***Category***

*Random Forest HOG*

***Random Forest***

**Author: Michał Pomarański, Po-Tsung Chiu, Chun-Jen Peng**

Version : 2.0

Publication : 22/01/2014

Copyright:

**Modeliosoft**

MINES Paristech

Table of Contents

[1 Introduction 3](#_Toc410005045)

[2 Dictionnaire 4](#_Toc410005046)

[2.1 Dictionnaire de forêts decisionelles Aléatoires 4](#_Toc410005047)

[2.2 Dictionnaire de L’histogramme de gradient orienté (HOG) 6](#_Toc410005048)

[3 Use Cases 7](#_Toc410005049)

[3.1 Actors 7](#_Toc410005050)

[3.2 Use Cases 7](#_Toc410005051)

[3.2.1 Use Case "Avant projet Use Case diagram" 8](#_Toc410005052)

[4 Package Index 9](#_Toc410005053)

[5 Package "RandomForestHOG" 10](#_Toc410005054)

[6 Package "HOG" 11](#_Toc410005055)

[6.1 Class "HOGParam" 12](#_Toc410005056)

[6.2 Class "HOG" 14](#_Toc410005057)

[6.3 Class "HOGAppli" 17](#_Toc410005058)

[7 Package "DecisionTree" 19](#_Toc410005059)

[7.1 Class "DecisionTree" 20](#_Toc410005060)

[7.2 Class "TreeNode" 23](#_Toc410005061)

[8 Package "RandomForest" 25](#_Toc410005062)

[8.1 Class "RandomForest" 26](#_Toc410005063)

[8.2 Class "RandomForestLearner" 28](#_Toc410005064)

[8.3 Class "MainRun" 31](#_Toc410005065)

[9 Package "NotifyingThread" 32](#_Toc410005066)

[9.1 Interface "ThreadCompleteListener" 33](#_Toc410005067)

[9.2 Class "NotifyingThread" 34](#_Toc410005068)

[10 Package "Utils" 35](#_Toc410005069)

[10.1 Class "ExtDataParser" 36](#_Toc410005070)

[10.2 Class "Helper" 37](#_Toc410005071)

[10.3 Class "DataVector" 38](#_Toc410005072)

# Introduction

Les ***forêts décisionnelles aléatoires*** (de l'anglais « *Random decision forest* ») ont été formellement proposées en [2001](http://fr.wikipedia.org/wiki/2001_en_informatique) par [Leo Breiman](http://fr.wikipedia.org/wiki/Leo_Breiman) et [Adèle Cutler](http://fr.wikipedia.org/w/index.php?title=Ad%C3%A8le_Cutler&action=edit&redlink=1). Elles font partie des techniques d'[apprentissage automatique](http://fr.wikipedia.org/wiki/Apprentissage_automatique). Cet algorithme combine les concepts de sous-espaces aléatoires et de « [*bagging*](http://fr.wikipedia.org/wiki/Bagging) ». L'algorithme des forêts d'arbres décisionnels effectuer un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents et des sous-ensembles totalement différents de variables (composantes des vecteurs de “features ").

Un ***histogramme de gradient orienté*** (HOG) est une [caractéristique](http://fr.wikipedia.org/wiki/Caract%C3%A9ristique_(vision_par_ordinateur)) utilisée en [vision par ordinateur](http://fr.wikipedia.org/wiki/Vision_par_ordinateur) pour la classification d’images et la détection visuelle d’objets d’une catégorie donnée. La technique calcule des [histogrammes](http://fr.wikipedia.org/wiki/Histogramme) locaux de l'orientation du [gradient](http://fr.wikipedia.org/wiki/Gradient) sur une grille dense, c'est-à-dire sur des zones régulièrement réparties sur l'image. Elle possède des points communs avec les ‘‘Scale-invariant feature transform’’ ([SIFT](http://fr.wikipedia.org/wiki/SIFT)), les [Shape contexts](http://fr.wikipedia.org/wiki/Shape_context) et les histogrammes d'orientation de contours, mais s'en diffère notamment par l'utilisation d'une grille dense. La méthode s'est montrée particulièrement efficace pour la [détection de personnes](http://fr.wikipedia.org/wiki/D%C3%A9tection_de_personne).

Nous allons utiliser les forêts décisionnelles aléatoires pour classifier des images qui ont été transformées en histogramme de gradient orienté.

# Dictionnaire

## Dictionnaire de forêts decisionelles Aléatoires

|  |  |  |
| --- | --- | --- |
| **Nom** | **Définition** | ***Nom UML*** |
| arbre de décision | flow-chart like structure  node => some attribute  branch => outcome of test  path from root to leaf => classification rule | Class DecisionTree |
| forêt d’arbres [décision](http://en.wikipedia.org/wiki/Flowchart)nels | a classifier algorithm which contains multiple decision trees and each tree vote for classification | Class RandomForest |
| L’apprentissage des arbres décisionnels | The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x1, …, xn with responses Y = y1 through yn, bagging repeatedly selects a bootstrap sample of the training set and fits trees to these samples | Method bootStrapSample  De la class DecisionTree |
| L’histogramme de gradient orienté (HOG) | a descriptor represented by histograms of the “Orientation” of gradients in the images | Histogram of Oriented Gradients |
| classification d’image | Classify an image into one of several categories | Method classify(Sample) in any subclass of Classifier |
| Classifier (6.4 p.19) | a class that implements a prototype of classifiers such as AdaBoost or MultilayerPerception | classifier.Classifier in jLevis |
| Learner (7.3 p.27) | an abstract class that reflects a learning system | learner.Learner in jLevis |
| Sample (9.9 p.50) | a class that reflects an image | sample.Sample in jLevis |
| jLevis | existing framework containing classification algorithms | jLevis |
| Vote majoritaire | la classification fournie par la forêt | Méthode voteMajor  de la classe RandomForest |
| Vote d’un arbre | chaque arbre vote pour une des classes | Méthode evaluate de la classe DecisionTree |
| bagging  ([bootstrap aggregating](http://en.wikipedia.org/wiki/Bootstrap_aggregating)) | also known as “[bootstrap aggregating”. A](http://en.wikipedia.org/wiki/Bootstrap_aggregating) [machine learning ensemble](http://en.wikipedia.org/wiki/Ensemble_learning)[meta-algorithm](http://en.wikipedia.org/wiki/Meta-algorithm) [designed to improve the stability and accuracy of](http://en.wikipedia.org/wiki/Bootstrap_aggregating) [machine learning](http://en.wikipedia.org/wiki/Machine_learning) [algorithms used in](http://en.wikipedia.org/wiki/Bootstrap_aggregating)[statistical classification](http://en.wikipedia.org/wiki/Statistical_classification) [and](http://en.wikipedia.org/wiki/Bootstrap_aggregating) [regression](http://en.wikipedia.org/wiki/Regression_analysis).  It also reduces [variance](http://en.wikipedia.org/wiki/Variance) and helps to avoid [overfitting](http://en.wikipedia.org/wiki/Overfitting) | Probablement un attribut de la classe DecisionTree, qui sera fixé lors de création, et utilisé dans méthode learn(Database) |
| arbre décisionnel classique | Un arbre de décision est un outil d'aide à la décision qui représente la situation plus ou moins complexe que l'on représente sous la forme graphique d'un arbre de façon à faire apparaître à l'extrémité de chaque branche (ou feuille) les différents résultats possibles en fonction des décisions prises à chaque étape. | Class DecisionTree |

