

About This Release

With the release of VisionPro Deep Learning 2.0, Cognex is announcing that the Deep Learning product is splitting into two major branches: 1.1.x and 2.x. The main difference between the two branches is how the Deep Learning software is integrated with other Cognex software. Besides the integration paths, there is no functional difference between VisionPro Deep Learning 1.1 and 2.0:

Integration	Cognex VisionPro Deep Learning 1.1	Cognex VisionPro Deep Learning 2.0
Cognex Designer ¹	O	X
Cognex VisionPro 10.0 (or higher)	X	O

For VisionPro Deep Learning 2.0, VisionPro Deep Learning supports the integration with VisionPro 10.0, the brand new VisionPro with enhanced features. Please see VisionPro documentation for the details about VisionPro 10.0.

Since VisionPro Deep Learning runtime workspace can be directly imported into VisionPro 10.0 QuickBuild **the integration with Cognex Designer (the Plug-in for Cognex Designer) is no longer supported from VisionPro Deep Learning 2.0.**

[New Content]

- VisionPro 10.0 Integration
 - VisionPro Deep Learning offers its tools (Blue, Red, Green tool in Runtime Workspace) for VisionPro 10.0 to be directly used in its QuickBuild as one of a VisionPro tool (DLRuntimeTool). See Integration with Cognex VisionPro and [Known Issues on page 4](#).

Note:

- The VisionPro 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 2.0 installer. The resources for the sample code (*VisionPro Deep Learning 2.0 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.

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.
- SuaKIT users can import SuaKIT classification and segmentation project into VisionPro Deep Learning. They can then take advantage of the other features of Cognex Deep Learning Studio, including the Blue Locate, OCR tools, and tool chaining.

Resolved Issues

This section describes changes and resolved issues in VisionPro Deep Learning 2.0.

- Fixed an issue that a plug-in that was built in Examples (C:\ProgramData\Cognex\VisionPro Deep Learning\2.0\Examples\c#\Example.Deep.Learning.Studio.GUI.Plugin) is not shown on Add Plugin window of Plugin Manager.

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.

Release - VisionPro Deep Learning 1.1.0 - 2.0

The End of Designer Support

The integration with Cognex Designer is no longer supported from VisionPro Deep Learning 2.0. For this particular update, the users who are required to use Designer should use VisionPro Deep Learning 1.1.

Past Releases - VisionPro Deep Learning 1.0.1 - 1.1.0

Change in Validation Loss Calculation

The base of calculating the Validation Loss in High Detail modes (Red Analyze and Green Classify) is changed from **128 iterations** to **0.125 epochs**. According to this change, the *Patience* parameter, which was calculated on 128 iterations (N times of 128 iterations) basis, is no longer used in High Detail modes and a new parameter named *Patience Epoch* is used instead. This new parameter is calculated on an epoch basis (N epochs). For the Green Classify High Detail tools already trained, you have to reconfigure the value of *Patience Epoch* manually due to this base change.

Loss Tracking in Log Messages

The records of the validation loss in High Detail modes (Green Classify and Red Analyze High Detail) are no longer available in Log Messages as its function of providing the validation loss history is replaced with the Loss Inspector.

Region Filter for Confusion Matrix

For all Red Analyze tools, the region filter is now also applied to the result of the confusion matrix. This application of the region filter is either performed by using filtered() expression.

Past Releases - VisionPro Deep Learning 1.0.0 - 1.0.1

The workspaces created in VisionPro Deep Learning 1.0.0 are fully compatible with VisionPro Deep Learning 1.0.1

