Release Notes

As of VisionPro Deep Learning 3.0, Cognex brings a couple of brand new features that steer your machine vision journey to the new round of excellence. For the updates made in API, see API.

Adaptation Mode for High Detail tools

Sometimes, the newly captured images in the front lines of your daily operation can deviate from the training images that constructed your VisionPro Deep Learning tool which handles the incoming images, due to many reasons. For example, the height of your cameras in your factory line can change as the camera mounting plates or rods are worn out, and this can lead to unexpected variations in incoming images compared to the initial images. In this case, the result of a VisionPro Deep Learning tool also starts to deviate against the expected result because the tool was trained on initial images, not on the new images that contain variations. Adaptation Mode was invented to handle this kind of problem by filling the performance gap between the result from training images and the result from newly captured images by adapting a previously trained tool to the new images with variations. Adaptation Mode is supported only for Red Analyze High Detail and Green Classify High Detail tool. See Adaptation Mode: Adapting to Line Variation for details.

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 in VisionPro Deep Learning 3.0.

- Red Analyze High Detail is evolved to provide higher accuracy. In many cases, the new architecture of Red Analyze High Detail shows higher accuracy than the old architecture.

- Green Classify High Detail is evolved to provide higher accuracy. Especially, the focus of Green Classify High Detail update is on the Large network model among Green Classify High Detail Network Model parameters (Small, Normal, Large), and it aims to provide the followings.

  - Higher Accuracy
  - Removal of Extra Large in Network Model parameters

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

Due to this upgrade, you need to consider the followings to use a Red Analyze High Detail or a Green Classify High Detail tool trained in the earlier VisionPro Deep Learning versions. For more details, see *Migrating to This Release* on page 20.

- Opening a workspace and a High Detail 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.

- Importing a Red Analyze High Detail tool (.vrta) or a Green Classify High Detail 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.

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. See Validation Set and Validation Loss for more details.

Automatic Parameter Search

A new mode for Parameter Search is introduced for a fast automatic search. Given a search range, the automatic mode (auto mode) automatically prunes the values of parameters to get you the best possible results within the shortest time. The existing parameter search is now called the manual mode. 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 Parameter Search is only supported for High Detail tools. See Parameter Search for details.

Label Check

If you deal with a large amount of image data, it is difficult to label all the data and control the labeling tasks. In this case, you are vulnerable to wrong or noisy labels and it takes so much cost to find wrong or noisy labels. For such a case, Cognex offers the Label Check feature to automatically review the goodness of the labels in your images. By simply clicking a few buttons, the label quality information is displayed for each view. 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 Focused Supervised, Red Analyze High Detail, and all Green Classify tool types. See Optimize Labeling with Label Check for more details.

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. See Cognex Label Import/Export Plug-in for details.

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. For details of TensorRT support, see *API Changelog* on page 34.

Green Classify High Detail Batch Processing Support for Runtime

Speeding up processing of a VisionPro Deep Learning tool can be achieved processing multiple images at once. To implement this idea, a way that divides the entire images into multiple batches and processes them in batches is adopted. This is called "batch processing" newly introduced in VisionPro Deep Learning 3.0 for Green Classify High Detail. Using the Batch Processing API in runtime can speed up the Green Classify High Detail processing via API. For more details about the batch processing API, see *API Changelog* on page 34.

View Browser - List Mode

The list mode of the View Browser is a handy tool that provides you an overlook over every view's information in detail. The existing View Browser is now called the grid mode, and you can switch back and forth between these 2 modes. 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. See View Browser - List Mode for details.

Other Improvements & Changes

UI/UX Improvements in Parameter Search

For both the auto and the manual mode, the full list of parameters are provided and you can enable or disable checkboxes to choose what parameters to be tested on the Search page. For High Detail tools, picking up the best parameter set is determined with the new validation loss formulation adopted in VisionPro Deep Learning 3.0. One another change is that the scale of validation loss now ranges from 0 to 1, replacing "from 0 to 100" scale. See Validation Set and Validation Loss (Red Analyze High Detail) and Validation Set and Validation Loss (Green Classify High Detail).

GUI Refinement

As of VisionPro Deep Learning 3.0 release, the GUI of VisionPro Deep Learning is renovated adopting modern UI style and standards. The most remarkable change is that the View Browser is now divided into 2 modes: the grid mode and the list mode. See GUI and API for more details.

