

暑期課程

基本影像處理

Day 8-1

指導教授：

顏淑惠、林慧珍

<http://163.13.127.10>

[http://pria.cs.tku.edu.
tw](http://pria.cs.tku.edu.tw)

指導教授：

涂瀞琹

[http://mail.tku.edu.tw
/cttu](http://mail.tku.edu.tw/cttu)

2015.07 陳毅仲

Outline

- ▶ ROI(Region of Interest)
 - ▶ Rect
 - ▶ rowRange / colRange
- ▶ 矩陣操作
 - ▶ Mat內function的使用
- ▶ 畫圖
 - ▶ 直線
- ▶ 作業

ROI (Region of Interest)

▶ Rect

- ▶ 指定一個矩形的區域，輸入左上角點的座標以及矩形長寬。
- ▶ `Rect(col座標, row座標, col長度, row長度);`
- ▶ 將Mat 根據 Rect 切出一個 ROI
- ▶ `Mat Img (Rect(Tp_x, Tp_y, Tp_width, Tp_height));`

ROI (Region of Interest)

(全域範圍)

```
#include<cv.h>
#include<highgui.h>
#include<opencv2/opencv.hpp>

using namespace std;
using namespace cv;

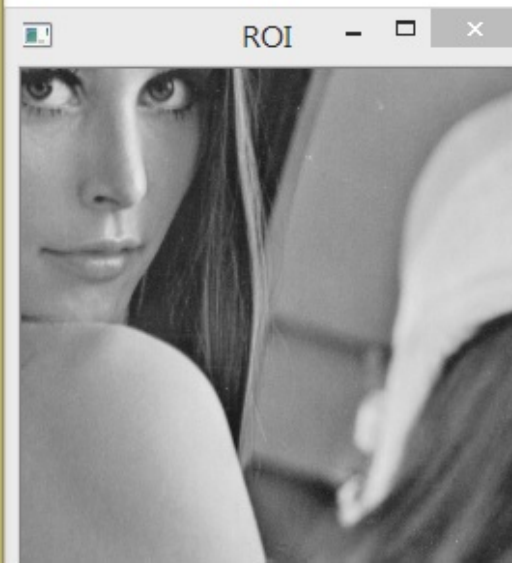
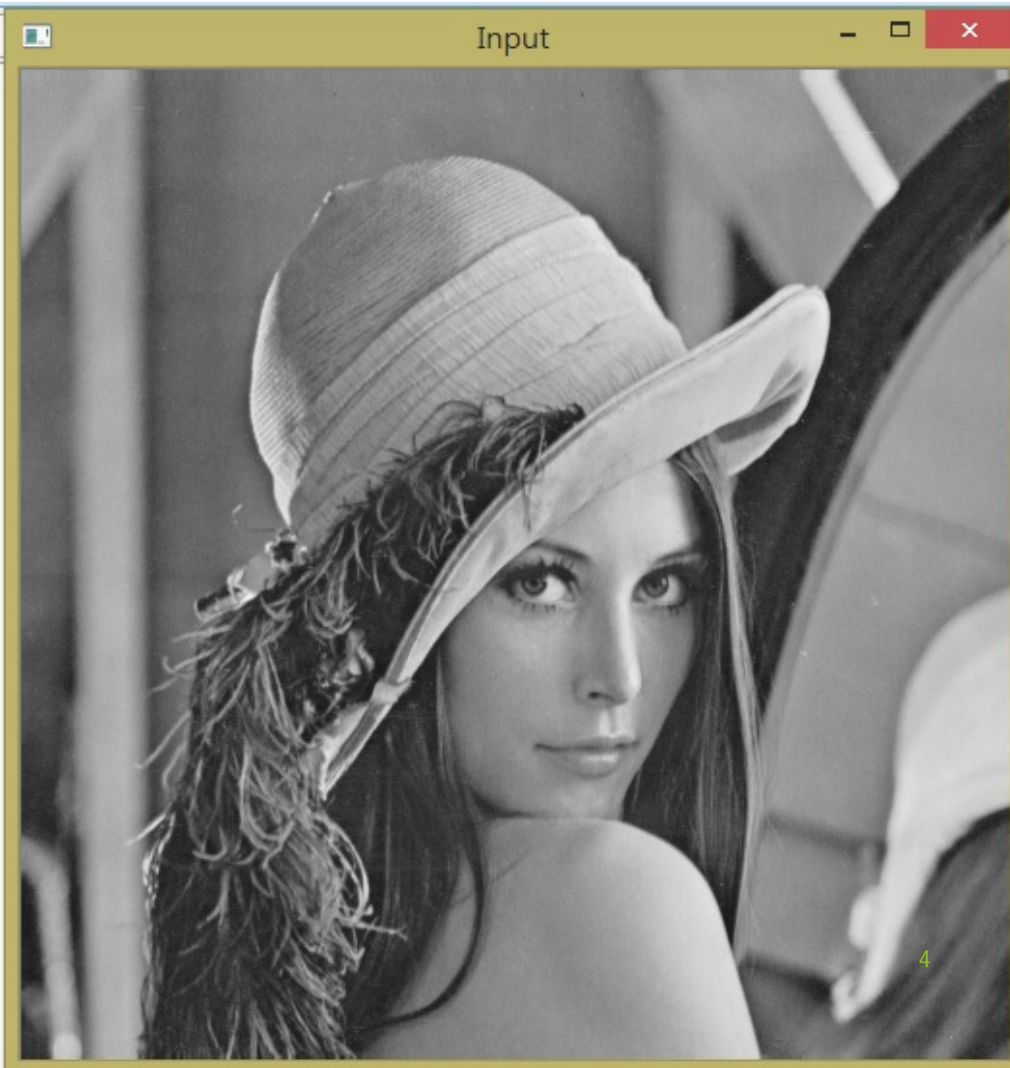
int main(){

    Mat input = imread("lena512_8bit.bmp",1);
    int col = input.cols;
    int row = input.rows;
    Mat roi = input(Rect(row/2,col/2,row/2,col/2)).clone();

    imshow("Input",input);
    imshow("ROI",roi);
    waitKey(0);

    return 0;

}
```