## Dictionnaire de L’histogramme de gradient orienté (HOG)

|  |  |  |
| --- | --- | --- |
| Histogram of oriented gradients | features of image pixels which are calculated with 1-D derivative masks | Attribut  \_histogram  de classe HOG |
| block normalization | a method to normalize the intensity of cells within blocks (larger region)  it results in better invariance to changes in illumination and shadowing | method blockNormalization of class HOG |
| 1-D derivative mask | kernels filtering the color or intensity data of images to compute gradient value  usually applied vertically and horizontally which have forms like:  [-1, 0, 1]\text{ and }[-1, 0, 1]^T.\,  https://lh3.googleusercontent.com/udUYZth8nhQg2ZuNy77_yDt1ldjZKgXbIA7tgdnptHCMOrRgzpDXf06NGfigABv-hfty7BIn3ebnSDmWeaZh888l3IK3WM76B_4lxTJvliwG09Xj-fQNsMjmSwoTZiQfuQ | x |
| Image Filter (9.3 p.45) | a class that provides DSP Image filters | x |
| Image Utils (9.4 p.45) | a class that provides utilities such as resize image, get image data, save to buffer...etc | x |
| Integral Image (9.5 p.46) | a class that helps us compute integral images | x |

# Use Cases

## Actors

| Actor | Description |
| --- | --- |
| Utilisateur final | the final user is a human operator who choose the images and give it to the system for classification |
| jLevis | existing framework containing classification algorithms |

Table 1 Table of Actors

## Use Cases

| Use-Cases | Description |
| --- | --- |
| Choisir les images | choose the images manually |
| Classification des images | give the chosen images for classification |
| Learn an image classifier | classify a set of images into different categories by random forest |
| Creation d'une forêt décisionnelle aléatoire | create a random forest which contains multiple decision trees and each tree vote for classification |
| choisir profondeur maximale des arbres de la forêt | choose the maximum depth of trees in the random forest |
| choisir nombre d’arbres de décision contenus dans la forêt aléatoire | choose the number of trees in this random tree |
| Creation d'une arbre de décision | create a flow-chart like structure which classifies information through each node, from root to leaf |
| Choisir un sous-ensemble aléatoire de données d'apprentissage | select a bootstrap sample of the training set and fits trees to these samples in the "bagging" process |
| Apprentissage d'une forêt | The training algorithm for random forests applies the general technique of bootstrap aggregating (bagging) to tree learners. |
| Fournir des données d'apprentissage | provide the data of the HoG process |
| Fournir HOG des images d'apprentissage | The technique which counts occurrences of gradient orientation in blocks of an image. |
| Informations sur la forêt | attributes of the random forest |
| Le nombre d'arbres dans la forêt construite | number of decision trees in the random forest |
| Calculer l'erreur | calculate the testing error number of trees have not been fit |
| Calculer l'erreur d'apprentissage | calculate the learning error number of trees have not been fit |
| Choisir classifier | choose a classifier |
| Charger et pré-traiter images | load the features of image pixels calculated by HOG |
| Pré-calculer HOG | use "Histogram of oriented gradients" (HOG) to calculate features of image pixels |
| Charger classifieur (depuis fichier) | load classifier from file |
| Sauvegarder classifieur (sur fichier) | Save the data and information of classifier |
| Choix du pour cent des variables à utiliser par chaque arbre | calculate the percentage of the variables to be used by each tree |
| Choix du subset de variables à utiliser | Choosing the subset of variables to be used as samples in a decision tree |

