

Random Forest HOG

Random Forest

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

Version : 2.0 1-1

~~Status :Draft~~

Publication : 07/11/201422/01/2014

Copyright:

Modeliosoft

MINES Paristech

Table of Contents

1	Introduction	4
2	Dictionnaire	5
2.1	Dictionnaire de forêts decisionelles Aléatoires.....	5
2.2	Dictionnaire de L'histogramme de gradient orienté (HOG).....	7
3	Use Cases.....	8
3.1	Actors	8
3.2	Use Cases.....	8
3.2.1	Use Case "Avant projet Use Case diagram"	9
4	Package Index	10
5	Package "RandomForestHOG"	11
6	Package "HOG"	12
6.1	Class "HOGParam"	13
6.2	Class "HOG"	15
6.3	Class "HOGAppli"	18
7	Package "DecisionTree"	20
7.1	Class "DecisionTree"	21
7.2	Class "TreeNode"	24
8	Package "RandomForest"	26
8.1	Class "RandomForest"	27
8.2	Class "RandomForestLearner"	29
8.3	Class "MainRun"	32
9	Package "NotifyingThread"	33
9.1	Interface "ThreadCompleteListener"	34
9.2	Class "NotifyingThread"	35
10	Package "Utils"	55
10.1	Class "ExtDataParser"	56
10.2	Class "Helper"	57
10.3	Class "DataVector"	58
1	Introduction	3
2	Dictionnaire	4
2.1	Dictionnaire de forêts decisionelles Aléatoires.....	4
2.2	Dictionnaire de L'histogramme de gradient orienté (HOG).....	6
3	Use Cases.....	7
3.1	Actors	7
3.2	Use Cases.....	7
3.2.1	Use Case "Avant projet Use Case diagram"	8

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型, 英文 (美國)

格式化: 預設段落字型, 英文 (美國)

格式化：預設段落字型

格式化: 預設段落字型

格式化：預設段落字型

格式化：預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化：預設段落字型

格式化：預設段落字型

4	Package Index	9
5	Package "RandomForestHOG"	10
6	Package "HOG"	11
6.1	Class "HOGParam"	12
6.2	Class "HOG"	13
7	Package "DecisionTree"	16
7.1	Class "DecisionTree"	17
7.2	Class "TreeNode"	18
8	Package "RandomForest"	19
8.1	Class "RandomForest"	20
8.2	Class "RandomForestLearner"	23
9	Préparation de la Validation	26
10	Répartition des tâches + Échéancier	27

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型

格式化: 預設段落字型, 英文 (美國)

格式化: 預設段落字型, 英文 (美國)

格式化: 預設段落字型

格式化: 預設段落字型

1 Introduction

Les **forêts décisionnelles aléatoires** (de l'anglais « *Random decision forest* ») ont été formellement proposées en 2001 par Leo Breiman et Adèle Cutler. Elles font partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de « *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 utilisée en vision par ordinateur pour la classification d'images et la détection visuelle d'objets d'une catégorie donnée. La technique calcule des histogrammes locaux de l'orientation du 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), les Shape contexts 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.

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

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

功能變數代碼變更

2 Dictionnaire

2.1 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 <u>décisionnels</u>	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 $X = x_1, \dots, x_n$ with responses $Y = y_1$ through y_n , 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

Random Forest HOG

		de la classe RandomForest
Vote d'un arbre	chaque arbre vote pour une des classes	Méthode evaluate de la classe DecisionTree
bagging (bootstrap aggregating)	also known as " <u>bootstrap aggregating</u> ". <u>A machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression</u> . It also reduces <u>variance</u> and helps to avoid <u>overfitting</u>	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

2.2 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	<p>kernels filtering the color or intensity data of images to compute gradient value usually applied vertically and horizontally which have forms like: $[-1, 0, 1]$ and $[-1, 0, 1]^T$.</p> $G_h(x, y) = f(x+1, y) - f(x-1, y) \quad \forall x, y$ $G_v(x, y) = f(x, y+1) - f(x, y-1) \quad \forall x, y$	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

3 Use Cases

3.1 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

3.2 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 maximal de la forêt profondeur maximale des arbres de la forêt	choose the maximum depth of trees in the random forest
choisir nombre maximale des arbres de décision 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 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

格式化: 間距 套用後: 0 點, 行距: 單行間距, 段落中不分頁
格式化: 間距 套用後: 0 點, 行距: 單行間距, 與下段同頁, 段落中不分頁

Random Forest HOG

Use-Cases	Description
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

3.2.1 Use Case "Avant projet Use Case diagram"

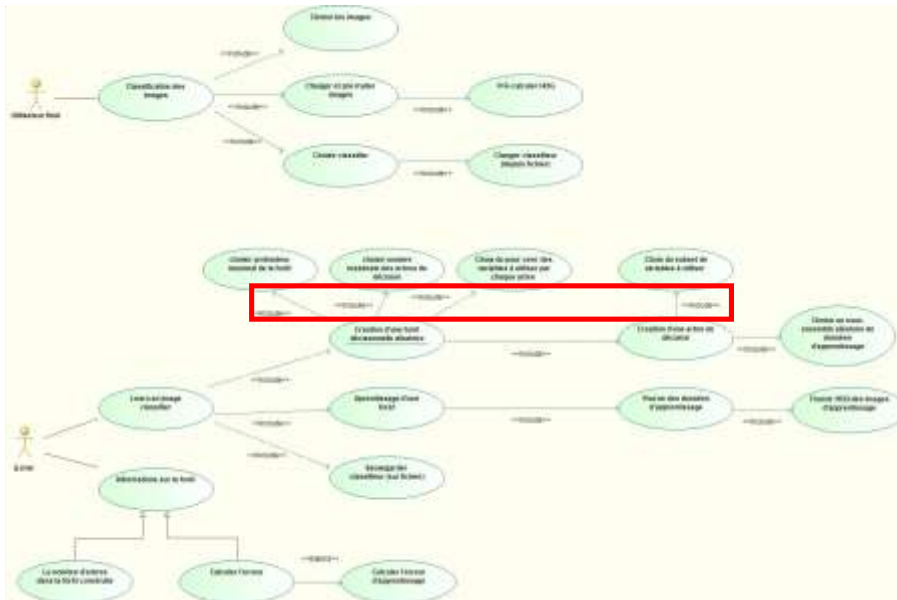


Figure 1 : Avant projet Use Case diagram

已註解 [fm1]: Plutôt des “include” que des “extend” dans le rectangle rouge

4 Package Index

[RandomForestHOG](#)

[HOG](#) [Package responsible for creating HOG representations of images](#)

[DecisionTree](#) [Package contains the classes used in decision tree](#)

[RandomForest](#) [Package contains class RandomForest and class RandomForestLearner.](#)

[NotifyingThread](#) [Implement listener for threading](#)

5 Package "RandomForestHOG"

from Package gforge
Stereotypes: Java Package

Name	Summary
HOG	Package responsible for creating HOG representations of images
DecisionTree	Package contains the classes used in decision tree
RandomForest	Package contains class RandomForest and class RandomForestLearner.
NotifyingThread	Implement listener for threading

