

Image Knowledge Builder

Image Learning and Recognition
powered by NeuroMem network

Version 2.5.4
Revised 04/8/2018



A. Contents

1	INTRODUCTION	4
1.1	WHAT CAN I DO WITH IMAGE KNOWLEDGE BUILDER?.....	4
2	INSTALLATION.....	4
3	GETTING STARTED.....	4
3.1	PRACTICE WITH OUR EXAMPLE PROJECTS	6
3.2	METHODOLOGY STEP-BY-STEP	6
4	EXAMPLE PROJECTS	7
4.1	GENERAL INSTRUCTIONS.....	7
4.2	BGA INSPECTION	7
4.3	FACE RECOGNITION ON SOCCER TEAMS AND THE FERRET DATABASE	7
4.4	ROAD MARKING DETECTION	8
4.5	INKJET CHARACTER RECOGNITION.....	8
4.6	GLASS SURFACE ANOMALY DETECTION.....	9
4.7	WAFER INSPECTION	10
4.8	INTRODUCTION TO IMAGE COMPRESSION WITH NEUROMEM.....	10
4.9	KANJI CHARACTER RECOGNITION	11
5	STARTING A NEW PROJECT	12
5.1	DEFINE YOUR CATEGORIES AND ASSOCIATED COLOR TAGS	12
5.2	NOMINAL SIZING AND FEATURE EXTRACTION	12
5.3	STATUS BAR	13
6	THE LEARNING MODES & TOOLS.....	13
6.1	SUPERVISED LEARNING WITH ANNOTATIONS.....	13
6.2	VIEW LOCAL CLASSIFICATION	13
6.3	VIEW KNOWLEDGE CONTENT	14
6.4	UNSUPERVISED LEARNING WITH CODEBOOK GENERATION	15
6.5	UNSUPERVISED LEARNING OF A GOLDEN TEMPLATE	15
7	THE RECOGNITION MODES AND TOOLS.....	16
7.1	COPING WITH CHANGE OF SCALE	16
8	ANNOTATION AND RESULT FILES.....	17
8.1	COMMON FORMAT	17
8.2	IMPORTING ANNOTATIONS.....	18
9	PROJECT AND KNOWLEDGE FILES.....	18
9.1	COGNISIGHT PROJECT FILE.....	18
9.2	PREFERENCE FILE	18
9.3	SPECIAL PROJECTS	18
9.3.1	<i>Default.csp</i>	18
9.3.2	<i>Last.csp</i>	18
10	APPENDIX A: IKB VERSION	19
11	APPENDIX B: WHAT IS NEW?.....	22
11.1	IKB VERSION 2.5.4	22
11.2	IKB VERSION 2.5.1	22
11.3	IKB VERSION 2.4.....	22
11.4	IKB VERSION 2.3.....	22

12	TROUBLESHOOTING	23
12.1	UNABLE TO LOAD DLL “COGNISIGHT_XXX.DLL”	23

Image Knowledge Builder is a product of General Vision, Inc. (GV)

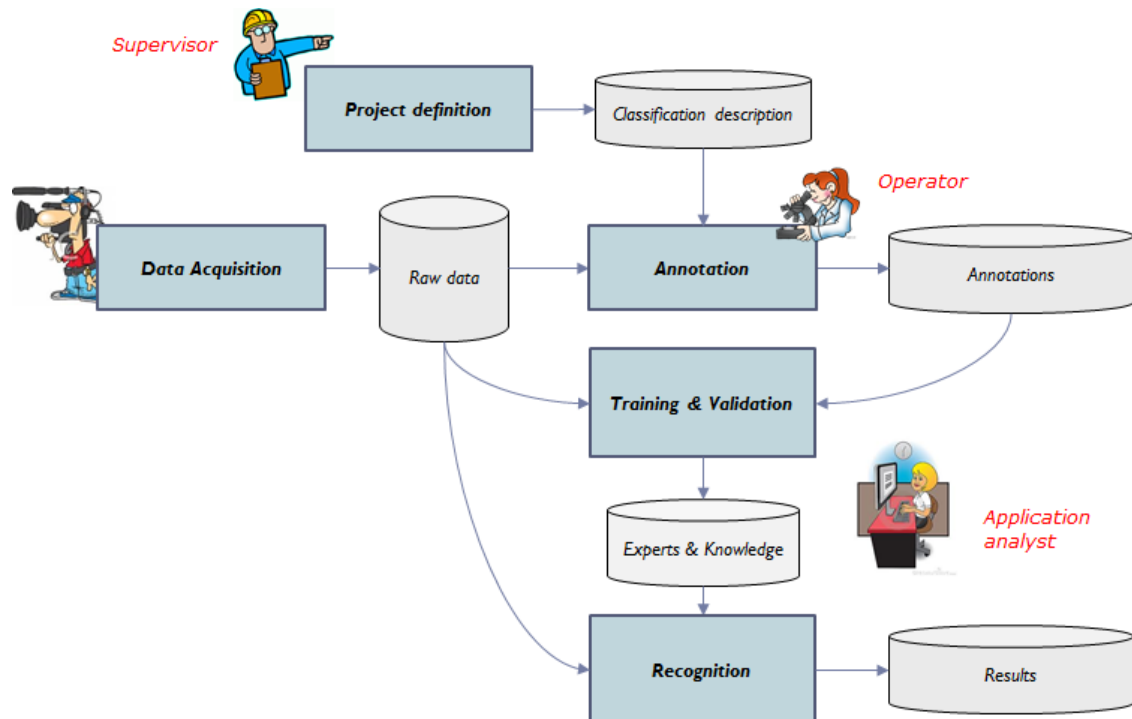
This manual is copyrighted and published by GV. All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of GV.