Applying Label Change for Green Classify

If you change the label of a single view or create one for a single unlabeled view, the current tool (a tool already trained and processed once) will not be automatically reprocessed anymore. Instead, after modifying or creating labels as necessary, you need to process the current tool manually.

Image Resize for Green Classify High Detail

The resize view (Resize Parameters on Green Classify High Detail Tool Parameters) UI/UX is improved to let you update the maximum buffer size (the maximum width and height of a view to be resized) whenever you want. You no longer need to use the inefficient hard-coded maximum buffer sizes in both the auto and the manual modes. Instead, you are guided to update it from the snapshot of the available GPU memory and the value of Network Model parameters, and this will help you make the most of GPU memory. For more details of the new image resizing, see Resize.

Validation Set Indicator and Filter for Validation Set

The views that are included in the validation set of a High Detail tool are now have special a indicator displayed on Image Display Area. It indicates these views belong to the validation set among the images of the training set. Also, new keywords 'validated'/ 'not validated' are introduced for Display Filter and Database Overview Filter to display/exclude only the views included in the validation set. For more details, see Prepare Validation Set (Green Classify High Detail) and Prepare Validation Set (Red Analyze High Detail).

Scale Change of Validation Loss

The scales Red Analyze High Detail and Green Classify High Detail validation loss are changed from "0 to 100" to "0 to 1." This change applies when you monitor the validation loss with loss graphs and use Parameter Search auto mode for both tools. See Loss Inspector and Auto Mode for more details.

Image Processing Logic Change for Red Analyze High Detail

The processing logic for Red Analyze High Detail is changed by implementing its new architecture upgrade, and Red Analyze High Detail now processes images with a different method. If you want to use the image-wise processing logic, you can use it by enabling Whole Image Processing checkbox on Processing parameters. For more details about this change, see Configure Processing Parameters.

Minimum Image Size Change for Green Classify High Detail

The minimum image size for Green Classify High Detail is relaxed from 96 x 96 to 64 x 64 (pixels). For more details, see Collect Images.

- Minimum Image Size for Training: 64 x 64
- Minimum Image Size for Processing: 64 x 64

Training Logic Change with Patience Epochs and Minimum Epochs

Patience Epochs and Minimum Epochs in the Training parameters of High Detail tools are applied differently during training from VisionPro Deep Learning 3.0. For more details, see Training Parameter Details: Patience Epochs & Minimum Epochs.

Designer Plug-in Support

In VisionPro Deep Learning 3.0, the Designer plug-in is restored with the support for the following new features introduced in VisionPro Deep Learning 3.0.

- TensorRT Optimization

- Whole Image Processing (Red Analyze High Detail)

For more details, see Using Runtime Workspace with Cognex Designer.

Miscellaneous.

- 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])`.

Note:

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

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

- Sample code is included with the VisionPro Deep Learning 3.0 installer. The resources for the sample code (*VisionPro Deep Learning 3.0 Sample Code Resource.msi*), which include images and workspaces, are available for download from the Cognex Support Site.

- For more information about new API functionality in this release, see the *Cognex VisionPro Deep Learning Suite .NET Library* and *Cognex VisionPro Deep Learning Suite C API* help files.

Resolved Issues

This section describes changes and resolved issues in VisionPro Deep Learning 3.0.

- Fixed an issue that the Red Analyze sort filter keyword `count(region!)` was applied to count the background for a Bad-labeled view without any labeled defect region. To fix this issue, the sort filter keyword is changed to `count (region![area > 0])` and applying this keyword will not count the background pixels as one labeled defect region for a Bad-labeled view that doesn't have any labeled defect region.

- Fixed an issue for Red Analyze High Detail that once it was trained, defect regions were detected in the masked pixels after processing.

- Fixed an issue for Blue Read that the model information displayed in VisionPro Deep Learning and in Parameter Search did not match each other.

- Fixed an issue that closing a workspace does not free the PC memory occupied by it.

- Fixed an issue that the Database Overview and marking information of a tool was not erased though you had reset the tool.

- Fixed an issue that the PC memory consumption continuously soared up when you executed `ViDi2.UI.MarkingOverlayExtensions.OverlayImage()` method.

Known Issues

The following issues have been identified in this release. For the API known issues, see

VisionPro Deep Learning 3.0

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.

