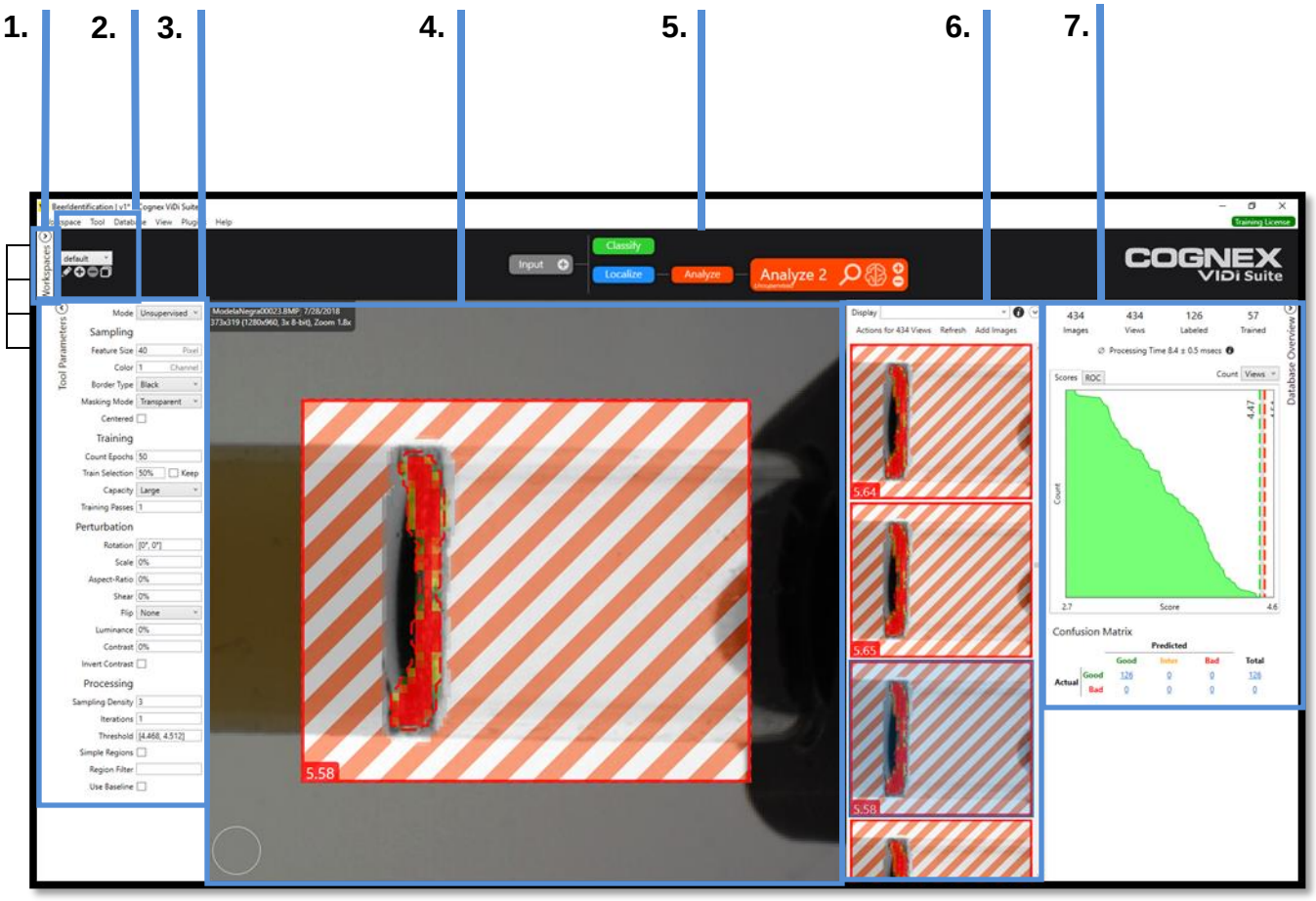
Contents

Lab Exercise 1.1 – ViDi Introduction

At the end of this lab exercise, Participants will be able to:

- Name the different panels in the ViDi GUI

The Participant will utilize the ViDi GUI to name each panel in ViDi Suite.



1.	
2.	
3.	
4.	
5.	
6.	
7.	

Lab Exercise 2.1 – ViDi Red – Analyze Tool Unsupervised

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi Red – Analyze tool to solve their application

The Participant will utilize the following ViDi tools to successfully complete this exercise:

- Red – Analyze Tool
  - Unsupervised

1. Textile Inspection
Red – Analyze Tool

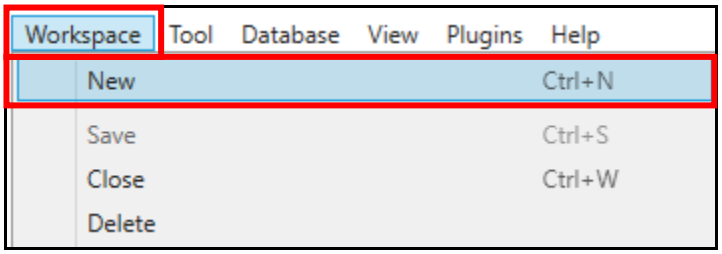
Inspecting textiles is a typical example of aesthetic inspection. Defects can be many different types, ranging in problems in the weaving, defective or bad yarn, stains and much more. The challenge is even bigger when the fabric is not perfectly aligned in front of the camera, it can shift and rotate quite dramatically.

This lab exercise shows a typical example using a couple of images from a textile which shows a relatively complex yet repetitive color pattern. The images are taken from random positions such that each shows a distinct precise view.

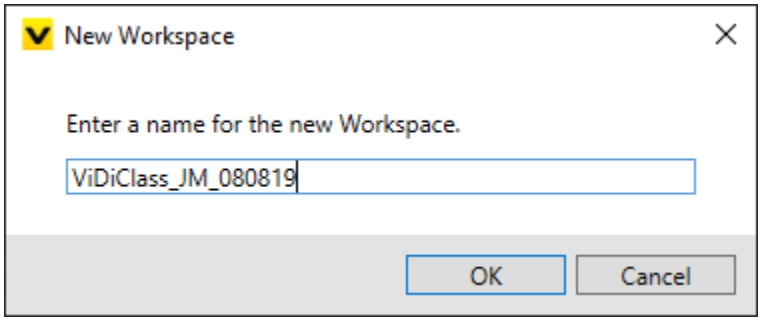
The images show a good degree of shear and rotation to demonstrate the Red – Analyze Tool's flexibility to cope with this.

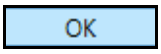
Follow the steps below to complete the lab exercise:

1. Create a Workspace by clicking on the **Workspace** menu and selecting **New**.



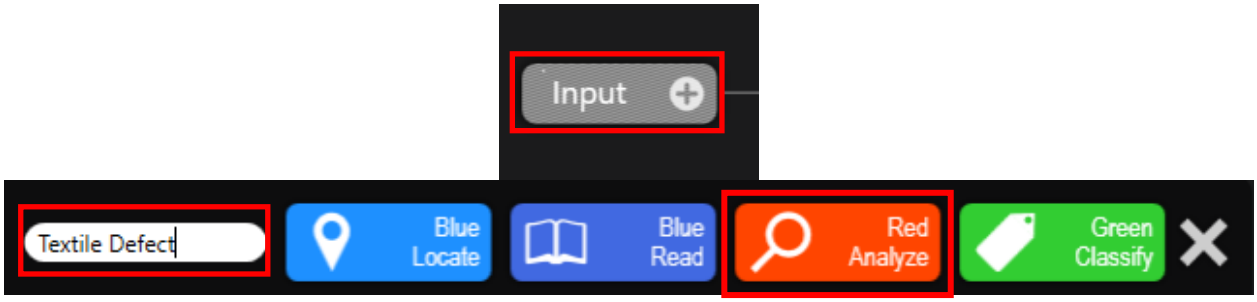
2. Name the Workspace in a recognizable manner.
   Suggested: ViDiClass_[First Initial][Last Initial]_MMDDYY.



3. Click the **OK** 

button.
NOTE: *We will be utilizing this Workspace through the class.*
4. Click the **Rename Stream**  button to rename the Default Stream to something more representative of the application.
For this example, we suggest using **Textile Example**.

