HIC: An image classification system based on supervised machine learning

Henning Christiansen
Research Group PLIS: Programming, Logic and Intelligent Systems
Roskilde University, Denmark
http://www.ruc.dk/henning/, henning@ruc.dk

Version 1.2.0, December 2016, and 1.2.1, February 2022

Version 1.2.1 is adapted from 1.2.0 by removing use of camera and the "What am I?" command, and is the one to use with newer Macs due to their non-flexible security facilities. Both version are (believed to be) valid with Windows, but 1.2.0 is more fun to use

1 Why HIC?

Giving a first year course in Artificial Intelligence, compressed in time to cover only 5 ECTS points, is quite a challenge, especially when the students are not assumed to have any programming experience in advance. To cope with this, the teacher has to select subjects and examples very carefully, so that the area is covered somehow and there is time to go in depth with the selected material to obtain some hands-on experience.

Machine learning is an important part of Artificial Intelligence, and HIC is an interactive system for image classification based on machine learning, developed especially to be used in this course given at Roskilde's Humanities-Technology Bachelor studies. It uses a supervised learning method developed by Marée et al [2], which is unusual in the sense that it is does not identify specific features of the data, and is highly generic in that sense. Instead it contains a lot of randomness and magically, by statistical regularities of the world (the law of big numbers is one such), it can actually produce reasonable results when used for the right problems.

The learning method produces ensembles of decision trees, called extra trees, short for extremely randomized trees (described in detail by [1]). Another good reason for using this method and an interactive system based on it, is that the central reference [2] is written in a way so it can be read by students without previous background in machine learning and programming; the only preparation for it in the course is an introduction to and exercises with elementary decision trees (see, e.g., [3, section 18.3]).

After the students have installed the HIC system and tested it for the sample images provided with the system, they are given an exercise in which they themselves provide the images and plan the training and testing phases. They are put into small groups of 3–4, and they are asked to produce series of photos of each other, experimenting with different parameter settings and ways of staging the photos to see what works best. This leads to discussions at a more general level about machine learning, supervised vs. unsupervised, and possible improvements of the results using more specific feature-based methods, as is standard for, e.g., face recognition in practice.

2 A little background

HIC has been written in Processing by the author in a quick and dirty way, so a few bugs may be expected here and there, and the source code is by no means intended as a showpiece in aesthetically and nicely structured programs. It is inspired by a much more advanced system PiXiT, developed by the Belgian company PEPITe, `http://www.pepite.be`, that has been used in previous versions of the course. PiXiT is not maintained any longer, which is reason for writing the new system HIC. In comparison, HIC is a lightweight system whose functionality is more limited and focused on being used in under specific teaching conditions.

Any student or teacher who happens to stumble upon this document and the HIC system are welcome to try it out, and the present author will be happy to receive feedback on your experiences with HIC. HIC is free software, you can redistribute it and/or modify it under the terms of the GNU General Public License. Details are found together with the source code, available at `http://www.ruc.dk/~henning/HIC`.

The remainder of this document explains in an informal way and by examples how to install and use the program.

3 Installing Processing

You need to have version 3 or higher of Processing installed on your computer. All (and more than) you need to know about Processing can be found here:

```
https://processing.org/
```

Locate the "Download" button, and following guidelines. You also need to install a library Processing named `video`.

Installation of the Processing library `video`: Start the Processing system, either by clicking its icon, or start HIC as described later (in which case it will notify you the first time that it misses `video`).

Locate a menu with the funny name "Sketch", go down to "Import Library..." and select "Add library". Then you see a window with lots of libraries; scroll down, select "Video" and press "Install", and it is done once and forall.

4 Download and install HIC

Go the webpage

`http://www.ruc.dk/~henning/HIC`

where you can find a link to the present document, and also to a packed file with a name of the form `HIC-Release`⟨*version-number*⟩`.zip`, at the moment of writing

`http://www.ruc.dk/~henning/HIC/HIC-Release1.2.zip`.

Download that file, unpack it and you will see a folder with the name `HIC-Release`⟨*version-number*⟩.

It has two subfolders, `HIC` containing the source code, and `SampleImages` containing sample images[1] for training and testing, organized in the way that HIC expects it. Place these folders on you hard disk wherever it is convenient.

