



Andhra Pradesh State Skill Development Corporation



SciLab

Image processing in Scilab

Sci lab

Image processing in Scilab

INTRODUCTION

Image processing and computer vision applications in a wide variety of science and technology fields. The IPCV Image Processing and Computer Vision toolbox caters to such requirements. The commands in this tool box enable even a novice to quick start image processing projects. This chapter gives a very brief introduction to the area of image processing and introduces some popular commands of IPCV toolbox.

INSTALLATION ON WINDOWS

The Scilab Image processing and computer vision toolbox can be downloaded from the website address www.atoms.Scilab.org. The home page of IPCV is shown below. Double clicking on the Image processing and computer vision toolbox, next scroll down and download the IPCV file. After downloading, starts the installation procedure in Scilab software. After IPCV is installed an “IPCV” will appear in the toolboxes menu. IPCV can be loaded into Scilab by clicking this option.



Fig. Image processing computer vision toolbox home page

BASIC IMAGE PROCESSING

For all our examples, we will use a color jpg file “flowers.jpg”. The reader should of course experiment with her own selection of an image. We assume that our test file is available in the default directory (verify your default directory using pwd). Please ensure that the file size is

not too large (test out the limit yourself you can handle the situation by using stack size command).

We will first see how the image file is opened and viewed. This is rather than straightforward.

example-1

```
x=imread(fullpath(getIPCVpath( )+ "/images/" + 'baboon.png'));  
imshow(x)
```

example-2

```
a=imread("F:\deskfiles\flowers.jpg") // change ("path name\image name dot(.) format name")  
imshow(a)
```

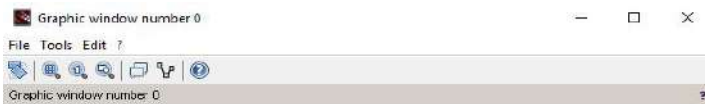


Fig. Showing an image using imshow ()

If you want to see the image pixel values, we can also use the disp() command to see the values of this image in the console window.

```
a=imread("F:\deskfiles\flowers.jpg")  
disp(a)
```

Scilab 6.0.2 Console									
119	117	139	137	131	144	148	147	31	1
175	144	135	148	150	143	147	136	43	9
169	172	146	137	146	150	160	99	27	18
85	160	177	155	149	164	156	58	7	24
19	99	171	181	165	157	101	36	18	28
0	9	91	165	171	150	35	33	47	29
14	11	17	91	153	76	4	17	27	29
10	9	1	19	52	16	3	10	16	17
0	3	0	0	0	0	5	6	7	6
0	0	0	0	0	0	5	4	4	4
0	0	0	2	0	0	0	0	3	8
0	1	0	0	0	0	0	0	0	8
0	9	5	0	0	0	0	0	0	3
0	0	0	0	5	0	12	5	1	0
2	2	1	1	1	0	12	11	7	0
2	2	1	1	1	1	4	4	2	0
1	1	1	1	1	1	14	11	8	5
1	1	1	1	1	1	12	8	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	2	0
0	0	0	0	0	0	4	4	0	0
0	0	0	0	0	0	0	0	0	4
0	0	1	2	0	0	0	16	58	103
3	0	0	0	11	25	86	104	129	147
0	0	4	41	88	120	148	156	163	163
15	46	84	121	148	162	150	149	147	146
94	133	152	153	147	143	146	143	140	139
135	146	145	140	138	139	137	141	143	140
136	133	133	136	138	140	130	135	138	137

Fig. showing pixel values using disp()command

As we have already mentioned, we would like to work with a gray image. Scilab lets you convert an rgb image to a gray image with the rgb2gray() command.

```
a=imread("F:\deskfiles\flowers.jpg")
```

```
b=rgb2gray(a)
```

```
imshow(b)
```

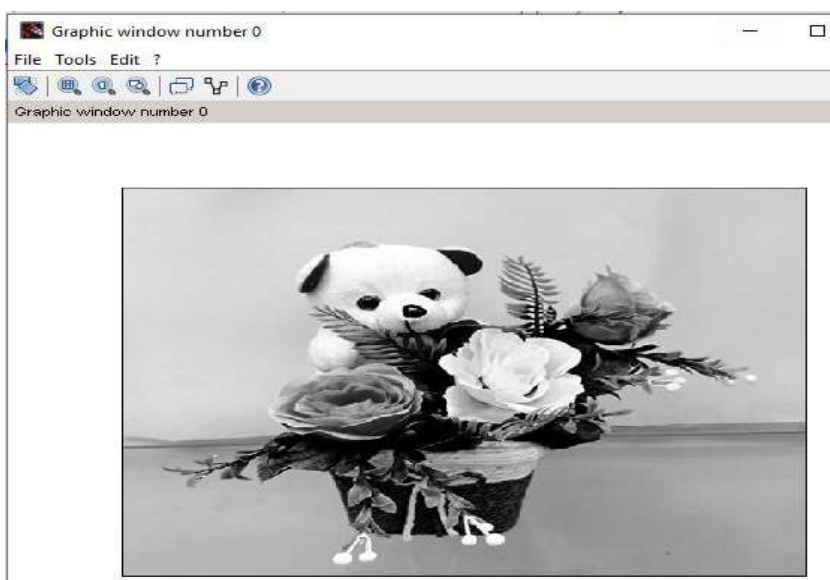


Fig. colour image to gray image using rgb2gray()