Table 2 Table of Use Cases

### Use Case "Avant projet Use Case diagram"

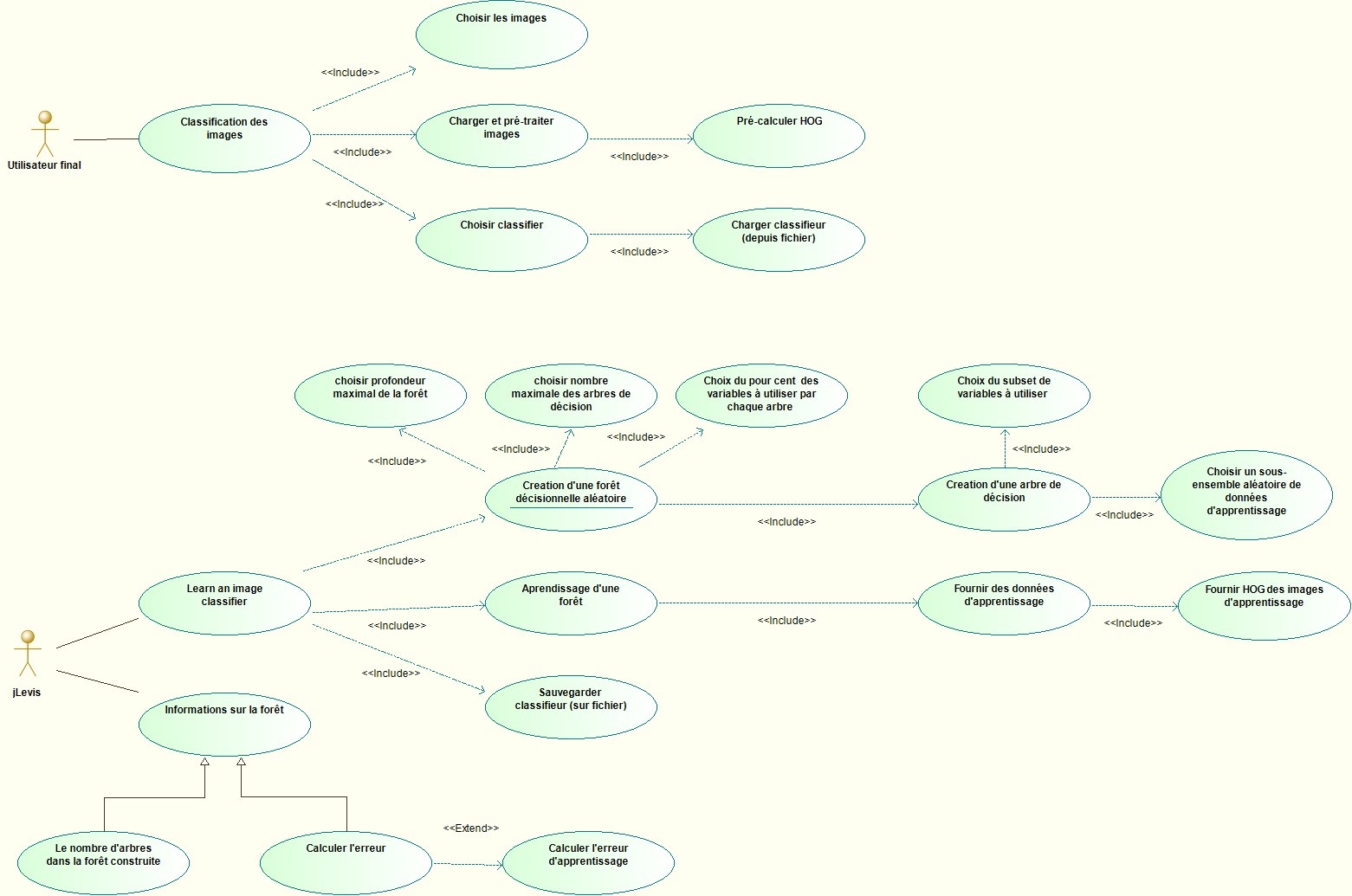


Figure 1 : Avant projet Use Case diagram

# Package Index

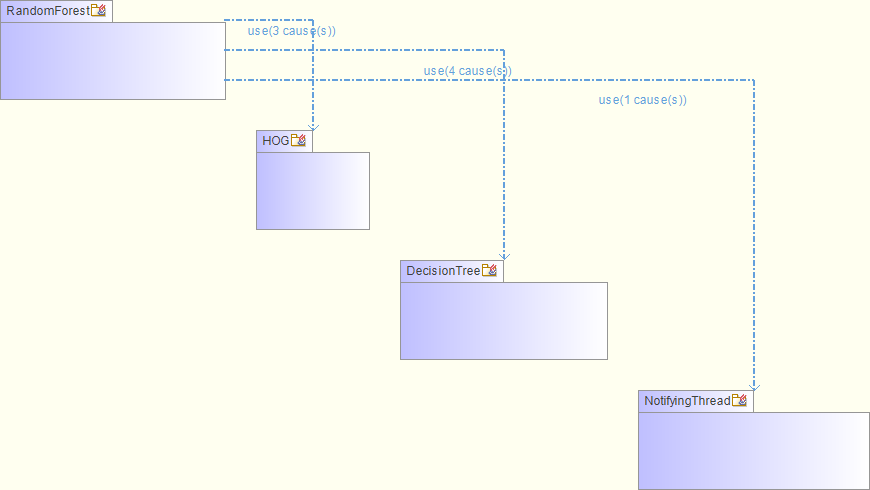
|  |  |
| --- | --- |
| [RandomForestHOG](#_3c0eec90-a163-4364-8ec1-855c885b184b) | Main Package |
| [HOG](#_3f901274-58e3-42e4-98b3-df441014db6d) | Package responsible for creating HOG representations of images |
| [DecisionTree](#_59aa1dec-a6d5-40e5-bf32-21f89bc84295) | Package contains the classes used in decision tree |
| [RandomForest](#_ffda6e8b-f5ec-48e6-9347-56af0b87eb91) | Package contains class RandomForest and class RandomForestLearner. |
| [NotifyingThread](#_7f56ec47-8192-48b7-ae3d-b8281cd23271) | Implement listener for threading |
| [Utils](#_7f56ec47-8192-48b7-ae3d-b8281cd23271) | Package contains utility classes |

# Package "RandomForestHOG"

Stereotypes: Java Package

| Name | Summary |
| --- | --- |
| [HOG](#_3f901274-58e3-42e4-98b3-df441014db6d) | Package responsible for creating HOG representations of images |
| [DecisionTree](#_59aa1dec-a6d5-40e5-bf32-21f89bc84295) | Package contains the classes used in decision tree |
| [RandomForest](#_ffda6e8b-f5ec-48e6-9347-56af0b87eb91) | Package contains class RandomForest and class RandomForestLearner. |
| [NotifyingThread](#_7f56ec47-8192-48b7-ae3d-b8281cd23271) | Implement listener for threading |

Table 3 Owned Packages of Package "RandomForestHOG"

 Figure 2 RandomForestHOG (subpackage\_structure\_autodiagram)

# Package "HOG"

from Package [RandomForestHOG](#_3c0eec90-a163-4364-8ec1-855c885b184b)

Stereotypes: Java Package

| Name | Summary |
| --- | --- |
| [HOGParam](#_b30f2d3b-3dd3-4928-8c2e-210cde6bac4c) | Class contains a set of parameters for HOG |
| [HOG](#_77b78d4d-6afc-4bc3-b438-316a59622fd8) | Class responsible for creating HOG representations of images |
| [HOGAppli](#_7d92f50b-23ca-492e-955c-d6ec3ca3f45d) | convert an image into HOG data set |

Table 4 Owned Classes of Package "HOG"

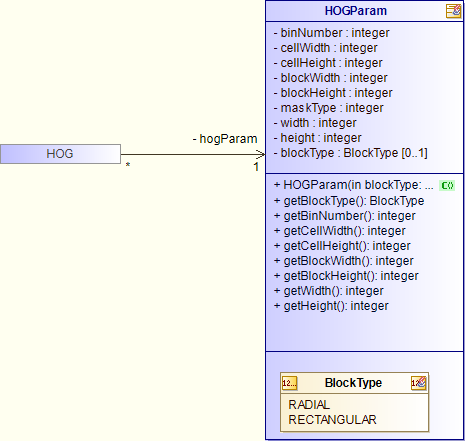
## Class "HOGParam"

from Package .RandomForestHOG.[HOG](#_3f901274-58e3-42e4-98b3-df441014db6d)

Stereotypes: Java Class

This class bundles the essential parameters to perform HOG.