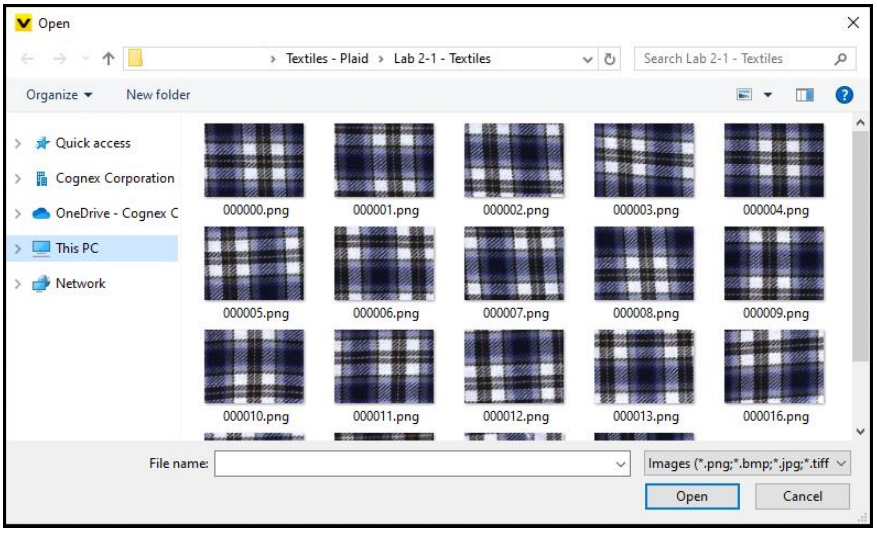
5. Add a **Red Tool** by first clicking on the Input button, adding the desired name, then clicking the Red – Analyze Tool Button.
The recommended name for this tool is **Textile Defect**.

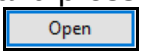


6. Once the Tool has been added the next step is to add the images. To do this click the **Add** 

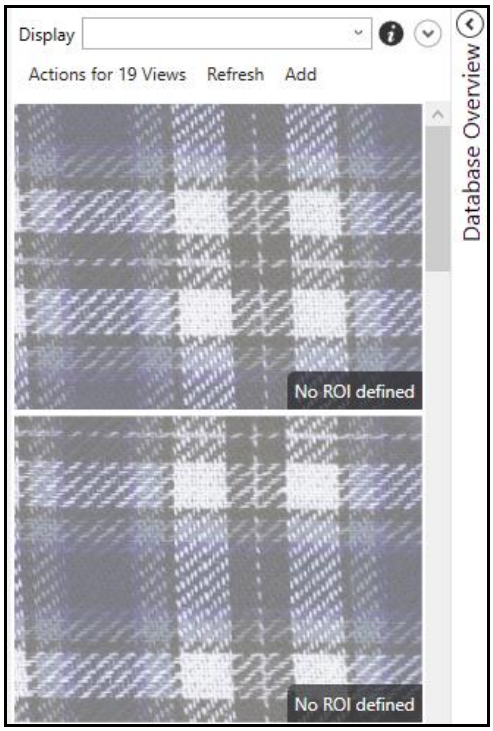
button in the View Panel.

7. Browse to the folder containing the desired images for the application. For this example, the folder will be **ViDi Student folder → Labs → Textile Tutorial**.



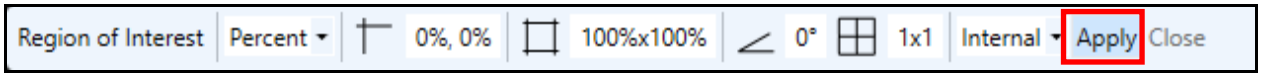
8. All of the images will be added, select the first image and press **<Ctrl + A>**, this will automatically select all of the images. Click the **Open**  button.

9. Once the images have been added, they will appear on the View Panel. Click an image to select.



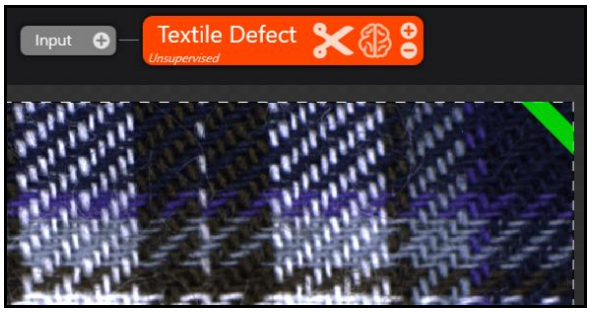
10. After selecting a View, you must define the Region of Interest (ROI). Once a view is selected, what you see below should appear on the screen.
    This bar allows the user to determine what ViDi will consider when interacting with this image. A border excluding regions of the image can be added, the angle of the ROI can be changed, and it may be split into multiple sections.

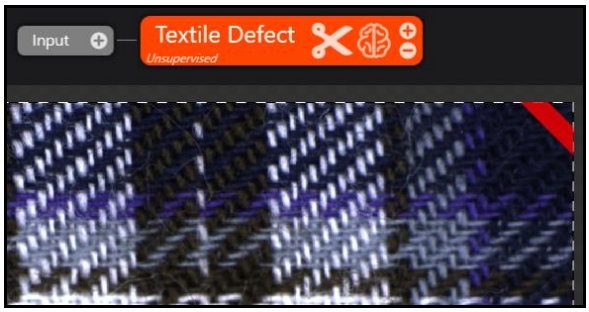
    For this example, click **Apply**, as we will be utilizing the whole image.



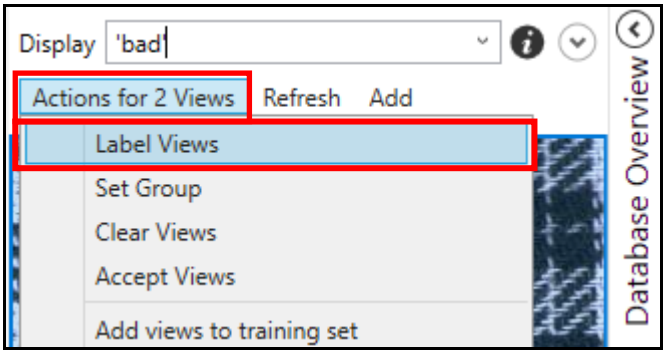
    NOTE: *Steps 4 – 10 are the standard method for adding images to a tool. As such they will be utilized for all of the following labs. The only differences may be the choice of color tool, tool name, image folder, and/or the ROI settings.*

11. Next, we will label the view.
    This can be done one of two ways.
    1. Clicking the image on the Main image window will Label it as a Good (Green Strike on Top Right) image, clicking a second time will label it as a Bad (Red Strike on Top Right) image, a third time will ask you if you would like to label it as Good again.





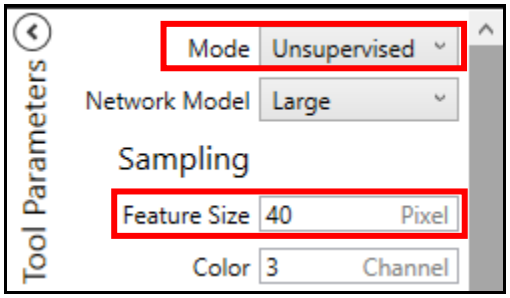
    2. Another method is to use the "**Actions for # Views**" option. This allows action to be taken on all views currently displayed in the View Selection Window. The views in this window can also be filtered by utilizing the "**Display**" option with filter functions.



12. Label all images without defects as Good.

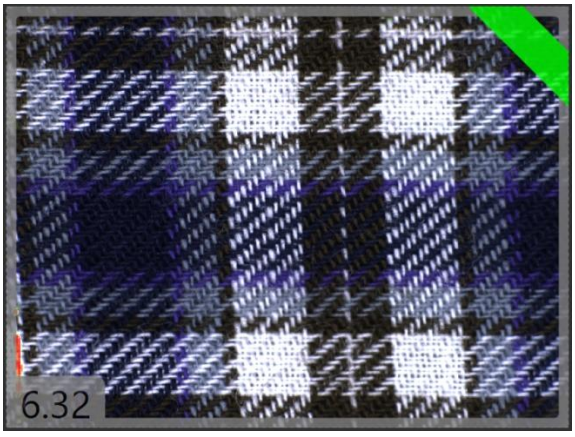
    NOTE: *This additional step is optional, since we are using the Red – Analyze tool in Unsupervised mode, label all images with defects as* ***Bad****. This can be done by searching display for "****bad****", all bad images will include "bad" in their file name.*

13. Typically, at this point we would adjust the tool parameters to better match our application, but for this example we will confirm that the **Mode** is set to *Unsupervised*, and the **Feature Size** is set to *40*.

