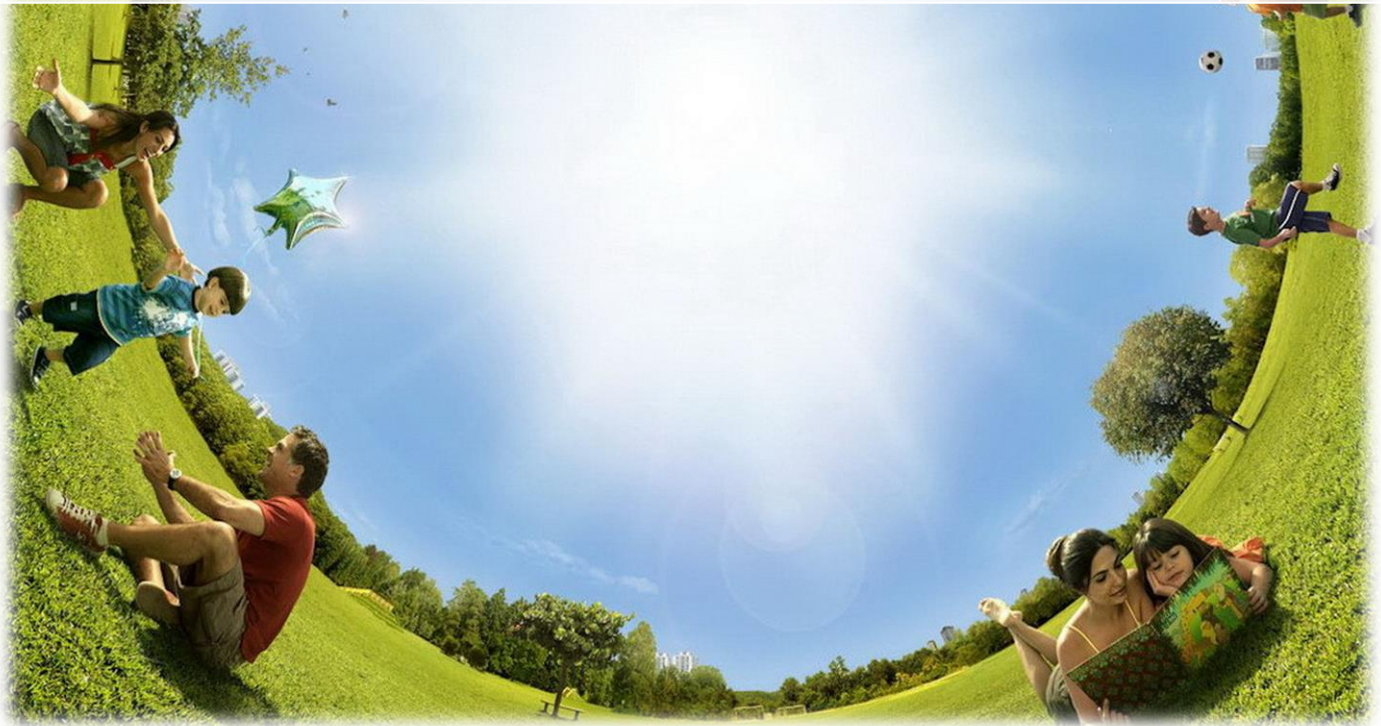


[Mã môn học] – [XLAS&VS]

Tháng 03/2018

# Giới thiệu OpenCV



Tài liệu này giới thiệu sinh viên làm quen với thư viện lập trình xử lý ảnh và thị giác máy tính OpenCV, hướng dẫn một số thao tác lập trình cơ bản truy xuất ảnh bằng OpenCV

Bộ môn **[Thị giác máy tính &  
Khoa học Robot]**  
Khoa Công nghệ thông tin  
ĐH Khoa học tự nhiên TP HCM



# MỤC LỤC

1. Hướng dẫn cài đặt và cấu hình .....	1
1.1. Bước 1: Download OpenCV.....	1
1.2. Bước 2: Giải nén.....	1
1.3. Bước 3: Cài đặt biến môi trường .....	1
1.4. Bước 4: Cấu hình project.....	2
1.5. Bước 5: Copy file dll vào thư mục chứa file exe.....	4
1.6. Bước 6: Chạy thử chương trình .....	4
2. Hướng dẫn thực hành OpenCV .....	5
2.1. Cấu trúc Mat .....	5
2.2. Khai báo ma trận.....	6
2.3. Các thao tác trên ma trận ảnh .....	6
Load ảnh .....	6
Lưu ảnh .....	7
Hiển thị ảnh lên cửa sổ .....	7
Truy xuất đến từng pixel trên ảnh.....	7
3. Tài liệu tham khảo .....	8

