

# OpenCV Tutorial C++

[Home](#) [OpenCV Lessons](#) [Reference Books](#) [About me](#)

## Write Image & Video to File

In this lesson, I am going to show you how to write images and videos to a file using OpenCV.  
If you have not install and configure OpenCV yet, please refer to [Installing & Configuring with Visual Studio](#).

### Write Image to File

In the following example, a yellow image is created and written into a file. Let's see how to do it with OpenCV.

```
////////////////////////////////////

#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace cv;
using namespace std;

int main( int argc, const char** argv )
{
    Mat img(650, 600, CV_16UC3, Scalar(0,50000, 50000)); //create an image ( 3 channels, 16 bit image depth, 650 high, 600 wide, (0,
    50000, 50000) assigned for Blue, Green and Red plane respectively. )

    if (img.empty()) //check whether the image is loaded or not
    {
        cout << "ERROR : Image cannot be loaded..!!" << endl;
        //system("pause"); //wait for a key press
        return -1;
    }

    vector<int> compression_params; //vector that stores the compression parameters of the image

    compression_params.push_back(CV_IMWRITE_JPEG_QUALITY); //specify the compression technique

    compression_params.push_back(98); //specify the compression quality

    bool bSuccess = imwrite("D:/TestImage.jpg", img, compression_params); //write the image to file

    if ( !bSuccess )
    {
        cout << "ERROR : Failed to save the image" << endl;
        //system("pause"); //wait for a key press
    }

    namedWindow("MyWindow", CV_WINDOW_AUTOSIZE); //create a window with the name "MyWindow"
    imshow("MyWindow", img); //display the image which is stored in the 'img' in the "MyWindow" window

    waitKey(0); //wait for a keypress

    destroyWindow("MyWindow"); //destroy the window with the name, "MyWindow"

    return 0;
}
////////////////////////////////////
```

You can download this OpenCV visual c++ project from [here](#). (The downloaded file is a compressed .rar folder. So, you have to extract it using [Winrar](#) or other suitable software)

The above program is very much similar to the program under 'Create a Blank Image & Display' section in the lesson of [Read & Display](#)

#### SITE MAP

[Home](#)

#### OpenCV Lessons

.. [What is OpenCV?](#)  
 .. [Installing & Configuring v](#)  
 .. [Basics of OpenCV API](#)  
 .. [Read & Display Image](#)  
 .. [Capture Video from File o](#)  
 .. [Write Image & Video to Fi](#)  
 .. [Filtering Images](#)  
 .. [Change Brightness of Im](#)  
 .. [Change Contrast of Image](#)  
 .. [Histogram Equalization](#)  
 .. [Smooth / Blur Images](#)  
 .. [How to Add Trackbar](#)  
 .. [How to Detect Mouse Clic](#)  
 .. [Rotate Image & Video](#)  
 .. [Color Detection & Object](#)  
 .. [Shape Detection & Trackir](#)

#### Reference Books

#### About Me

#### GOOGLE+ FOLLOWERS

OpenCV Tutorials

Follow



639 have us in circles

782

#### FACEBOOK FOLLOWERS

[Like](#) [Share](#) 2,256 people  
what your friends think

#### SEARCH THIS BLOG

Image. If you need further clarifications of any OpenCV functions which are not explained here, please refer to Read & Display Image lesson.

#### New OpenCV functions which are not found earlier

- **bool** `imwrite( const string& filename, InputArray img, const vector<int>& params=vector<int>())`

The function saves the image in the variable '**img**' to a file, specified by '**filename**'. If this function fails to save the image, it will return false. On success of writing the file to the harddisk, it will return true.

Parameters -

- **filename** - specify the location and name of the file to be saved
- **img** - hold the image which is going to save
- **params** - This is a int vector to which you have to insert some int parameters specifying the format of the image
  - **JPEG format** - You have to push\_back **CV\_IMWRITE\_JPEG\_QUALITY** first and then a number between 0 and 100 (higher is the better). If you want the best quality output, use 100. I have used 98 in the above sample program. But higher the value, it will take longer time to write the image
  - **PNG format** - You have to push\_back **CV\_IMWRITE\_PNG\_COMPRESSION** first and then a number between 0 and 9 (higher is the better compression, but slower).