- Due to several issues related to GUI and training, all users of VisionPro Deep Learning 3.0 MUST use NVIDIA driver that is of:

  - 466.77 or higher version for NVIDIA GeForce® series if the issues above continue to occur.

  - 466.77 or higher version for NVIDIA RTX / Quadro® series if the issues above continue to occur.

- With the minimum required GeForce® RTX driver version 466.77, an error can occur with some models of NVIDIA GPUs. In this case, please use the higher GeForce® RTX driver version. For example, if your GPU's GRAM size is equal to or less than 10GB, it is recommended to upgrade the driver version to 516.59.

Operating System (OS)

- Due to OS compatibility issues caused from NVIDIA drivers, Windows Server 2016 is not supported on VisionPro Deep Learning 3.0. For the users with Windows Server 2016 environment, upgrading from Windows Server 2016 to Windows Server 2019 is required for the utilization of VisionPro Deep Learning 3.0 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 3.0, but you cannot modify this.

- See the FAQ for the workaround for this issue.

- It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio 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.

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 Focused Supervised, 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 High Detail, resizing views takes a little more time when it is the first processing of the tool.

- For Green Classify High Detail, the processing preparation takes a little more time when it is the first processing of the tool.

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.

Parameter Search

- In Home page, if you set 1-depth directory path (example: "E:/," or "A:/") in the Path, Parameter Search is terminated with an error and you can't re-open it. Please refer to Unable to start Parameter Search after setting 1-depth directory path at Path in its Home page for the workaround of this issue.

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

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

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.

    ○ See the FAQ for the workaround for this issue.

    ○ It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio.

- 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.

Parameter Search

- In Home page, if you set 1-depth directory path (example: "E:/," or "A:/") in the Path, Parameter Search is terminated with an error and you can't re-open it. Please refer to Unable to start Parameter Search after setting 1-depth directory path at Path in its Home page for the workaround of this issue.

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 VPDL 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.
    - See the FAQ for the workaround for this issue.
    - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio.

- 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 VPDL 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.

  - See the FAQ for the workaround for this issue.

  - It is recommended to delete "Group" in the previous workspaces and import to Cognex Deep Learning Studio.

- 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 may affect existing VisionPro Deep Learning projects when migrating projects from previous releases to the current release.

2.1 (2.1.1) → 3.0

Removal of "Extra Large" from Green Classify High Detail Network Models

Due to the Green Classify High Detail architecture update in VisionPro Deep Learning 3.0, the Extra Large option in Network Model parameters for a Green Classify High Detail is removed from VisionPro Deep Learning. If you want to use a Green Classify High Detail 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 High Detail 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 High Detail 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.

High Detail Backwards Compatibility

When you open a Red Analyze High Detail or Green Classify High Detail 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 High Detail tool (.vrta) or a Green Classify High Detail 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 High Detail, Red Analyze High Detail Quick, Green Classify High Detail, or Green Classify High Detail Quick 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 High Detail 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. See Import a SuaKIT Classification Project and Import a SuaKIT Segmentation Project for more details.

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 3.0. 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. For more information, see *API Deprecated Features* on page 48 .

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

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

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

In 4.1.0 and later, it must now be expressed as the following:

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

> **Note**:
> This syntax can also more easily be expressed by using the `in` operator. For example, the following syntax could be used to simplify the expression:
>
> ```
> x in [0,view.width] and y in [0,view.height]
> ```

Changes to the Viewing of Overlay Graphics

In the 3.4.1 releases, the keyboard shortcuts used to show and hide overlay graphics in the Image Display Area have been changed. The new keystrokes are the following:

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

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

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

Changes to Tool Training Workflow

In the 3.4.0 release, there is a new method for specifying the images/views that will be used to train a tool. If you will be re-training a tool that was trained in a previous release, you will need to configure the tool to use the new training set paradigm, via the Select Training Set dialog. Please refer to the 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 version 2.1, these constraints were interpreted to apply to the position of the origin of the model coordinates. In version 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 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. For more details, see .

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. See for more details.

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. See Heatmap and GUI and API for the details.

- 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. See for the details.

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 .vvb 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). See Integration with Cognex VisionPro and .

- 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. See Red Analyze Focused Supervised - Confusion Matrix for an example of the F1 score of confusion matrix.