ROI (Region of Interest)

▶ rowRange / colRange

- ▶ 將Mat 根據 rowRange / colRange 切出一個 ROI
- ▶ `Mat Img.rowRange(StartRow , EndRow).colRange(StartCol, EndCol);`

ROI (Region of Interest)

(全域範圍)

```
#include<cv.h>
#include<highgui.h>
#include<opencv2/opencv.hpp>

using namespace std;
using namespace cv;

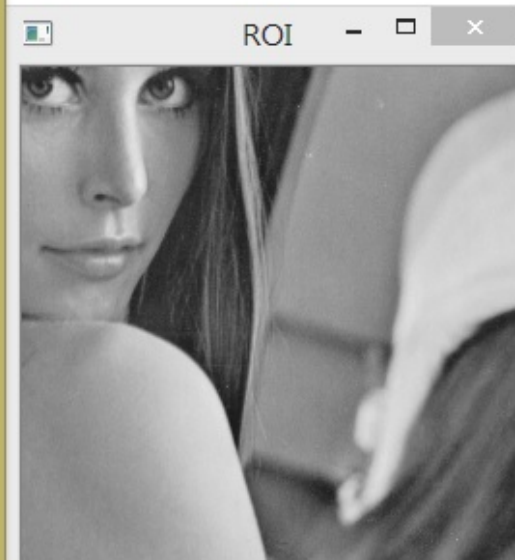
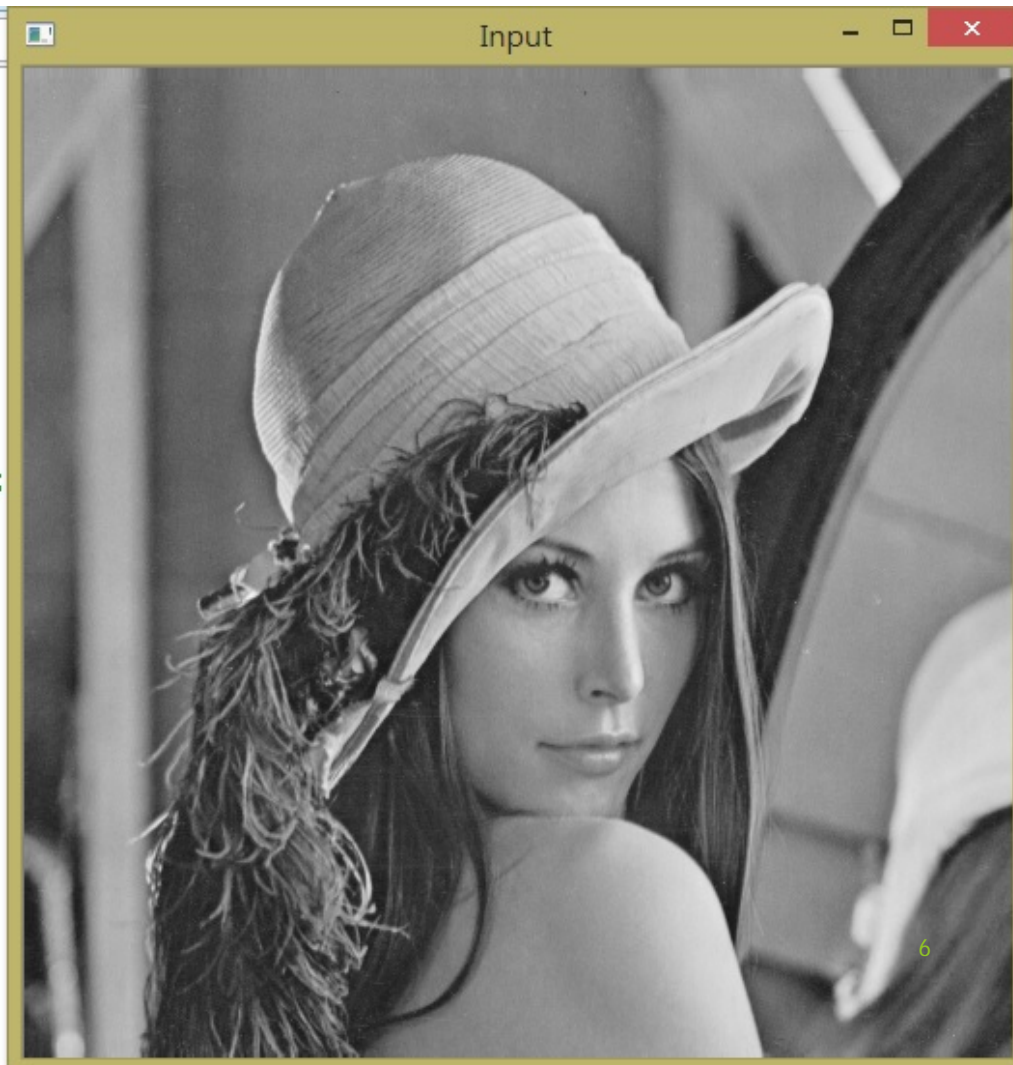
int main(){

    Mat input = imread("lena512_8bit.bmp",1);
    int col = input.cols;
    int row = input.rows;
    //Mat roi = input(Rect(row/2,col/2,row/2,col/2)).clone();
    Mat roi = input.rowRange(row/2,row).colRange(col/2,col);

    imshow("Input",input);
    imshow("ROI",roi);
    waitKey(0);

    return 0;

}
```



矩陣操作

► 取得矩陣資訊

- `Mat input;`
- `input.rows;` `//` 矩陣的高
- `input.cols;` `//` 矩陣的寬
- `input.types();` `//` 矩陣的型態 (CV_8U/ CV_32FC3 / CV8UC3 ...)
- `input.channels();` `//` 矩陣的通道數

矩陣操作

► 矩陣的運算

► Mat Input;