# GIỚI THIỆU OPENCV

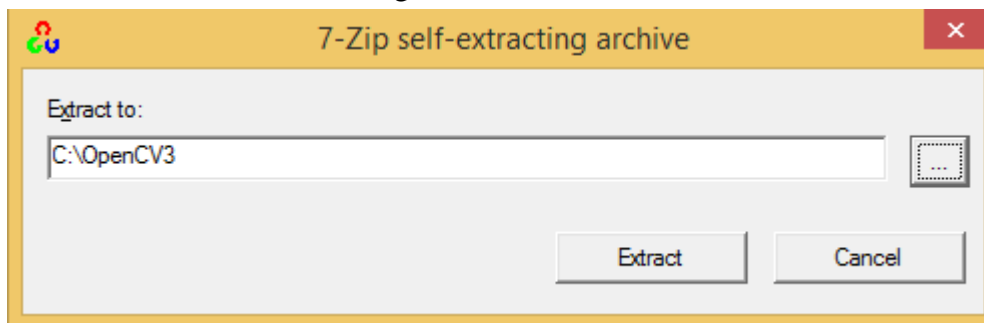
## 1. Hướng dẫn cài đặt và cấu hình

### 1.1. Bước 1: Download OpenCV

- Phiên bản: 3.4.1
- Download tại trang web <https://opencv.org/releases.html>

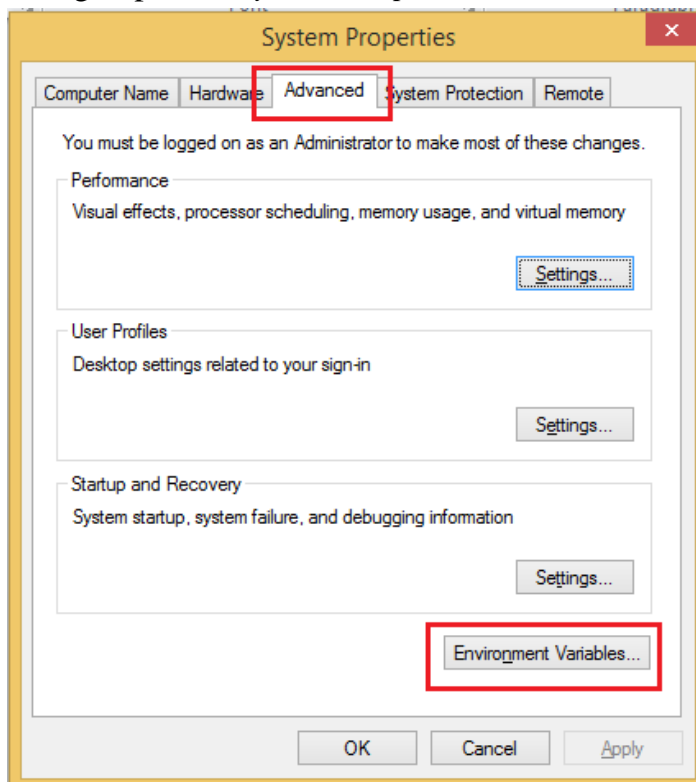
### 1.2. Bước 2: Giải nén

- Click vào file cài đặt để giải nén

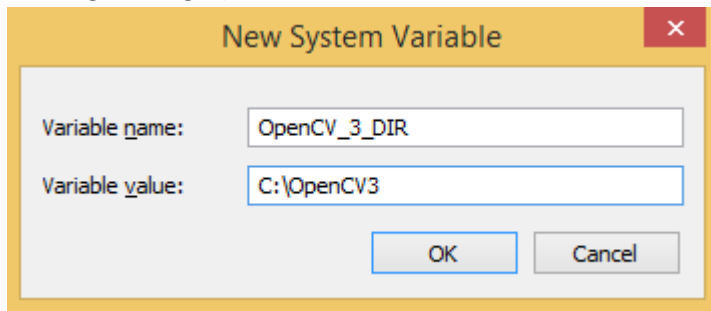


### 1.3. Bước 3: Cài đặt biến môi trường

- Click chuột phải vào Computer, chọn *Properties*
- Chọn tiếp *Advanced System Setting*
- Trong hộp thoại *System Properties*, chọn thẻ *Advance*, chọn tiếp *Environment Variables*



