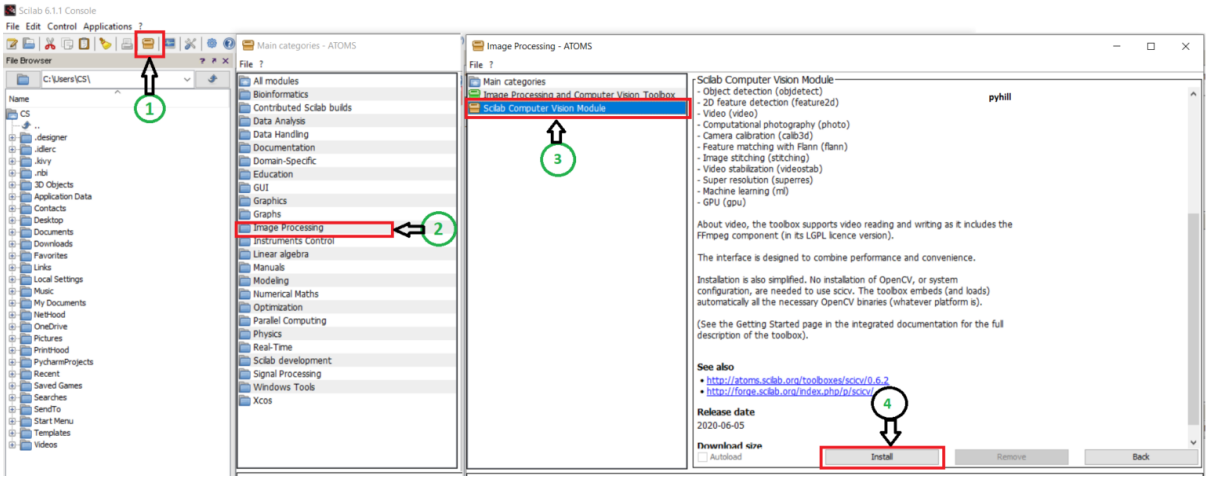
SEM-6-PRACTICALS-DIGITAL IMAGE PROCESSING

Binary Image Processing and Color Image processing.

For this practical, we need Scilab Computer Vision Module to follow the below step and install the module and restart the scilab.



Binary Image Processing.