The image format is chosen depending on the file name extension. Only images with 8 bit or 16 bit unsigned single channel or 3 channel ( CV\_8UC1, CV\_8UC3, CV\_8SC1, CV\_8SC3, CV\_16UC1, CV\_16UC3) with '**BGR**' channel order, can be saved. If the depth or channel order of the image is different, use '**Mat::convertTo()**' or '**cvtColor**' functions to convert the image to supporting format before using **imwrite** function.

The above program is very much similar to the program under 'Create a Blank Image & Display' section in the lesson of Read & Display Image. If you need further clarifications of any OpenCV functions which are not explained here, please refer to Read & Display Image lesson.

#### Summary

This program creates an yellowish image ( 3 channels, 16 bit image depth, 650 high, 600 wide, (0, 50,000, 50,000) assigned for BGR channels). Because the image has 16 bit depth, you can use values between 0 and 65535 for each element in each channel. I have used 50,000 for each element in the green and red planes which gives the yellowish color. You can try different values.

Then it specifies the compressing technique. I have used JPEG as the compression technique in the above example. Then it saves the image in the "**D:/TestImage.jpg**" location. Then it displays the newly created image in a window and wait indefinitely until an user presses a key.

## Write Video to File

In the following example, a video is captured from the webcam and written into a file. Let's see how to do it with OpenCV.

```

////////////////////////////////////
#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace cv;
using namespace std;

int main(int argc, char* argv[])
{
    VideoCapture cap(0); // open the video camera no. 0

    if (!cap.isOpened()) // if not success, exit program
    {
        cout << "ERROR: Cannot open the video file" << endl;
        return -1;
    }

    namedWindow("MyVideo", CV_WINDOW_AUTOSIZE); //create a window called "MyVideo"

    double dWidth = cap.get(CV_CAP_PROP_FRAME_WIDTH); //get the width of frames of the video
    double dHeight = cap.get(CV_CAP_PROP_FRAME_HEIGHT); //get the height of frames of the video

    cout << "Frame Size = " << dWidth << "x" << dHeight << endl;

    Size frameSize(static_cast<int>(dWidth), static_cast<int>(dHeight));

    VideoWriter oVideoWriter ("D:/MyVideo.avi", CV_FOURCC('P','I','M','I'), 20, frameSize, true); //initialize the VideoWriter object

    if ( !oVideoWriter.isOpened() ) //if not initialize the VideoWriter successfully, exit the program
    {
        cout << "ERROR: Failed to write the video" << endl;
        return -1;
    }

    while (1)
    {
        Mat frame;

```

```

bool bSuccess = cap.read(frame); // read a new frame from video

if (!bSuccess) //if not success, break loop
{
    cout << "ERROR: Cannot read a frame from video file" << endl;
    break;
}

oVideoWriter.write(frame); //writer the frame into the file

imshow("MyVideo", frame); //show the frame in "MyVideo" window

if (waitKey(10) == 27) //wait for 'esc' key press for 30ms. If 'esc' key is pressed, break loop
{
    cout << "esc key is pressed by user" << endl;
    break;
}
}

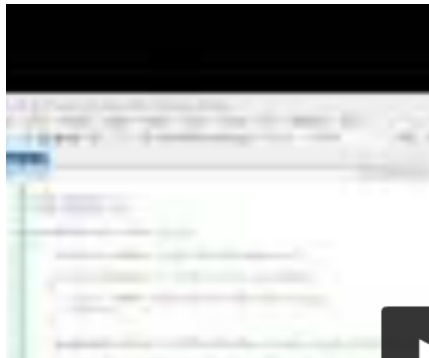
return 0;
}

```

////////////////////////////////////

You can download this OpenCV visual c++ project from [here](#). (The downloaded file is a compressed .rar folder. So, you have to extract it using [Winrar](#) or other suitable software)

Here is how I record my video from my webcam and save to a file in D drive



New OpenCV functions which are not found earlier are explained here

- **Size** frameSize(**static\_cast<int>**(dWidth), **static\_cast<int>**(dHeight))

Create a **Size** object with a given width and height. Here I have cast the width and height to integers because they are originally double values and the **Size** constructor does not accept double values as its parameters.

