Object Class and Concept Recognition for Content-Based Image Retrieval: the Software Package

Yi Li

Computer Science & Engineering

University of Washington

yi@cs.washington.edu

November 16, 2004

About this Documentation

This documentation describes how to set up and execute the software package I designed and implemented for my Ph.D. research, object and concept recognition for Content-based Image Retrieval.

Release Notes

Please note the latest updates of this documentation:

- **Edition 0.1** The first version has the basic setup and a step-by-step example.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 System Requirements

The system is implemented by JAVA. Most code design uses java JDK 1.4.1, but it is expected that the code works with any version of JDK 1.5.x and 1.4.x. However, there is no guarantee that the code works with any other version except 1.4.1.

Enable "`assert`" keyword if you use JDK 1.4.x. If use "JBuilder", find this option under "Project" → "Project Properties..." → "Build" → "Java", and then "Enable assert keyword", as shown in Figure 1.1

1.2 System Setup

1.2.1 Path Configuration

- config.properties

  In script file of config.properties, define pathes of

  IMAGE_PATH the directory to store images

  FEATURE_PATH the directory to store the generated fetures for images

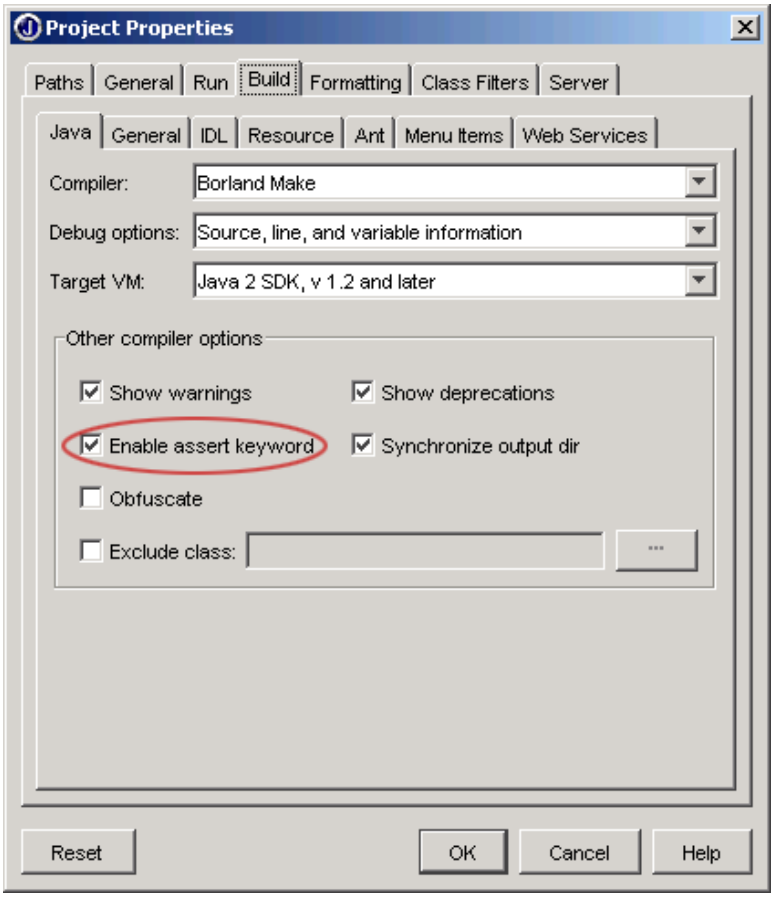

Figure 1.1: Enable "`assert`" keyword in JBuilder

REPRESENTATION_PATH the directory to store the feature representation for images

LEARNING_PATH the directory to store the learning results.

Other java class parameters can also be defined in config.properties, which will be detailed later.

- CLASSPATH A script file `setup.bat` shows an example to setup `CLASSPATH`. You need change `PROJECTPATH`, `PROJECTNAME`, `PACKAGEPATH`, and JBUILDERPATH to the correct locations. `"JBuilder"` is assumed to be used here.

Chapter 2

Step by Step Example

2.1 Image Descriptions GUI

This application provides utilities to construct or modify descriptions or labels for images. A standard `.desc` file can be generated as a result.

Usage

```
java yi.iu.imgdb.desc.gui.ImageDescApplication
```

2.2 Image Feature Generation

This section describes how to generate image features

2.2.1 Color Segmentations

Generate Color Segmentations

Indivisual Process

Usage

`java yi.iu.imgdb.learn.FeatureUtil$Generate -i` *image_name* `-f cs`

Example

```
java yi.iu.imgdb.learn.FeatureUtil$Generate -i
groundtruth/arborgreens/arborgreens01.jpg -f cs
```

Output

- `rst/ft/groundtruth/arborgreens/`
  `arborgreens01.jpg.s256.r128.KMeansHSISeg.CIELabLabeledRgnFts.ser`

  segmentation result written in java object format

- `rst/ft/groundtruth/arborgreens/`
  `arborgreens01.jpg.s256.r128.KMeansHSISeg.CIELabLabeledRgnFts.cdemo.gif`

  segmentation result shown in mean colors

- `rst/ft/groundtruth/arborgreens/`
  `arborgreens01.jpg.s256.r128.KMeansHSISeg.CIELabLabeledRgnFts.gdemo.gif`

  segmentation result shown in pseudo-colors

Batch Process

Usage

`perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Generate` *set_name.list* `-f cs`
`-h` *set_name*

Example

```
perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Generate t.list -f cs -h t
```

Output

- 3 outputs for each indivisual image as described above in "Indivisual Process" section
- `rst/ft/t.KMeansHSISeg.CIELabLabeledRgnFts.html`

  A html file shows the running results.