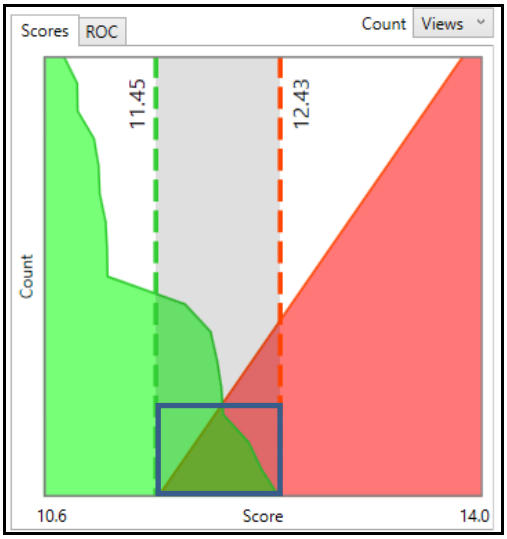


14. Train the tool by clicking the **Train the Tool** button.

15. At this point we should have images and statistics results. We will analyze these to see which actions need to be taken to get perfect, or near perfect results.
Below are some examples of Good and Bad results that have *not* been classified, this is indicated by the gray Border.

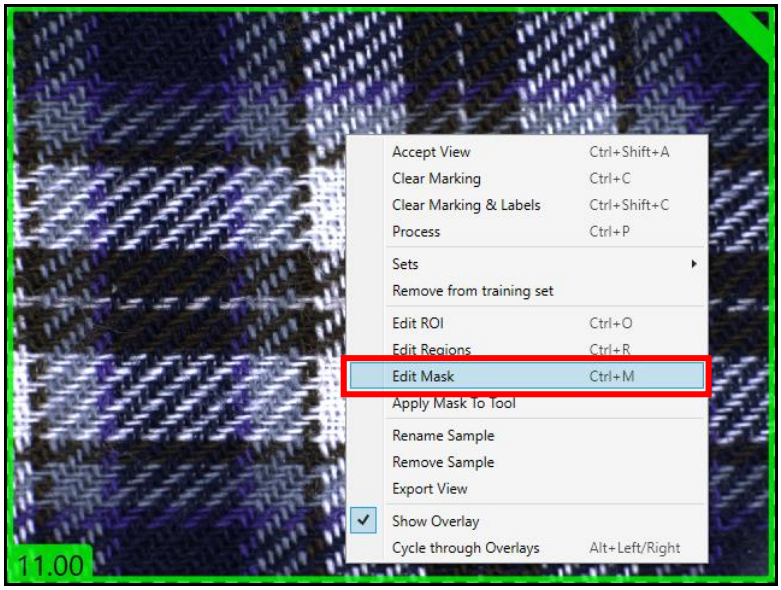


This is because there is an area where a score is between Good and Bad and the tool is unable to detect which is which. This can be Viewed in the **Scores** Graph, which is found in the Result Tab.



16. Note, that on the unclassified good image the heatmap shows defects along the edge. This is because ViDi is looking outside of the image, which right now by default is set to Black. To avoid blocking ViDi from looking at these images we need to add a mask.

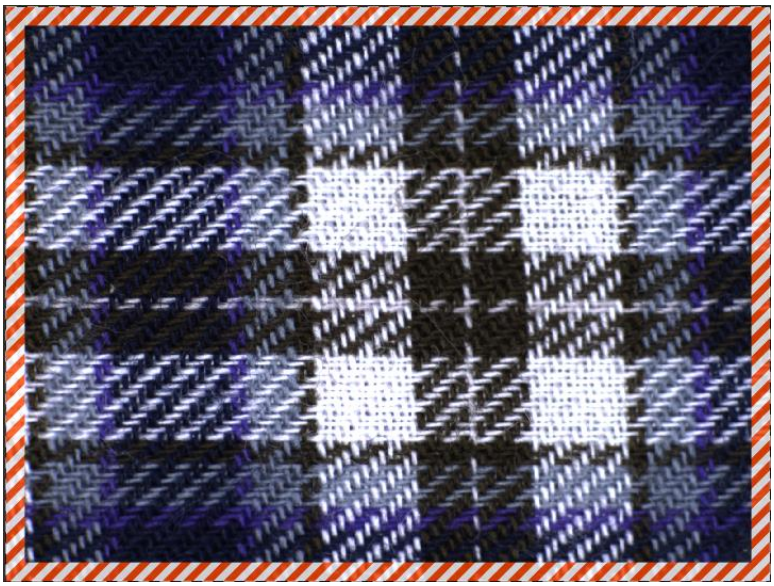
    To do this right click on the image in the main image window and select **Edit Mask**. The menu below will pop up on the image.



    The **Mask** toolbar displays.

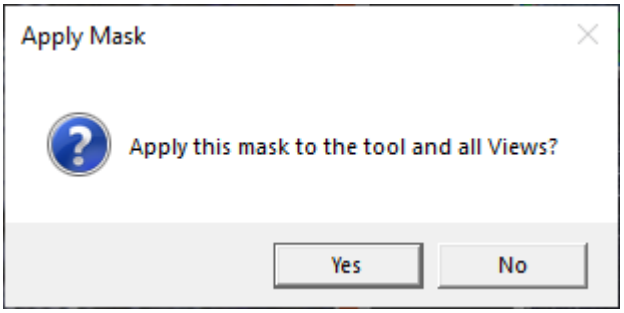
17. Click the **Add Mask** button.

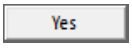
A diagonal striped border will appear around the image with a width equal to the setting in the toolbar (40).



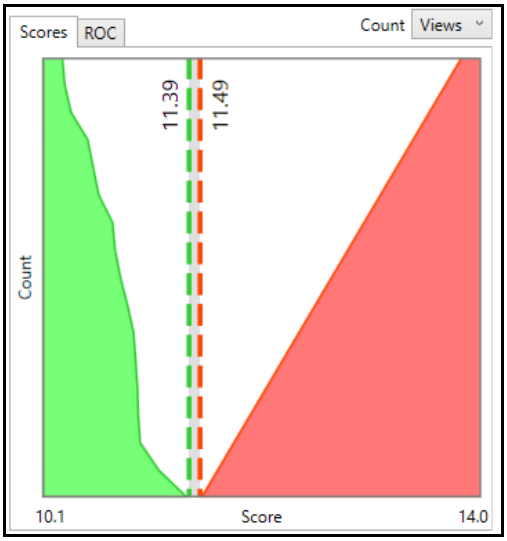
18. Click **Apply** on the Mask Tool Bar.

The **Apply Mask** dialog displays.



19. Click the **Yes**  button to **Apply this mask to the tall and all Views?**
20. Retrain by clicking the **Train the Tool** button.

Your results should look similar to the one below. All Good images should now be maked Good and all Bad images should be marked as Bad.



Confusion Matrix

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	17	0	0	17
	Bad	0	0	2	2

Lab Exercise 3.1 – ViDi Red – Analyze Tool Supervised

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi Red – Analyze Tool in Supervised mode to solve their application

The Participant will utilize the following ViDi tools to successfully complete this exercise:

- Red – Analyze Tool
  - Supervised

2. Red – Analyze Tool *Supervised*
Texture Inspection

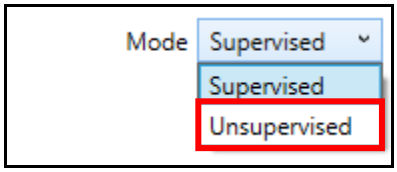
Textures often have a pseudo random character. They can be partially described by some basic characteristics – such as the direction or width of brush strokes. However, they also have a random character which changes from one sample to the next. The challenge is to accept these normal variations while still detecting small, yet visually disturbing defects.

While the Red – Analyze tool in its unsupervised mode can find anomalies in complex surfaces and textures, it requires that the anomalies are visually sufficiently salient to be separable from the background texture. If this is not the case, or essentially, if an unsupervised red tool is not responding to certain anomalies or defects as required or expected, you will need to switch to supervised mode.