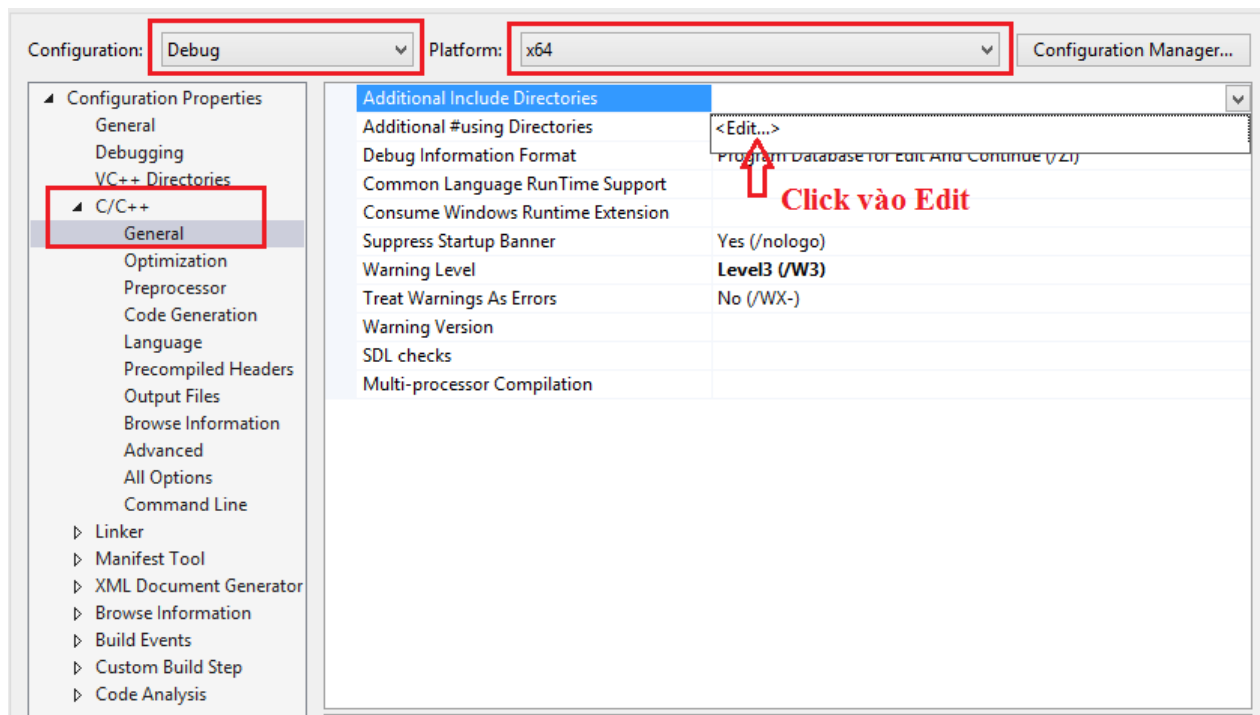
-Trong khung *System Variables*, chọn *New* và điền vào thông tin giống như hình

