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/ml/ParamGrid.html)
* [Next Class](http://docs.google.com/org/opencv/ml/StatModel.html)
* [Frames](http://docs.google.com/index.html?org/opencv/ml/RTrees.html)
* [No Frames](http://docs.google.com/RTrees.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#2et92p0) |
* Constr |
* [Method](#3dy6vkm)
* Detail:
* Field |
* Constr |
* [Method](#3rdcrjn)

org.opencv.ml

## Class RTrees

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.ml.StatModel](http://docs.google.com/org/opencv/ml/StatModel.html)
      * [org.opencv.ml.DTrees](http://docs.google.com/org/opencv/ml/DTrees.html)
        + org.opencv.ml.RTrees
* public class RTrees  
  extends [DTrees](http://docs.google.com/org/opencv/ml/DTrees.html)  
  The class implements the random forest predictor. SEE: REF: ml\_intro\_rtrees

### Field Summary

### Fields inherited from class org.opencv.ml.[**DTrees**](http://docs.google.com/org/opencv/ml/DTrees.html)[PREDICT\_AUTO](http://docs.google.com/org/opencv/ml/DTrees.html#PREDICT_AUTO), [PREDICT\_MASK](http://docs.google.com/org/opencv/ml/DTrees.html#PREDICT_MASK), [PREDICT\_MAX\_VOTE](http://docs.google.com/org/opencv/ml/DTrees.html#PREDICT_MAX_VOTE), [PREDICT\_SUM](http://docs.google.com/org/opencv/ml/DTrees.html#PREDICT_SUM)

### Fields inherited from class org.opencv.ml.[**StatModel**](http://docs.google.com/org/opencv/ml/StatModel.html)[COMPRESSED\_INPUT](http://docs.google.com/org/opencv/ml/StatModel.html#COMPRESSED_INPUT), [PREPROCESSED\_INPUT](http://docs.google.com/org/opencv/ml/StatModel.html#PREPROCESSED_INPUT), [RAW\_OUTPUT](http://docs.google.com/org/opencv/ml/StatModel.html#RAW_OUTPUT), [UPDATE\_MODEL](http://docs.google.com/org/opencv/ml/StatModel.html#UPDATE_MODEL)

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/ml/RTrees.html#__fromPtr__(long))(long addr) |
| static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) | [**create**](http://docs.google.com/org/opencv/ml/RTrees.html#create())() Creates the empty model. |
| int | [**getActiveVarCount**](http://docs.google.com/org/opencv/ml/RTrees.html#getActiveVarCount())() SEE: setActiveVarCount |
| boolean | [**getCalculateVarImportance**](http://docs.google.com/org/opencv/ml/RTrees.html#getCalculateVarImportance())() SEE: setCalculateVarImportance |
| double | [**getOOBError**](http://docs.google.com/org/opencv/ml/RTrees.html#getOOBError())() Returns the OOB error value, computed at the training stage when calcOOBError is set to true. |
| [TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) | [**getTermCriteria**](http://docs.google.com/org/opencv/ml/RTrees.html#getTermCriteria())() SEE: setTermCriteria |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**getVarImportance**](http://docs.google.com/org/opencv/ml/RTrees.html#getVarImportance())() Returns the variable importance array. |
| void | [**getVotes**](http://docs.google.com/org/opencv/ml/RTrees.html#getVotes(org.opencv.core.Mat,%20org.opencv.core.Mat,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) samples, [Mat](http://docs.google.com/org/opencv/core/Mat.html) results, int flags) Returns the result of each individual tree in the forest. |
| static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) | [**load**](http://docs.google.com/org/opencv/ml/RTrees.html#load(java.lang.String))(java.lang.String filepath) Loads and creates a serialized RTree from a file Use RTree::save to serialize and store an RTree to disk. |
| static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) | [**load**](http://docs.google.com/org/opencv/ml/RTrees.html#load(java.lang.String,%20java.lang.String))(java.lang.String filepath, java.lang.String nodeName) Loads and creates a serialized RTree from a file Use RTree::save to serialize and store an RTree to disk. |
| void | [**setActiveVarCount**](http://docs.google.com/org/opencv/ml/RTrees.html#setActiveVarCount(int))(int val) getActiveVarCount SEE: getActiveVarCount |
| void | [**setCalculateVarImportance**](http://docs.google.com/org/opencv/ml/RTrees.html#setCalculateVarImportance(boolean))(boolean val) getCalculateVarImportance SEE: getCalculateVarImportance |
| void | [**setTermCriteria**](http://docs.google.com/org/opencv/ml/RTrees.html#setTermCriteria(org.opencv.core.TermCriteria))([TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) val) getTermCriteria SEE: getTermCriteria |