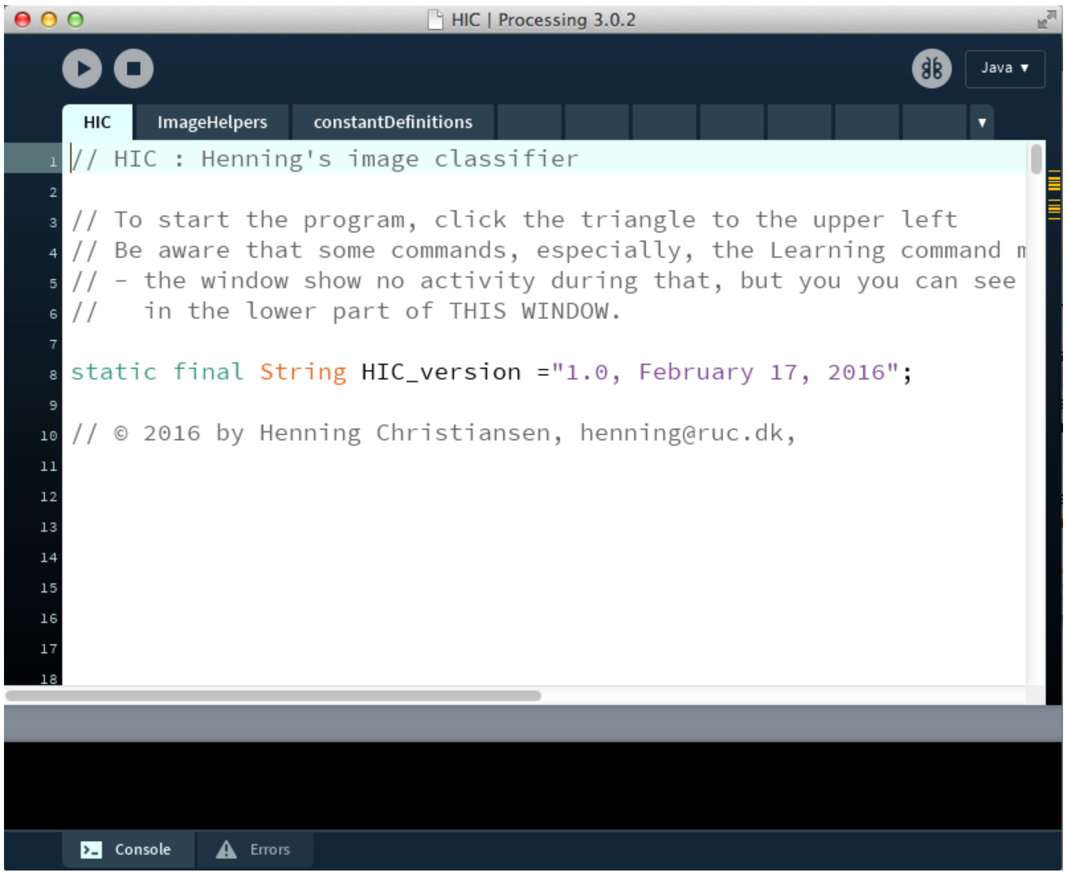
It is not necessary, but it can be useful to make an alias or short-cut to the file `HIC.pde` found inside the `HIC` folder.

5 Starting HIC

Double-click the `HIC.pde` file in the folder `HIC`, or use the alias/shortcut to it, if you made one as described above.

If Processing works as expected, a window pops up with the title "`HIC |` ....". (For some strange reason, the Processing system may also open another window with a weird name — close that and forget everything about it.) Now you should see a window that looks like this:

[1]Kindly made available by James Z. Wang, `http://wang.ist.psu.edu/docs/home.shtml`.



