JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/imgcodecs/Imgcodecs.html)
* [No Frames](http://docs.google.com/Imgcodecs.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* [Constr](#2et92p0) |
* [Method](#tyjcwt)
* Detail:
* [Field](#1t3h5sf) |
* [Constr](#1opuj5n) |
* [Method](#2nusc19)

org.opencv.imgcodecs

## Class Imgcodecs

* java.lang.Object
  + org.opencv.imgcodecs.Imgcodecs
* public class Imgcodecs  
  extends java.lang.Object

### Field SummaryFields

| Modifier and Type | Field and Description |
| --- | --- |
| static int | [**CV\_CVTIMG\_FLIP**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_CVTIMG_FLIP) |
| static int | [**CV\_CVTIMG\_SWAP\_RB**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_CVTIMG_SWAP_RB) |
| static int | [**CV\_IMWRITE\_EXR\_TYPE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_EXR_TYPE) |
| static int | [**CV\_IMWRITE\_JPEG\_CHROMA\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_JPEG_CHROMA_QUALITY) |
| static int | [**CV\_IMWRITE\_JPEG\_LUMA\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_JPEG_LUMA_QUALITY) |
| static int | [**CV\_IMWRITE\_JPEG\_OPTIMIZE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_JPEG_OPTIMIZE) |
| static int | [**CV\_IMWRITE\_JPEG\_PROGRESSIVE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_JPEG_PROGRESSIVE) |
| static int | [**CV\_IMWRITE\_JPEG\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_JPEG_QUALITY) |
| static int | [**CV\_IMWRITE\_JPEG\_RST\_INTERVAL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_JPEG_RST_INTERVAL) |
| static int | [**CV\_IMWRITE\_PAM\_FORMAT\_BLACKANDWHITE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_FORMAT_BLACKANDWHITE) |
| static int | [**CV\_IMWRITE\_PAM\_FORMAT\_GRAYSCALE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_FORMAT_GRAYSCALE) |
| static int | [**CV\_IMWRITE\_PAM\_FORMAT\_GRAYSCALE\_ALPHA**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_FORMAT_GRAYSCALE_ALPHA) |
| static int | [**CV\_IMWRITE\_PAM\_FORMAT\_NULL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_FORMAT_NULL) |
| static int | [**CV\_IMWRITE\_PAM\_FORMAT\_RGB**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_FORMAT_RGB) |
| static int | [**CV\_IMWRITE\_PAM\_FORMAT\_RGB\_ALPHA**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_FORMAT_RGB_ALPHA) |
| static int | [**CV\_IMWRITE\_PAM\_TUPLETYPE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PAM_TUPLETYPE) |
| static int | [**CV\_IMWRITE\_PNG\_BILEVEL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_BILEVEL) |
| static int | [**CV\_IMWRITE\_PNG\_COMPRESSION**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_COMPRESSION) |
| static int | [**CV\_IMWRITE\_PNG\_STRATEGY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_STRATEGY) |
| static int | [**CV\_IMWRITE\_PNG\_STRATEGY\_DEFAULT**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_STRATEGY_DEFAULT) |
| static int | [**CV\_IMWRITE\_PNG\_STRATEGY\_FILTERED**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_STRATEGY_FILTERED) |
| static int | [**CV\_IMWRITE\_PNG\_STRATEGY\_FIXED**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_STRATEGY_FIXED) |
| static int | [**CV\_IMWRITE\_PNG\_STRATEGY\_HUFFMAN\_ONLY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_STRATEGY_HUFFMAN_ONLY) |
| static int | [**CV\_IMWRITE\_PNG\_STRATEGY\_RLE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PNG_STRATEGY_RLE) |
| static int | [**CV\_IMWRITE\_PXM\_BINARY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_PXM_BINARY) |
| static int | [**CV\_IMWRITE\_WEBP\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_IMWRITE_WEBP_QUALITY) |
| static int | [**CV\_LOAD\_IMAGE\_ANYCOLOR**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_LOAD_IMAGE_ANYCOLOR) |
| static int | [**CV\_LOAD\_IMAGE\_ANYDEPTH**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_LOAD_IMAGE_ANYDEPTH) |
| static int | [**CV\_LOAD\_IMAGE\_COLOR**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_LOAD_IMAGE_COLOR) |
| static int | [**CV\_LOAD\_IMAGE\_GRAYSCALE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_LOAD_IMAGE_GRAYSCALE) |
| static int | [**CV\_LOAD\_IMAGE\_IGNORE\_ORIENTATION**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_LOAD_IMAGE_IGNORE_ORIENTATION) |
| static int | [**CV\_LOAD\_IMAGE\_UNCHANGED**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#CV_LOAD_IMAGE_UNCHANGED) |
| static int | [**IMREAD\_ANYCOLOR**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_ANYCOLOR) |
| static int | [**IMREAD\_ANYDEPTH**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_ANYDEPTH) |
| static int | [**IMREAD\_COLOR**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_COLOR) |
