

V1KU-CS-SDK

Image Recognition Development kit
for the V1KU board

Version 1.0



Distributed by General Vision Inc.

Revised Oct 2009

License Agreement

This SDK is published by General Vision

This License Agreement is a legal agreement between you (LICENSEE) and General Vision Ltd 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' 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.

Distribution

The distribution of an application written with the SOFTWARE PRODUCT requires the purchase of one run-time license per system.

Contact

www.General-Vision.com

Table of Contents

LICENSE AGREEMENT	1
TABLE OF CONTENTS	2
4	
PACKAGE CONTENTS	4
V1KU DRIVERS.....	4
BIN FOLDER.....	4
EXAMPLES IN C++	4
EXAMPLES IN VB6	4
EXAMPLES IN C#.....	4
GETTING STARTED WITH THE SDK	5
5 SIMPLE COMPONENTS.....	5
<i>Image Frame, the memory space where the image is stored</i>	5
<i>ROI, the Region of Interest to learn and recognize</i>	5
<i>KN, the Knowledge</i>	6
<i>ROS, the Region of Search</i>	6
<i>VO, the Visual Objects identified by the engine</i>	7
OVERVIEW.....	8
CLASSES AND FUNCTIONS	9
IMAGE CLASS AND FUNCTIONS	9
<i>GetImageInfo (word Width, word Height)</i>	9
<i>LoadJPEG (char * Filename)</i>	9
<i>Display(HDC ImageHDC, HBITMAP ImageHWnd)</i>	9
<i>SaveJPEG (char * Filename, word Quality)</i>	9
ROI CLASS AND FUNCTIONS	9
<i>Init</i>	9
<i>MoveROI (word Left, word Top)</i>	9
<i>Set (word *Width, word *Height, word BlockWidth, word BlockHeight)</i>	9
<i>Get(word Left, word Top, word *Width, word *Height, word BWidth, word BHeight)</i>	9
<i>Size (word *Width, word *Height, word KeepRatio)</i>	10
<i>Learn (word Category)</i>	10
<i>Recognize (word *Distance, word *Category)</i>	10
<i>ReadFeature</i>	10
ROS CLASS AND FUNCTIONS	10
<i>Init</i>	10
<i>Search(word Mode)</i>	10
<i>ReadObject(word Index, word X, word Y, word Distance, word Category)</i>	10
<i>SetArea (word Left, word Top, word Width, word Height)</i>	11
<i>GetArea (word Left, word Top, word Width, word Height)</i>	11
<i>SetScanning (word StepX, word StepY)</i>	11
<i>GetObjectCoverage(word Left, word Top, word Width, word Height)</i>	11
KNOWLEDGE CLASS AND FUNCTIONS.....	12
<i>Init</i>	12
<i>Load (char *Filename)</i>	12
<i>Save (char *Filename)</i>	12
<i>ReadModel (word *NeuronID)</i>	12
<i>ReadNeuron (word *NeuronID, NEURON *Neuron)</i>	12
<i>WriteToFlash</i>	12
<i>ReadFromFlash</i>	12
SENSOR CLASS AND FUNCTIONS	13
<i>Connect (byte DeviceID)</i>	13

<i>Grab</i>	13
<i>ReadImage</i>	13
<i>WriteImage</i>	13
<i>SetSensor(word Gain, word Shutter, word Trigger)</i>	13

Package contents

V1KU drivers

The USB drivers of the V1KU sensor board are installed in C:\Windows\System32\CyLoad by the Diagnostics utility supplied with the board. They are composed of the following two files: CyUSB.sys and GV_USB.inf.

Bin folder

Depending on your development platform, you can interface to the V1KU board through the V1KU_CS.dll or the V1KU_CSnet.dll. Both DLLs are very similar in term of data structure and function calls. They have different dependencies which are presented below.

V1KU_CS.dll

This DLL is intended for the standard Windows development environment such as C, C++ and Visual Basic. It comes with associated library and header files:

- V1KU_CS.lib, library to link to your project
- V1KU_CS.h
- CyAPi.lib (Cypress usb)
- Jpeglib

V1KU_CSnet.dll

This DLL is intended for the .NET development environment such as VB_Net and C#. It comes with associated DLLs:

- V1KU_BASICnet.dll
- CYUSB.dll

Examples in C++

These examples use the V1KU_CS.dll (to copy in C:\Windows\System32) and its associated library and header file.

- Simple vector recognition
- Simple video recognition example requiring the V1KU connectivity

Examples in VB6

These examples use the V1KU_CS.dll (to copy in C:\Windows\System32) and its associated Visual Basic modules (*.bas) stored in a Common folder.