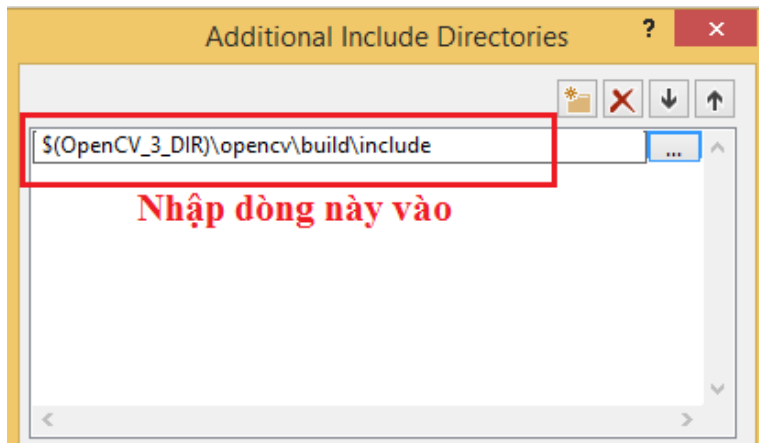


-Khởi động lại máy để hệ thống cập nhật biến môi trường

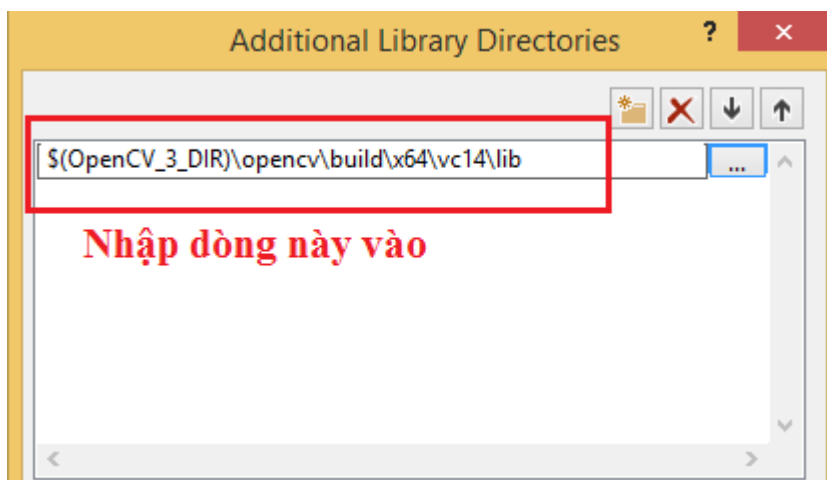
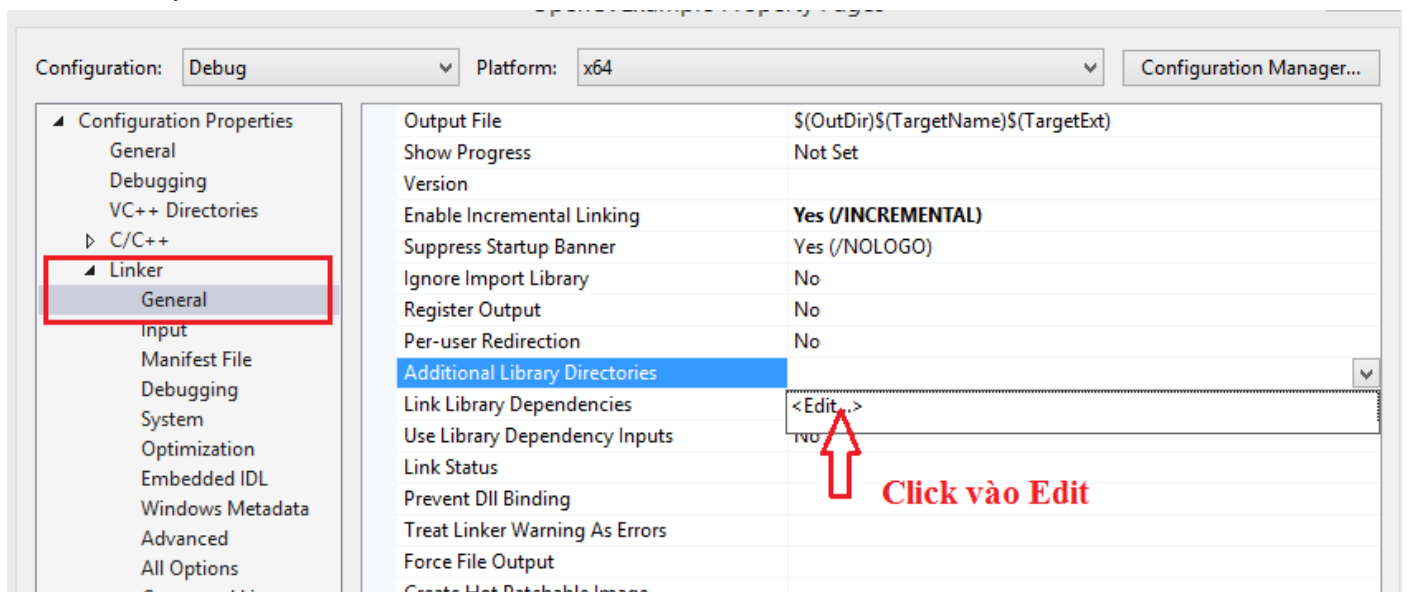
#### 1.4. Bước 4: Cấu hình project