Past Releases - ViDi 4.1 to VisionPro Deep Learning 1.0.0

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 3.4.1 to 4.1

Changes to the Feature Filter Syntax

In the 4.1.0 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 4.1.0 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]
```

Changes to the Viewing of Overlay Graphics

In the 3.4.1 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.

Known Issues

The following issues have been identified in this release:

[GPU]

- Internal issues on supporting NVIDIA driver versions.
Due to internal issues related to NVIDIA drivers, all users of VisionPro Deep Learning 2.0 **MUST** use NVIDIA driver according to the followings:
 - 461.09 or higher version for NVIDIA GeForce® series.
 - R460 U2 (461.09) or higher version for NVIDIA RTX / Quadro® series.

- C API malfunctioning depending on NVIDIA drivers.
Due to internal issues related to runtime, C API may not work as intended. To prevent this issue, all users of C API **MUST** follow the steps below.
 - Remove the currently installed NVIDIA driver from your PC.
 - Reboot your PC and install the **DCH** NVIDIA driver of which version is compatible with VisionPro Deep Learning 2.0 (461.09 or higher, or R460 U2 (461.09) or higher for NVIDIA RTX / Quadro®)
- Processing Time could be slower compared to 1.0.0 under old GPUs.
Processing Time could be slower compared to 1.0.0 when using NVIDIA GeForce® RTX™ 20 series or older series due to NVIDIA internal issues.

[OS]

- OS compatibility issues on Windows Server 2016
Due to OS compatibility issues caused from NVIDIA drivers, Windows Server 2016 is not supported on VisionPro Deep Learning 2.0. For the users with Windows Server 2016 environment, upgrading from Windows Server 2016 to Windows Server 2019 is required for the utilization of VisionPro Deep Learning 2.0 servers. Windows Server 2019 is only supported for 'servers' configured to use the Deep Learning Client/Server functionality.

[Deep Learning Tools]

- Erasing maskings/regions in the mask/region editor pane (*Edit Mask or Edit Region* in right-click popup menu on the image display area) applies Gaussian smoothing on all the regions/maskings. This causes the entire maskings/regions to shift slightly and their sizes to be changed slightly. This issue has been an existing issue from VisionPro ViDi.
- For Red Analyze Supervised modes (High Detail and Focused), applying the region filter could affect the result of Auto Threshold to be compromised in generating an optimal threshold for the best F1 score.
- For Red Analyze High Detail, generally, you can guess that processing time increases when you reduce the Downsampling Size, but the opposite case can happen but this is not a bug.
- For Red Analyze High Detail, if the tool crashes with some unknown reason, images in this tool are deleted when the workspace that contains this tool is imported.
- For Red Analyze High Detail, changing a value in Processing Parameter in Tool Parameters and rolling back to the original value could affect the result at 4 decimal places.
- For Red Analyze Tool, the area of a marked defect region and the area of a labeled defect region which is created by accepting (right-click on a marked region → click *Accept Region*) that marked region, is slightly different from each other.
- For Red Analyze Supervised, if there were an extremely enormous amount of defect regions per view, especially when the tool is not trained enough, VisionPro Deep Learning becomes slow and the expected value of processing time becomes inaccurate. Please take the tool re-trained to learn the image information fairly enough.
- High Detail mode uses a different resize algorithm to SuaKIT. (Green Classify Tool in High Detail mode only)
To use an 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 touches the left side of the image, please set 4 pixels wider ROI than the 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. The vvb 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 vvb file size=94MB
 - feature size=10, sampling density=10, 8192x8192 image: 1 vvb file size=1GB
 - feature size=40, sampling density=3, 8192x8192 image: 1 vvb file size=23KB

[API]

- Several minor issues with using APIs. See FAQ for the details.
- Running Example.Runtime.Remote.Console returns results to only 4 decimal places, not 6 as in the previous VisionPro Deep Learning release.

[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 by processing the tools again and save the workspace.
- Boost::filesystem::rename error.
This error occurs when the user or system repetitively saves the workspace in a short time. This is just an alert that there was an issue while saving. You can solve this problem only by saving the workspace again.
- Group is shown in the workspace imported from ViDi Suite.
 - If you import workspaces including "Group" feature which are made 4.1 and before, "Group" is still displayed in Cognex Deep Learning Studio 2.0, but you cannot modify this.
 - See the FAQ for the workaround for this issue.
 - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio 2.0.
- 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 (neither turn off the remote environment nor lock screen) while you are creating a report.

[Migration]

- For SuaKIT plugin, The result of processing between the imported SuaKIT Segmentation project in VisionPro Deep Learning and the original SuaKIT Segmentation project could be different due to the internal logic change of VisionPro Deep Learning2.0.

[Integration]

- Due to the currently limited support of VisionPro 10, the external ROI is not supported for a DLRuntimeTool Edit Control Panel in VisionPro 10 for the time being.

[VisionPro Deep Learning Service]