| static int | [**IMREAD\_GRAYSCALE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_GRAYSCALE) |
| static int | [**IMREAD\_IGNORE\_ORIENTATION**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_IGNORE_ORIENTATION) |
| static int | [**IMREAD\_LOAD\_GDAL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_LOAD_GDAL) |
| static int | [**IMREAD\_REDUCED\_COLOR\_2**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_REDUCED_COLOR_2) |
| static int | [**IMREAD\_REDUCED\_COLOR\_4**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_REDUCED_COLOR_4) |
| static int | [**IMREAD\_REDUCED\_COLOR\_8**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_REDUCED_COLOR_8) |
| static int | [**IMREAD\_REDUCED\_GRAYSCALE\_2**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_REDUCED_GRAYSCALE_2) |
| static int | [**IMREAD\_REDUCED\_GRAYSCALE\_4**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_REDUCED_GRAYSCALE_4) |
| static int | [**IMREAD\_REDUCED\_GRAYSCALE\_8**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_REDUCED_GRAYSCALE_8) |
| static int | [**IMREAD\_UNCHANGED**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMREAD_UNCHANGED) |
| static int | [**IMWRITE\_EXR\_TYPE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_EXR_TYPE) |
| static int | [**IMWRITE\_EXR\_TYPE\_FLOAT**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_EXR_TYPE_FLOAT) |
| static int | [**IMWRITE\_EXR\_TYPE\_HALF**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_EXR_TYPE_HALF) |
| static int | [**IMWRITE\_JPEG\_CHROMA\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_JPEG_CHROMA_QUALITY) |
| static int | [**IMWRITE\_JPEG\_LUMA\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_JPEG_LUMA_QUALITY) |
| static int | [**IMWRITE\_JPEG\_OPTIMIZE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_JPEG_OPTIMIZE) |
| static int | [**IMWRITE\_JPEG\_PROGRESSIVE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_JPEG_PROGRESSIVE) |
| static int | [**IMWRITE\_JPEG\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_JPEG_QUALITY) |
| static int | [**IMWRITE\_JPEG\_RST\_INTERVAL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_JPEG_RST_INTERVAL) |
| static int | [**IMWRITE\_PAM\_FORMAT\_BLACKANDWHITE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_FORMAT_BLACKANDWHITE) |
| static int | [**IMWRITE\_PAM\_FORMAT\_GRAYSCALE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_FORMAT_GRAYSCALE) |
| static int | [**IMWRITE\_PAM\_FORMAT\_GRAYSCALE\_ALPHA**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_FORMAT_GRAYSCALE_ALPHA) |
| static int | [**IMWRITE\_PAM\_FORMAT\_NULL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_FORMAT_NULL) |
| static int | [**IMWRITE\_PAM\_FORMAT\_RGB**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_FORMAT_RGB) |
| static int | [**IMWRITE\_PAM\_FORMAT\_RGB\_ALPHA**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_FORMAT_RGB_ALPHA) |
| static int | [**IMWRITE\_PAM\_TUPLETYPE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PAM_TUPLETYPE) |
| static int | [**IMWRITE\_PNG\_BILEVEL**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_BILEVEL) |
| static int | [**IMWRITE\_PNG\_COMPRESSION**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_COMPRESSION) |
| static int | [**IMWRITE\_PNG\_STRATEGY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_STRATEGY) |
| static int | [**IMWRITE\_PNG\_STRATEGY\_DEFAULT**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_STRATEGY_DEFAULT) |
| static int | [**IMWRITE\_PNG\_STRATEGY\_FILTERED**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_STRATEGY_FILTERED) |
| static int | [**IMWRITE\_PNG\_STRATEGY\_FIXED**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_STRATEGY_FIXED) |
| static int | [**IMWRITE\_PNG\_STRATEGY\_HUFFMAN\_ONLY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_STRATEGY_HUFFMAN_ONLY) |
| static int | [**IMWRITE\_PNG\_STRATEGY\_RLE**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PNG_STRATEGY_RLE) |
| static int | [**IMWRITE\_PXM\_BINARY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_PXM_BINARY) |
| static int | [**IMWRITE\_TIFF\_COMPRESSION**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_TIFF_COMPRESSION) |
| static int | [**IMWRITE\_TIFF\_RESUNIT**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_TIFF_RESUNIT) |
| static int | [**IMWRITE\_TIFF\_XDPI**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_TIFF_XDPI) |
| static int | [**IMWRITE\_TIFF\_YDPI**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_TIFF_YDPI) |
| static int | [**IMWRITE\_WEBP\_QUALITY**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#IMWRITE_WEBP_QUALITY) |