Table 3 Owned Packages of Package "RandomForestHOG"

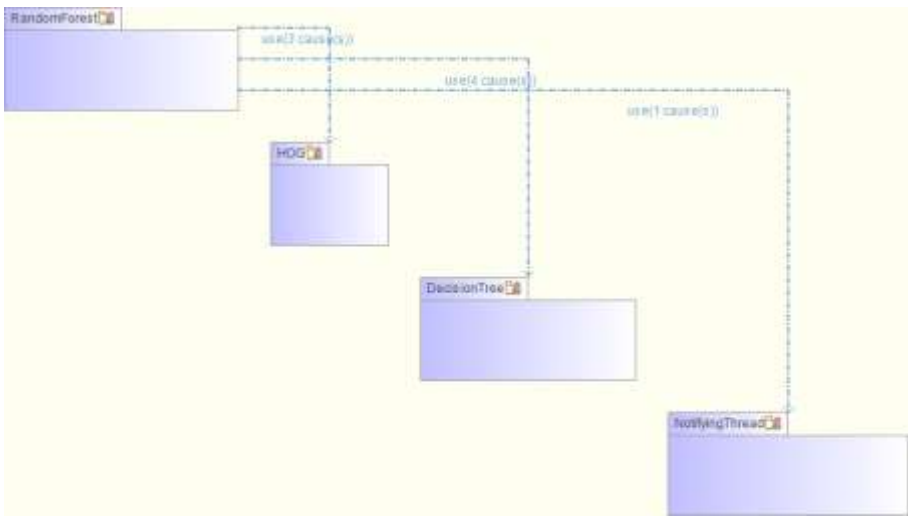


Figure 2 RandomForestHOG (subpackage structure autodiagram)

格式化: 内文

格式化: 内文

6 Package "HOG"

from Package gforge.RandomForestHOG

Stereotypes: Java Package

Name	Summary
<u>HOGParam</u>	<u>Class contains a set of parameters for HOG</u>
<u>HOG</u>	<u>Class responsible for creating HOG representations of images</u>
<u>HOGAppli</u>	<u>convert an image into HOG data set</u>

Table 4 Owned Classes of Package "HOG"

格式化: Texte

格式化: 内文

6.1 Class "HOGParam"

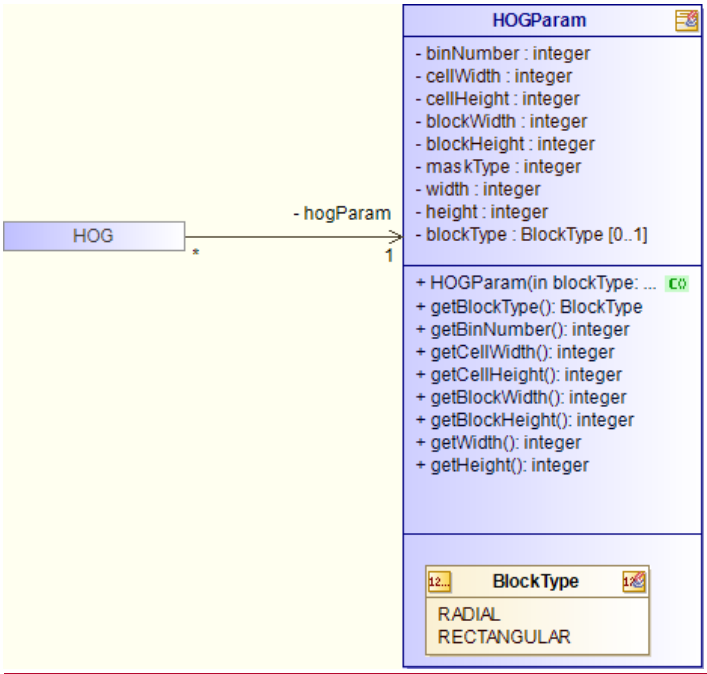
from Package gforge.RandomForestHOG.HOG

Stereotypes: Java Class

This class bundles the essential parameters to perform HOG.

← 格式化: 内文

Figure 3 HOGParam



Random Forest HOG

Name	Description
<u>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)</u>	<u>initialize instance of HOGParam with manual setting</u>
<u>BlockType getBlockType ()</u>	<u>return blockType variable</u>
<u>integer getBinNumber ()</u>	<u>return binNumber variable</u>
<u>integer getCellWidth ()</u>	<u>return cellWidth variable</u>
<u>integer getCellHeight ()</u>	<u>return cellHeight variable</u>
<u>integer getBlockWidth ()</u>	<u>return blockWidth variable</u>
<u>integer getBlockHeight ()</u>	<u>return blockHeight variable</u>
<u>integer getWidth ()</u>	<u>return width variable</u>
<u>integer getHeight ()</u>	<u>return height variable</u>

Table 5 Operations of Class "HOGParam"

格式化: 内文

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

Table 6 Attributes of Class "HOGParam"

格式化: 内文

Name	Values	Description
<u>BlockType</u>	<u>RADIAL</u> <u>RECTANGULAR</u>	<u>BlockType contains two type, RADIAL and RECTANGULAR</u>

Table 7 Owned Enumerations of Class "HOGParam"