- Common folder including *.bas
- Simple vector recognition example
- Simple image recognition example

Examples in C#

Simple vector recognition example

Simple video recognition example

Getting started with the SDK

5 simple components

Learning examples selected in images triggers the automatic model generator built into the CogniMem neurons. Their contents produces a knowledge which can be saved and displayed.

Reading the response of the CogniMem neurons over many different positions in the image produces a map of “recognized” Visual Objects which can be further studied and consolidated to produce a final decision.

The V1KU_CS Development Kit lets you perform these operations with minimum efforts by manipulating five simple components:

- Image
- ROI or Region of Interest
- KN, or Knowledge
- ROS, or Region of Search
- VO, or Visual Objects

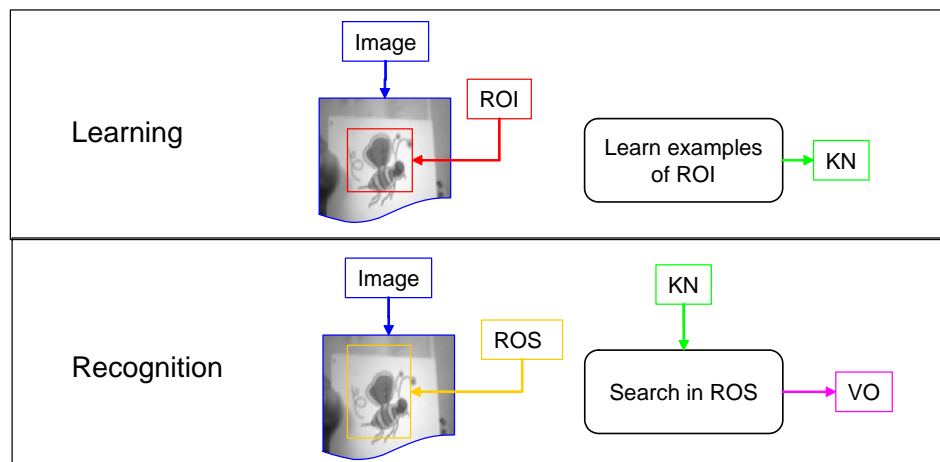


Image Frame, the memory space where the image is stored

- The V1KU board is capable of acquiring live video frame coming from its Aptina monochrome sensor and it can also receive image files from a host. In both cases, the pixel data is stored to a memory page of the on-board SDRAM. There is no limit to the size of the images which can be loaded from file as long as they occupy less than 16 MB.
-
- Rev3 of the CogniSight engine is presently limited to the extraction of a grey level signature from image. As a consequence, the WriteImage function converts a color image to a grey-level image at the time of the transfer.

ROI, the Region of Interest to learn and recognize

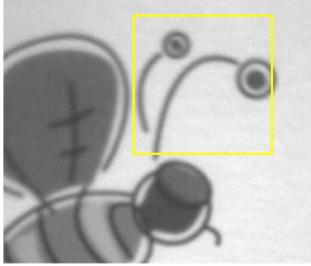



The Region Of Interest (ROI) is the smallest rectangular area including your object. It is used by the CogniSight engine to extract a signature which becomes the pattern vector learned or recognized by the CogniMem neurons.

The ROI is defined by a standard set of coordinates including Left and Top position and a Width and Height. It also has two additional attributes called BlockWidth and BlockHeight used by the CogniSight

engine to extract the signature vector of the ROI. This size of block defines the internal geometry of the ROI and how it can be divided into up to 256 basic groups of pixels

Example of different signature extracted from a same ROI size:

The three configurations shown below feature a geometry of 16x16 blocks. The total number of blocks amounts to 256 which makes the maximum usage of the neurons' memory capacity. The configurations 2 and 3 use non square blocks which produces a blurry effects in the case of this particular example, but can be of interest if the pattern to detect has a specific orientation.

Source image and ROI size	Feature extracted with a geometry of ...		
	16x16 blocks of 8x8 pixels 	16x16 blocks of 4x16 pixels 	16x16 blocks of 16x4 pixels 

- REMARK1: The Size function of the SDK calculates automatically an optimized distribution of square blocks in the ROI, so you do not have to worry about it.
- REMARK2: If you intend to control your own block size, you must ensure that their total number inside the ROI does not exceed 256.

It is recommended that the ROI be limited to the smallest area including the discriminant portion of your object and that it excludes any unnecessary background area.

KN, the Knowledge

The knowledge is built by the neurons as they learn examples. It is defined by their contents which include the learned vector data, its category, a context value and influence field at the time the knowledge is saved. Additional training can be done at a later time, thus expanding the knowledge base.

