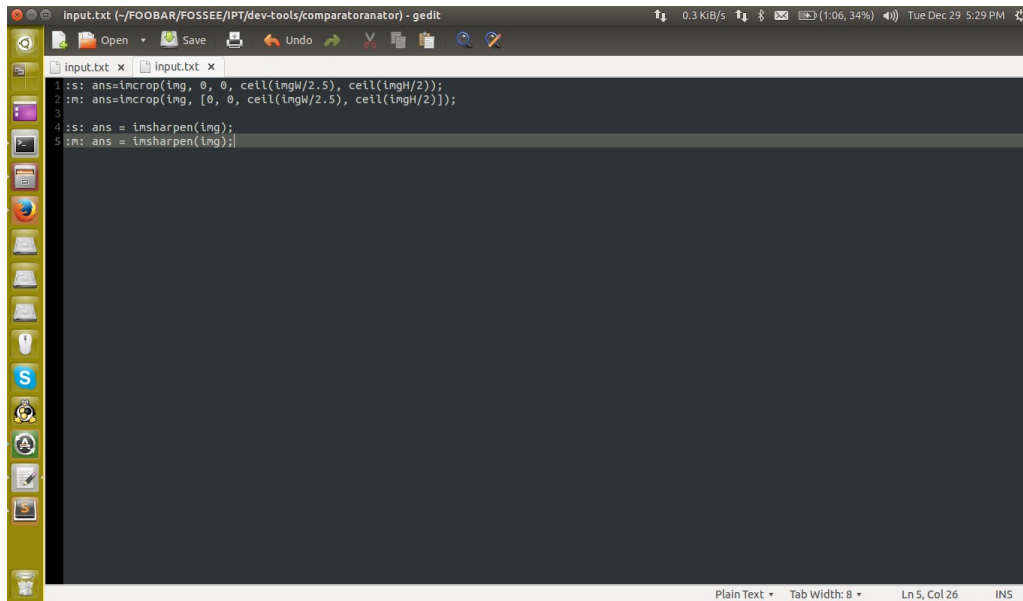


First of all, a 3d matrix in matlab and a list in scilab will not match. Thus, for the tool to work, your functions must return a matrix for an image. To do this, use the migrator tool in dev-tools/migrator.

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.



```
1:s: ans=incrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
2:m: ans=incrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);
3:
4:s: ans = imsharpen(img);
5:m: ans = imsharpen(img);
```

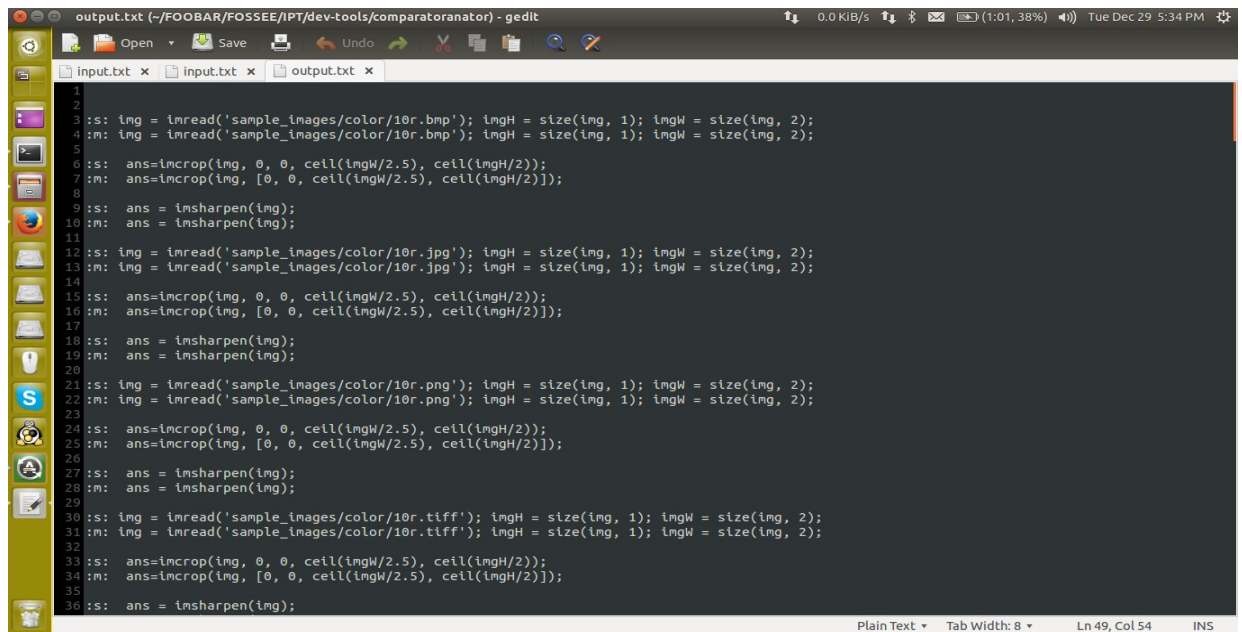
Since the functions are tested for multiple images, input.txt will use the variables img, imgH (image height) and imgW (image width) to generate specific commands.

