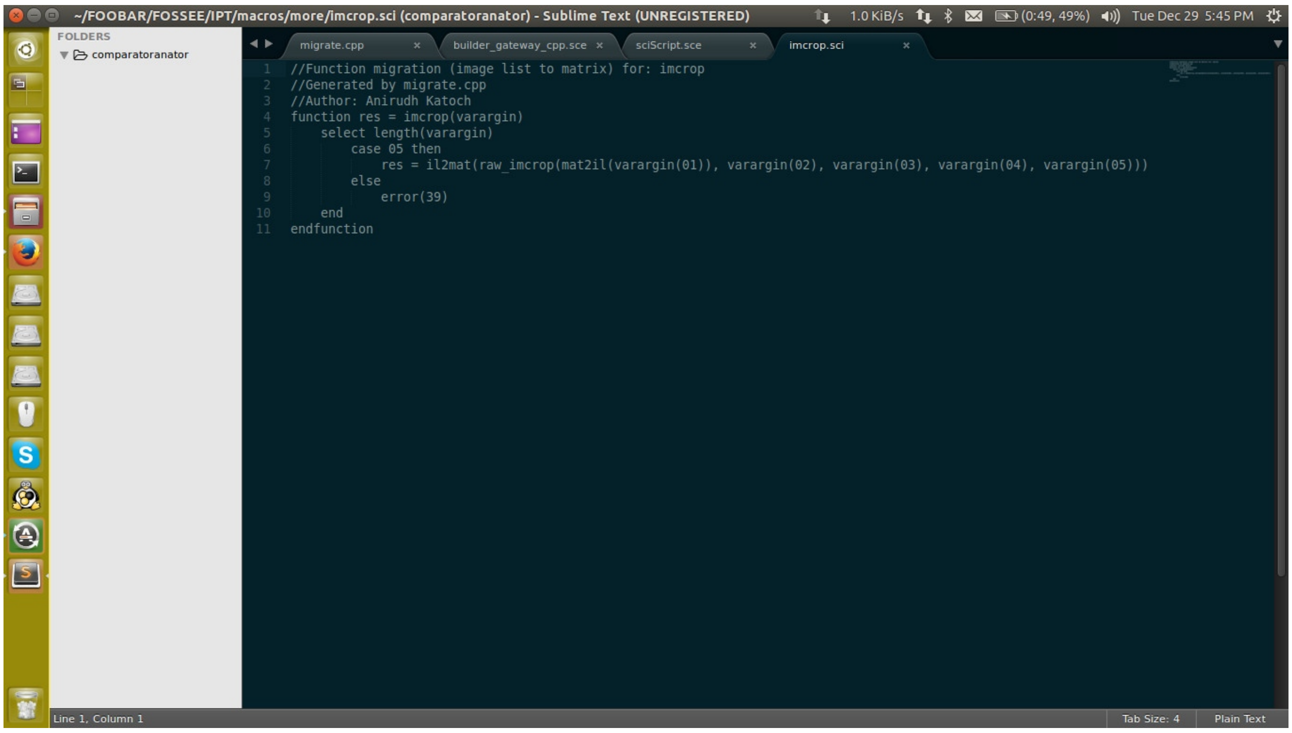
Here is how the comparator works:
It runs a specific command on matlab, and another specific command on scilab.
It then compares the last value of “ans” on each system. For most expressions or commands, this is the output you're expecting. Assignments and “expressions with just one variable name” do not change the value of ans, and this gives nothing to compare. It is good practice to explicitly include “ans = function()” in the commands that you will be running.
The method of comparison can be inspected (and changed) in macros/cmp.sci

Needless to say, you need matlab on your system.

Also needless to say, a 3d matrix in matlab and a list in scilab will not match. Thus, for the comparator to work, your functions must return a 3d matrix for an image (or 2d if only one channel) when you want to return an image.

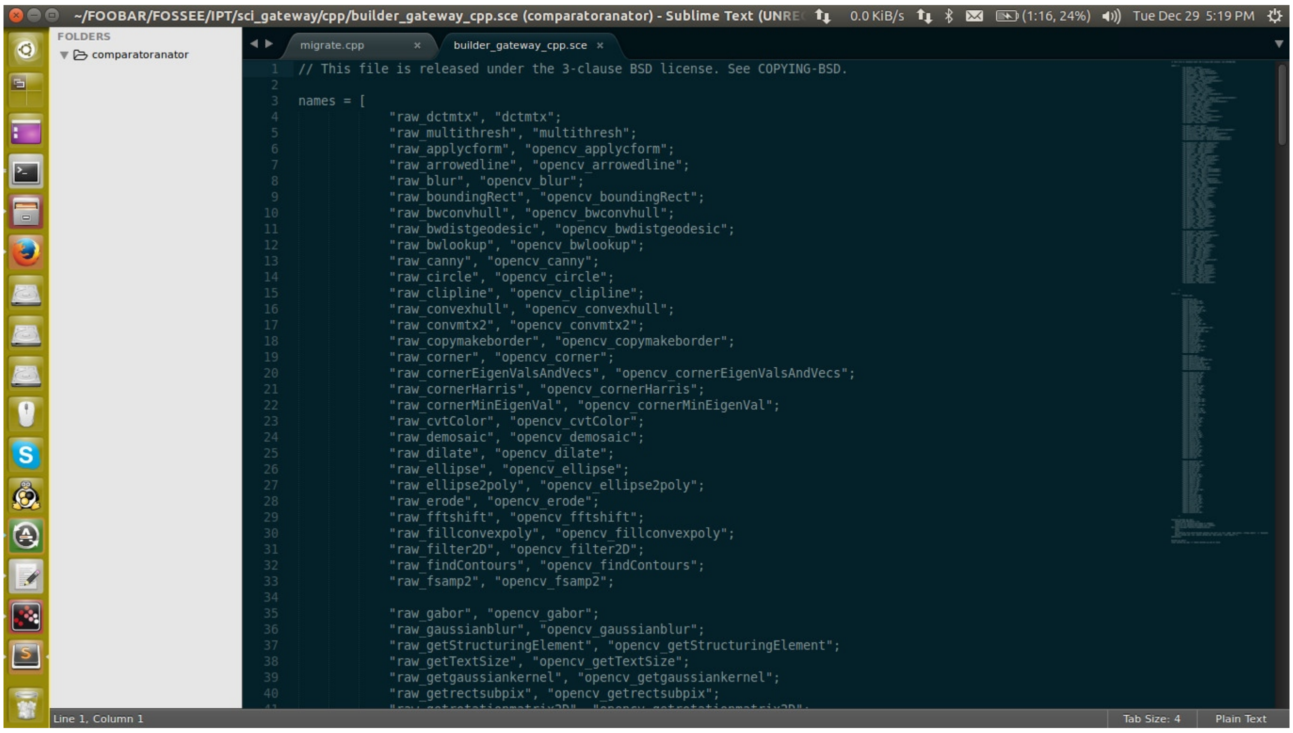
This can be accomplished by wrapping your function in a scilab function and use the functions mat2il and il2mat to convert something from a matrix to an image list or an image list to a matrix respectively.
Many functions have already been wrapped. They can be seen under macros/more/



```
//Function migration (image list to matrix) for: imcrop
//Generated by migrate.cpp
//Author: Anirudh Katoch
function res = imcrop(varargin)
    select length(varargin)
        case 05 then
            res = il2mat(raw_imcrop(mat2il(varargin(01)), varargin(02), varargin(03), varargin(04), varargin(05)))
        else
            error(39)
    end
endfunction
```