- **VideoWriter::VideoWriter(const string& filename, int fourcc, double fps, Size frameSize, bool isColor=true)**

This is the constructor of the VideoWriter class. It initializes the object with following parameters

- **const string& filename** - Specify the name and the location of the output file. The video stream is written into this file
- **int fourcc** - specify the 4 character code for the codec which is used to compress the video. Your computer may not be supported some codecs. So, if you fail to save the video, please try other codecs. Here are some popular codecs.
  - CV\_FOURCC('D', 'I', 'V', '3') for DivX MPEG-4 codec
  - CV\_FOURCC('M', 'P', '4', '2') for MPEG-4 codec
  - CV\_FOURCC('D', 'I', 'V', 'X') for DivX codec
  - CV\_FOURCC('P', 'I', 'M', '1') for MPEG-1 codec
  - CV\_FOURCC('I', '2', '6', '3') for ITU H.263 codec
  - CV\_FOURCC('M', 'P', 'E', 'G') for MPEG-1 codec

Complete list of codec can be found [here](#)

Here I have used **CV\_FOURCC('P', 'I', 'M', '1')** as the four character code of codec to compress the video. If the output file cannot be opened, you can try different four character code of codecs.

For **Windows users**, it is possible to use -1 instead of the above codecs in order to choose compression method and additional compression parameters from a dialog box. It is a best method for Microsoft Windows users.

- **double fps** - frames per seconds of the video stream. I have used 20. You can try different values. But the codec should support the fps value. So, use an appropriate value.
- **Size frameSize** - Size object which specify the width and the height of each frame of the video stream.
- **bool isColor** - If you want to save a color video, pass the value as true. Otherwise false. Remember codec should support whatever value, you pass. In the above example, you have to give true as the 5th argument. Otherwise it will not work.

- **if ( !oVideoWriter.isOpened() )**

Check whether the VideoWriter object initialize successfully. If not exit the program immediately.

- **void write(const Mat& image)**

Write a frame to the video stream. The size of the frame should be same as the size you specified when initializing the VideoWriter object.

All other OpenCV functions have been discussed in earlier lessons. So, if you are not familiar with those OpenCV functions yet, please go through [Capture Video from File or Camera](#)

## Next Lesson: Filtering Images

## Previous Lesson: Capture Video from File or Camera

Posted by Shermal Fernando



+5 Recommend this on Google

Is This Helpful :      Yes (1)      No (0)

## 18 comments:



**LeGoek** June 13, 2013 at 2:02 AM

In your post you give the image a fixed file name. But let's say I run an opencv operation in a loop, like corner detection three times. Do you have an idea to have files named like: NewPic01.jpg, NewPic02.jpg and NewPic03?

[Reply](#)

[Replies](#)



**Shermal Fernando**

June 13, 2013 at 8:57 AM

Yes, you can change your file name in a loop. You have to concatenate "NewPic" and your loop variable to create your file name.

[Reply](#)



**h2w** July 17, 2013 at 4:58 PM

Typo : If I am not mistaken, the scalar value of your created image (0, 500 000, 500 000) in your code exceeded the size limit of 16 bit unsigned integer but that value in your explanation is correct. :)

[Reply](#)

[Replies](#)



**Shermal Fernando**

July 18, 2013 at 9:12 AM

Thank you very much. It was a mistake. I will correct it in the code.

[Reply](#)



**Anonymous** July 18, 2013 at 12:16 AM

man this is a great ah...,

[Reply](#)



**Anonymous** August 19, 2013 at 12:41 PM

The video plays faster then what it captured! In other words, it is in speed motion!

[Reply](#)

[Replies](#)



**Anonymous** August 19, 2013 at 6:12 PM

try installing different codecs specially if your in windows environment

[Reply](#)

**lill altaweel** September 29, 2013 at 1:56 AM