### Methods inherited from class org.opencv.ml.[**DTrees**](http://docs.google.com/org/opencv/ml/DTrees.html)[getCVFolds](http://docs.google.com/org/opencv/ml/DTrees.html#getCVFolds()), [getMaxCategories](http://docs.google.com/org/opencv/ml/DTrees.html#getMaxCategories()), [getMaxDepth](http://docs.google.com/org/opencv/ml/DTrees.html#getMaxDepth()), [getMinSampleCount](http://docs.google.com/org/opencv/ml/DTrees.html#getMinSampleCount()), [getPriors](http://docs.google.com/org/opencv/ml/DTrees.html#getPriors()), [getRegressionAccuracy](http://docs.google.com/org/opencv/ml/DTrees.html#getRegressionAccuracy()), [getTruncatePrunedTree](http://docs.google.com/org/opencv/ml/DTrees.html#getTruncatePrunedTree()), [getUse1SERule](http://docs.google.com/org/opencv/ml/DTrees.html#getUse1SERule()), [getUseSurrogates](http://docs.google.com/org/opencv/ml/DTrees.html#getUseSurrogates()), [setCVFolds](http://docs.google.com/org/opencv/ml/DTrees.html#setCVFolds(int)), [setMaxCategories](http://docs.google.com/org/opencv/ml/DTrees.html#setMaxCategories(int)), [setMaxDepth](http://docs.google.com/org/opencv/ml/DTrees.html#setMaxDepth(int)), [setMinSampleCount](http://docs.google.com/org/opencv/ml/DTrees.html#setMinSampleCount(int)), [setPriors](http://docs.google.com/org/opencv/ml/DTrees.html#setPriors(org.opencv.core.Mat)), [setRegressionAccuracy](http://docs.google.com/org/opencv/ml/DTrees.html#setRegressionAccuracy(float)), [setTruncatePrunedTree](http://docs.google.com/org/opencv/ml/DTrees.html#setTruncatePrunedTree(boolean)), [setUse1SERule](http://docs.google.com/org/opencv/ml/DTrees.html#setUse1SERule(boolean)), [setUseSurrogates](http://docs.google.com/org/opencv/ml/DTrees.html#setUseSurrogates(boolean))

### Methods inherited from class org.opencv.ml.[**StatModel**](http://docs.google.com/org/opencv/ml/StatModel.html)[calcError](http://docs.google.com/org/opencv/ml/StatModel.html#calcError(org.opencv.ml.TrainData,%20boolean,%20org.opencv.core.Mat)), [empty](http://docs.google.com/org/opencv/ml/StatModel.html#empty()), [getVarCount](http://docs.google.com/org/opencv/ml/StatModel.html#getVarCount()), [isClassifier](http://docs.google.com/org/opencv/ml/StatModel.html#isClassifier()), [isTrained](http://docs.google.com/org/opencv/ml/StatModel.html#isTrained()), [predict](http://docs.google.com/org/opencv/ml/StatModel.html#predict(org.opencv.core.Mat)), [predict](http://docs.google.com/org/opencv/ml/StatModel.html#predict(org.opencv.core.Mat,%20org.opencv.core.Mat)), [predict](http://docs.google.com/org/opencv/ml/StatModel.html#predict(org.opencv.core.Mat,%20org.opencv.core.Mat,%20int)), [train](http://docs.google.com/org/opencv/ml/StatModel.html#train(org.opencv.core.Mat,%20int,%20org.opencv.core.Mat)), [train](http://docs.google.com/org/opencv/ml/StatModel.html#train(org.opencv.ml.TrainData)), [train](http://docs.google.com/org/opencv/ml/StatModel.html#train(org.opencv.ml.TrainData,%20int))