► Input.t(); // 轉置矩陣

► Input.inv(); //反矩陣

```
#include<cv.h>
#include<highgui.h>
#include<opencv2/opencv.hpp>

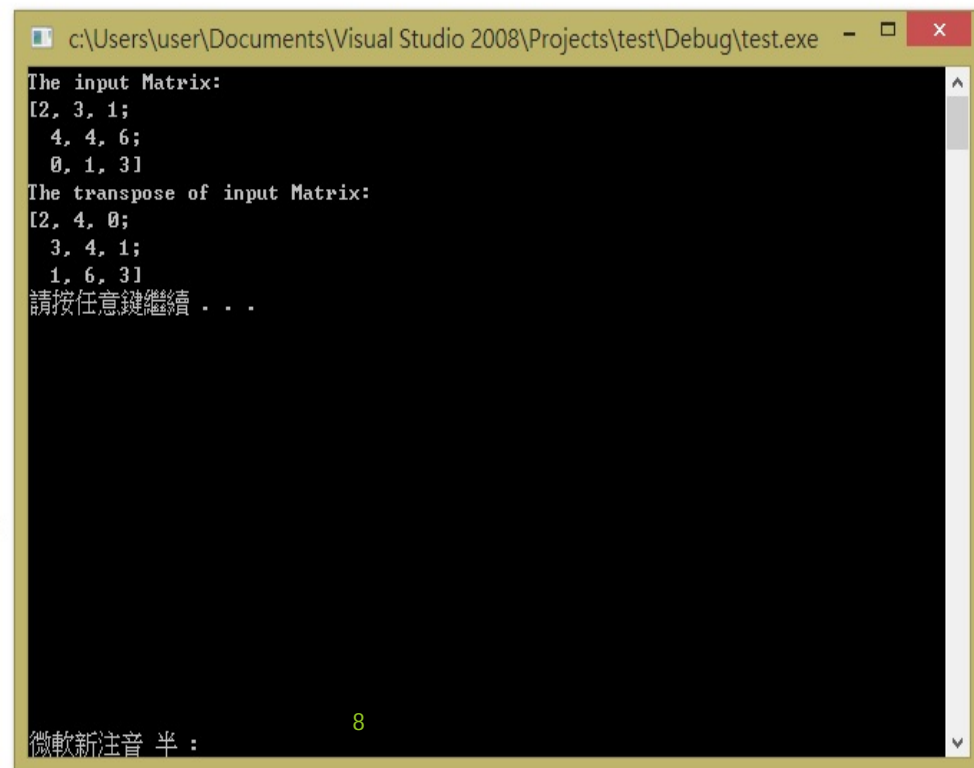
using namespace std;
using namespace cv;

int main(){
    Mat input(3,3,CV_32S,Scalar(0));

    input.at<int>(0,0) = 2;
    input.at<int>(0,1) = 3;
    input.at<int>(0,2) = 1;
    input.at<int>(1,0) = 4;
    input.at<int>(1,1) = 4;
    input.at<int>(1,2) = 6;
    input.at<int>(2,0) = 0;
    input.at<int>(2,1) = 1;
    input.at<int>(2,2) = 3;

    cout<<"The input Matrix: "<<endl<<input<<endl;
    cout<<"The transpose of input Matrix: "<<endl<<input.t()<<endl;

    system("pause");
    return 0;
}
```



```
c:\Users\user\Documents\Visual Studio 2008\Projects\test\Debug\test.exe
The input Matrix:
[2, 3, 1;
 4, 4, 6;
 0, 1, 3]
The transpose of input Matrix:
[2, 4, 0;
 3, 4, 1;
 1, 6, 3]
請按任意鍵繼續 . . .
```


矩陣操作

▶ 矩陣的運算

▶ 相加 $\rightarrow \text{Output} = \text{Input1} + \text{Input2}$

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} + \begin{array}{|c|c|} \hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 2 & 0 \\ \hline \end{array}$$

