

# Release Information

The VisionPro Deep Learning 4.1 version introduces the following features, enhancements, and changes:

- Reduced the training and preparation time for the Standard type tools. The reductions are more significant for Red Analyze tools that have larger images in the training set.
- A stage called optimizing resources is added before training. This stage can lead to faster training speeds during repeated training runs.
- Improved the system RAM and storage drive usage of Cognex Deep Learning Studio. This improvement provides better performance in areas such as labeling, tool training, tool processing, storage, and runtime export.
- Production mode is now available again. You can access it by clicking **Production Mode** in the **Workspace** toolbar menu same as before.
- The heatmap is now displayed in production mode for the Standard type Green Classify tool.
- Updated the Codemeter version from 7.51 to 8.20a.
- Added the **Optimized for Green/Red Standard tool** toggle to the Cognex Deep Learning Studio toolbar area, allowing you to easily toggle the [Optimized GPU Memory](#) option on and off.
- Introduced the heatmap-based label generator plugin. This plugin helps to convert heatmap results from Few Sample mode as labels.

**Note:**

- Sample code is included with the VisionPro Deep Learning 4.1 installer. The resources for the sample code (*VisionPro Deep Learning 4.1 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 resolved issues in the VisionPro Deep Learning 4.1 release.

### Resolved Tool Issues

The following tool issues are resolved in VisionPro Deep Learning 4.1:

Issue Type	Description
General tool issue	Attempting to train multiple tools repeatedly resulted in an internal error.
	Moving the Region of Interest (ROI) outside the view after rotating it in the <b>Edit ROI</b> window caused a crash.
	Exports failed with an error when the runtime workspace size was large.
Red Analyze tool issue	The training process proceeded and terminated abnormally in Fast and Accurate mode, even when the image labeling conditions were not met.
	Error pop-up appeared during training in Standard type when adding masks to images larger than a certain size.
	The auto threshold setting occasionally failed to apply.
	The inverted mask was not immediately applied after tool chaining a Red Analyze tool.
	The tooltip text for several parameters of the StandardRed Analyze tool displayed incorrectly.
Green Classify tool issue	The class names in the confusion matrix were not updated when switching between multiple Green Classify tools with two classes.
	The tooltip text for several parameters of the Standard type Green Classify tool displayed incorrectly.

### Other Resolved Issues

The following issues are resolved in VisionPro Deep Learning 4.1:

Issue Type	Description
API documentation issue	The description for the heatmap section of <i>IGreenView</i> in the C# API documentation was displayed incorrectly.
	Fixed the description for Anomaly Detection (Legacy type Red Analyze) in the tooltype enumeration section of the C# API documentation and added the appropriate explanation.
Designer issue	Error occurred while importing a runtime workspace created by inputting images in folder units from VisionPro Deep Learning into the Designer.
GPU memory issue	The GPU memory was not released after processing a Standard type tool in Cognex Deep Learning Studio.
Label checker issue	The label checker results did not fit within the ROI after resizing the ROI to be smaller than the image.
License issue	Processing using the runtime API failed with an error when using the Legacy type Red Analyze/Green Classify-only license.
VisionPro issue	The heatmap for the Green Classify tool did not display correctly in tool chains set to Blue Locate - Green Classify in VisionPro.

# Known Issues

The following issues have been identified in this release.

## VisionPro Deep Learning 4.1

- Converting to TensorRT with NVIDIA® GeForce® RTX 5080 fails due to CUDA errors and `AccessViolationException`.
- When using an RTX 50 series GPU, the Runtime API consumes more than twice as much GPU memory during processing compared to GPUs with CUDA Compute Capability 9.0 or earlier.  
  
In particular, using the `Task (ThreadPool)` method provided in the API Sample Code consumes more GPU memory. This can be resolved by modifying the `Thread` method.

## VisionPro Deep Learning 4.0

- Each time you perform training in the GUI, a memory leak of about 3-5 MB occurs, which is reset by restarting the GUI.
- Windows 10, version 1809 has an issue when running the VisionPro Deep Learning application. Cognex recommends using Windows 10, version 22h2.
- The **Create Report** option in the **Database** menu does not work correctly in Cognex Deep Learning Studio. Tag, Score, and Overlay images by View do not appear in the generated report.

## VisionPro Deep Learning 3.3.0

There is a score difference between the GUI and the runtime when using 16-bit images with the Green Classify Standard, Green Classify High Detail Quick, and the Red Analyze Standard tools.

## VisionPro Deep Learning 3.2

### GPU

- 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.
- When optimizing the **Small Network Model** with the Tensor RT feature on NVIDIA RTX 1080 Ti or NVIDIA RTX 2080 Ti GPU models, there is a spike in processing time.
- Due to several issues related to GUI and training, all users of VisionPro Deep Learning 4.1 MUST use NVIDIA driver that is of:
  - After VisionPro Deep Learning 4.1, recommended GPU driver version range is from 528.02 to 532.03, with the tested version being 528.02. Using driver versions beyond 532.03 may result in slower training and processing times in Standard .

## Cognex Deep Learning Studio (GUI)

- 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.
- “No License Found” error occasionally occurs when you try to open 2 different versions of VisionPro Deep Learning, or launch Parameter Search and VisionPro Deep Learning, at the same time.

## Workspace

- Boost::filesystem::rename 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 3.0, but you cannot modify this.
  - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio 3.0.
- 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.

## Tools - General

- If a tool is not saved well because of unspecific reasons, you cannot export or clone the workspace or the tool. You can avoid this problem by processing the tools again and save the workspace.
- For all tools, if the tool crashes with some unknown reason, images in this tool are deleted when the workspace that contains this tool is imported.
- If the name of a set in Database Overview contains a single quote ('), you cannot change the name of this set.
- If the name of a set in Database Overview contains a single quote ('), an error occurs when you try to filter views on View Browser by clicking the icon of this set in Database Overview.
- When using 16-bit images with High Detail tools, the runtime result is incorrect.

## Tools - Red Analyze

- 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, 8192x8192 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
- 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, 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 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 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.
- 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 Legacy, its Database Overview crashes when there are multiple types of labeled defect regions.

## Tools - Green Classify

- Once you are done with training High Detail mode of Green Classify Tool, you cannot change the name of the class.
- For Green Classify Standard, resizing views takes a little more time when it is the first processing of the tool.
- For Green Classify Standard, the processing preparation takes a little more time when it is the first processing of the tool.
- When using the Green Classify Standard tool, the execution time for the label checker significantly increases compared to using 8-bit 4-color channel images.

## Tools - Blue Read

- For Blue Read, the markings are displayed on the view regardless of the Marking checkbox in Image File Information is whether enabled or disabled. This issue has existed since ViDi 4.1.

## Cognex Deep Learning Parameter Search

- In Home page, if you set 1-depth directory path (example: "E:/" or "A:/") in the Path, Cognex Deep Learning Parameter Search is terminated with an error and you can't re-open it.

## 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.

## VisionPro Deep Learning 2.1 (2.0, 2.1.1)

### GPU

- 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.

## Operating System (OS)

- Due to OS compatibility issues caused from NVIDIA drivers, Windows Server 2016 is not supported on VisionPro Deep Learning. 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 servers. Windows Server 2019 is only supported for 'servers' configured to use the Deep Learning Client/Server functionality.

## Cognex Deep Learning Studio (GUI)

- 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.
- “No License Found” error occasionally occurs when you try to open 2 different versions of VisionPro Deep Learning, or launch Parameter Search and VisionPro Deep Learning, at the same time.

## Workspace

- Boost::filesystem::rename 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, but you cannot modify this.
  - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio.
- 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.

## Tools - General

- If a tool is not saved well because of unspecific reasons, you cannot export or clone the workspace or the tool. You can avoid this problem by processing the tools again and save the workspace.
- For all tools, if the tool crashes with some unknown reason, images in this tool are deleted when the workspace that contains this tool is imported.
- If the name of a set in Database Overview contains a single quote ('), you cannot change the name of this set.
- If the name of a set in Database Overview contains a single quote ('), an error occurs when you try to filter views on View Browser by clicking the icon of this set in Database Overview.

## Tools - Red Analyze

- 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, 8192x8192 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
- 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, 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 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 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.
- 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.
- 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.

## Tools - Green Classify

- Once you are done with training High Detail mode of Green Classify Tool, you cannot change the name of the class.

## Tools - Blue Read

- For Blue Read, the markings are displayed on the view regardless of the Marking checkbox in Image File Information is whether enabled or disabled. This issue has existed since ViDi 4.1.

## Cognex Deep Learning Parameter Search

- In Home page, if you set 1-depth directory path (example: "E:/," or "A:/") in the Path, Cognex Deep Learning Parameter Search is terminated with an error and you can't re-open it.

## 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.

## Migration from SuaKIT or The Past VisionPro Deep Learning Versions

- 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 VPD runtime API, please use SuaKIT resize API before processing.
- 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 Learning 1.1.

## Integration with VisionPro

- 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.

## Plug-in

- For Designer, the added heatmap checkbox is initialized on Deep Learning Runtime Blocks of the tools other than Red Analyze. Tools other than Red Analyze were supposed not to support heatmap function.

## VisionPro Deep Learning 1.1 (1.1.1)

### GPU

- 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.
- Internal issues on supporting NVIDIA driver versions. Due to internal issues related to NVIDIA drivers, all users of VisionPro Deep Learning 1.1.0 MUST use NVIDIA driver that is of:
  - 461.09 or higher version for NVIDIA GeForce® series.
  - R460 U2 (461.09) or higher version for NVIDIA RTX / Quadro® series.
- 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.
  1. Remove the currently installed NVIDIA driver from your PC.
  2. Reboot your PC and install the DCH NVIDIA driver of which version is compatible with VisionPro Deep Learning 1.1.0 (461.09 or higher, or R460 U2 (461.09) or higher for NVIDIA RTX / Quadro®)

### Operating System (OS)

- Due to OS compatibility issues caused from NVIDIA drivers, Windows Server 2016 is not supported on VisionPro Deep Learning. 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 servers. Windows Server 2019 is only supported for 'servers' configured to use the Deep Learning Client/Server functionality.

### Cognex Deep Learning Studio (GUI)

- 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.

### Workspace

- Boost::filesystem::rename 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, but you cannot modify this.
- It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio.
- 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.

## Tools - General

- If a tool is not saved well because of unspecific reasons, you cannot export or clone the workspace or the tool. You can avoid this problem by processing the tools again and save the workspace.
- For all tools, if the tool crashes with some unknown reason, images in this tool are deleted when the workspace that contains this tool is imported.
- If the name of a set in Database Overview contains a single quote ('), you cannot change the name of this set.
- If the name of a set in Database Overview contains a single quote ('), an error occurs when you try to filter views on View Browser by clicking the icon of this set in Database Overview.

## Tools - Red Analyze

- 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, 8192x8192 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
- 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, 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 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 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.
- 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.

- 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.

## Tools - Green Classify

- Once you are done with training High Detail mode of Green Classify Tool, you cannot change the name of the class.

## Tools - Blue Read

- For Blue Read, the markings are displayed on the view regardless of the Marking checkbox in Image File Information is whether enabled or disabled. This issue has existed since ViDi 4.1.

## 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.

## Migration from SuaKIT or The Past VisionPro Deep Learning Versions

- 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 VPD runtime API, please use SuaKIT resize API before processing.
- 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 Learning 1.1.

## Integration with VisionPro

- 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.

## Plug-in

- For Designer, the added heatmap checkbox is initialized on Deep Learning Runtime Blocks of the tools other than Red Analyze. Tools other than Red Analyze were supposed not to support heatmap function.

## VisionPro ViDi 3.1.0 ~ VisionPro Deep Learning 1.0.1

### GPU

- 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.

- Internal issues on supporting NVIDIA driver versions. Due to internal issues related to NVIDIA drivers, all users of VisionPro Deep Learning 1.0.1 MUST use NVIDIA driver that is of:
  - 457.09 or higher version for NVIDIA GeForce® series.
  - R460 U2 (461.09) or higher version for NVIDIA RTX / Quadro® series.
- 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.
  1. Remove the currently installed NVIDIA driver from your PC.
  2. Reboot your PC and install the DCH NVIDIA driver of which version is compatible with VisionPro Deep Learning 1.0.1 (457.09 or higher, or R460 U2 (461.09) or higher for NVIDIA RTX / Quadro®)
- 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.

## Operating System (OS)

- Due to OS compatibility issues caused from NVIDIA drivers, Windows Server 2016 is not supported on VisionPro Deep Learning. 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 servers. Windows Server 2019 is only supported for 'servers' configured to use the Deep Learning Client/Server functionality.

## Cognex Deep Learning Studio (GUI)

- 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.

## Workspace

- Boost::filesystem::rename 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, but you cannot modify this.
  - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio.
- 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.

## Tools - General

- If a tool is not saved well because of unspecific reasons, you cannot export or clone the workspace or the tool. You can avoid this problem by processing the tools again and save the workspace.

## Tools - Red Analyze

- 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, 8192x8192 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
- 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.
- 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.

## Tools - Green Classify

- Once you are done with training High Detail mode of Green Classify Tool, you cannot change the name of the class.

## Tools - Blue Read

- For Blue Read, the markings are displayed on the view regardless of the Marking checkbox in Image File Information is whether enabled or disabled. This issue has existed since ViDi 4.1.

## 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.

## Migration from SuaKIT or The Past VisionPro Deep Learning Versions

- 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.

## Integration with VisionPro

- 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.

## Migrating to This Release

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

### Migrating to VisionPro Deep Learning 4.0

Standard type tools are not compatible with the tools created in previous versions of VisionPro Deep Learning. The following changes apply when upgrading to 4.0:

- High Detail architecture tools are discarded and set to Standard type Accurate mode.
- Focused architecture tools keep the same conditions as the existing tool. For example, a tool trained with Focused architecture and medium network model is set to Legacy type medium network model.
- If using a High Detail architecture tool in the runtime workspace, you must retrain the tool in VisionPro Deep Learning 4.0.