- MultipleDevicesPerTool is no longer supported for VisionPro Deep Learning. Therefore you will have an “One or more error occurred” error if you select “--gpu-mode =MultipleDevicesPerTool” for service. Please choose another option to run VisionPro Deep Learning Service.
- For VisionPro Deep Learning Service, Export Runtime Workspace to Service on the right-click pop-up menu on Workspaces panel is currently disabled. To export runtime workspace, please go to and select *Workspace - Export Runtime Workspace to Service* instead.
- For VisionPro Deep Learning Service, when hosting a service with 2 different server ports (Client 1 connects via Port 1 and Client 2 connects via Port 2), the “Lock” icon does not appear on a workspace in Workspaces for a client when this workspace has been already used by one another.

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 2019	✓	✓	✓	✓	✓	✓	✓

Note:



- The use of the VisionPro Deep Learning software or the Deep Learning Service on a virtual machine (VM) is not supported.
- Due to the issues from the NVIDIA driver, Windows Server 2016 is not supported for VisionPro Deep Learning 2.0 Server/Client mode. See [Known Issues](#) for more details and the workaround.

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 the higher specification than Intel Core i7. When selecting a CPU, a higher CPU clock speed rate and multiple core processors will result in faster 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)

For training, we recommend GPUs that have 10GB or larger GPU Memory (1080 Ti, 2080 Ti, 3080).

For the processing of High Detail modes, GPUs that have 8GB or larger GPU Memory are recommended.

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 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

i 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 the higher specification than Intel Core i7. When selecting a CPU, a higher CPU clock speed rate and multiple core processors will result in faster 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.

-

For training, we recommend GPUs that have 10GB or larger GPU Memory (1080 Ti, 2080 Ti, 3080).

- 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™ 3080 Ti GPUs, which each have 10 GB of memory, the PC should have 40 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.

i 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 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.

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 the higher specification than Intel Core i7. When selecting a CPU, a higher CPU clock speed rate and multiple core processors will result in faster 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 High Detail modes, 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 for processing, VisionPro Deep Learning 2.0 only supports NVIDIA GPUs. Consider the following when choosing an NVIDIA GPU:

At a minimum, an NVIDIA® CUDA® enabled GPU is required.

When selecting an NVIDIA GPU, a unit with a higher core clock frequency, CUDA cores and Tensor cores will result in faster computational speeds. Due to this, Cognex strongly recommends the use of NVIDIA RTX / Quadro® and Tesla GPUs for the following reasons:

- These GPUs support the compute-optimized Tesla Compute Cluster (TCC) mode.
- These GPUs are designed and tested for continuous duty-cycle operation.
- These GPUs undergo rigorous testing and qualification by NVIDIA.

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 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

i 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 increases performance of runtime workspaces.

NVIDIA Driver Support

- GeForce® RTX driver greater than or equal to 461.09
- NVIDIA RTX / Quadro® / Data Center driver greater than or equal to R460 U2 (461.09) (Optimal Driver for Enterprise).
 - It is highly recommended to visit <https://www.nvidia.com/Download/Find.aspx> to find the right versions for your GPU as your GPU may require higher version than R460 U2 (461.09).

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

Supported NVIDIA GPUs

VisionPro Deep Learning 2.0 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 / RTX 2080 Ti / RTX 3080

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
	RTX 3060		
	RTX 3060 Ti		
	RTX 3070	Tested	
	RTX 3080	Tested	Recommended
	RTX 3090	Tested	Recommended
NVIDIA RTX / 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*		

Models known not to work:

- GTX 1660 and GTX 1650: Please note that low performance and slow training time with unknown reason.

GPU	Model	Tested by Cognex	Recommended for High Performance
<p>Note: When selecting a GPU, Cognex only supports NVIDIA GPUs.</p> <ul style="list-style-type: none"> • • VisionPro Deep Learning 2.0 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 461.09. • 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). • When training and processing High Detail mode, a requirement of 8 GB of GPU memory is for runtime operation, and 10 GB of GPU memory is recommended for training. • The NVIDIA GeForce Titan V, NVIDIA RTX / Quadro® GV100 and Tesla V100 include NVIDIA Tensor Cores. VisionPro Deep Learning 2.0 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

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 VisionPro Support

VisionPro Deep Learning is qualified to support the following Cognex VisionPro release:

- VisionPro 10.0

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.

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.


 ViDi.NET.UI

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 Red Analyze High Detail mode and Loss Inspector are added in VisionPro Deep Learning 2.0, there are some changes to API as well.

C/C++ API

Major changes of C/C++ API include access to and interaction with Red Analyze High Detail mode and Loss Inspector.

Red Analyze High Detail