For information about ownership, copyrights, warranties and liabilities, refer to the document [Standard Terms And Conditions Of Sale](#) or contact us at www.general-vision.com.

1 Introduction

1.1 [What can I do with Image Knowledge Builder?](#)

- Learn objects and surfaces under a supervised mode
 - o Edit and save annotations in your images or video frames
 - o Train the neurons in a step by step or batch mode
- Build a knowledge to recognize objects
 - o Learning annotations edited by the user
 - o Building codebook (unsupervised mode)
- Generate outputs showing the recognized objects
 - o Visual color and text overlay
 - o Map of the hit locations
 - o Map of the category locations
 - o Map of confidence
 - o Text list of the recognized objects or anomalies

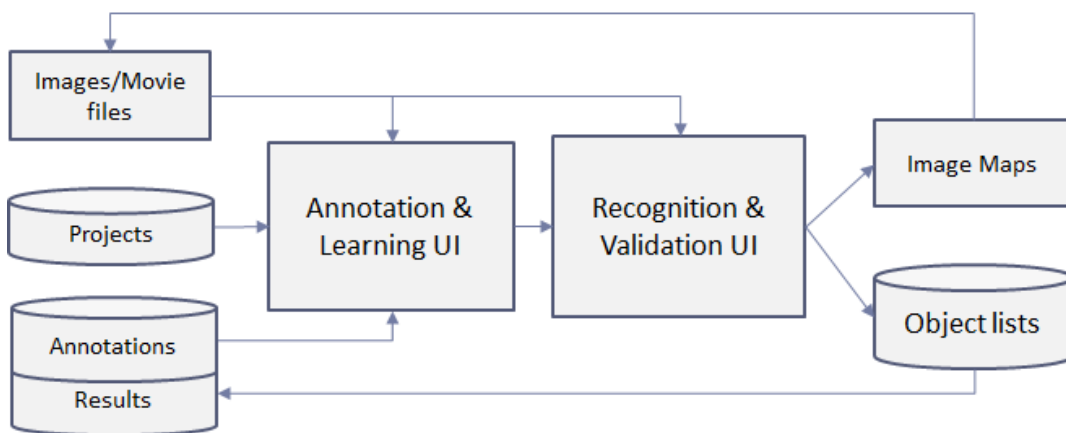


2 Installation

- Double click at the setup.exe file and follow the instructions on the screen.
- After the display of the panel "Installing GV Image Knowledge Builder...", you may get a pop-up message asking you to confirm the installation of the software from an Unknown Publisher. Click Yes to proceed.
- Upon termination, click at Start menu \All Programs\General Vision\Image Knowledge Builder

3 Getting Started

The main panel is divided into 2 areas:



Three learning modes and five recognition modes are presently available in the Image Knowledge Builder. The table below indicates the learning methods appropriate for the envisioned recognition methods.

Recognition mode / Learning modes	Hit Points	Clusters	Anomalies	Map of Models	Template Matching
Annotations	x	x	x		
Codebook			x	x	
Golden template					x

3.1 [Practice with our example projects](#)

Image Knowledge Builder is supplied with example projects and their accompanying images and videos. They are usually organized per topic and the name of the project file matches the name of the folder where you will find the images used for the training and the verification.

Our example projects are described in a [next chapter](#).

3.2 [Methodology Step-by-Step](#)

- Select a folder of images to use for training
 - o Default location c:\MyDocuments\General Vision\Images
- Select a project file if any already exist for your application
 - o Default location c:\MyDocuments\General Vision\Images
- Observations:
 - o The status bar at the bottom of the screen reports the settings of the project and the size of the knowledge (i.e. number of committed neurons)
 - o If any annotation file has the same rootname as the project in the image folder, the annotations associated to the image on screen are displayed.
- Possible actions on the first image
 - o Learn existing annotations
 - o Edit and learn new annotations
 - o Recognize a region or the entire image
- Go to next image to observe the recognition results obtained with the current knowledge
- Possible actions on the next image
 - o Learn existing annotations to teach the neurons with new examples
 - o Edit and learn new annotations
 - o Recognize a region or the entire image
- Observations:
 - o If the scanning steps are too high, the report may skip some locations which should be recognized positively
 - o Check Auto recognition and Full Image option allow to automate the search across multiple images
 - o Annotations are saved to a log file as soon as they are learned. This log file can be saved for later or further use
- Try building a new knowledge delivering better accuracy using the same annotations, but a different feature
 - o Go back to the first image
 - o Clear the current knowledge
 - o Open the Project/Settings menu and change your parameters
 - o Proceed again with the learning of annotations

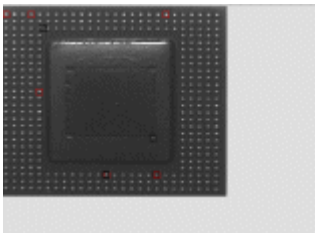
4 Example projects

4.1 [General Instructions](#)

The proposed examples can be reviewed in a few minutes as follows:

- Use the navigation bar to open an image stored in C:\\yourUser\\Documents\\General Vision\\Images**Topic**.
- Select File\\Open Project from the Main menu and look for an existing *.prj file with the **Topic** label.
- A summary of the project definition is displayed in the Status bar at the bottom of the screen: size of the knowledge, type of feature extraction, image scanning mode.
- If neurons are committed, their models can be viewed in the View/Knowledge menu, or you can click directly at the Reco button to view the results of the recognition.
- If annotations for the project exist in the selected folder of images, they are overlaid in the image.
- If no annotations exist for the selected image, you can edit your own manually.
- You can click at the Learn button and when done at the Reco button.

