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](http://docs.google.com/org/opencv/objdetect/BaseCascadeClassifier.html)
* [Next Class](http://docs.google.com/org/opencv/objdetect/HOGDescriptor.html)
* [Frames](http://docs.google.com/index.html?org/opencv/objdetect/CascadeClassifier.html)
* [No Frames](http://docs.google.com/CascadeClassifier.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#2s8eyo1)

org.opencv.objdetect

## Class CascadeClassifier

* java.lang.Object
  + org.opencv.objdetect.CascadeClassifier
* public class CascadeClassifier  
  extends java.lang.Object  
  Cascade classifier class for object detection.

### Constructor SummaryConstructors

| Constructor and Description |
| --- |
| [**CascadeClassifier**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#CascadeClassifier())() |
| [**CascadeClassifier**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#CascadeClassifier(java.lang.String))(java.lang.String filename) Loads a classifier from a file. |

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [CascadeClassifier](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#__fromPtr__(long))(long addr) |
| static boolean | [**convert**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#convert(java.lang.String,%20java.lang.String))(java.lang.String oldcascade, java.lang.String newcascade) |
| void | [**detectMultiScale**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale(org.opencv.core.Mat,%20org.opencv.core.MatOfRect))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects) Detects objects of different sizes in the input image. |
| void | [**detectMultiScale**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20double))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor) Detects objects of different sizes in the input image. |
| void | [**detectMultiScale**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20double,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors) Detects objects of different sizes in the input image. |
| void | [**detectMultiScale**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20double,%20int,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors, int flags) Detects objects of different sizes in the input image. |
| void | [**detectMultiScale**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20double,%20int,%20int,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize) Detects objects of different sizes in the input image. |
| void | [**detectMultiScale**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20double,%20int,%20int,%20org.opencv.core.Size,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize) Detects objects of different sizes in the input image. |
| void | [**detectMultiScale2**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale2(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections) |
| void | [**detectMultiScale2**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale2(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20double))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor) |
| void | [**detectMultiScale2**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale2(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20double,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors) |
| void | [**detectMultiScale2**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale2(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20double,%20int,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors, int flags) |
| void | [**detectMultiScale2**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale2(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20double,%20int,%20int,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize) |
| void | [**detectMultiScale2**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale2(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20double,%20int,%20int,%20org.opencv.core.Size,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize) |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights) This function allows you to retrieve the final stage decision certainty of classification. |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble,%20double))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor) This function allows you to retrieve the final stage decision certainty of classification. |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble,%20double,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors) This function allows you to retrieve the final stage decision certainty of classification. |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble,%20double,%20int,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags) This function allows you to retrieve the final stage decision certainty of classification. |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble,%20double,%20int,%20int,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize) This function allows you to retrieve the final stage decision certainty of classification. |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble,%20double,%20int,%20int,%20org.opencv.core.Size,%20org.opencv.core.Size))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize) This function allows you to retrieve the final stage decision certainty of classification. |
| void | [**detectMultiScale3**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#detectMultiScale3(org.opencv.core.Mat,%20org.opencv.core.MatOfRect,%20org.opencv.core.MatOfInt,%20org.opencv.core.MatOfDouble,%20double,%20int,%20int,%20org.opencv.core.Size,%20org.opencv.core.Size,%20boolean))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize, boolean outputRejectLevels) This function allows you to retrieve the final stage decision certainty of classification. |
| boolean | [**empty**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#empty())() Checks whether the classifier has been loaded. |
| int | [**getFeatureType**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#getFeatureType())() |
| long | [**getNativeObjAddr**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#getNativeObjAddr())() |
| [Size](http://docs.google.com/org/opencv/core/Size.html) | [**getOriginalWindowSize**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#getOriginalWindowSize())() |
| boolean | [**isOldFormatCascade**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#isOldFormatCascade())() |
| boolean | [**load**](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html#load(java.lang.String))(java.lang.String filename) Loads a classifier from a file. |

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

### Constructor Detail

#### CascadeClassifier public CascadeClassifier()

#### CascadeClassifier public CascadeClassifier(java.lang.String filename) Loads a classifier from a file.Parameters:filename - Name of the file from which the classifier is loaded.