Figure 3 HOGParam



| Name | Description |
| --- | --- |
| HOGParam (In blockType BlockType,In binNumber integer,In cellWidth integer,In cellHeight integer,In blockWidth integer,In blockHeight integer,In maskType integer,Inout width integer,Inout height integer) | initialize instance of HOGParam with manual setting |
| BlockType getBlockType () | return blockType variable |
| integer getBinNumber () | return binNumber variable |
| integer getCellWidth () | return cellWidth variable |
| integer getCellHeight () | return cellHeight variable |
| integer getBlockWidth () | return blockWidth variable |
| integer getBlockHeight () | return blockHeight variable |
| integer getWidth () | return width variable |
| integer getHeight () | return height variable |

Table 5 Operations of Class "HOGParam"

| Name | Description |
| --- | --- |
| nBin : [1..1] integer | number of bins for creating histogram |
| cellWidth : [1..1] integer | width of each cell |
| cellHeight : [1..1] integer | height of each cell |
| blockWidth : [1..1] integer | width of each block |
| blockHeight : [1..1] integer | height of each block |
| maskType : [1..1] integer | type of the mask to calculate gradients |
| width : [1..1] integer | number of pixels for the width of each image |
| height : [1..1] integer | number of pixels for the height of each image |
| blockType : [0..1] BlockType | shape of the block |

Table 6 Attributes of Class "HOGParam"

| Name | Values | Description |
| --- | --- | --- |
| BlockType | RADIAL  RECTANGULAR | BlockType contains two type, RADIAL and RECTANGULAR |

Table 7 Owned Enumerations of Class "HOGParam"

## Class "HOG"

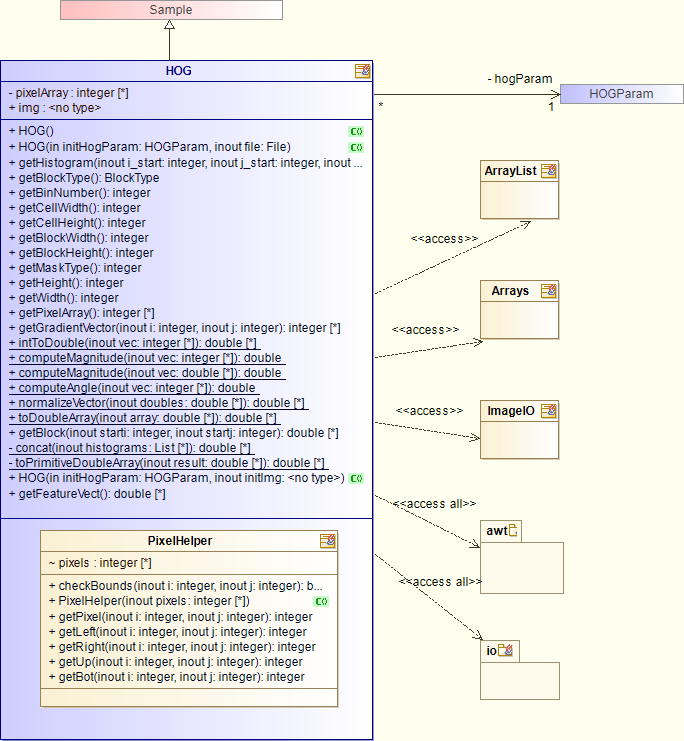
from Package .RandomForestHOG.[HOG](#_3f901274-58e3-42e4-98b3-df441014db6d)

Inherits from: [Sample](#_cc4ac9ce-f533-4988-a291-00d126112807)

Stereotypes: Java Class

This class implements the operations involved in HOG, and create arrays of pixel's data of each stage.

Figure 4 HOG



| Name | Description |
| --- | --- |
| HOG () | initialize HOG instance with default parameter setting |
| HOG (In initHogParam HOGParam,Inout file File) | initialize HOG instance with certain parameter setting |
| HOG (In initHogParam HOGParam,Inout initImg ) |  |
| double getHistogram (Inout i\_start integer,Inout j\_start integer,Inout end\_i integer,Inout end\_j integer) | calculate the histogram of a block and return a double array |
| BlockType getBlockType () | return blockType of HOGParam |
| integer getBinNumber () | return binNumber of HOGParam |
| integer getCellWidth () | return cellWidth of HOGParam |
| integer getCellHeight () | return cellHeight of HOGParam |
| integer getBlockWidth () | return blockWidth of HOGParam |
| integer getBlockHeight () | return blockHeight of HOGParam |
| integer getMaskType () | return maskType of HOGParam |
| integer getHeight () | return height of HOGParam |
| integer getWidth () | return width of HOGParam |
| integer getPixelArray () | return the grayscale of the image |
| integer getGradientVector (Inout i integer,Inout j integer) | get the gradient vector of each pixel |
| double intToDouble (Inout vec integer) | convert an integer array to a double array |
| double computeMagnitude (Inout vec integer) | compute the magnitude of the gradient vector |
| double computeMagnitude (Inout vec double) | compute the magnitude of the gradient vector |
| double computeAngle (Inout vec integer) | compute the angle of the gradient vector |
| double normalizeVector (Inout doubles double) | normalize the gradient vector |
| double toDoubleArray (Inout array double) | convert primitive double array to Double array |
| double getBlock (Inout starti integer,Inout startj integer) | getting historams from each cell in a block |
| double concat (Inout histograms List) | concat all historam data in the same block into a double array |
| double toPrimitiveDoubleArray (Inout result double) | convert a Double array to a primitive double array |
| double getFeatureVect () | return all the histogram data of an image |

Table 8 Operations of Class "HOG"

| Name | Description |
| --- | --- |
| \_featureVector : [0..\*] integer | array of feature vectors of each pixel after calculation |
| img : [1..1] |  |

Table 9 Attributes of Class "HOG"

| Name | Description |
| --- | --- |
| ->hogParam : [1..1] [HOGParam](#_b30f2d3b-3dd3-4928-8c2e-210cde6bac4c) | set of HOG parameters used in HOG computation |

Table 10 Associations of Class "HOG"

| Name | Summary |
| --- | --- |
| [PixelHelper](#_d47babce-5ef4-4949-a3db-8ed5403b6789) |  |

Table 11 Owned Classes of Class "HOG"

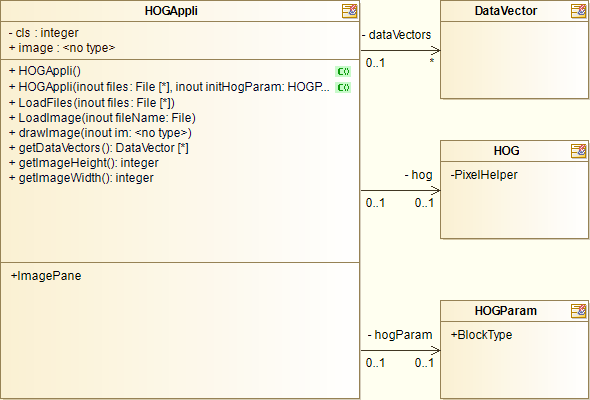
## Class "HOGAppli"

from Package .RandomForestHOG.[HOG](#_3f901274-58e3-42e4-98b3-df441014db6d)