### Constructor SummaryConstructors

| Constructor and Description |
| --- |
| [**Imgcodecs**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#Imgcodecs())() |

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**imdecode**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imdecode(org.opencv.core.Mat,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) buf, int flags) Reads an image from a buffer in memory. |
| static boolean | [**imencode**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imencode(java.lang.String,%20org.opencv.core.Mat,%20org.opencv.core.MatOfByte))(java.lang.String ext, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) buf) Encodes an image into a memory buffer. |
| static boolean | [**imencode**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imencode(java.lang.String,%20org.opencv.core.Mat,%20org.opencv.core.MatOfByte,%20org.opencv.core.MatOfInt))(java.lang.String ext, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) buf, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) params) Encodes an image into a memory buffer. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**imread**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imread(java.lang.String))(java.lang.String filename) Loads an image from a file. |
| static [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**imread**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imread(java.lang.String,%20int))(java.lang.String filename, int flags) Loads an image from a file. |
| static boolean | [**imreadmulti**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imreadmulti(java.lang.String,%20java.util.List))(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> mats) Loads a multi-page image from a file. |
| static boolean | [**imreadmulti**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imreadmulti(java.lang.String,%20java.util.List,%20int))(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> mats, int flags) Loads a multi-page image from a file. |
| static boolean | [**imwrite**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imwrite(java.lang.String,%20org.opencv.core.Mat))(java.lang.String filename, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img) Saves an image to a specified file. |
| static boolean | [**imwrite**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imwrite(java.lang.String,%20org.opencv.core.Mat,%20org.opencv.core.MatOfInt))(java.lang.String filename, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) params) Saves an image to a specified file. |
| static boolean | [**imwritemulti**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imwritemulti(java.lang.String,%20java.util.List))(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> img) |
| static boolean | [**imwritemulti**](http://docs.google.com/org/opencv/imgcodecs/Imgcodecs.html#imwritemulti(java.lang.String,%20java.util.List,%20org.opencv.core.MatOfInt))(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> img, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) params) |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### CV\_CVTIMG\_FLIP public static final int CV\_CVTIMG\_FLIPSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_CVTIMG_FLIP)

#### CV\_CVTIMG\_SWAP\_RB public static final int CV\_CVTIMG\_SWAP\_RBSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_CVTIMG_SWAP_RB)

#### CV\_IMWRITE\_EXR\_TYPE public static final int CV\_IMWRITE\_EXR\_TYPESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_EXR_TYPE)

#### CV\_IMWRITE\_JPEG\_CHROMA\_QUALITY public static final int CV\_IMWRITE\_JPEG\_CHROMA\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_JPEG_CHROMA_QUALITY)

#### CV\_IMWRITE\_JPEG\_LUMA\_QUALITY public static final int CV\_IMWRITE\_JPEG\_LUMA\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_JPEG_LUMA_QUALITY)

#### CV\_IMWRITE\_JPEG\_OPTIMIZE public static final int CV\_IMWRITE\_JPEG\_OPTIMIZESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_JPEG_OPTIMIZE)

#### CV\_IMWRITE\_JPEG\_PROGRESSIVE public static final int CV\_IMWRITE\_JPEG\_PROGRESSIVESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_JPEG_PROGRESSIVE)

#### CV\_IMWRITE\_JPEG\_QUALITY public static final int CV\_IMWRITE\_JPEG\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_JPEG_QUALITY)

#### CV\_IMWRITE\_JPEG\_RST\_INTERVAL public static final int CV\_IMWRITE\_JPEG\_RST\_INTERVALSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_JPEG_RST_INTERVAL)

#### CV\_IMWRITE\_PAM\_FORMAT\_BLACKANDWHITE public static final int CV\_IMWRITE\_PAM\_FORMAT\_BLACKANDWHITESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_FORMAT_BLACKANDWHITE)

#### CV\_IMWRITE\_PAM\_FORMAT\_GRAYSCALE public static final int CV\_IMWRITE\_PAM\_FORMAT\_GRAYSCALESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_FORMAT_GRAYSCALE)

#### CV\_IMWRITE\_PAM\_FORMAT\_GRAYSCALE\_ALPHA public static final int CV\_IMWRITE\_PAM\_FORMAT\_GRAYSCALE\_ALPHASee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_FORMAT_GRAYSCALE_ALPHA)

#### CV\_IMWRITE\_PAM\_FORMAT\_NULL public static final int CV\_IMWRITE\_PAM\_FORMAT\_NULLSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_FORMAT_NULL)

#### CV\_IMWRITE\_PAM\_FORMAT\_RGB public static final int CV\_IMWRITE\_PAM\_FORMAT\_RGBSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_FORMAT_RGB)

#### CV\_IMWRITE\_PAM\_FORMAT\_RGB\_ALPHA public static final int CV\_IMWRITE\_PAM\_FORMAT\_RGB\_ALPHASee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_FORMAT_RGB_ALPHA)

#### CV\_IMWRITE\_PAM\_TUPLETYPE public static final int CV\_IMWRITE\_PAM\_TUPLETYPESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PAM_TUPLETYPE)

#### CV\_IMWRITE\_PNG\_BILEVEL public static final int CV\_IMWRITE\_PNG\_BILEVELSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_BILEVEL)

#### CV\_IMWRITE\_PNG\_COMPRESSION public static final int CV\_IMWRITE\_PNG\_COMPRESSIONSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_COMPRESSION)

#### CV\_IMWRITE\_PNG\_STRATEGY public static final int CV\_IMWRITE\_PNG\_STRATEGYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_STRATEGY)

#### CV\_IMWRITE\_PNG\_STRATEGY\_DEFAULT public static final int CV\_IMWRITE\_PNG\_STRATEGY\_DEFAULTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_STRATEGY_DEFAULT)

#### CV\_IMWRITE\_PNG\_STRATEGY\_FILTERED public static final int CV\_IMWRITE\_PNG\_STRATEGY\_FILTEREDSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_STRATEGY_FILTERED)

#### CV\_IMWRITE\_PNG\_STRATEGY\_FIXED public static final int CV\_IMWRITE\_PNG\_STRATEGY\_FIXEDSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_STRATEGY_FIXED)

#### CV\_IMWRITE\_PNG\_STRATEGY\_HUFFMAN\_ONLY public static final int CV\_IMWRITE\_PNG\_STRATEGY\_HUFFMAN\_ONLYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_STRATEGY_HUFFMAN_ONLY)

#### CV\_IMWRITE\_PNG\_STRATEGY\_RLE public static final int CV\_IMWRITE\_PNG\_STRATEGY\_RLESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PNG_STRATEGY_RLE)

#### CV\_IMWRITE\_PXM\_BINARY public static final int CV\_IMWRITE\_PXM\_BINARYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_PXM_BINARY)

#### CV\_IMWRITE\_WEBP\_QUALITY public static final int CV\_IMWRITE\_WEBP\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_IMWRITE_WEBP_QUALITY)

#### CV\_LOAD\_IMAGE\_ANYCOLOR public static final int CV\_LOAD\_IMAGE\_ANYCOLORSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_LOAD_IMAGE_ANYCOLOR)

#### CV\_LOAD\_IMAGE\_ANYDEPTH public static final int CV\_LOAD\_IMAGE\_ANYDEPTHSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_LOAD_IMAGE_ANYDEPTH)

#### CV\_LOAD\_IMAGE\_COLOR public static final int CV\_LOAD\_IMAGE\_COLORSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_LOAD_IMAGE_COLOR)

#### CV\_LOAD\_IMAGE\_GRAYSCALE public static final int CV\_LOAD\_IMAGE\_GRAYSCALESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_LOAD_IMAGE_GRAYSCALE)

#### CV\_LOAD\_IMAGE\_IGNORE\_ORIENTATION public static final int CV\_LOAD\_IMAGE\_IGNORE\_ORIENTATIONSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_LOAD_IMAGE_IGNORE_ORIENTATION)

#### CV\_LOAD\_IMAGE\_UNCHANGED public static final int CV\_LOAD\_IMAGE\_UNCHANGEDSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.CV_LOAD_IMAGE_UNCHANGED)

#### IMREAD\_ANYCOLOR public static final int IMREAD\_ANYCOLORSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_ANYCOLOR)

#### IMREAD\_ANYDEPTH public static final int IMREAD\_ANYDEPTHSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_ANYDEPTH)

#### IMREAD\_COLOR public static final int IMREAD\_COLORSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_COLOR)

#### IMREAD\_GRAYSCALE public static final int IMREAD\_GRAYSCALESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_GRAYSCALE)

#### IMREAD\_IGNORE\_ORIENTATION public static final int IMREAD\_IGNORE\_ORIENTATIONSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_IGNORE_ORIENTATION)

#### IMREAD\_LOAD\_GDAL public static final int IMREAD\_LOAD\_GDALSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_LOAD_GDAL)

#### IMREAD\_REDUCED\_COLOR\_2 public static final int IMREAD\_REDUCED\_COLOR\_2See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_REDUCED_COLOR_2)

#### IMREAD\_REDUCED\_COLOR\_4 public static final int IMREAD\_REDUCED\_COLOR\_4See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_REDUCED_COLOR_4)

#### IMREAD\_REDUCED\_COLOR\_8 public static final int IMREAD\_REDUCED\_COLOR\_8See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_REDUCED_COLOR_8)

#### IMREAD\_REDUCED\_GRAYSCALE\_2 public static final int IMREAD\_REDUCED\_GRAYSCALE\_2See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_REDUCED_GRAYSCALE_2)

#### IMREAD\_REDUCED\_GRAYSCALE\_4 public static final int IMREAD\_REDUCED\_GRAYSCALE\_4See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_REDUCED_GRAYSCALE_4)

#### IMREAD\_REDUCED\_GRAYSCALE\_8 public static final int IMREAD\_REDUCED\_GRAYSCALE\_8See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_REDUCED_GRAYSCALE_8)

#### IMREAD\_UNCHANGED public static final int IMREAD\_UNCHANGEDSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMREAD_UNCHANGED)

#### IMWRITE\_EXR\_TYPE public static final int IMWRITE\_EXR\_TYPESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_EXR_TYPE)

#### IMWRITE\_EXR\_TYPE\_FLOAT public static final int IMWRITE\_EXR\_TYPE\_FLOATSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_EXR_TYPE_FLOAT)

#### IMWRITE\_EXR\_TYPE\_HALF public static final int IMWRITE\_EXR\_TYPE\_HALFSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_EXR_TYPE_HALF)

#### IMWRITE\_JPEG\_CHROMA\_QUALITY public static final int IMWRITE\_JPEG\_CHROMA\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_JPEG_CHROMA_QUALITY)

#### IMWRITE\_JPEG\_LUMA\_QUALITY public static final int IMWRITE\_JPEG\_LUMA\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_JPEG_LUMA_QUALITY)

#### IMWRITE\_JPEG\_OPTIMIZE public static final int IMWRITE\_JPEG\_OPTIMIZESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_JPEG_OPTIMIZE)

#### IMWRITE\_JPEG\_PROGRESSIVE public static final int IMWRITE\_JPEG\_PROGRESSIVESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_JPEG_PROGRESSIVE)

#### IMWRITE\_JPEG\_QUALITY public static final int IMWRITE\_JPEG\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_JPEG_QUALITY)

#### IMWRITE\_JPEG\_RST\_INTERVAL public static final int IMWRITE\_JPEG\_RST\_INTERVALSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_JPEG_RST_INTERVAL)

#### IMWRITE\_PAM\_FORMAT\_BLACKANDWHITE public static final int IMWRITE\_PAM\_FORMAT\_BLACKANDWHITESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_FORMAT_BLACKANDWHITE)

#### IMWRITE\_PAM\_FORMAT\_GRAYSCALE public static final int IMWRITE\_PAM\_FORMAT\_GRAYSCALESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_FORMAT_GRAYSCALE)

#### IMWRITE\_PAM\_FORMAT\_GRAYSCALE\_ALPHA public static final int IMWRITE\_PAM\_FORMAT\_GRAYSCALE\_ALPHASee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_FORMAT_GRAYSCALE_ALPHA)

#### IMWRITE\_PAM\_FORMAT\_NULL public static final int IMWRITE\_PAM\_FORMAT\_NULLSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_FORMAT_NULL)

#### IMWRITE\_PAM\_FORMAT\_RGB public static final int IMWRITE\_PAM\_FORMAT\_RGBSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_FORMAT_RGB)

#### IMWRITE\_PAM\_FORMAT\_RGB\_ALPHA public static final int IMWRITE\_PAM\_FORMAT\_RGB\_ALPHASee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_FORMAT_RGB_ALPHA)

#### IMWRITE\_PAM\_TUPLETYPE public static final int IMWRITE\_PAM\_TUPLETYPESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PAM_TUPLETYPE)

#### IMWRITE\_PNG\_BILEVEL public static final int IMWRITE\_PNG\_BILEVELSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_BILEVEL)

#### IMWRITE\_PNG\_COMPRESSION public static final int IMWRITE\_PNG\_COMPRESSIONSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_COMPRESSION)

#### IMWRITE\_PNG\_STRATEGY public static final int IMWRITE\_PNG\_STRATEGYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_STRATEGY)

#### IMWRITE\_PNG\_STRATEGY\_DEFAULT public static final int IMWRITE\_PNG\_STRATEGY\_DEFAULTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_STRATEGY_DEFAULT)

#### IMWRITE\_PNG\_STRATEGY\_FILTERED public static final int IMWRITE\_PNG\_STRATEGY\_FILTEREDSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_STRATEGY_FILTERED)

#### IMWRITE\_PNG\_STRATEGY\_FIXED public static final int IMWRITE\_PNG\_STRATEGY\_FIXEDSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_STRATEGY_FIXED)

#### IMWRITE\_PNG\_STRATEGY\_HUFFMAN\_ONLY public static final int IMWRITE\_PNG\_STRATEGY\_HUFFMAN\_ONLYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_STRATEGY_HUFFMAN_ONLY)

#### IMWRITE\_PNG\_STRATEGY\_RLE public static final int IMWRITE\_PNG\_STRATEGY\_RLESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PNG_STRATEGY_RLE)

#### IMWRITE\_PXM\_BINARY public static final int IMWRITE\_PXM\_BINARYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_PXM_BINARY)

#### IMWRITE\_TIFF\_COMPRESSION public static final int IMWRITE\_TIFF\_COMPRESSIONSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_TIFF_COMPRESSION)

#### IMWRITE\_TIFF\_RESUNIT public static final int IMWRITE\_TIFF\_RESUNITSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_TIFF_RESUNIT)

#### IMWRITE\_TIFF\_XDPI public static final int IMWRITE\_TIFF\_XDPISee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_TIFF_XDPI)

#### IMWRITE\_TIFF\_YDPI public static final int IMWRITE\_TIFF\_YDPISee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_TIFF_YDPI)

#### IMWRITE\_WEBP\_QUALITY public static final int IMWRITE\_WEBP\_QUALITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgcodecs.Imgcodecs.IMWRITE_WEBP_QUALITY)

### Constructor Detail

#### Imgcodecs public Imgcodecs()

### Method Detail

#### imdecode public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) imdecode([Mat](http://docs.google.com/org/opencv/core/Mat.html) buf, int flags) Reads an image from a buffer in memory. The function imdecode reads an image from the specified buffer in the memory. If the buffer is too short or contains invalid data, the function returns an empty matrix ( Mat::data==NULL ). See cv::imread for the list of supported formats and flags description. **Note:** In the case of color images, the decoded images will have the channels stored in **B G R** order.Parameters:buf - Input array or vector of bytes.flags - The same flags as in cv::imread, see cv::ImreadModes. Returns:automatically generated

#### imencode public static boolean imencode(java.lang.String ext, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) buf) Encodes an image into a memory buffer. The function imencode compresses the image and stores it in the memory buffer that is resized to fit the result. See cv::imwrite for the list of supported formats and flags description.Parameters:ext - File extension that defines the output format.img - Image to be written.buf - Output buffer resized to fit the compressed image. Returns:automatically generated

#### imencode public static boolean imencode(java.lang.String ext, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [MatOfByte](http://docs.google.com/org/opencv/core/MatOfByte.html) buf, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) params) Encodes an image into a memory buffer. The function imencode compresses the image and stores it in the memory buffer that is resized to fit the result. See cv::imwrite for the list of supported formats and flags description.Parameters:ext - File extension that defines the output format.img - Image to be written.buf - Output buffer resized to fit the compressed image.params - Format-specific parameters. See cv::imwrite and cv::ImwriteFlags. Returns:automatically generated

#### imread public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) imread(java.lang.String filename)

Loads an image from a file. imread The function imread loads an image from the specified file and returns it. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format), the function returns an empty matrix ( Mat::data==NULL ). Currently, the following file formats are supported:

* + - * Windows bitmaps - \\*.bmp, \\*.dib (always supported)
      * JPEG files - \\*.jpeg, \\*.jpg, \\*.jpe (see the \*Note\* section)
      * JPEG 2000 files - \\*.jp2 (see the \*Note\* section)
      * Portable Network Graphics - \\*.png (see the \*Note\* section)
      * WebP - \\*.webp (see the \*Note\* section)
      * Portable image format - \\*.pbm, \\*.pgm, \\*.ppm \\*.pxm, \\*.pnm (always supported)
      * Sun rasters - \\*.sr, \\*.ras (always supported)
      * TIFF files - \\*.tiff, \\*.tif (see the \*Note\* section)
      * OpenEXR Image files - \\*.exr (see the \*Note\* section)
      * Radiance HDR - \\*.hdr, \\*.pic (always supported)
      * Raster and Vector geospatial data supported by GDAL (see the \*Note\* section)