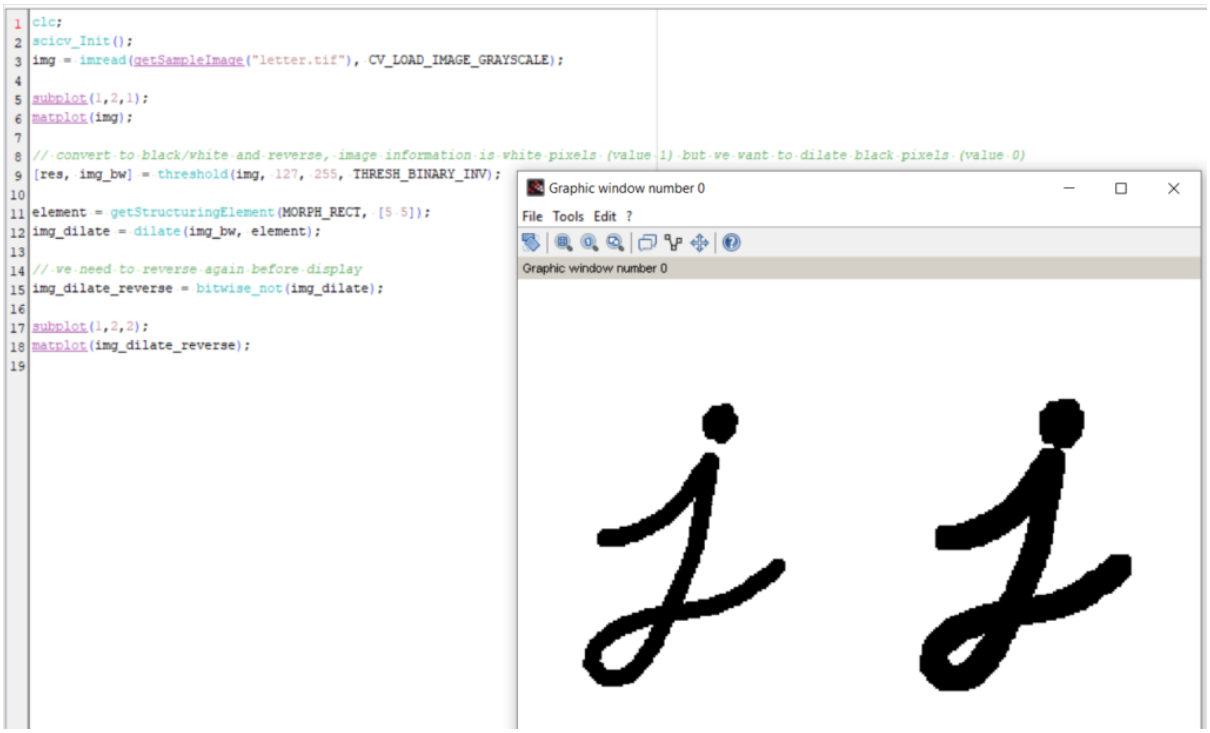
A. Image Dilation

Code:

```
clc;
scicv_Init();
img = imread(getSampleImage("letter.tif"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);
```

```
// convert to black/white and reverse, image information is white pixel
[res, img_bw] = threshold(img, 127, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_dilate = dilate(img_bw, element);

// we need to reverse again before display
img_dilate_reverse = bitwise_not(img_dilate);

subplot(1,2,2);
matplot(img_dilate_reverse);
```

Output:-



B.Image Erosion

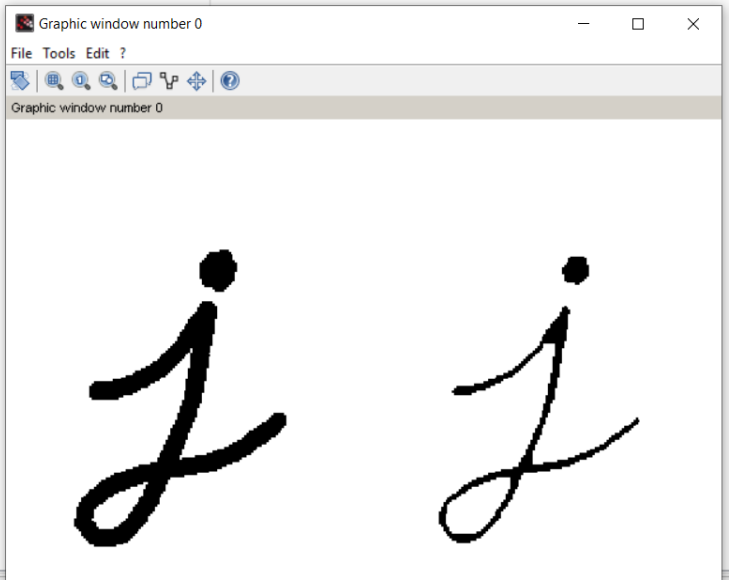
Code:

```
clear;
scicv_Init();

img = imread(getSampleImage("letter.tif"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);
```

```
// convert to black/white and reverse, image information is white pixel
[res, img_bw] = threshold(img, 127, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_erode = erode(img_bw, element);

// we need to reverse again before display
img_erode_reverse = bitwise_not(img_erode);

subplot(1,2,2);
matplot(img_erode_reverse);

delete_Mat(img);
delete_Mat(img_bw);
delete_Mat(element);
delete_Mat(img_erode);
delete_Mat(img_erode_reverse);
```

Output:-

```
clear;
scicv_Init();

img = imread(getSampleImage("letter.tif"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);

// convert to black/white and reverse, image information is white pixels (value 1) but we want to erode black pixels (value 0)
[res, img_bw] = threshold(img, 127, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_erode = erode(img_bw, element);

// we need to reverse again before display
img_erode_reverse = bitwise_not(img_erode);

subplot(1,2,2);
matplot(img_erode_reverse);

delete_Mat(img);
delete_Mat(img_bw);
delete_Mat(element);
delete_Mat(img_erode);
delete_Mat(img_erode_reverse);
```



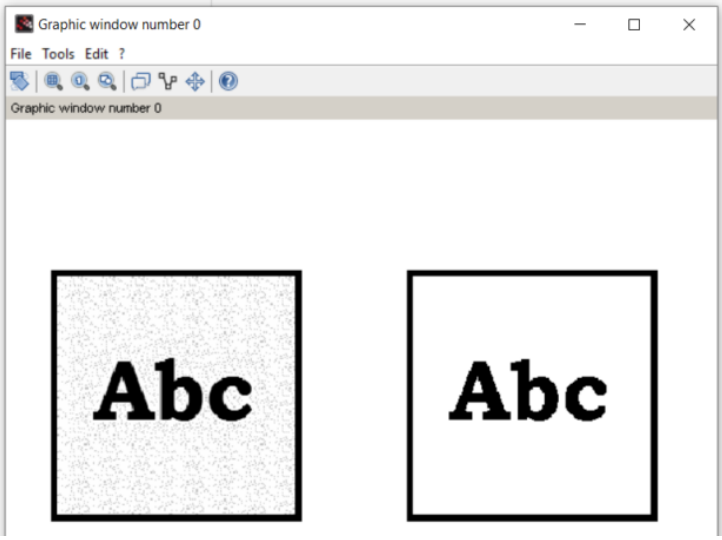
C. Image Opening

Code:-

```
clc;
scicv_Init();
// Remove noise with an opening filter
img = imread(getSampleImage("noise.png"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);

// convert to black/white and reverse, image information is white pixel
[res, img_bw] = threshold(img, 128, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_open = morphologyEx(img_bw, MORPH_OPEN, element);

// we need to reverse again before display
img_open_reverse = bitwise_not(img_open);

subplot(1,2,2);
matplot(img_open_reverse);

delete_Mat(img);
delete_Mat(img_bw);
delete_Mat(element);
delete_Mat(img_open);
delete_Mat(img_open_reverse);
```