Stereotypes: Java Class

load files, caculate the HOGs and generate data sets

Figure 5 HOGAppli Class diagram



| Name | Description |
| --- | --- |
| HOGAppli () | set up hogParam |
| HOGAppli (Inout files File,Inout initHogParam HOGParam) | input file and setup hogParam |
| LoadFiles (Inout files File) | recursively load all files in the input directory |
| LoadImage (Inout fileName File) | load and resize image and compute its HOG |
| drawImage (Inout im ) | show image on the screen |
| DataVector getDataVectors () | return dataVector variable |
| integer getImageHeight () | return height of the image |
| integer getImageWidth () | return width of the image |

Table 12 Operations of Class "HOGAppli"

| Name | Description |
| --- | --- |
| cls : [1..1] integer | the class of the input image |
| image : [1..1] | input image |

Table 13 Attributes of Class "HOGAppli"

| Name | Description |
| --- | --- |
| ->dataVectors : [0..\*] [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) | list of data from HOG |
| ->hog : [0..1] [HOG](#_77b78d4d-6afc-4bc3-b438-316a59622fd8) | the HOG instance |
| ->hogParam : [0..1] [HOGParam](#_b30f2d3b-3dd3-4928-8c2e-210cde6bac4c) | the hogParam instance |

Table 14 Associations of Class "HOGAppli"

| Name | Summary |
| --- | --- |
| [ImagePane](#_9b7fe7ec-70fe-4b90-9d57-2a7fce2722cb) | ImagePane extends JPanel |

Table 15 Owned Classes of Class "HOGAppli"

# Package "DecisionTree"

from Package .[RandomForestHOG](#_3c0eec90-a163-4364-8ec1-855c885b184b)

Stereotypes: Java Package

Create decision tree and classify testing data

| Name | Summary |
| --- | --- |
| [DecisionTree](#_61261918-d6ad-4d4d-a19f-e6c7088f5dd6) | Class contains interface for using Decision Trees. |
| [TreeNode](#_67321fdf-07c5-4b25-973e-2c0c213fa851) | Class defines the nodes of a tree data structure |

Table 16 Owned Classes of Package "DecisionTree"

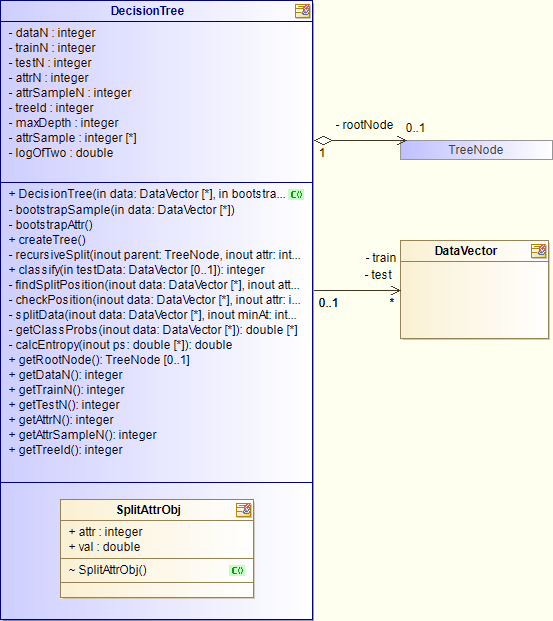
## Class "DecisionTree"

from Package .RandomForestHOG.[DecisionTree](#_59aa1dec-a6d5-40e5-bf32-21f89bc84295)

Stereotypes: Java Class

This class implements operations about decision tree in random forest learning.

Figure 6 DecisionTree



| Name | Description |
| --- | --- |
| DecisionTree (In data DataVector,In bootstrapRate double,In attrSampleN integer,In maxDepth integer,In treeId integer) | constructs a decision tree from a data matrix. |
| bootStrapSample (In data DataVector) | create a boostrap sample of size trainN |
| bootStrapAttr () | selects attributes of size attrN from sample |
| createTree () | creates the decision tree according to the specifications of random forest trees |
| recursiveSplit (In parent TreeNode,Inout attr integer) | critical function to create the decision tree with the selected attributes of bootstrap sample |
| findSplitPosition (Inout data DataVector,Inout attr integer,Inout attrObj SplitAttrObj) | get attribute index and value to split data |
| double checkPosition (Inout data DataVector,Inout attr integer,Inout val double) | return the entropy of current split position |
| List splitData (Inout data DataVector,Inout minAt integer,Inout minAtVal double) | split data with given attribute index and value |
| double getClassProbs (Inout data DataVector) | get probabilities of classes |
| double calcEntropy (Inout ps double) | calculate the entropy given classes' probabilities |
| integer classify (In testData DataVector) | traverses the tree and returns the prediction of the given test data |
| TreeNode getRootNode () | return rootNode |
| integer getDataN () | return dataN |
| integer getTrainN () | return trainN |
| integer getTestN () | return testN |
| integer getAttrN () | return attrN |
| integer getAttrSampleN () | return attrSampleN |
| integer getTreeId () | return treeId |

Table 17 Operations of Class "DecisionTree"

| Name | Description |
| --- | --- |
| dataN : [1..1] integer | total size of the data (training and testing) |
| trainN : [1..1] integer | size of the bootstrap samples to train (assigned by RFLearner) |
| attrN : [1..1] integer | size of the attributes to train (assigned by RFLearner) |
| attrN : [1..1] integer | size of all attributes |
| attrSampleN : [1..1] integer | size of bootstrapped attributes to train |
| treeId : [1..1] integer | the number of this tree |
| maxDepth : [1..1] integer | maximum level of decision tree |
| attrSample : [0..\*] integer | an array of bootstrapped attributes' index |
| logOfTwo : [1..1] double | log 2 |

Table 18 Attributes of Class "DecisionTree"

| Name | Description |
| --- | --- |
| ->rootNode : [0..1] [TreeNode](#_67321fdf-07c5-4b25-973e-2c0c213fa851) | root node of this decision tree |
| ->train : [0..\*] [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) | training dataset |
| ->test : [0..\*] [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) | testing dataset |

Table 19 Associations of Class "DecisionTree"

| Name | Summary |
| --- | --- |
| [SplitAttrObj](#_022c19a8-d263-4638-a209-2cfb9bad6764) | Attribute index and value |