    ◦ 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 Focused Supervised

    ◦ 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. See Red Analyze Focused Supervised - Neural Network Processing (Inferencing) for an example of the Auto Threshold.

- 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. See Blue Locate - Review Markings for an example of the overlay checkbox for Blue Locate tool.

- 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. See Blue Locate - Neural Network Training for an example of the restore parameters button for Blue Locate tool.

- 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. See Configure Multiple GPUs for the details.

- 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 be consist of RTX™ 30 series of the same type and of the same specification. See Configure Multiple GPUs for more details.

  ○ Currently, there are some known issues relating to NVIDIA driver versions and Processing Time. See *Known Issues* on page 7 for more details.

- 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/absense 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.
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.

33

API Changelog

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

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 High Detail. Using the following Runtime APIs can speed up the Green Classify High Detail 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.

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
```

```csharp
// 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 exploit TensorRT for Runtime API, go through the following steps. The TensorRT for Runtime API is supported only for Red Analyze High Detail and Green Classify High Detail runtime environment.

1. In the training environment, create a workspace, stream, and a High Detail 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. For the more details of the training and runtime environment, see Environments.

2. Export a runtime workspace which includes the trained High Detail tool. You also can create a runtime workspace in GUI (Workspace > Export Runtime Workspace). For the more details of the runtime workspace, see Runtime Workspace.

3. Deploy the runtime workspace and its High Detail tool on the device on your front line. For the more details of the runtime deployment, see Runtime Workspace and Runtime Deployment.