- Khởi động *Visual Studio*, tạo project *Visual C++ Console Application*
- Click chuột phải vào project chọn *Properties*
- Cần chú ý chọn chế độ (*Configuration*) là *Debug* hay *Release* hoặc *Platform* là *x64* hay *Win32*
- Add thư mục Include

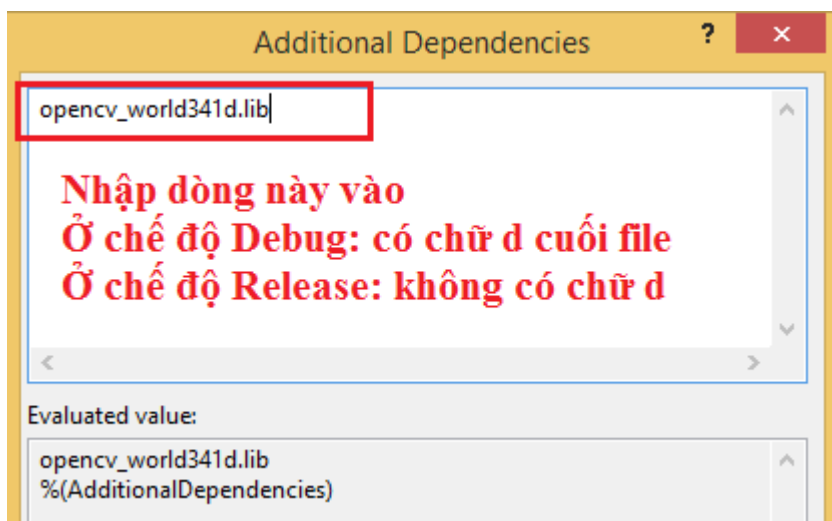
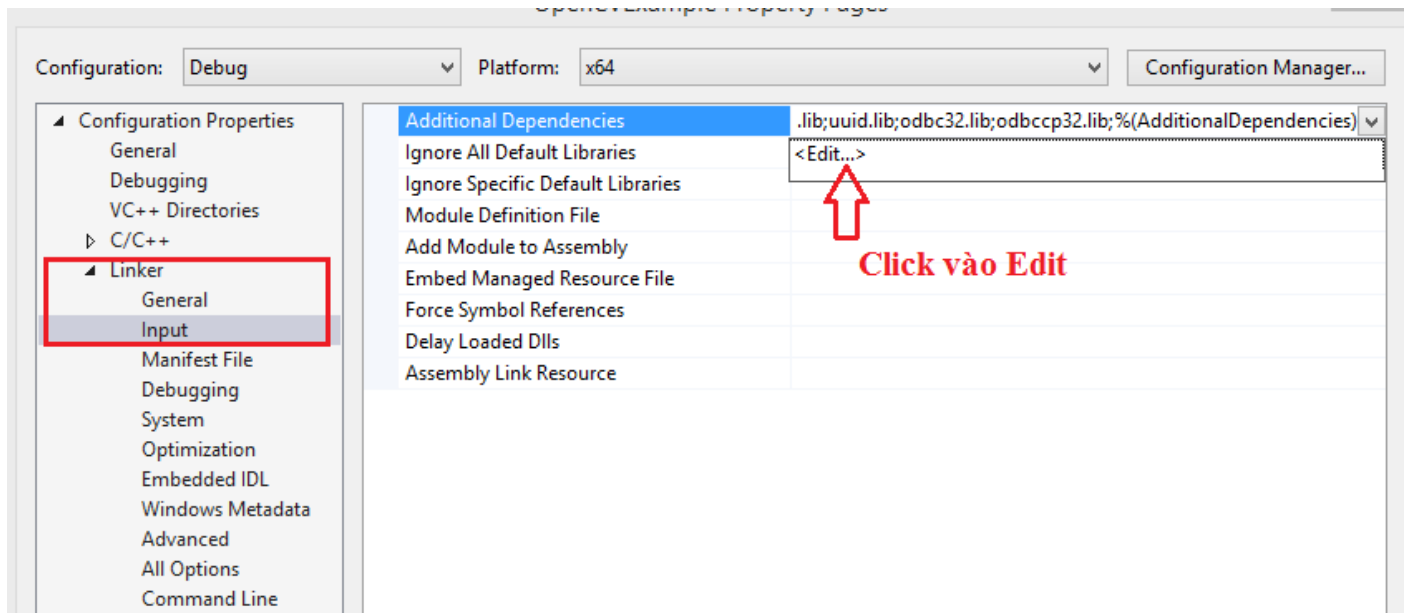