Table 20 Owned Classes of Class "DecisionTree"

## Class "TreeNode"

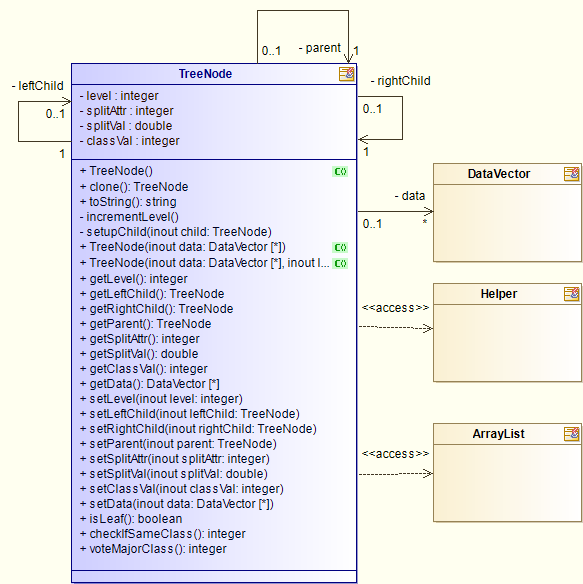
from Package .RandomForestHOG.[DecisionTree](#_59aa1dec-a6d5-40e5-bf32-21f89bc84295)

Implements: [Cloneable](#_72e0c561-4e1e-4bb1-a674-1d91d6f65226)

Stereotypes: Java Class

This class contains the elements of a node in tree data structure.

Figure 7 TreeNode



| Name | Description |
| --- | --- |
| TreeNode () | constructs the tree node |
| TreeNode (Inout data DataVector) | constructs the tree node |
| TreeNode (Inout data DataVector,Inout level integer,Inout splitAttr integer,Inout splitVal double) | constructs the tree node |
| TreeNode clone () | copy TreeNode instance |
| string toString () | display TreeNode as string |
| incrementLevel () | increment level |
| setupChild (Inout child TreeNode) | set up child nodes |
| integer getLevel () | return level |
| TreeNode getLeftChild () | return leftChild |
| TreeNode getRightChild () | return rightChild |
| TreeNode getParent () | return parent |
| integer getSplitAttr () | return splitAttr |
| double getSplitVal () | return splitVal |
| integer getClassVal () | return classVal |
| DataVector getData () | return data |
| setLevel (Inout level integer) | set level |
| setLeftChild (Inout leftChild TreeNode) | set leftChild |
| setRightChild (Inout rightChild TreeNode) | set rightChild |
| setParent (Inout parent TreeNode) | set parent |
| setSplitAttr (Inout splitAttr integer) | set splitAttr |
| setSplitVal (Inout splitVal double) | set splitVal |
| setClassVal (Inout classVal integer) | set classVal |
| setData (Inout data DataVector) | set data |
| boolean isLeaf () | check if current node is leaf |
| integer checkIfSameClass () | check if data is the same class |
| integer voteMajorClass () | get majority of the class of current data |

Table 21 Operations of Class "TreeNode"

| Name | Description |
| --- | --- |
| level : [1..1] integer | current level in the tree |
| splitAttr : [1..1] integer | attribute to split on |
| splitVal : [1..1] double | value to split splitAttr |
| classVal : [1..1] integer |  |

Table 22 Attributes of Class "TreeNode"

| Name | Description |
| --- | --- |
| ->leftChild : [0..1] [TreeNode](#_67321fdf-07c5-4b25-973e-2c0c213fa851) | left child of this node |
| ->rightChild : [1..1] [TreeNode](#_67321fdf-07c5-4b25-973e-2c0c213fa851) | right child of this node |
| ->parent : [1..1] [TreeNode](#_67321fdf-07c5-4b25-973e-2c0c213fa851) | parent of this node |
| ->data : [0..\*] [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) | data at current level |

Table 23 Associations of Class "TreeNode"

# Package "RandomForest"

from Package .[RandomForestHOG](#_3c0eec90-a163-4364-8ec1-855c885b184b)

Stereotypes: Java Package

| Name | Summary |
| --- | --- |
| [RandomForest](#_81236832-24f2-4e63-9b25-09e05cf5c4b1) | Class define the classifier generated by random forest. |
| [RandomForestLearner](#_f13ea57b-2648-48bf-8e5e-b1319a05eaba) | Class responsible for random forest learning |
| [MainRun](#_9529dee4-15ba-48ea-911d-4b1953bb1a0a) | Main class |

Table 24 Owned Classes of Package "RandomForest"

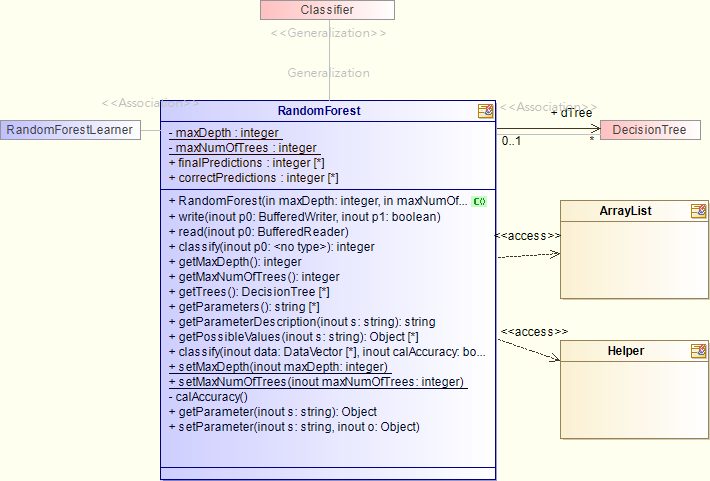
## Class "RandomForest"

from Package .RandomForestHOG.[RandomForest](#_ffda6e8b-f5ec-48e6-9347-56af0b87eb91)

Stereotypes: Java Class

This class contains information about the classifier trained with random forest and provides operation to classify images.

