Day 1 影像處理基礎入門

Outline

- Outline OpenCV 2基本操作
- Mat class基本介紹
- Point及vector資料結構

OpenCV 2基本操作

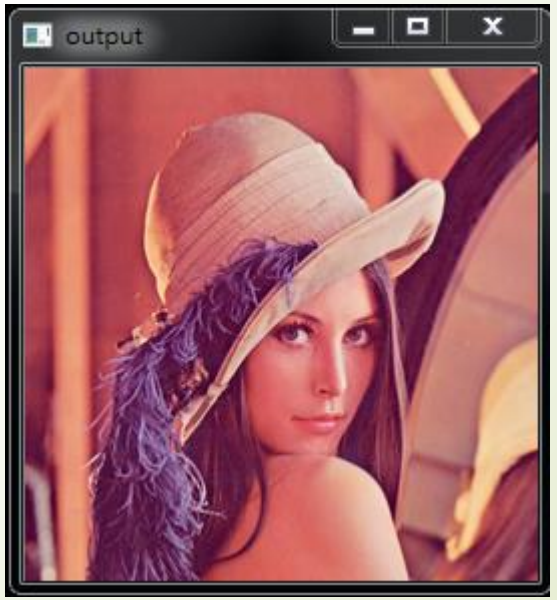
圖片讀取、顯示和存檔

- Mat input=imread("lena_24bit.bmp",1); //1表示彩圖(3 byte)
- Mat input2=imread("lena_24bit.bmp",0); //0表示灰階圖(1 bytes)

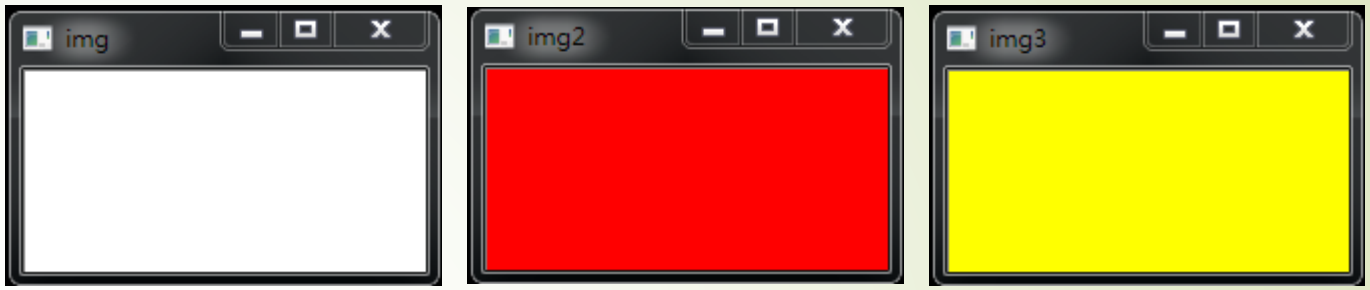
- namedWindow("output" ,1); //1:自動調整視窗大小 0:手動調整視窗大小
- imshow("output" , input); //在output視窗顯示image
- waitKey(500);//畫面顯示500毫秒後繼續執行程式碼

- imshow("output2" , input);
- waitKey(0);//0為按任一鍵繼續執行程式碼

- imwrite("test_img.bmp" ,input);