格式化: 内文

6.2 Class "HOG"

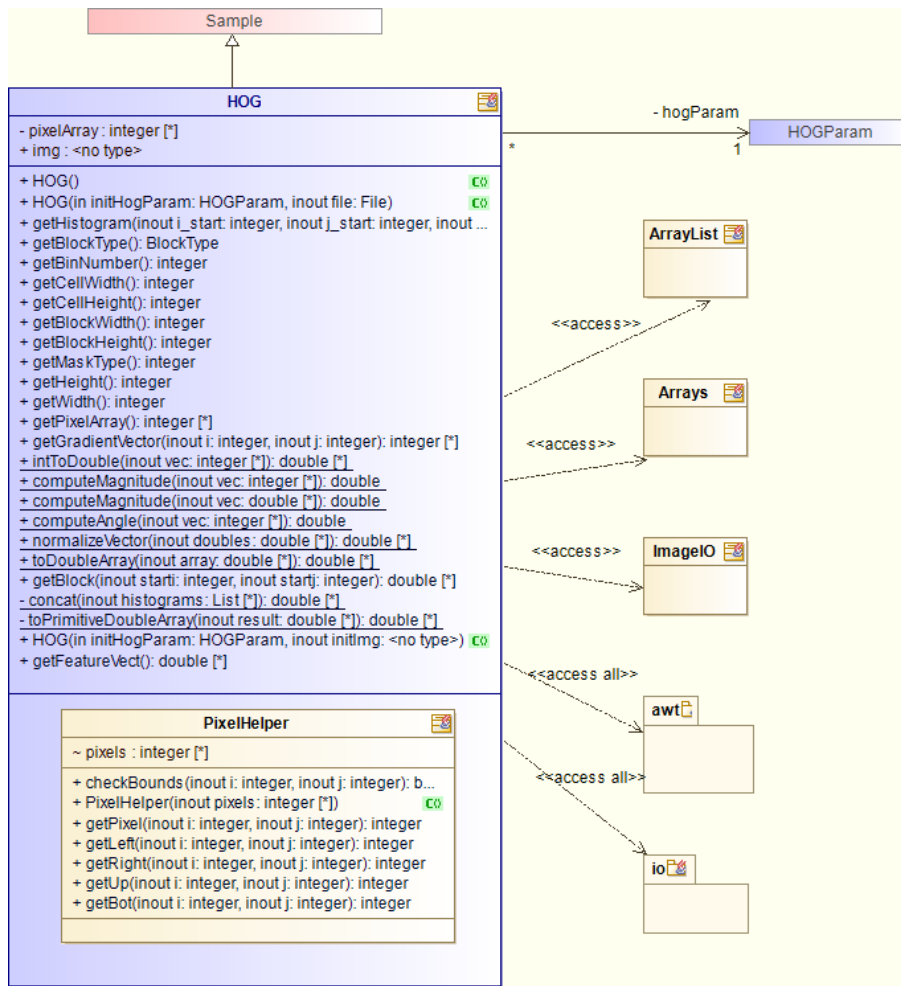
from Package gforge.RandomForestHOG.HOG

Inherits from: Sample

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

Table 8 Operations of Class "HOG"

← 格式化: 内文

Random Forest HOG

Name	Description
featureVector : [0..*] integer	array of feature vectors of each pixel after calculation
img : [1..1]	

Table 9 Attributes of Class "HOG"

← 格式化: 内文

Name	Description
->hogParam : [1..1] HOGParam	set of HOG parameters used in HOG computation

Table 10 Associations of Class "HOG"

← 格式化: 内文

Name	Summary
PixelHelper	

Table 11 Owned Classes of Class "HOG"

← 格式化: 内文

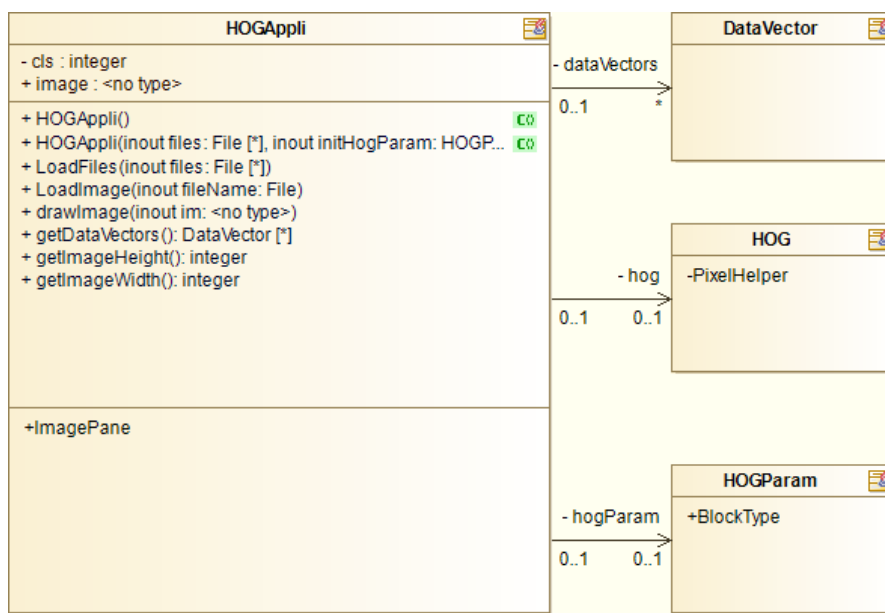
6.3 Class "HOGAppli"

from Package gforge.RandomForestHOG.HOG

Stereotypes: Java Class

load files, caculate the HOGs and generate data sets

Figure 5 HOGAppli Class diagram



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

Table 12 Operations of Class "HOGAppli"

Random Forest HOG

Name	Description
<u>cls : [1..1] integer</u>	<u>the class of the input image</u>
<u>image : [1..1]</u>	<u>input image</u>

Table 13 Attributes of Class "HOGAppli"

← 格式化: 内文

Name	Description
<u>->dataVectors : [0..*] DataVector</u>	<u>list of data from HOG</u>
<u>->hog : [0..1] HOG</u>	<u>the HOG instance</u>
<u>->hogParam : [0..1] HOGParam</u>	<u>the hogParam instance</u>

Table 14 Associations of Class "HOGAppli"

← 格式化: 内文

Name	Summary
<u>ImagePane</u>	<u>ImagePane extends JPanel</u>

Table 15 Owned Classes of Class "HOGAppli"

7 Package "DecisionTree"

from Package *gforge.RandomForestHOG*

Stereotypes: Java Package

Create decision tree and classify testing data

Name	Summary
<u>DecisionTree</u>	<u>Class contains interface for using Decision Trees.</u>
<u>TreeNode</u>	<u>Class defines the nodes of a tree data structure</u>

Table 16 Owned Classes of Package "DecisionTree"

格式化: 内文

格式化: 内文

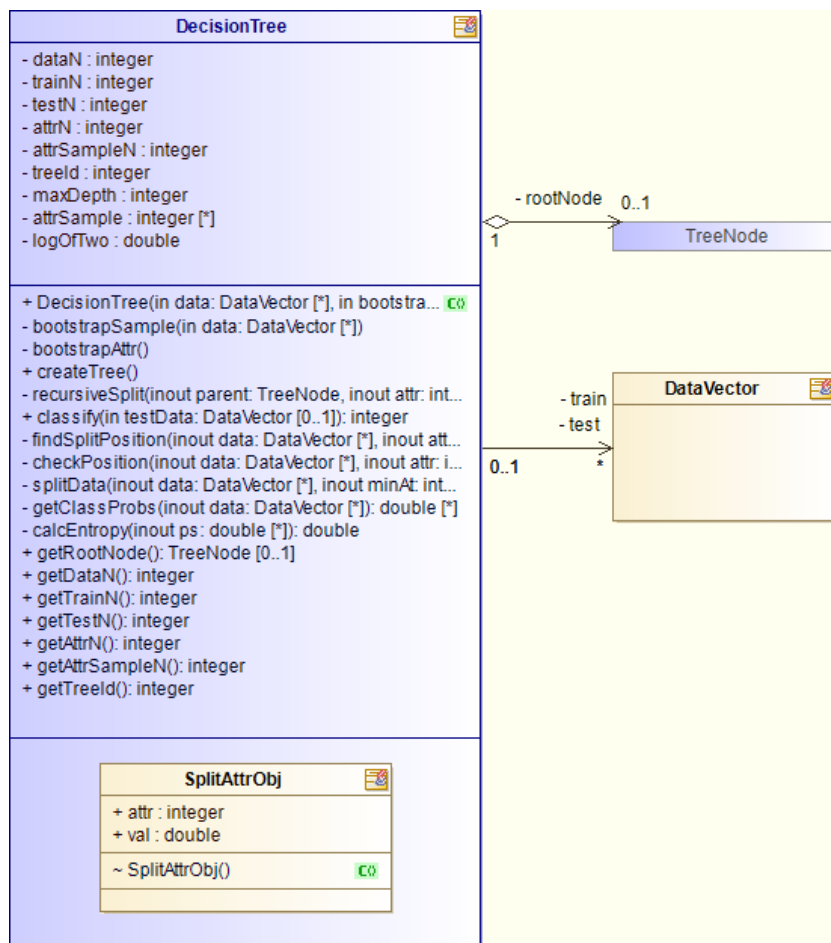
7.1 Class "DecisionTree"

from Package gforge.RandomForestHOG.DecisionTree

Stereotypes: Java Class

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

Figure 6 DecisionTree



Random Forest HOG

Name	Description
DecisionTree (In data DataVector, In bootstrapRate double, In attrSampleN integer, In maxDepth integer, In treeld integer)	constructs a decision tree from a data matrix.
bootStrapSample (In data DataVector)	create a bootstrap 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 getTreeld ()	return treeld