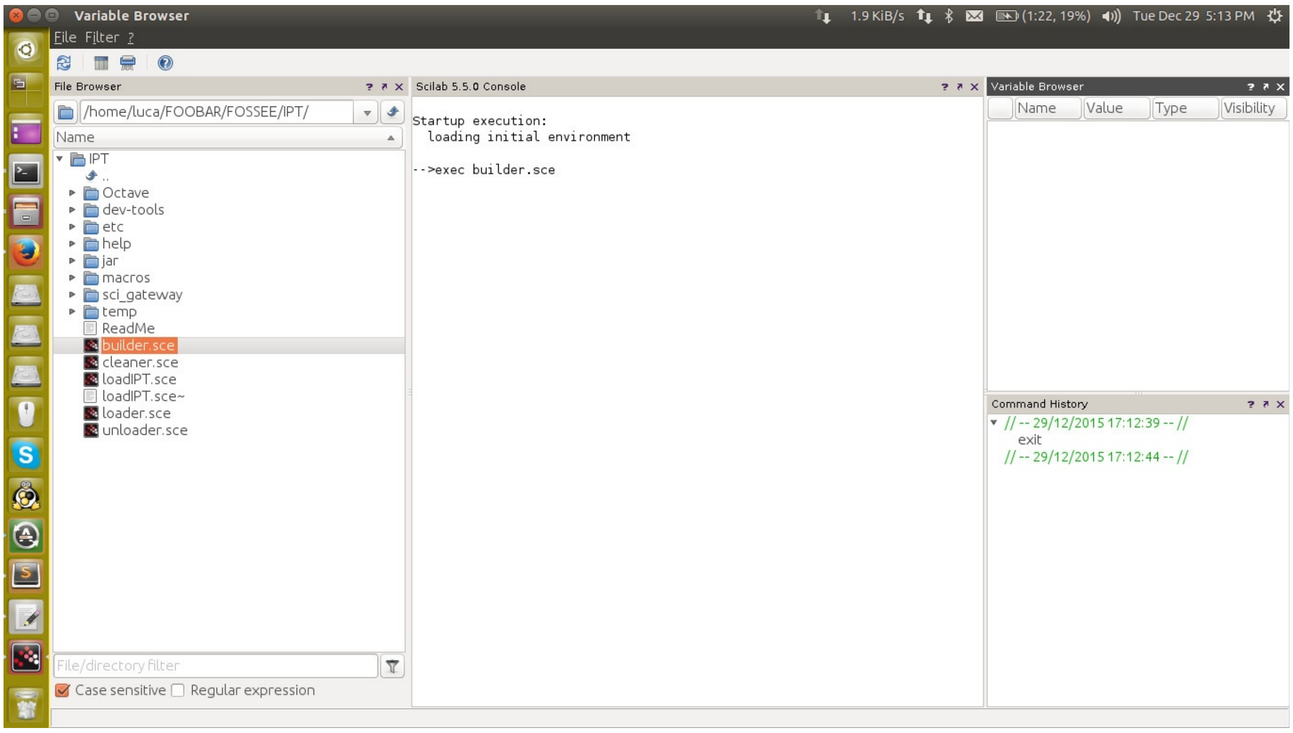
You can make the wrapper functions by hand, or use the migrator tool. Instructions regarding the migrator tool can be find in dev-tools/migrator/input.txt
For examples of the wrapper functions, inspect the functions in macros/more/
(Tip: All scripts in the macros/more/ folder will be executed by loadIPT.sce, so you can just add your wrapper function there instead of explicitely putting them in a loader.)

This is what my builder_gateway_cpp.sce looks like. As you see, I renamed all the binded functions to avoid name conflicts with the wrappers.