4.2 [BGA inspection](#)



Example in machine vision where a non-linear classifier is a simple remedy to learn a diversity of examples in a few mouse clicks. Try to achieve the same results in less than a minute with a threshold technique!

1. Click Browse in the Navigation toolbar and load the image stored in Images/BGA folder.
 - a. Assuming that the Learn method is set to the default “Annotations”, the program should automatically find and load the annotations associated to this image and stored in the file BGA_GT.txt stored in the same folder.
 - b. Note that the annotations include “background” annotations (in black). They behave as counter examples to ensure that the neurons learning solder balls (in red) do not over generalize too much.
2. Select Load project in the File menu and load the file bga.prj stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is empty. The size of the region to learn and recognize is 12x12 pixels and the feature extraction a monochrome and subsample.
3. Checked the Auto checkbox to automatically recognize the image when a Learn operation or Load image is executed. Verify that the Reco method is set to Objects.
4. Click Learn to learn the annotations seen on in the learning panel. Verify that all the solder balls are highlighted in the right panel. Some appear more reddish than other when a solder ball is recognized at 2 adjacent positions.

4.3 [Face recognition on soccer teams and the ferret database](#)



A great example for cooperative face detection when people are facing forward.

Observe how 7 models can recognize soccer players from four different teams as well as faces extracted from the ferret database!

1. Click Browse in the Navigation toolbar and load the image stored in Images/Faces_SoccerPlayers folder
2. Select Load project in the File menu and load the file faces_soccerplayers_kn7.prj stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is already built and consists of 7 neurons. The size of the region used to learn and recognize the characters is 27x 36 pixels. The feature extraction a monochrome normalized subsample.
3. The knowledge can be displayed by clicking "View Knowledge" button.
4. Click the Reco button to inspect the entire image. Note that this can take a few seconds in simulation.
5. Select the Auto checkbox next to the Reco button. Go to the next image by clicking at the ">" button in the navigation bar. Review the results. Proceed to next image.
6. Using the navigation bar, select the first image supplied in the Images/faces_ferret folder. The Ferret database is a known repository of faces looking at the camera from approximately the same distance. The region of interest is clearly too small in this case to cover the eye and nose area of the person. In the navigation bar, reduce the image by a scale 1/3 and you should observe that the face is then detected.
7. Go through several images of this same folder by clicking at the ">" button in the navigation bar.

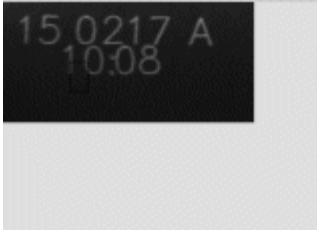
4.4 [Road Marking detection](#)



Monitor the edge and center lane while driving on Hwy 101

1. Change the data source in the Navigation toolbar to work on "Movie" and load the RoadTrip.avi file stored in Videos/RoadTrip folder
2. Select Load project in the File menu and load the roadtrip.prj file stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is empty. The size of the region to learn and recognize as the lane marking is 24 x 8 pixels. The feature extraction a color normalized subsample.
3. Click the Learn button to build the knowledge
4. Click the Reco button to inspect the entire image. Note that this can take a few seconds in simulation.
5. Select the Auto checkbox next to the Reco button. Go to the next image by clicking at the ">" button in the navigation bar. Review the results. Proceed to next image.

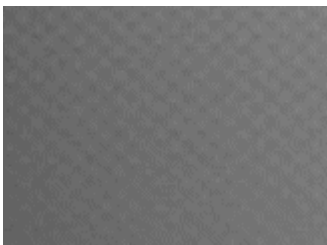
4.5 [Inkjet character recognition](#)



This example quickly demonstrates how the feature extraction can impact the ease of deployment of an application, in particular limit the number of examples to teach without compromising the accuracy of the recognition.

1. Click Browse in the Navigation toolbar and load the first image stored in Images/OCR_inkjet folder: Im_110217A_1008.jpg.
 - a. Assuming that the Learn method is set to the default “Annotations”, the program should automatically find and load the annotations associated to this image and stored in the file Im_110217A_1008_GT.txt stored in the same folder.
2. Select Load project in the File menu and load the file ocr_inkjet.prj stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is empty. The size of the region to learn and recognize is 36x60 pixels and the feature extraction a monochrome and normalized subsample.
3. Checked the Auto checkbox to automatically recognize the image when a Learn operation or Load image is executed. Verify that the Reco method is set to Objects.
4. Click Learn to learn the annotations seen on in the learning panel. Verify that the digits are highlighted with the proper categories in the right panel. The color legend can be seen and changed in the Options/Settings menu.
5. Click at the Next arrow in the navigation bar to proceed with the next image. Notice that on the third image, a new annotation appears for the digit “6” which has still not been taught. Click Learn to add it to the current knowledge composed of 7 neurons as reported in the status bar. You can review the results on the next and previous images navigating through the various images supplied in the OCR_inkjet folder.
6. Now, let’s repeat the same routine, but with a slightly different feature extraction:
 - a. Erase the knowledge by selecting the Clear command in the Knowledge menu
 - b. Open the Options/Settings panel and uncheck the “Normalize” checkbox. This means that the amplitude of the subsample will not be expanded to the range [0, 255].
7. Load the first image again with its annotations, learn and review the results of the recognition:
 - a. Select Options/Details at cursor and move the cursor over the image
 - b. Observe the number of misfiring such as in (320,90)
 - c. Observe how the number of uncertainties such as in (54,70)

4.6 [Glass surface anomaly detection](#)



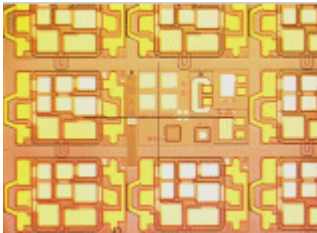
This example is a good illustration of the power of the NeuroMem RBF classifier to detect novelty which in this case represents the occurrence of an anomaly.