Table 17 Operations of Class "DecisionTree"

格式化: 内文

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
treeld : [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	root node of this decision tree
->train : [0..*] DataVector	training dataset
->test : [0..*] DataVector	testing dataset

Table 19 Associations of Class "DecisionTree"

← 格式化: 内文

Name	Summary
SplitAttrObj	Attribute index and value

Table 20 Owned Classes of Class "DecisionTree"

← 格式化: 内文

7.2 Class "TreeNode"

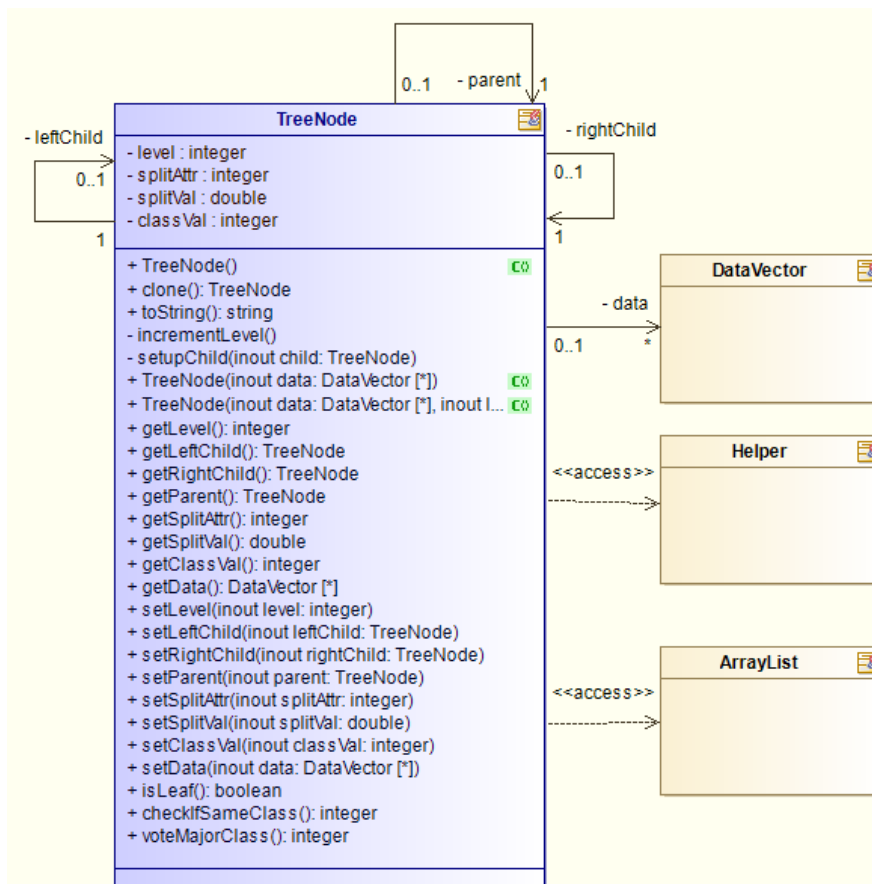
from Package gforge.RandomForestHOG.DecisionTree

Implements: Cloneable

Stereotypes: Java Class

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

Figure 7 *TreeNode*



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

Table 21 Operations of Class "TreeNode"

格式化: 内文

Name	Description
level : [1..1] integer	<u>current level in the tree</u>
splitAttr : [1..1] integer	<u>attribute to split on</u>
splitVal : [1..1] double	<u>value to split splitAttr</u>
classVal : [1..1] integer	

Table 22 Attributes of Class "TreeNode"

格式化: 内文

Name	Description
->leftChild : [0..1] TreeNode	<u>left child of this node</u>
->rightChild : [1..1] TreeNode	<u>right child of this node</u>
->parent : [1..1] TreeNode	<u>parent of this node</u>
->data : [0..*] DataVector	<u>data at current level</u>

Table 23 Associations of Class "TreeNode"

8 Package "RandomForest"

from Package gforge.RandomForestHOG

Stereotypes: Java Package

格式化: Texte

Name	Summary
<u>RandomForest</u>	Class define the classifier generated by random forest.
<u>RandomForestLearner</u>	Class responsible for random forest learning
<u>MainRun</u>	Main class

Table 24 Owned Classes of Package "RandomForest"

格式化: 内文

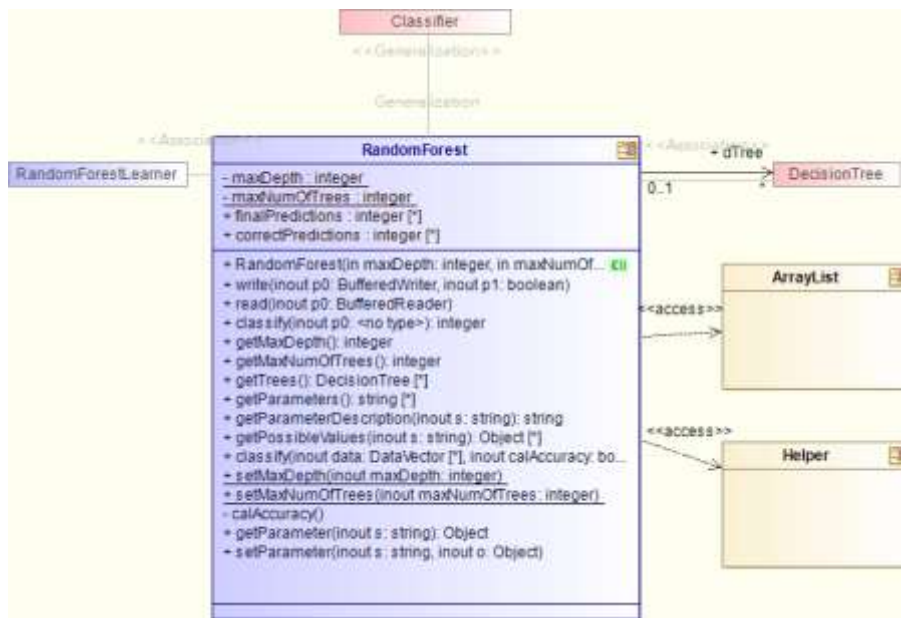
8.1 Class "RandomForest"

from Package gforge.RandomForestHOG.RandomForest

Stereotypes: Java Class

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

Figure 8 RandomForest



Random Forest HOG

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

Table 25 Operations of Class "RandomForest"