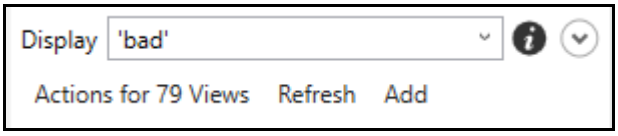
Follow the steps below to complete the lab exercise:

1. Follow steps 4 – 10 from exercise 2.1 changing the following:
   Stream Name: Texture Example
   Tool Name: Defect Detect
   Tool: Red – Analyze
   Image Folder: Texture
   ROI: Full Image

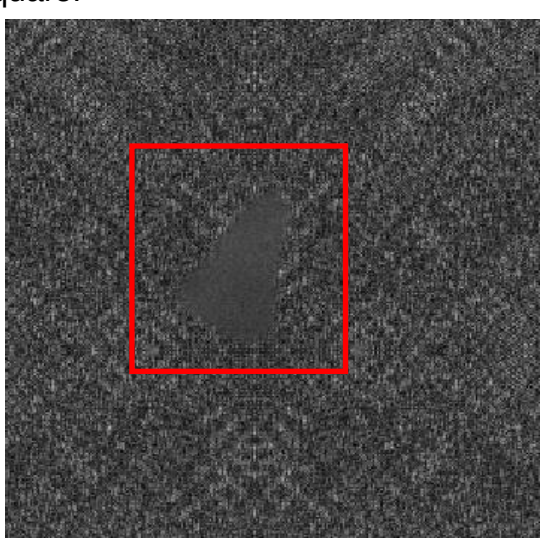
2. Change the Tool Mode from Unsupervised to **Supervised**.

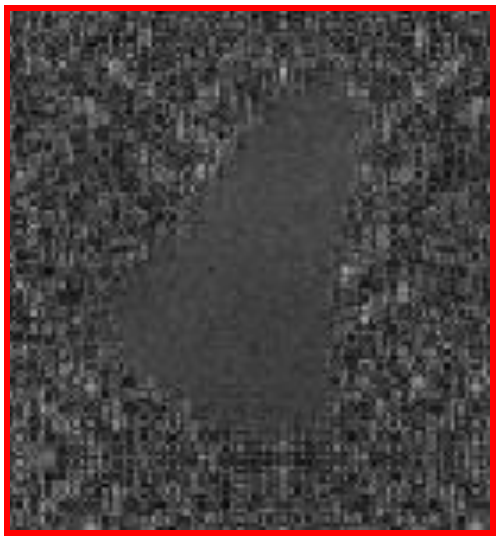


3. Filter the View Results by using the '**bad**' search term on the Display field. This will only display the images with defects.

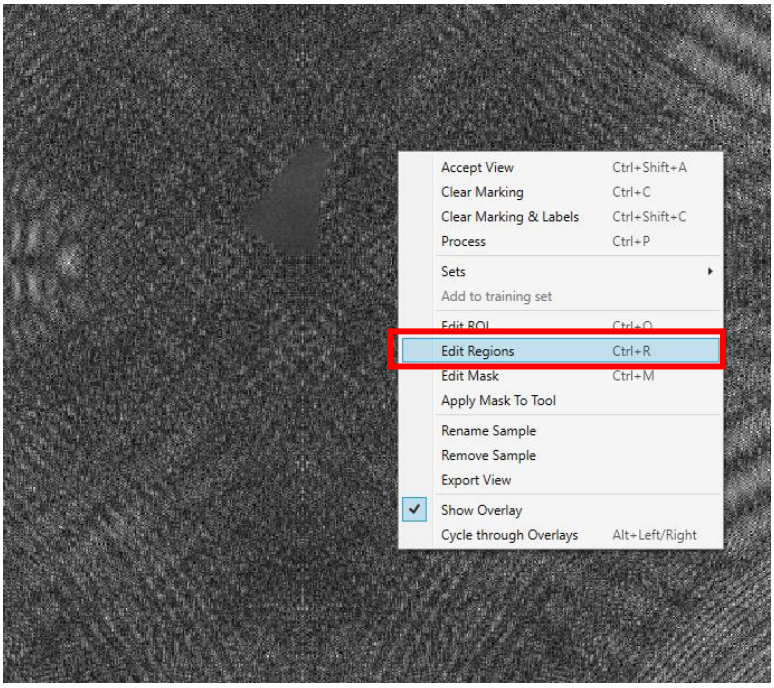


4. Select an Image by clicking on it in the View Panel.
5. Identify the defect on the image. The example below has its defect inside of the square.



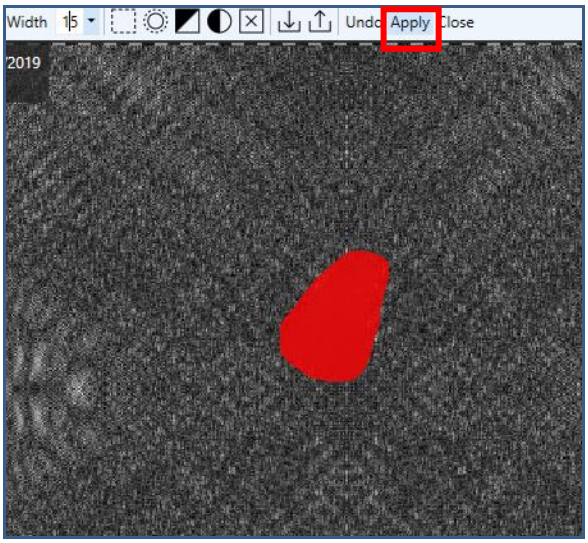


6. Right Click on the image inside of the Image Panel and select **Edit Regions**.

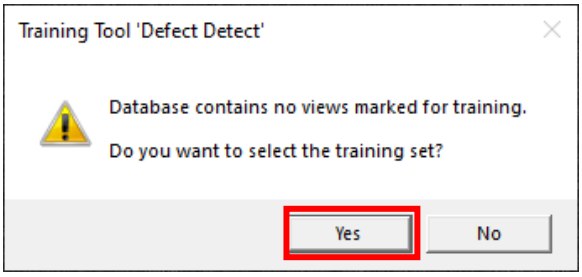


The Region toolbar will display.

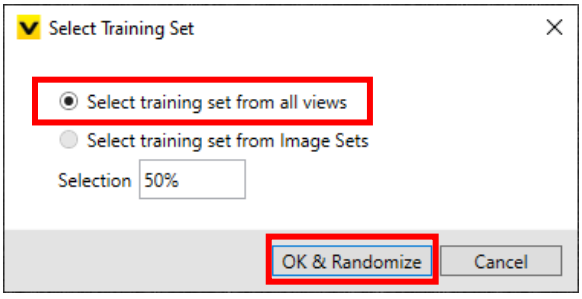
7. For this example, we will be selecting the **Line Tool**. This will allow us to mark by clicking and dragging our mouse across the defect until it is filled in. Fill in the mistake as is displayed in the image below. Once this is done click **Apply**.
NOTE: *The width may be too large at 40, lower the width to 15. You will want to fill in right at line of the defect, small mistakes are fine but it's better to avoid them.*



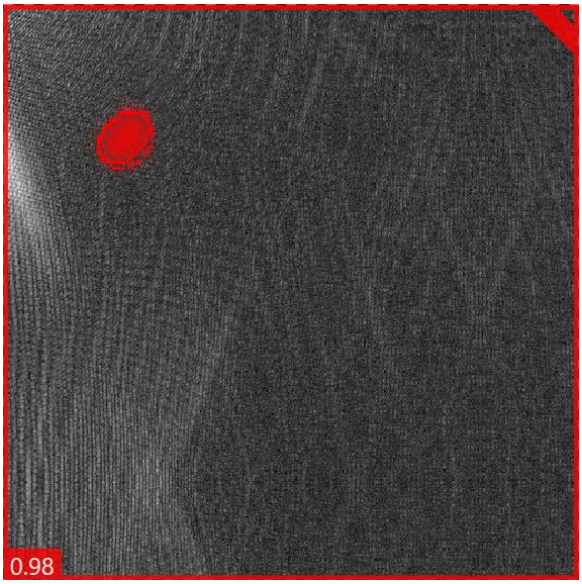
8. Repeat this process for approximately 30 images.
9. Train the system by clicking the **Train the Tool** button.
The Warning dialog displays – click **Yes** to select the training set.