The learning is unsupervised and consists of using the neurons to build a codebook of good texture pattern from a reference image. In new images, the program only reports the locations where no neuron fires, thus indicating an anomaly in the glass pattern.

1. Change the data source in the Navigation toolbar to “Image” and load the Glass0_good.png file stored in Images/Surface_Glass folder
2. Select Load project in the File menu and load the surface_glass.prj file stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is empty. The size of the glass area to learn and recognize is 16x16 pixels. The feature extraction a monochrome subsample.

3. Select the “Codebook” learning mode to build the knowledge and click the Learn button. Using the Maxif of 1500, and a learning step of 16 along the horizontal and vertical axes, 12 neurons are committed.
4. Select the “Anomalies” recognition mode, check the Auto checkbox and click the Reco button.
5. No anomaly is reported in the good image.
6. Go to the next image by clicking at the “>” button in the navigation bar. Anomalies should be reported in the other three images.
7. You can practice with the same routine using bigger or smaller values of the maximum influence field and observe the differences in the number of committed neurons and number of discrepancies identified in recognition.

4.7 [Wafer inspection](#)



This example illustrates the use of an unsupervised learning method to learn each block of a reference image such as a wafer, a printed material, etc and detect significant local differences in other similar images.

1. Click Browse in the Navigation toolbar and load the image Good.tiff stored in Images/Wafers folder
2. Select Load project in the File menu and load the file wafer_inspection.prj stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is empty. The size of the region to learn and recognize is 16x16 pixels and the feature extraction with the value 3 means that the subsample includes the color information. The maximum influence field of 2000 is an important parameter in this example and tunes the conservatism of the neurons.
3. Select the “Golden template” learning mode to build the knowledge and click the Learn button. Using the Maxif of 2000, and a learning step of 16 along the horizontal and vertical axes, 674 neurons are committed.
4. Select the “Anomalies” recognition mode, check the Auto checkbox and click the Reco button.
5. No anomaly is reported in the good image.
6. Go to the next image by clicking at the “>” button in the navigation bar. Anomalies should be reported in the other three images.
7. You can practice with the same routine using bigger or smaller values of the maximum influence field and observe the differences in the number of committed neurons and number of discrepancies identified in recognition.

4.8 [Introduction to image compression with NeuroMem](#)



This example illustrates the use of an unsupervised learning method to build a codebook of reference codes over one or multiple images, and reconstruct simpler images by replacing the original blocks by their closest matches.

1. Click Browse in the Navigation toolbar and load the first image stored in Images/MiscBWImages folder
2. Select Load project in the File menu and load the file compression.prj stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is empty. The size of the region to learn and recognize is 9x9 pixels and the feature extraction is a monochrome subsample includes the color information.

3. Select the “Codebook” learning mode to build the knowledge and click the Learn button. Using the Maxif of 2000, approximately 50 neurons are committed.
4. Select the “MapOfModels” recognition mode, check the Auto checkbox and click the Reco button.
5. A simplified image based of only 50 patches is generated.
6. Go to the next image by clicking at the “>” button in the navigation bar and observe the same compression effect.
7. If blocks appear in black in an image, click Learn. A few neurons should be added to the knowledge. They correspond to the blocks not recognized in the previous codebook.
8. You can practice with the same routine using smaller values of the maximum influence field and observe the differences in the number of committed neurons and the reduction of the error between the source and its Transform image.
9. You can practice with the same routine using smaller patches of pixels and observe a finer granularity and lesser error in the Transform images.
10. You can practice with the same routine using the images of the MiscColorImages folder. Change the feature to Color Subsample in the Options/Settings menu if you wish to build a codebook of color patches.

4.9 [Kanji Character recognition](#)



This example demonstrates that the NeuroMem network is (1) a good classifier which can easily discriminate between many models, and also (2) the recognition time is not impacted by the number of models which constitute the knowledge base (360 in this case).

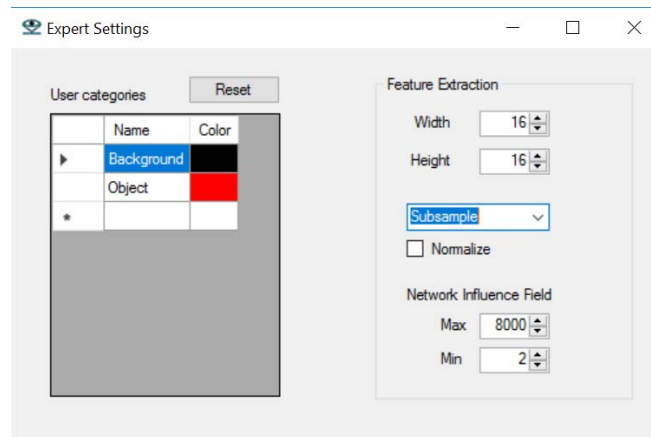
Warning! If using a simulation of the NeuroMem network, the recognition may take several seconds. On the small image, the window of interest is moved across 68,432 positions and at each position, the block of pixels (256 bytes) is compared with 360 models. Assuming that each comparison between an input pixel and the same pixel stored in a neuron takes 4 operations, the recognition of the entire image is equivalent to 25.2 billions of operations. The CM1K chip will execute this in 2 seconds or 12.5 Giga ops.