格式化: 内文

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

Table 26 Attributes of Class "RandomForest"

格式化: 内文

Name	Description
__->dTree : [0..*] DecisionTree	<u>trees in random forest</u>

Table 27 Associations of Class "RandomForest"

格式化: 内文

8.2 Class "RandomForestLearner"

from Package gforge.RandomForestHOG.RandomForest

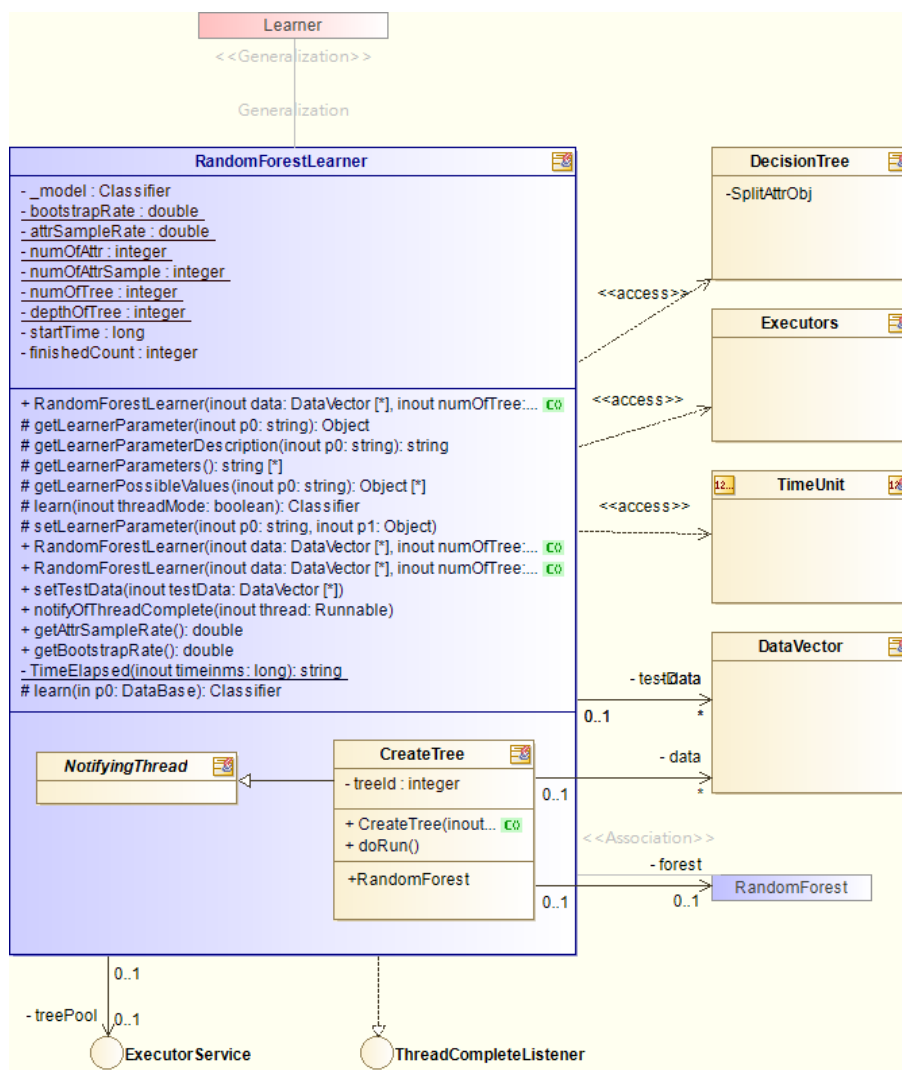
Implements: ThreadCompleteListener

Stereotypes: Java Class

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

格式化: 内文

Figure 9 RandomForestLearner



Random Forest HOG

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

Table 28 Operations of Class "RandomForestLearner"

格式化: 内文

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

Table 29 Attributes of Class "RandomForestLearner"

格式化: 内文

Random Forest HOG

Name	Description
->data : [0..*] DataVector	training dataset
->testData : [0..*] DataVector	testing dataset
->treePool : [0..1] ExecutorService	thread pool for learning

Table 30 Associations of Class "RandomForestLearner"

← 格式化: 内文

← 格式化: (中文) 中文 (台灣)

Name	Summary
CreateTree	Create decision tree
NotifyingThread	Implement listener for threading
DataVector	

Table 31 Owned Classes of Class "RandomForestLearner"

← 格式化: 内文

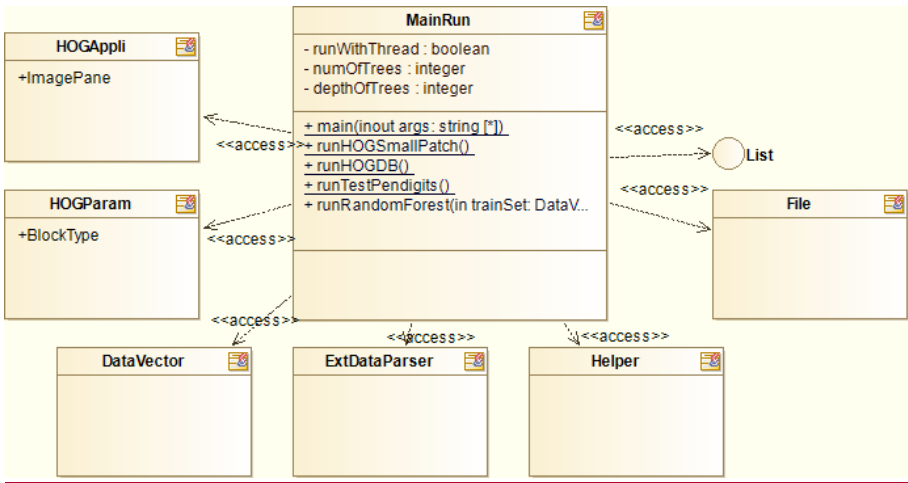
8.3 Class "MainRun"

from Package gforge.RandomForestHOG.RandomForest
Stereotypes: Java Class

main running class with general testing

格式化: 内文

Figure 10 MainRun Class diagram



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

Table 32 Operations of Class "MainRun"

格式化: 内文

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

Table 33 Attributes of Class "MainRun"

9 Package "NotifyingThread"

from Package `gforge.RandomForestHOG`

Stereotypes: Java Package

contain abstract class and interface for setting up listener of threads

格式化: 内文

Name	Summary
<u>ThreadCompleteListener</u>	<u>Interface for thread listeners</u>

Table 34 Owned Interfaces of Package "NotifyingThread"

格式化: 内文

Name	Summary
<u>NotifyingThread</u>	<u>Implement listener for threading</u>

Table 35 Owned Classes of Package "NotifyingThread"