2.2.2 Texture Segmentations

Generate Texture Segmentations

Indivisual Process

Usage

`java yi.iu.imgdb.learn.FeatureUtil$Generate -i` *image_name* `-f ts`

Example

```
java yi.iu.imgdb.learn.FeatureUtil$Generate -i
groundtruth/arborgreens/arborgreens01.jpg -f ts
```

Output

- `rst/ft/groundtruth/arborgreens/`

  `arborgreens01.jpg.s256.r128.ColorGuidedTextureSegNCC.`

  `TextureLabeledRgnFts.ser`

  segmentation result written in java object format

- `rst/ft/groundtruth/arborgreens/`

`arborgreens01.jpg.s256.r128.ColorGuidedTextureSegNCC.`

`TextureLabeledRgnFts.gdemo.gif`

segmentation result shown in pseudo-colors

Batch Process

Usage

`perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Generate` *set_name.list* `-f ts -h` *set_name*

Example

```
perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Generate t.list -f ts -h t
```

Output

- 2 outputs for each indivisual image as described above in "Indivisual Process" section
- `rst/ft/t.ColorGuidedTextureSegNCC.TextureLabeledRgnFts.html`

  A html file shows the running results.

2.2.3 Structure Features

generate Structure features (line clusters)

Indivisual Process

Usage

`java yi.iu.imgdb.learn.FeatureUtil$Generate -i` *image_name* `-f st`

Example

```
java yi.iu.imgdb.learn.FeatureUtil$Generate -i
groundtruth/campusinfall/campusinfall14.jpg -f st
```

Output

- `rst/ft/groundtruth/campusinfall/`
  `campusinfall14.jpg.LineStructGenerator.LineClusterSets.ser`

  line clustering result written in java object format

- `rst/ft/groundtruth/campusinfall/`
  `campusinfall14.jpg.LineStructGenerator.LineClusterSets.cdemo.gif`

  line clusters shown in different colors

- `rst/ft/groundtruth/campusinfall/`
  `campusinfall14.jpg.LineStructGenerator.LineClusterSets.gdemo.gif`

  line clusters shown in different colors and enclosed by polygons

Batch Process

Usage

`perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Generate` *set_name.list* `-f st -h` *set_name*

Example

```
perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Generate t.list -f st -h t
```

Output

- 3 outputs for each indivisual image as described above in "Indivisual Process" section

- `rst/ft/t.LineStructGenerator.LineClusterSets.html`

  A html file shows the running results.

2.3 Image Feature Representation

This section describes how to represent image features

2.3.1 Color Segmentation Representation

Represent Color Segmentation features

Indivisual Process

Usage

`java yi.iu.imgdb.learn.FeatureUtil$Represent -i` *image_name* `-f cs`

Example

```
java yi.iu.imgdb.learn.FeatureUtil$Represent -i
groundtruth/arborgreens/arborgreens01.jpg -f cs
```

Output

- `rst/represent/groundtruth/arborgreens/`
  `arborgreens01.jpg.s256.r128.KMeansHSISeg.FeatureRepresentationList.m0.ser`

  feature representation result written in java object format

Batch Process

Usage

`perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Represent` *set_name.list* `-f cs`
`-h` *set_name*

Example

```
perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Represent t.list -f cs -h t
```

Output

- 1 output for each indivisual image as described above in "Indivisual Process" section
- `rst/represent/t.KMeansHSISeg.FeatureRepresentationList.html`

  A html file having links to the original images

2.3.2 Texture Segmentation Representation

Represent Texture Segmentation features

Indivisual Process

Usage

`java yi.iu.imgdb.learn.FeatureUtil$Represent -i` *image_name* `-f ts`

Example

```
java yi.iu.imgdb.learn.FeatureUtil$Represent -i
groundtruth/arborgreens/arborgreens01.jpg -f ts
```

Output

- `rst/represent/groundtruth/arborgreens/`
  `arborgreens01.jpg.s256.r128.ColorGuidedTextureSegNCC.FeatureRepresentationList.m0.ser`

  feature representation result written in java object format

Batch Process

Usage

`perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Represent` *set_name.list* `-f ts -h` *set_name*

Example

```
perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Represent t.list -f ts -h t
```

Output

- 1 output for each indivisual image as described above in "Indivisual Process" section
- `rst/represent/t.ColorGuidedTextureSegNCC.FeatureRepresentationList.html`