Figure 8 RandomForest



| Name | Description |
| --- | --- |
| RandomForest (In maxDepth integer,In maxNumOfTrees integer) | the constructor of random forest |
| write (Inout p0 BufferedWriter,Inout p1 boolean) | implement write method of Classifier |
| read (Inout p0 BufferedReader) | implement read method of Classifier |
| integer classify (Inout p0 ) | classify dataset p0 |
| integer classify (Inout data DataVector,Inout calAccuracy boolean) | classify dataset and calculate accuracy |
| integer getMaxDepth () | return maxDepth |
| integer getMaxNumOfTrees () | return maxNumOfTrees |
| DecisionTree getTrees () | return dTree |
| string getParameters () | return forest parameter |
| string getParameterDescription (Inout s string) | return forest parameter description |
| Object getPossibleValues (Inout s string) | return possible values |
| setMaxDepth (Inout maxDepth integer) | set maxDepth |
| setMaxNumOfTrees (Inout maxNumOfTrees integer) | set maxNumOfTrees |
| calAccuracy () | calculate accuracy of classified predictions |
| Object getParameter (Inout s string) | return forest parameters |
| setParameter (Inout s string,Inout o Object) | set forest parameters |

Table 25 Operations of Class "RandomForest"

| Name | Description |
| --- | --- |
| maxDepth : [1..1] integer | maximum depth of tree, set to -1 if depth has no limit |
| maxNumOfTrees : [1..1] integer | maximum number of trees in random forest |
| finalPredictions : [0..\*] integer | forest predictions of test data |
| correctPredictions : [0..\*] integer | actual class of test data |

Table 26 Attributes of Class "RandomForest"

| Name | Description |
| --- | --- |
| ->dTree : [0..\*] [DecisionTree](#_61261918-d6ad-4d4d-a19f-e6c7088f5dd6) | trees in random forest |

Table 27 Associations of Class "RandomForest"

## Class "RandomForestLearner"

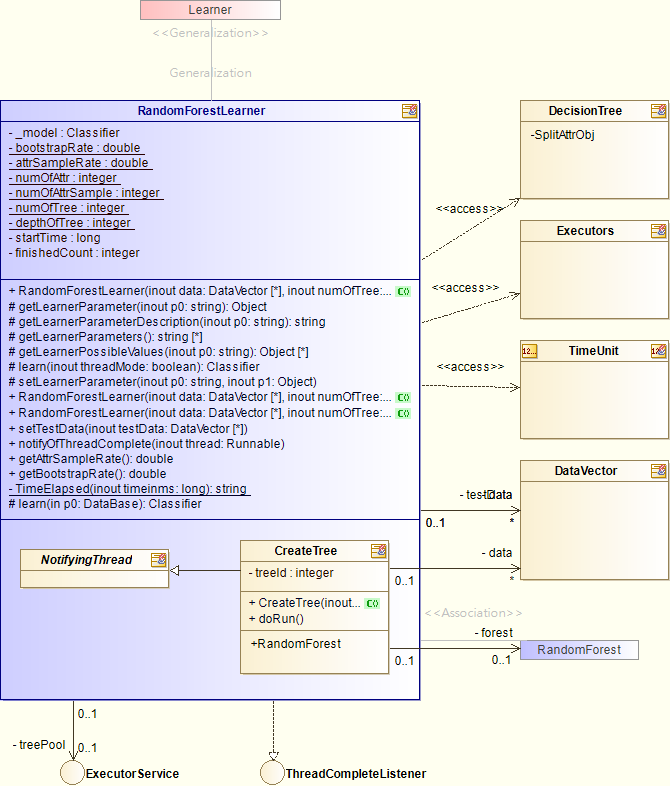
from Package .RandomForestHOG.[RandomForest](#_ffda6e8b-f5ec-48e6-9347-56af0b87eb91)

Implements: [ThreadCompleteListener](#_4793b759-d68c-45cd-9adc-47d2c9a2a913)

Stereotypes: Java Class

This class implements the learning algorithm of random forest inheriting from learner

Figure 9 RandomForestLearner



| Name | Description |
| --- | --- |
| RandomForestLearner (Inout data DataVector,Inout numOfTree integer,Inout depthOfTree integer) | The constructor of RandomForestLearner |
| RandomForestLearner (Inout data DataVector,Inout numOfTree integer,Inout depthOfTree integer,Inout attrSampleRate double) | constructor of RandomForestLearner |
| RandomForestLearner (Inout data DataVector,Inout numOfTree integer,Inout depthOfTree integer,Inout attrSampleRate double,Inout bootstrapRate double) | constructor of RandomForestLearner |
| Object getLearnerParameter (Inout p0 string) | implements getLearnerParameter method of Learner |
| string getLearnerParameterDescription (Inout p0 string) | implements getLearnerParameterDescription method of Learner |
| string getLearnerParameters () | implements getLearnerParameters method of Learner |
| Object getLearnerPossibleValues (Inout p0 string) | implements getLearnerPossibleValues method of Learner |
| Classifier learn (Inout threadMode boolean) | implements learn method of Learner and trains a random forest classifier |
| Classifier learn (In p0 DataBase) | train decision trees of random forest |
| setLearnerParameter (Inout p0 string,Inout p1 Object) | implements setLearnerParameter method of Learner |
| setTestData (Inout testData DataVector) | set testData |
| notifyOfThreadComplete (Inout thread Runnable) | callback method when thread finished |
| double getAttrSampleRate () | return attrSampleRate |
| double getBootstrapRate () | return bootstrapRate |
| string TimeElapsed (Inout timeinms long) | calculate time elapsed after timeinms |

Table 28 Operations of Class "RandomForestLearner"

| Name | Description |
| --- | --- |
| \_model : [1..1] Classifier | random forest mode |
| bootstrapRate : [1..1] double | ratio to split training and testing data |
| attrSampleRate : [1..1] double | ratio of bootstrapped attribute sample |
| numOfAttr : [1..1] integer | number of all attributes |
| numOfAttrSample : [1..1] integer | number of attribute samples |
| numOfTree : [1..1] integer | number of trees in random forest |
| depthOfTree : [1..1] integer | maximum depth of decision tree |
| startTime : [1..1] long | starting time of learn() |
| finishedCount : [1..1] integer | count of finished thread |

Table 29 Attributes of Class "RandomForestLearner"

| Name | Description |
| --- | --- |
| ->data : [0..\*] [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) | training dataset |
| ->testData : [0..\*] [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) | testing dataset |
| ->treePool : [0..1] [ExecutorService](#_e14d910e-86ce-4936-afd4-0cf095b3d1ad) | thread pool for learning |

Table 30 Associations of Class "RandomForestLearner"

| Name | Summary |
| --- | --- |
| [CreateTree](#_6a3a9384-4429-4893-bb47-2998f42f7436) | Create decision tree |
| [NotifyingThread](#_baa6354a-dcce-41f7-8277-96ee6145afeb) | Implement listener for threading |
| [DataVector](#_4ccce523-4653-4dc2-a5bc-b4bfbbe95a34) |  |

Table 31 Owned Classes of Class "RandomForestLearner"