1. Click Browse in the Navigation toolbar and load the image stored in Images/Kanji folder
2. Select Load project in the File menu and load the file kanji360.prj stored in the Projects folder
 - a. In the status bar at the bottom of the screen, notice that the knowledge is already built and consists of 360 neurons. The size of the region used to learn and recognize the characters is 16x16 pixels. The feature extraction with the value 0 corresponds to a monochrome subsample.
3. Select Options/Overview at Cursor. Move the cursor over a character and view the result of the recognition in the Cursor Info box of the main panel. Note that positive recognition of a character only occurs when well centered over it.
4. The knowledge can be displayed by clicking “View Knowledge” button. Note that in this case, each character is represented by 3 models shifted by one pixel each. This training was done intentionally to allow the detection of a character at different offsets.
5. Click the Reco button to inspect the entire image. Note that this can take a few seconds in simulation, since the image is entirely scanned with a step of 1 pixel per column and 2 pixels per row, and at each position the block of 16x16 pixels is matched against 360 models.
6. Review the results in the Options/Results window

5 Starting a new project

5.1 Define your categories and associated color tags

A new project starts with a single generic category labeled “Object”, the other option being the “Background” category or “null” category. Names can be changed and validated by typing the Enter key. Depending on the colors of your images, you can change the colors which will be used to highlight your objects. For example, if your image has a red background, you may want to select a blue or green color to highlight your objects. Click at the color cell and select a new color in the color wheel.



Examples of categories for a part inspection

- Acceptable (category 1)
- Recyclable (category 2)

Or

- Grade A (category 1)
- Grade B (category 2)
- Grade C (category 3)

Examples of categories for a traffic monitoring application:

- Car (category 1)
- Truck (category 2)
- Van (category 3)

5.2 Nominal sizing and Feature extraction

The Region Of Interest (ROI) is the smallest rectangular area including your object. It should be limited to the smallest discriminant portion of your object and exclude any unnecessary background area.

For example, if a mechanical part can be classified as good or defective based on the presence or absence of a screw at a given position in the part, the region of interest should not be defined as the whole part but rather limited to the rectangle where the screw is expected.

The feature extraction is the method used to convert the pixel data within the region of interest into a vector of 256 bytes ready for broadcast to the neurons:

Subsample	Monochrome subsample, or average intensity of up to 256 blocks fitting inside the ROI
Histogram	Grey histogram, or the number of pixels per 256 grey-level values.
HistoCumul	Grey histogram cumulative
SubsampleRGB	Color subsample, or average Red, Green and Blue intensities of up to 85 blocks fitting inside the ROI
HistogramRGB	Color histogram, or sequence of the red, green and blue histograms, each of 85 values.
HistoCumulRGB	Color histogram cumulative
Composite profile	Average of the lines of pixels and columns of pixels
Horizontal profile	Average of the lines of pixels
Vertical profile	Average of the columns of pixels

Regardless of the feature you select, you have the option to normalize the amplitude of the feature vector or not. For a better understanding of the “Normalize” effect, take a look at the OCR_inkjet project described later in this manual.

5.3 [Status bar](#)

Once you close the Settings panel, the categories are reported in the “Category” combo box and the other parameters in the Status bar at the bottom of the screen.

6 [The Learning modes & Tools](#)

6.1 [Supervised learning with annotations](#)

Select the “Annotations” learning mode

Option 1: Annotate an object

- Select the category of the object in the dropdown list above the Teaching panel
- Move the cursor over the object and Right Click with the mouse button
- To delete, the annotation, press Shift + Right click

Option 2: Annotate an area or surface

- Select the category of the area in the dropdown list above the Teaching panel
- Draw the area pressing down the left mouse button
- Select the stepx and stepy you want to use when learning an area
- To delete, the annotation, press Shift +Right click
- To reset to the full frame, press Shift + Left click

You can choose to learn at any time by clicking the Learn button, but it is recommended to first annotate all the regions of interest and learn them all at once.

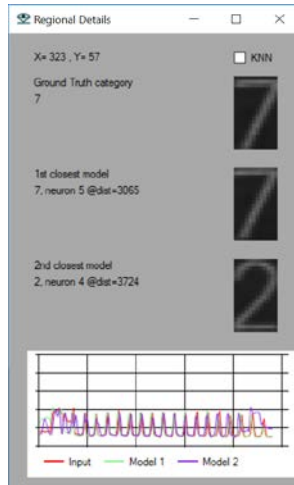
The annotations are automatically appended to a log file which can be saved from the Annotations menu.

If the “Auto” checkbox next to the Learn button is marked, the annotations loaded with an image are automatically learned. Combine this setting with the use of the Play mode in the navigation bar and you can learn automatically all the “annotated images within a given folder.

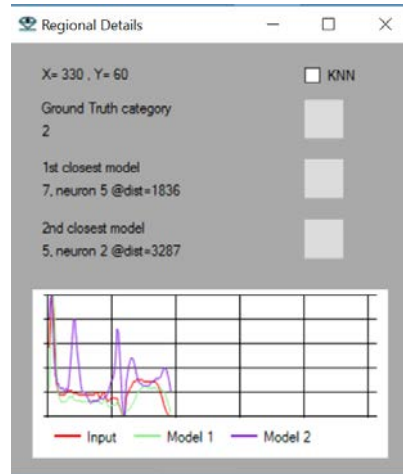
6.2 [View Local Classification](#)

To help understand the recognition at specific locations, select Options/ Details at Cursor. As you move the cursor, the feature extracted from its location is displayed and, if applicable, the models of the first and second closest firing neurons along with the distance between the feature and their models. This utility can be useful to correct mis-firing neurons which recognize a feature vector with an incorrect category at a specific location.

Report of subsample vectors



Report of composite profile vectors



6.3 [View Knowledge Content](#)

The View Knowledge menu allows reviewing the contents of the neurons, the pattern they hold, their category and active influence field (AIF). This utility can be helpful to understand why some erroneous or uncertain classifications occur.

