

Introducing Cognex Deep Learning™

- VisionPro Deep Learning is Cognex's new PC-based deep learning software product. It is the direct successor to VisionPro ViDi™. The first release of VisionPro Deep Learning adds a new "High Detail" mode to the existing ViDi Classifier, which is now known as "Focused" mode. High Detail mode was originally developed as part of the SuaKIT™ product. It provides highly accurate results for a wide range of challenging applications, while Focused mode typically trains and runs faster. Users can switch between modes without relabeling the images. This makes it easy to determine which mode offers the best combination of accuracy & speed for any particular application.
- VisionPro Deep Learning is compatible with VisionPro ViDi, including API, project loading and security dongles. Some changes have been made to Cognex Designer® support. See [Using Runtime Workspace with Cognex Designer](#) topic for details.
- SuaKIT users can import SuaKIT classification project into VisionPro Deep Learning 1.0. (VisionPro Deep Learning 1.0 doesn't support the import of the other SuaKIT projects yet.). They can then take advantage of the other features of Cognex Deep Learning Studio, including the Blue Locate and OCR tools, tool chaining, and access to VisionPro® and Designer.

About This Release

[New Content]

- Two modes for the Green Classify Tool.
 - High Detail mode (from SuaKIT), which provides higher classification accuracy on many challenging applications.
 - Focused mode (from ViDi) for simpler applications that need faster training or runtime execution.
 - A Heat Map for High Detail mode, which visually indicates which portions of the image were most influential in the classification decision.
 - Validation sets for High Detail mode. These provide feedback on model performance and are used for training optimization.
 - You can enable/disable GPU memory optimization for use with High Detail mode and other tools.
- Simpler labeling controls, including
 - Selecting multiple samples when label or accepting views, clearing markings and labels, processing, or managing image sets.
 - Automatically re-using the last label name by default, rather than requiring you to explicitly type each new label.

[Deprecated features]

- "Set Group" Feature is not supported from this release.
- Changes in Designer functionality
 - Deep Learning Plugin that ships with Deep Learning 1.0 is fundamentally different than previous versions.
 - Training is not supported from within Designer. Instead, training needs to be done from the Cognex Deep Learning Studio, and the resulting runtime workspace loaded into Designer.
 - You can process images with Deep Learning Runtime Block. You can check more information on [Migrating to This Release](#) topic.
 - Deep Learning Runtime Block does not support an External Fixture.

Note:

- The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.

When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.



- Sample code is included with the VisionPro Deep Learning 1.0.0 installer. The resources for the sample code (*VisionPro Deep Learning 1.0 Sample Code Resource.msi*), which include images and workspaces, are available for download from the [Cognex Support Site](#).
- For more information about new API functionality in this release, see the *Cognex VisionPro Deep Learning Suite .NET Library* and *Cognex VisionPro Deep Learning Suite C API* help files.

Resolved Issues

This section describes changes and resolved issues in VisionPro Deep Learning 1.0.0.

- Resolved an issue with feature size indicator does not change after changing the value of the feature size in the sampling tool parameter.
- Resolved an issue with "Save new version" in the workspace list does not act in the Blue Read Tool.
- Resolved an issue with name of Set are shown as original name in the image viewer even though users change the name.

Migrating to This Release

This topic contains information about changes that may affect existing VisionPro Deep Learning projects when migrating projects from previous releases to the current release.

Architecture of Existing Green Classify Tool

When you open workspaces created from previous version (ViDi Suite) and if there is a Green Classify Tool, you would see as a Green Classify Tool in Focused mode. Existing architecture of Green Classify Tool is called Focused mode and a new architecture of Green Classify Tool is called High Detail mode.

Keeping Existing names of APIs

Since VisionPro Deep Learning inherits ViDi, it retains the name ViDi in many parts of the APIs

Internal version of APIs

Product version is 1.0.0 but, 5.0.0 is used for the version in APIs and binaries.

No longer supported or Deprecated APIs

GPUMode.MultipleDevicesPerTool is no longer supported. For more information, see [Breaking Changes to the API](#).

You can see deprecated APIs here([C and .NET API Deprecated and Removed Feature](#)).

Changes to Designer functionality

A project built with the 4.1 (or previous) version of the Designer Deep Learning Plugin is not compatible with the Cognex Deep Learning Plugin. To force the correct plugin to be used, use the Project Plugin Configurator in Designer's Utilities.