格式化: 内文

9.1 Interface "ThreadCompleteListener"

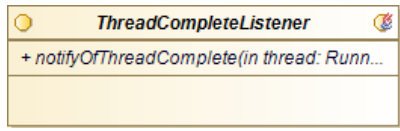
from Package gforge.RandomForestHOG.NotifyingThread

Stereotypes: Java Interface

Notify listeners when thread finished

← 格式化: 内文

Figure 11 ThreadCompleteListener Class diagram



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

Table 36 Operations of Interface "ThreadCompleteListener"

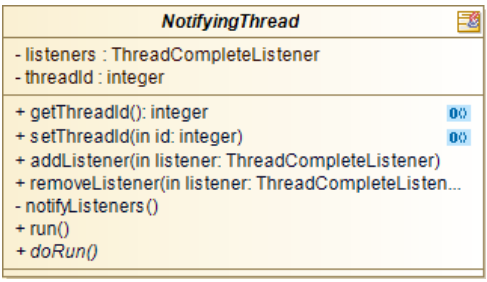
← 格式化: 内文

9.2 Class "NotifyingThread"

from Package *gforge.RandomForestHOG*.*NotifyingThread*
Stereotypes: Java Class

Implement Runnable as abstract class to set up listener of threads

Figure 12 NotifyingThread Class diagram



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

Table 37 Operations of Class "NotifyingThread"

格式化: 内文

Name	Description
<u>listeners : [1..1] ThreadCompleteListener</u>	<u>listeners of finished thread</u>
<u>threadId : [1..1] integer</u>	<u>id of current thread</u>

4 Table 38 Attributes of Class "NotifyingThread"Package Index

格式化: 標號, 不在段落前分頁

<u>RandomForestHOG</u>	<i>Package contains main packages for random forest classification on HOG</i>
<u>HOG</u>	<i>Package responsible for creating HOG representations of images</i>
<u>DecisionTree</u>	<i>Package contains the classes used in decision tree</i>

格式化: 標號

格式化: 標號

格式化: 標號

Random Forest HOG

RandomForest

~~Package contains class RandomForest and class RandomForestLearner.~~

格式化: 標號

格式化: 標號

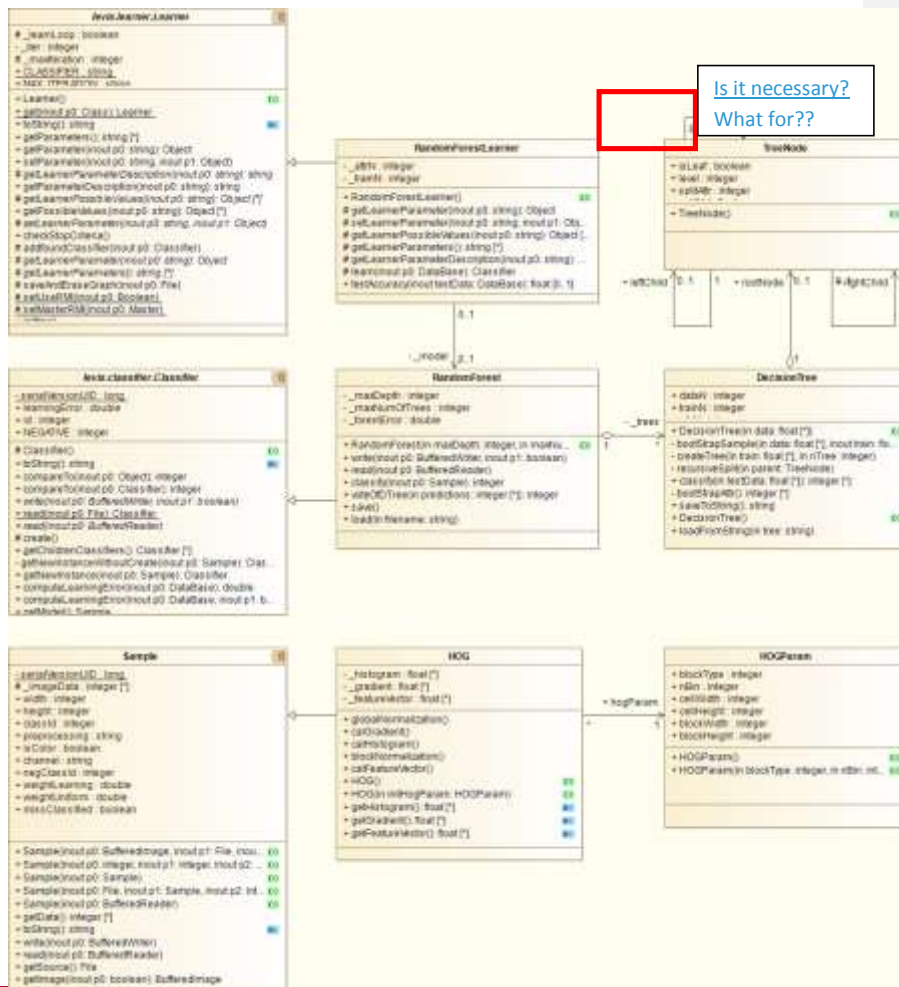
格式化: 標號, 不在段落前分頁

5

~~Package "RandomForestHOG"~~

~~6~~ ~~from Package gforge~~

~~7 Figure 122 Random Forest HOG Class diagram~~



8.

9

Random Forest HOG

10 Name	11 Summary
12 HOG	13 Package responsible for creating HOG representations of images
14 DecisionTree	15 Package contains the classes used in decision tree
16 RandomForest	17 Package contains class RandomForest and class RandomForestLearner.

~~18~~ ~~Table 3 Owned Packages of Package "RandomForestHOG"~~

~~19~~ ~~Package "HOG"~~

~~20~~ ~~from Package gforge.RandomForestHOG~~

~~21~~

22 Name	23 Summary
24 HOGParam	25 Class contains a set of parameters for HOG
26 HOG	27 Class responsible for creating HOG representations of images

~~28~~ ~~Table 4 Owned Classes of Package "HOG"~~

~~29~~

~~30~~

~~31~~

~~32~~

~~33~~

~~34~~

~~35~~

~~36~~

~~37~~

~~38~~

~~39~~

~~40~~

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

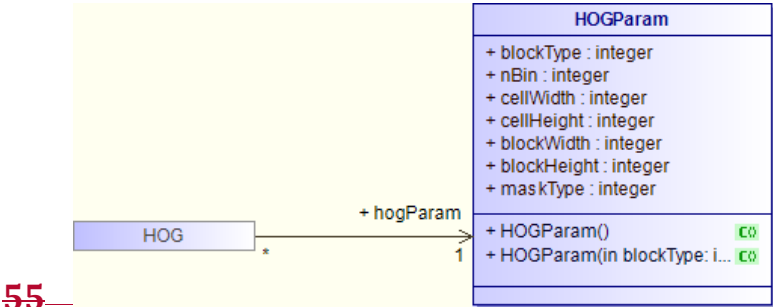
格式化: 標號, 不在段落前分頁

41
42
43
44
45
46
47
48
49

50 *Class "HOGParam"*
51 *from Package gforge.RandomForestHOG.HOG*
52 *This class bundles the essential parameters to perform HOG.*