-Add thư mục chứa file lib



-Add file lib



- Nhấn *Apply*, nhấn *OK*
- Sau đó build project để tạo file exe

### 1.5. Bước 5: Copy file dll vào thư mục chứa file exe

- Copy các file *opencv\_world341d.dll* (có chữ d cuối file nếu chạy chế độ Debug) từ thư mục *C:\OpenCV3\opencv\build\x64\vc14\bin* vào thư mục chứa file exe.
- Nếu chạy chế độ Release thì copy file *opencv\_world341.dll* (không có chữ d cuối file)

### 1.6. Bước 6: Chạy thử chương trình

- Copy đoạn code sau vào file cpp chứa hàm main

```
#include "stdafx.h"
#include "opencv2/opencv.hpp"
#include "opencv2/highgui/highgui.hpp"
using namespace cv;
```



```
int main()
{
    Mat image = imread("D:\\lena.png", CV_LOAD_IMAGE_COLOR);

    namedWindow("Show Image");
    imshow("Show Image", image);

    waitKey(0);
    return 0;
}
```

- Nếu chương trình show được ảnh lên thành công, xem như đã cài đặt và cấu hình xong cho OpenCV

## 2. Hướng dẫn thực hành OpenCV

Bắt đầu từ phiên bản OpenCV 2.0 trở lên, ảnh và ma trận được lưu chung bằng cấu trúc Mat

### 2.1. Cấu trúc Mat

```
class CV_EXPORTS Mat
{
public:
    // ... a lot of methods ...
    ...

    /*! includes several bit-fields:
        - the magic signature
        - continuity flag
        - depth
        - number of channels
    */
    int flags;
    /// the array dimensionality, >= 2
    int dims;
    /// the number of rows and columns or (-1, -1) when the array has more than 2 dimensions
    int rows, cols;
    /// pointer to the data
    uchar* data;

    /// pointer to the reference counter;
    // when array points to user-allocated data, the pointer is NULL
    int* refcount;

    // other members
    ...
};
```

## 2.2. Khai báo ma trận

```
Mat::Mat();//khai báo ma trận rỗng
Mat::Mat(int rows, int cols, int type);// khai báo ma trận theo số dòng, số cột
Mat::Mat(Size size, int type);//khai báo ma trận theo kích thước
Mat::Mat(int rows, int cols, int type, const Scalar& s);//khai báo ma trận có giá trị
mặc định s
```

Giải thích tham số:

- rows: số dòng ma trận tương ứng chiều dài của ảnh
- cols: số cột ma trận tương ứng chiều rộng của ảnh
- type: kiểu dữ liệu của ma trận được viết theo cấu trúc sau

CV\_<số\_bit><Kiểu\_dữ\_liệu><số\_kênh\_màu>

- + Số bit: số bit trong 1 kênh màu: VD: 8bit, 16bit, 32bit,...
- + Kiểu dữ liệu: kiểu dữ liệu của từng phần tử trong ma trận tương ứng (UC: kiểu unsigned, SC: kiểu signed, FC: kiểu số thực).

- + Số kênh màu: số lượng kênh màu trong ảnh (giá trị từ 1 đến 4)

- VD:

- + CV\_8UC1: ảnh có 1 kênh màu, 8 bit mỗi kênh màu, mỗi phần tử kiểu unsigned char.
- + CV\_8UC3: ảnh có 3 kênh màu, 8 bit mỗi kênh màu, mỗi phần tử kiểu unsigned char.
- + CV\_16SC1: ảnh 1 kênh màu, 16 bit mỗi kênh màu, mỗi phần tử kiểu signed short.
- + CV\_32FC3: ảnh 3 kênh màu, 32 bit mỗi kênh màu, mỗi phần tử kiểu float.