### Methods inherited from class org.opencv.core.[**Algorithm**](http://docs.google.com/org/opencv/core/Algorithm.html)[clear](http://docs.google.com/org/opencv/core/Algorithm.html#clear()), [getDefaultName](http://docs.google.com/org/opencv/core/Algorithm.html#getDefaultName()), [getNativeObjAddr](http://docs.google.com/org/opencv/core/Algorithm.html#getNativeObjAddr()), [save](http://docs.google.com/org/opencv/core/Algorithm.html#save(java.lang.String))

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

### Method Detail

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

#### create public static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) create() Creates the empty model. Use StatModel::train to train the model, StatModel::train to create and train the model, Algorithm::load to load the pre-trained model.Returns:automatically generated

#### getActiveVarCount public int getActiveVarCount() SEE: setActiveVarCountReturns:automatically generated

#### getCalculateVarImportance public boolean getCalculateVarImportance() SEE: setCalculateVarImportanceReturns:automatically generated

#### getOOBError public double getOOBError() Returns the OOB error value, computed at the training stage when calcOOBError is set to true. If this flag was set to false, 0 is returned. The OOB error is also scaled by sample weighting.Returns:automatically generated

#### getTermCriteria public [TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) getTermCriteria() SEE: setTermCriteriaReturns:automatically generated

#### getVarImportance public [Mat](http://docs.google.com/org/opencv/core/Mat.html) getVarImportance() Returns the variable importance array. The method returns the variable importance vector, computed at the training stage when CalculateVarImportance is set to true. If this flag was set to false, the empty matrix is returned.Returns:automatically generated

#### getVotes public void getVotes([Mat](http://docs.google.com/org/opencv/core/Mat.html) samples, [Mat](http://docs.google.com/org/opencv/core/Mat.html) results, int flags) Returns the result of each individual tree in the forest. In case the model is a regression problem, the method will return each of the trees' results for each of the sample cases. If the model is a classifier, it will return a Mat with samples + 1 rows, where the first row gives the class number and the following rows return the votes each class had for each sample.Parameters:samples - Array containing the samples for which votes will be calculated.results - Array where the result of the calculation will be written.flags - Flags for defining the type of RTrees.

#### load public static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) load(java.lang.String filepath) Loads and creates a serialized RTree from a file Use RTree::save to serialize and store an RTree to disk. Load the RTree from this file again, by calling this function with the path to the file. Optionally specify the node for the file containing the classifierParameters:filepath - path to serialized RTree Returns:automatically generated

#### load public static [RTrees](http://docs.google.com/org/opencv/ml/RTrees.html) load(java.lang.String filepath, java.lang.String nodeName) Loads and creates a serialized RTree from a file Use RTree::save to serialize and store an RTree to disk. Load the RTree from this file again, by calling this function with the path to the file. Optionally specify the node for the file containing the classifierParameters:filepath - path to serialized RTreenodeName - name of node containing the classifier Returns:automatically generated

#### setActiveVarCount public void setActiveVarCount(int val) getActiveVarCount SEE: getActiveVarCountParameters:val - automatically generated

#### setCalculateVarImportance public void setCalculateVarImportance(boolean val) getCalculateVarImportance SEE: getCalculateVarImportanceParameters:val - automatically generated

#### setTermCriteria public void setTermCriteria([TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) val) getTermCriteria SEE: getTermCriteriaParameters:val - 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/ml/ParamGrid.html)
* [Next Class](http://docs.google.com/org/opencv/ml/StatModel.html)
* [Frames](http://docs.google.com/index.html?org/opencv/ml/RTrees.html)
* [No Frames](http://docs.google.com/RTrees.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#2et92p0) |
* Constr |
* [Method](#3dy6vkm)
* Detail:
* Field |
* Constr |
* [Method](#3rdcrjn)

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