In workspaces previously made with multiclass segmentation, the application unifies the class color to red. You can select each class and assign different colors to them.

### Migrating to VisionPro Deep Learning 3.3.0

- From the VisionPro Deep Learning 3.3.0 release, all tools must be retrained after each version update, including minor version updates.
- The High Detail Quick architecture of the Green Classify tool is deprecated.
- The *Extra Large* option of the **Network Model** parameter is deprecated for the Red Analyze and Green ClassifyStandard tools.
- When you import a previously created workspace to VisionPro Deep Learning 3.3.0, **Maximum Epochs** and **Patience Epochs** are imported as default parameters.

### 3.0 → 3.1

#### Re-optimization Required When Using TensorRT

When using TensorRT with VisionPro Deep Learning 3.1, please note that even if you have already optimized TensorRT with VisionPro Deep Learning 3.0, you must optimize it again for VisionPro Deep Learning 3.1.

### 2.1 (2.1.1) → 3.0

#### Removal of "Extra Large" from Green Classify Standard Network Models

Due to the Green Classify Standard architecture update in VisionPro Deep Learning 3.0, the Extra Large option in Network Model parameters for a Green Classify Standard is removed from VisionPro Deep Learning. If you want to use a Green Classify Standard in VisionPro Deep Learning 3.0, saved with the Extra Large network size in the previous VisionPro Deep Learning versions, import the tool and change its Network Model parameter from a blank to another one (Small, Normal, or Large).

#### Removal of Red Analyze High Detail Quick

Due to internal reasons, Red Analyze High Detail Quick is not supported in VisionPro Deep Learning 3.0. When you open a Red Analyze High Detail Quick tool trained in an earlier VisionPro Deep Learning version on VisionPro Deep Learning 3.0:

- This tool will be opened as a Red Analyze Standard tool.
- The trained network of this tool will be lost.
- The views and their labels of this tool will be preserved.
- The Tool Parameter values will be reset to the default values of a Red Analyze Standard tool.

Also, importing a Red Analyze High Detail Quick tool trained in an earlier VisionPro Deep Learning version into a VisionPro Deep Learning 3.0 workspace is disabled.

## Standard Backwards Compatibility

When you open a Red Analyze Standard or Green Classify Standard tool trained in an earlier VisionPro Deep Learning version on VisionPro Deep Learning 3.0:

- The trained network of this tool will be lost.
- The views and their labels of this tool will be preserved.
- The Tool Parameter values will be preserved.

When you import a Red Analyze Standard tool (.vrta) or a Green Classify Standard tool (.vgta) trained in an earlier VisionPro Deep Learning version into a VisionPro Deep Learning 3.0 workspace:

- The trained network of this tool will be lost.
- The views and their labels of this tool will be preserved.
- The Tool Parameter values will be preserved.

## Runtime Workspace Backwards Compatibility

The runtime workspace created in the earlier versions of VisionPro Deep Learning 3.0 that contains a Red Analyze Standard, Red Analyze High Detail Quick, Green Classify Standard, or Green Classify High Detail Quick tool can't be imported in VisionPro Deep Learning 3.0. In this case, you need to create a new runtime workspace in VisionPro Deep Learning 3.0.

## SuaKIT Segmentation and Classification Projects Compatibility

As it becomes unable to import the Standard tools created and trained in the earlier versions of VisionPro Deep Learning 3.0, it is also not available to import the trained models (neural networks) of existing segmentation or classification projects from SuaKIT into VisionPro Deep Learning 3.0.

## 2.0 → 2.1 (2.1.1)

### Termination of CIE1 Deployment Support

The CIE1 deployment target is officially no longer supported by VisionPro Deep Learning 2.1 as VisionPro Deep Learning does not support smart cameras. The support for CIE1 deployment target has unofficially been terminated since VisionPro Deep Learning 2.0, but the option was visible in VisionPro Deep Learning GUI in 2.0.

## 1.1 → 2.0

### Termination 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.

## 1.0.1 → 1.1

### 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.

## 1.0 → 1.0.1

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

## VisionPro ViDi 4.1 → VisionPro Deep Learning 1.0

### 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.

### 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.

## 3.4.1 → 4.1

### Changes to the Feature Filter Syntax

In the VisionPro ViDi 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 VisionPro ViDi 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 VisionPro ViDi 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 VisionPro ViDi 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 Train Neural Network section of each tool.

For the Blue Locate Tool, a change was made in the behavior of X and Y position clamping when using model matching. In VisionPro ViDi version 2.1, these constraints were interpreted to apply to the position of the origin of the model coordinates. In version VisionPro ViDi 3.0, they are interpreted as constraints on the position of the centroid of the nodes in the model.

# Release History

This topic lists the features and resolved issues in previous releases of VisionPro Deep Learning.

## VisionPro Deep Learning 4.0

- Introduced Few Sample mode for the Standard type Green Classify tool.  
This mode provides high performance with a small set of images, sometimes fewer than ten images per class.

**Note:** Few Sample mode does not support the outlier score.

- Introduced Robust mode for the Standard type tools.  
This mode allows the tool to adapt to changes on the production line and to product variants that have similar kinds of defects. Select this mode if you want to use the tool on different production lines and new products without retraining the tool.

**Note:** Robust mode replaces Adaptation mode in VisionPro Deep Learning. Adaptation mode is deprecated and removed from the application.

- Introduced outlier score for the Standard type Green Classify tool.  
The outlier score shows how much a view deviates from the other views in the training set. A high outlier score can indicate that an anomaly happened on the production line, for example, the lighting conditions changed.  
In the display filter, filter the score with the `outlier_score` syntax.
- Introduced the Accelerated Training as an expert mode parameter for Standard type tools.  
Accelerated training optimizes GPU resources and can contribute to speed improvements, but performance degradation can occur depending on the data.
- Introduced multiclass segmentation for the Standard and Legacy type Red Analyze tools. Multiclass segmentation is no longer a preview-mode feature.
- Introduced the Int8 option to Tensor RT speed optimization. The Int8 option improves processing time but it can reduce performance. You can also check the runtime processing time in Cognex Deep Learning Studio.

## Renewed Tool Categorization

VisionPro Deep Learning 4.0 changed the categorization of the tools. From this version, the tools have the categories of type and mode instead of architectures.

Standard type tools are not compatible with the tools created in previous versions of VisionPro Deep Learning. The following changes apply when upgrading to 4.0:

- High Detail architecture tools are discarded and set to Standard type Accurate mode.
- Focused architecture tools keep the same conditions as the existing tool. For example, a tool trained with Focused architecture and medium network model is set to Legacy type medium network model.
- If using a High Detail architecture tool in the runtime workspace, you must retrain the tool in VisionPro Deep Learning 4.0.

The new tool types:

Tool	Type	Mode	Network Model	Corresponding 3.3 Network

Red Analyze	Standard	Fast	N/A	Red Analyze High Detail architecture, small network model
		Accurate		Red Analyze High Detail architecture, normal network model
		Robust		N/A
	Unsupervised	N/A	Small	Red Analyze Unsupervised architecture, small network model
			Medium	Red Analyze Unsupervised architecture, normal network model
			Large	Red Analyze Unsupervised architecture, large network model
	Legacy	N/A	Small	Red Analyze Supervised architecture, small network model
			Medium	Red Analyze Supervised architecture, normal network model
			Large	Red Analyze Supervised architecture, large network model
Green Classify	Standard	Fast	N/A	Green Classify High Detail architecture, small network model
		Accurate		Green Classify High Detail architecture, normal network model
		Robust		N/A
		Few Sample		N/A
	Legacy	N/A	Small	Green Classify Supervised, small network model
			Medium	Green Classify Supervised, normal network model
			Large	Green Classify Supervised, large network model

## Enhancements

- Improved the grid ROI rule. The grid divides the ROI into several views. The following rules apply to all views that the grid creates:
  - Green Classify tool: The new views have the same class as the original view.
  - Red Analyze tool: The region remains in the same position as the label of the original view.
    - If a new view does not have a drawn region after you apply the grid, the view is labeled as Good.
    - If the original view did not have a label before applying the grid, the new views will not have a label either.
  - Blue Locate and Blue Read tools: The tools find the feature based on the new view.
- When adding a downstream tool after the Blue Locate tool, the default value for pixel is changed from 100x100 to fitted size from the Blue Locate tool when editing the ROI.
- The IOU metric is added to the Region Area Metrics in the **Database Overview**.
- Introduced shortcuts to the drawing tools in the Edit ROI, Edit Mask, and Edit Region toolbars.
- Added the `mismatch` display filter to the Red Analyze tool.

## Changes

- In previous versions, sometimes the processing speeds shown in Cognex Deep Learning Studio and runtime environments were different. From this version, both environments show the same processing speed.
- Heatmap for the Standard type Green Classify tool is available in the runtime environment.
- The **Speed Optimization** parameter is added to the **Tool Parameters** panel under the Processing section. This parameter is applicable for Standard type tools, and it is the same as Tensor RT speed optimization in the runtime environment.
- Introduced an error message that appears if you do not have the minimum required number of views for training or label quality check
- The **Threshold** parameter of the Red Analyze tool changed. The tool now uses a single threshold instead of T1 and T2.
- The default option for the GPU Memory Optimization option changed to disabled.
- When using the Green Classify tool, you can tag with the number keys. If you press the **1** number key, you can apply the first tag in the tag list.
- When using multiple defect classes with the Red Analyze tool, you can change between classes in the **Database Overview** using shortcuts. The Database Overview displays the Score, ROC, and Confusion Matrix only for the selected class. For example, pressing **ALT + CTRL + 1** selects the first class in the list, and pressing **ALT + CTRL + 0** selects the tenth class.
- You can check Untrained Views from the Region Area Metrics.

## Resolved Issues

This section describes resolved issues in the VisionPro Deep Learning4.0 release.

- For Standard type tools, memory usage increases during repeated processing with the runtime API.
- Fixed the issue where the Blue Locate tool slowed down when there were more than 400 features in one image. The UI temporarily froze during image scrolling and selection.
- Fixed the issue where training sometimes failed with the error `extracted view size is wrong`.
- Fixed the issue where the angle value of the ROI was not fixed when setting the ROI in the downstream tool after training the Blue Locate tool.
- Fixed the issue where the external area of a view was set as the view during grid ROI.
- Fixed the issue where an application error occurred when no view was selected in the list due to the display filter.
- Fixed the issue where the regions outside the ROI were used as labeled regions. Now, any regions outside the ROI are removed after training.


## VisionPro Deep Learning 3.3.0

- Improved Red Analyze and Green Classify High Detail tools

Compared to the previous versions, the improved Red Analyze and Green Classify High Detail tools can be trained to the same level of performance faster. This feature affects the following tool parameters of the Red Analyze and Green Classify High Detail tools:

- The **Maximum Epochs** expert mode training parameter is introduced.
  - The available value range for the **Patience Epochs** expert mode training parameter is changed to 0–100.
  - The **Epoch Count** and the **Minimum Epochs** training parameters are deprecated.
- Resizing option for Red AnalyzeHigh Detail tools

This release introduces the Resizing tool parameters to the Red AnalyzeHigh Detail tool to allow for resizing large images. With the Resizing parameters, VisionPro Deep Learning downsizes the image, the marking, and label sizes to make the training and processing time faster. Using smaller image sizes can decrease the accuracy of the tool.

 **Note:** The same functionality is also available through the API but it has a different name: adaptive downsampling.

- Import mask

This release introduces an import mask feature. With this feature, you can apply the same mask file to several views at the same time.

- Hand tool

This release introduces the hand tool to the **Edit Mask** and **Edit Region** editors. With the hand tool, you can pan through an image, allowing for more precise masking and region editing.

## Enhancements

- The processing time of Red Analyze High Detail tool is faster compared to previous releases.
- The *Small* option of the **Network Model** parameter is improved. The improved small network model size allows for faster processing.
- Resizing options are added to the Parameter Search utility for Green Classify High Detail tool. The resizing options are the same as the Resizing parameters available in the VisionPro Deep Learning application when training a Green ClassifyHigh Detail tool.
- The VisionPro Deep Learning *Software Help* document is updated with a restructured table of contents and revised sections.

## Changes

- From the VisionPro Deep Learning 3.3.0 release, all tools must be retrained after each version update, including minor version updates.
- From the VisionPro Deep Learning 3.3.0 release, Cognex provides a documentation installer that is separate from the product installer. To get the latest documentation, you must download and install the documentation installer when it becomes available after each release.
- VisionPro Deep Learning now supports Windows Server 2022.
- The High Detail Quick architecture of the Green Classify tool is deprecated.
- The **Downsampling Size** processing parameter of the Red AnalyzeHigh Detail tool is deprecated.

- The *Extra Large* option of the **Network Model** parameter is deprecated for the Red Analyze and Green ClassifyHigh Detail tools.
- The **Whole Image Processing** parameter and patch cropping processing is deprecated. From this release, all tools use whole image processing by default.
- The logic used when resizing images is changed. Now, regardless of the image size ratio, VisionPro Deep Learning sets the ratio based on the maximum image height and width. This change only affects Green Classify High Detail tools.