~/FOOBAR/FOSSEE/IPT/sci_gateway/cpp/builder_gateway_cpp.sce (comparatoranator) - Sublime Text

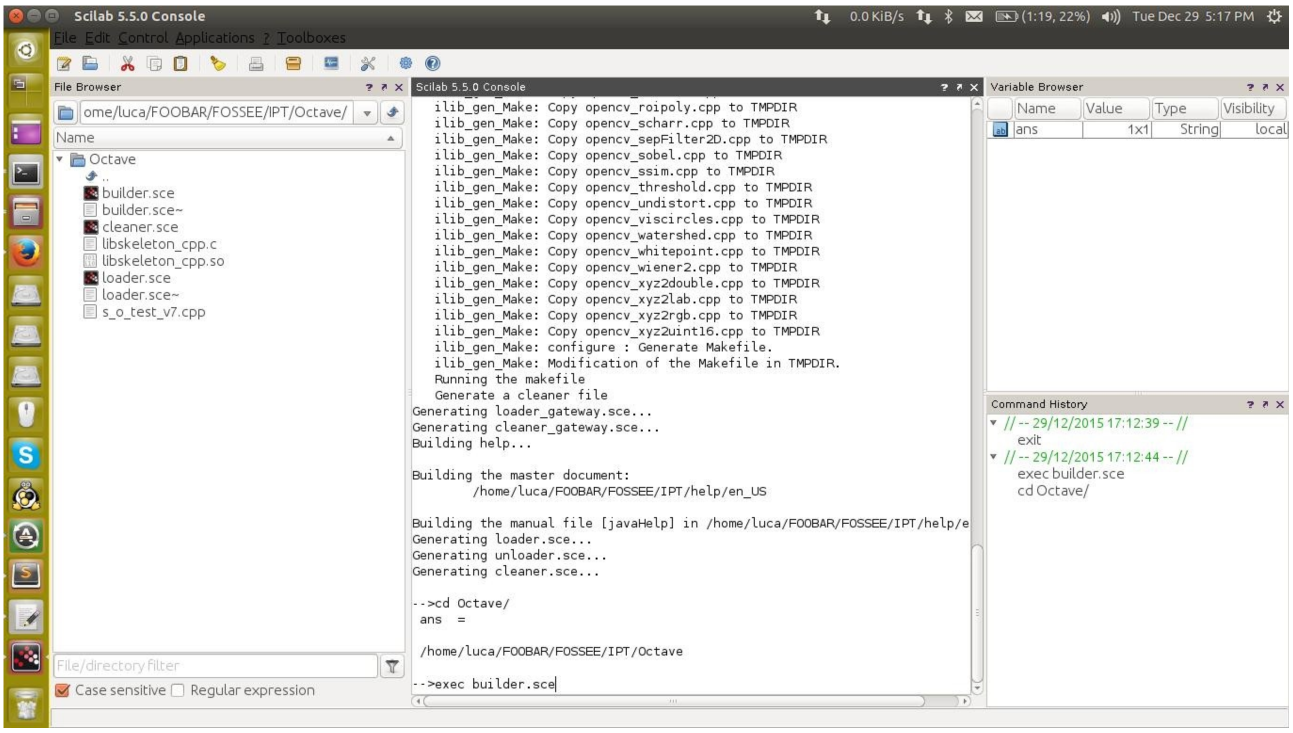
```
// This file is released under the 3-clause BSD license. See COPYING-BSD.

names = [
            "raw_dctmtx", "dctmtx";
            "raw_multithresh", "multithresh";
            "raw_applycform", "opencv_applycform";
            "raw_arrowedline", "opencv_arrowedline";
            "raw_blur", "opencv_blur";
            "raw_boundingRect", "opencv_boundingRect";
            "raw_bwconvhull", "opencv_bwconvhull";
            "raw_bwdistgeodesic", "opencv_bwdistgeodesic";
            "raw_bwlookup", "opencv_bwlookup";
            "raw_canny", "opencv_canny";
            "raw_circle", "opencv_circle";
            "raw_clipline", "opencv_clipline";
            "raw_convexhull", "opencv_convexhull";
            "raw_convmtx2", "opencv_convmtx2";
            "raw_copymakeborder", "opencv_copymakeborder";
            "raw_corner", "opencv_corner";
            "raw_cornerEigenValsAndVecs", "opencv_cornerEigenValsAndVecs";
            "raw_cornerHarris", "opencv_cornerHarris";
            "raw_cornerMinEigenVal", "opencv_cornerMinEigenVal";
            "raw_cvtColor", "opencv_cvtColor";
            "raw_demosaic", "opencv_demosaic";
            "raw_dilate", "opencv_dilate";
            "raw_ellipse", "opencv_ellipse";
            "raw_ellipse2poly", "opencv_ellipse2poly";
            "raw_erode", "opencv_erode";
            "raw_fftshift", "opencv_fftshift";
            "raw_fillconvexpoly", "opencv_fillconvexpoly";
            "raw_filter2D", "opencv_filter2D";
            "raw_findContours", "opencv_findContours";
            "raw_fsamp2", "opencv_fsamp2";

            "raw_gabor", "opencv_gabor";
            "raw_gaussianblur", "opencv_gaussianblur";
            "raw_getStructuringElement", "opencv_getStructuringElement";
            "raw_getTextSize", "opencv_getTextSize";
            "raw_getgaussiankernel", "opencv_getgaussiankernel";
            "raw_getrectsubpix", "opencv_getrectsubpix";
```

Before we start the comparatoranator, we must ensure all our functions for scilab are built properly. Run the builder for the cpp gateway and also run the builder for callOctave. You may need to change some paths in the builder.sce for callOctave



```
Startup execution:
  loading initial environment

-->exec builder.sce
```

```
// -- 29/12/2015 17:12:39 -- //
exit
// -- 29/12/2015 17:12:44 -- //
```