53

54 *Figure 3 HOGParam*



56
57

Random Forest HOG

58—Name	59—Description
60—HOGParam {}	61—initialize instance of HOGParam with default setting
62—HOGParam (In blockType integer, In nBin integer, In cellWidth integer, In cellHeight integer, In blockWidth integer, In blockHeight integer, In maskType integer)	63—initialize instance of HOGParam with manual setting

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

65—

66—

67—Name	68—Description
69—blockType : {1..1} integer	70—the type of block (either rectangular or radial)
71—nBin : {1..1} integer	72—number of bins for creating histogram
73—cellWidth : {1..1} integer	74—width of each cell
75—cellHeight : {1..1} integer	76—height of each cell
77—blockWidth : {1..1} integer	78—width of each block
79—blockHeight : {1..1} integer	80—height of each block
81—maskType : {1..1} integer	82—type of the mask to calculate gradients

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 不在段落前分頁

84—

85—

86—Class "HOG"

87—from Package gforge.RandomForestHOG.HOG

88—Inherits from: Sample

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

90—Figure 4 HOG

~~92~~

~~93~~

~~94~~

~~95~~

~~96~~

~~97~~

~~98~~

~~99~~

~~100~~

~~101~~

~~102~~

~~103~~ *Interaction "HOG computation"*

~~104~~ *Figure 5 HOG computation*

Random Forest HOG

106

[illegible]

130

131-

[illegible]

141-

142-

格式化: 標號, 無, 不在段落前分頁

~~148~~ Package "DecisionTree"

149 ~~from Package gforge.RandomForestHOG~~

~~150~~

格式化: 標號, 無, 不在段落前分頁

151 <i>Name</i>	152 <i>Summary</i>
155 <i>TreeNode</i>	156 <i>Class defines the nodes of a tree data structure</i>

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

~~158~~

~~159~~

~~160~~

~~161~~

~~162~~

~~163~~

~~164~~

~~165~~

~~166~~

~~167~~

~~168~~

~~169~~

~~170~~

~~171~~

~~172~~

~~173~~

~~174~~

~~175~~

~~176~~

~~177~~

179 *Class "DecisionTree"***180** *from Package gforge.RandomForestHOG, DecisionTree***181** *This class implements operations about decision tree in random forest learning.***182** *Figure 6 DecisionTree***183**

184 <i>Name</i>	185 <i>Description</i>
186 <i>DecisionTree()</i>	187
188 <i>DecisionTree (In data float)</i>	189 <i>constructs a decision tree from a data matrix.</i>
190 <i>bootstrapSample (In data float, In out train float, In out test float)</i>	191 <i>create a bootstrap sample of size trainN while leaving others as test data create a bootstrap sample of size trainN</i>
192 <i>integer bootstrapAttr()</i>	193 <i>selects attributes of size attrN from sample</i>
194 <i>createTree (In train float, In nTree integer)</i>	195 <i>creates the decision tree according to the specifications of random forest trees</i>
196 <i>recursiveSplit (In parent TreeNode)</i>	197 <i>critical function to create the decision tree with the selected attributes of bootstrap sample</i>
198 <i>integer classify (In testData float)</i>	199 <i>traverses the tree and returns the prediction of the given test data</i>
200 <i>string saveToString()</i>	201 <i>traverses through tree nodes and returns the string representation of current decision tree</i>
202 <i>loadFromString (In tree string)</i>	203 <i>reconstruct this decision tree object using the string previously generated by toString() reconstruct this decision tree object using the string previously generated by toString()</i>
205 <i>Name</i>	206 <i>Description</i>
207 <i>dataN : [1, 1] integer</i>	208 <i>total size of the data (training and testing)</i>

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化

格式化

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

已註解 [fm4]: Where and how is stored the result of bootstrapping the trainingSet ?? Should there be sth like an array of used_sample_IDs ?

已註解 [fm4]: Where and how is stored the result of bootstrapping the trainingSet ?? Should there be sth like an array of used_sample_IDs ?

已註解 [fm3]: Unclear what exactly are these parameters...

已註解 [fm3]: Unclear what exactly are these parameters...

已註解 [fm5]: Unclear WHAT exactly is returned by bootstrapAttr()... And is it stored somewhere in the instance ? More important and even less clear: how do you deal with bootstrapping of examples and random selection of variables ? D

已註解 [fm5]: Unclear WHAT exactly is returned by

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化

格式化

格式化

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 醒目提示

格式化: 醒目提示

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

已註解 [fm6]: Really needed ? Can be recovered from Dataset

已註解 [fm6]: Really needed ? Can be recovered from Dataset

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化

格式化

格式化

格式化

Random Forest HOG

205 Name	206 Description
209 trainN : [1..1] integer	210 size of the bootstrap samples to train (assigned by RFlearner)
211 attrN : [1..1] integer	212 size of the attributes to train (assigned by RFlearner)
214 Name	215 Description
216 →rootNode : [0..1] TreeNode	217 root node of this decision tree

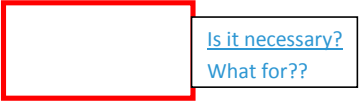
~~219~~ Class "TreeNode"

~~220~~ from Package gforge.RandomForestHOG.DecisionTree

~~221~~ This class contains the elements of a node in tree data structure.

~~222~~

~~223~~ Figure 7 TreeNode



~~224~~

225 Name	226 Description
227 TreeNode()	228 constructs the tree node

~~230~~

~~231~~

232 Name	233 Description
234 isLeaf : [1..1] boolean	235 indicates if the node is a leaf
236 level : [1..1] integer	237 current level in the tree
238 splitAttr : [1..1] integer	239 attribute to split on
240 splitVal : [1..1] float	241 value to split splitAttr

已註解 [fm6]: Really needed ? Can be recovered from Dataset inside recursiveSplit() method [?]

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

已註解 [fm7]: In case it is a leaf, you probably need at least one attribute to store either 1 associated classID, or associated probabilities for each class

格式化: 標號, 無, 不在段落前分頁

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 醒目提示

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

244-

245-Name	246-Description
247- <u>→leftChild : [0..1]TreeNode</u>	248- <u>left child of this node</u>
249- <u>→rightChild : [1..1]TreeNode</u>	250- <u>right child of this node</u>
251- <u>→parent : [1..1]TreeNode</u>	252- <u>parent of this node</u>

254-Package "RandomForest"

255-from Package gforge.RandomForestHOG

256-

257-Name	258-Summary
259- <u>RandomForest</u>	260- <u>Class define the classifier generated by random forest.</u>
261- <u>RandomForestLearner</u>	262- <u>Class responsible for random forest learning</u>

264-

265-

266-

267-

268-

269-

270-

271-

272-

273-

274-

275-

276-

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

已註解 [fm8]: Is this association really necessary ? What for ??

已註解 [fm8]: Is this association really necessary ? What for ??