You cannot train in Designer, you can import Cognex Deep Learning Studio Runtime Workspace(.vrws file) created from Cognex Deep Learning Studio 1.0 and process the images.

The Runtime Block includes only one output pin. Individual Tools in the Tool Chain are not visible in the UI. And there are no preconfigured result pins so you need script for every single application.

Deep Learning Runtime Block does not support an External Fixture.

Past Releases - ViDi 3.4 to 4.1

Changes to the Viewing of Overlay Graphics

In the 3.4.0 and 4.1.0 releases, the keyboard shortcuts used to show and hide overlay graphics in the Image Display Area have been changed. The new keystrokes are the following:

- The *Alt* key plus the *Right Arrow* key or the *Alt* key plus the *Left Arrow* key

These are used to cycle through the available combinations of display graphics (labels, markings, labels and markings, or no overlay graphics).

In previous releases, the space bar could be used to show or hide all overlay graphics, but that is no longer supported.

Changes to Tool Training Workflow

In the 3.4.0 release, there is a new method for specifying the images/views that will be used to train a tool. If you will be re-training a tool that was trained in a previous release, you will need to configure the tool to use the new training set paradigm, via the Select Training Set dialog. Please refer to the Training Set and Use the Select Training Set Dialog topics.

Changes to the Feature Filter Syntax

In the 3.4.0 release, the syntax for the Feature Filter Processing tool parameter was modified. If you were using `width` and `height` in your expression, they must now be qualified with `view`, such as `view.width` and `view.height`.

For example, in version 3.2 and earlier, syntax used to limit the found features to those where the center of the feature lies within the ROI would have been the following:

```
x < width and x > 0 and y > 0 and y < height
```

In 3.4.0 and later, it must now be expressed as the following:

```
x < view.width and x > 0 and y > 0 and y < view.height
```

Note:

This syntax can also more easily be expressed by using the `in` operator. For example, the following syntax could be used to simplify the expression:

```
x in [0,view.width] and y in [0,view.height]
```

Known Issues

The following Known Issues have been identified in this release:

[Deep Learning Tools]

- **Multiple GPUs are not supported efficiently for High Detail mode.**
You cannot get an advantage of using multiple GPUs when you proceed training or processing the other tool specifically with Green Classify Tool in High Detail mode at the same time. For example, when training or processing job of Focused mode and High Detail mode are simultaneously being processed, each job could be slower than just training or processing one job at a time.
- **High Detail mode uses different resize algorithm to SuaKIT. (Green Classify Tool in High Detail mode only)**
To use imported SuaKIT model with VPDL runtime API, please use SuaKIT resize API before processing.

- **Once you are done with training High Detail mode of Green Classify Tool, you cannot change the name of the class.**
- **Processing results are different between GPU and CPU.**
The difference is in the second decimal place of the result. For example, When you process with CPU and GPU respectively, different results can be obtained, such as 0.80 and 0.81.
- **Defect regions do not extend to the left side of the view in Red Analyze Tool.**
To avoid this problem, when the defect area touch the left side of the image, please set 4 pixel more wider ROI than original image size. For example, if you have (x=0,y=0,w=100,h=100) image and it has detected area touch to left side of the image, please set (x=-4, y=0, w=104, h=100) ROI size.
- **When you set the smaller feature size (especially smaller than 10) and train the Red Analyze tool (both in Supervised and Unsupervised mode), the workspace is getting bigger.**
vzb file which is for saving heat map within the workspace is the main reason. : for example,
 - feature size=10, sampling density=3, 8192x819 image: 1 vzb file size=94MB
 - feature size=10, sampling density=10, 8192x8192 image: 1 vzb file size=1GB
 - feature size=40, sampling density=3, 8192x8192 image: 1 vzb file size=23KB

[Workspace]

- **If the tool is not saved well because of unspecific reasons, you cannot export or clone the workspace/tool.**
You can avoid this problem with processing the tools again and save the workspace.
- **Boost::filesystem::rename error**
This error occurs when the user or system repetitively save the workspace in a short time. This is just an alert that there was an issue while saving. You can solve this problem only with saving the workspace again.

[Big Workspace]

- **Especially with bigger than 10GB workspaces,**
 - Cloning work does not end when you clone tools in different big workspaces at the same time.
 - Some images are not included in the created report.

[Report]

- **During creating a report when you log off your PC, some of images are shown black.**
To avoid this issue, please do not log off your PC (turn off remote environment or lock screen) while you are creating a report.