## Resolved Issues

- Fixed an issue where the scores in the API and the scores in the GUI were different for Green Classify High Detail tools.
- Fixed an issue where the validation set was not displayed when switching to High Detail from Focused after configuring the training set.
- Fixed an issue with the Cognex Label Import/Export Plug-in where the software crashed when importing approximately 11000 labels.
- Fixed the following issues related to resizing images with different aspect ratios using the Green Classify High Detail tool:
  - When refreshing the Resizing parameters, the *Max Buffer Size* parameter is updated, but the input field for the *Resize to* parameter remains unchanged.
  - When clicking the **See preview** link, the view size displayed in the popup window is not updated.
- Fixed an issue related to getting an error when importing a workspace where training was conducted with Korean characters in the image names.
- Fixed an issue where an out of memory error occurred when using a runtime workspace from versions prior to VisionPro Deep Learning 3.2 after TensorRT optimization in VisionPro Deep Learning 3.2 during API processing, even when the image sizes were the same.
- The range for inputting parameter values in the Parameter Search utility is standardized across the UI.

## VisionPro Deep Learning 3.2.1

VisionPro Deep Learning 3.2.1 is a hotfix release of the VisionPro Deep Learning 3.2 release.

### Enhancements

- The AI Labeling Tool can be used in the service client environment.
- The AI Labeling Tool has the undo function.
- The Green Classify High Detail and the Green Classify High Detail Quick tools also support processing with images smaller than 64 x 64 pixels.
- The Green Classify High Detail and the Green Classify High Detail Quick tools also support training only with images smaller than 64 x 64 pixels.

### Resolved Issues

- Fixed an issue where the 100 x 100 ROI value set by the user was randomized in the ROI setting window.
- Fixed an issue where the VisionPro Deep Learning software was slow when loading the database overview when the number of classes was eight or more.

- Fixed an issue where the resize size set by the user could not be imported into the Parameter Search in Auto mode. Previously, the manually set resize size was overwritten by the size calculated by Parameter Search, but now this does not happen anymore.
- Fixed an issue with the Green Classify High Detail and the Green Classify High Detail Quick tools where the score was different between the training API and the runtime API. Now both show the same score.
- Fixed an issue with the Green Classify High Detail Quick tool where the marking did not change to label after clearing a label in a view which had marking and accepting the view.
- Fixed a compatibility issue between VisionPro Deep Learning 3.2.0 and VisionPro 9.20
- Fixed an issue where the Green Classify High Detail mode parameter Validation Set Ratio could not be changed from 20%.

## VisionPro Deep Learning 3.2.0

- AI labeling tool  
In the AI labeling tool, you can use two key tools: AI Click Tool and AI Box Tool. These tools simplify labeling by automatically recognizing and labeling regions, eliminating the requirement for manual drawing.
- Red Analyze processing speed  
The processing speed of the Red Analyze was enhanced, particularly in Red Analyze High Detail.
- Faster image addition  
Adding images takes reduced time, resulting in faster handling when adding a large number of images.

## Enhancements

- GPU Clock Stabilizer  
The feature to maintain a stable GPU clock in a deployment environment can be helpful in multi-GPU scenarios or in situations where clock instability leads to reduced processing speed or spikes. For more details, see [API Changelog on page 34](#) or the API documentation.

## Resolved Issues

- Fixed an issue where CPU memory gradually increased when repeating training and processing of High Detail.
- Fixed an issue where an internal error occurred during the creation of a runtime workspace, preventing its creation when there was a previously trained model using the large network in the Green Classify High Detail from versions prior to VisionPro Deep Learning 3.1.1.
- Fixed an issue where High Detail was not supported in ViDi.ITool.RegionOfInterest.
- Fixed an error when training tools with a 1-channel 16-bit image.
- Fixed an internal error that occurred during the Parameter Search when starting the search for Green Classify High Detail with Large model.

## VisionPro Deep Learning 3.1.2

- Hotfix release of VisionPro Deep Learning 3.1.

## Resolved Issues

- Fixed an issue where tools trained using Green Classify High Detail tool's "large" modes in VisionPro Deep Learning 3.0 were not processing in VisionPro Deep Learning 3.1.1.

- With the use of CLIP, you can now import a JSON file containing a "bad" tag without regions into Red Analyze.
- Using CLIP, fixed an issue where the region name of a "bad" label in the Red Analyze Legacy appeared as "bad" instead of "defect" during label export and import.
- Using CLIP, fixed an issue of JSON file format labels not being imported in the Green Classify.
- The tooltip for the "mode" parameter in the High Detail parameters has been updated.

## VisionPro Deep Learning 3.1.1

- Windows 11 is now supported
  - Now VisionPro Deep Learning 4.1 supports Windows 11
- RTX™ 40 series are tested
  - VisionPro Deep Learning supports on RTX™ 40 series which are RTX 4070 Ti, RTX 4080 and RTX 4090. It is tested on RTX 4080 and 4090.
- Minor improvements
  - Improved the processing speed for High Detail tool by optimizing the internal logic, including the mask functionality.
  - Due to internal logic improvements, the training speed of Red Analyze High Detail tool has been partially enhanced.
  - In the Auto mode of Parameter search, remaining and elapsed time are displayed during the search.

## Resolved Issues

- Fixed an issue in Red Analyze High Detail where the view score would display as 0 after applying a mask and processing with the API.
- Fixed an issue in the Red Analyze High Detail where the processing results differed when using Whole Image processing, although differences may still occur within an acceptable margin of error.
- Fixed an issue where out of memory pop-up occurred during processing after completing training with Whole Image Processing on in Red Analyze High Detail, which prevented saving of the model. The tool now automatically turns off Whole Image Processing and proceeds with processing if it cannot perform the function.
- Fixed an issue where the application would intermittently crash when loading a runtime workspace with multiple High Detail and using multi-GPU.
- Fixed an issue where the Reuse mask option was not applied in the ROI settings for downstream tools when Red Analyze was the preceding tool.
- Fixed an issue where the label check did not run when using Green Classify with more than three tags.
- Restored the pan function that was missing in the image viewer.
- Fixed an issue where Cognex Deep Learning Studio abnormally terminates when clicking on an empty column in the image list.
- Fixed an issue where Cognex Deep Learning Studio abnormally terminates when entering an invalid color value for the tag after setting the tag in Green Classify.
- Updated the tooltip description regarding the processing time displayed on the Cognex Deep Learning Studio, as it was previously shown in view unit instead of image unit.
- Fixed an issue where entering -1 into the command line failed to correctly deactivate the auto-save feature in the service-client

- Fixed an issue where the C++ sample code, Example.AdaptationMode.HD, was not functioning properly.
- Fixed an issue that caused an error when attempting to switch from Auto mode to Manual mode in Parameter search, if there were spaces in the tags.
- Fixed an issue where results were not displayed after conducting a search in auto mode with a tool in the downstream in Parameter search.

## VisionPro Deep Learning 3.1.0 (Beta)

- Beta release for Window 11

## VisionPro Deep Learning 3.0.1

- Adaptation Mode for Standard tools
  - Adaptation Mode is supported only for Red Analyze and Green Classify High Detail tool.
- Red Analyze High Detail and Green Classify High Detail Architecture Upgrade
  - The neural network architecture of Red Analyze High Detail and Green Classify High Detail is updated.
- Sampling without "Stride"
  - The sampling logic for Red Analyze High Detail training evolved that it can deliver the same performance without the Stride-based sampling. Due to this upgrade, the Stride in the Training parameters of Red Analyze High Detail is deprecated from VisionPro Deep Learning 3.0.
- Backwards Compatibility
- Change of Validation Loss Calculation
  - Another change from this architecture update is the new validation loss calculation for Red Analyze High Detail, which is used to select the best neural network during validation.
- Automatic Parameter Search
  - While the manual mode tests all combinations of the listed parameters and their value ranges, the auto mode cuts off unnecessary tests based on performance criteria and thus it can save more time for getting you the best values of provided parameters. The auto mode of is only supported for tools. See Parameter Search for details.
- Label Check
  - The label quality value is scaled from 0 to 100. The lower the value, the noisier or the worse each label is. You can dramatically reduce the time spent reviewing the labels by sorting images in ascending order by the label quality value. Label Check is only supported for Red Analyze Legacy, Red Analyze High Detail, and all Green Classify tool types.
- Cognex Label Import/Export Plugin (CLIP)
  - CLIP is a new plug-in that supports importing/exporting masks and labels of a VisionPro Deep Learning workspace in VisionPro Deep Learning internal and external formats. You can export masks, labels, or images in a VisionPro Deep Learning workspace in VisionPro Deep Learning internal format or png file format, and also import masks or labels in various formats into a VisionPro Deep Learning workspace with this plug-in.
- TensorRT Support for Runtime

- [The NVIDIA TensorRT](#) is a library that boosts the inference speed of a deep neural network running on NVIDIA GPUs. The VisionPro Deep Learning now supports TensorRT for the runtime API, boosting the processing speed of each tool. TensorRT for Runtime is only supported in High Detail tools.
- Green Classify Standard Batch Processing Support for Runtime
  - Using the Batch Processing API in runtime can speed up the Green Classify High Detail processing via API.
- View Browser - List Mode
  - The list mode presents an information table that offers much more details about a view's status including marking, label, sets, or a view's index. You also can star a view if it is an important view or write and edit a memo for each view.
- Other Improvements & Changes
  - UI/UX Improvements in Parameter Search
  - GUI Refinement
  - Applying Label Change for Green Classify
  - Image Resize for Green Classify High Detail
  - Validation Set Indicator and Filter for Validation Set
  - Scale Change of Validation Loss
  - Image Processing Logic Change for Red Analyze High Detail
  - Minimum Image Size Change for Green Classify High Detail
  - Training Logic Change with Patience Epochs and Minimum Epochs
  - Designer Plug-in Support
  - Red Analyze sort filter keyword `count(region!)` in View Browser which sort the views by the number of labeled defect regions is changed to `count(region![area > 0])`.

## VisionPro Deep Learning 2.1.1

- Fixed an issue that the execution of VisionPro 10.1 C# API method `Cognex.Vision.Startup.Shutdown()` is not completed within an expected time when using VisionPro Deep Learning Tool Block.
- Fixed an issue that the Focused tools (Blue Locate, Blue Read, Green Classify Focused, Red Analyze Focused Supervised, Red Analyze Focused Unsupervised) occasionally expose inconsistent results given exactly the same configurations.
- Fixed an issue that the processing of Blue Locate or Blue Read rarely gets slower.
  - To solve this issue, we allowed access to a new processing parameter called `max_scan_iterations` in the runtime API. With this parameter, you can limit the maximum search time to speed up the processing.

## Resolved Issues

- Fixed an issue that the execution of VisionPro 10.1 C# API method `Cognex.Vision.Startup.Shutdown()` is not completed within an expected time when using VisionPro Deep Learning Tool Block.
- Fixed an issue that the Focused tools (Blue Locate, Blue Read, Green Classify Focused, Red Analyze Focused Supervised, Red Analyze Focused Unsupervised) occasionally expose inconsistent results given exactly the same configurations.
- Fixed an issue that the processing of Blue Locate or Blue Read rarely gets slower.

## VisionPro Deep Learning 2.1

- Red Analyze High Detail Quick and Green Classify High Detail Quick
  - Faster than High Detail modes in training
  - Stay solid in task accuracy
  - More convenient with much less Tool Parameters to be tuned by hand.
- Revamped Parameter Search
  - The user interface of Parameter Search has been completely renovated to enhance the usability of the tool.
- Heatmap Overlay on Image Display Area for Green Classify High Detail and Green Classify High Detail Quick
  - For Green Classify High Detail and Green Classify High Detail Quick, the heatmaps are not only visible through View Inspector but directly in Image Display Area by enabling Heatmap checkbox in Image File Information.
- Green Classify API Example Codes are updated for getting label information from images (RAPID XML)
  - C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c++\Example.Training.HDGreen
  - C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c++\Example.Training.Green
- Improved tool loading speed into Workspace for a Red Analyze tool when it has significantly many detected regions (more than hundreds or thousands of regions) or regions with complex shapes per image.
- Enhanced document contents.

## Resolved Issues

- Fixed an issue that some runtime API example codes were unable to be run because of missing DLL files.
- Fixed an issue in High Detail Mode tools that the training time keeps increasing when training multiple High Detail mode tools in sequence.
- Fixed an issue in Parameter Search that searching parameters for a Red Analyze tool already trained in VisionPro Deep Learning unnecessarily copies existing .vrb files and thus grows the size of the current workspace.
- Fixed an issue in C API that 32-bit images (8-bit, 4 channel) added to a workspace by C API look bizarre when they are viewed in GUI.
- Fixed an issue in Loss Graph that the Elapsed Time in Status table is not in accordance with the elapsed time in Progress bar.
- Fixed an issue in Loss Graph that the Elapsed Epoch in Status table and Epoch in the loss graph are displayed incorrectly.
- Fixed an issue that the value of ViDi2.View.maskCache keeps increasing and consumes memory.
- For High Detail modes, fixed an issue that processing with GPU designation is not available when calling ISample.Process().
- For VisionPro Deep Learning Client/Server environment, fixed an issue that VisionPro Deep Learning Client/Server cannot accept "16-bit depth, png/tiff format" image file as its input image.
- For runtime workspaces, fixed an issue that users cannot get the label information of Green Classify High Detail with Runtime.IGreenTool.KnownTags
- For Red Analyze, fixed an issue that after changing mode of tool operation (Supervised, Unsupervised) the class of a region is displayed incorrectly.

- For Red Analyze, fixed an issue that after changing mode of tool operation (Supervised, Unsupervised) a region cannot be modified with Edit Region.
- For Red Analyze High Detail, fixed an issue that if you change the mode of a Red Analyze tool to Red Analyze High Detail, add images, set ROI, and train with the default tool parameters, the score for a view is not generated.

## VisionPro Deep Learning 2.0

- 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).
- Cognex Designer Support Termination
  - 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.

## Resolved Issues

- 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.