4. Run TensorRT Optimization API to optimize the High Detail tools in your runtime workspace with TensorRT, and save your runtime workspace. The optimization normally takes under 10 minutes to be completed. After the first run of optimization, the tool's neural network is optimized to the specific NVIDIA GPU model you use, and 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.Write($"The workspace with the optimized tool is saved at
     {savePath}.");
22   ...
```

If you want to use the *Classification Batch Processing for Runtime API* 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

5. 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.

> (i) **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 3.0, 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 Known Issues

- For the adaptation task, it is unable to build its API example codes in Visual Studio 2015. Use Visual Studio 2017 or 2019 instead to build them and run the examples. See Adaptation Mode: Adapting to Line Variation for more details of the adaptation.

- For the training of High Detail tools, the memory consumption can soar up and cause unexpected behaviors. To solve this, you need to use different DLLs for the initialization of High Detail training. For more details, see PC memory consumption soars up when you train High Detail tools with API.

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
.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

Library	Obsolete Name	New Name
.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.Occurences	IModel.Occurrences
.NET	INode.Names	INode.Fielding
.NET	ITool.Parameters	ITool.ParameterBase
.NET	IFeature.Id	IFeature.Name
.NET	IRedView.Deviation	IRedView.Score
.NET	ITag.Key	ITag.Name
.NET	ITag.Value	ITag.Score
.NET	IWorkspace.Name	IWorkspace.UniqueName
.NET	IWorkspace.LastModification	
.NET	IToolDatabase.SetTrainFlag(string filter, string group)	IToolDatabase.SetTrainFlag(string filter, bool flag)
.NET	IToolDatabase.GetOverlayImage(ViewKey, string, uint)	
.NET	ILibraryAccess.ToolGetOverlay(WorkspaceName, StreamName, ToolName, string, uint, string, ILibraryAccessContext)	
.NET	IStream.AddImage(Image, string) IStream.AddImage(System.IO.Stream, string) GetImage(string, uint)	
.NET	ITool.Mask	ITool.RegionOfInterest.Mask
.NET	ITool.Parameters	ITool.ParametersBase
.NET	IToolParameters.TrainSelection	
.NET	IToolParameters.KeepTrainSelection	
.NET	BorderType.White BorderType.Reflect BorderType.Wrap	
.NET	Node.Names Node.IDs	Node.Fielding
.NET	ProductCode.OptimizedMemory	ProductCode.Standard ProductCode.Advanced

Library	Obsolete Name	New Name
.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()

Documentation Updates

VisionPro Deep Learning 3.0

- The document structure and contents organization is reformed for the clear understanding, quick and easy access to the necessary contents, and future enhancement and rearrangement of the contents.

- Fixed a typo "8192x819" to "8192x8192" in *Known Issues* on page 7 - *Tools - General* on page 8 section

- Fixed contents into that the Expert Mode does not support Edit Mask and Edit Region in Red Analyze . For example, see Edit Mask.

- Updates about Adaptation Modes

    ◦ Adaptation Mode: Adapting to Line Variation

    ◦ General Workflow

    ◦ Green Classify High Detail Mode Parameter

    ◦ Red Analyze High Detail Mode Parameter

- Updates About Label Check

    ◦ Optimize Labeling with Label Check

- Updates on Parameter Search and its Auto mode.

    ◦ Parameter Search

    ◦ Manual Mode

    ◦ Auto Mode

- Updates on VisionPro Deep Learning GUI refinement

    ◦ GUI and API

    ◦ Interpret Results (all tools)

- Updates on Red Analyze High Detail Validation Loss Formula

    ◦ Validation Set and Validation Loss

- Updates about Indicators (Training Set Indicator, Validation Set Indicator) on Image Display Area, Updates about Filters for Validation Set

    ◦ GUI Overview

    ◦ Prepare Validation Set (Red Analyze High Detail)

    ◦ Prepare Validation Set (Green Classify High Detail)

    ◦ Default Display Filters

    ◦ Tool-Specific View-Level Variables

- Updates about View Browser List Mode for all tools

    ◦ Define Region of Interest (ROI) and Mask to Construct View (all tools)

    ◦ Collect Images (all tools)

    ◦ Create an Image Sample Set (all tools)

    ◦ Modify Training Set (all tools)

    ◦ Review Markings (Red Analyze, Green Classify)

    ◦ Create Label (Labeling) (Red Analyze, Green Classify)

    ◦ View Browser - Grid Mode

    ◦ View Browser - List Mode

- Updates about TensorRT for Runtime API and Batch Processing for Runtime Green Classify

    ◦ *API Changelog* on page 34

- Updates about Cognex Label Import/Export Plugin

    ◦ Cognex Label Import/Export Plug-in

- Updates about Image Processing Logic Change for Red Analyze High Detail

    ◦ Configure Processing Parameters

- Updates about Minimum Image Size for Green Classify High Detail and Red Analyze High Detail.

    - Collect Images (Green Classify High Detail)

    - Collect Images (Red Analyze High Detail)

- Updates about Validation Loss Scale Change

    ◦ Validation Set and Validation Loss (Red Analyze High Detail)

    ◦ Validation Set and Validation Loss (Green Classify High Detail)

- Updates about Image Resize Improvement for Green Classify High Detail

    ◦ Resize