[Plugins]

- **You can add same multiple plugins using Manage Plugins.**

[VisionPro Deep Learning Service]

- **MultipleDevicesPerTool is no longer supported for VisionPro Deep Learning. Therefore you will have “One or more error occurred” error, if you select “--gpu-mode =MultipleDevicesPerTool” for service.**
Please choose other option to run VisionPro Deep Learning Service.

[Deprecated features]

- **Group Feature is not supported from this release.**
 - If you import workspaces including "Group" feature which are made 4.1 and before, "Group" is still displayed in Cognex Deep Learning Studio 1.0, but you cannot modify this.
 - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio 1.0.

System Requirements

These topics cover the computer operating system and hardware requirements to use VisionPro VisionPro Deep Learning in your application.

Operating System Requirements

Microsoft® Windows® Operating System	English	Chinese	Japanese	Korean	French	German	Spanish
Windows 10 Professional (64-bit)	✓	✓	✓	✓	✓	✓	✓
Windows Server 2016	✓	✓	✓	✓	✓	✓	✓

Note:



- Windows Server 2016 is only supported for servers configured to use the Deep Learning client/server functionality. Configuring clients on the Windows Server 2016 OS is not supported.
- The use of the VisionPro Deep Learning software or the Deep Learning Service on a virtual machine (VM) is not supported.

Computer Hardware Requirements

When assembling PCs for your VisionPro Deep Learning application, refer to the following general guidelines for either a deployment or training PC (see the separate topics for specific requirements applicable to those configurations):

- CPU

We recommend Higher than Intel Core i7. When selecting a CPU, a higher CPU clock speed rate and multiple core processors will result in higher runtime tool execution. If your application will be relying on the Blue Locate tool, it is more sensitive to clock speed rates, particularly with complex model matching applications.

- System Memory (RAM)

We recommend 32GB or higher.

Graphics Processing Unit (GPU)

We recommend 11GB and higher GPU (1080 Ti, 2080 Ti) for training High Detail mode of Green Classify Tool.

For processing, 8GB and higher GPU is recommended for High Detail mode of Green Classify Tool.

When selecting a GPU, Cognex only supports NVIDIA GPUs.

-

- Power Supply

When selecting your power supply, include a 25% margin above the requirement to accommodate the system and GPU power requirements, i.e. select a power supply that is 1.25 times the system and GPU power requirements.

- USB 2.0

A USB port for a permanent connection to the Cognex Security Dongle via USB 2.0.

Note:

The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex

- ① Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.

When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.

Training PC Requirements

The following requirements describe the suggested system for the VisionPro Deep Learning application training and development PC.

- CPU

We recommend Higher than Intel Core i7. When selecting a CPU, a higher CPU clock speed rate and multiple core processors will result in higher runtime tool execution. If your application will be relying on the Blue Locate tool, it is more sensitive to clock speed rates, particularly with complex model matching applications.

GPU

When selecting a GPU, Cognex only supports NVIDIA GPUs.

-

We recommend 11GB and higher GPU (1080 Ti, 2080 Ti) for training especially for Green Classify Tool in High Detail mode.

- System Memory (RAM)

First, 32GB or more than 32GB is recommended.

When selecting the system RAM, specify the greater of:

- The sum of all GPU memory. For example, if you have four NVIDIA GeForce® RTX™ 2080 Ti GPUs, which each have 11 GB of memory, the PC should have 44 GB of RAM.
- The RAM should be one and a half times the typical workspace size. For example, if your typical workspace is 20 GB, the PC should have a minimum of 30 GB of RAM.

① **Note:** You should have more than 50GB to train the tool with 15,000 images of 1K X 1K.

- System Storage

Cognex recommends the use of a solid-state drive (SSD), we recommend 100GB + as a free space.

- USB 2.0

USB port for a permanent Cognex Security Dongle (containing the training license) connection via USB 2.0.

Note:

The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex

i Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.

When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.

Deployment PC Requirements

The following requirements describe the suggested system for the deployment of your VisionPro Deep Learning application on a runtime PC.

- CPU

We recommend Higher than Intel Core i7. When selecting a CPU, a higher CPU clock speed rate and multiple core processors will result in higher runtime tool execution. If your application will be relying on the Blue Locate tool, it is more sensitive to clock speed rates, particularly with complex model matching applications.

For Green Classify tool in High Detail mode, if you have only CPU; not GPU, you cannot process the tool.

If a single core performs multi-tasks, processing may be jumped up. Therefore, it is necessary to use a sufficient amount of resources or allocate the single task to a single core.