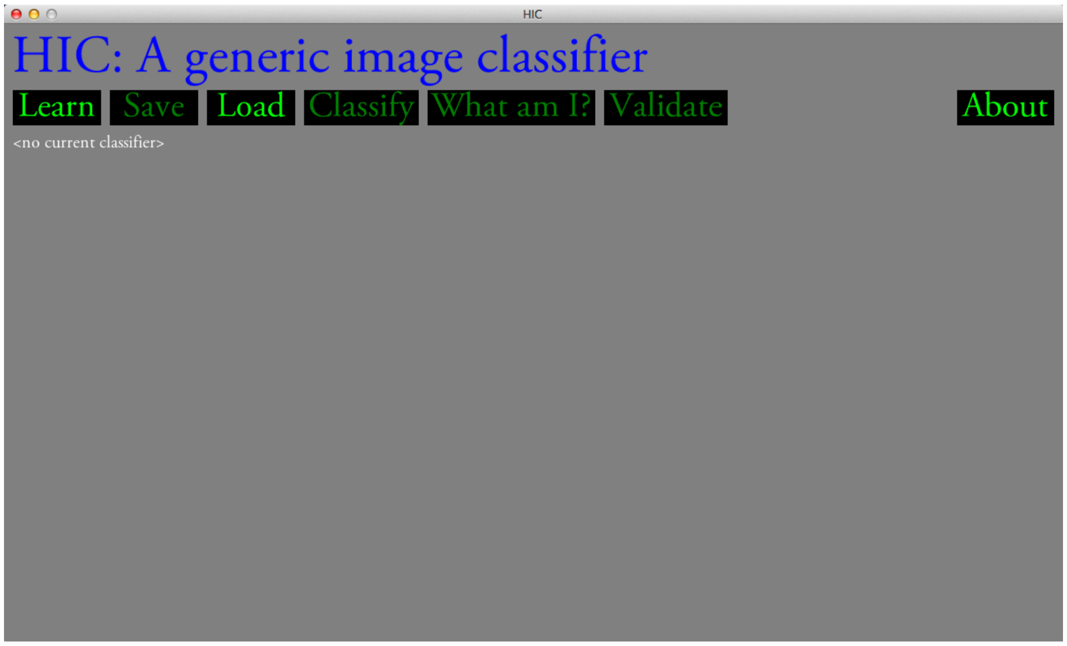
```
// HIC : Henning's image classifier

// To start the program, click the triangle to the upper left
// Be aware that some commands, especially, the Learning command m
// - the window show no activity during that, but you you can see
//   in the lower part of THIS WINDOW.

static final String HIC_version ="1.0, February 17, 2016";

// © 2016 by Henning Christiansen, henning@ruc.dk,
```

Now the Processing system is running, and to start HIC within Processing, click the triangle in the upper left; the following window should appear.



Almost at the top, you see a row of command buttons with green labels. Some of them are *active*, having a clear green font, while others are currently inactive, with a dimmed green colour.

To stop HIC, either close its window or click the square to the left of the above mentioned triangle in the Processing window (and if that does not work, use the "force quite" facility (or what it called in your computer's operating system)).

The Processing window is not important for now, except for the lower black field which shows a trace of what HIC is doing. When something goes wrong, it can be useful to have a look at that.

Before you can use HIC, you must have a collection of images organized in a specific way explained in the following.

6 How to organize your image files

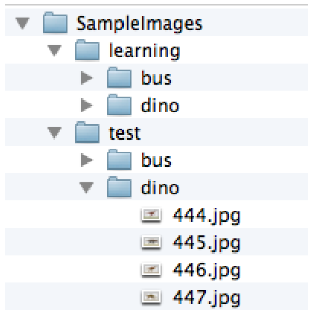
HIC must be trained with a collection of images which you have classifed already. Images in the following formats can be used: `.jpg`, `.gif`, `.tga`, `.png`.

It is not necessary to down-scale the images for HIC to digest them, and if you (unexpectedly) run into memory problems, it is possible have them scaled automatically when HIC reads them in, one by one; more about that later.

The folder `SampleImages`, delivered with the HIC system as described above, exemplifies how the files should be organized. You must decide two or more possible classes to which an image can belong. The way to inform HIC about your classes and your classification of the training images, is to place all images of a given class in a specific folder whose name determines the name of that class.

The folder `SampleImages` contains two sub-folders, `learning` and `validation`: In agreement with good practice in supervised machine learning, different data sets are assumed for training and for validation.

Both training and validation images are organized in the same way, one folder for each class. Each of the included sample images depicts either a bus or a dinosaur, coresponding to the two classes `bus` and `dino`.



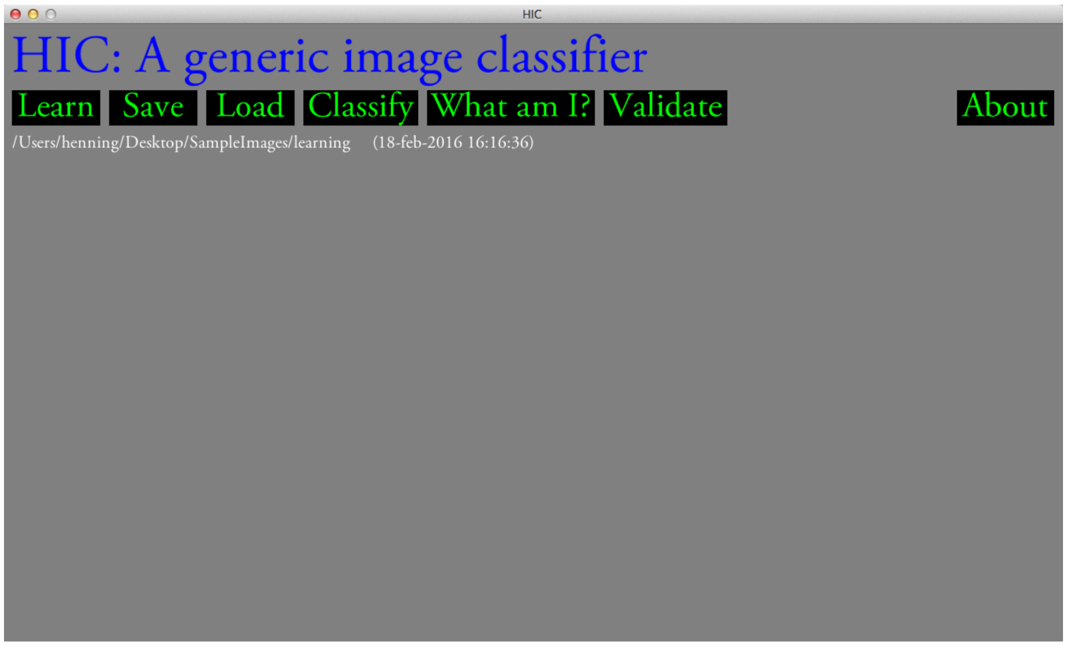
```
▼ SampleImages
    ▼ learning
        ▶ bus
        ▶ dino
    ▼ test
        ▶ bus
        ▼ dino
            444.jpg
            445.jpg
            446.jpg
            447.jpg
```

Having our training data organized in this way, we can how start playing with HIC.

7 The commands of HIC

7.1 Learn

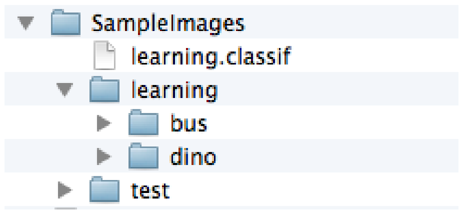
This command is used for the training phase. You are asked to select a folder of training data, organized in subfolders as described above. If you use many and large images, it may take some time. When it has finished (and nothing went wrong), HIC's memory contains a trained model, that can be used for classification of new images; it is called the *current model*. The text line below the commands indicates the training catalogue and a time stamp (as well as the fact that there is a current model now).



/Users/henning/Desktop/SampleImages/learning (18-feb-2016 16:16:36)

7.2 Save

This command is active when HIC's memory contains a trained model. Using "Save", you can store the trained model in a file and read it in later (without having to spend time on training it again, each time you want to use it). This file will get the same name as the training folder and the extension `.classif`, in the same place as the training folder.



In case you save more that one version of the model for the training same folder, the files names will be added a sequential number, e.g., `learning_3.classif`.

