

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

影像卷積(Convolution)

濾波(filtering)是影像處理的一個基本操作，目的在選擇性的提取重要訊息，用在影像銳化、去除雜訊或提取感興趣的視覺特徵，這類處理需尋訪影像每個像素，從此像素和相鄰像素得到新的像素值。

核心(kernel)：基本上核心是一個固定大小，其中心為錨點(anchor point)的二維矩陣，以下為一高斯濾波的核心，大小和矩陣數值依需求而變。

0	-1	0
-1	4	-1
0	-1	0

卷積(convolution)：是核心與圖的每個重疊像素間的運算，目的是要計算影像中某個位置的結果，主要有以下4個步驟：

1. 將核心的錨點放在輸入圖的某個像素上。
2. 各鄰近像素乘上相對應的核心係數後加總。
3. 將計算結果放在輸出圖的錨點像素上。
4. 對輸入圖的每個像素重複進行以上動作。

我們用以上的核心和卷積的概念，對一張灰階8位元影像進行銳化，我們將輸出圖和來源圖的第一行、第一列、最後一行、最後一列不做處理，所以尋訪像素時留了1像素的寬度，使用ptr0函式讀取指定列的第一個像素，主要有3個步驟：

1. 先複製圖案，讓不處理的第一行、第一列、最後一行、最後一列，輸出圖和輸入圖像素值相同。
2. 接著使用四個指標，前三個指標指向來源圖上中下列，第四個指向輸出圖目前處理的列數，之後逐列逐列進行濾波計算。

3. 用`saturate_cast`將每次的計算結果限定在合理範圍，以本例來說就是從0到255，超過255會設定成255，小於0會設定成0。

OpenCV 限定合理範圍：`template< ... > _Tp saturate_cast(_Tp2 v)`

- `v`：輸入參數，會讓此值在合理範圍。
- `saturate_cast`使用模板，所以呼叫時要指定像素深度。

影像卷積

```
#include <cstdio>
#include <opencv2/opencv.hpp>
using namespace cv;

void sharpen(const Mat &src, Mat &dst);

int main(){
    Mat src = imread("lena.jpg", CV_LOAD_IMAGE_UNCHANGED);
    Mat dst;
    sharpen(src, dst);

    namedWindow("window1");
    imshow("window1", src);
    imshow("window2", dst);
    waitKey(0);

    return 0;
}

void sharpen(const Mat &src, Mat &dst){
    src.copyTo(dst);
    const int nChannels = src.channels();
    int heightLimit = src.rows - 1;
    int widthLimit = nChannels * (src.cols-1);
    for(int iH=1; iH<heightLimit; iH++){
        const uchar *prePtr = src.ptr<const uchar>(iH-1);
        const uchar *curPtr = src.ptr<const uchar>(iH);
        const uchar *nextPtr = src.ptr<const uchar>(iH+1);
        uchar *dstPtr = dst.ptr<uchar>(iH);
        for(int iW=nChannels; iW<widthLimit; iW++){
            dstPtr[iW] = saturate_cast<uchar>(5*curPtr[iW]-curPtr[iW-
nChannels]-curPtr[iW+nChannels]-prePtr[iW]-nextPtr[iW]);
        }
    }
}
```



掃描影像

OpenCV提供`filter2D`函式方便濾波計算，我們依濾波值定義一個核矩陣，之後呼叫`filter2D`函式時輸入此核矩陣當參數。

```
void filter2D(InputArray src, OutputArray dst, int ddepth, InputArray kernel,  
Point anchor=Point(-1,-1), double delta=0, intborderType=BORDER_DEFAULT)
```

- `src`：輸入圖。
- `dst`：輸出圖，和輸入圖的尺寸、通道數相同。
- `ddepth`：輸出圖深度。
- `kernel`：使用的核心。
- `anchor`：錨點，預設為核心中央。

上面的卷積函式可改寫成以下函式，兩者結果相同：

```
#include <cstdio>
#include <opencv2/opencv.hpp>
using namespace cv;

void sharpen2(const Mat &src, Mat &dst);

int main(){
    Mat src = imread("lena.jpg",CV_LOAD_IMAGE_UNCHANGED);
    Mat dst;
    sharpen2(src, dst);

    namedWindow("window1");
    imshow("window1", src);
    imshow("window2", dst);
    waitKey(0);

    return 0;
}

void sharpen2(const Mat &src, Mat &dst){
    Mat kernel(3,3,CV_32F,Scalar(0));
    kernel.at<float>(1,1) = 5.0;
    kernel.at<float>(0,1) = -1.0;
    kernel.at<float>(2,1) = -1.0;
    kernel.at<float>(1,0) = -1.0;
    kernel.at<float>(1,2) = -1.0;
    filter2D(src,dst,src.depth(),kernel);
}
```





[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教學](#)

📅 2015-11-25 👤 阿宅 📁 OpenCV, 影像操作 🔖 Convolution, filter2D, 影像卷積

0 Comments

猴子遇到0與1! 程式學習筆記

 Login ▾ Recommend Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON 猴子遇到0與1! 程式學習筆記

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

文件對話框(QFileDialog)

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s
=
QFileDialog::getOpenFileName(this, tr

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress