Image Knowledge Builder

Image learning and recognition based on the CogniMem neural network technology

Version 1.0

A product of General Vision Inc.

LICENSE AGREEMENT

This License Agreement is a legal agreement between you (LICENSEE) and General Vision Inc. (General Vision) for the enclosed computer software, and any associated media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT").

License, rights and limitations
General Vision grants to the LICENSEE a revocable, non-transferable, and non-exclusive license to use the SOFTWARE PRODUCT. This license may not be shared or used concurrently on different computers. You may install one copy of the SOFTWARE PRODUCT, or any prior version, on a single computer for use by a single user. LICENSEE may make one copy of SOFTWARE PRODUCT for backup purposes. No other copies shall be made without General Vision's prior written consent.

Ownership and Copyright
The SOFTWARE PRODUCT, including all supporting documentation, is a proprietary product of General Vision. General Vision retains all title and copyrights in and to the SOFTWARE PRODUCT, the accompanying printed materials or electronic documentation and any complete or partial copies of the SOFTWARE PRODUCT.

Limited Warranty
General Vision warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials or electronic documentation and General Vision will make reasonable efforts to solve any problem issues arising out of the intended use of the SOFTWARE PRODUCT. To the extent allowed by applicable law, implied warranties on the SOFTWARE PRODUCT, if any, are limited to ninety (90) days. General Vision disclaim all other warranties and conditions, either express or implied, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose, with regards to the SOFTWARE PRODUCT.

Limitation of Liability
To the maximum extent permitted by law, in no event shall General Vision be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the SOFTWARE PRODUCT or the provision of or failure to provide support services, even if General Vision has been advised of the possibility of such damages. In any case, General Vision's entire liability under any provision of the license agreement shall be limited to the amount actually paid by you for the SOFTWARE PRODUCT.

Terms and termination
This license agreement is effective from the date of receipt and shall remain in full force until terminated. Without prejudice to any other right, General Vision may terminate this LICENSE AGREEMENT if you fail to comply with any of the terms and conditions of this license agreement. In such event, you must destroy all complete and partial copies of the SOFTWARE PRODUCT in your possession.

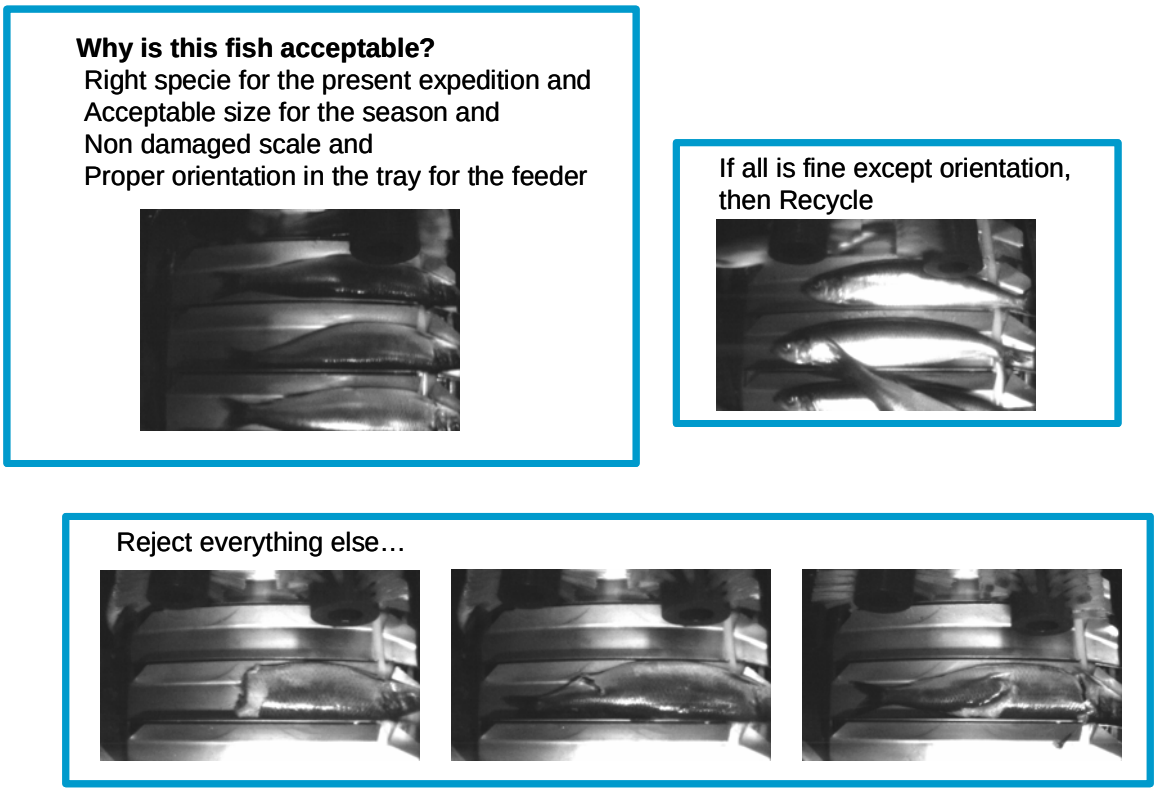
Trademarks
Image Knowledge Builder is a trademark of General Vision Inc.
Product names listed are trademarks of their respective manufacturers.

Support
support@general-vision.com
Tel 707-765-6150

TABLE OF CONTENTS

INTRODUCTION

One visual example is often better than long descriptions.



This "trainability" makes it possible to transfer a human expertise built on years of practice, with no need for intermediaries such as interpreters, consultants and programmers. Still, there is no unique way to recognize a family of objects and several teaching session might obtain the same results, only with more or less examples, more or less accuracy, more or less invariance to variations, etc.

Image Knowledge Builder (IKB) is an amazingly simple application which lets you try and validate different teaching strategies based on many images and settings to be more or less conservative, detail oriented, etc. Using your own images, you can build a knowledge by selecting relevant visual examples to learn and then test how this resulting knowledge performs in recognition on other images. A teaching session can be refined within a few iterations until you achieve the recognition accuracy requested by your application. You can then save the knowledge into a file (*.ckf) and distribute it to other systems with CogniMem networks.

Let's take the example of defect detection. Because it is impossible to model all the possible representations of defective parts, you can try two teaching methods: (a) be very conservative at teaching good parts and consider everything slightly different as defective. (b) be moderate at teaching defects so that more or less similar defects can be picked up. The selection of a teaching method can be influenced by the frequency of occurrence of the events or objects to recognize. If your production has 0.02% of defects, it will be difficult to capture these defects and a conservative teaching might be wiser.

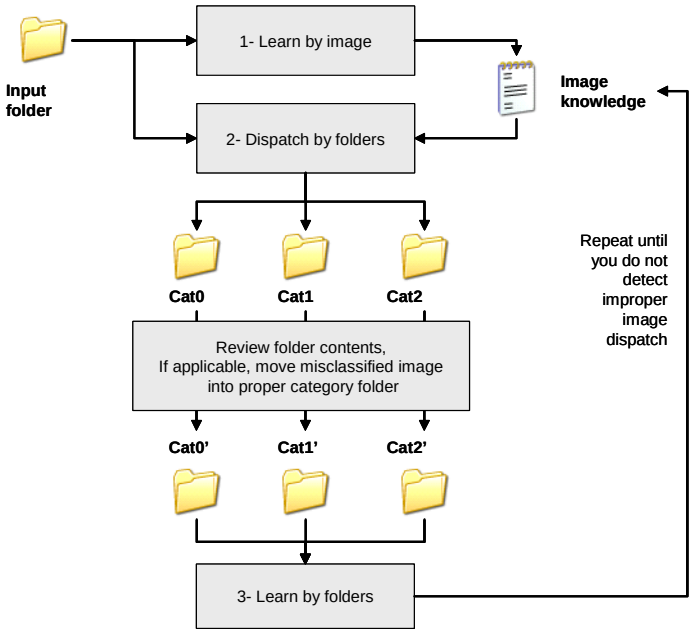
About CogniMem

CogniMem is a neural network featuring a highly-adaptive non-linear classifier and automatic model generator. The knowledge stored into its neurons can be saved and distributed. CogniMem is implemented in hardware with patented and fully-parallel neuron architecture. This means that its recognition speed is independent from the number of neurons in use and that high-speed recognition is possible for both simple and complex applications. Finally, the availability of the CogniMem chip can significantly reduce cost, size and power consumption issues for real-world applications.

New methodology for image recognition

Image Knowledge Builder lets you manipulate large amount of images with no efforts to (a) define the relevant examples to learn and (b) validate thoroughly the recognition accuracy of the resulting image knowledge base.

Phase of development	**Tasks and Tools**
Definition	Definition of the application and constraints. Collection of images representing all the possible scenarios of the application. This can be done with any video acquisition system, but preferably using the same sensor as the one used on the final system.
Training and validation	Definition of a learning strategy, targeted accuracy and throughput. Image Knowledge Builder to build and validate image knowledge bases recognizing the images collected in Phase1.
Implementation	Speed performance validation, definition of final hardware.



INSTALLATION

The Image Knowledge Builder program is installed by default into C:\program Files\General Vision.

If you own a full license, it is protected with a license file associated to the volume number of the hard disk on which the application will be installed. This file, cplib.lic, must be copied to the C:\Windows\System32 folder of the hard disk.

If your IKB installer does not come with your licence, please contact your vendor or General Vision to request one. You will be asked to show a proof of purchase and to provide your hard disk volume number. The later can be obtained by typing "Vol" under Start\Run\Cmd panel.

We recommend that you store your images to learn and/or recognize at a different location such as MyDocuments folder or MyPicture folders, or even the C drive.

Image Knowledge Builder is supplied with a folder of sample images called IKB Images. It contains sets of images arranged per theme and their default project files (*.ikp).

STARTING A NEW PROJECT

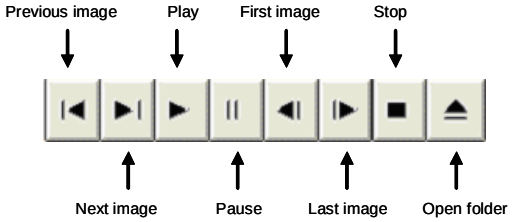
Organizing your image files


images

Images saved in JPEG or Bitmap format. When you start a new project, we recommend that you arrange your images into two folders: A Training Set and a Testing Set. The training set should be used to store the images used for the learning. Eventually they can be sorted into subfolders per category. The testing set should be used to validate the accuracy of the recognition.

Navigating through images

The navigation toolbar of IKB let's you select images from a folder and move through them step by step or continuously.



SETTING A NEW PROJECT

Size a region of interest

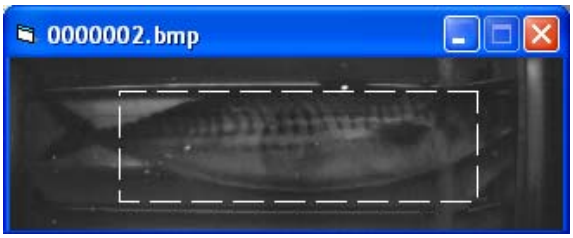
The first step when starting a new project is to define the size of your region of interest (ROI). This region can be a rectangle with a size such that it contains the objects or patterns to classify.

For example, if you want to differentiate good or defective parts based on the presence or absence of a few screws in the center of the part, you do not have to include the edges of the part in the region, but rather want to select the smallest area containing all the screws to identify.

In the case of texture classification, the region should be the minimum size of the repetitive pattern.

Define the size and location of the region of interest (ROI) using the mouse cursor. It should be equal to the minimum matrix of discrimination between your objects. The default region of interest is a square of 16x16 pixels located in the upper left corner of the image.

When selecting the region of interest, use the right mouse button to resize it and the left mouse button to change its position.



The ROI must be a rectangle with a size such that it contains the objects or patterns to classify. Depending on your application, you may have the choice to include a layer of background pixels, or on the contrary to exclude any background and edge information.

For example, if you want to differentiate good or defective parts based on the presence or absence of a few screws in the center of the part, you do not have to include the edges of the part in the expert region, but rather want to select the smallest area containing all the screws to identify.

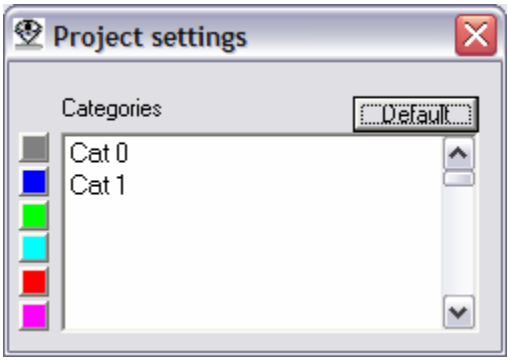
In the case of texture classification, the ROI size should be the minimum size of the repetitive pattern. The smaller the size, the more sensitivity to possible variations in the learned patterns (to either assimilate or discriminate them).

Changing your region of interest

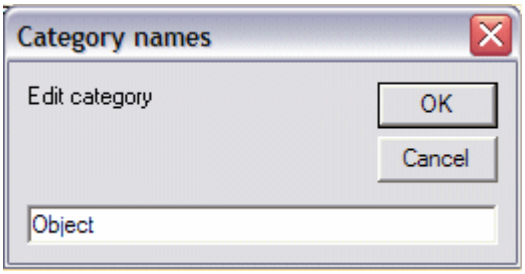
If you are interested in experiencing with scale invariance for an application you can easily do so using the Zoom In and Zoom out buttons accessible the main panel. If a knowledge exists at the time the region is resized, IKB maintains its aspect ratio to ensure compatibility with the existing knowledge. If the displayed region is not suitable the only alternative to set its size freely is to clear the knowledge using the Clear knowledge command from the Train menu.

Define the categories of objects

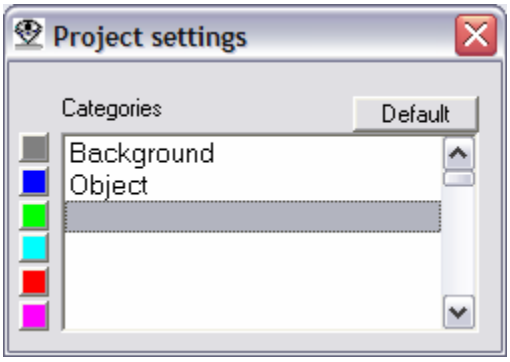
Define the possible categories to recognize in the View\Project Settings menu. The default names of the categories are Cat0 and Cat1.



To change a name, simply double-click inside its text and a dialog box will appear so you can edit a new name.



To add a new category, double click on the next blank line of the category names (i.e. the one next to the green button will become your third category, etc)



Examples of categories:
Project 0 .....Background, Object
Project 1 .....Good, Bad
Project 2 .....Accept, Reject, Recycle
Project 3 .....Present, Absent
Project 4 .....Grade A, Grave B, Grade C
Project 5 .....Event 1, Event 2, Event 3, Event 4
The second step is to edit the names of the categories of objects you wish to classify.

Following are suggestions of categories you can teach per type of inspection:

- Presence/Absence
  - Category 1= Present at the center of the region of inspection
  - Category Null= Absent or not at the center of the region of inspection
- Conformity/Defect detection
  - Category 1= Acceptable part at the center of the region of inspection
  - Category 2= Non-acceptable part at the center of the region of inspection
  - Category Null= Nothing, or part not positioned at the center of the region of inspection
- Part identification/sorting
  - Category 1= Part #ABC at the center of the region of inspection
  - Category 2= Part #DEF at the center of the region of inspection
  - Category 3= Part #GHI at the center of the region of inspection
  - Category Null= Nothing or parts not centered in the region of inspection
- Video Monitoring
  - Category 1= Event for incoming person
  - Category 2= Event for incoming car
  - Category 3= Event for incoming pet
  - Category Null= Nothing happening

IMPORTANT:
The first category in the list (Category 0) is reserved for the Null category. It allows teaching what does not belong to a known category, but rather to some background information. Make sure to keep this concept in mind when you rename this particular category. You can see it as teaching counter-examples to avoid mis-classifications.

Select preferred colors

You can also change the color associated to each category by clicking at the color box shown to the left of the category.

You can edit up to 256 categories. Their names are saved in both the project settings files and knowledge files. The colors of the categories are only saved in the project settings files.

TRAIN A RECOGNITION ENGINE

A simple learning mechanism

The following paragraph describes the simple learning behavior of the CogniMem neural network so you can quickly understand the mechanism of its built-in model generator.

When a new example of category A is presented for learning, the neural network first attempts to recognize it.

- If the example is not recognized by any existing neurons, a new neuron is automatically added to the network in order to store the new example and its category value A.
- If the example is recognized by one or more neurons and they all agree that it matches a category A, then the new example is discarded since it does not add any new information to the existing knowledge base.
- If the example is recognized by several neurons where one or more identify it with a category other than A, these neurons which are in disagreement with the category to learn automatically reduce their similarity domain to exclude the new example. This corrective action changes the knowledge base by making certain neurons more conservative in their classification process.

As a result, a learning operation can have the following impact on a knowledge base:

- Add a new neuron
- reduce the similarity of existing neurons
- reduce the similarity domain of existing neurons and add a new neuron
- do nothing

It is important to realize that when the similarity domain of neurons are reduced, it might very well happen that an example which was recognized with the correct category at an earlier time is no longer recognized as such because the neuron which originally recognized the said example now excludes it. Therefore repeating the learning of all examples until the number of neurons reaches a constant is a good method to build a knowledge.

Have I learned enough?

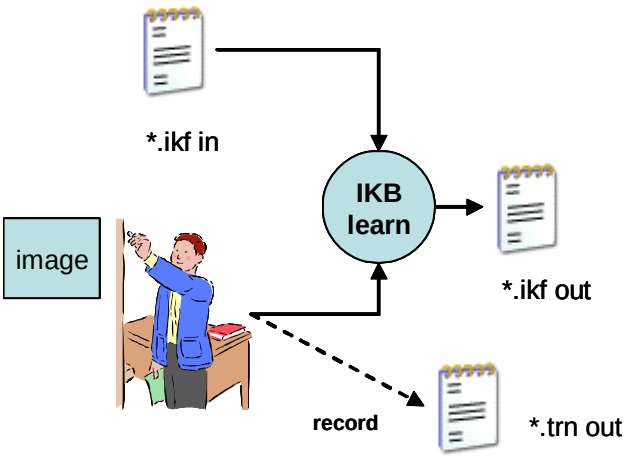
The learning curve should be asymptotic. It tells you that you can build a knowledge capable of generalization, and therefore capable of recognizing objects never seen before but similar to learned examples. Obviously, the number of neurons must become steady and be less than the number of taught examples.

For a given set of examples, you can generate multiple knowledge bases using different region size. A knowledge with fewer (and bigger) neurons has better generalization capabilities, but can generate classification with uncertainties if not mistakes. A knowledge with more neurons (and smaller) has a more conservative behavior, will make less mistakes, but will classify new situations as unknown rather than associating them to known examples.

Tracability of the learned examples

The images selected for a training session do not necessarily all contribute to the enrichment of the knowledge. Furthermore, this influence depends on the examples previously learned.

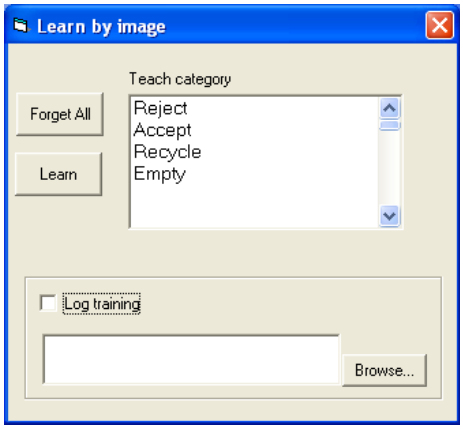
Learn images manually



Once you have defined the size of a region of interest and its possible categories, you can screen image and position the ROI over relevant examples. Choose a category button and the example is learned, possibly adding a new model to the existing knowledge. Note that you cannot resize the ROI between two learning operation, but you can move it at any time. You can learn more than one example per image, if applicable.

When you click at a category, the region in the image is outlined with the color associated to the selected category.

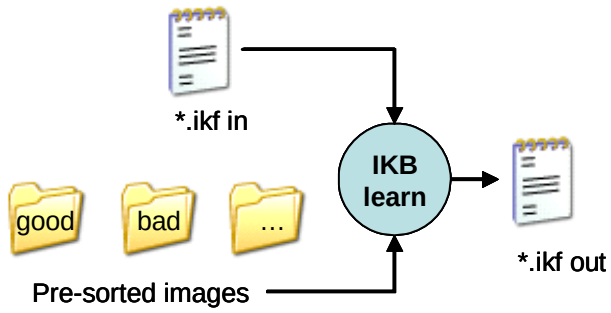
IMPORTANT:
The first category in the list is reserved for the Null category. It allows teaching what does not belong to a known category. You can see it as teaching counter-examples to avoid mis-classifications.
.
You need to click the Learn button so the example is actually learned. If the example brings novelty to the current recognition engine, a new neuron is added to the knowledge.
Teaching examples of Category 0 never adds any neuron to the current knowledge, but it helps delimit the influence field of existing neurons and make an engine more conservative and accurate.



Whenever you teach a new example, IKB automatically re-learns all the examples previously taught and still saved in the training buffer and this until the knowledge reaches a constant neuron number. As a result, you may notice that learning a single new example generates more than one neuron. Refer to the paragraph "learning mechanism" for more information.

Learn from image folders

Requirements:
- Images must be sorted per category and saved into one folder per category
- The region of interest is the same in all the sorted images



A simple way to build an image knowledge automatically is to dispatch images in advance per category. Once you have defined the size and position of a region of interest, learning can be executed automatically.

Learning by folder is a very convenient utility if you can sort in advance the images you wish to use for training and if the object to learn always appears in the same region of interest..

For each category defined in your project, you can select a folder containing images of this category.

Clicking the Start button launches the learning operation. All the regions of interest in the selected folders (checked) are learned. If a knowledge is already loaded in the application, you will be prompted to erase it or to add to it.

Looking at the number of processed images, you can notice that the learning of the folders is repeated a certain number of time. This operation allows building a knowledge which is invariant to the order in which the examples are taught.

Before clicking the Start button, you can select one of the two options:

Option1
Record the images which will contribute to the creation of neurons. These images are automatically saved in a sub-folder of their folder of origin. For example, if the learning of image_003 belonging to the "Novelty" folder adds a neuron to the existing knowledge, IKB copies image_003 in Novelty/Ref_Novelty.

Option2
Record the images which will contribute to the creation of neurons into an annotation file. This annotation file of your choice.

Q & A: The learning never stops and has to be interrupted manually

This can occur if the set of images to learn contains errors or inconsistencies. For example, the knowledge cannot converge if a same image is stored by mistake in two different Category folders. The "faulty" examples can be identified if you select to record a training file. Their names should appear repeatedly at the end of the file.

Invariance to the sequence of the examples

IKB performs iterative learning which ensures that the knowledge is invariant to the order in which the examples are taught.

When using the Learn by Folder method, the program repeats the learning of each folder until no more neurons gets added to the knowledge between two iterations.

When using the Learn by Image method, the program stores all the examples in a log file and repeats the learning of all of them until no more neurons gets added to the knowledge between two iterations. The log

file can contain up to 1000 examples and is cleared automatically when it is full or upon a Forget operation.

TEST A RECOGNITION ENGINE

Once a knowledge exists (i.e. your project contains at least one neuron), you can perform the following recognition operations:

1) Review the classification of images by screening images manually with the navigation toolbar. The recognized category (or categories, in case of uncertainty) is reported in the main panel. Also the region outline takes the color associated to the first category.
2) View images by contents skipping to images with a given category
3) Sort images by contents moving or duplicating them into category subfolders
4) Recognize images by folders

Validating a knowledge

The validation of a knowledge must be done on an amount of images much larger than the one used to teach the recognition engine.

Recognize what you have taught

A first validation of a knowledge consists of recognizing the examples learned to build this knowledge.

The classification of these examples should reveal that a majority of them are positively identified with the correct category and that a minority of them are classified with an uncertainty between the correct category and other possible categories.

The percentage of uncertain classifications can be an indication that the examples and/or their selected region of interest are not necessarily the best.

Recognize new examples

The true accuracy of a knowledge is rated by observing its classification of examples never seen before. It is realistic to expect false classifications at first, but CogniMem can easily correct these inaccuracies by learning new examples.

The IKB sorting utilities let you locate mis-classifications very quickly and correct them in a few mouse clicks.

Note 1: Multiple iterations might be necessary to obtain a robust knowledge which can adapt to the variations of your objects.

Note 2: In addition to the examples pre-sorted in your Training Set folders, your knowledge also depends on the size and location of the region of interest (ROI). If you are not satisfied with the accuracy of a knowledge, you can easily build a new one using the same examples but with a slightly different ROI.

Complete or expand a knowledge

If you move the mis-classified images back to the correct folder under the Learning Set and run the "Learn by Folders" again, repeat until all classifications are correct.

An incorrect classification is due by one of the following:
- an insufficient training
- an erroneous training

- a misleading training

To help you identify such problems, you can review the content of your knowledge file through the View Models utility, displaying the patterns stored in the neurons and their associated categories.

Sort by contents

This utility lets you select a folder of images and dispatch these images into specific folders depending on the category of the region of interest.

The default button automatically creates sub-folders with the name of the categories.

Once the dispatch has been done, you can easily review the accuracy of the classification by opening the category folders under Windows Explorer.

Example: The screen to the right shows that a mackerel (image00049.jpg in the Testing\Recycle folder) has been dispatched into the Recycle folder, when the application requires that a mackerel be ejected (i.e. the catch of the day being for herrings).

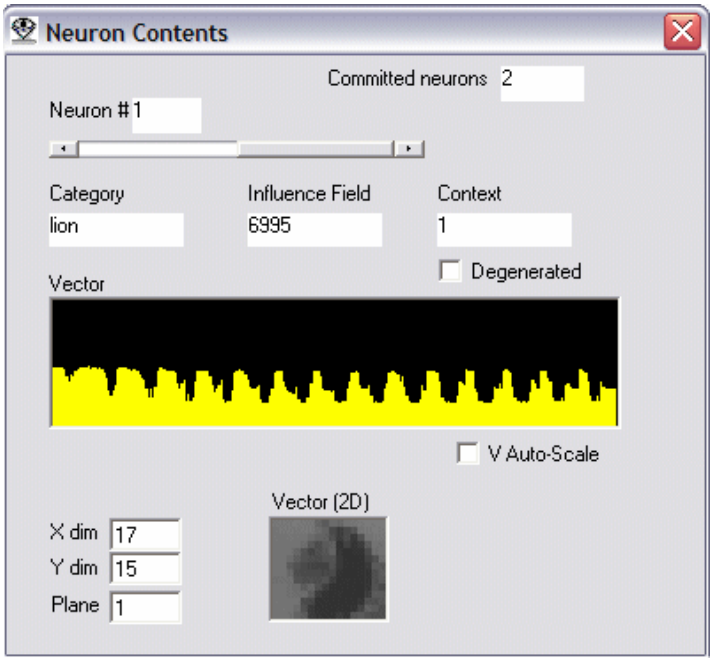
To correct this mis-classification, you will have to teach this example of a mackerel and associate it to the Recycle category. This is done by simply dragging the image from the Testing\Recycle folder into the Training\Reject folder. The next time, you will run the utility "Learn by Folder", this image should be dispatched properly.

With the image sorting, you can quickly notice if the images recognized as "Category A" really belong to this category or not. If so, you can assume that the engine is well trained to identify category A. On the other hand, you will have to verify that images of category A cannot be found in the other folders. Now, if you notice that images stored in "Category A" often belong to category C, you can immediately teach more examples of objects belonging to category C. These observations help you identify the categories of objects which require more teaching due to the complexity or diversity of their features. You can also easily locate the categories which introduce confusion and need to be taught with a conservative approach. These observations can also lead you to define intermediary categories or, on the contrary, to merge categories together.

KNOWLEDGE CONTENT MANAGEMENT

View models generated by the neural network

The View Model menu allows to review the contents of the neurons which represent the knowledge. This utility can be helpful to understand why some erroneous or uncertain classifications occur. It can also point to examples which are irrelevant because showing too much background.



The content of a neuron comprises a Category, an Influence Field, a Context, a Degenerated flag if applicable, and of course a Vector.

The 1D vector extracted from the images by the CogniMem chip is actually a sub-sampling of the pixels within a region of interested. Knowing the dimensions of this region (Xdim x Y dim x Plane), the 1D vector can be plotted as a 2D icon.

In the example to the left, the neuron #1 describes the lion. The vector was extracted from a region of 17 x 15 pixels in a monochrome image (plane=1).

The **influence field** is an attribute of the neuron limiting its generalization capabilities. It is a dynamic attribute entirely controlled by the neuron itself. From the time the neuron is created, its influence field can only reduce as new examples are taught and the knowledge gets more accurate. The smaller the influence field the more conservative the neuron. For a complete description, please refer to the CogniMem Reference Guide.

Using the example above, if a vector extracted from a new image is distant from the vector stored in the neuron #1 by a value less than 6995, the neuron #1 will fire and report that the input vector is similar to a category "lion". All neurons holding a learned vector will behave the same. The Easy_Video_Trainer interface returns the category of the neuron with the best match. However more than one neuron can fire with identical or different categories. For more information, please refer to the CogniMem technology Reference Guide.

The **context** is equivalent to an index to a specific knowledge. In the case of Easy_Video_Trainer only one knowledge can be built at a time, so Context is always equal to 1.

Remark: Plane=1 for monochrome images, and 3 for color images.

Looking at the reference patterns stored in the neurons can reveal a lack of examples showing discriminant features, or an abundance of examples which are too similar.

Save /Load Knowledge File

CogniMem Knowledge files (*.ckf) contain the names of your categories, the size of your region of interest and the knowledge to classify it (i.e. the models stored in the neurons).

These commands automatically save the current project file and image knowledge file. The latter can have an empty knowledge.

INDEX