### Method Detail

#### \_\_fromPtr\_\_ public static [CascadeClassifier](http://docs.google.com/org/opencv/objdetect/CascadeClassifier.html) \_\_fromPtr\_\_(long addr)

#### convert public static boolean convert(java.lang.String oldcascade, java.lang.String newcascade)

#### detectMultiScale public void detectMultiScale([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects) Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image. to retain it. cvHaarDetectObjects. It is not used for a new cascade. The function is parallelized with the TBB library. **Note:**

* + - * (Python) A face detection example using cascade classifiers can be found at opencv\_source\_code/samples/python/facedetect.py

#### detectMultiScale public void detectMultiScale([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor) Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.scaleFactor - Parameter specifying how much the image size is reduced at each image scale. to retain it. cvHaarDetectObjects. It is not used for a new cascade. The function is parallelized with the TBB library. **Note:**

* + - * (Python) A face detection example using cascade classifiers can be found at opencv\_source\_code/samples/python/facedetect.py

#### detectMultiScale public void detectMultiScale([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors) Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it. cvHaarDetectObjects. It is not used for a new cascade. The function is parallelized with the TBB library. **Note:**

* + - * (Python) A face detection example using cascade classifiers can be found at opencv\_source\_code/samples/python/facedetect.py

#### detectMultiScale public void detectMultiScale([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors, int flags) Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it.flags - Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade. The function is parallelized with the TBB library. **Note:**

* + - * (Python) A face detection example using cascade classifiers can be found at opencv\_source\_code/samples/python/facedetect.py

#### detectMultiScale public void detectMultiScale([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize) Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it.flags - Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.minSize - Minimum possible object size. Objects smaller than that are ignored. The function is parallelized with the TBB library. **Note:**

* + - * (Python) A face detection example using cascade classifiers can be found at opencv\_source\_code/samples/python/facedetect.py

#### detectMultiScale public void detectMultiScale([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize) Detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles.Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it.flags - Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.minSize - Minimum possible object size. Objects smaller than that are ignored.maxSize - Maximum possible object size. Objects larger than that are ignored. If maxSize == minSize model is evaluated on single scale. The function is parallelized with the TBB library. **Note:**

* + - * (Python) A face detection example using cascade classifiers can be found at opencv\_source\_code/samples/python/facedetect.py

#### detectMultiScale2 public void detectMultiScale2([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections)Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.numDetections - Vector of detection numbers for the corresponding objects. An object's number of detections is the number of neighboring positively classified rectangles that were joined together to form the object. to retain it. cvHaarDetectObjects. It is not used for a new cascade.

#### detectMultiScale2 public void detectMultiScale2([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor)Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.numDetections - Vector of detection numbers for the corresponding objects. An object's number of detections is the number of neighboring positively classified rectangles that were joined together to form the object.scaleFactor - Parameter specifying how much the image size is reduced at each image scale. to retain it. cvHaarDetectObjects. It is not used for a new cascade.

#### detectMultiScale2 public void detectMultiScale2([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors)Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.numDetections - Vector of detection numbers for the corresponding objects. An object's number of detections is the number of neighboring positively classified rectangles that were joined together to form the object.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it. cvHaarDetectObjects. It is not used for a new cascade.

#### detectMultiScale2 public void detectMultiScale2([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors, int flags)Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.numDetections - Vector of detection numbers for the corresponding objects. An object's number of detections is the number of neighboring positively classified rectangles that were joined together to form the object.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it.flags - Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.

#### detectMultiScale2 public void detectMultiScale2([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize)Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.numDetections - Vector of detection numbers for the corresponding objects. An object's number of detections is the number of neighboring positively classified rectangles that were joined together to form the object.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it.flags - Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.minSize - Minimum possible object size. Objects smaller than that are ignored.

#### detectMultiScale2 public void detectMultiScale2([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) numDetections, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize)Parameters:image - Matrix of the type CV\_8U containing an image where objects are detected.objects - Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.numDetections - Vector of detection numbers for the corresponding objects. An object's number of detections is the number of neighboring positively classified rectangles that were joined together to form the object.scaleFactor - Parameter specifying how much the image size is reduced at each image scale.minNeighbors - Parameter specifying how many neighbors each candidate rectangle should have to retain it.flags - Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.minSize - Minimum possible object size. Objects smaller than that are ignored.maxSize - Maximum possible object size. Objects larger than that are ignored. If maxSize == minSize model is evaluated on single scale.

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generated

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generatedscaleFactor - automatically generated

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generatedscaleFactor - automatically generatedminNeighbors - automatically generated

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generatedscaleFactor - automatically generatedminNeighbors - automatically generatedflags - automatically generated

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generatedscaleFactor - automatically generatedminNeighbors - automatically generatedflags - automatically generatedminSize - automatically generated

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generatedscaleFactor - automatically generatedminNeighbors - automatically generatedflags - automatically generatedminSize - automatically generatedmaxSize - automatically generated

#### detectMultiScale3 public void detectMultiScale3([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [MatOfRect](http://docs.google.com/org/opencv/core/MatOfRect.html) objects, [MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) rejectLevels, [MatOfDouble](http://docs.google.com/org/opencv/core/MatOfDouble.html) levelWeights, double scaleFactor, int minNeighbors, int flags, [Size](http://docs.google.com/org/opencv/core/Size.html) minSize, [Size](http://docs.google.com/org/opencv/core/Size.html) maxSize, boolean outputRejectLevels) This function allows you to retrieve the final stage decision certainty of classification. For this, one needs to set outputRejectLevels on true and provide the rejectLevels and levelWeights parameter. For each resulting detection, levelWeights will then contain the certainty of classification at the final stage. This value can then be used to separate strong from weaker classifications. A code sample on how to use it efficiently can be found below: Mat img; vector<double> weights; vector<int> levels; vector<Rect> detections; CascadeClassifier model("/path/to/your/model.xml"); model.detectMultiScale(img, detections, levels, weights, 1.1, 3, 0, Size(), Size(), true); cerr << "Detection " << detections[0] << " with weight " << weights[0] << endl;Parameters:image - automatically generatedobjects - automatically generatedrejectLevels - automatically generatedlevelWeights - automatically generatedscaleFactor - automatically generatedminNeighbors - automatically generatedflags - automatically generatedminSize - automatically generatedmaxSize - automatically generatedoutputRejectLevels - automatically generated

#### empty public boolean empty() Checks whether the classifier has been loaded.Returns:automatically generated

#### getFeatureType public int getFeatureType()

#### getNativeObjAddr public long getNativeObjAddr()

#### getOriginalWindowSize public [Size](http://docs.google.com/org/opencv/core/Size.html) getOriginalWindowSize()

#### isOldFormatCascade public boolean isOldFormatCascade()

#### load public boolean load(java.lang.String filename) Loads a classifier from a file.Parameters:filename - Name of the file from which the classifier is loaded. The file may contain an old HAAR classifier trained by the haartraining application or a new cascade classifier trained by the traincascade application. Returns:automatically generated

* [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](http://docs.google.com/org/opencv/objdetect/BaseCascadeClassifier.html)
* [Next Class](http://docs.google.com/org/opencv/objdetect/HOGDescriptor.html)
* [Frames](http://docs.google.com/index.html?org/opencv/objdetect/CascadeClassifier.html)
* [No Frames](http://docs.google.com/CascadeClassifier.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#2s8eyo1)

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