## VisionPro Deep Learning 1.1

- Red Analyze High Detail
  - A new solution with a powerful new architecture for supervised segmentation tasks that were previously only handled by Red Analyze Supervised Mode. With the addition of Red Analyze High Detail Mode, Red Analyze tools are now reorganized and fall into the following category:

Name	Architecture	Mode
Red Analyze High Detail	High Detail	Supervised
Red Analyze Supervised	Focused	Supervised
Red Analyze Unsupervised	Focused	Unsupervised

- Loss Inspector
  - A monitoring tool of the validation loss change during training that guides you in training High Detail Modes (Red Analyze High Detail and Green Classify High Detail mode).
- F1 Score of Confusion Matrix for Red Analyze tools
  - The F1 score of the confusion matrix is introduced to show the result summary of the confusion matrix in Red Analyze tools.
  - Unlike the precision, recall, and F-score in Region Area Matrix which are calculated on a "pixel" basis, the F1 score of a confusion matrix is calculated from each "item in the confusion matrix table", aiming to suggest the performance of this tool with respect to the confusion matrix result.
- Auto Threshold for Red Analyze High Detail and Red Analyze Legacy
  - The T1 and T2 threshold values that maximize the **F1 score of confusion matrix** on Database Overview by following one of the 4 criteria in the Count dropdown menu. These threshold values are automatically calculated for you by enabling a checkbox.

- Region List for Red Analyze tools
  - The region list is introduced to help you find and check the defect regions (both labeled defects and markings) in a view more easily and conveniently and thus enhance your usability of Red Analyze tools.
- Overlay Checkbox
  - For all tools, the overlay checkbox is introduced for a handy alternative to "Cycle through Overlays" in the right-click pop-up menu on Image Display Area.
- Restore Parameters
  - For all tools, the restore parameter button is introduced for an easier "roll-back" of the set of parameter values in the Tool Parameters pane toward those that were used in the last training session.
- Wibu CodeMeter Runtime update from 6.70a to 7.20b to resolve [the known security vulnerabilities](#) (WIBU-200521-01, WIBU-200521-02, WIBU-200521-03, WIBU-200521-05, WIBU-200521-06).

## Resolved Issues

- Fixed an issue in Parameter Search that some texts in Database Overview are overlapped to each other and displayed incorrectly.
- Multiple GPUs are now fully supported efficiently for High Detail modes.
- Fixed a bug that the multiple additions of the same plugin with Manage Plugins.
- Fixed a bug that VisionPro Deep Learning is not launched when the version of nvml.dll (C:\Windows\System32\nvml.dll) is lower than the NVIDIA graphics driver version.
- Fixed a bug in Blue Read that the overlay of marking and label is not correctly displayed.
- Fixed a bug in Red Analyze that the area under ROC curve is different from AUC value in Database Overview - ROC tab.
- Fixed an issue in Designer for Red Analyze that the workflow of Deep Learning Runtime Block of Red Analyze slows down due to heatmap processing. It is fixed by adding a 'heatmap' checkbox and the users now can control the processing speed by enabling/disabling the checkbox.
- Fixed some naming inconsistencies in C:\ProgramData\Cognex\VisionPro Deep Learning\1.1\Examples\c#
  - The assembly name for Example.Training.HDRed.Console is modified to the right name.
  - The project file name "Example.ViDi.Suite.GUI.Plugin" is changed to "Example.Deep.Learning.Studio.GUI.Plugin"
  - The assembly name with "space" delimiter is changed to the one with "period/full-stop" delimiter.

## VisionPro Deep Learning 1.0.1

- Support for NVIDIA GeForce® RTX™ 30 series.
  - VisionPro Deep Learning 1.0.1 supports RTX 3090, RTX 3080 and RTX 3070 for all tasks.
  - If you're exploiting multiple GPUs, all of them should consist of RTX™ 30 series of the same type and of the same specification.
  - Currently, there are some known issues relating to NVIDIA driver versions and Processing Time.
- Support for Optimized GPU Memory in Parameter Search

- If you use Focused Mode in Parameter Search, you can set and use Optimized GPU Memory.
- If you use High Detail Mode in Parameter Search, Optimized GPU Memory is automatically disabled for the efficiency of High Detail Mode.

## Resolved Issues

- Resolved an issue that the prediction result of a cloned Red Analyze Tool is incorrect when Low Precision is enabled.
- Resolved an issue that labeling is mistakenly applied on images, not views, after receiving views from an upstream tool.
- Resolved an issue that the space between "Image Record" blocks you from using ImageRecord feature with DeepLearningRuntimeBlock inside Designer.

## VisionPro Deep Learning 1.0

- 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.

## Resolved Issues

- 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.

## VisionPro ViDi 4.1.0

- Additional Blue Locate tool enhancements, including the following:
  - New handles on the feature graphic labels to make resizing easier.
  - Select either rectangular or circular feature graphic labels.
  - Display or hide the name of feature graphic labels.
  - Copy and paste feature label graphics, as well as cloning a feature graphic label by clicking on it.
  - Auto-generate the creation of a layout model by selecting features and selecting Create Model.
- Additional configuration options for assigning the GPU mode and memory, along with an auto-save feature and training image cache path, can now be set from the Options dialog, which is available in the Starting Cognex ViDi Suite startup dialog.
- The ViDi Suite GUI has been localized into six languages: French, German, Japanese, Korean, Simplified Chinese and Spanish. The language can be selected from the Locale drop-down within the Options dialog.

## Resolved Issues

- To improve the ability to determine if a view is labeled or not, a new property, `IView.IsLabeled`, has been added to the .NET API.
- Resolved an issue with large images and the Red Analyze tool in Unsupervised mode that could result in an invalid exception "Database is Locked" error.

## VisionPro ViDi 3.4.1

- Blue Locate tool  
The Blue Locate tool has undergone significant overhaul, including the following:
  - Create a Layout Model, which provides presence/absence verification of features in areas of an image.
  - New label, marking and model graphics to both increase visibility and usability by making labeled and marked features clearer, and handles to enhance graphic interaction when moving and resizing labels.
  - Feature identifiers can now be up to 140 characters.
  - Improved capabilities for oriented and scaled features.
  - Improved workflow when labeling images/views, with the tool remembering the previously labeled feature, allowing you to start with one image, create a label, and then apply that label to remainder of your images.
  - Improved performance of node models, along with a new Model Editor GUI.
  - The tool can now create a region of interest for downstream tools based on a feature, a match, or a node or layout model.
- Training workflow
  - There is a new paradigm for defining the images/views that will be used to train tools, allowing the process to be more transparent and controllable through integration with the new image sample sets.
- Image sample sets
  - The ability to create and manage images/views as training and testing sample sets has been added, which allows for better management of the images/views used to train and test tools.
- Database Overview filtering

- While in Expert Mode, it is now possible to filter results and statistics in the Database Overview, using the syntax of the Display filters.
- Blue Read tool
  - Capability to identify and read rotated characters.
- Red Analyze tool
  - An eraser tool has been added which allows you to remove areas of defect regions and masks.
- In addition to the above new features, there have been numerous performance and stability improvements.

## Resolved Issues

- Resolved an issue where exporting or cloning a large workspace could cause the ViDi Suite GUI to become unresponsive or unexpectedly quit.
- Resolved an issue where a Blue Locate or Blue Read tool's graphical display of markings could be offset from its actual location due to the language settings of the user's PC.
- Resolved an issue with the Blue Read tool's Label Views dialog not displaying the names of Regex models that had been created.
- Resolved an issue with the Red Analyze tool's generated reports not displaying the markings in the report.

## VisionPro ViDi 3.3.0

- General stability and performance improvements.
- Improved support for NVIDIA® Turing™ Tensor Cores for the Red Analyze and Green Classify tools with the Low Precision parameter enabled.
- Two new model types have been added for the Blue Read tool.
  - A regular expression model, which allows you to specify string matches based on regular expressions. This type of model offers faster, more flexible model matching for applications with linear arrangements of text than the existing node model.
  - A string model, which allows you to match linear strings based on the length of the text string.
- Improved performance for image displays that include a large number (500+) of feature graphics.
- Improved readability of the Green Classify tool's class labels when creating reports for the tool.
- Simplified controls for rotating a tool's Region of Interest (ROI).
- Support for Cognex VisionPro version 9.5 has been added.

## Resolved Issues

Issue	Description
AIC-1861	Corrected an issue with NVIDIA drivers 411.31 and newer. For more information, please visit the Cognex Support Site VisionPro ViDi Software Driver Support page.
AIC-1801	The Set Group option has been restored to the Actions for N Views menu in the View Browser, which allows you to create a group of images when using the Red Analyze tool in Unsupervised Mode.
AIC-1201	Corrected an issue where a workspace would not function correctly if a stream contained a single quote in the name.

Issue	Description
AIC-1223	Corrected an issue where the ViDi Suite GUI would become unresponsive when applying a region for a Red Analyze tool with numerous views.
AIC-1387	Corrected an issue where the ViDi Suite GUI would not terminate properly after attempting to connect to a runtime-only service.

## VisionPro ViDi 3.2.1

- This release introduces the ViDi Client/Server functionality, allows you to use the training and runtime implementations of the ViDi software on multiple PCs, through the use of a server/client model. This functionality gives you the option of using a remote service for runtime inference and/or training, instead of using the GPU on your local PC. This can be useful if you have a more powerful GPU on the server, or perhaps you do not have a GPU installed locally.
- This functionality supports up to 4 clients for deployment operation, and up to 8 clients for development.
- This release provides enhancements and speed improvements for the ViDi runtime:
  - Support for NVIDIA® Tensor Cores has been added, and the GPU Optimization feature also now supports Tensor Cores.
  - The Low Precision parameter has been added to the Red Analyze and Green Classify tools, which optimizes the tools for speed during runtime operation.
- The Feature Filter has been added to the Blue Locate and Blue Read tools, which allows you to filter found features by ID, X, Y, size and orientation.
- The ViDi GUI and the ViDi Help file have been translated into Simplified Chinese. To view the GUI and Help in Chinese, the OS language setting must be set to Simplified Chinese.
- Support for up to 32MP images (up to 16k in height or width) has been added.

## VisionPro ViDi 3.1.0

### Tool and GUI Changes

- The ViDi Blue Read Tool has been added for Optical Character Recognition (OCR) applications.
- Speed improvements and increased performance for the Blue Locate Tool when matching models.
- The Feature Size parameter is now interactive and can be graphically resized within the GUI.
- Labeling with the Blue tools has been improved to include multi-feature selection and interaction.
- The Blue tools have improved flexibility for model optimization.
- In the results, a Confusion Matrix has been added for the Blue Locate and Blue Read tools.
- A new feature/match and region-focus display mode has been added when using the Database Explorer.
- Checks have been added to provide feedback on wrongly labeled views.
- The examples have been updated, and there is a new example for the Blue Read tool.
- The Green Classify Tool now includes the Exclusive parameter, which, when disabled, allows you to label a single image with multiple labels (indicating that the image belongs to multiple classes).
- The Red Analyze Tool – Statistics have been enhanced for better readability.
- There have been numerous issues resolved, and additional enhancements and improvements to the ViDi software.

## Processing and Support Changes

- The GPU Memory Optimization is now activated by default (when using either the Windows Display Driver Model or Tesla Compute Cluster drivers). This feature can be activated and/or deactivated via command line arguments and the API.
- Command line arguments have been added to allow you to select the number of GPUs to utilize at startup of ViDi, to specify the GPU mode and to specify the amount of GPU memory to pre-allocate.
- There is a new Cognex VisionPro wrapper assembly, which provides data conversion between VisionPro and ViDi Suite, offering the following:
  - Efficient image conversion
  - Combined graphics
  - Fixturing conversion
  - Calibration conversion
- Support for Cognex Designer 4.0 has been added, with the addition of a ViDi Sub Task block, which contains the ViDi Suite tools and the ability to program the tools within the Cognex Designer IDE.
- Support for Cognex VisionPro 9.3 has been added.
- Support for Microsoft® Windows® 10 Professional (64-bit).
- Support for Microsoft® Windows® 7 Professional (64-bit).
- The performance when loading training Workspaces, utilizing on-demand loading of tools, has been improved.
- A new Workspace archive format, based on ZIP files, has been implemented.

# API Changelog

This topic lists the API changes in previous releases of VisionPro Deep Learning.

## VisionPro Deep Learning 4.1

### View

VisionPro Deep Learning 4.1 introduces the following view-related changes:

#### Added

The `ViDi2.IRedView.FilteredScores` (`ReadOnlyDictionary<string, double>`) C# parameter is added to support using filtered scores for each class.

#### Changed

The threshold is changed to be applied to each class. Add `TagThresholds` to use the threshold for each predicted tag.

The following C# parameters are added and changed:

- The `ViDi2.IGreenView.TagThresholds` (`ReadOnlyDictionary<string, double>`) parameter is added.
- `ViDi2.IGreenView.Threshold` (`double`)

**Note:** An exception is thrown only when the values of all tags in `TagThresholds` are not the same: The thresholds for each class are set differently. Use `IGreenView.TagThresholds` to verify.

### Tool Parameters

VisionPro Deep Learning 4.1 introduces the following tool parameter-related changes:

#### Added TagThresholds

Added the `TagThresholds` parameter to use the threshold for each predicted tag, implemented on the `IGreenStandardParameters` parameters.

The following C# parameters are changed:

- `ViDi2.Runtime.IGreenLegacyParameters.Threshold` (`double`)
- `ViDi2.Training.IGreenLegacyParameters.Threshold` (`double`)

**Note:** An exception is thrown only when the values of all tags in `TagThresholds` are not the same: The thresholds for each class are set differently. Use `IGreenView.TagThresholds` to verify.

- `ViDi2.Runtime.IGreenStandardParameters.ProbabilityThreshold` (`double`)
- `ViDi2.Training.IGreenStandardParameters.ProbabilityThreshold` (`double`)

**Note:** An exception is thrown only when the values of all tags in `TagThresholds` are not the same: The thresholds for each class are set differently. Use `IGreenView.TagThresholds` to verify.