- Removing Red Analyze High Detail Quick Contents

    ◦ *Removal of Red Analyze High Detail Quick* on page 20

- Removing Stride parameter from Red Analyze High Detail Training parameters

    ◦ Training Parameters

- Updates about Migrating SuaKIT Segmentation and Classification Projects into VisionPro Deep Learning

    ◦ Import a SuaKIT Segmentation Project

    ◦ Import a SuaKIT Classification Project

- Updates on FAQ

- Updates on VisionPro Deep Learning 3.0 Known Issues

    ◦ Tools - Red Analyze

    ◦ Tools - Green Classify

    ◦ Parameter Search

    ◦ GPU

- Updates on VisionPro Deep Learning 2.1 Known Issues

    ○ Parameter Search

- Updates on High Detail training logic regarding Patience Epochs and Minimum Epochs in Training parameters.

    ○ Training Parameter Details: Patience Epochs & Minimum Epochs

- Updates on Best Epoch update logic in Loss Inspector

    ○ Status Variables

- Updates on the compatible VisionPro version

    ○ Integration with Cognex VisionPro

- Updates on the GPU driver versions for NVIDIA RTX / Quadro® / Data Center driver

    ○ [Pre-requisites: NVIDIA® GPU Driver](#)

- Updates on Designer Plug-in support

    ○ [Using Runtime Workspace with Cognex Designer](#)

VisionPro Deep Learning 2.1

- Updates on Red Analyze Tool and Green Classify Tool for newly added High Detail Quick modes

- Updates on Edit Deployment Target page that introduces the CIE1 deployment is no longer available

- Updates on Parameter Search and Use the Parameter Search Utility

- Updates on Image File and Format Requirements.

- Updates on the contents explaining statistical results

    ○ Statistics

    ○ False Positive and False Negative Results

    ○ Blue Locate Tool – Statistics

    ○ Blue Read Tool – Statistics

    ○ Green Classify Tool – Confusion Matrix

    ○ Green Classify Tool – F Score

    ○ Red Analyze Tool – Statistics – Confusion Matrix

    ○ Red Analyze Tool – Statistics – Region Area Metrics

- Updates on the model scores for Blue Locate/Read

    ○ Blue Locate Tool – Node Model

    ○ Blue Locate Tool – Layout Model

    ○ Blue Read Tool – Node Model

    ○ Blue Read Tool – Regex Model

    ○ Blue Read Tool – String Model

- Updates on C#, C++ example codes (...\ProgramData\Cognex\VisionPro Deep Learning\2.1\Examples\). There were many improvements in example codes and some of the notable changes are:

  ○ The unit of angle for input is changed from "degree" to "radian" (example: Perturbation of Red Analyze)

  ○ For Green Classify, the codes that load label information per image is added.

  ○ The parameter path used for *vidi_runtime_get_parameter()* and *vidi_runtime_set_parameter()* is changed to be a "full path" which should include "tool/" within it.
  Example: "*workspaces/(workspace_name)/streams/(stream_name)/tools/(tool_name)/**tool**/processing/sampling_density*"

  > **Note**: When either "roi/", "runtime_parameters/", or "post_process_parameters/" are used in the parameter path, this path should not include "tool/".
  > ⓘ Example 1: "*workspaces/(workspace_name)/streams/(stream_name)/tools/(tool_name)/**roi**/...*"
  > Example 2: "*workspaces/(workspace_name)/streams/(stream_name)/tools/(tool_name)/**runtime_parameters**/...*"

- Updates on page contents.

VisionPro Deep Learning 2.0

- Updates for VisionPro Deep Learning 2.0 release

  ○ The documentation for the integration with Cognex Designer is no longer provided

VisionPro Deep Learning 1.1

- Updates for VisionPro Deep Learning 1.1 release

- Designer Plugin script examples for each tool have been updated for VisionPro Deep Learning 1.1.

- Updates of the comments in C#, C++ example codes (...\ProgramData\Cognex\VisionPro Deep Learning\1.1\Examples\)

- Updates about how to use multiple GPUs for VisionPro Deep Learning in Multiple GPUs Utilization.

- Updates about the introduction to the various training parameters for VisionPro Deep Learning in Training Parameters

- Updates about how to import SuaKIT projects in Import SuaKIT Project.

VisionPro Deep Learning 1.0.1

- Updates for VisionPro Deep Learning 1.0.1 release

- Updates for Migrating To This Release

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

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

Preparation

This topic covers the computer operating systems, hardware requirements, dongle configuration, licenses, and legal notices you need to be prepared with before using VisionPro Deep Learning for your application.

Pre-requisites: Operating System

Microsoft® Windows® Operating System	English	Chinese	Japanese	Korean	French	German	Spanish
Windows 10 Professional (64-bit)	✓	✓	✓	✓	✓	✓	✓
Windows Server 2019	✓	✓	✓	✓	✓	✓	✓

ⓘ **Note**: The use of the VisionPro Deep Learning software or the Deep Learning Service on a virtual machine (VM) is not supported.

ⓘ **Note**: Due to the issues from the NVIDIA driver, Windows Server 2016 is not supported for VisionPro Deep Learning3.0 Server/Client mode. See for more details and the workaround.

Pre-requisites: Training PC Requirements

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

- CPU

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

- GPU

  When selecting a GPU, Cognex only supports NVIDIA GPUs.

  For training, we recommend GPUs that have 10GB or larger GPU Memory (1080 Ti, 2080 Ti, 3080). For the processing of High Detail modes and High Detail Quick modes, GPUs that have 8GB or larger GPU Memory are recommended.