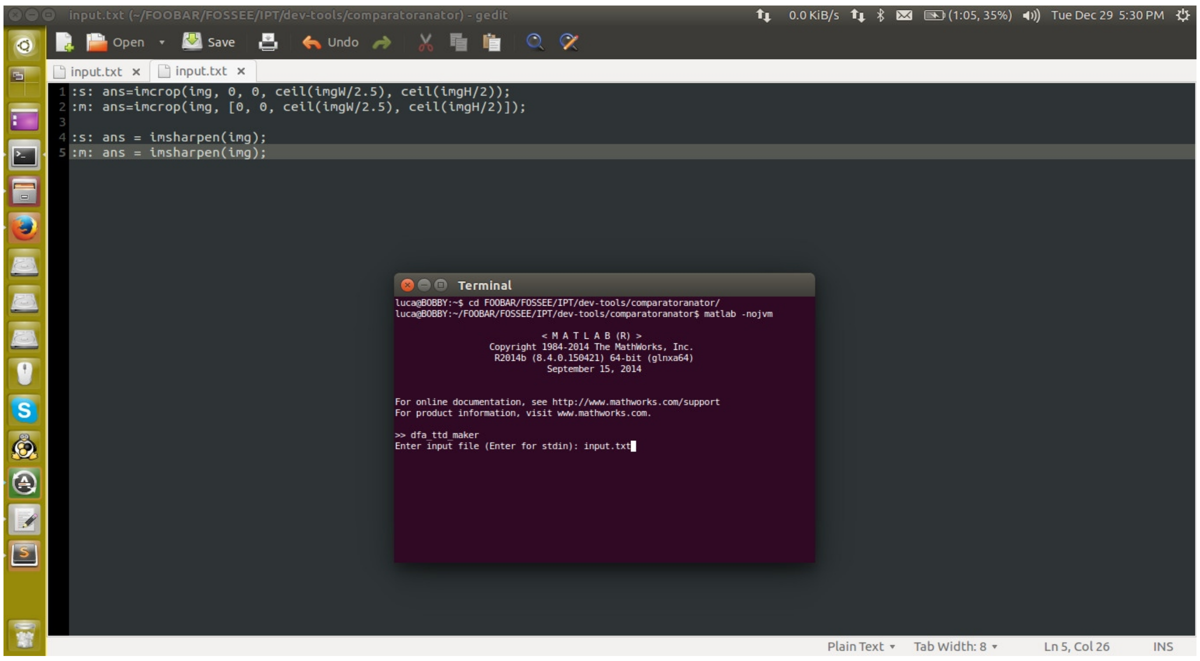


```
   ilib_gen_Make: Copy opencv_roipoly.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_scharr.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_sepFilter2D.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_sobel.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_ssim.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_threshold.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_undistort.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_viscircles.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_watershed.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_whitepoint.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_wiener2.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_xyz2double.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_xyz2lab.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_xyz2rgb.cpp to TMPDIR
   ilib_gen_Make: Copy opencv_xyz2uint16.cpp to TMPDIR
   ilib_gen_Make: configure : Generate Makefile.
   ilib_gen_Make: Modification of the Makefile in TMPDIR.
   Running the makefile
   Generate a cleaner file
Generating loader_gateway.sce...
Generating cleaner_gateway.sce...
Building help...

Building the master document:
        /home/luca/FOOBAR/FOSSEE/IPT/help/en_US

Building the manual file [javaHelp] in /home/luca/FOOBAR/FOSSEE/IPT/help/e
Generating loader.sce...
Generating unloader.sce...
Generating cleaner.sce...

-->cd Octave/
 ans  =

 /home/luca/FOOBAR/FOSSEE/IPT/Octave

-->exec builder.sce
```

Once we have built the toolbox, we can start working on the actual comparator. Since it very hacky right now, there are a few steps that need to be taken first. I'll also explain a few things.

output.txt will contain a list of scilab command (after an :s:) and a matlab command (after an :m: on the next line) pairs. A comparison will be done after each pair of such comamnds. So, each pair of :s: & :m: will give a match, mismatch or a “Nothing to compare” (i.e. ans was not given a value for one of them.)

Somewhere during the making of the comparinatoranator, we decided that each command must run for a variety of images. Therefore, we also made an input.txt where you can specify the commands to run, and have access to the variables img (image), imgH (image height) and imgW (image width). It generates output.txt for you, where it loads every image in the given folders, assigns the right values to img, imgH and imgW, and runs the commands from input.txt for each one. You may think of output.txt as a cartisian product of input.txt and the images in the folders.



```
: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);
```

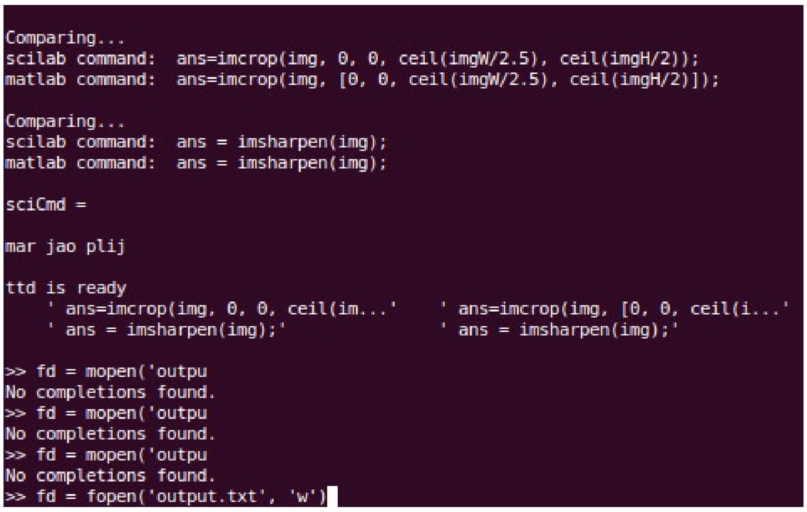
```
luca@BOBBY:~$ cd FOOBAR/FOSSEE/IPT/dev-tools/comparatoranator/
luca@BOBBY:~/FOOBAR/FOSSEE/IPT/dev-tools/comparatoranator$ matlab -nojvm

                              < M A T L A B (R) >
                    Copyright 1984-2014 The MathWorks, Inc.
                     R2014b (8.4.0.150421) 64-bit (glnxa64)
                              September 15, 2014


For online documentation, see http://www.mathworks.com/support
For product information, visit www.mathworks.com.

>> dfa_ttd_maker
Enter input file (Enter for stdin): input.txt
```

Here in the screenshot you can see my input.txt.
MATLAB AND SCILAB MAY HAVE DIFFERENT ROUNDING RULES, SO PLEASE USE "ceil" FUNCTION WHEN OPERATING ON imgW OR imgH