The following C# parameters are added:

- `Vidi2.Runtime.IGreenLegacyParameters.TagThresholds` (`IRoDictionary<string, double>`)
- `Vidi2.Training.IGreenLegacyParameters.TagThresholds` (`IRoDictionary<string, double>`)
- `Vidi2.Runtime.IGreenStandardParameters.TagThresholds` (`IRoDictionary<string, double>`)
- `Vidi2.Training.IGreenStandardParameters.TagThresholds` (`IRoDictionary<string, double>`)

## Changed the Value Range of Epoch Counts

You can now set the epoch count to a minimum value of 1. In previous versions, the minimum value was 50.

The following parameters are changed:

- **C++ parameter:** `training_parameters/count_epochs`
- **C# parameter:** `Vidi2.Training.IToolParametersStandard.CountEpochs`

## Removed

The `training/red/exclusive` C++ parameter was removed.

## API Functions

VisionPro Deep Learning 4.1 introduces the following API function-related changes:

### Added

The following C# functions are added to directly import and export the tool archive as a file:

- `Vidi2.Training.ITool.ImportFromFile`
- `Vidi2.Training.ITool.ExportToFile`

The following functions are added to convert the heat map to the polygons or labels for the Red Analyze tool:

- **C++**
  - `vidi_training_green_heatmap_to_polygons`
  - `vidi_training_green_heatmap_to_labels`
- **C#**
  - `Vidi2.Training.IDatabase.HeatmapToPolygons`
  - `Vidi2.Training.IDatabase.HeatmapToLabels`

## Changed

The `fetch_from_cache(bool)` function is added to use cache when getting the resource image.

The following parameters are changed:

- **C++ parameter:** `vidi_resource_get_image`
- **C# parameter:** `Vidi.NET.Local.ILibraryAccess.ResourceGetImage`

The name of the following parameters changed:

- `gpu_hdm` to `gpu_standard`
- `gpuHdm` to `gpuStandard`

The following parameters are changed:

- **C++ parameter:** `vidi_runtime_open_workspace_from_file2`
- **C# parameter:** `ViDi2.Runtime.IWorkspaceList.Add`

The schema of the runtime workspace export option is changed. For more information, see `vidi_training_export_runtime_workspace_to_file2` or `vidi_training_export_runtime_workspace_to_memory2` in the API reference.

The `visual_debugger_on` C++ parameter is renamed to `heat_map_on`.

## VisionPro Deep Learning 4.0

### Introduced Tool Parameters and API Functions

#### Multiclass Segmentation

Introduced support for multiclass segmentation. The `threshold` parameter is deprecated and replaced by `region_thresholds` parameter.

C++ parameters (JSON-formatted strings):

- `post_process_parameters/red/region_thresholds`
- `post_process_parameters/red/region_colors`
- `post_processing/red/region_thresholds`
- `post_processing/red/region_colors`

C# parameters:

- `region_thresholds`
  - `ViDi2.Training.IRedStandardParameters.RegionThresholds`
  - `ViDi2.Training.IRedToolParameters.RegionThresholds`
  - `ViDi2.Training.IAnomalyDetectionParameters.RegionThresholds`
  - `ViDi2.Runtime.IRedStandardParameters.RegionThresholds`
  - `ViDi2.Runtime.IRedToolParameters.RegionThresholds`
  - `ViDi2.Runtime.IAnomalyDetectionParameters.RegionThresholds`
- `region_colors`
  - `ViDi2.Training.IRedStandardParameters.ClassColors`
  - `ViDi2.Training.IRedToolParameters.RegionColors`
  - `ViDi2.Training.IAnomalyDetectionParameters.RegionColors`
  - `ViDi2.Runtime.IRedStandardParameters.ClassColors`
  - `ViDi2.Runtime.IRedToolParameters.RegionColors`
  - `ViDi2.Runtime.IAnomalyDetectionParameters.RegionColors`

#### Heatmap for Training

Introduced support for heatmap in the training API.

- **C++ parameter:** `training_parameters/heat_map_on`
- **C# parameter:** `ViDi2.Training.IGreenStandardParameters.HeatMapOn`

## Outlier Score

Introduced support for outlier score in the training API.

- C++ parameter: `training_parameters/outlier_check_on`
- C# parameter: `ViDi2.Training.IGreenStandardParameters.OutlierOn`

## Accelerated Training

Introduced support for the accelerated training option in the training API.

- C++ parameter: `training_parameters/accelerated_training`
- C# parameter: `ViDi2.Training.IToolParametersStandard.AcceleratedTraining`

## Runtime Export Option for Outlier Score and Runtime Heatmap

Introduced runtime export option for Outlier Score and Runtime Heatmap.

C++ functions:

- `vidi_training_export_runtime_workspace_to_memory2`
- `vidi_training_export_runtime_workspace_to_file2`

Set the additional export option with the `export_option` parameter when you want to use heatmap or outlier score in runtime.

- @param `export_option`: You must provide a JSON-formatted string to set the `outlier_check_on` and `heat_map_on` flag for each tool.

C# functions:

- `ViDi2.Training.IWorkspace.ExportRuntimeWorkspace`
- `ViDi2.Training.IWorkspace.ExportRuntimeWorkspaceToFile`

## TensorRT conversion in training API

Introduced support for TensorRT conversion for both training and runtime. Run TensorRT conversion in the training API with this function. All parameters are the same as in the runtime API.

- C++ function: `vidi_training_tool_convert_trt`
- C# function: `ViDi2.Training.ITool.OptimizeTensorRT`

## Changed Tool Parameters and API Functions

### TensorRT related parameter changes

Replaced the boolean parameter `process_with_trt` with string parameter `trt_mode` to introduce the "int8" option for TensorRT optimization.

C++ parameters:

- `runtime_parameters/trt_mode` (string)
  - "none" option: Do not use TensorRT optimization.
  - "basic" option: Use default TensorRT optimization.
  - "int8" option: Use int8 TensorRT optimization.
- `vidi_training_tool_convert_trt`

- `vidi_runtime_tool_convert_trt`
- `@param mode` parameter: Optimization mode in string.
  - "basic" option: Same as the previous version without this parameter.
  - "int8" option: Mode for int8 TensorRT optimization.

C# parameters:

- `ViDi2.Training.IToolParametersStandard.TensorRTMode`
- `ViDi2.Runtime.IToolParametersStandard.TensorRTMode`
- `ViDi2.Training.ITool.OptimizeTensorRT`
- `ViDi2.Runtime.ITool.OptimizeTensorRT`

## Parameters Changed Related to Renewed Tool Categorization

Removed the `network_type` parameter and changed the `training_mode` parameter.

C++ parameters:

- `training_parameters/training_mode` (string)
  - "accurate" option: Same as "standard" in the previous version.
  - "fast" option: Same as "quick" in the previous version.
  - "robust" option: Replaces "adaptation". The parameters "basic" and "adaptation" are deprecated.
  - "fewsample" option: Use this option to train a Green Classify tool with a small sample set of images.
- `vidi_training_calculate_max_buffer_size`
  - "model\_capacity" option: Removed as we adopt renewed tool categorization.

C# parameters:

- `ViDi2.Training.IToolParametersStandard.Mode`
- `ViDi2.Training.IGreenTool.CalculateMaxBuffSize`

## Deprecated Features

The following features are deprecated:

- Similarity
- Incremental
- Adaptation
- Runtime heatmap

Heatmap is now supported for both training and runtime. To use heatmap in runtime, use the `export_option` parameter in `vidi_training_export_runtime_workspace_to_memory2` or the `vidi_training_export_runtime_workspace_to_file2` function.

- Incremental learning

For more information, see [API Deprecated Features on page 54](#).

## Updated Project Example Names

Due to the change in the type and mode of the tools, the names of the example code also changed between the VisionPro Deep Learning 3.3.0 and 4.0.0 versions.

### C# API Examples

Path: `C:\ProgramData\Cognex\VisionPro Deep Learning\4.0\Examples\c#`

Outdated Name	New Name
<code>Example.Runtime.HDGreen.Console</code>	<code>Example.Runtime.GreenStandard.Console</code>
<code>Example.Runtime.HDRed.Console</code>	<code>Example.Runtime.RedStandard.Console</code>
<code>Example.Training.HDGreen.Console</code>	<code>Example.Training.GreenStandard.Console</code>
<code>Example.Training.HDRed.Console</code>	<code>Example.Training.RedStandard.Console</code>
<code>Example.Runtime.HDGreen.Batched.Console</code>	<code>Example.Runtime.GreenStandard.Batched.Console</code>
<code>Example.Runtime.OptimizeHDTool.Console</code>	<code>Example.Runtime.OptimizeStandardTool.Console</code>

### C API Examples

Path: `C:\ProgramData\Cognex\VisionPro Deep Learning\4.0\Examples\c++`

Outdated Name	New Name
<code>Example.Runtime.HDGreen</code>	<code>Example.Runtime.GreenStandard</code>
<code>Example.Runtime.HDRed</code>	<code>Example.Runtime.RedStandard</code>
<code>Example.Training.HDGreen</code>	<code>Example.Training.GreenStandard</code>
<code>Example.Training.HDRed</code>	<code>Example.Training.RedStandard</code>
<code>Example.Runtime.HDGreen.Batched</code>	<code>Example.Runtime.GreenStandard.Batched</code>
<code>Example.Runtime.OptimizeHDTool</code>	<code>Example.Runtime.OptimizeStandardTool</code>

## VisionPro Deep Learning 3.3

### Deprecated *extra\_large* Capacity

- In the `vidi_training.h vidi_training_tool_train2` C++ function, the *extra\_large* option is no longer valid for the `training_artifact_key` parameter.
- In the `ViDi2.Training.IToolParametersHighDetail.NetworkModel` C# function, the *extra\_large* option is no longer valid.

### Deprecated *extra\_small* Capacity

- In the `vidi_training.h vidi_training_tool_train2` C++ function, the *extra\_small* option is no longer valid for the `training_artifact_key` parameter.
- In the `ViDi2.Training.IToolParametersHighDetail.NetworkModel` C# function, the *extra\_small* option is no longer valid.

## Deprecated *green\_quick* and *red\_quick* Tool Types

- In the `vidi_training.h vidi_runtime_set_parameter` C++ function, the `tool_type/green_quick` and `tool_type/red_quick` options are no longer valid.
- The `ViDi2.ITool.ToolType.GreenQuickHighDetail` and `ViDi2.ITool.ToolType.RedQuickHighDetail` C# function properties are no longer valid and will be removed from the future release.

## Deprecated *use\_patchcrop\_prediction* Parameter in *runtime\_parameters*

- In the `vidi_runtime.h vidi_runtime_set_parameter` and `vidi_runtime.h vidi.runtime_tool_set_parameter` C++ functions, the `runtime_parameters/use_patchcrop_prediction` parameter is no longer valid.
- The `ViDi2.Training.IRedHighDetailParameters.UsePatchcropPrediction` and `ViDi2.Runtime.IRedHighDetailParameters.UsePatchcropPrediction` C# function properties are removed.

## Deprecated *max\_downsampling\_size* Parameter in *runtime\_parameters*

- In the `vidi_runtime.h vidi_runtime_set_parameter` and `vidi_runtime.h vidi.runtime_tool_set_parameter` C++ functions, the `runtime_parameters/max_downsampling_size` parameter is no longer valid.
- The `ViDi2.Training.IRedHighDetailParameters.MaxDownSamplingSize` and `ViDi2.Runtime.IRedHighDetailParameters.MaxDownSamplingSize` C# function properties are removed.

## Deprecated *minimum\_epochs* Parameter in *training\_parameters*

- In the `vidi_training.h vidi_training_set_parameter` and `vidi_training.h vidi.training_tool_set_parameter` C++ functions, the `training_parameters/minimum_epochs` parameter is no longer valid.
- The `ViDi2.Training.IToolParametersHighDetail.MinimumEpochs` C# function property is no longer valid and will be removed from the future release.

## Changed the Internal Model Architecture for *small* Capacity Model

- In the `vidi_training.h vidi_training_tool_train2` C++ function, the `small` option is changed for the `training_artifact_key` parameter.
- In the `ViDi2.Training.IToolParametersHighDetail.NetworkModel` C# function, the `small` option is changed.

## Changed the Mechanism and Valid Value Range of the *count\_epochs* and *patience* *training\_parameters*

- In the `vidi_training.h vidi_training_set_parameter` and the `vidi_training.h vidi.training_tool_set_parameter` C++ functions:
  - The valid value range of the `training_parameters/count_epochs` parameter is changed to [50, 100000].
  - The valid value range of the `training_parameters/patience` parameter is changed to [0, 100].

- The `ViDi2.Training.IToolParametersHighDetail.CountEpochs` and the `ViDi2.Training.IToolParametersHighDetail.Patience` C# functions are changed.

For more information about the new mechanism, see the *VisionPro Deep Learning Software Help*.

## Introduced Parameters for the *Auto Resize Mode*

- In the `vidi_training.h vidi_training_set_parameter` and the `vidi_training.h vidi_training_tool_set_parameter` C++ functions, the `training_parameters/adaptive_downsampling` parameter is introduced with the following details:
  - Type: float
  - Valid range: [0, 1]
  - Description: Scale to resize the views for training and processing. You can manually set the 0 and the 1 values. Value 1 is the default value, resizing is disabled. Value 0 enables resizing in auto mode. In auto mode, the software automatically finds and uses the proper size ratio.
- In the `vidi_runtime.h vidi_runtime_get_parameter` and the `vidi_runtime.h vidi_runtime_tool_get_parameter` C++ functions, the `runtime_parameters/adaptive_downsampling` parameter is introduced with the following details:
  - Type: float
  - Valid range: [0, 1]
  - Description: This is a read-only parameter that is automatically set to the same value as the `training_parameters/adaptive_downsampling` parameter when the runtime workspace is exported.