- Ví dụ khai báo ma trận

```
//Ma trận a có kích thước 100x100, 1 kênh màu, mỗi phần tử unsigned char, ảnh xám
Mat a = Mat(100, 100, CV_8UC1);

//Ma trận a có kích thước 200x200, 3 kênh màu, mỗi phần tử unsigned char, ảnh màu RGB
Mat a = Mat(200, 200, CV_8UC3, Scalar(0));

// Ma trận a có kích thước 120x240, 1 kênh màu, mỗi phần tử là số thực 32bit (float)
Mat a = Mat(Size(120, 240), CV_32FC1);

// Ma trận a có kích thước 120x240, 1 kênh màu, mỗi phần tử là số thực 64bit (double)
Mat a = Mat(Size(120, 240), CV_64FC1);
```

## 2.3. Các thao tác trên ma trận ảnh

### Load ảnh

```
//load ảnh màu
Mat image = imread("filename.jpg", CV_LOAD_IMAGE_COLOR);

//load ảnh xám
Mat image = imread("filename.jpg", CV_LOAD_IMAGE_GRAYSCALE);
```



## Lưu ảnh

```
//lưu đối tượng image xuống file
imwrite("filename.jpg", image);
```

## Hiển thị ảnh lên cửa sổ

```
//tạo cửa sổ với tên "Show Image"
namedWindow("Show Image");
//hiển thị ảnh lên cửa sổ "Show Image"
imshow("Show Image", image);
//chờ người dùng nhấn phím để đóng cửa sổ
waitKey(0);
```

## Truy xuất đến từng pixel trên ảnh

OpenCV lưu ma trận ảnh xám

	Column 0	Column 1	Column ...	Column m
Row 0	0,0	0,1	...	0, m
Row 1	1,0	1,1	...	1, m
Row ...	...,0	...,1	...	..., m
Row n	n,0	n,1	n,...	n, m

OpenCV lưu ma trận ảnh màu theo thứ tự BGR (màu Blue trước, màu Green, rồi đến màu Red)

	Column 0			Column 1			Column ...			Column m		
Row 0	0,0	0,0	0,0	0,1	0,1	0,1	...	...	...	0, m	0, m	0, m
Row 1	1,0	1,0	1,0	1,1	1,1	1,1	...	...	...	1, m	1, m	1, m
Row ...	...,0	...,0	...,0	...,1	...,1	...,1	...	...	...	..., m	..., m	..., m
Row n	n,0	n,0	n,0	n,1	n,1	n,1	n,...	n,...	n,...	n, m	n, m	n, m

## Cách 1 : dùng câu lệnh at

```
void salt(cv::Mat &image, int n) {
    for (int k = 0; k < n; k++) {
        int i = rand() % image.cols; //tương ứng tọa độ phương ngang x của ảnh
        int j = rand() % image.rows; //tương ứng tọa độ phương ngang y của ảnh
        if (image.channels() == 1) { // gray-level image
            //truy xuất đến tọa độ ảnh (i,j)
            image.at<uchar>(j, i) = 255;
        }
        else if (image.channels() == 3) { // color image
            //truy xuất đến tọa độ ảnh (i,j) kênh màu 0 (blue)
            image.at<cv::Vec3b>(j, i)[0] = 255;
            //truy xuất đến tọa độ ảnh (i,j) kênh màu 1 (green)
            image.at<cv::Vec3b>(j, i)[1] = 255;
            //truy xuất đến tọa độ ảnh (i,j) kênh màu 2 (red)
            image.at<cv::Vec3b>(j, i)[2] = 255;
        }
    }
}
```

```
}  
}
```

### Cách 2: dùng con trỏ

```
void colorReduce(cv::Mat &image, int div = 64) {  
    int nl = image.rows; // number of lines  
    int nc = image.cols * image.channels();  
    for (int j = 0; j<nl; j++) {  
        // lấy địa chỉ của dòng thứ j  
        uchar* data = image.ptr<uchar>(j);  
        for (int i = 0; i<nc; i++) {  
            // xử lý trên mỗi pixel  
            data[i] = data[i] / div*div + div / 2;  
        } // end of line  
    }  
}
```

### 3. Tài liệu tham khảo

- The OpenCV Tutorial
- The OpenCV Reference Manual.
- OpenCV 2 Computer Vision Application Programming Cookbook
- Learning OpenCV