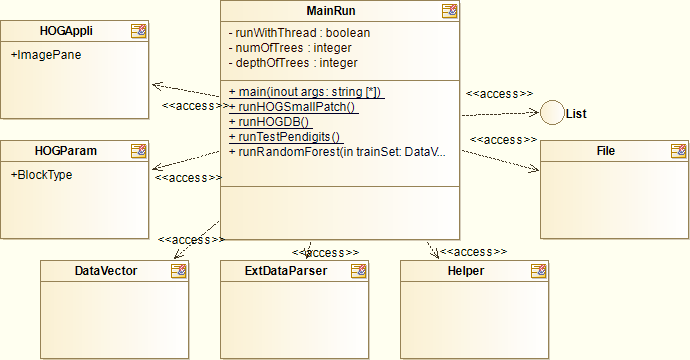
## Class "MainRun"

from Package .RandomForestHOG.[RandomForest](#_ffda6e8b-f5ec-48e6-9347-56af0b87eb91)

Stereotypes: Java Class

main running class with general testing

Figure 10 MainRun Class diagram



| Name | Description |
| --- | --- |
| main (Inout args string) | main method to start general test |
| runHOGSmallPatch () | test small patch of images |
| runHOGDB () | test images of DB |
| runTestPendigits () | test handwritten digits |
| runRandomForest (In trainSet DataVector,In testSet DataVector) | start random forest learning |

Table 32 Operations of Class "MainRun"

| Name | Description |
| --- | --- |
| runWithThread : [1..1] boolean | set to train random forest with threads |
| numOfTrees : [1..1] integer | number of decision trees in random forest |
| depthOfTrees : [1..1] integer | maximum depth of decision tree |

Table 33 Attributes of Class "MainRun"

# Package "NotifyingThread"

from Package .[RandomForestHOG](#_3c0eec90-a163-4364-8ec1-855c885b184b)

Stereotypes: Java Package

contain abstract class and interface for setting up listener of threads

| Name | Summary |
| --- | --- |
| [ThreadCompleteListener](#_4793b759-d68c-45cd-9adc-47d2c9a2a913) | Interface for thread listeners |

Table 34 Owned Interfaces of Package "NotifyingThread"

| Name | Summary |
| --- | --- |
| [NotifyingThread](#_39ad72c4-e6a0-4c94-879f-738d17b3aa02) | Implement listener for threading |

Table 35 Owned Classes of Package "NotifyingThread"

## Interface "ThreadCompleteListener"

from Package .RandomForestHOG.[NotifyingThread](#_7f56ec47-8192-48b7-ae3d-b8281cd23271)

Stereotypes: Java Interface

Notify listeners when thread finished

Figure 11 ThreadCompleteListener Class diagram



| Name | Description |
| --- | --- |
| notifyOfThreadComplete (In thread Runnable) | notify listener when thread finished |

Table 36 Operations of Interface "ThreadCompleteListener"

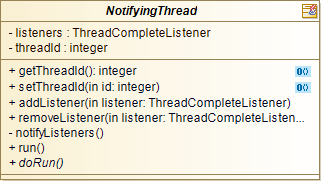
## Class "NotifyingThread"

from Package .RandomForestHOG.[NotifyingThread](#_7f56ec47-8192-48b7-ae3d-b8281cd23271)

Stereotypes: Java Class

Implement Runnable as abstract class to set up listener of threads

Figure 12 NotifyingThread Class diagram



| Name | Description |
| --- | --- |
| integer getThreadId () | return threadId |
| setThreadId (In id integer) | set threadId |
| addListener (In listener ThreadCompleteListener) | add new listener when starting new thread |
| removeListener (In listener ThreadCompleteListener) | remove listener |
| notifyListeners () | notify all listeners when current thread finished |
| run () | implement run of Runnable |
| doRun () | actual run method to implement |

Table 37 Operations of Class "NotifyingThread"

| Name | Description |
| --- | --- |
| listeners : [1..1] ThreadCompleteListener | listeners of finished thread |
| threadId : [1..1] integer | id of current thread |

Table 38 Attributes of Class "NotifyingThread"

# Package "Utils"

Stereotypes: Java Package

| Name | Summary |
| --- | --- |
| [ExtDataParser](#_21c88acb-2d6a-49ad-9a0f-7882ec793d92) | Parser for external testing data |
| [Helper](#_617915e1-75d1-45fd-88bc-f4bd08c36b60) | General helper methods |
| [DataVector](#_45afbcb8-7873-4f64-8b12-03950ccc7132) | Customed format for data vector |

Table 39 Owned Classes of Package "Utils"

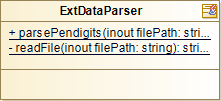
## Class "ExtDataParser"

from Package .[Utils](#_fdaca482-dd91-46f7-963e-a27c18d93d35)

Stereotypes: Java Class

Various parser for testing data in assets/external folder

Figure 13 ExtDataParser Class diagram



| Name | Description |
| --- | --- |
| DataVector parsePendigits (Inout filePath string,Inout type integer) | parse pendigits dataset |
| string readFile (Inout filePath string) | read dataset file given filePath |

Table 40 Operations of Class "ExtDataParser"

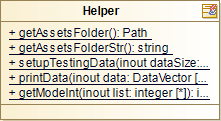
## Class "Helper"

from Package .[Utils](#_fdaca482-dd91-46f7-963e-a27c18d93d35)

Stereotypes: Java Class

contain utility methods for all classes

Figure 14 Helper Class diagram



| Name | Description |
| --- | --- |
| Path getAssetsFolder () | return Path object of asset folder |
| string getAssetsFolderStr () | return string of path of asset folder |
| DataVector setupTestingData (Inout dataSize integer,Inout attrSize integer,Inout classSize integer) | generate testing data with format compatible with DecisionTree |
| printData (Inout data DataVector) | print list of DataVector |
| integer getModeInt (Inout list integer) | return the mode of an integer list |

Table 41 Operations of Class "Helper"

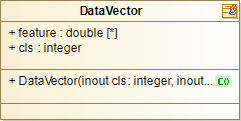
## Class "DataVector"

from Package .[Utils](#_fdaca482-dd91-46f7-963e-a27c18d93d35)

Stereotypes: Java Class

contain class and features of a data vector

Figure 15 DataVector Class diagram



| Name | Description |
| --- | --- |
| DataVector (Inout cls integer,Inout feature double) | constructor of DataVector |

Table 42 Operations of Class "DataVector"

| Name | Description |
| --- | --- |
| feature : [0..\*] double | features (attributes) of data vector |
| cls : [1..1] integer | class (label) of data vector |

Table 43 Attributes of Class "DataVector"