```

1
2 ...
3 // Red HD mode
4 std::stringstream param_ss;
5 param_ss << "workspaces/" << workspace_name << "/streams/" << stream_name
  << "/tools/" << tool_name << "/tool_type";
6 status = vidi_training_set_parameter(param_ss.str().c_str(), "red_sua");
7 ...

```

Loss Graph

```

1 ...
2 // Get lossgraph data
3 status = vidi_training_get_loss_graph_data(workspace_name, stream_name, tool_
  name, true, 0, &lossgraph_buffer);
4 CHECK_STATUS(status);
5
6 rapidxml::xml_document<> doc;
7 doc.parse<0>(lossgraph_buffer.data);
8
9 rapidxml::xml_node<>* lossgraph_node = doc.first_node("loss_graph_data");
10 assert(lossgraph_node);
11
12 // check if it has "data" node
13 if (lossgraph_node->first_node("data") == nullptr)
14     break;
15
16 rapidxml::xml_node<>* last_node = lossgraph_node->last_node("data");
17 assert(last_node);
18
19 rapidxml::xml_attribute<>* elapsedEpoch = last_node->first_attribute("current_
  epoch");
20 rapidxml::xml_attribute<>* bestEpoch = last_node->first_attribute("min_epoch");
21 rapidxml::xml_attribute<>* elapsedLoss = last_node->first_attribute("current_loss");
22 rapidxml::xml_attribute<>* bestLoss = last_node->first_attribute("min_loss");
23 ...

```

Please refer to *Examples (C:\ProgramData\Cognex\VisionPro Deep Learning\2.0\Examples\c++\Example.Training.HDRed, Example.Training.HDGreen)* for the details of usage.

Please refer to C/C++ API document (C:\Program Files\Cognex\VisionPro Deep Learning\Cognex\VisionPro Deep Learning\2.0\Develop\docs) for the details of APIs.

Header File	API	Description
vidi_training.h	vidi_training_clear_backup_loss_graph_data function	Clear the backed up loss graph data for the High Detail Mode Tool
vidi_training.h	vidi_training_get_loss_graph_data function	Get the loss graph data for the High Detail Mode Tool
vidi_training.h	vidi_training_restore_loss_graph_data function	Restore the backed up loss graph data for the High Detail Mode Tool

.NET API

Major changes of .NET API include access to and interaction with Red Analyze High Detail mode and Loss Inspector.

Red Analyze High Detail

```

1 | ...
2 |
3 | // creates a new workspace
4 |     var workspace = control.Workspaces.Add("workspace");
5 |
6 |     //creates a stream
7 |     var stream = workspace.Streams.Add("default");
8 |
9 |     //creates a red tool at the root of the toolchain
10 |     var hdRedTool = stream.Tools.Add("red_HDM", ToolType.Red) as IRedTool;
11 |
12 | ...

```

Loss Graph