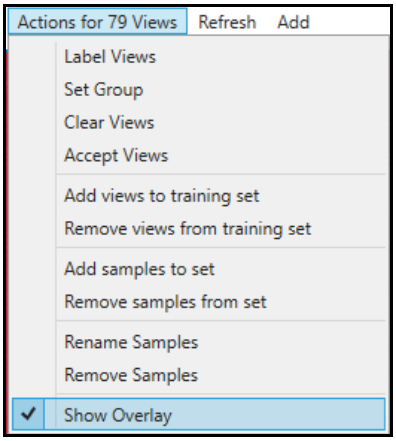
The **Select Traning Set** dialog displays – click the **Select training set from all views** radio button and click the **OK & Randomize** button to select the training set.



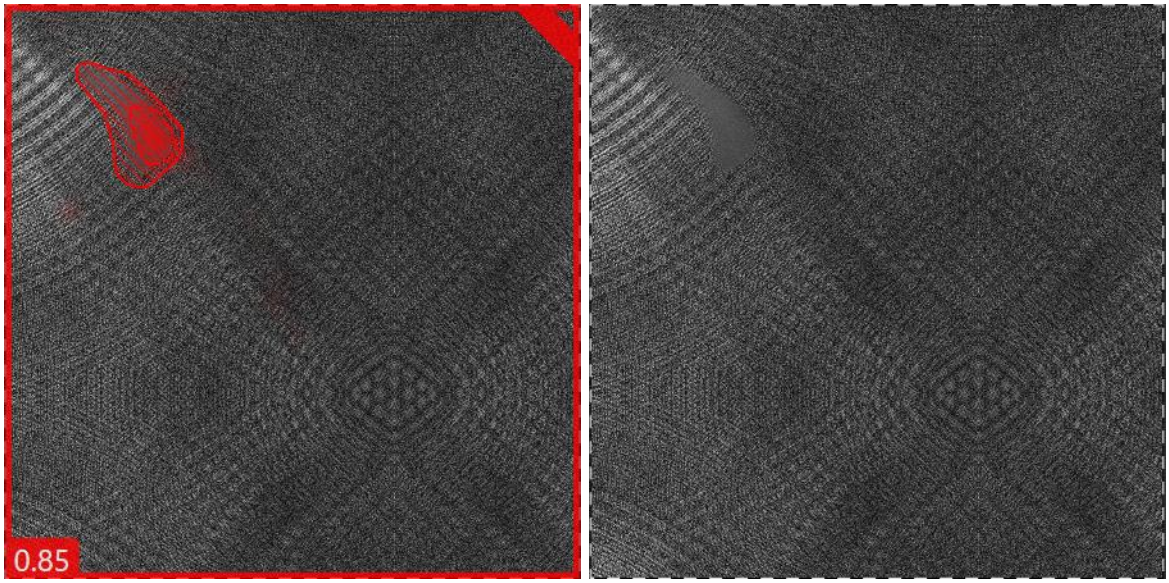
10. Review the results for the Bad images by searching the display for '**bad**', as this will not be close to perfect.
    Any defect that is detected will be outlined by a dashed red border. Defects that you have labeled will appear with diagonal stripes across them, as shown in the image below. There may be a small difference between Labeled and Marked defects.



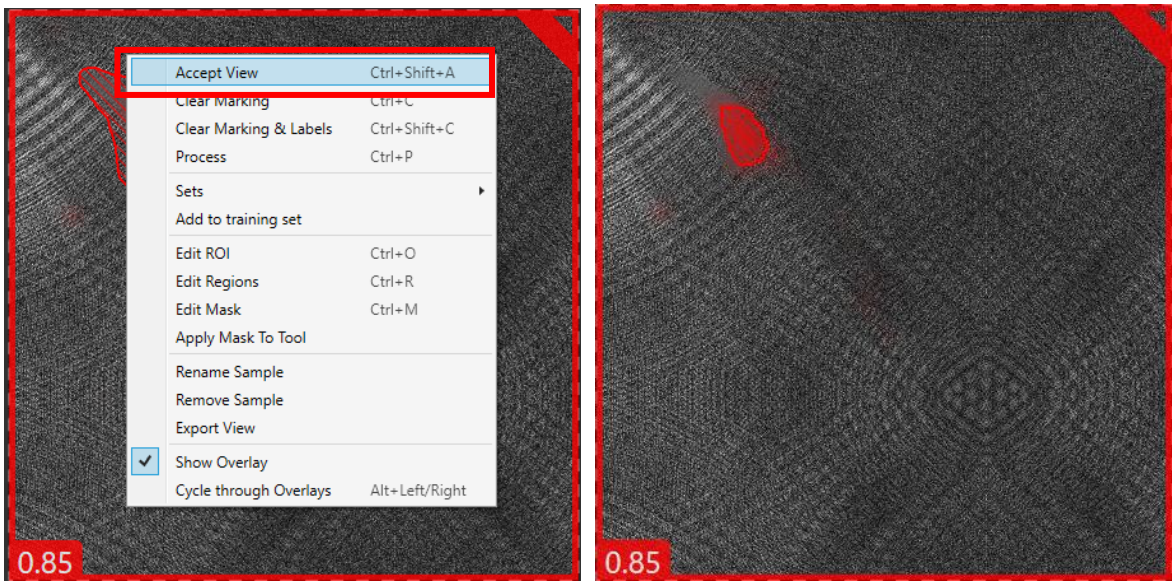
    Defects that were not labeled and images that have not been touched will be marked with a red dashed border, but nothing on the inside.
11. To confirm that the image is correctly marked deselect **Show Overlay** from the Actions for # Views.
    NOTE: *Using the Alt + arrow keys will also remove the overlay.*



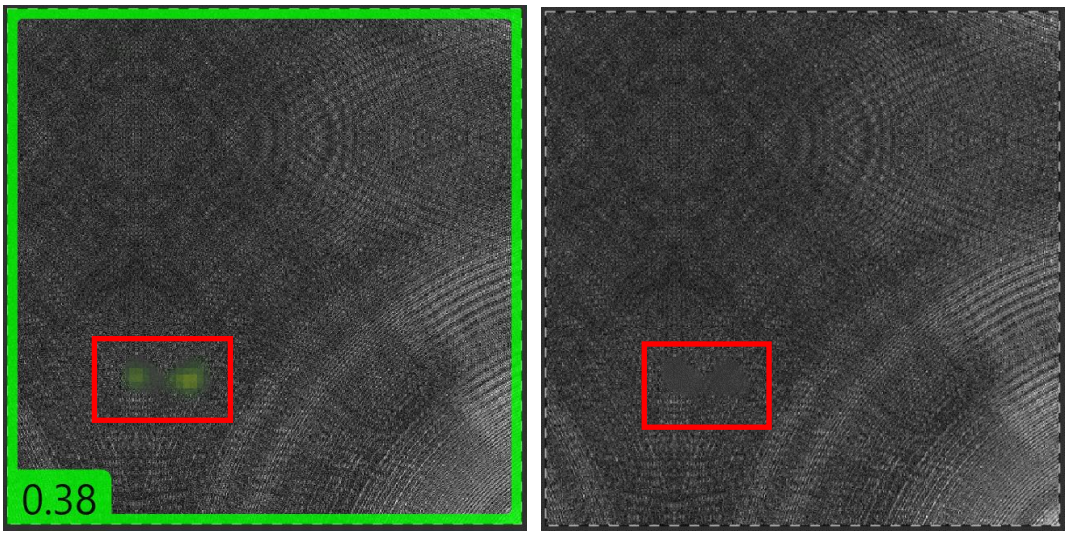
12. Any defect that is detected will be outlined by a dashed red border. Defects that you have labeled will appear with diagonal stripes across them, as shown in the image below. There may be a small difference between Labeled and Marked defects.