Output:-

```
clc;
scicv_Init();
//Remove noise with an opening filter
img = imread(getSampleImage("noise.png"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);

//convert to black/white and reverse, image information is white pixels (value 1) but we want to filter black pixels (value 0)
[res, img_bw] = threshold(img, 128, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_open = morphologyEx(img_bw, MORPH_OPEN, element);

//we need to reverse again before display
img_open_reverse = bitwise_not(img_open);

subplot(1,2,2);
matplot(img_open_reverse);

delete_Mat(img);
delete_Mat(img_bw);
delete_Mat(element);
delete_Mat(img_open);
delete_Mat(img_open_reverse);
```



D . Image Closing.

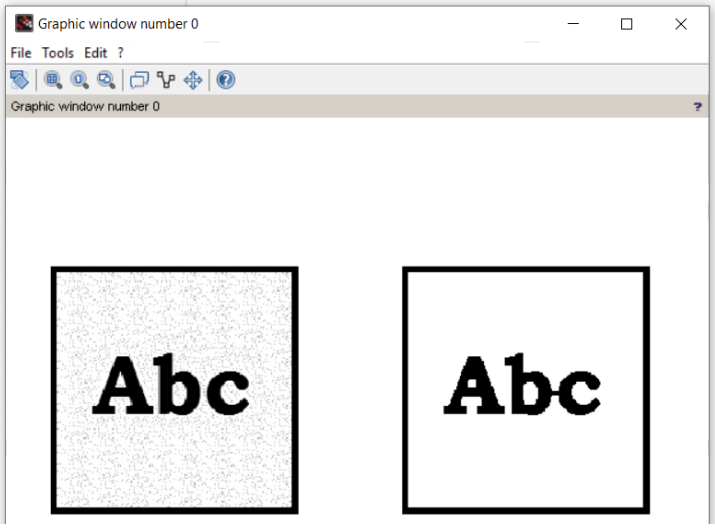
Code:-

```
clc;
scicv_Init();
// Remove noise with an opening filter
img = imread(getSampleImage("noise.png"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);

// convert to black/white and reverse, image information is white pixel
[res, img_bw] = threshold(img, 128, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_close = morphologyEx(img_bw, MORPH_CLOSE, element);

// we need to reverse again before display
img_close_reverse = bitwise_not(img_close);

subplot(1,2,2);
matplot(img_close_reverse);

delete_Mat(img);
delete_Mat(img_bw);
delete_Mat(element);
```

```
delete_Mat(img_open);
delete_Mat(img_open_reverse);
```

Output:-

```
clc;
scicv_Init();
//Remove noise with an opening filter
img = imread(getSampleImage("noise.png"), CV_LOAD_IMAGE_GRAYSCALE);

subplot(1,2,1);
matplot(img);

//convert to black/white and reverse, image information is white pixels (value 1) but we want to filter black pixels (value 0)
[res, img_bw] = threshold(img, 128, 255, THRESH_BINARY_INV);

element = getStructuringElement(MORPH_RECT, [5 5]);
img_close = morphologyEx(img_bw, MORPH_CLOSE, element);

//we need to reverse again before display
img_close_reverse = bitwise_not(img_close);

subplot(1,2,2);
matplot(img_close_reverse);

delete_Mat(img);
delete_Mat(img_bw);
delete_Mat(element);
delete_Mat(img_open);
delete_Mat(img_open_reverse);
```



← PREVIOUS

Practical 9

NEXT →

Project in Python – Colour Detection using Pandas & OpenCV

Practical 1

Practical 2

Practical 3

Practical 4

Practical 5

Practical 6

Practical 7

Practical 8

Practical 9

Practical 10

About Us

Welcome to Pyhill Education we will offer interactive python courses. Learn from a team of Pyhill with videos lessons and fun coding challenges and projects. we will

Great Quotes

-> "The best way to predict the future is to create it."

-> To err is human, but to really foul things up you need a computer.

-> "Imagination is more important than knowledge."

Quick links

Home

BLOG

MUMBAI-UNIVERSITY

AMAZON OFFERS

ABOUT US

Search

Follow us

also provide Mumbai university computer science BSc-Cs and MSc-Cs Notes and Practicals. Pyhill Education.