GPU

When selecting a GPU, Cognex only supports NVIDIA GPUs.

-

In addition, these GPUs offer longer-term availability and driver stability.

8 GB of GPU memory is recommended for runtime.

- PCIe Lanes

Cognex recommends a minimum of x8 PCIe lanes. However, a PCIe x16 has the potential to reduce cycle time by approximately 10 ms, relative to a PCIe x8 (based on a 5 MP image).

- USB 2.0

USB port for a permanent Cognex Security Dongle (containing the runtime license) connection via USB 2.0.

Note:

The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex

i Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.

When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.

NVIDIA® GPU Requirements

The following information covers the requirements when utilizing an NVIDIA GPU with your VisionPro Deep Learning application. The GPU is utilized by Deep Learning during the development of your application, typically with the training of tools. In addition, a GPU can also be used during runtime deployment, where it increase performance of runtime workspaces.

NVIDIA Driver Support

- NVIDIA GeForce® driver greater than or equal to 431.86 (Creator Ready)
- NVIDIA Quadro® driver greater than or equal to 431.94 (Optimal Driver for Enterprise)

Note: For more information about supported drivers, please visit the [Cognex Support Site](#).

Supported NVIDIA GPUs

VisionPro Deep Learning 1.0 aims to support the most common GPU families, but it's not practical to test every model or driver. The table below indicates those models we have explicitly tested and confirmed work, as well as those few models and families that do not work. Most other common NVIDIA GPUs other than very new models have been used successfully by customers, even if they haven't been explicitly tested. In addition, we share specifications for minimum GPU capability, as well as recommendations high performance GPUs for demanding applications, such as those using the new High Detail mode of the Green Classify Tool.

- Minimum recommended performance: GTX 1060 6GB
- Recommended for high performance: GTX 1080 Ti and RTX 2080 Ti

GPU	Model	Tested by Cognex	Recommended for High Performance
NVIDIA GeForce®	GTX 1060 6GB	Tested	
	GTX 1070		
	GTX 1080	Tested	
	GTX 1080 Ti	Tested	Recommended
	RTX™ 2070		
	RTX 2080		
	RTX 2080 Ti	Tested	Recommended
NVIDIA Quadro®	P2000		
	P4000	Tested	
	P5000		Recommended
	GV100*		Recommended
	RTX 4000		
	RTX 5000		Recommended
	RTX 6000		Recommended
	RTX 8000		Recommended
	Titan V*		Recommended
	Titan RTX		Recommended
NVIDIA Tesla®	V100*		

GPU	Model	Tested by Cognex	Recommended for High Performance
Models known not to work: <ul style="list-style-type: none"> <li data-bbox="174 281 1345 308">• GTX 1660 and GTX 1650: Please note that low performance and slow training time with unknown reason. 			
<p data-bbox="166 338 227 365">Note:</p> <p data-bbox="243 363 910 390">When selecting a GPU, Cognex only supports NVIDIA GPUs.</p> <ul style="list-style-type: none"> <li data-bbox="220 396 232 424">• <li data-bbox="220 438 1345 499">• VisionPro Deep Learning 1.0.0 supports the use of any NVIDIA CUDA® enabled GPU that provides a compute capability of 3.0 and higher, and requires a driver version greater than or equal to 431.86. <li data-bbox="220 520 1387 615">• CUDA compute capability 3.0 or higher is required to use the core VisionPro Deep Learning functionality. The Low Precision parameter requires CUDA compute capability 6.1 or higher (for more information, visit the NVIDIA website). <li data-bbox="220 636 1345 697">• When training and processing High Detail mode, a requirement of 8 GB of GPU memory is for runtime operation, and 11 GB of GPU memory is recommended for training. <li data-bbox="220 718 1387 842">• The NVIDIA GeForce Titan V, Quadro GV100 and Tesla V100 include NVIDIA Tensor Cores. VisionPro Deep Learning 1.0.0 will automatically utilize Tensor Cores if they are present, provided you have a Standard or Advanced security license. Please contact your Cognex Sales Representative for information about enabling the functionality on your Cognex Security Dongle. 			

API Development Requirements

The following software and components are necessary for developing VisionPro Deep Learning custom applications through the VisionPro Deep Learning API:

- Microsoft® Visual Studio® 2015, 2017 or 2019
- Microsoft .NET Framework 4.7.2

i **Note:** For more information about the VisionPro Deep Learning API, please refer to the *Cognex Deep Learning .NET Documentation* and the *Cognex Deep Learning C Documentation* help files.