**Note:**

* + - * The function determines the type of an image by the content, not by the file extension.
      * In the case of color images, the decoded images will have the channels stored in **B G R** order.
      * When using IMREAD\_GRAYSCALE, the codec's internal grayscale conversion will be used, if available. Results may differ to the output of cvtColor()
      * On Microsoft Windows\\* OS and MacOSX\\*, the codecs shipped with an OpenCV image (libjpeg, libpng, libtiff, and libjasper) are used by default. So, OpenCV can always read JPEGs, PNGs, and TIFFs. On MacOSX, there is also an option to use native MacOSX image readers. But beware that currently these native image loaders give images with different pixel values because of the color management embedded into MacOSX.
      * On Linux\\*, BSD flavors and other Unix-like open-source operating systems, OpenCV looks for codecs supplied with an OS image. Install the relevant packages (do not forget the development files, for example, "libjpeg-dev", in Debian\\* and Ubuntu\\*) to get the codec support or turn on the OPENCV\_BUILD\_3RDPARTY\_LIBS flag in CMake.
      * In the case you set \*WITH\_GDAL\* flag to true in CMake and REF: IMREAD\_LOAD\_GDAL to load the image, then the [GDAL](http://www.gdal.org) driver will be used in order to decode the image, supporting the following formats: [Raster](http://www.gdal.org/formats\_list.html), [Vector](http://www.gdal.org/ogr\_formats.html).
      * If EXIF information is embedded in the image file, the EXIF orientation will be taken into account and thus the image will be rotated accordingly except if the flags REF: IMREAD\_IGNORE\_ORIENTATION or REF: IMREAD\_UNCHANGED are passed.
      * By default number of pixels must be less than 2^30. Limit can be set using system variable OPENCV\_IO\_MAX\_IMAGE\_PIXELS

Parameters:filename - Name of file to be loaded. Returns:automatically generated

#### imread public static [Mat](http://docs.google.com/org/opencv/core/Mat.html) imread(java.lang.String filename, int flags)

Loads an image from a file. imread The function imread loads an image from the specified file and returns it. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format), the function returns an empty matrix ( Mat::data==NULL ). Currently, the following file formats are supported:

* + - * Windows bitmaps - \\*.bmp, \\*.dib (always supported)
      * JPEG files - \\*.jpeg, \\*.jpg, \\*.jpe (see the \*Note\* section)
      * JPEG 2000 files - \\*.jp2 (see the \*Note\* section)
      * Portable Network Graphics - \\*.png (see the \*Note\* section)
      * WebP - \\*.webp (see the \*Note\* section)
      * Portable image format - \\*.pbm, \\*.pgm, \\*.ppm \\*.pxm, \\*.pnm (always supported)
      * Sun rasters - \\*.sr, \\*.ras (always supported)
      * TIFF files - \\*.tiff, \\*.tif (see the \*Note\* section)
      * OpenEXR Image files - \\*.exr (see the \*Note\* section)
      * Radiance HDR - \\*.hdr, \\*.pic (always supported)
      * Raster and Vector geospatial data supported by GDAL (see the \*Note\* section)

**Note:**

* + - * The function determines the type of an image by the content, not by the file extension.
      * In the case of color images, the decoded images will have the channels stored in **B G R** order.
      * When using IMREAD\_GRAYSCALE, the codec's internal grayscale conversion will be used, if available. Results may differ to the output of cvtColor()
      * On Microsoft Windows\\* OS and MacOSX\\*, the codecs shipped with an OpenCV image (libjpeg, libpng, libtiff, and libjasper) are used by default. So, OpenCV can always read JPEGs, PNGs, and TIFFs. On MacOSX, there is also an option to use native MacOSX image readers. But beware that currently these native image loaders give images with different pixel values because of the color management embedded into MacOSX.
      * On Linux\\*, BSD flavors and other Unix-like open-source operating systems, OpenCV looks for codecs supplied with an OS image. Install the relevant packages (do not forget the development files, for example, "libjpeg-dev", in Debian\\* and Ubuntu\\*) to get the codec support or turn on the OPENCV\_BUILD\_3RDPARTY\_LIBS flag in CMake.
      * In the case you set \*WITH\_GDAL\* flag to true in CMake and REF: IMREAD\_LOAD\_GDAL to load the image, then the [GDAL](http://www.gdal.org) driver will be used in order to decode the image, supporting the following formats: [Raster](http://www.gdal.org/formats\_list.html), [Vector](http://www.gdal.org/ogr\_formats.html).
      * If EXIF information is embedded in the image file, the EXIF orientation will be taken into account and thus the image will be rotated accordingly except if the flags REF: IMREAD\_IGNORE\_ORIENTATION or REF: IMREAD\_UNCHANGED are passed.
      * By default number of pixels must be less than 2^30. Limit can be set using system variable OPENCV\_IO\_MAX\_IMAGE\_PIXELS

Parameters:filename - Name of file to be loaded.flags - Flag that can take values of cv::ImreadModes Returns:automatically generated