If an application uses a single definition of ROI, the committed neurons hold a pattern which represent a same signature. The knowledge simply has to be associated to the definition of the ROI used to teach the examples.

Rev1 of this SDK is intended for applications dealing with one ROI at a time, but it will evolve to support multiple definitions of ROIs. In this case, the neurons must be assigned a different context per type of ROI prior to being taught. The knowledge will then be associated to a series of ROI definitions.

ROS, the Region of Search

The Region Of Search (ROS) is a rectangular area in which the CogniSight engine will look for known objects. The ROS can be the entire image.

The ROS is describes by the following attributes

- Size (up to the full frame, must be greater than or equal to ROI)
- Position (ROS must be included in the image frame)

The Search function of the CogniSight engine moves the ROI over the ROS in a raster displacement based on a user-defined step X and Y and such that the ROI never extends outside the ROS. At each displacement, a signature vector is extracted and sent to the neurons for recognition. If the vector is recognized, the engine appends the results to the list of VO (see below).



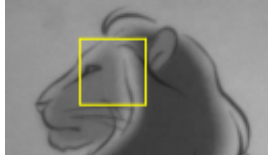

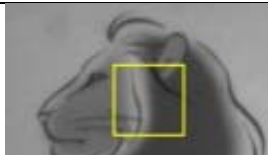

VO, the Visual Objects identified by the engine

The attributes of the Visual Objects are the following:

- Context, identifier of the context in which the object is recognized (links to the definition of the ROI and the neurons holding examples of this ROI)
- X,Y, center position of the object
- Distance and Category, outputs of the neuron with the best match
- Recognition status:0, unknown; 4, uncertain; 8, identified

*Distance refers to the distance between the input vector and the vector stored in the neuron. The distance and category values are read consecutively. A distance 0 means an exact match. The higher the distance, the farther the match between the input pattern and the pattern stored in the firing neuron. A Distance value equal to 65535 or 0xFFFF means that the input pattern is not recognized by any neuron and the associated category is therefore null.

Overview

Description		Instructions and Observations
Size and position the ROI to include the eye of the lion and learn as category 1		SizeROI MoveROI LearnROI (1) Returns Neurons=1
Recognize the ROI without moving it		RecognizeROI returns Distance =0 (exact match) Category=1
Move the ROI a few pixels away and recognize		RecognizeROI returns Distance =1892 Category=1
Move the ROI over the ear of the lion and learn as category 2.		MoveROI LearnROI (2) Returns Neurons=2
Move the ROI in between the eye and the ear of the lion.		RecognizeROI returns Distance =-1 (unknown response) Category=0
Overview of the recognized objects		RecognizeROS Returns ObjectCount= 55 ReadObject 55 times and annotate in image

Classes and Functions

Image Class and Functions

This class is the one that differs the most between the standard and the .Net DLLs. In the .Net environment the SDK takes advantage of the built-in bitmap file management of the .Net platform and therefore provides for very simple access functions such as LoadFromSensor and LoadFromFile returning a bitmap image ready for display.

GetImageInfo (word Width, word Height)

Read the spatial resolution of the last image loaded in memory either through a LoadFromFile function or through a Grab function.

LoadJPEG (char * Filename)

Std DLL only

Load an image saved in a JPEG file.

Display(HDC ImageHDC, HBITMAP ImageHWnd)

Std DLL only

Display the image pointed by the current image descriptor in a picture box.

SaveJPEG (char * Filename, word Quality)

Std DLL only

Save the image stored in the frame buffer to a JPEG file. Filename is the path of the destination file. Quality is a compression factor and can range between 0 and 100. Default is 70.

ROI Class and Functions

Init

Set the ROI to the default values which are Left=200, Top=120, Width=340; Height=220, BlockWidth=20 and BlockHeight=20;

MoveROI (word Left, word Top)

Move the region of interest so its upper-left corner becomes the pixel (Left,Top).

Set (word *Width, word *Height, word BlockWidth, word BlockHeight)

Set the attributes of the ROI to the input values. Width and Height are rounded up if necessary to be a multiple of the BlockWidth and BlockHeight. The function returns an error code 0 if successful. An error code of 1 indicates that there are more than 256 blocks inside the ROI which is not compatible with the feature extraction of the CogniSight engine. In this case, the Size function will adjust the block size so the ROI fits no more than 256.

Get(word Left, word Top, word *Width, word *Height, word BWidth, word BHeight)

Read the attributes of the ROI.

Size (word *Width, word *Height, word KeepRatio)

Size the region of interest to the specified Width and Height and calculate automatically the best block size. Width and Height are rounded up if necessary to be a multiple of the BWidth and BHeight.