The AIF is a dynamic attribute entirely controlled by the neuron itself. From the time the neuron is created, its influence field can be reduced as new and contradicting examples are taught.

NeuronID	Category	Model	Active IF	Degenerated
1	cloud		2686	<input type="checkbox"/>
2	grapes		12268	<input type="checkbox"/>
3	soil		6649	<input type="checkbox"/>
4	soil		6948	<input type="checkbox"/>
5	soil		8322	<input type="checkbox"/>
6	sky		8936	<input type="checkbox"/>
7	soil		8993	<input type="checkbox"/>
8	soil		7395	<input type="checkbox"/>

6.4 [Unsupervised learning with Codebook generation](#)

Select a Region to Learn with the mouse cursor in the “Learning” panel, or Shift click to reset to the full image. Let the neurons decide which patches of pixels represent novelty.

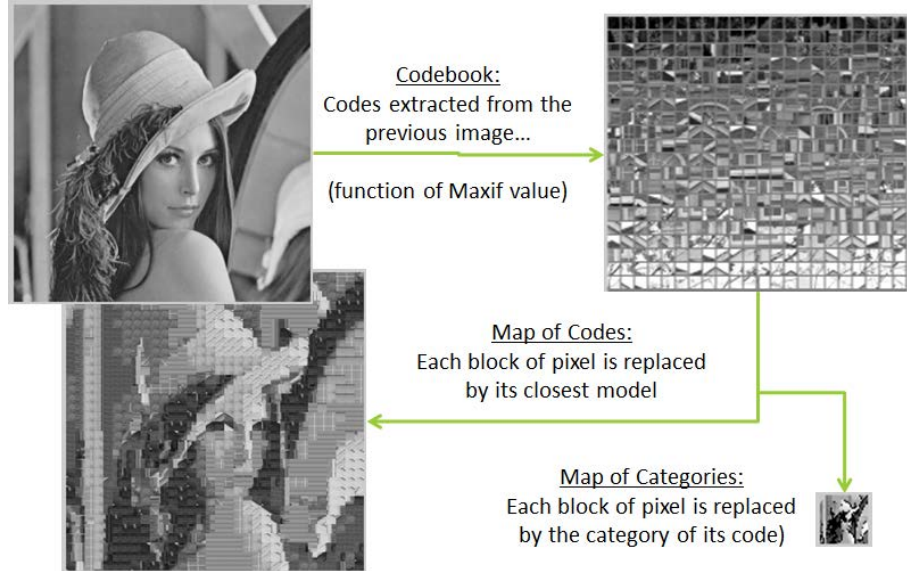
The knowledge or contents of the neurons is then called a codebook, or a book of codes (or features) which the neurons have learned and detected as significant.

The codebook can be used to generate simpler transform images such as a map of models, a map of code IDs, a map of confidence, etc.

Important parameter for this learning mode:

The smaller the Maxif, the more detailed the resulting codebook, but evidently the less generalization.

You can edit the Maxif under the Project/Settings menu.



6.5 [Unsupervised learning of a Golden Template](#)

Select a Region to Learn with the mouse cursor in the “Learning” panel, or Shift click to reset to the full image. Let the neurons build a codebook and an associated list of codes describing the golden template image.

The use of this learning method is only relevant if you wish to execute a template matching recognition. The latter will mark the codes in a new image which do not match the list of codes from the golden template. It is an anomaly detection considering both the recognition of a code, but also its proper location to match the golden template.

Important parameter for this learning mode:

The smaller the Maxif, the more detailed the resulting codebook, but evidently the less generalization.

You can edit the Maxif under the Project/Settings menu.

Important remark:

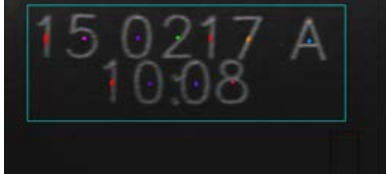


In this current version of IKB, you can only build a knowledge referring to a single golden template.

7 The Recognition modes and Tools

The recognition is executed by moving the ROI over the ROS using a raster displacement and a given step along the horizontal and vertical axis.

The IKB can format the results of the scanning with different format:

- Clusters
- Anomalies
- Map of the categories

Objects, or all locations with a positive recognition (given the sampling steps)	Clusters, or groups of locations recognized with a same category (given the sampling steps)	Transform image reporting the recognition at all scanned positions (the unknown is shown with the color assigned to the category 0 in the Settings; white in this example)
		

IKB-Pro only

- Map of the closest models (useful when the neurons contain a codebook)
- Map of the distances (inversely proportional to the confidence of the neurons)
- Map of the neuron identifiers

7.1 [Coping with change of scale](#)

Alternative #1:

- Learn at a given ROI size (as specified in the Options/Settings menu)
- In recognition, the image scale can be changed in the navigation bar at the bottom of the screen.

The project faces_soccerplayer is a good illustration of this approach: the ROI has been sized to learn on the images of the soccer players. However, if you load an image from the folder faces_Ferret, you can notice that the ROI size is too small, but if you scale down the image by 3, then it becomes suitable and the face is recognized. Refer to manual paragraph 4.2.3.

Alternative #2:

- If the feature extraction can be a subsampling, it is scale invariant if the number of blocks within the ROI remain the same. It is then possible to learn objects at different compatible ROI sizes (as specified in the Options/Settings menu). For example, an ROI of 64x64 with blocks of 4x4 creates a vector of 16x16 components, but similarly an ROI of 128x128 with blocks of 8x8 also creates a vector of 16x16 components.
- In recognition, the neurons holding models of the objects at the proper scale will fire.

8 Annotation and Result files

Annotations are generated and appended to a Log file whenever you click the Learn button under the “Annotations” learning mode. The file can be cleared, saved or loaded from the Annotations menu.