Cognex Designer and VisionPro Support

VisionPro Deep Learning is qualified to support the following Cognex Designer and VisionPro releases:

- Designer 4.3.1
- VisionPro 9.6

Image File and Format Requirements

Image File Formats

- PNG
- BMP
- TIFF
- JPEG

Image Channels

- 1 to 4

Bit Depth

- 8-bit
- 16-bit

VisionPro Deep Learning API Changes

The topics in this section provide a change log due to updates to .NET Core.

New Targets

- Non-UI NuGet packages now target NET Standard 2.0
- UI assemblies and executables now target .NET Framework 4.7.2
- Non-UI executables now target .NET Core 3.1

NuGet Packages

- NuGet packages are project-based, and have dependencies.

Use of NuGet Packages

VisionPro Deep Learning .NET library is compiled targeting a cross-platform framework, wherever possible.

All of the UI-related assemblies are compiled targeting either .NET Framework 4.7.2 or NET Standard 2.0.

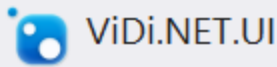
Since VisionPro Deep Learning inherits ViDi, it retains the name ViDi in many parts of the APIs

Assembly	Framework Target
ViDi.NET.Local	NET Standard 2.0
ViDi.NET.Logging	NET Standard 2.0
ViDi.NET.Remote	NET Standard 2.0
ViDi.NET.Remote.Service	NET Standard 2.0
ViDi.NET.Remote.Client	NET Standard 2.0
ViDi.NET	NET Standard 2.0
ViDi.NET.Base	NET Standard 2.0
ViDi.NET.Common	NET Standard 2.0
ViDi.NET.Extensions	NET Standard 2.0
ViDi.NET.Interfaces	NET Standard 2.0
ViDi.NET.UI.Interfaces	.NET Framework 4.7.2
ViDi.NET.GUI	.NET Framework 4.7.2
ViDi.NET.UI	.NET Framework 4.7.2
ViDi.NET.VisionPro	.NET Framework 4.7.2

NuGet packages were introduced to simplify which assemblies are required to use a certain Cognex Deep Learning Studio functionality.

For example, the ViDi.NET.UI now shows exactly which assemblies are required, along with the requisite licenses for 3rd party assemblies.

Note: Dependencies may require additional dependencies.



Versions - 1

<input type="checkbox"/>	Project	Version
<input type="checkbox"/>	Example.Runtime.Console.MultiGPU\Example.Runtime.Console.Multi	
<input type="checkbox"/>	Example.Runtime.Console\Example.Runtime.Console.csproj	
<input type="checkbox"/>	Example.Runtime.GUI\Example.Runtime.GUI.csproj	5.0.0.23161-DEV
<input type="checkbox"/>	Example.Runtime.Parameters\Example.Runtime.Parameters.csproj	
<input type="checkbox"/>	Example.Training.Console\Example.Training.Console.csproj	
<input type="checkbox"/>	Example.Training.GUI.Plugin\Example.ViDi.Suite.GUI.Plugin.csproj	
<input type="checkbox"/>	Example.Training.GUI\Example.Training.GUI.csproj	
<input type="checkbox"/>	Example.VisionPro.Integration\Example.VisionPro.Integration.csproj	5.0.0.23161-DEV

Installed: 5.0.0.23161-DEV

Uninstall

Version: Latest prerelease 5.0.0.23161-DEV

Install

Options

Description

Interfaces for ViDi controls

Version: 5.0.0.23161-DEV**Owner(s):** Cognex Corporation**Author(s):** Cognex Corporation**License:** [View License](#)**Date published:** Wednesday, July 15, 2020 (7/15/2020)**Project URL:** <https://www.cognex.com/>

Dependencies

- ▲ .NETFramework,Version=v4.7.2
 - ViDi.NET.Common (>= 5.0.0.23161-DEV)
 - ViDi.NET.Extensions (>= 5.0.0.23161-DEV)
 - ViDi.NET.Interfaces (>= 5.0.0.23161-DEV)
 - ViDi.NET.Logging (>= 5.0.0.23161-DEV)
 - ViDi.NET.UI.Extensions (>= 5.0.0.23161-DEV)
 - ViDi.NET.UI.Interfaces (>= 5.0.0.23161-DEV)
 - ViDi.NET (>= 5.0.0.23161-DEV)
 - AForge (>= 2.2.5)
 - AForge.Imaging (>= 2.2.5)
 - MvvmLightLibs (>= 5.4.1.1)
 - System.Windows.Interactivity.WPF (>= 2.0.20525)

Note:

- VisionPro Deep Learning NuGet packages are located in the following directory:

C:\ProgramData\Cognex\VisionPro Deep Learning\X.X\Examples\packages

(C:) > ProgramData > Cognex > VisionPro Deep Learning > 1.0 > Examples > packages

Name	Date modified	Type
UserMessage.10.20.3122.201.nupkg	7/15/2020 1:23 PM	NUPKG File
ViDi.Cloud.Client.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
ViDi.Cloud.Common.5.0.0.23161-DEV.nu...	7/15/2020 1:19 PM	NUPKG File
ViDi.NET.5.0.0.23161-DEV.nupkg	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Base.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
ViDi.NET.Common.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
ViDi.NET.Data.5.0.0.23161-DEV.nupkg	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Extensions.5.0.0.23161-DEV.nup...	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Interfaces.5.0.0.23161-DEV.nupkg	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Local.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
ViDi.NET.Logging.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
ViDi.NET.Remote.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
ViDi.NET.Remote.Client.5.0.0.23161-DEV....	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Remote.Service.5.0.0.23161-DEV...	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Training.UI.5.0.0.23161-DEV.nup...	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.Training.UI.Interfaces.5.0.0.2316...	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.UI.5.0.0.23161-DEV.nupkg	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.UI.Extensions.5.0.0.23161-DEV.n...	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.UI.Interfaces.5.0.0.23161-DEV.nu...	7/15/2020 1:20 PM	NUPKG File
ViDi.NET.VisionPro.5.0.0.23161-DEV.nupkg	7/15/2020 1:19 PM	NUPKG File
Wibu.CodeMeter.WibuCmNET.6.70.3164....	7/15/2020 1:23 PM	NUPKG File

The NuGet feed is automatically set in the examples, but must be set to a global Visual Studio NuGet package feed.

- Cognex highly recommends backing up the NuGet packages to another secure repository (or its equivalent) and to keep a copy. This location should then be configured as a new NuGet feed. For more information, see the following Microsoft topic: [Hosting your own NuGet feeds](#)



CAUTION: A NuGet will be overwritten if a new version is installed. If both versions are unavailable, a manual NuGet upgrade must be performed.

Localization

The Chinese translations are now located in zh-Hans, instead of zh-CHS.

Breaking Changes to the API

- As a new mode of Green Classify Tool which is High Detail mode added, there are some changes to API as well.

Assembly	Namespace	Interface or Enumeration Member	Description
ViDi.NET.Interfaces	ViDi2	ToolType.GreenHighDetail	A new enumerator element for Green Classify Tool in High Detail model
ViDi.NET.Interfaces	ViDi2.Training	IGreenHighDetailParameters	Training parameters for Green Classify Tool in High Detail mode
ViDi.NET.Interfaces	ViDi2.Runtime	IGreenHighDetailParameters	Runtime parameters for Green Classify Tool in High Detail mode
ViDi.NET.Interfaces	ViDi2	HeatMap	HeatMap for Green Classify Tool in High Detail mode

- Following codes are for how to get Bitmap from HeatMap.

```
sample.Process(greenTool); // mode of greenTool should be High-detail to get HeatMap.
sample is ISample

IGreenMarking greenMarking = sample.Markings[greenTool.Name] as IGreenMarking;
foreach (IGreenView view in greenMarking.Views)
{
    IImage heatmap = view.HeatMap; // get HeatMap
    System.Drawing.Bitmap bitmap = heatmap.Bitmap; // get Bitmap
    // you can use bitmap for whatever you want
}
```

- Below list is no longer supported.

Assembly	Namespace	Interface or Enumeration Member	Description
ViDi.NET.Interfaces	ViDi2	GpuMode.MultipleDevicesPerTool	A member for using multiple GPUs per tool

API Known-Issues

Due to an issue with CodeMeter .NET Standard, in which a hang could be introduced when first calling a ViDi2.Remote, the following is added in App.config:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
        <dependentAssembly>
            <assemblyIdentity name="WibuCmNET" publicKeyToken="01d86e1eb0c69c23"
culture="neutral"/>
            <bindingRedirect oldVersion="6.0.0.0-8.0.0.0" newVersion="7.0.3933.502"/>
        </dependentAssembly>
    </assemblyBinding>
</runtime>
```