Manually write input.txt. Once your input.txt is ready, use your terminal to navigate to the comparatoranator folder and run "matlab -nojvm"

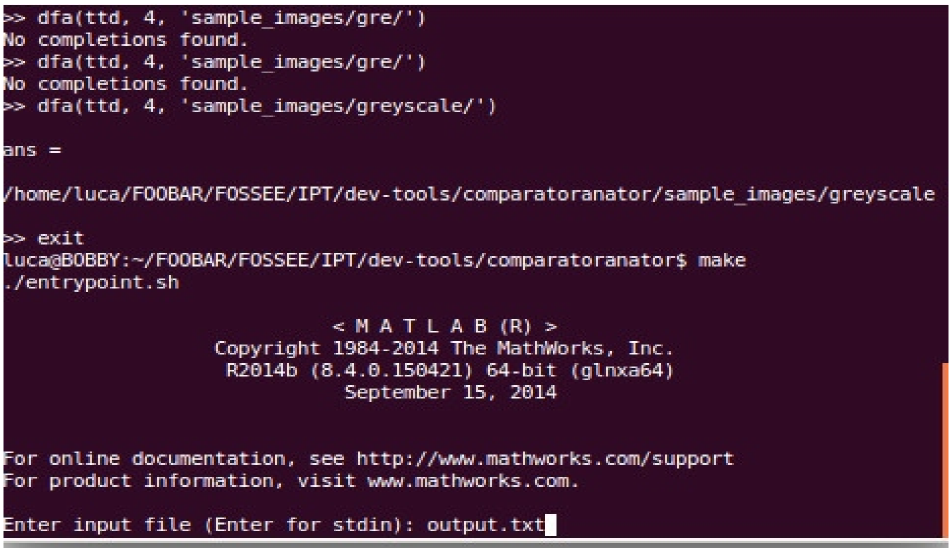
Within the matlab console, run "dfa_ttd_maker". When it asks for input, type in "input.txt"
This will create an N x 2 cell matrix with scilab and matlab commands form input.txt as strings.

After this, open output.txt for writing with the matlab command "fd = fopen('output.txt', 'w')"



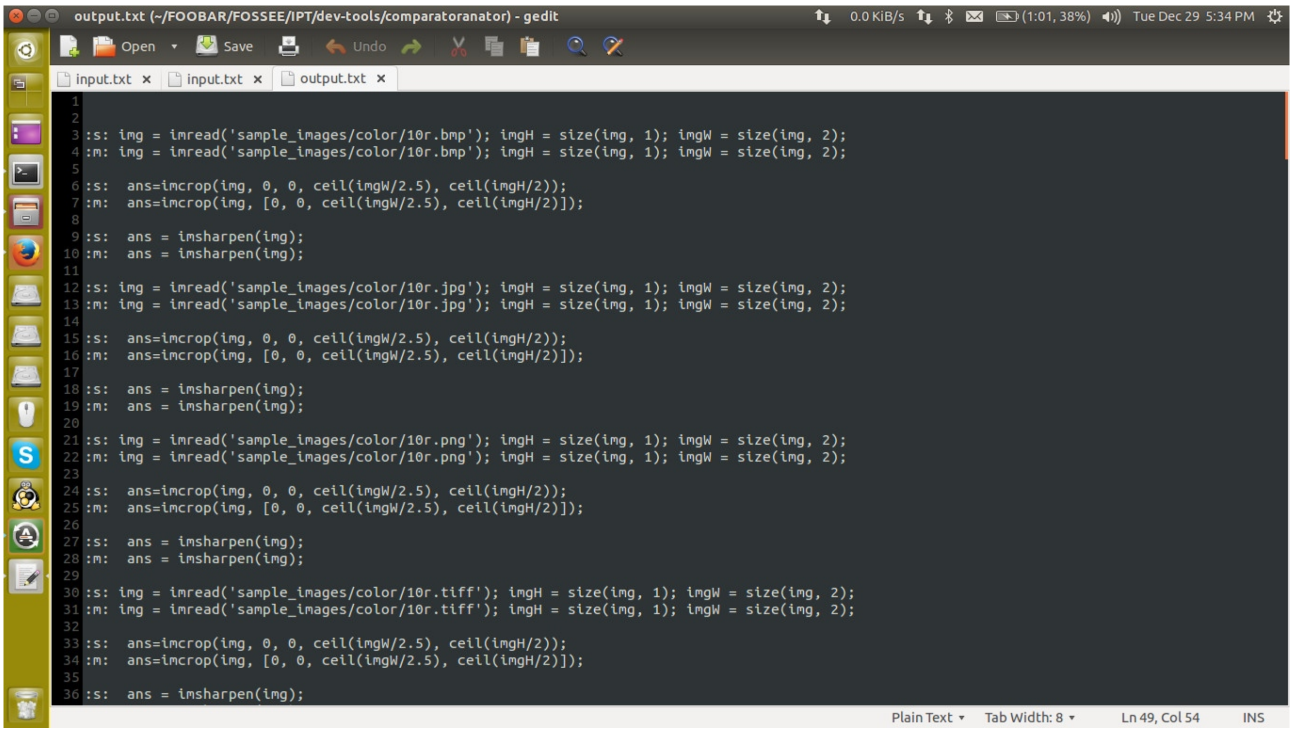
```
Comparing...
scilab command:  ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
matlab command:  ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);

Comparing...
scilab command:  ans = imsharpen(img);
matlab command:  ans = imsharpen(img);

sciCmd =

mar jao plij

ttd is ready
    ' ans=imcrop(img, 0, 0, ceil(im...'    ' ans=imcrop(img, [0, 0, ceil(i...'
    ' ans = imsharpen(img);'               ' ans = imsharpen(img);'

>> fd = mopen('outpu
No completions found.
>> fd = mopen('outpu
No completions found.
>> fd = mopen('outpu
No completions found.
>> fd = fopen('output.txt', 'w')
```