Results are generated and appended to a Log file when you click the Recognize button and if the selected Recognition mode is not a map style. The file can be cleared, saved or loaded from the Annotations menu.

The Annotations and Results files have the same format which is described in the next paragraph.

They are saved in the same folder as the last processed image. As a consequence, you will be prompted with the option to save your annotations or results files whenever you select a new folder of images or the image source type.

8.1 Common format

Annotations and Results both have the same format described below. They can be either comma delimited or tab delimited.

A header listing a version number, the labels associated to the category values and a listing of the colors associated to the category values, followed by one line item per annotation:

```
string sourceFile; // image file name, movie file name
long indexFile = -1; // a user defined index, a frame number in case of movie file, etc.
decimal Scale = 1; // scale at which annotation is edited
int Type = 0; // 0=object, 1=area
int Left = 0;
int Top = 0;
int Width = 0;
int Height = 0;
int Category = 1;
```

<u>Example of GT annotation file</u>	<u>Example of Results file</u>
Version, 0317	Version, 0317
<i>CatNbr,4</i>	<i>CatNbr,4</i>
<i>Nobody</i>	<i>Nobody</i>
<i>John</i>	<i>John</i>
<i>Debbie</i>	<i>Debbie</i>
<i>Thomas</i>	<i>Thomas</i>
<i>CatColors,4</i>	<i>CatColors,4</i>
<i>0,0,0</i>	<i>0,0,0</i>
<i>255,0,0</i>	<i>255,0,0</i>
<i>0,255,0</i>	<i>0,255,0</i>
<i>0,0,255</i>	<i>0,0,255</i>
Annotations,10	Results,10
Im_110217A_1008.jpg,0,1,37,28,36,60,1	Im_110217A_1008.jpg,0,1,37,28,36,60,1
Im_110217A_1008.jpg,0,1,88,28,36,60,5	Im_110217A_1008.jpg,0,1,88,28,36,60,5
Im_110217A_1009.jpg,0,1,160,28,36,60,10	Im_110217A_1009.jpg,0,1,160,28,36,60,10
Im_110217A_1016.jpg,0,1,388,32,36,60,11	Im_110217A_1016.jpg,0,1,388,32,36,60,11
Im_110217A_1016.jpg,0,1,286,86,36,60,8	Im_110217A_1016.jpg,0,1,286,86,36,60,8
Im_110217A_1016.jpg,0,1,126,99,36,60,0	Im_110217A_1016.jpg,0,1,126,99,36,60,0
Im_110217A_1016.jpg,0,1,37,21,36,60,0	Im_110217A_1016.jpg,0,1,37,21,36,60,0
Im_110217A_1026.jpg,0,1,315,89,36,60,6	Im_110217A_1026.jpg,0,1,315,89,36,60,6

8.2 [Importing annotations](#)

You may import annotations files generated outside of the IKB if they verify the following format:

- Must be comma delimited format
- 1st line = "Version, 0317"
- 2nd line = "Annotations" followed by the number of annotation appended in the file

Next lines are the annotations which can have different formats depending on the version of the IKB which generated them originally:

- Annotations accumulated over several images or movie files:
 - Sequence of 9 items: Image name, Index, Scale, Type, Left, Top, Width, Height, Category
 - Sequence of 8 items: Image name, Index, Scale, Left, Top, Width, Height, Category
- Annotations within a same image:
 - Sequence of 6 items: Index, Scale, Left, Top, Width, Height, Category
- Annotations within a same movie file
 - Sequence of 7 items: Image name, Index, Scale, Left, Top, Width, Height, Category

9 [Project and Knowledge files](#)

Projects defined under Image_Knowledge_Builder can be saved to disk in a project file or to the Flash memory of the camera.

A project is composed of two files associated to the same rootname:

- A project file (*.csp) containing the description of the recognition engine
- A settings file (*.prf) with the description of the GUI preferences

9.1 [CogniSight Project file](#)

- The *.csp file describes the recognition engine. It contains all the information necessary to resume the recognition of the images on an embedded system including the knowledge, the region of interest, the region of search and sensor settings

9.2 [Preference file](#)

- *_settings.csv describes the settings of the graphical user interface including the names and colors associated to the category values, as well as the preferred learning and recording settings.

9.3 [Special Projects](#)

9.3.1 *Default.csp*

At program startup, the application looks for a project named default.csp in the Projects folder. When found, this project is automatically loaded.

9.3.2 *Last.csp*

Whenever a Learn operation is executed, the application backs up the current project under the name last.csp in the Projects folder.

10 Appendix A: IKB version

	Access	IKB Trial	IKB	IKB Pro
Image sources and navigation				
Load image file	Navigation toolbar	x	x	x
Load movie file	Navigation toolbar			x
Navigate through folder of the last selected image	Navigation toolbar	x	x	x
Navigate through frames of the last selected video	Navigation toolbar			x
Play/Pause	Navigation toolbar	x	x	x
Pre-processing				
Image scaling	Main menu	x	x	x
Learning functions and utilities				
Selection of global network settings	Project/Settings	x	x	x
Choice of feature extractions on color or monochrome image	Project//Settings	x	x	x
Choice of feature extractions with normalization on/off	Project/Settings	x	x	x
Choice of scanning steps for learning of areas (annotated areas, codebook, golden template)	Main/Step control	x	x	x
Teach object with annotations	Main/Learn button	x	x	x
Teach area or surface with annotations	Main/Learn button	x	x	x
Build codebook from an ROS	Main/Learn button		x	x
Learn golden template from an ROS	Main/Learn button			x
Find saliencies in image	Main/Learn button			x
Load image to neurons	Main/Learn button			x
Learn an image obtaining categories from an 8-bit GT image	Main/Learn button			x
Report committed neurons	Main/Status bar	x	x	x
Display feature extraction at cursor location	Main/Status bar	x	x	x
Undo last learning	Knowledge/Undo menu	x	x	x
Auto checkbox to learn each time a new image is loaded	Main/Auto checkbox	x	x	x
Load Ground truth image	Options/Load GT image			x
Image annotations				
User defined category names and color tags	Project/Settings	x	x	x
Annotate an object	Right Click	x	x	x
Annotate an area	Left Draw	x	x	x
Delete annotation (object and area)	Shift Right Click	x	x	x
Reset area to full frame	Shift Left Click	x	x	x
View all or filtered annotations	Annotations/View	x	x	x

