OpenCV 的输入输出数组

Dezeming Family

2023 年 4 月 17 日

目录

一 介绍

在使用 OpenCV 的函数时，我们经常会遇到 OpenCV 的输入输出数组，比如：

```
InputArray
OutputArray
InputArrayOfArrays
OutputArrayOfArrays
InputOutputArray
InputOutputArrayOfArrays
```

那么本文就以 OpenCV3.4.2 为例，来详细解读一下这些类之间的关系。

在 opencv2/core/mat.hpp 文件中，有下面的一些定义：

```
typedef const _InputArray& InputArray;
typedef InputArray InputArrayOfArrays;
typedef const _OutputArray& OutputArray;
typedef OutputArray OutputArrayOfArrays;
typedef const _InputOutputArray& InputOutputArray;
typedef InputOutputArray InputOutputArrayOfArrays;
```

以及：

```
class CV_EXPORTS _InputArray {
    ......
}
class CV_EXPORTS _OutputArray : public _InputArray {
    ......
}
class CV_EXPORTS _InputOutputArray : public _OutputArray {
    ......
}
```

大家可能会好奇，为什么输入和输出数组都是常量引用：

```
typedef const _InputArray& InputArray;
typedef const _OutputArray& OutputArray;
typedef const _InputOutputArray& InputOutputArray;
```

按理来说，输出不应该是常量，因为需要对其进行改动。但这里是常量，是因为改动并不会改变 OutputArray 本身的值，而是改变里面存储的内容，比如：

```
void _OutputArray::clear() const {
    int k = kind();
    if( k == MAT ) {
        CV_Assert(!fixedSize());
        ((Mat*)obj)->resize(0);
        return;
    }
    release();
}
```

虽然 clear() 函数是常量函数，但是它可以清理掉 OutputArray 内部存储的内容。

我们下面开始详细分析每个类的功能和实现。

二　各个类的基本功能和结构

2 1　_InputArray

因为后面的其他类都是继承自该类，所以该类的讲解自然会比后续章节多一些内容。

_InputArray 有一堆不同参数的拷贝构造函数，使得当函数的参数是 InputArray 时，我们可以传入各种类型的数据，比如 cv::Mat、std::vector<cv::Mat>，甚至是 std::vector<std::vector<cv::Point»：

```
1 _InputArray(const Mat& m);
2 _InputArray(const std::vector<Mat>& vec);
3 template<typename _Tp> _InputArray(const std::vector<std::vector<_Tp> >& vec
    );
```

以拷贝构造函数的参数为 Mat 为例，会进行如下初始化：

```
1 inline _InputArray::_InputArray(const Mat& m) { init(MAT+ACCESS_READ, &m); }
```

当函数中需要使用 _InputArray 中的 Mat 时，则可以使用 getMat() 成员函数，比如：

```
1 Mat image = _image.getMat();
```

如果要获得 vector 就没那么简单了，它需要先用 total() 函数统计数组元素个数，然后使用 getMat() 获得元素指针。之后再根据每个元素实际的类型进行转换，比如如果 InputArray 存储的内容是：

```
1 std::vector<std::vector<cv::Point>>
```

那么就可以通过 cvMakeSeqHeaderForArray() 函数将每个 std::vector<cv::Point> 进行转换。

2 2　_OutputArray

该类继承自 _InputArray，除了本来的函数，还有 create() 函数，用于创建设定大小的数据。以及 assign() 函数，用于赋值。

2 3　_InputOutputArray

该类并没有实现什么特别的内容，

三　小结

我们有些内容并没有详细介绍，比如 cvMakeSeqHeaderForArray() 函数是如何操作的，但是由于一方面它的实现比较底层，而且涉及内容比较复杂，所以暂时先不再过多介绍，以后可能会在其他文章中描述。

参考文献

[1] xxx