After this, run "dfa(ttd, fd, 'sample_images/color')" and "dfa(ttd, fd, 'sample_images/greyscale')"
You can skip one of them if you want only color or only greyscale.
Then you can exit matlab by running the command "exit"



```
>> dfa(ttd, 4, 'sample_images/gre/')
No completions found.
>> dfa(ttd, 4, 'sample_images/gre/')
No completions found.
>> dfa(ttd, 4, 'sample_images/greyscale/')

ans =

/home/luca/FOOBAR/FOSSEE/IPT/dev-tools/comparatoranator/sample_images/greyscale

>> exit
luca@BOBBY:~/FOOBAR/FOSSEE/IPT/dev-tools/comparatoranator$ make
./entrypoint.sh

                              < M A T L A B (R) >
                   Copyright 1984-2014 The MathWorks, Inc.
                    R2014b (8.4.0.150421) 64-bit (glnxa64)
                              September 15, 2014


For online documentation, see http://www.mathworks.com/support
For product information, visit www.mathworks.com.

Enter input file (Enter for stdin): output.txt
```

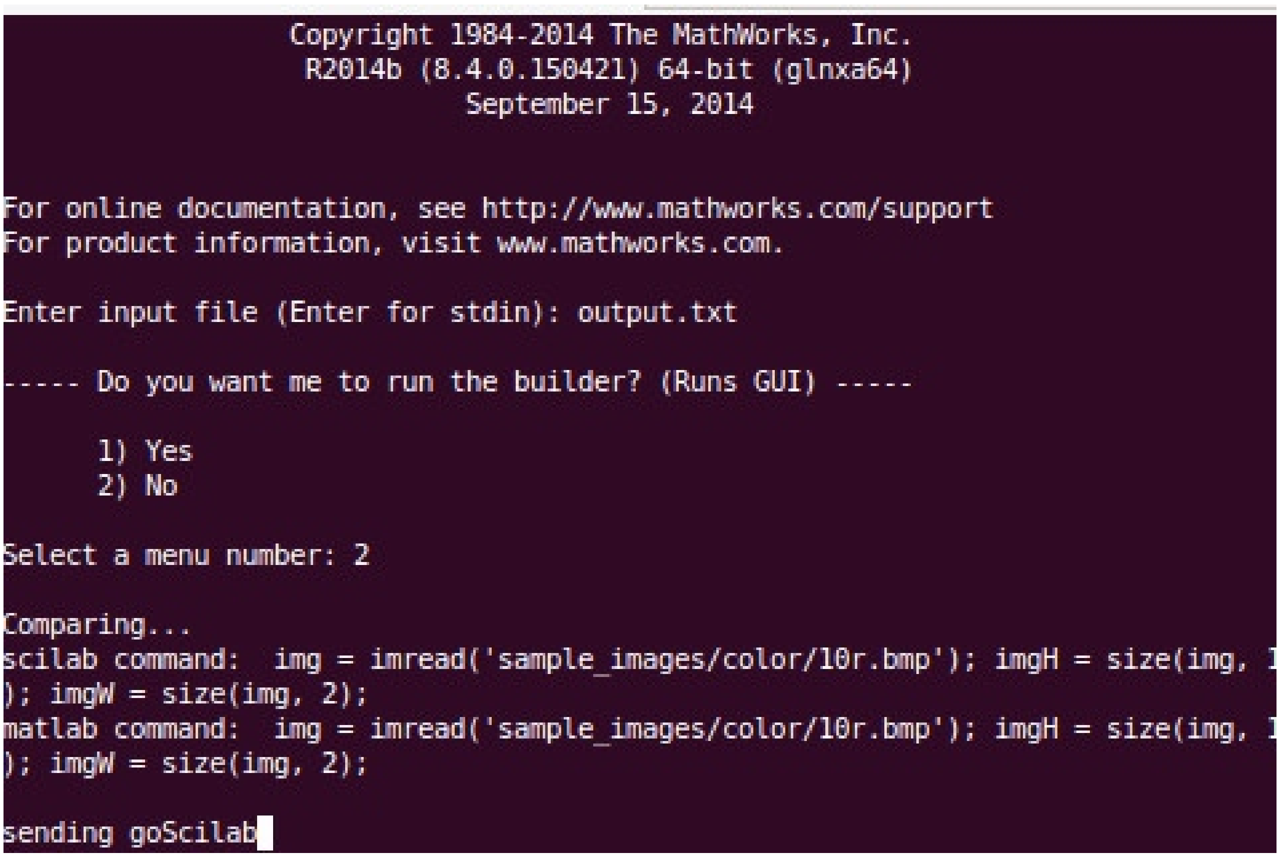
If you inspect output.txt at this point, you will find it populated with commands.