如何宣告



- int row=100; //圖片高度
- int col=200; // 圖片寬度

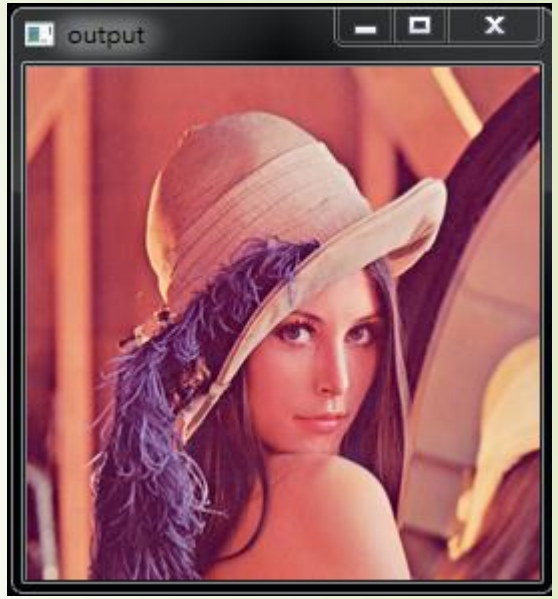
- Mat img(row, col, CV_8UC1, Scalar(255));
  - //CV_8UC1, CV_8UC2, CV_8UC3　　**Unsigned 8bits uchar 0~255**
  - //CV_32FC1，CV_32FC2，CV_32FC3　**Float 32bits float -1.18*10-38~3.40*10-38**
  - //CV_64FC1，CV_64FC2，CV_64FC3　**Double 64bits double**
- Size matrixsize = Size(col, row);
- Mat img2(matrixsize.height,matrixsize.width,CV_8UC3,Scalar(0,0,255));
- Mat img3(matrixsize,CV_8UC3,Scalar(0,255,255));

如何取值 (1/2)


```
Mat input=imread( "lena_24bit.bmp",0);
Mat output(input.size(), CV_8UC1, Scalar(0));
for (int row=0;row<input.rows;row++)
{
    for(int col=0;col<input.rows;row++)
    {
        output.at<uchar>(row,col)=input.at<uchar>(row,col);
    }
}
```

```
//CV_8UC1, CV_8UC2, CV_8UC3 uchar
//CV_32FC1 · CV_32FC2 · CV_32FC3 float
//CV_64FC1 · CV_64FC2 ·  double
```

如何取值 (2/2)




```
Mat input=imread( "lena_24bit.bmp",1);
Mat output(input.size(), CV_8UC3, Scalar(0,0,0));
for (int row=0;row<input.rows;row++)
{
    for(int col=0;col<input.cols;cols++)
    {
    for(int channel=0;channel<input.channels();channel++)
        output.at<Vec3b>(row,col)[channel]=input.at<Vec3b>(row,col)[channel];
    }
}
```

	C1	C2	C3
uchar	uchar	Vec2b	Vec3b
short	short	Vec2s	Vec3s
int	int	Vec2i	Vec3i
float	float	Vec2f	Vec3f
double	double	Vec2d	Vec3d

Point

```
point a:[0.33, 0]
point b:[0, 0.45]
point c:3, 4
```

- Point2f a(0.33,0.0),b(0.0,0.45);
- //Point_<float> a(0.33,0.0),b(0.0,0.45);
- //Point_<double> a(0.33,0.0),b(0.0,0.45); or Point2d a(0.33,0.0),b(0.0,0.45);
- Point c=(a+b)*10.0;
- //Point_<int> c=(a+b)*10.0;
- cout<<"point a:"<<a<<endl;
- cout<<"point a:"<<a<<endl;
- cout<<"point c:"<<c.x<<","<<c.y<<endl;

vector

- vector<int> sequence1; //empty vector
- vector<int> sequence2(5); // 5 stack 初始為0
- // vector<資料結構> 變數名稱(初始堆疊個數)
- for(int i=0;i<sequence2.size();i++)
- {
  - cout<<"初始 sequence2["<<i<<"] :"<<sequence2[i]<<endl;
  - sequence2[i]=i+1; //修改 i th stack值
  - cout<<"修改後 sequence2["<<i<<"] :"<<sequence2[i]<<endl;
  - sequence1.push_back(sequence2[i]); //push一個stack進去
  - cout<<"sequence1["<<i<<"] :"<<sequence1[i]<<endl;
- }

```
初始 sequence2[0] :0
修改後 sequence2[0] :1
sequence1[0] :1
初始 sequence2[1] :0
修改後 sequence2[1] :2
sequence1[1] :2
初始 sequence2[2] :0
修改後 sequence2[2] :3
sequence1[2] :3
初始 sequence2[3] :0
修改後 sequence2[3] :4
sequence1[3] :4
初始 sequence2[4] :0
修改後 sequence2[4] :5
sequence1[4] :5
```