**Note:** You cannot call the `vidi_runtime_set_parameter/vidi_runtime_tool_set_parameter` function for this parameter.

- The following C# functions are introduced:
  - `ViDi2.Training.IRedHighDetailParameters.AdaptiveDownsampling`
  - `ViDi2.Runtime.IRedHighDetailParameters.TrainedAdaptiveDownsampling`

For example, see *Examples/c#/Example.Training.HDRed.Console*, lines 156-158

## Removed the Default Parameter for the *width, height* Parameters of the `vidi_runtime_tool_convert_trt` function

- The `vidi_runtime.h vidi_runtime_tool_convert_trt` function is changed compared to the previous release as follows:
  - VisionPro Deep Learning 3.2.0:
    - `VIDI_INT width = 0`
    - `VIDI_INT height=0`
  - VisionPro Deep Learning 3.3.0:
    - `VIDI_INT width,`
    - `VIDI_INT height,`

- The `Vidi2.Runtime.ITool.OptimizeTensorRT C#` function is changed as follows:
  - VisionPro Deep Learning 3.2.0:
    - `int width = 0, int height = 0`
  - VisionPro Deep Learning 3.3.0:
    - `int width, int height,`

These parameters are the target size of the optimized TensorRT model. If the image sizes for processing are the same, use the exact same image width and height. However, if the image sizes are different, set each width and height based on the largest width and height among the images to be processed.

## VisionPro Deep Learning 3.2

### GPU Clock Stabilizer

First, you can try adjusting the GPU clock settings to prevent lowering the GPU clock. Please change the 'Power management mode' in Nvidia Control Panel > Manage 3D settings to 'Prefer maximum performance. However, if the GPU clock still cannot be maintained, consider using together with the new feature called GPU Clock Stabilizer.

The feature to maintain a stable GPU clock in a deployment environment can be helpful in multi-GPU scenarios or in situations where clock instability leads to reduced processing speed or spikes.

For more details, see the below example codes.

#### Note:



- This feature is initially disabled by default. You can control its activation or deactivation through the API.
- It should be employed after Compute Device initialization. Attempting to use it before initialization will result in the C API returning a value other than `VIDI_SUCCESS`, while the .NET API will trigger an exception.

#### C++ Example of GPU Clock Stabilizer

```

1  ...
2      /** @brief disable stabilize feature */
3      #define VIDI_STABILIZE_OFF 0
4
5      /** @brief enable stabilize feature on GPUs */
6      #define VIDI_STABILIZE_GPU 1
7
8      /** @brief stabilize the compute devices
9          * @param mode mode of operation
10         * @return 0 if passed, otherwise an error_code that can be
11         used with vidi_get_error_message()
12         *
13         * @see VIDI_STABILIZE_GPU, VIDI_STABILIZE_OFF
14         * This method must be called after the compute devices have
15         been initialized.
16         */

```

```

15 |          VIDI_DLLEXPORT  VIDI_UINT  vidi_stabilize_compute_device(VIDI_
    | INT mode);
16 |
17 |          ///////////////////////////////////////////////////////////////////
18 |          // example
19 |          ///////////////////////////////////////////////////////////////////
20 |
21 |          // Turns on.
22 |          vidi_stabilize_compute_device(VIDI_STABILIZE_GPU);
23 |
24 |          // Turns off
25 |          vidi_stabilize_compute_device(VIDI_STABILIZE_OFF);
26 |          ...

```

### C# Example of GPU Clock Stabilizer

```

1 | ...
2 |         /// <summary>
3 |         /// Offers the possibility to stabilize the compute
    | devices when the control is created, without unloading the vidi dll
4 |         /// example: mode = "StabilizeMode.Off"
5 |         ///         mode = "StabilizeMode.GPU"
6 |         /// This can only be called after compute devices
    | initialized.
7 |         /// </summary>
8 |         void StabilizeComputeDevices(Enum mode);
9 |
10 |
11 |         ///////////////////////////////////////////////////////////////////
12 |         // example
13 |         ///////////////////////////////////////////////////////////////////
14 |
15 |         // Turns on
16 |         control.StabilizeComputeDevices(StabilizeMode.GPU);
17 |
18 |         // Turns off
19 |         control.StabilizeComputeDevices(StabilizeMode.Off);
20 |         ...

```

## vidi\_initialize2() is Deprecated

vidi\_initialize2() function has been deprecated.

Starting from version VisionPro Deep Learning 3.2.0, you can load and use both the training API and runtime API simultaneously. Although the vidi\_initialize2() API still exists, it does not perform the intended operation.

Deprecated `vidi_initialize2()`

```
1 | VIDI_DLLEXPORT VIDI_UINT vidi_initialize2(VIDI_INT compute_mode, VIDI_STRING
   | compute_devices, VIDI_INT cuda_load_mode);
```

## VisionPro Deep Learning 3.0

Among the new features introduced in VisionPro Deep Learning 3.0, some of them are only provided through API as they enhance runtime functionality.

### Classification Batch Processing for Runtime API

Processing multiple views at once through batch processing is now available for Green Classify Standard. Using the following Runtime APIs can speed up the Green Classify Standard processing via API.

The steps to use classification batch processing API are summarized as follows.

1. Fix the batch size and prepare the images to process. The maximum batch size is dependent on the current available GPU memory.
2. Create a sample.
3. Add the prepared images to the sample.
4. Run batch processing.
5. Check the processing results.

For more details of utilizing batch processing API, see the below example codes.

**Note:**

- If you select a batch size larger than the maximum batch size which is calculated based on the available memory of the current GPU in Step 1, the batch processing will fail to start.
- Set the batch size and perform dummy processing twice initially. The processing time might be slow for the first two runs due to the warm-up.

C++ Example of Batch Processing API for Runtime (Green High Detail)

```
1 | ...
2 | // The maximum batch size depends on the available GPU memory size.
3 | // let's suppose the available maximum batch size is 12.
4 | constexpr int batch_size = 12;
5 |
6 | // you need to set "runtime_parameters/batch_size" as the batch size you
   | chose above.
7 | // "workspace": the name of your workspace
8 | // "default": the name of the stream in your workspace
9 | // "Classify": the name of the Green High Detail tool in your stream.
10 | string batch_size_str = std::to_string(batch_size);
11 | vidi_runtime_tool_set_parameter("workspace", "default", "Classify", "runtime_
   | parameters/batch_size", batch_size_str.c_str());
```

```

12  ...
13  // you also need to prepare the same number of the images as the batch size
14  vector<VIDI_IMAGE> images(batch_size);
15  for (int i = 0; i < batch_size; i++)
16  {
17      ...
18      status = vidi_load_image(image_path.c_str(), &images[i]);
19      ...
20  }
21  ...
22  // create a sample for batch processing. Note that a sample is the basic unit
    of processing task.
23  // "my_sample": the name of the sample you create.
24  status = vidi_runtime_create_batched_sample("workspace", "default", "my_
    sample");
25  ...
26  // add the prepared images to the sample
27  for (int i = 0; i < batch_size; i++)
28  {
29      ...
30      status = vidi_runtime_batched_sample_add_image
    ("workspace", "default", "my_sample", &images[i]);
31      ...
32  }
33  ...
34  // execute batch processing
35  status = vidi_runtime_batched_sample_process
    ("workspace", "default", "Classify", "my_sample", "");
36  ...
37  // get result of batch processing
38  status = vidi_runtime_get_batched_sample("workspace", "default", "my_
    sample", &result_buffer);
39  ...

```

#### C# Example of Batch Processing API for Runtime (Green High Detail)

```

...
// The maximum batch size depends on the available GPU memory size.
// let's suppose the available maximum batch size is 12.
int batch_size = 12;
...
// you also need to prepare the same number of the images as the batch size
// the paths list should contain the paths of images
List<IImage> imgs = new List<IImage>();
for (int i = 0; i < batch_size; ++i)

```

```

    imgs.Add(new LibraryImage(paths[i]));

// create a sample for batch processing. Note that a sample is the basic unit of
processing task.
// "Classify": the name of the Green High Detail tool in your stream.
using (IBatchedSample sample = stream.CreateBatchedSample())
{
    // add images to sample
    for (int i = 0; i < batch_size; ++i)
        sample.AddImage(imgs[i]);

    ITool greenTool = stream.Tools["Classify"];

    // set batch size parameter
    var param = greenTool.ParametersBase as
ViDi2.Runtime.IGreenHighDetailParameters;
    param.BatchSize = batch_size;

    // run processing
    sample.Process(greenTool);

    // get processing result of batch
    for (int i = 0; i < batch_size; ++i)
    {
        var tags = sample.Tags(greenTool, i);
        foreach (var tag in tags)
            Console.WriteLine($"tag: name={tag.Key}, score={tag.Value}");
    }
}

```

**Note:** Batch processing should not be used in a tool chain. If you want to use batch processing in a chain of tools, create a runtime workspace for each tool and chain them.

## NVIDIA TensorRT Support for Runtime API

[NVIDIA TensorRT](#) is an NVIDIA SDK that boosts the inference speed of a deep learning application. It optimizes the inference speed of a deep neural network by NVIDIA GPU model. From VisionPro Deep Learning 3.0, Runtime API supports TensorRT to boost the processing speed at runtime. To use TensorRT for Runtime API, go through the following steps. The TensorRT for Runtime API is supported only for Red Analyze Standard and Green Classify Standard runtime environment.

1. In the training environment, create a workspace, stream, and a Standard tool. Train the tool to make it ready to be deployed in the runtime environment. The training environment can be either VisionPro Deep Learning GUI or API. You can create a workspace, stream, and a High Detail tool and train the tool in GUI.
2. Export a runtime workspace which includes the trained Standard tool. You also can create a runtime workspace in GUI (Workspace > Export Runtime Workspace).
3. Deploy the runtime workspace and its Standard tool on the device on your front line.

4. Run TensorRT Optimization API to optimize the Standard tools in your runtime workspace with TensorRT, then save your runtime workspace.

**Note:** If the image sizes are the same, use the exact same image width and height. However, if the image sizes are different, set each width and height based on the largest width and height among the images to be processed.

- The optimization normally takes under 10 minutes to be completed. After the first run of optimization, the neural network of the tool is optimized to the specific NVIDIA GPU model you use. From this point, you can process as many images you want with faster speed.

#### C++ Example of TensorRT Optimization API for Runtime (Red High Detail and Green High Detail)

```

1 | ...
2 | // open the given workspace.
3 | // "workspace": the name of your runtime workspace
4 | // "..\\..\\..\\resources\\runtime\\Green High-detail Tool.vrws": the path to the
   | runtime workspace file
5 | status = vidi_runtime_open_workspace_from_file("workspace",
   | "..\\..\\..\\resources\\runtime\\Green High-detail Tool.vrws");
6 | ...
7 |
8 | // optimize the High-detail tool
9 | // "default": the name of the stream in your workspace
10 | // "Classify": the name of the Green High Detail tool in your stream.
11 | clog << "Start optimization. It will take a few minutes." << endl;
12 | status = vidi_runtime_tool_convert_trt
   | ("workspace", "default", "Classify", 0);
13 | ...
14 |
15 | // save runtime workspace with the optimized tool
16 | string save_path = "..\\..\\..\\resources\\runtime\\Green High-detail Tool
   | optimized.vrws";
17 | status = vidi_runtime_save_workspace("workspace", save_path.c_str());
18 | clog << "The workspace with the optimized tool is saved at " << save_path.c_
   | str() << endl;
19 | ...

```

#### C# Example of TensorRT Optimization API for Runtime (Red High Detail and Green High Detail)

```

1 | ...
2 | // Open a runtime workspace from file
3 | // the path to this file relative to the example root folder
4 | // and assumes the resource archive was extracted there.
5 | // "workspace": the name of your runtime workspace
6 | ViDi2.Runtime.IWorkspace workspace = control.Workspaces.Add("workspace",
   | "..\\..\\..\\resources\\runtime\\Green High-detail Tool.vrws");
7 |
8 | // Store a reference to the stream 'default'
9 | IStream stream = workspace.Streams["default"];
10 |
11 | // "Classify": the name of the Green High Detail tool in your stream.

```

```

12 var greenTool = stream.Tools["Classify"] as ViDi2.Runtime.IGreenTool;
13
14 // Optimizes the High-Detail tool with TensorRT. It takes some time to
    // optimize the tool.
15 greenTool.OptimizeTensorRT(0);
16
17 // Save runtime workspace with the optimized tool.
18 // You can use this workspace to process with the optimized tool.
19 string savePath = "..\\..\\..\\..\\resource\\runtime\\Green High-detail Tool
    Optimized.vrws";
20 workspace.Save(savePath);
21 Console.WriteLine($"The workspace with the optimized tool is saved at
    {savePath}.");
22 ...

```

If you want to use the [Classification Batch Processing for Runtime API on page 44](#) together with TensorRT-optimized processing, you need to set batch size before calling the optimization.