- System Memory (RAM)

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

  When selecting the system RAM, specify the greater of:

  - The sum of all GPU memory. For example, if you have four NVIDIA GeForce® RTX™ 3080 Ti GPUs, which each have 10 GB of memory, the PC should have 40 GB of RAM.

  - The RAM should be one and a half times the typical workspace size. For example, if your typical workspace is 20 GB, the PC should have a minimum of 30 GB of RAM.

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

- System Storage

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

- Power Supply

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

- USB 2.0

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

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

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

Pre-requisites: Deployment PC Requirements

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

- CPU

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

  For High Detail modes and High Detail Quick, if you have only CPU; not GPU, you cannot process the tool.

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

- GPU

    When selecting a GPU for processing, VisionPro Deep Learning3.0 only supports NVIDIA GPUs. Consider the following when choosing an NVIDIA GPU:

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

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

    - These GPUs support the compute-optimized Tesla Compute Cluster (TCC) mode.

    - These GPUs are designed and tested for continuous duty-cycle operation.

    - These GPUs undergo rigorous testing and qualification by NVIDIA.

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

    8 GB of GPU memory is recommended for runtime. For the processing of High Detail modes and High Detail Quick modes, GPUs that have 8GB or larger GPU Memory are recommended.

- System Memory (RAM)

    We recommend 32GB or higher.

- PCIe Lanes

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

- Power Supply

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

- USB 2.0

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

> **Note**:
> The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.
>
> When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.

Pre-requisites: NVIDIA® GPU Device Requirements

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

VisionPro Deep Learning3.0 aims to support the most common GPU families, but it's not practical to test every model or driver. The table below indicates those models we have explicitly tested and confirmed work, as well as those few models and families that do not work. Most other common NVIDIA GPUs other than very new models have been used successfully by customers, even if they haven't been explicitly tested. In addition, we share specifications for minimum GPU capability, as well as recommendations high performance GPUs for demanding applications, such as those using Green Classify High Detail or Red Analyze High Detail.

- Minimum recommended performance: GTX 1060 6GB

- Recommended for high performance: GTX 1080 Ti / RTX 2080 Ti / RTX 3080

GPU	Model	Tested by Cognex	Recommended for High Performance
NVIDIA GeForce®	GTX 1060 6GB	Tested	
	GTX 1070		
	GTX 1080	Tested	
	GTX 1080 Ti	Tested	Recommended
	RTX™ 2070		
	RTX 2080		
	RTX 2080 Ti	Tested	Recommended
	RTX 3060		
	RTX 3060 Ti		
	RTX 3070		
	RTX 3070 Ti		
	RTX 3080	Tested	Recommended
	RTX 3080 Ti	Tested	
	RTX 3090	Tested	Recommended
NVIDIA RTX / Quadro®	P2000		
	P4000	Tested	
	P5000		Recommended
	GV100*		Recommended
	RTX 4000		
	RTX 5000		Recommended
	RTX 6000		Recommended
	RTX 8000		Recommended
	Titan V*		Recommended
	Titan RTX		Recommended
NVIDIA Tesla®	V100*		

GPU	Model	Tested by Cognex	Recommended for High Performance
Models known not to work:				
• GTX 1660 and GTX 1650: Please note that low performance and slow training time with unknown reason.				

> **Note**:
> - When selecting a GPU, Cognex only supports NVIDIA GPUs.
>
> - VisionPro Deep Learning3.0 supports the use of any NVIDIA CUDA® enabled GPU that provides a compute capability of 3.0 and higher, and requires a driver version greater than or equal to 466.77.
>
> - CUDA compute capability 3.0 or higher is required to use the core VisionPro Deep Learning functionality. The Low Precision parameter requires CUDA compute capability 6.1 or higher (for more information, visit the [NVIDIA](#) website).
>
> - When training and processing High Detail mode and High Detail Quick mode, a requirement of 8 GB of GPU memory is for runtime operation, and 10 GB of GPU memory is recommended for training.
>
> - The NVIDIA GeForce Titan V, NVIDIA RTX / Quadro® GV100 and Tesla V100 include NVIDIA Tensor Cores. VisionPro Deep Learning3.0 will automatically utilize Tensor Cores if they are present, provided you have a Standard or Advanced security license. Please contact your Cognex Sales Representative for information about enabling the functionality on your Cognex Security Dongle.

Pre-requisites: NVIDIA® GPU Driver

- GeForce® RTX driver greater than or equal to 466.77

- NVIDIA RTX / Quadro® / Data Center driver greater than or equal to 466.77 (Optimal Driver for Enterprise).