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 標號, 無, 不在段落前分頁

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 標號, 無, 不在段落前分頁

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 不加底線, 字型色彩: 自動, 醒目提示

格式化: 不加底線, 字型色彩: 自動, 醒目提示

格式化: 醒目提示

格式化: 醒目提示

格式化: 不加底線, 字型色彩: 自動, 醒目提示

格式化: 不加底線, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 字型: 11 點, 字型色彩: 自動, 醒目提示

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

~~278~~

~~279~~

~~280~~

~~281~~

~~282~~

~~283~~

~~284~~

~~285~~ *Class "RandomForest"*

~~286~~ *from Package gforge.RandomForestHOG.RandomForest*

~~287~~ *Inherits from: Classifier*

~~288~~ *This class contains information about the classifier trained with random forest and provides operation to classify images.*

~~289~~ *Figure 8 RandomForest*

~~290~~

~~291~~ *Interaction "RandomForest-load"*

~~292~~ *Figure 9 load*

~~294~~ ~~Interaction "RandomForest save"~~

~~295~~ ~~Figure 10 save~~

~~296~~

~~297~~

~~298~~ ~~Interaction "RandomForest classify"~~

~~299~~ ~~Figure 11 Interaction Sequence diagram~~

~~300~~

343-

344-

345-

346-

347- *Class "RandomForestLearner"*

348- *from Package gforge.RandomForestHOG.RandomForest*

349- *Inherits from: Learner*

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

351-

352- *Figure 12 RandomForestLearner*

353-

354-

355-

356-

358-

359-

360-

361-

362-

363-

364-Interaction "RandomForestLearner.learn"

365-Figure 13 learn

Random Forest HOG

367 Name	368 Description
369 RandomForestLearner ()	370 The constructor of RandomForestLearner
371 Object getLearnerParameter (Inout p0 string)	372 implements getLearnerParameter method of Learner
373 string getLearnerParameterDescription (Inout p0 string)	374 implements getLearnerParameterDescription method of Learner
375 string getLearnerParameters ()	376 implements getLearnerParameters method of Learner
377 Object getLearnerPossibleValues (Inout p0 string)	378 implements getLearnerPossibleValues method of Learner
379 Classifier learn (Inout p0 DataBase)	380 implements learn method of Learner and trains a random forest classifier
381 setLearnerParameter (Inout p0 string, Inout p1 Object)	382 implements setLearnerParameter method of Learner
383 float testAccuracy (Inout testData DataBase)	384 tests the accuracy of random forest after forest creation

~~386~~

~~387~~

388 Name	389 Description
390 _attrN : [1..1] integer	391 size of attributes to bootstrap
392 _trainN : [1..1] integer	393 size of data to create bootstrap sample

~~395~~

~~396~~

397 Name	398 Description
399 -> _model : [0..1] RandomForest	400 the random forest classifier model for learner to train

已註解 [fm13]: It seems a bit strange that RandomForest, supposed to be only a Classifier, not a Learner, contains methods like createForest(dataSet), which is related to LEARNING. It would probably be wiser that RandomForest(...) constructor would create the treeNum DecisionTrees (all initially empty) of the forest , and then learn(dataSet) in RandomForestLearner would call, DIRECTLY on every tree, bootStrapSample(), then bootStrapAttrib(), and finally recursiveSplit(float[][] randomizedTrainingData, rootNode, maxDepth)

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化: 標號, 無, 不在段落前分頁

格式化表格

格式化: 標號, 無, 不在段落前分頁

10Package "Utils"

from Package gforge
Stereotypes: Java Package

格式化: Texte

Name	Summary
<u>ExtDataParser</u>	<u>Parser for external testing data</u>
<u>Helper</u>	<u>General helper methods</u>
<u>DataVector</u>	<u>Customed format for data vector</u>

Table 37 Owned Classes of Package "Utils"

格式化: 内文

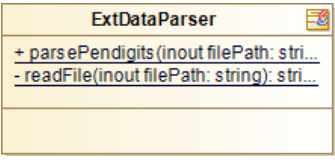
10.1 Class "ExtDataParser"

from Package gforge.Utils
Stereotypes: Java Class

Various parser for testing data in assets/external folder

格式化: 内文

Figure 17 ExtDataParser Class diagram



Name	Description
<u>DataVector parsePendigits (Inout filePath string,Inout type integer)</u>	<u>parse pendigits dataset</u>
<u>string readFile (Inout filePath string)</u>	<u>read dataset file given filePath</u>

Table 38 Operations of Class "ExtDataParser"

格式化: 内文

10.2 Class "Helper"

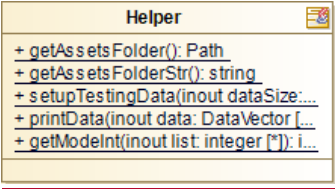
from Package gforge.Utils

Stereotypes: Java Class

contain utility methods for all classes

格式化: 内文

Figure 18 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 getModelInt (Inout list integer)	return the mode of an integer list

Table 39 Operations of Class "Helper"

格式化: 内文

10.3 Class "DataVector"

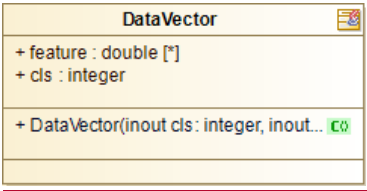
from Package gforge.Utils

Stereotypes: Java Class

contain class and features of a data vector

格式化: 内文

Figure 19 DataVector Class diagram



Name	Description
<u>DataVector (Inout cls integer,Inout feature double)</u>	<u>constructor of DataVector</u>

Table 40 Operations of Class "DataVector"

Name	Description
<u>feature : [0..*] double</u>	<u>features (attributes) of data vector</u>
<u>cls : [1..1] integer</u>	<u>class (label) of data vector</u>

Table 41 Attributes of Class "DataVector"

401- Table 23 Associations of Class "RandomForestLearner"

格式化: 標號, 不在段落前分頁

402-

403-

404-

405-

406-

407-

~~408—~~

~~409—~~

~~410—~~

~~411—Préparation de la Validation~~

~~412—~~

~~413—Using framework JUnit as validation tool.~~

~~414—Create basic test cases for methods and attributes value.~~

~~415—~~

~~416—Ex: For TreeNode we test if the parent.leftNode, parent.rightNode are not null, or if parent.isLeaf is false when this.isLeaf is true. For DecisionTree we test if saveToString() returns the correct string representation of simple trees.~~

~~417—~~

~~418—Create boundary test cases methods and attributes value.~~

~~419—~~

~~420—Create simple training data for DecisionTree and see if it creates the tree with best performance.~~

~~421—~~

~~422—4. Create simple training data for RandomForestLearner and see if the forest matches the pre-calculated forest.~~

~~423—~~

~~424—~~

~~425—~~

~~426—~~

~~427—~~

~~428—~~

~~429—~~

~~430—~~

~~431—~~

~~432—~~

~~433—~~

434 ~~Répartition des tâches + Échéancier~~

~~435—Michał Pomarański : testing program, random forest, test cases~~

~~436—Po-Tsung CHIU : decision tree, tree node, random forest, test cases~~

~~437—Chun-Jen PENG : HOG, random forest, test cases~~

Random Forest HOG



438