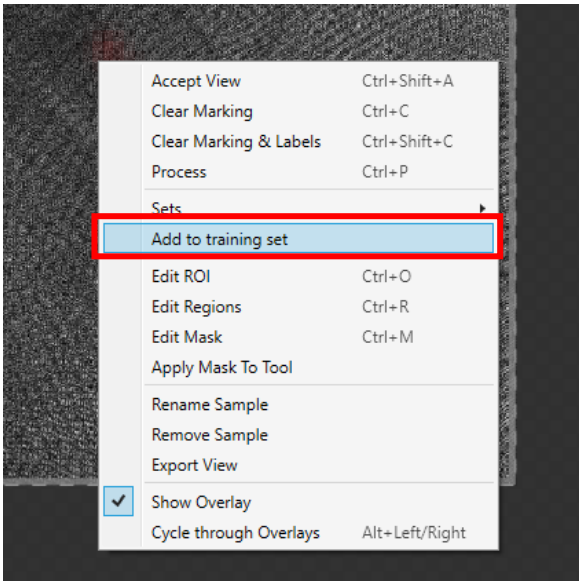
13. Check for these and add them to your set of labeled images by right clicking and selecting **Accept View**. Once this is done the Mark should now have diagonal stripes, meaning it has been added as a labeled image, increasing your possible training set.



14. Through checking bad images you will come across some images that have not been properly marked.

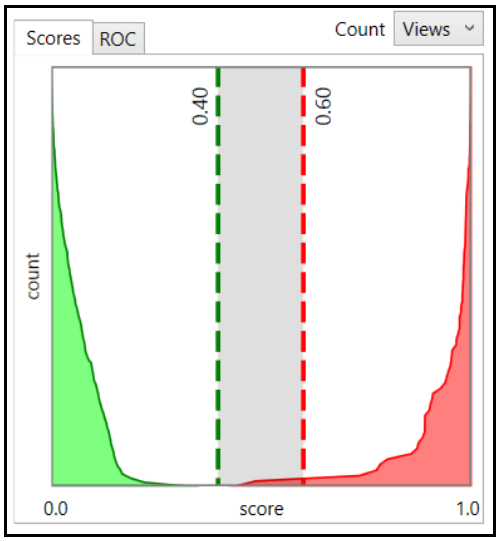


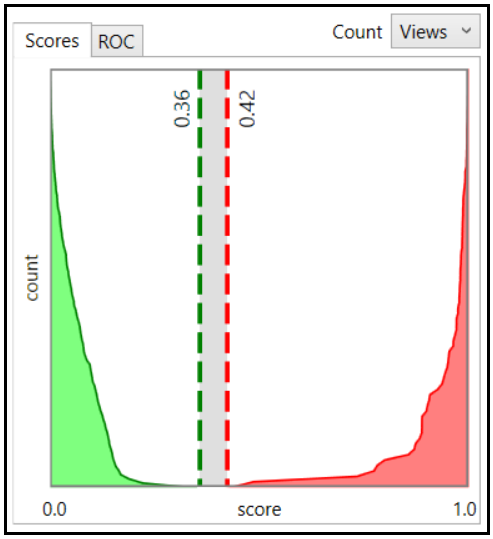
15. For these cases you will have to label them yourself. To do this right click and select **Edit Regions** and mark the defect as was done before.
Given that ViDi has trouble detecting the defects in these images they should be added to the training set. To do this right click on the image, and select **Add to Training set** from the list. This will always include this image in the training set giving ViDi a better change to learn it.



16. Label all the good images by searching the Display field for "**not 'bad'**". Use the "**Actions for Views**" option to label all images as good.
17. Retrain by hitting the **Train the Tool** Button.

18. Review and adjust the score graph.





3. Red – Locate Tool *Supervised* Battery Detection and Classification

This is the first part in a two-part lab. This lab exercise will improve a classification task by masking the changing background, we will touch on this in a later section. In this section we will

Follow the steps below to complete the lab exercise:

1. Add a new stream called **Battery Example.**
2. Add a **Red – Analyze Tool** to the stream called Battery Masking.
2. Set the parameters:
    - **Image Folder**: Battery
    - **Mode**: Supervised
    - **Feature Size**: 80
    - **Epoch Count**: 5 – 10
    - **Train Selections**: 100%
    - Perturbation Parameters
        - **Rotation** [-20°, 20°]
        - **Scale** 20%
        - **Aspect-ratio** 5%
        - **Shear** 5%
        - **Luminance** 10%
    - **Simple Regions**: ON
3. Label approximately 20-30 images utilizing the same method as described in lab. 3.1. In this example the battery is the defect and what we are looking to detect.
4. **Train** the tool.
5. Review and Adjust the results to ensure all batteries are being detected properly.
6. We will return to this example in a later section to demonstrate some of the amazing capabilities that the Red – Analyze tool has to offer.

Lab Exercise 4.1 – ViDi Blue – Locate Tool

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi tools to solve their application

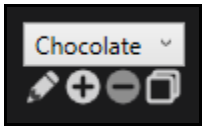
The Participant will utilize the following ViDi tools to successfully complete this lab exercise:

- Blue – Locate Tool

1. Chocolate Assortment Inspection

Follow the steps below to complete the lab exercise:

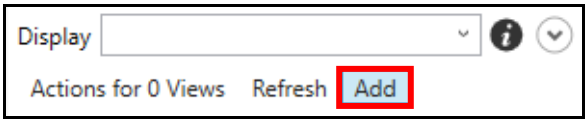
1. Click the **Add Stream** button, to add a Stream to the Workspace.
2. Click the **Rename Stream** button, rename the Stream *Chocolate* and press the **<Enter>** key.



3. Click the **Input** button to view the available ViDi tools.

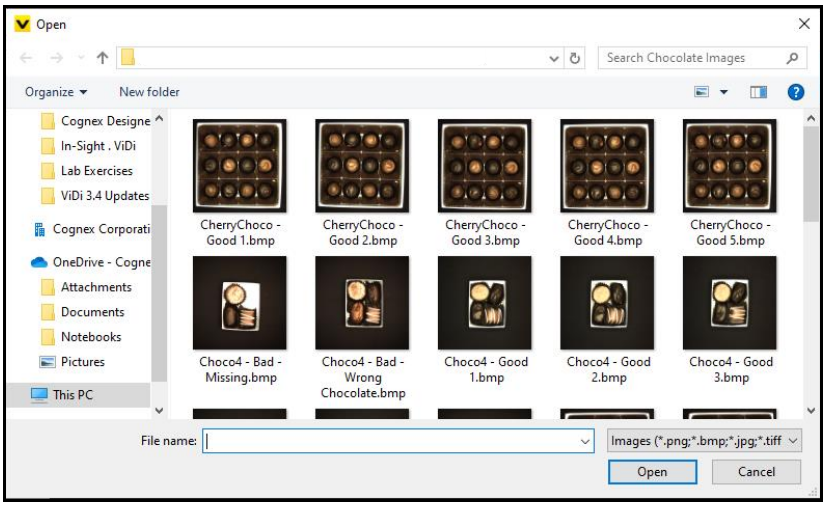
4. Enter *Chocolate_Verification* in the Tool name field and click the **Blue Locate** tool to select.
   The **Blue Locate** tool is added to the Stream.

5. Click the **Add** button in the View Panel Stream.

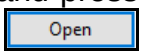


6. Browse to the folder containing the Chocolate images.

   NOTE: *The images are in the ViDi Student Folder → Lab 4-2 Chocolates.*

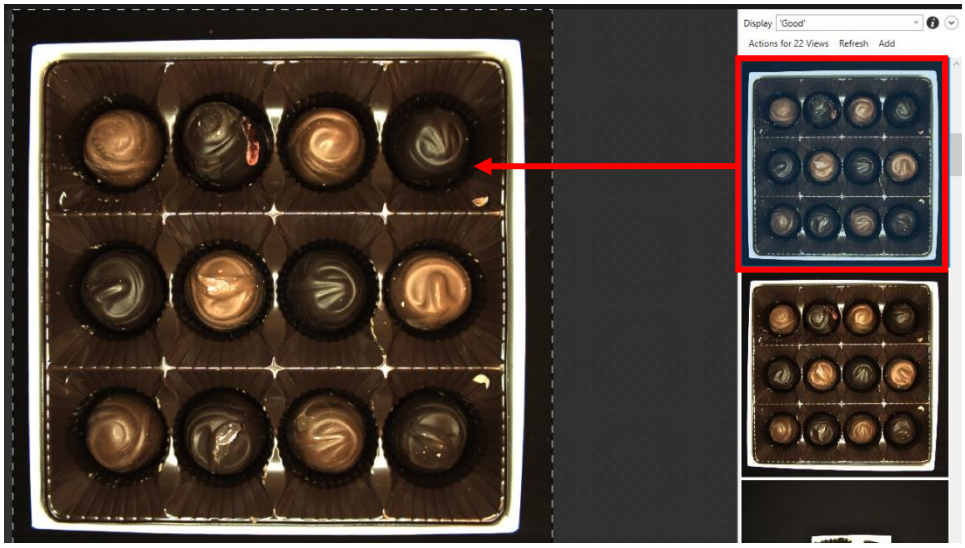


7. All of the images will be added to the Workspace, select the first image and press **<Ctrl + A>**, this will automatically select all of the images. Click the **Open** button.