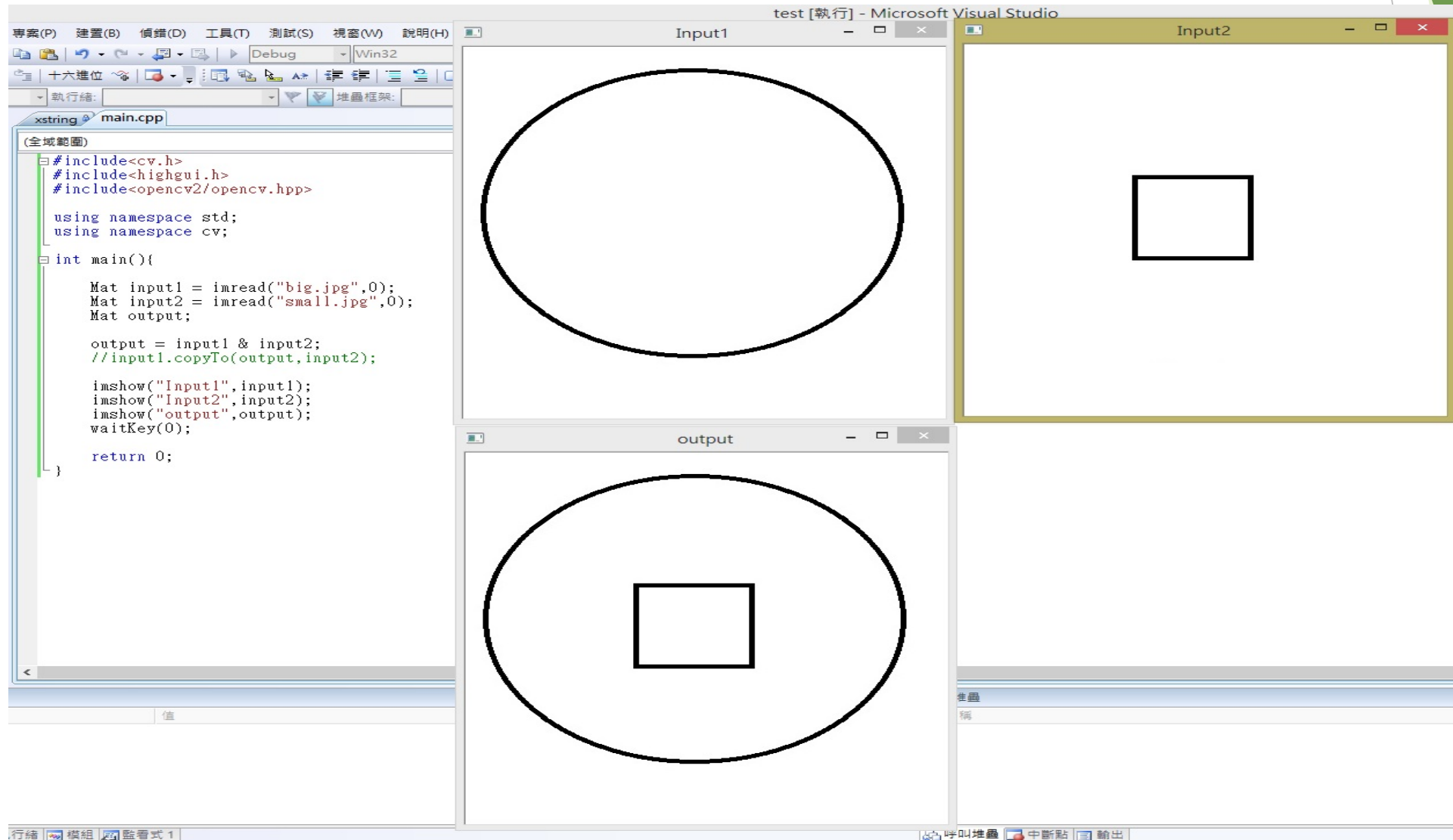
▶ AND $\rightarrow \text{Output} = \text{Input1} \& \text{Input2}$

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \& \begin{array}{|c|c|} \hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 1 & 0 \\ \hline \end{array}$$

▶ OR $\rightarrow \text{Output} = \text{Input1} \mid \text{Input2}$

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \mid \begin{array}{|c|c|} \hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 0 \\ \hline \end{array}$$

矩陣操作



畫圖

➤ Line

▶ `line(img, pt1, pt2, color, thickness=1, lineType=8, shift=0) ;`

想畫上線
的圖
(Mat)

線的第一
個點
(Point)

線的最後
一個點
(Point)

顏色
Scalar(B,G,R)
Scalar(Gray)

線的寬度
(int)

可省略

作業8-1

