  0  255  0    0    0
  0  255  0  79  255    0    0    0    0    0  0  255  0    0    0
  0  255  0  74  255    0    0    0    0    0  0  255  0    0    0
  0  255  0  70  255  255  255  255  255  255  0  255  0  255  255
  0  255  0  52  255  255  255  255  255  255  0  255  0  255  255
  0  255  0  69  255  255  255  249  255  255  0  255  0  255  255
  0  255  0  62  255  255  255  252  255  255  0  255  0  255  255
  0  255  0  61  255  255  255  252  255  255  0  255  0  255  255


and


(:,:,1)

  245  85  90  0  43    0    3    0    0    0  0  255  0    0    0
  252  57  90  0  49    0    3    7    6    0  0  255  0    0    0
  241  85  90  0  43    0    4    0    0    0  0  255  0    0    0
  251  71  90  0  48    0    4    0    0    0  0  255  0    0    0
  253  65  90  0   0    0    4    0    0    0  0  255  0    0    0
  245  84  90  0   0  255  255  255  255  255  0  255  0  255  255
  251  59  90  0   0  253  255  255  255  255  0  255  0  255  255
  253  71  90  0   0  255  255  251  255  255  0  255  0  255  255
:
```

[Note: The above tool was coded and tested with Matlab 2014b and Scilab 5.5.0.]