    ○ It is highly recommended to visit https://www.nvidia.com/Download/Find.aspx to find the right versions for your GPU as your GPU may require higher version than 466.77.

Pre-requisites: API Development Requirements

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

- Microsoft® Visual Studio® 2015, 2017 or 2019

- Microsoft .NET Framework 4.7.2

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

Pre-requisites: Image File and 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.

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

Maximum Image Size and Dimensions

The following guidelines will help you determine how best to configure your Deep Learning application, within these operational parameters. The figures in the table below (for example the maximum number of images per stream) are experimentally tested and the values vary more or less on which hardware settings you rely. Generally, the larger your memory (not GPU RAM) and hard disk capacity you have, the more the number and the size of the image you can handle.

- Maximum image size (Channel parameter set to 1 to 4)
    - Images up to 32MP (up to 16k in height or width) are supported
    - Images up to 32k pixels wide (in one dimension, dependent upon GPU resources)
    - Images over 32k are not supported
- Maximum number of images per stream for Red Analyze
    - 1000
- Maximum number of defects per image for Red Analyze
    - 1000

Tip: As the number and the size of images are getting larger, the training and processing time for all images increases as well.

Install VisionPro Deep Learning

To successfully install VisionPro Deep Learning, perform the following:

1. Attach the Cognex Security Dongle to a USB port on the computer that will be used to develop the vision application.

2. Download the CognexVisionPro Deep Learning installer from the Cognex support page.

3. Run the VisionPro Deep Learning installer and follow the prompts.

> **Note**:
> Choosing the Custom option will allow you to install selected features:
>
> - The Wibu Runtime Server, which is needed to connect to the USB Cognex Security Dongle.
>
> - The main VisionPro Deep Learning application (this is required).
>
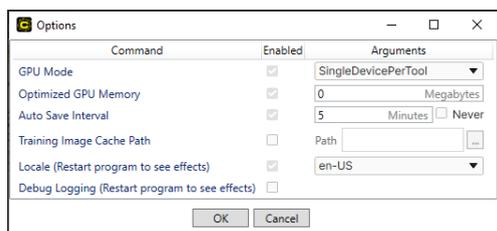> - The VisionPro Deep Learning Developer API.
>
> - The VisionPro, Designer and VisionPro Deep Learning software require that a valid Cognex Security Dongle be installed directly to PCs running the software during all phases of operation (programming, processing, training, testing, etc.). Any attempts to temporarily remove, substitute, or share a Cognex Security Dongle may cause your system to operate incorrectly, and may result in the loss of data.
>
>   When VisionPro Deep Learning is configured for the Client/Server functionality, and a computer has been configured as a server, the Cognex Security Dongle must be attached to the server, but the client(s) does not need a Cognex Security Dongle.

Launch VisionPro Deep Learning

When launching the Deep Learning GUI, there are several options that you can use to control the GPU Mode, which GPU device to use and the allocation of GPU memory, in addition to project settings.

At the Starting Cognex Deep Learning screen, press the **Options** button.

Command	Settings
GPU Mode	Specifies the GPU mode to be used by the application.
	SinglePerDeviceTool — A single GPU is used for training and processing an image.
	MultipleDevicesPerTool — *MultipleDevicesPerTool* that aimed to provide processing an image on multiple devices has no longer been supported since VisionPro Deep Learning 1.0.0.
	NoSupport — Specifies that a GPU will not be used.
	(i) **Note**: This option conflicts with --gpu-devices and/or --optimized-gpu-memory.
Optimized GPU Memory	Specifies the size of the pre-allocated optimized memory buffer. This setting is activated by default, with the default size of 2GB.
	💡 **Tip**: **Compute Devices** You can set Optimized GPU Memory in the Help > Compute Devices option. Compute Devices window contains the information about computing resources. 1. **Available Devices** This shows the GPUs that are used for VisionPro Deep Learning. You can check list of multiple devices and GRAM information here. 2. **Optimized GPU Memory** You can enable/disable GPU memory optimization for the use of Focused mode tools. See NVIDIA GPU Selection and Configuration for the details.
Auto Save Interval	Specifies how often a workspace will be auto-saved. The default is every 5 minutes.
Training Image Cache Path	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 Training Image Cache Path, the conversion is only performed once and the converted images are stored in a local cache directory. The cache directory should be local, and 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.
Locale	Specifies the language to use throughout the VisionPro Deep Learning GUI.
Debug Logging	Specifies whether or not debug logs should be activated for the project.

CognexVisionPro Support

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

- VisionPro9.8

- VisionPro 9.8

For more information about the integration with CognexVisionPro, see Integration with Cognex VisionPro.

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 © 2020. 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.

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, BGAII, 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.