```

1 | ...
2 |
3 | // 1. Get loss graph data using GetLossGraphData method
4 |     var lossGraph = hdRedTool.GetLossGraphData(true);
5 |     if (lossGraph.Count() != 0)

```

```

6      System.Console.WriteLine($"{lossGraph.Last().CurrentEpoch,15} {lossGraph.Last
7      ().MinEpoch,15} {lossGraph.Last().CurrentLoss,15} {lossGraph.Last().MinLoss,15} {"
8      ",15}");
...

```

Please refer to Examples (*C:\ProgramData\Cognex\VisionPro Deep Learning\2.0\Examples\c#\Example.Training.HDRRed.Console, Example.Training.HDRRed.Console*) for the details of usage.

Please refer to .NET API document (*C:\Program Files\Cognex\VisionPro Deep Learning\Cognex\VisionPro Deep Learning\2.0\Develop\docs*) for the details of APIs.

Assembly	Namespace	Interface, Member, or API	Description
ViDi.NET.Interfaces	ViDi2.Runtime / ViDi2.Training	IRedHighDetailParameters	Runtime parameters for high-detail red tool
ViDi.NET.Interfaces	ViDi2.Training	ILibraryAccess.ClearBackupLossGraphData Method	Clear the backed up loss graph data for the High Detail Mode Tool
ViDi.NET.Interfaces	ViDi2.Training	ILibraryAccess.GetLossGraphData Method	Get loss graph data for High-detail red/green tool
ViDi.NET.Interfaces	ViDi2.Training	ILibraryAccess.RestoreLossGraphData Method	Restore backed up loss graph data for High Detail Mode 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>
</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
.NET	IGreenRegionOfInterest.Filter	IGreenRegionOfInterest.Parameters.Filter
.NET	IMatch.FeatureString	IReadModelMatch.FeatureString
.NET	IMatch.PrettifiedString	IReadModelMatch.PrettifiedString

Library	Obsolete Name	New Name
.NET	IMatch.Points	INodeModelMatch.NodePoints
.NET	IMatch.Pose	INodeModelMatch.Pose
.NET	IMatch.Position	IReadModelMatch.Position
.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	BorderType.White BorderType.Reflect BorderType.Wrap	
.NET	Node.Names Node.IDs	Node.Fielding
.NET	ProductCode.OptimizedMemory	ProductCode.Standard ProductCode.Advanced
.NET	CalibratedRedView.Deviation	
.NET	TrainingFlagMode.Auto	
.NET	RedTool.IRedTool.Parameters	RedTool.IRedTool.BaseParameters
C	vidi_runtime_init_sample()	vidi_runtime_create_sample()

Library	Obsolete Name	New Name
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 Cognex 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 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

VisionPro Deep Learning 2.0

- Updates for VisionPro Deep Learning 2.0 release
 - The documentation for the integration with Cognex Designer is no longer provided

VisionPro Deep Learning 1.1

- Updates for VisionPro Deep Learning 1.1 release
- Designer Plugin script examples for each tool have been updated for VisionPro Deep Learning 1.1.
- Updates of the comments in C#, C++ example codes (...\\ProgramData\\Cognex\\VisionPro Deep Learning\\1.1\\Examples\\)
- Updates about how to use multiple GPUs for VisionPro Deep Learning in Multiple GPUs Utilization.
- Updates about the introduction to the various training parameters for VisionPro Deep Learning in Training Parameters
- Updates about how to import SuaKIT projects in Import SuaKIT Project.

VisionPro Deep Learning 1.0.1

- Updates for VisionPro Deep Learning 2.0 release
- Updates for Migrating To This Release

The migration records from ViDi Suite 3.4 to ViDi Suite 4.1.0 have been reorganized for the correct migration history.

You can access the latest VisionPro Deep Learning Help on the Cognex Support website:

<https://support.cognex.com/en/documentation/deep-learning>

¹ VisionPro Deep Learning 1.1 supports Designer 4.4