```

    </dependentAssembly>
  </assemblyBinding>
</runtime>
</configuration>

```

C and .NET API Deprecated and Removed Features

The following list of C and .NET functions are being deprecated, and will either be renamed or removed in the next major release.

Library	Obsolete Name	New Name
.NET	ToolDatabase.GetOverlayImage()	
.NET	ViewKey.FileName	FrameKey.SampleName
.NET	LibraryAccess.SampleGetOverlayImage()	
.NET	IManualRegionOfInterest.External	IManualRegionOfInterest.Parameters.Mode
.NET	IManualRegionOfInterest.Mode	IManualRegionOfInterest.Parameters.Mode
.NET	IManualRegionOfInterest.Units	IManualRegionOfInterest.Parameters.Units
.NET	IManualRegionOfInterest.Offset	IManualRegionOfInterest.Parameters.Offset
.NET	IManualRegionOfInterest.Size	IManualRegionOfInterest.Parameters.Size
.NET	IManualRegionOfInterest.Angle	IManualRegionOfInterest.Parameters.Angle
.NET	IManualRegionOfInterest.SplittingGrid	IManualRegionOfInterest.Parameters.SplittingGrid
.NET	IBlueRegionOfInterest.Model	IBlueRegionOfInterest.Parameters.MatchFilter
.NET	IBlueRegionOfInterest.Models	
.NET	IBlueRegionOfInterest.TargetMode	IBlueRegionOfInterest.Parameters.TargetMode
.NET	IBlueRegionOfInterest.FeatureFilter	IBlueRegionOfInterest.Parameters.FeatureFilter
.NET	IBlueRegionOfInterest.MatchFilter	IBlueRegionOfInterest.Parameters.MatchFilter
.NET	IBlueRegionOfInterest.MatchItems	IBlueRegionOfInterest.Parameters.MatchItems
.NET	IBlueRegionOfInterest.MatchItemsMode	IBlueRegionOfInterest.Parameters.MatchItemsMode
.NET	IBlueRegionOfInterest.Centered	IBlueRegionOfInterest.Parameters.Centered
.NET	IBlueRegionOfInterest.Units	IBlueRegionOfInterest.Parameters.Units
.NET	IBlueRegionOfInterest.Offset	IBlueRegionOfInterest.Parameters.Offset
.NET	IBlueRegionOfInterest.Size	IBlueRegionOfInterest.Parameters.Size
.NET	IBlueRegionOfInterest.Angle	IBlueRegionOfInterest.Parameters.Angle
.NET	IBlueRegionOfInterest.Scaled	IBlueRegionOfInterest.Parameters.Scaled
.NET	IBlueRegionOfInterest.Scale	IBlueRegionOfInterest.Parameters.Scale
.NET	IRedRegionOfInterest.ExtractRegions	IRedRegionOfInterest.Parameters.ExtractRegions
.NET	IRedRegionOfInterest.RegionFilter	IRedRegionOfInterest.Parameters.RegionFilter
.NET	IRedRegionOfInterest.Units	IRedRegionOfInterest.Parameters.Units
.NET	IRedRegionOfInterest.Border	IRedRegionOfInterest.Parameters.Border
.NET	IRedRegionOfInterest.MaskRegions	IRedRegionOfInterest.Parameters.MaskRegions
.NET	IRedRegionOfInterest.FailOnFilter	IRedRegionOfInterest.Parameters.FailOnFilter
.NET	IRedRegionOfInterest.RecycleMask	IRedRegionOfInterest.Parameters.RecycleMask
.NET	IRedRegionOfInterest.InvertMask	IRedRegionOfInterest.Parameters.InvertMask
.NET	IRedRegionOfInterest.GrowShrinkMask	IRedRegionOfInterest.Parameters.GrowShrinkMask

Library	Obsolete Name	New Name
.NET	IGreenRegionOfInterest.Filter	IGreenRegionOfInterest.Parameters.Filter
.NET	IMatch.FeatureString	IReadModelMatch.FeatureString
.NET	IMatch.PrettifiedString	IReadModelMatch.PrettifiedString
.NET	IMatch.Points	INodeModelMatch.NodePoints
.NET	IMatch.Pose	INodeModelMatch.Pose
.NET	IMatch.Position	IReadModelMatch.Position
.NET	IMatch.Orientation	INodeModelMatch.Orientation
.NET	IMatch.Scale	INodeModelMatch.Scale
.NET	IMatch.AspectRatio	INodeModelMatch.AspectRatio
.NET	IMatch.Shear	INodeModelMatch.Shear
.NET	IModel.Occurences	IModel.Occurrences
.NET	INode.Names	INode.Fielding
.NET	ITool.Parameters	ITool.ParameterBase
.NET	IFeature.Id	IFeature.Name
.NET	IRedView.Deviation	IRedView.Score