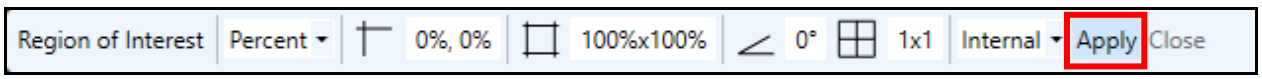


8. Once the images have been added, they will appear in the View Panel.
9. Click an image to select.

   The **Image** will display in the Image Panel.



10. Define the Region of Interest (ROI). In this example we will be using the entire image, so click **Apply**.

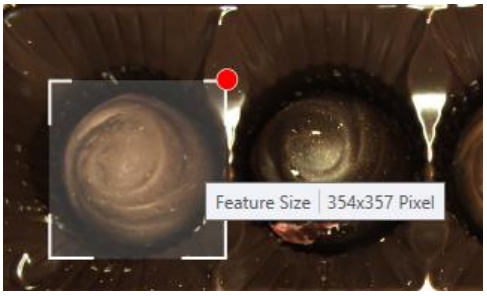


11. The **ROI** is defined.

12. Open the **Tool Parameters**. Since the original image was color, we must change the **Color** Channel from 1 to **3** because we want to use all of the data contained in the original image.

13. Move the **Feature Size** box over the first chocolate. Drag the red circle to the desired size (approximately the same size as the chocolate) and move it to be centered over the first chocolate.

    NOTE: *The Feature Size box is found in the lower left hand side of the Workspace.*

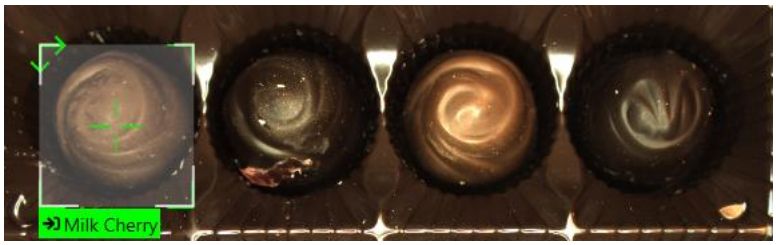


14. Click on the image outside of your feature size box.

    NOTE: *If you zoom in and out the Feature Size box will return to the corner.*

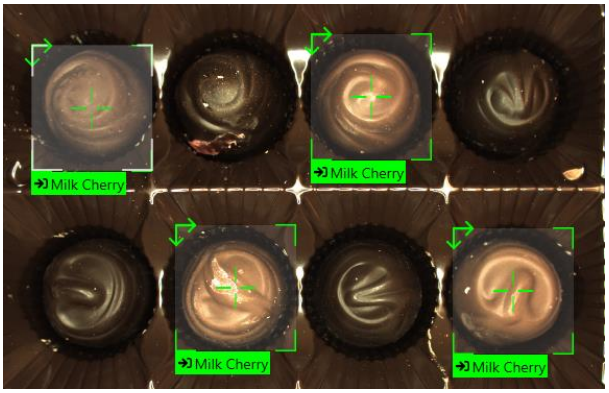
    A green rectangle the same size as your feature size box displays. The box is labeled 0 by default.

15. Move the box over the first chocolate and click the **0** to open the text box. Label this chocolate *Milk Cherry*.

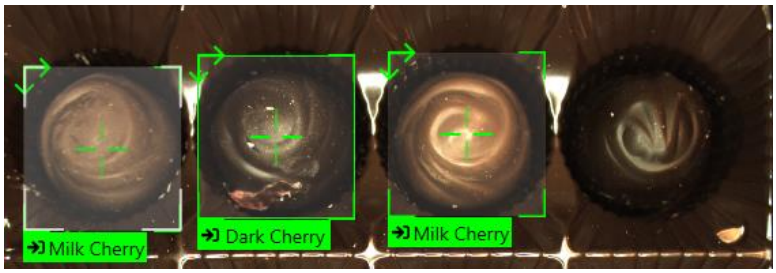


16. Click on the remaining *Milk Cherry* chocolates in the box to label.

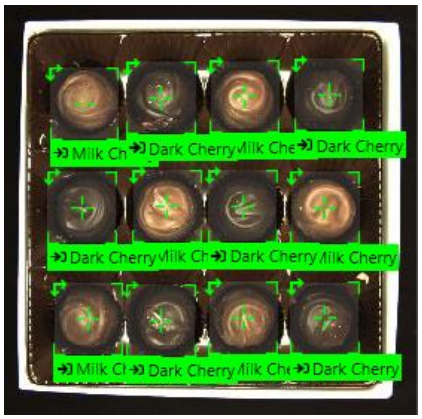
    NOTE: *As you click on each chocolate the labeled box will display – make sure that each one is centered over the corresponding chocolate.*



17. Click on the Dark Chocolate and change the label to *Dark Cherry.*



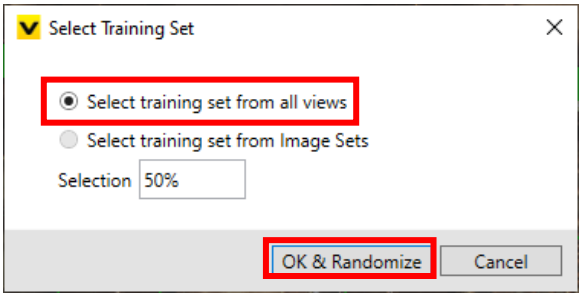
18. Click on the remaining *Dark Cherry* chocolates in the box to label.
NOTE: *Since this assortment is consistent there is no need to label more than one assortment.*



19. Move onto the next chocolate assortment in your image database and label the chocolates.
NOTE: *Use the names of the chocolates found in the Resources when labeling the chocolates in the assortments.*
20. Label at least three images with the names of each chocolate.

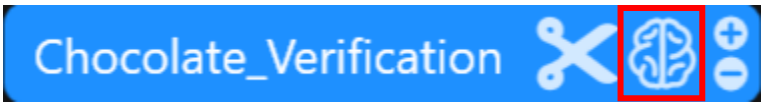
21. Repeat steps 20 and 21 to label the rest of the chocolate assortments.
22. Select the Training Set by clicking on the Training Set parameter.

The **Select Training Set** dialog displays.



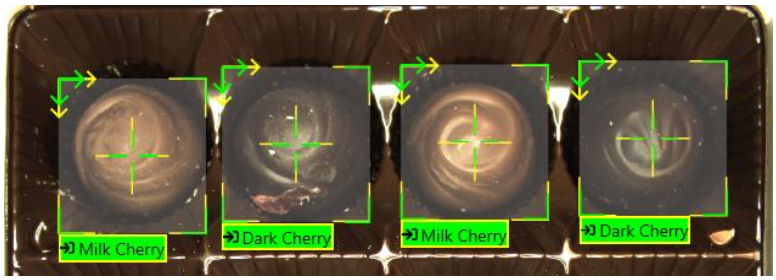
23. **Select training set from all views** and click the **OK & Randomize**  button.

24. Click the **Train** button to train.



The **Analysis** begins.

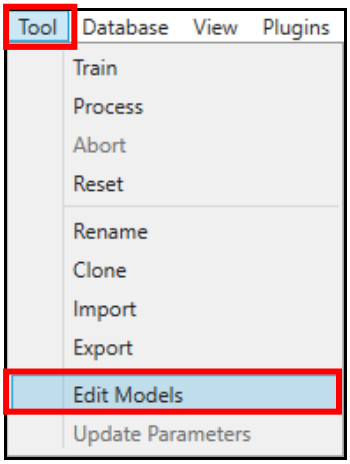
25. Once the training is complete scroll through your images to view the results.
26. Confirm that all chocolates are correctly marked and label any chocolates that were not marked.
Notice on the images in addition to the green labels there are also yellow markings recognizing the different chocolates.