Review list of annotations	Annotations menu	x	x	x
Save annotations	Annotations menu		x	x
Automatic counter examples				x
Automatic adjacent examples				x
Local recognition functions and utilities				
Display feature vector at cursor position	Options/Cursor Info	x	x	x
Report GT and recognized category, if applicable	Options/Cursor Info	x	x	x
Report feature vector, firing models and ground truth, if applicable	Results/Details at cursor	x	x	x
Image recognition functions and outputs				
Select Region of Search on screen (ROS)	Main	x	x	x
Choice of scanning steps to screen the ROS	Main/Step control	x	x	x
Choice to skip the next step if recognized	Main/Skip box	x	x	x
Report recognized objects with their category color tags	Main/Reco button	x	x	x
Report clusters of recognized objects with their category color tags	Main/Reco button	x	x	x
Report anomaly locations	Main/Reco button or Auto checkbox	x	x	x
Transform blocks of image with closest match	Main/Reco button or Auto checkbox	x	x	x
Find differences with a golden template	Main/Reco button or Auto checkbox			x
Generate map of categories	Main/Reco button			x
Generate map of confidence	Main/Reco button			x
Generate map of identifiers	Main/Reco button			x
Auto checkbox to recognize each time a new image is loaded	Main/Auto checkbox	x	x	x
Recognition of multiple Region of Search				x
Image Results				
Save a map image	Main/File/Save Transform menu			x
View all or filtered results	Results/View	x	x	x
Save results	Results/View		x	x
Snap to the closest recognized object	Main/Mouse key	x	x	x
Knowledge display and file management				
Display all models in a thumbnail	Knowledge/View	x	x	x
Display models per category in a thumbnail	Knowledge/View	x	x	x
Display all neurons' content in a table	Knowledge/View	x	x	x
Clear all neurons	Knowledge/Clear	x	x	x
Restore the knowledge as it was before the last learning	Knowledge/Undo	x	x	x

Project file management				
Save to /Restore from file	Main/File menu		x	x
Choice of the neuron platform	Knowledge/Platform menu		x	x
Examples with Installer				
Example BGA	Main/File menu	x	x	x
Example Inkjet OCR	Main/File menu	x	x	x
Example Kanji	Main/File menu	x	x	x
Example Face Reco on soccer teams & Ferret	Main/File menu	x	x	x
Example glass surface anomaly detection	Main/File menu	x	x	x
Example wafer inspection w. template matching	Main/File menu			x

11 Appendix B: What is new?

11.1 [IKB Version 2.5.4](#)

The program can load annotations saved in comma delimited and tab delimited format. The annotations loaded from file are automatically converted to the current scale selected on the main panel. When scaling an image, the location of existing annotations and/or results are updated accordingly. Their size is not since this is related to the project settings.

Automatic option to fit the region of search to the entire image

11.2 [IKB version 2.5.1](#)

Fixed bug preventing to edit steps and scales with more than 1 digit

Addition of a Settings button to access the project settings directly from the main panel

Addition of 3 display options to report the recognized objects and anomalies: a single point, a box, a shade

Bug fix and Improved display of the maps of the categories, distances, and identifiers: The maps are shown with their true dimension in pixel after the decimation by the selected scanning steps. The maps are pasted into a background image to clearly show their position relative to the original image.

The outline of the region of search is always visible even if the region is set to the entire image. This prevents the mis-interpretation of results due to an incorrect ROS selection.

11.3 [IKB version 2.4](#)

The Arrow cursors can be used to move the Region of Search in the recognition window.

The Details at Cursor panel has been improved so it can report more than the first 2 firing neurons if application. The plot display has been improved.

The last project can be retrieved in the C:\MyDocuments\General Vision\Project folder. This project is saved automatically before each learning operation.

The learning mode through Annotations now supports the annotations of single objects (ROI, Region of Search) but also areas (ROS, Region of Scan). The key controls have slightly been changed as described under the [Learning and Utilities](#) paragraph.

11.4 [IKB version 2.3](#)

- Menus have been re-arranged
- Data logging functionalities have been added
- Fix the bug with the Knowledge\Undo last learning command
- The Settings panel allows to edit the Minimum Influence Field
- The Auto-Learn check box is active and triggers the automatic learning of the annotations associated to the newly loaded image.

- The Auto-Reco check box is active and triggers the automatic savings of the recognized objects (or clusters or anomalies) to a "results" file (see below).
- Annotations are now appended in a single file which can be saved and reviewed in a new panel. This file must reside in the same folder as the images it relates to.
- Results are now appended in a single file which can be saved and reviewed in a new panel. This functionality replaces the former "View Results" menu.
- The feature vectors extracted from annotations and/or results can be exported in a csv format compatible with NeuroMem Knowledge Builder

12 Troubleshooting

12.1 [Unable to load DLL "CogniSight_xxx.dll"](#)

Download [Visual C++ Redistributable for Visual Studio 2015](#) from the Microsoft Download Center.

Apply (execute) both files:

vc_redist.x86

vc_redist.x64