[Matlab and Scilab have different rounding rules. So please use 'ceil()' while using imgH or imgW].

After manually writing input.txt, do the following steps:

Generating output.txt

1. Run matlab using the command 'matlab -nojvm', from within the comparator directory.
2. Run, 'dfa_ttd_maker', and give 'input.txt' as the input file.
3. fd = fopen('output.txt', 'w');
4. Next, to include colored images in the comparison, run 'dfa(ttd, fd, 'sample_images/color/)', and for greyscale images, run 'dfa(ttd, fd, 'sample_images/greyscale/)'.
5. Exit Matlab using 'exit'.




The screenshot shows a Gedit editor window with the title bar "output.txt (-/FOOBAR/FOSSEE/IPT/dev-tools/comparatoranator) - gedit". The window contains a MATLAB script with 36 lines of code. The script reads and processes three different image files: '10r.bmp', '10r.jpg', and '10r.tiff'. For each file, it performs a crop operation using `imcrop` and a sharpening operation using `imsharpen`. The script uses `size` to get the dimensions of the images and `ceil` to calculate the crop coordinates. The output of the script is saved in 'output.txt'.

```
1
2
3: s: img = imread('sample_images/color/10r.bmp'); imgH = size(img, 1); imgW = size(img, 2);
4: m: img = imread('sample_images/color/10r.bmp'); imgH = size(img, 1); imgW = size(img, 2);
5
6: s: ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
7: m: ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);
8
9: s: ans = imsharpen(img);
10: m: ans = imsharpen(img);
11
12: s: img = imread('sample_images/color/10r.jpg'); imgH = size(img, 1); imgW = size(img, 2);
13: m: img = imread('sample_images/color/10r.jpg'); imgH = size(img, 1); imgW = size(img, 2);
14
15: s: ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
16: m: ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);
17
18: s: ans = imsharpen(img);
19: m: ans = imsharpen(img);
20
21: s: img = imread('sample_images/color/10r.png'); imgH = size(img, 1); imgW = size(img, 2);
22: m: img = imread('sample_images/color/10r.png'); imgH = size(img, 1); imgW = size(img, 2);
23
24: s: ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
25: m: ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);
26
27: s: ans = imsharpen(img);
28: m: ans = imsharpen(img);
29
30: s: img = imread('sample_images/color/10r.tiff'); imgH = size(img, 1); imgW = size(img, 2);
31: m: img = imread('sample_images/color/10r.tiff'); imgH = size(img, 1); imgW = size(img, 2);
32
33: s: ans=imcrop(img, 0, 0, ceil(imgW/2.5), ceil(imgH/2));
34: m: ans=imcrop(img, [0, 0, ceil(imgW/2.5), ceil(imgH/2)]);
35
36: s: ans = imsharpen(img);
```

Running the comparison(s)

Now, open a terminal session inside dev-tools/comparatoranator and type 'make'.

Give 'output.txt' as the input file. Next, specify whether you want to build the toolbox or not.



The screenshot shows a terminal window with the following text:

```
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
```

When the terminal says '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 0 4 0 0 0 0 255 0 0 0
255 94 0 0 4 9 3 6 5 0 0 0 255 0 0 0
255 110 0 0 0 0 0 0 0 0 0 0 255 0 0 0
255 100 0 0 0 0 0 0 0 0 0 0 255 0 0 0
255 96 0 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 0 255 0 0 0
0 87 255 98 0 3 0 6 6 0 0 0 255 0 0 0
0 104 255 104 0 0 0 0 0 0 0 0 255 0 0 0
0 94 255 96 0 0 0 0 0 0 0 0 255 0 0 0
0 89 255 92 0 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 70 0 255 255 255 255 255 0 255 0 255 255
0 94 255 70 0 255 255 255 255 255 0 255 0 255 255
(:,:,3)
0 255 0 88 255 0 0 0 0 0 0 0 255 0 0 0
0 255 0 80 255 0 0 4 5 0 0 0 255 0 0 0
0 255 0 87 255 0 0 0 0 0 0 0 255 0 0 0
0 255 0 70 255 0 0 0 0 0 0 0 255 0 0 0
0 255 0 74 255 0 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 60 255 255 255 249 255 255 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 0 255 0 0 0
252 97 90 0 49 0 3 7 6 0 0 0 255 0 0 0
241 85 90 0 43 0 4 0 0 0 0 0 255 0 0 0
251 71 90 0 48 0 4 0 0 0 0 0 255 0 0 0
252 65 90 0 0 0 4 0 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.]