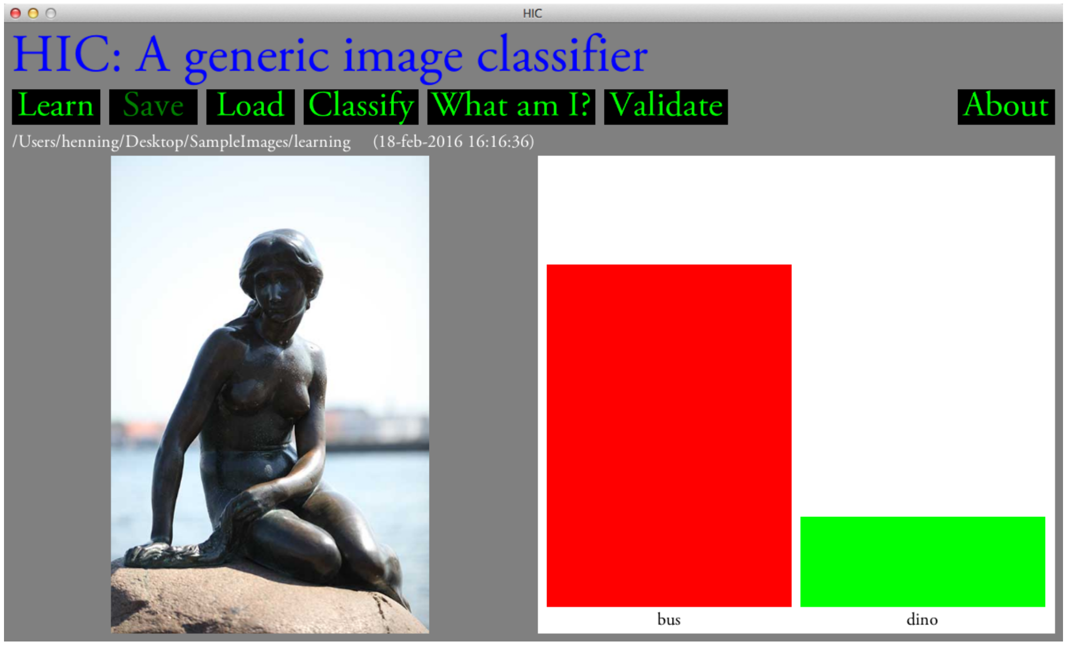
7.3 Load

You will be asked to select a file that contains a trained model, i.e., a file that has been created earlier by the “Save” command. That model becomes the current model in HIC.

7.4 Classify

You will be asked to select an image file, and HIC will classify it according to the current model. The answer is presented as a histogram, showing to which extent the image seems to belong to each possible category – this is defined precisely in [2], but you can think of it as a probability.

The following screen shot shows how the Little Mermaid[2] is viewed through a model trained on busses and dinos.


HIC
HIC: A generic image classifier
Learn
Save
Load
Classify
What am I?
Validate
About
/Users/henning/Desktop/SampleImages/learning (18-feb-2016 16:16:36)
bus
dino

7.5 What am I? [Not available in version 1.2.1]

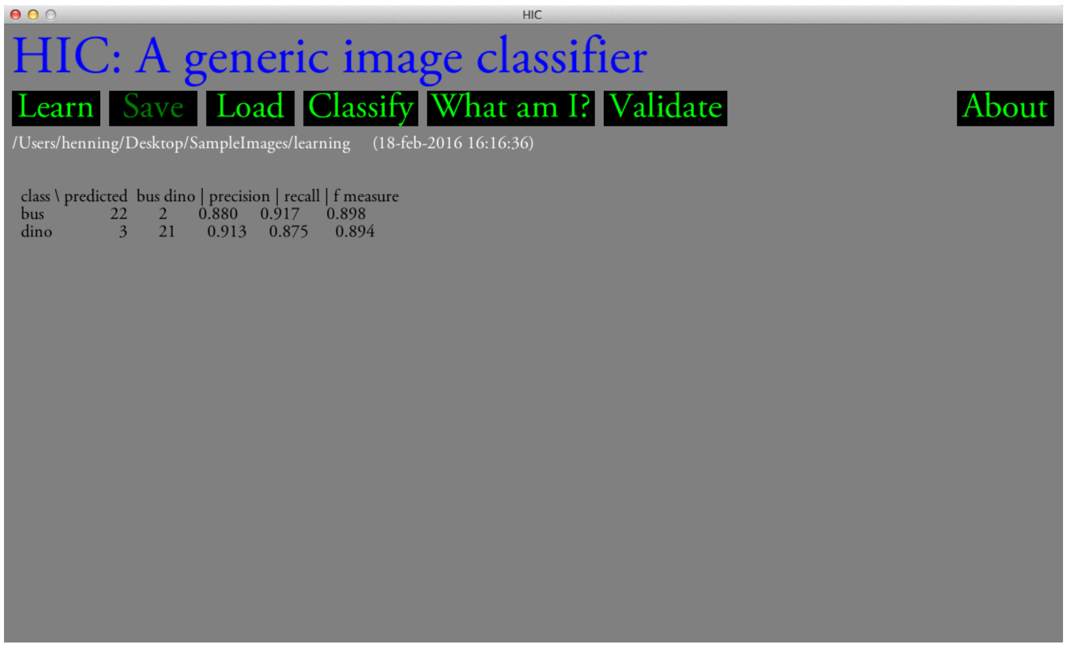
Try it out!

7.6 Validate

Validates the current model on a collection of test data that are classified in advance, i.e., it compares the classes predicted by a trained model with those classes given by a knowledgeable human, defining a golden standard.

You will be asked to select a folder with training data, assumed it to be organized similarly to the training data as described above in section 6. The result is given in the form a *confusion matrix*, together with the numbers for *precision*, *recall* and *f measure* for each category; check out your course material or Wikipedia for definitions of these notions. See the following example.

[2]Photo by the author.



