

OpenCV- Introduction

Introduction

Computer vision which goes beyond image processing, helps to obtain relevant information from images and make decisions based on that information. In other words, computer vision is making the computer see as humans do. Basic steps for a typical computer vision application as follows.

1. Image acquisition
2. Image manipulation
3. Obtaining relevant information
4. Decision making

OpenCV is an open source C++ library for image processing and computer vision. It is free for both commercial and non-commercial use. Therefore, you can use the OpenCV library even for your commercial applications. It is a library mainly aimed at real time processing.

Key Features:

- Optimized for real time image processing & computer vision applications
- Primary interface of OpenCV is in C++
- There are also C, Python and JAVA full interfaces
- OpenCV applications run on Windows, Android, Linux, Mac and iOS
- Optimized for Intel (and AMD) processors

C++ API

Header files

You will need to include only the `opencv2/opencv.hpp` header file in your program. That header file will include all the other necessary header files for your application. Therefore, you do not need to bother thinking which header files should be included for your program anymore.

```
#include <opencv2/opencv.hpp>
```

Namespace

All OpenCV classes and functions are in `cv` namespace. So, you have to do one of the following

Add the using

```
#include <opencv2/opencv.hpp>
using namespace cv;
int main(int argc, char** argv)
{
    // Read the image file
    Mat image = imread("D:/My OpenCV Website/Eagle.jpg");
    return 0;
}
```

```
#include <opencv2/opencv.hpp>
int main(int argc, char** argv)
{
    // Read the image file
    cv::Mat image = cv::imread("D:/My OpenCV Website/Eagle.jpg");
}
```

```
    return 0;
}
```

Data Types of Arrays

Data type of an array defines the number of channels, the number of bits allocated for each element and how the value of an element is represented using those bits. If an array represents an image, each elements of the array are pixels of the image.

Any single channel array should belong to one of following data types.

- CV_8U - 8 bit unsigned integer
- CV_8S - 8 bit signed integer
- CV_16U - 16 bit unsigned integer
- CV_16S - 16 bit signed integer
- CV_32S - 32 bit signed integer
- CV_32F - 32 bit floating point number
- CV_64F - 64 bit float floating point number

We can define all of above data types for multi-channel arrays. OpenCV supports up to 512 channels. Here I am going to show you how to define CV_8U data type for multi-channel arrays.

- CV_8UC1 - Single channel array with 8 bit unsigned integers which is exactly same as CV_8U
- CV_8UC2 - 2 channel array with 8 bit unsigned integers
- CV_8UC3 - 3 channel array with 8 bit unsigned integers
- CV_8UC4 - 4 channel array with 8 bit unsigned integers
- CV_8UC(n) - n channel array with 8 bit unsigned integers (n can be from 1 to 512))