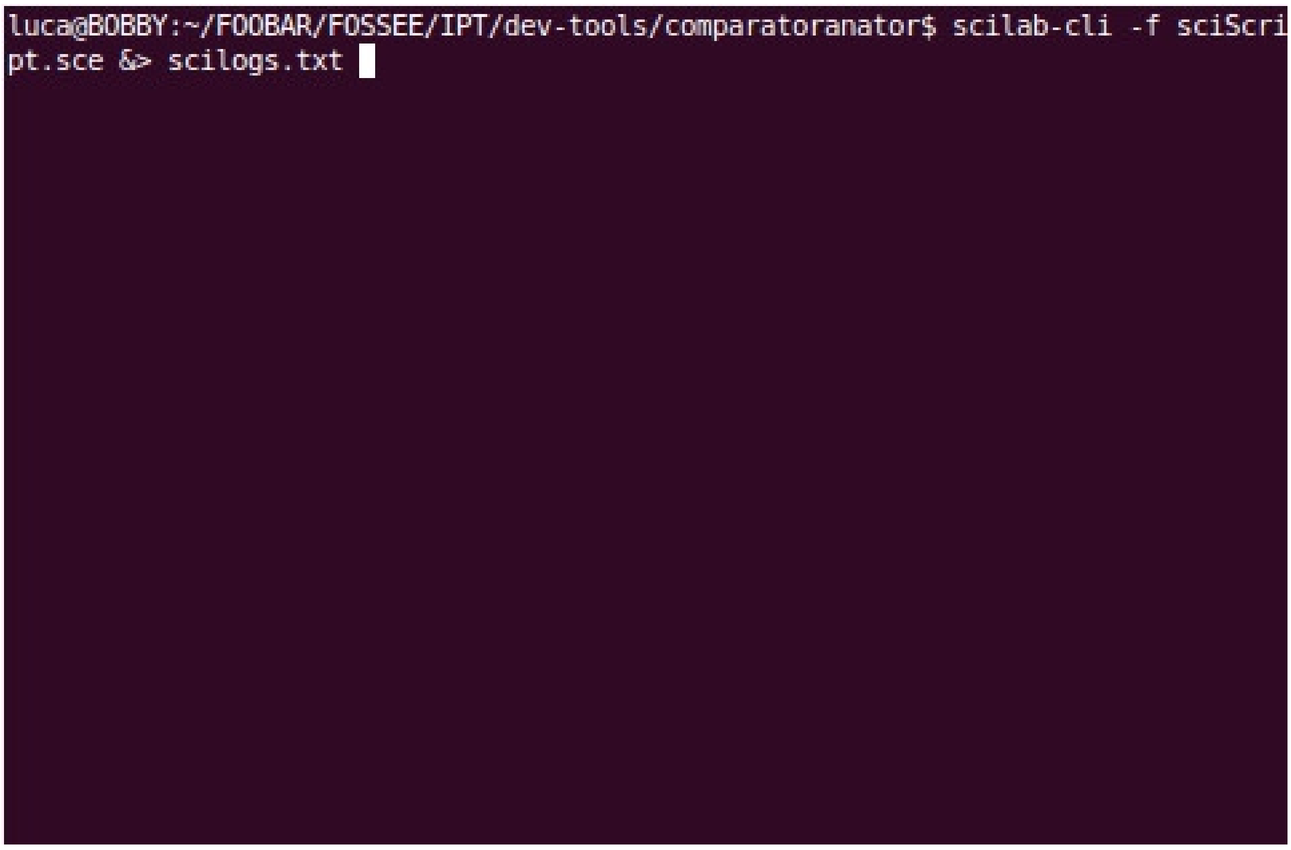
```


:s: img = imread('sample_images/color/10r.bmp'); imgH = size(img, 1); imgW = size(img, 2);
:m: img = imread('sample_images/color/10r.bmp'); imgH = size(img, 1); imgW = size(img, 2);

: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);

:s: img = imread('sample_images/color/10r.jpg'); imgH = size(img, 1); imgW = size(img, 2);
:m: img = imread('sample_images/color/10r.jpg'); imgH = size(img, 1); imgW = size(img, 2);

: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);

:s: img = imread('sample_images/color/10r.png'); imgH = size(img, 1); imgW = size(img, 2);
:m: img = imread('sample_images/color/10r.png'); imgH = size(img, 1); imgW = size(img, 2);

: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);

:s: img = imread('sample_images/color/10r.tiff'); imgH = size(img, 1); imgW = size(img, 2);
:m: img = imread('sample_images/color/10r.tiff'); imgH = size(img, 1); imgW = size(img, 2);

: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);
```

Now, on the terminal, run the command "make". This will launch matlabScript.m
When asked for an input file, type in "output.txt"
When asked if you want to run the builder, type 2 (no), as we already ran the builder before and dont need to run it again until we edit a cpp file.

When the terminal says "Sending goScilab", at this point, open another terminal window or tab, navigate to the same directory (comparinatoranator), and run the command "scilab-cli -f sciScript.sce &> scilogs.txt"

