

imencode与imdecode

imencode编码：

```
bool cv::imencode(  
    const String& ext,  
    InputArray img,  
    std::vector<uchar>& buf,  
    const std::vector<int>& params=std::vector<int>()  
)
```

参数说明：

ext-定义输出文件格式的扩展名

img-需要被编码的图像

buf-输出的缓存区，类型是vector

parms-被编码的格式和压缩率，类型是vector

parms为两个参数：第一个是编码的flag，一个是编码质量

编码质量如下：

JPEG，它的压缩率范围（**cv_imwrite_jpeg_quality**）从0到100（越大越好）。默认值是95。100为没有压缩。

对于**WEBP**来说，它的压缩范围（**cv_imwrite_webp_quality**）从1到100（越大越好）。默认情况下（不含任何参数）和质量在100以上，则使用无损压缩

png，可以压缩级别（**cv_imwrite_png_compression**）从0到9。更高的值意味着更小的尺寸和更长的压缩时间。默认值是3

PPM、**PGM**、或**PBM**，它可以是一个二进制格式的标志（**cv_imwrite_pxm_binary**），0或1。默认值是1。

例子：

```
std::vector<int> parm = std::vector<int>(2);  
parm[0] = IMWRITE_JPEG_OPTIMIZE;  
parm[1] = 95;  
cv::imencode(".jpg",mat, buf, parm);
```

imdecode解码：

```
Mat cv::imdecode(  
    InputArray buf,  
    int flags  
)
```

参数说明：

buf-输入解压的buf

flags-和**imread()**的**flags**是一样的

CV_LOAD_IMOSE_COLOR-如果设置，始终将图像转换为彩色图像

CV_LOAD_IMAGE_GRAYSCALE如果设置，始终将图像转换为灰度图像

dst -解码矩阵的可选输出占位符。不填则是NULL

例子:

```
std::vector<uchar> decode;  
Mat image = imdecode(decode, CV_LOAD_IMAGE_COLOR); // 图像解码
```