- If KeepRatio=1, the function keeps the same distribution of blocks inside the region, thus ensuring consistency with an existing knowledge built with an ROI using the same aspect ratio.
- If KeepRatio=0, the function finds the square block such that the region can fit as many as 256 of them, thus using the maximum capacity of the neurons' memory.

The function returns an error code 0 if successful. An error code of 1 indicates that the input Width or Height have invalid values.

Learn (word Category)

Learn the current region of interest as belonging to the category Category and return the number of committed neurons after this operation.

Recognize (word *Distance, word *Category)

Recognize the current region of interest and reads the Distance and Category of the neuron with the best match, if any. The function returns the Recognition Status value (0=unknown, 4=uncertain, 8=identified)..

ReadFeature

Read the signature vector extracted from the current region of interest and returns its length.

Std DLL	Returns the length of the vector and a pointer to the vector data
Net DLL	Return a bitmap with a resolution equal to the number of blocks along the vertical and horizontal axis

ROS Class and Functions

Init

Set the ROS to the default values which are Left=0, Top=0, Width=752; Height=480, StepX=16 and StepY=16.

Search(word Mode)

Move the ROI over the ROS using a step increment and for each position append the recognized object, if applicable, to the VO list. The function returns the number of objects stored in the VO list.

The Mode input specifies which type of recognition to report:

Mode=0, all recognized objects (identified or uncertain)

Mode=1, all non recognized objects

Mode=2, recognized objects without uncertainty

Mode=4, recognized objects with uncertainty

ReadObject(word Index, word X, word Y, word Distance, word Category)

Read the information of the object number Index in the list of VO generated by the last Search function.

SetArea (word Left, word Top, word Width, word Height)

Define the position and size of the ROS.

GetArea (word Left, word Top, word Width, word Height)

Define the position and size of the ROS.

SetScanning (word StepX, word StepY)

Define the length of the scanning step along the horizontal and vertical axis to move the ROI over the ROS. If only one step is specified the value applies to the displacement along the two axes.

GetObjectCoverage(word Left, word Top, word Width, word Height)

NET DII Only

Return the coordinates of the rectangle including all the objects of the VO list. This function can be of interest to adapt the size of the ROS programmatically.

Knowledge Class and Functions

Init

Clear the contents of the neurons including registers and memory.

Load (char *Filename)

Load from file the definition of the ROI and the contents of the neurons. This function returns the number of committed neurons.

Save (char *Filename)

Save the definition of the ROI and the contents of the neurons to a file. The default extension is *.knf.

ReadModel (word *NeuronID)

NET only

Read the contents of a particular neuron indexed as NeuronID and retruns its bitmap image.

ReadNeuron (word *NeuronID, NEURON *Neuron)

Read the contents of a particular neuron indexed as NeuronID and stores its registers in the Neuron structure described below. This function is useful to review the knowledge stored in the neurons for display and analysis purposes.

Neuron

```
{
  byte Context..... Context of the neuron (bit 6-0 = context value, bit 7 = Norm)
  byte Vector[MAXVECLENGTH]  Vector stored in the neuron, up to 256 components of 8-bit
  word AIF ..... Active Influence Field
  word Identifier ..... Identifier
  word Category ..... Category
}
```

WriteToFlash

Save the knowledge to the Flash memory of the V1KU board.

ReadFromFlash

Read the knowledge from the Flash memory of the V1KU board.

Sensor Class and Functions

Connect (*byte DeviceID*)

Executing this function is necessary to start using the V1KU board.

Std DLL DeviceID =0 for simulation, DeviceID =1 for V1KU interface via USB
Net DLL DeviceID= identification number of the V1KU (default = 0)

Grab

Save the next frame to memory.

ReadImage

Transfer the image residing in the SDRAM of the V1KU board to the memory space allocated to the CogniSight engine on the PC host.

WriteImage

Transfer the image residing in the memory space allocated to the CogniSight engine on the PC host to the SDRAM of the V1KU board. The execution of this function updates the CS_FWIDTH and CS_FHEIGHT registers of the V1KU board accordingly.

SetSensor(*word Gain, word Shutter, word Trigger*)

Set the most three common video acquisition settings of the Micron sensor:

- Gain is the sensor analog gain. Its value ranges between 16 and 64. If Gain is set to 0, the function set the Auto Gain and Auto Exposure ON.
- Shutter is the total integration time of the sensor expressed in number of rows. Its value ranges between 1 and 480. This value multiplied by 30 gives the exposure time of the sensor in micro seconds. If Shutter is set to 0, the function set the Auto Gain and Auto Exposure ON.
- Trigger sets the Micron sensor in trigger mode. If this parameter is set to 0, the Micron sensor grabs video continuously.