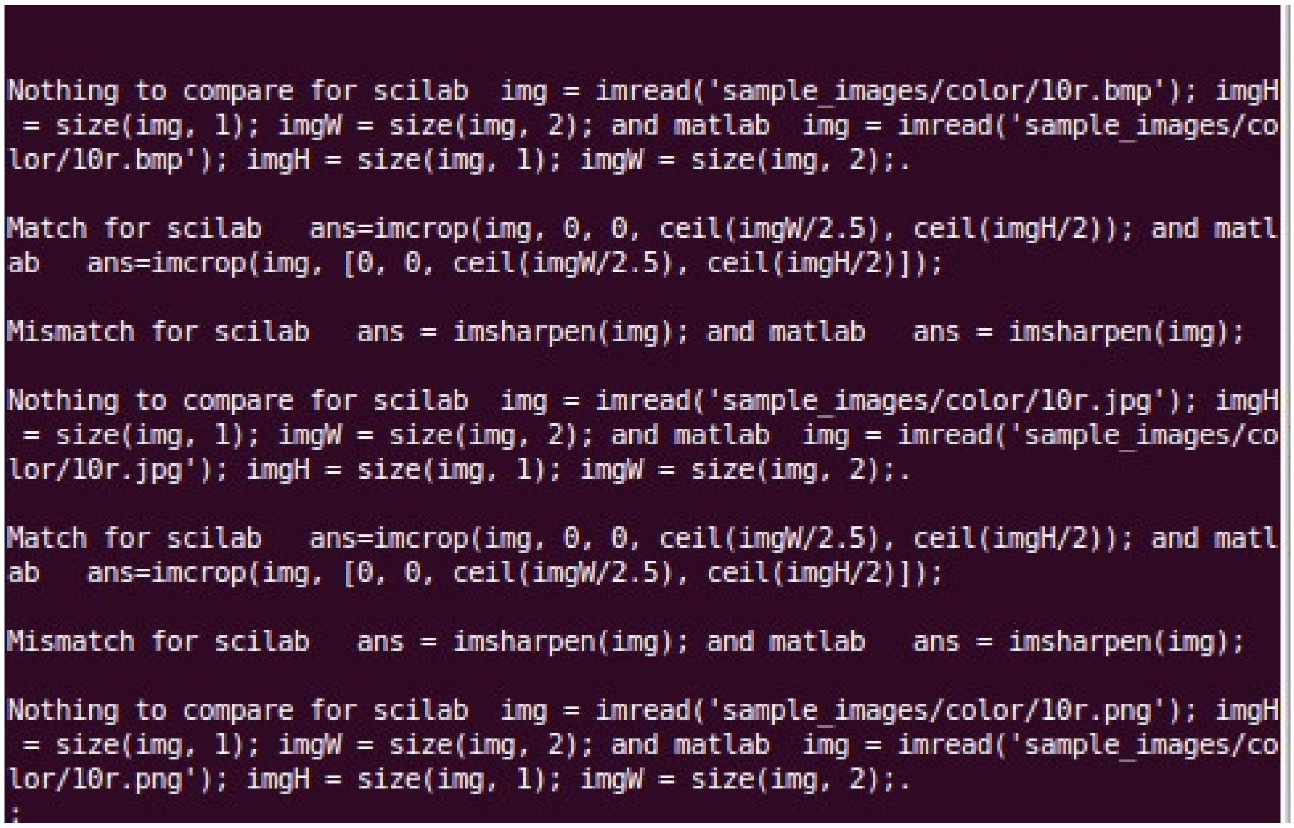


```
                      Copyright 1984-2014 The MathWorks, Inc.
                       R2014b (8.4.0.150421) 64-bit (glnxa64)
                                 September 15, 2014


For online documentation, see http://www.mathworks.com/support
For product information, visit www.mathworks.com.

Enter input file (Enter for stdin): output.txt

----- Do you want me to run the builder? (Runs GUI) -----

      1) Yes
      2) No

Select a menu number: 2

Comparing...
scilab command:   img = imread('sample_images/color/10r.bmp'); imgH = size(img, 1
); imgW = size(img, 2);
matlab command:   img = imread('sample_images/color/10r.bmp'); imgH = size(img, 1
); imgW = size(img, 2);

sending goScilab
```



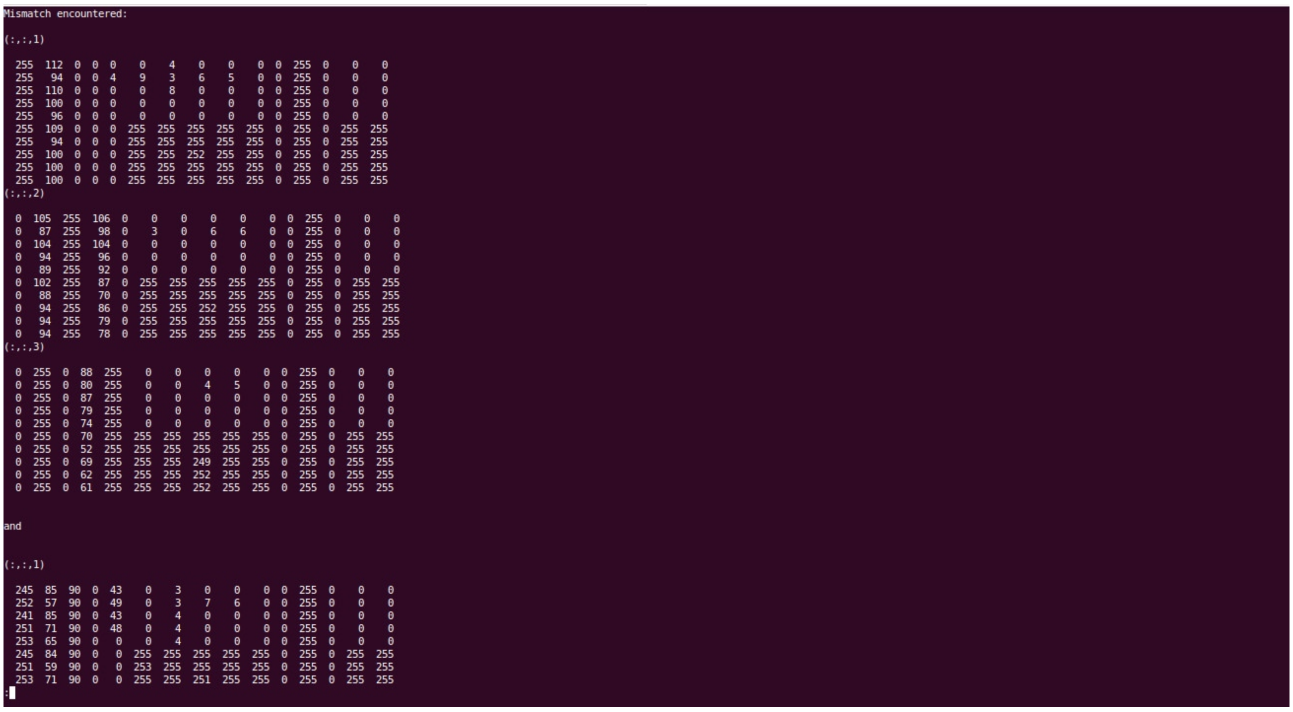
```
luca@BOBBY:~/FOOBAR/FOSSEE/IPT/dev-tools/comparatoranator$ scilab-cli -f sciScri
pt.sce &> scilogs.txt
```

After the matlab script stops running, you will see the results in logs.txt
It will only show you whether command pairs matched, mismatched or resulted in nothing to compare. As seen in the screenshot here, imcrop matches for scilab and matlab, but imsharpen doesnt.



```
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);.
```

If you want to know the exact results from a comparison and why something mismatched, then I have some bad news for you. Scilab currently doesn't allow saving lists, tlists or mlists in .mat files. However, disp() works for them.
So the method being employed here is to go through the scilogs.txt file. It displays the outputs from both matlab and scilab when a mismatch is encountered.



```
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
:
```