# Learn OpenCV by Examples

OpenCV simplified for beginners by the use of examples. Learn OpenCV with basic implementation of different algorithms.



## 2D Convolution / Creating new filter

OpenCV function filter2D is used to create new linear filters.

void **filter2D**(InputArray src, OutputArray dst, int ddepth, InputArray kernel, Point anchor=Point(-1,-1), double delta=0, int borderType=BORDER_DEFAULT )
Parameters:

- **src** – input image.
- **dst** – output image of the same size and the same number of channels as `src`.
- **ddepth** – desired depth of the destination image; if it is *negative*, it will be the same as `src.depth()`; the following combinations of `src.depth()` and `ddepth` are supported:
    - `src.depth()` = `CV_8U`, `ddepth` = -1/`CV_16S`/`CV_32F`/`CV_64F`
    - `src.depth()` = `CV_16U`/`CV_16S`, `ddepth` = -1/`CV_32F`/`CV_64F`
    - `src.depth()` = `CV_32F`, `ddepth` = -1/`CV_32F`/`CV_64F`
    - `src.depth()` = `CV_64F`, `ddepth` = -1/`CV_64F`

  when `ddepth=-1`, the output image will have the same depth as the source.
- **kernel** – convolution kernel (or rather a correlation kernel), a single-channel floating point matrix; if you want to apply different kernels to different channels, process

SEARCH CONTENTS OF THIS BLOG

Search

POPULAR POSTS

Find Contour

Basic drawing examples

Line Detection by Hough Line Transform

Face Detection using Haar-Cascade Classifier

5 Perspective Transform

6 Sobel Edge Detection

them individually.

- **anchor** – anchor of the kernel that indicates the relative position of a filtered point within the kernel; the anchor should lie within the kernel; default value (-1,-1) means that the anchor is at the kernel center.
- **delta** – optional value added to the filtered pixels before storing them in `dst`.
- **borderType** – pixel extrapolation method (see `borderInterpolate()` for details).

A kernel is a fixed size array of numerical coefficients along with an *anchor point* in that array.

The code provided below is slight modification of code provided in OpenCV documentation.
**Steps:**

1. Load image
2. Create a kernel to convolve with the input matrix ( here all elements of kernel is equal; so performs a low pass filter operation)
3. Apply convolution (filter2D)
4. Draw contours

**Functions:**
filter2D, imshow, imread, waitKey.

## Example:

---

```
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <stdlib.h>
#include <stdio.h>

using namespace cv;

void conv2(Mat src, int kernel_size)
{
    Mat dst,kernel;
    kernel = Mat::ones( kernel_size, kernel_size, CV_32F )/ (float)

    /// Apply filter
    filter2D(src, dst, -1 , kernel, Point( -1, -1 ), 0, BORDER_DEFA
    namedWindow( "filter2D Demo", CV_WINDOW_AUTOSIZE );imshow( "fil
}

int main ( int argc, char** argv )
{
    Mat src;

    /// Load an image
    src = imread( "1.jpg" );
```



## CATEGORIES

```
24      if( !src.data )  { return -1; }
25
26      conv2(src,3);
27
28      waitKey(0);
29      return 0;
30  }
```

---



Labels: Filter

# No comments:

# Post a Comment



Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)