#### C++ Example of Setting Batch Size Before Calling TensorRT Optimization (Green High Detail)

```

1 // Optimize with batchSize=16
2 // The 5th(width) and 6th(height) parameter of vidi_runtime_tool_convert_trt()
    // are for target width and height for optimization
3 // If width=0 or height=0 is provided, the trained size is used. So we just
    // recommend use width=0, height=0 in normal case.
4 int gpu_index = 0;
5 int batch_size = 16;
6 vidi_runtime_tool_convert_trt("workspace", "default", "Analyze", gpu_
    index, 0, 0, batch_size);

```

#### C# Example of Setting Batch Size Before Calling TensorRT Optimization (Green High Detail)

```

1 // Optimize with batchSize=16
2 // The second(width) and third(height) parameter of OptimizeTensorRT() are for
    // target width and height for optimization
3 // If width=0 or height=0 is provided, the trained size is used. So we just
    // recommend use width=0, height=0 in normal case.
4 int gpuIndex = 0;
5 int batchSize = 16;
6 greenTool.OptimizeTensorRT(gpuIndex, 0, 0, batchSize);

```

Directory to C++ Full Example Codes:

C:/ProgramData/Cognex/VisionPro Deep Learning/3.0/Examples/c++/Example.Runtime.OptimizeHDTool

C:\ProgramData\Cognex\VisionPro Deep Learning\3.0\Examples\c++\Example.Runtime.HDGreen.Batched

Directory to C# Full Example Codes:

*C:/ProgramData/Cognex/VisionPro Deep Learning/3.0/Examples/c#/Example.Runtime.OptimizeHDTool.Console*

*C:\ProgramData\Cognex\VisionPro Deep Learning\3.0\Examples\c#\Example.Runtime.HDGreen.Batched.Console*

Load your runtime workspace in your application and deploy it on the device to process images with the TensorRT-optimized tool. To do this, you need to set a parameter before processing.

#### C++ Example of Setting Parameter to Use TensorRT Optimized-model for Processing (Red High Detail and Green High Detail)

```

1 // If you want to process with optimized tool, you have to set ProcessTensorRT
  to true.
2 // Setting this option to true will trigger TensorRT processing.
3 status = vidi_runtime_tool_set_parameter
  ("workspace", "default", "Analyze", "runtime_parameters/process_with_
  trt", "true");
4 ...
5 status = vidi_runtime_sample_process("workspace", "default", "Analyze", "my_
  sample", "");

```

#### C# Example of Setting Parameter to Use TensorRT Optimized-model for Processing (Red High Detail and Green High Detail)

```

1 // If you want to process with optimized tool, you have to set
  ProcessTensorRT to true.
2 // Setting this option to true will trigger TensorRT processing.
3 var hdParam = hdTool.ParametersBase as
  ViDi2.Runtime.IToolParametersHighDetail;
4 hdParam.ProcessTensorRT = true;
5 ...
6 sample.Process(hdTool);

```

See the example codes below to process images with a runtime workspace.

Directory to C# Example Codes:

- *C:/ProgramData/Cognex/VisionPro Deep Learning/3.0/Examples/c#/Example.Runtime.HDGreen.Console*
- *C:/ProgramData/Cognex/VisionPro Deep Learning/3.0/Examples/c#/Example.Runtime.HDRed.Console*

Directory to C++ Example Codes:

- *C:/ProgramData/Cognex/VisionPro Deep Learning/3.0/Examples/c++/Example.Runtime.HDGreen*
- *C:/ProgramData/Cognex/VisionPro Deep Learning/3.0/Examples/c++/Example.Runtime.HDRed*

**Note:** TensorRT Optimization API (`vidi_runtime_tool_convert_trt`, `.OptimizeTensorRT`) should be called again after your NVIDIA GPU model is changed because its optimization works by NVIDIA GPU model. If you changed your NVIDIA GPU model in your device, repeat the Step 4 and 5.

## VisionPro Deep Learning 2.1.1

In VisionPro Deep Learning 2.1.1, a new processing parameter `max_scan_iterations` is added only in API for Blue Locate and Blue Read. This parameter limits the maximum number of iterations for image scanning during processing and providing a fixed number for this parameter speeds up the processing.

**Note:** The features or the character features can be less detected than expected when the value of `max_scan_iterations` is set too low. The appropriate value for this parameter is dependent on the images.

You can get the value of this parameter by:

- `vidi_runtime_get_parameter`
- `vidi_runtime_tool_get_parameter`

You can change the value of this parameter by:

- `vidi_runtime_set_parameter`
- `vidi_runtime_tool_set_parameter`

The parameter path for `max_scan_iterations` is:

- `processing/blue/max_scan_iterations`

## Examples

This topic introduces the code examples of `max_scan_iterations` parameter for C and .NET API

C API

```
1 | VIDI_UINT status = vidi_runtime_tool_set_parameter
   | ("WORKSPACE", "STREAM", "TOOL", "processing/blue/max_scan_iterations", "40");
```

.NET API

```
1 | libraryAccess.SetToolParameter("WORKSPACE", "STREAM", "TOOL",
   | "processing/blue/max_scan_iterations", "40"); // libraryAccess is
   | ILibraryAccess
```

## VisionPro Deep Learning 2.1

As High Detail Quick modes are added in VisionPro Deep Learning 4.1, there are some changes to API as well.

### Green Classify High Detail Quick Training - C API

Green High Detail Quick Training with C API

```
1 | // Green high-detail-quick mode
2 | ...
```

```

3 param_ss << "workspaces/" << workspace_name << "/streams/" << stream_name
  << "/tools/" << tool_name << "/tool_type";
4 status = vidi_training_set_parameter(param_ss.str().c_str(), "green_quick");
5 ...
6

```

To train Green Classify High Detail Quick, the parameter path should be set to "green\_quick" for `vidi_training_set_parameter`.

The example of parameter path:

- `workspaces/WORKSPACE_NAME/streams/STREAM_NAME/tools/TOOL_NAME/tool_type`

To configure the Tool Parameters for Green Classify High Detail Quick, the parameter path also should be set. For example, if you want to change Epoch Count in Training Parameters, set the parameter path as:

- `workspaces/WORKSPACE_NAME/streams/STREAM_NAME/tools/TOOL_NAME/training_parameters/count_epochs`

See `C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c++\Example.Training.HDGreen\example_training_hdgreen.cpp` for the more detailed examples of training C API codes.

## Green Classify High Detail Quick Training - .NET API

Green High Detail Quick Training with .NET API

```

1 ...
2 // Green high-detail-quick mode
3 hdGreenTool.Database.Tool.Type = ToolType.GreenQuickHighDetail;
4 ...
5

```

To train Green Classify High Detail Quick, the `ViDi2.Training.ITool.Type` should be set to `ToolType.GreenQuickHighDetail`.

To configure the Tool Parameters for Green Classify High Detail Quick, the `ViDi2.Training.IGreenHighDetailParameters` should be set. For example, you should set the value of `CountEpochs` for `IGreenHighDetailParameters`.

See `C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c#\Example.Training.HDGreen.Console\Program.cs` for the more detailed examples of training .NET API codes.

## Green Classify High Detail Quick Processing - C API

The way to process Green Classify High Detail Quick with API is the same as those of Green Classify Focused and Green Classify High Detail. See `C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c++\Example.Runtime.HDGreen\example_runtime_hdgreen.cpp` for the details.

## Green Classify High Detail Quick Processing - .NET API

The way to process Green Classify High Detail Quick with API is the same as those of Green Classify Focused and Green Classify High Detail. See `C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c#\Example.Runtime.HDGreen.Console\Program.cs` for the details.

## Red Analyze High Detail Quick Training - C API

Red High Detail Quick Training with C API

```

1 | ...
2 | // Red high-detail quick mode
3 | param_ss << "workspaces/" << workspace_name << "/streams/" << stream_name
   | << "/tools/" << tool_name << "/tool_type";
4 | status = vidi_training_set_parameter(param_ss.str().c_str(), "red_quick");
5 | ...
6 |
```

To train Red Analyze High Detail Quick, the parameter path should be set to "red\_quick" for `vidi_training_set_parameter`.

The example of parameter path:

- `workspaces/WORKSPACE_NAME/streams/STREAM_NAME/tools/TOOL_NAME/tool_type`

To configure the Tool Parameters for Red Analyze High Detail Quick, the parameter path also should be set. For example, if you want to change Epoch Count in Training Parameters, set the parameter path as:

- `workspaces/WORKSPACE_NAME/streams/STREAM_NAME/tools/TOOL_NAME/training_parameters/count_epochs`

See `C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c++\Example.Training.HDRed\example_training_hdred.cpp` for the more detailed examples of training C API codes.

## Red Analyze High Detail Quick Training - .NET API

Red High Detail Quick Training with .NET API

```

1 | ...
2 | // Red high-detail quick mode
3 | hdRedTool.Database.Tool.Type = ToolType.RedQuickHighDetail;
4 | ...
```

To train Red Analyze High Detail Quick, the `ViDi2.Training.ITool.Type` should be set to `ToolType.RedQuickHighDetail`.

To configure the Tool Parameters for Red Analyze High Detail Quick, the `ViDi2.Training.IRedHighDetailParameters` should be set. For example, you should set the value of `CountEpochs` for `IRedHighDetailParameters`.

See *C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c#\Example.Training.HDRed.Console\Program.cs* for the more detailed examples of training .NET API codes.

## Red Analyze High Detail Quick Processing - C API

The way to process Red Analyze High Detail Quick with API is the same as those of Red Analyze Focused Supervised and Red Analyze High Detail. See *C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c++\Example.Runtime.Red\example\_runtime\_red.cpp* for the details.

## Red Analyze High Detail Quick Processing - .NET API

The way to process Red Analyze High Detail Quick with API is the same as those of Red Analyze Focused Supervised and Red Analyze High Detail. See *C:\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\c#\Example.Runtime.HDRed.Console\Program.cs* for the details.

## API Deprecated 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

Library	Obsolete Name	New Name
.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
.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.Occurrences	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)	

Library	Obsolete Name	New Name
.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()
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()
C	vidi_initialize2()	

Library	Obsolete Name	New Name
C++	vidi_runtime_green_set_similarity_reference_views	
C++	vidi_runtime_green_remove_similarity_reference_views	
C++	vidi_runtime_green_list_similarity_reference_views	
C++	vidi_runtime_green_train_similarity_model	
C++	vidi_training_green_set_similarity_reference_views	
C++	vidi_training_green_list_similarity_reference_views	
C++	vidi_training_green_train_similarity_model	
C++	vidi_training_tool_set_incremental_learning	
C++	vidi_training_tool_unset_incremental_learning	
C++	vidi_training_tool_adapt	
C++	vidi_training_tool_set_pretrained_model	
C++	vidi_training_tool_get_pretrained_model_info	
C++	runtime_parameters/heat_map_on	
C++	training_parameters/incremental	
C++	runtime_parameters/process_with_trt (boolean)	runtime_parameters/trt_mode (string)
C++	training_parameters/network_type	
C++	training/red/exclusive	
C++	gpu_hdm	gpu_standard
C++	gpu_Hdm	gpu_Standard
C++	visual_debugger_on	
C++	visual_debugger_on	heat_map_on
C#	ViDi2.IGreenTool.SimilarityReferences	
C#	ViDi2.IGreenTool.TrainSimilarityModel	
C#	ViDi2.Training.SetIncrementalTrain	
C#	ViDi2.Training.UnsetIncrementalTrain	
C#	ViDi2.Training.ITool.Adapt	
C#	ViDi2.Training.ITool.SetPretrainedModel	
C#	ViDi2.Training.ITool.GetPretrainedModelInfo	
C#	ViDi2.Runtime.IGreenHighDetailParameters.HeatMapOn	
C#	ViDi2.Training.IToolParameters.Incremental	

## API Known Issues

The Home page in the .NET/C Documentation files is outdated. The names of the example projects in the `C:\ProgramData\Cognex\VisionPro Deep Learning\4.0\Examples\` directory changed.

In the VisionPro Deep Learning .NET Documentation `.chm` file:

- From `Example.Runtime.HDGreen.Console` to `Example.Runtime.GreenStandard.Console`
- From `Example.Runtime.HDRed.Console` to `Example.Runtime.RedStandard.Console`

- From `Example.Training.HDGreen.Console` to `Example.Training.GreenStandard.Console`
- From `Example.Training.HDRed.Console` to `Example.Training.RedStandard.Console`

The following example projects are missing from the documentation:

- `Example.Runtime.Console.MultiGPU2`
- `Example.Runtime.GreenStandard.Batched.Console`
- `Example.Runtime.OptimizeStandardTool.Console`

In the VisionPro Deep Learning .C Documentation *.chm* file:

- From `Example.Runtime.HDGreen` to `Example.Runtime.GreenStandard`
- From `Example.Runtime.HDRed` to `Example.Runtime.RedStandard`
- From `Example.Training.HDGreen` to `Example.Training.GreenStandard`
- From `Example.Training.HDRed` to `Example.Training.RedStandard`

The following example projects are missing from the documentation:

- `Example.Runtime.MultiGPU2`
- `Example.Runtime.GreenStandard.Batched`
- `Example.Runtime.OptimizeStandardTool`

# System Requirements

The following subsections describe the software and hardware requirements to use VisionPro Deep Learning.

## PC Requirements

Both PCs for training and PCs for deployment purposes require a fast multi-core CPU, large system memory (RAM), and a strong NVIDIA® GPU. Training PCs require larger GPU and system memory, and a large amount of free SSD space.