#### imreadmulti public static boolean imreadmulti(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> mats) Loads a multi-page image from a file. The function imreadmulti loads a multi-page image from the specified file into a vector of Mat objects.Parameters:filename - Name of file to be loaded.mats - A vector of Mat objects holding each page, if more than one. SEE: cv::imread Returns:automatically generated

#### imreadmulti public static boolean imreadmulti(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> mats, int flags) Loads a multi-page image from a file. The function imreadmulti loads a multi-page image from the specified file into a vector of Mat objects.Parameters:filename - Name of file to be loaded.flags - Flag that can take values of cv::ImreadModes, default with cv::IMREAD\_ANYCOLOR.mats - A vector of Mat objects holding each page, if more than one. SEE: cv::imread Returns:automatically generated

#### imwrite public static boolean imwrite(java.lang.String filename, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img)

Saves an image to a specified file. The function imwrite saves the image to the specified file. The image format is chosen based on the filename extension (see cv::imread for the list of extensions). In general, only 8-bit single-channel or 3-channel (with 'BGR' channel order) images can be saved using this function, with these exceptions:

* + - * 16-bit unsigned (CV\_16U) images can be saved in the case of PNG, JPEG 2000, and TIFF formats
      * 32-bit float (CV\_32F) images can be saved in TIFF, OpenEXR, and Radiance HDR formats; 3-channel (CV\_32FC3) TIFF images will be saved using the LogLuv high dynamic range encoding (4 bytes per pixel)
      * PNG images with an alpha channel can be saved using this function. To do this, create 8-bit (or 16-bit) 4-channel image BGRA, where the alpha channel goes last. Fully transparent pixels should have alpha set to 0, fully opaque pixels should have alpha set to 255/65535 (see the code sample below).
      * Multiple images (vector of Mat) can be saved in TIFF format (see the code sample below).

If the format, depth or channel order is different, use Mat::convertTo and cv::cvtColor to convert it before saving. Or, use the universal FileStorage I/O functions to save the image to XML or YAML format. The sample below shows how to create a BGRA image, how to set custom compression parameters and save it to a PNG file. It also demonstrates how to save multiple images in a TIFF file: INCLUDE: snippets/imgcodecs\_imwrite.cppParameters:filename - Name of the file.img - (Mat or vector of Mat) Image or Images to be saved. Returns:automatically generated

#### imwrite public static boolean imwrite(java.lang.String filename, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) params)

Saves an image to a specified file. The function imwrite saves the image to the specified file. The image format is chosen based on the filename extension (see cv::imread for the list of extensions). In general, only 8-bit single-channel or 3-channel (with 'BGR' channel order) images can be saved using this function, with these exceptions:

* + - * 16-bit unsigned (CV\_16U) images can be saved in the case of PNG, JPEG 2000, and TIFF formats
      * 32-bit float (CV\_32F) images can be saved in TIFF, OpenEXR, and Radiance HDR formats; 3-channel (CV\_32FC3) TIFF images will be saved using the LogLuv high dynamic range encoding (4 bytes per pixel)
      * PNG images with an alpha channel can be saved using this function. To do this, create 8-bit (or 16-bit) 4-channel image BGRA, where the alpha channel goes last. Fully transparent pixels should have alpha set to 0, fully opaque pixels should have alpha set to 255/65535 (see the code sample below).
      * Multiple images (vector of Mat) can be saved in TIFF format (see the code sample below).

If the format, depth or channel order is different, use Mat::convertTo and cv::cvtColor to convert it before saving. Or, use the universal FileStorage I/O functions to save the image to XML or YAML format. The sample below shows how to create a BGRA image, how to set custom compression parameters and save it to a PNG file. It also demonstrates how to save multiple images in a TIFF file: INCLUDE: snippets/imgcodecs\_imwrite.cppParameters:filename - Name of the file.img - (Mat or vector of Mat) Image or Images to be saved.params - Format-specific parameters encoded as pairs (paramId\_1, paramValue\_1, paramId\_2, paramValue\_2, ... .) see cv::ImwriteFlags Returns:automatically generated

#### imwritemulti public static boolean imwritemulti(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> img)

#### imwritemulti public static boolean imwritemulti(java.lang.String filename, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> img, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) params)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/imgcodecs/Imgcodecs.html)
* [No Frames](http://docs.google.com/Imgcodecs.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* [Constr](#2et92p0) |
* [Method](#tyjcwt)
* Detail:
* [Field](#1t3h5sf) |
* [Constr](#1opuj5n) |
* [Method](#2nusc19)

Generated on 2021-04-02 03:15:03 / OpenCV 3.4.14