.NET	ITag.Key	ITag.Name
.NET	ITag.Value	ITag.Score
.NET	IWorkspace.Name	IWorkspace.UniqueName
.NET	IWorkspace.LastModification	
.NET	IToolDatabase.SetTrainFlag(string filter, string group)	IToolDatabase.SetTrainFlag(string filter, bool flag)
.NET	IToolDatabase.GetOverlayImage(ViewKey, string, uint)	
.NET	ILibraryAccess.ToolGetOverlay(WorkspaceName, StreamName, ToolName, string, uint, string, ILibraryAccessContext)	
.NET	IStream.AddImage(Image, string) IStream.AddImage(System.IO.Stream, string) GetImage(string, uint)	
.NET	ITool.Mask	ITool.RegionOfInterest.Mask
.NET	ITool.Parameters	ITool.ParametersBase ?????
.NET	IToolParameters.TrainSelection	
.NET	IToolParameters.KeepTrainSelection	
.NET	Model.Occurences	Model.Occurrences
.NET	BorderType.White BorderType.Reflect BorderType.Wrap	
.NET	Tag.Key	Tag.Name
.NET	Tag.Value	Tag.Score
.NET	Workspace.Name	Workspace.UniqueName Workspace.DisplayName
.NET	Node.Names Node.IDs	Node.Fielding
.NET	Model.Occurences	Model.Occurrences
.NET	Model.AllOccurences	Model.AllOccurrences

Library	Obsolete Name	New Name
.NET	ProductCode.OptimizedMemory	ProductCode.Standard ProductCode.Advanced
.NET	DatabaseExplorerViewModel.SetViewGroup	
.NET	Viewer.Database	
.NET	DatabaseOverview	DatabaseOverviewViewModel
.NET	WorkspaceBrowser	WorkspaceBrowserViewModel ??????
.NET	IWorkspaceBrowser	IWorksapceBrowserViewModel ??????
.NET	CalibratedFeature.Id	
.NET	CalibratedRedView.Deviation	
.NET	TrainingFlagMode.Auto	
.NET	RedTool.IRedTool.Parameters	RedTool.IRedTool.BaseParameters????**
C	vidi_runtime_init_sample()	vidi_runtime_create_sample()
C	vidi_runtime_process()	vidi_runtime_sample_process()
C	vidi_runtime_process_sample()	vidi_runtime_sample_process()
C	vidi_runtime_get_image()	vidi_runtime_sample_get_image()
C	vidi_runtime_sample_get_overlay()	Use get_overlay for the specific tool.
C	vidi_training_tool_get_status()	vidi_training_stream_list_tools()
C	vidi_training_stream_process_sample()	vidi_training_sample_process()
C	vidi_training_blue_add_feature()	vidi_training_set_feature()
C	vidi_training_blue_add_model()	vidi_training_blue_create_model()
C	vidi_training_tool_set_mask()	vidi_training_roi_set_mask()
C	vidi_training_tool_get_mask()	vidi_training_roi_get_mask()
C	vidi_training_init_sample()	vidi_training_add_sample()
C	vidi_training_process_sample()	vidi_training_sample_process()
C	vidi_training_sample_get_overlay()	
C	vidi_training_tool_get_overlay()	
C	occurences	occurrences
C	names	fielding
C	red/mode	training/network_model
C	vidi_training_tool_train()	vidi_training_tool_train2()

Software Installation

To successfully install VisionPro Deep Learning, perform the following:

1. Attach the Cognex Security Dongle to a USB port on the computer that will be used to develop the vision application.
2. Download the VisionPro Deep Learning installer from the [Cognex support page](#).
3. Run the VisionPro Deep Learning installer and follow the prompts.

Note:

Choosing the Custom option will allow you to install selected features:

- The Wibu Runtime Server, which is needed to connect to the USB Cognex Security Dongle.
- The main VisionPro Deep Learning application (this is required).
- The VisionPro Deep Learning Developer API.



- The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.

When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.

Documentation Updates

You can access the latest VisionPro Deep Learning Help on the Cognex Support website:
<https://support.cognex.com/en/documentation/deep-learning>