HIC: A generic image classifier

/Users/henning/Desktop/SampleImages/learning   (18-feb-2016 16:16:36)

```
class \ predicted  bus dino | precision | recall | f measure
bus              22    2      0.880     0.917      0.898
dino              3   21       0.913     0.875      0.894
```

The formatting is not perfect, but readable. We notice, e.g., that out of 24 true busses, 22 of them are classified correctly and 2 wrongly. For bus recognition, the recall of this trained model is 0.880, the precision 0.917 and the f measure 0.898.

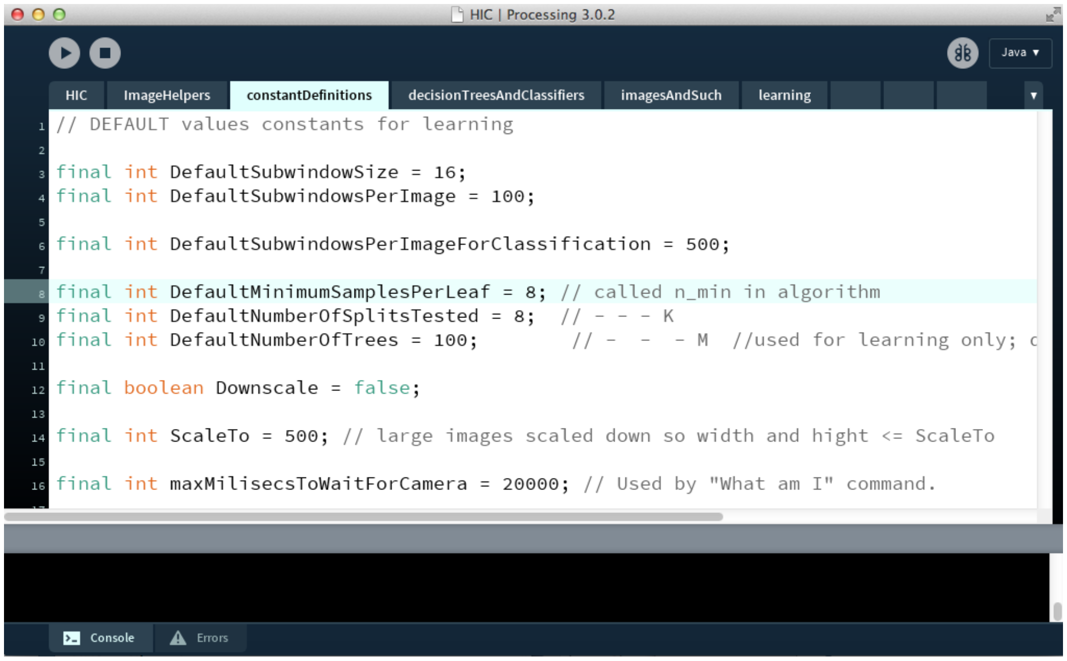
It is not possible to copy text from the HIC window, but the text is also printed in the Processing window (the lower, black part; section 5), so it can be copied and pasted into another text document.

7.7 About

Displays is short explanation about the HIC system.

8 Adjusting parameters for training and classifikation

There is no nice GUI in which you can adjust parameters, but it can easily been done in the Processing code. In the Processing window, select the tab `constantDefinitions`, and you will see the following text.



```
// DEFAULT values constants for learning

final int DefaultSubwindowSize = 16;
final int DefaultSubwindowsPerImage = 100;

final int DefaultSubwindowsPerImageForClassification = 500;

final int DefaultMinimumSamplesPerLeaf = 8; // called n_min in algorithm
final int DefaultNumberOfSplitsTested = 8;  // - - - K
final int DefaultNumberOfTrees = 100;        // -  -  - M  //used for learning only; c

final boolean Downscale = false;

final int ScaleTo = 500; // large images scaled down so width and hight <= ScaleTo

final int maxMilisecsToWaitForCamera = 20000; // Used by "What am I" command.
```

The first group of constants, all starting with `Default`... are explained in the paper [2]. The two next ones can be used if you want HIC to downscale the images for they are processed. There may be two reasons for downscaling:

- If you camera delivers images of, say, 20 or more Mpixels, it may be relevant to save some memory space. However, HIC does not store all images, but process them one by one, so this is likely not a problem.
- Down-scaling means data reduction which under some circumstances may improve precision and recall.

Litteratur

[1] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[2] Raphaël Marée, Pierre Geurts, Justus H. Piater, and Louis Wehenkel. Random subwindows for robust image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 34–40. IEEE Computer Society, 2005.

[3] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach. Third Edition*. Pearson International Editions, 2010.