  A html file having links to the original images

2.3.3 Structure Representation

Represent Structure features

Indivisual Process

Usage

`java yi.iu.imgdb.learn.FeatureUtil$Represent -i` *image_name* `-f st`

Example

```
java yi.iu.imgdb.learn.FeatureUtil$Represent -i
groundtruth/arborgreens/arborgreens01.jpg -f st
```

Output

- `rst/represent/groundtruth/arborgreens/`
  `arborgreens01.jpg.LineStructGenerator.FeatureRepresentationList.m4.ser`
  feature representation result written in java object format

Batch Process

Usage

`perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Represent` *set_name.list* `-f st -h` *set_name*

Example

```
perl bp.pl yi.iu.imgdb.learn.FeatureUtil$Represent t.list -f st -h t
```

Output

- 1 output for each indivisual image as described above in "Indivisual Process" section
- `rst/represent/t.LineStructGenerator.FeatureRepresentationList.html`
  A html file having links to the original images

2.4 Image Data Center Construction

This section describes how to construct image data center, which stores the image feature vectors and image labels.

Usage

`java -Xms64m -Xmx512m yi.iu.imgdb.learn.ImgDataCenter -d` *set_name.desc* `-r` *represent_desc* `[-o` *output.idc*`]`

Generate image data center on image data set described by *set_name.desc* and with the feature representations specified by *represent_desc*. Save the constructed image data center in java object format with the name of *output.idc*, if specified.

Example

```
java -Xms64m -Xmx512m yi.iu.imgdb.learn.ImgDataCenter -d t.desc -r cs
```

Generate image data center on image data set described by t.desc and with the color segmentation feature representations.

Output

- `rst/represent/t.KMeansHSISeg.FeatureRepresentationList.idc`

  constructed image data center in java object format

Example

```
java -Xms64m -Xmx512m yi.iu.imgdb.learn.ImgDataCenter -d t.desc -r cs;ts;st
```

Generate image data center on image data set described by t.desc and with the color segmentation feature representations, texture segmentation feature representations, and structure feature representations.

Output

- `rst/represent/t.KMeansHSISeg+ColorGuidedTextureSegNCC+LineStructGenerator.FeatureRepresentationList.idc`

  constructed image data center in java object format

Example

```
java -Xms64m -Xmx512m yi.iu.imgdb.learn.ImgDataCenter -d t.desc -r cs;ts;st
-o t.cs+ts+st.idc
```

Generate image data center on image data set described by t.desc and with the color segmentation feature representations, texture segmentation feature representations, and structure feature representations.

Output

- `rst/represent/t.cs+ts+st.idc`

  constructed image data center in java object format with the specified name.

2.5 Learning Experiment

This section describes how to perform learning experiments.

Usage

`java -Xms64m -Xmx512m yi.iu.imgdb.learn.StatModelFoldExperiment -d` *data_center.idc* `-a` *learn_adapter* `[-n` *n_fold*`] [-o` *obj_list*`] [-r` *rst_output˙file*`] [-e` *output_experiment*`]`

Perform learning experiment using the learning method specified *learn_adapter* on the image data center specified by *data_center.idc*. The experiment uses cross-validation of *n_fold*. If *n_fold* is not specified, the experiment will be performed on 5-folds and only one training-testing around will be excuted. In other words, four of 5-folds will be used as training set and the other fold will be used as testing set. Use *obj_list* to perform experiments on specified objects. You can also specify the name to output the experiment results and the name to output the whole experiment, including training sets, testing sets and trained models of different arounds.

Example

```
java -Xms64m -Xmx512m yi.iu.imgdb.learn.StatModelFoldExperiment
-d sd.ct1.mm.bg.idc -a yi.iu.imgdb.learn.em.GMMAdapter -n 5
```

Perform learning experiments on the image data set used in my icpr04 paper [1] with 10 general object classes, using EM variant approach. See section 4.1 of my cvpr05 paper [2] submission for details.

Output

- `rst/learn/sd.ct1.mm.bg.GMMAdapter.png`

  ROC chart

- `rst/learn/sd.ct1.mm.bg.GMMAdapter.rst`

  Testing results saved in java object format

Example

```
java -Xms64m -Xmx512m yi.iu.imgdb.learn.StatModelFoldExperiment
-d gt.51.ts+cs+st.idc -a yi.iu.imgdb.learn.GMMNNAdapter -o cannon_beach -n 5
```

Perform learning experiments on the image data set of groundtruth database, using the Generative/Discriminative approach. Only learn the object of *canon beach* See section 4.4 of my cvpr05 paper [2] submission for details.

- `rst/learn/gt.51.ts+cs+st.GMMNNAdapter.cannon_beach.png`

  ROC chart

- `rst/learn/gt.51.ts+cs+st.GMMNNAdapter.cannon_beach.rst`

  Testing results saved in java object format

Bibliography

[1] Y. Li, J. Bilmes, and L. G. Shapiro. Object class recognition using images of abstract regions. In *Proceedings of the International Conference on Pattern Recognition*. IEEE Computer Society, 2004.

[2] Y. Li, J. Bilmes, and L. G. Shapiro. A generative/discriminative learning algorithm for object recognition in content-based image retrieval. In *CVPR submission*, 2005.