Requirement	Description
CPU	<p>Intel Core i7 or higher.</p> <ul style="list-style-type: none"> <li>The Blue Locate tool is sensitive to clock speed rates, particularly with complex model matching applications.</li> <li>If a single core performs multiple tasks, processing time can increase. Therefore, use a sufficient amount of resources or allocate only one task to a single core.</li> </ul>
GPU	<p>Cognex recommends using a GPU for higher performance and for access to Standard tools. You can use Legacy mode tools without a GPU, but you cannot process Standard and Standard tools.</p> <p>VisionPro Deep Learning has the following GPU requirements:</p> <ul style="list-style-type: none"> <li>VisionPro Deep Learning4.1 supports all NVIDIA® CUDA®-enabled GPUs with compute capability of 6.1 and higher. For more information about GPU compute capability, see the <a href="#">NVIDIA</a> website.</li> <li>10 GB or larger GPU RAM is required for a training PC.</li> <li>8 GB or larger GPU RAM is required for a deployment PC.</li> </ul> <p><b>Note:</b> For training and processing Standard tools, Cognex recommends more than 10 GB.</p> <p>Cognex recommends using a GPU driver version from 528.02 to 532.03, with the following considerations:</p> <ul style="list-style-type: none"> <li>The tested driver version is 528.02.</li> <li>Driver versions from 532.03 to 546.01 are not recommended.</li> <li>You can use a driver version higher than 546.01 if necessary. When using a version higher than 546.01, you must disable Shared GPU memory usage in the NVIDIA Control Panel. For more information, see <a href="#">Disable Shared GPU Memory</a>.</li> </ul> <p><b>Note:</b> Using driver versions beyond 532.03 can result in slower training and processing times in Standard mode.</p> <p>Visit the <a href="#">NVIDIA Driver Downloads</a> website to find the right driver version for your GPU.</p> <p>For more information about selecting GPUs for your application, see <a href="#">Guidelines for GPU Selection on page 60</a>.</p>
System memory (RAM)	<p>VisionPro Deep Learning has the following minimum system memory requirements:</p> <ul style="list-style-type: none"> <li>64 GB RAM or larger is required for a training PC.</li> <li>32 GB RAM or larger is required for a deployment PC.</li> </ul>
System storage	<p>SSD with 200 GB or more free space is required for a training PC.</p>
PCIe lanes	<p>Minimum x8 PCIe lanes. A x16 PCIe lane can reduce cycle time by approximately 10 ms, relative to a PCIe x8 (based on a 5 MP image).</p>

Power supply	1.25 times larger than the system and GPU power requirements.
USB 2.0	USB port for a permanent Security Key connection via USB 2.0.

## Guidelines for GPU Selection

VisionPro Deep Learning4.1 supports all NVIDIA® CUDA®-enabled GPUs with compute capability of 6.1 and higher. For more information about GPU compute capability, see the [NVIDIA](#) website.

Cognex recommends selecting an NVIDIA® GPU with a higher core clock frequency, higher memory, and a higher number of CUDA cores and Tensor cores because these models provide faster computational speeds.

**i Note:** To use GPU driver versions higher than 546.01, you need to disable the System Fallback setting in the NVIDIA Control Center. See [Disable Shared GPU](#).

## Tested GPU Models

The following list includes the GPU models that Cognex has tested. Cognex aims to support most common GPU families, but we do not test every model. If the GPU model meets the compute capability requirements, you can use it. Cognex customers use most common GPUs other than very new models, even if the GPUs have not been tested by Cognex.

- NVIDIA® GeForce® RTX 3080
- NVIDIA® GeForce® RTX 3080 Ti
- NVIDIA® GeForce® RTX 3090
- NVIDIA® GeForce® RTX 4080
- NVIDIA® GeForce® RTX 4090
- NVIDIA® Quadro® P4000

## Not Recommended GPU Models

Cognex does not recommend the following GPU models because they provide low performance and slow training time for unknown reasons:

- NVIDIA® GeForce® GTX 1650
- NVIDIA® GeForce® GTX 1660

## Supported Operating Systems

Microsoft® Windows® Operating System	English	Chinese	Japanese	Korean	French	German	Spanish
Windows 11 Professional (64-bit)	✓	✓	✓	✓	✓	✓	✓
Windows 10 Professional (64-bit)							
<b>Note:</b> Windows 10, version 1809 has an issue when running the VisionPro Deep Learning application. Cognex recommends using Windows 10, version 22h2.	✓	✓	✓	✓	✓	✓	✓
Windows Server 2019	✓	✓	✓	✓	✓	✓	✓
Windows Server 2022	✓	✓	✓	✓	✓	✓	✓

**Note:** You cannot use VisionPro Deep Learning or the Deep Learning Service on a virtual machine.

**Note:** Due to issues from the NVIDIA driver, Windows Server 2016 is not supported for VisionPro Deep Learning 4.1 Server/Client mode. See [Known Issues on page 3](#) for more details and the workaround.

## API Development Requirements

You need the following software and components for developing VisionPro Deep Learning custom applications through the VisionPro Deep Learning API:

- Microsoft® Visual Studio® 2017, 2019, or 2022
- Microsoft .NET Framework 4.7.2
- For VisionPro and Designer users, Microsoft .NET Framework 4.8

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

## Supported Designer Release

VisionPro Deep Learning 4.1 supports Cognex Designer 4.5.2.

For Designer, VisionPro Deep Learning supports TensorRT Optimization, which makes processing faster. To use this feature for Deep Learning Runtime Block in Designer, see the sample projects packaged in the VisionPro Deep Learning plug-in for Designer.

**Note:** You can manage your runtime workspace created by Cognex Deep Learning Studio, but you cannot control the Cognex Deep Learning Studio GUI inside of Designer.

## Supported VisionPro Release

VisionPro Deep Learning 4.1 supports VisionPro 9.25.

## Image Format Requirements

The supported input image channels, bit depths, and image file formats in VisionPro Deep Learning and the image channels, bit depths, and image file formats converted after importing to VisionPro Deep Learning are described in the table below. The image channels, bit depths, and image file formats not described in this table are not supported in VisionPro Deep Learning.

**Note:** When using 16-bit images with Standard tools, the runtime result is incorrect.

Before Importing into VisionPro Deep Learning			After Importing into VisionPro Deep Learning		
Input Image Channel	Input Image Bit Depth	Input Image File Format	Imported Image Channel	Imported Image Bit Depth	Imported Image File Format
1 ch	8-bit	bmp	1 ch	8-bit	png
1 ch	8-bit	jpg	1 ch	8-bit	png
1 ch	8-bit	png	1 ch	8-bit	png
1 ch	8-bit	tiff	1 ch	8-bit	png
1 ch	16-bit	png	1 ch	16-bit	png
1 ch	16-bit	tiff	1 ch	16-bit	png

3 ch	8-bit	bmp	3 ch	8-bit	png
3 ch	8-bit	jpg	3 ch	8-bit	png
3 ch	8-bit	png	3 ch	8-bit	png
3 ch	8-bit	tiff	3 ch	8-bit	png
3 ch	16-bit	png	3 ch	16-bit	png
3 ch	16-bit	tiff	3 ch	16-bit	png
4 ch	8-bit	bmp	4 ch	8-bit	png
4 ch	8-bit	png	4 ch	8-bit	png
4 ch	8-bit	tiff	4 ch	8-bit	png
4 ch	16-bit	png	4 ch	16-bit	png
4 ch	16-bit	tiff	4 ch	16-bit	png

## Image Size and Dimensions

The figures in the table below, for example, the maximum number of images per stream, are experimentally tested and the values vary depending on your hardware configuration.

When working with images, observe the following considerations:

- As the number and the size of images get larger, the training and processing time for all images increases as well.
- For Standard type Green Classify tools, images are automatically resized for training and processing to maximum 16 MP.
- The more memory (not GPU RAM) and hard disk capacity you have, the more and larger images you can handle.
- Using several hundred features per image with the Blue Locate tool slows down the user interface considerably.

Tool	Tool Type	Minimum Size	Maximum Size (Channel Parameter Set to 1 to 4)	Maximum Images per Stream	Maximum Trained Features per Image	Maximum Discrete Labeled Features	Maximum Number of Classes	Maximum Number of Defects per Image
Blue Locate	N/A	N/A	<ul style="list-style-type: none"> <li>◦ Images up to 32 MP (up to 16k in height or width) are supported.</li> <li>◦ Images up to 32k pixels wide (in one dimension, dependent upon GPU resources).</li> <li>◦ Images over 32k are not supported.</li> </ul>	25000	1000	256	N/A	N/A
Blue Read				25000	N/A	N/A	200	N/A
Green Classify	Legacy	N/A		1000	N/A	N/A	N/A	1000
	Standard			96 x 96 (pixels)				

# Install VisionPro Deep Learning

To install VisionPro Deep Learning:

1. Attach the Security Key to a USB port on the computer that you want to use to develop the vision application.

**Note:** The Security Key must remain connected to the computer during all phases of operation. For more information, see [Software Security](#).

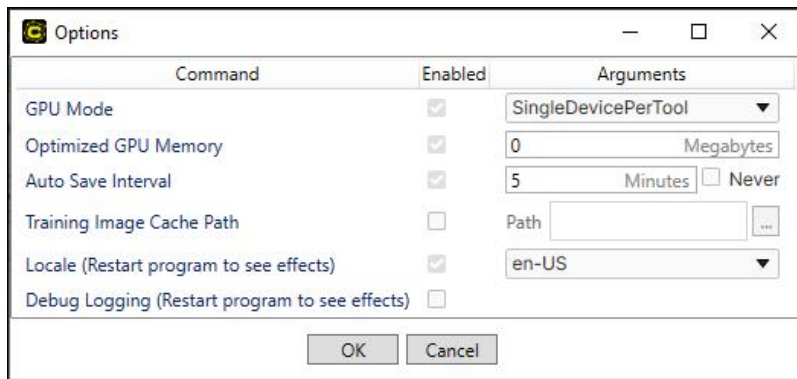
2. Download the latest VisionPro Deep Learning installer from the [Cognex support page](#).
3. Extract the contents of the .zip file.
4. Run the executable file in the extracted folder to start the VisionPro Deep Learning installer.
5. After accepting the license agreement, enter your user name and company name.
6. Select your installation type:
  - **Complete:** all VisionPro Deep Learning features are installed.
  - **Custom:** select the VisionPro Deep Learning features that you want to install.
7. Specify the install directory. The default directory is `C:\Program Files\Cognex\VisionPro Deep Learning\4.1`.
8. If you have selected a custom installation, you can uncheck the checkbox for any VisionPro Deep Learning components that you do not want to install.



9. Click **Install** at the final prompt and wait for the installation to complete.

# Launch VisionPro Deep Learning

In the launcher, you can choose to run VisionPro Deep Learning locally or by connecting to a server. By selecting **Options**, you can reach several options that control the GPU Mode, which GPU device to use, the allocation of GPU memory, in addition to project settings.



Command	Description
GPU Mode	<p>Specifies the GPU mode to be used by the application. The following options are available:</p> <p>SingleDevicePerTool    The application uses a single GPU for training and processing an image.</p> <p>MultipleDevicesPerTool    This option is not supported in the current release.</p> <p>NoSupport    The application does not use a GPU.</p> <p><b>Note:</b> This option conflicts with --gpu-devices and/or --optimized-gpu-memory.</p>
Optimized GPU Memory	<p>Specifies the size of the preallocated optimized memory buffer.</p> <p><b>Note:</b> You can also reach the <b>Optimized GPU Memory</b> option from the GUI by selecting the <b>Compute Devices</b> option of the <b>Help</b> menu. For more information, see <a href="#">GPU Memory Optimization</a>.</p>
Auto Save Interval	<p>Specifies how often a workspace is saved automatically. The default is every 5 minutes.</p>
Training Image Cache Path	<p>Specifies a cache location for training images. This is useful if the images in a stream are in non-raster format (for example JPG, PNG, or GIF), since they must be converted to a raster (BMP) format for training. By default, this conversion happens multiple times for each image. By enabling the <b>Training Image Cache Path</b> option, the application only converts the images once then stores the converted images in a local cache directory. Use a local cache directory, preferably on a Solid-State Drive (SSD). This option can also speed up training in cases where the workspace is stored on a slower drive or a remote storage device.</p>
Locale	<p>Specifies the language of the VisionPro Deep Learning GUI.</p>

Command	Description
Debug Logging	Specifies whether or not debug logs are enabled for the project.

# Legal Notices

The software described in this document is furnished under license, and may be used or copied only in accordance with the terms of such license and with the inclusion of the copyright notice shown on this page. Neither the software, this document, nor any copies thereof may be provided to, or otherwise made available to, anyone other than the licensee. Title to, and ownership of, this software remains with Cognex Corporation or its licensor. Cognex Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Cognex Corporation. Cognex Corporation makes no warranties, either express or implied, regarding the described software, its merchantability, non-infringement or its fitness for any particular purpose.

The information in this document is subject to change without notice and should not be construed as a commitment by Cognex Corporation. Cognex Corporation is not responsible for any errors that may be present in either this document or the associated software.

Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, nor transferred to any other media or language without the written permission of Cognex Corporation.

Copyright © 2025. Cognex Corporation. All Rights Reserved.

Portions of the hardware and software provided by Cognex may be covered by one or more U.S. and foreign patents, as well as pending U.S. and foreign patents listed on the Cognex web site at: [cognex.com/patents](http://cognex.com/patents).

---

The following are registered trademarks of Cognex Corporation:

Cognex, 2DMAX, Advantage, AlignPlus, Assemblyplus, Check it with Checker, Checker, Cognex Vision for Industry, Cognex VSOC, CVL, DataMan, DisplayInspect, DVT, EasyBuilder, Hotbars, IDMax, In-Sight, Laser Killer, MVS-8000, OmniView, PatFind, PatFlex, PatInspect, PatMax, PatQuick, SensorView, SmartView, SmartAdvisor, SmartLearn, UltraLight, Vision Solutions, VisionPro, VisionView

The following are trademarks of Cognex Corporation:

The Cognex logo, 1DMax, 3D-Locate, 3DMax, BGAll, CheckPoint, Cognex VSoC, CVC-1000, FFD, iLearn, In-Sight (design insignia with cross-hairs), In-Sight 2000, InspectEdge, Inspection Designer, MVS, NotchMax, OCRMax, PatMax RedLine, ProofRead, SmartSync, ProfilePlus, SmartDisplay, SmartSystem, SMD4, VisiFlex, Xpand

Portions copyright © Microsoft Corporation. All rights reserved.

Portions copyright © MadCap Software, Inc. All rights reserved.

Other product and company trademarks identified herein are the trademarks of their respective owners.