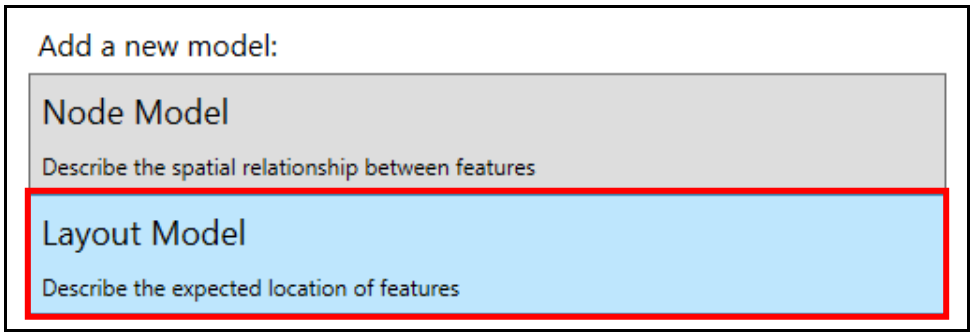
Build the Models for the Four Types of Chocolates

Follow the steps below to complete the lab exercise:

1. Select the first chocolate assortment, from the **Tool** menu select **Edit Models**.

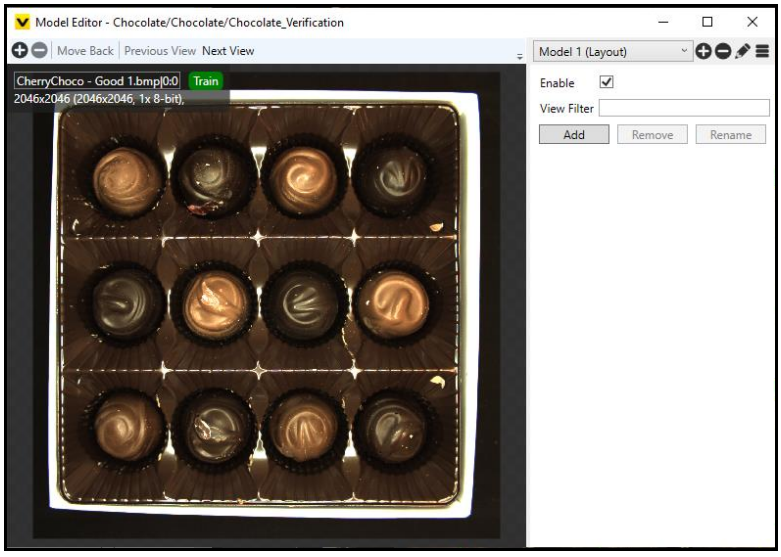


The **Model Editor** displays.



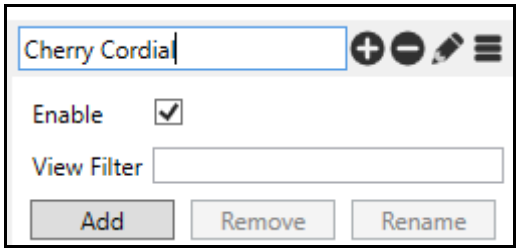
2. Select **Layout Model**.

The selected assortment displays in the **Model Editor**.

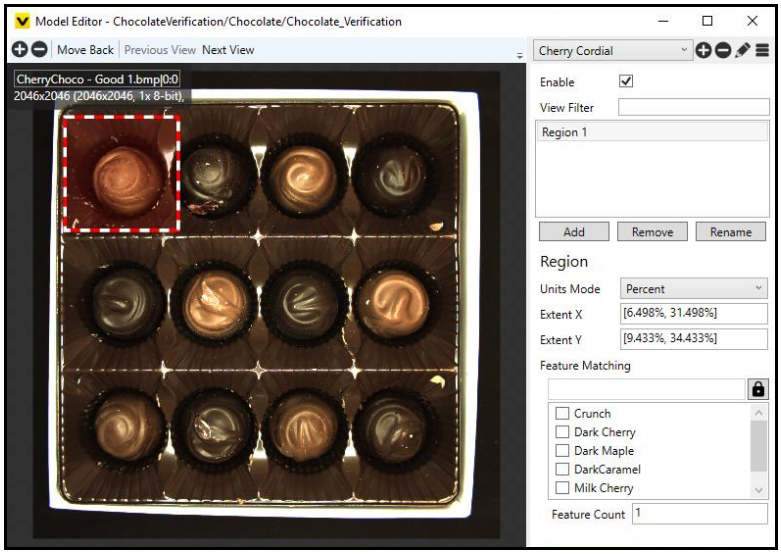


3. Click the **pencil**  icon to rename the model.

NOTE: *Use the names of the assortments found in the Resources when renaming the four chocolate models.*

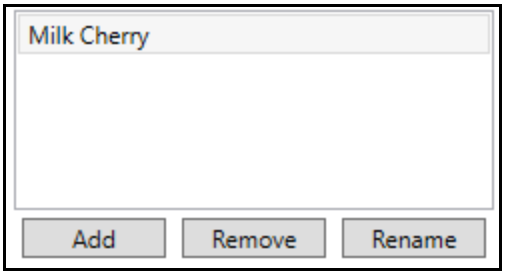


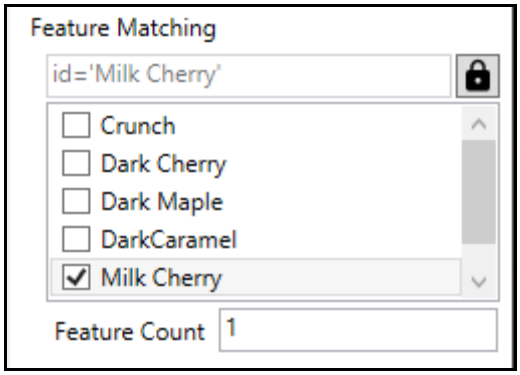
4. Click the **Add** Add button.
5. Move the ROI to outline the first chocolate.



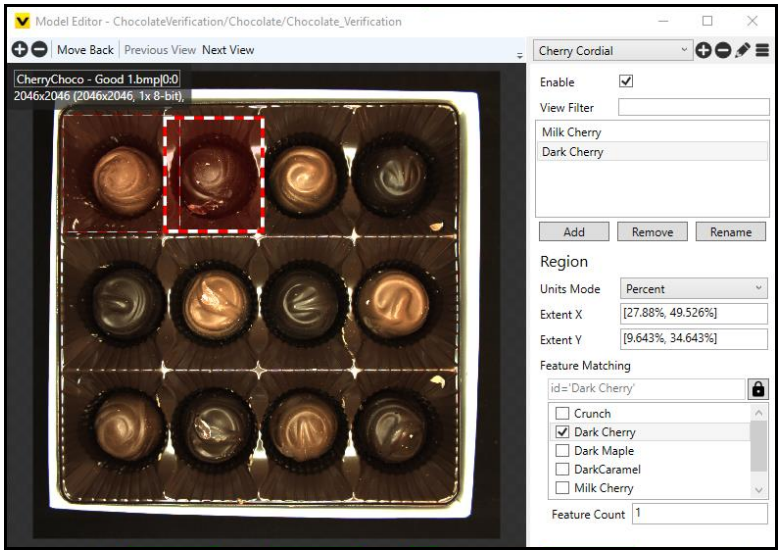
7. Click the **Rename** Rename button to update the name of Region 1 to *Milk Cherry* and check the Milk Cherry box in the Feature Matching box.

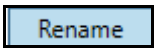
   NOTE: *There is one feature in each ROI, so the Feature Count is 1.*

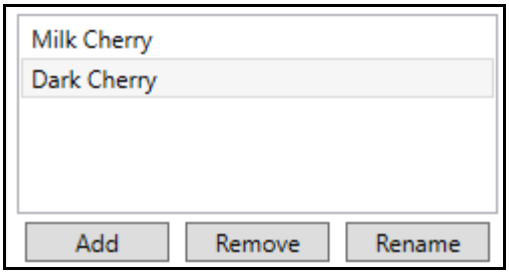


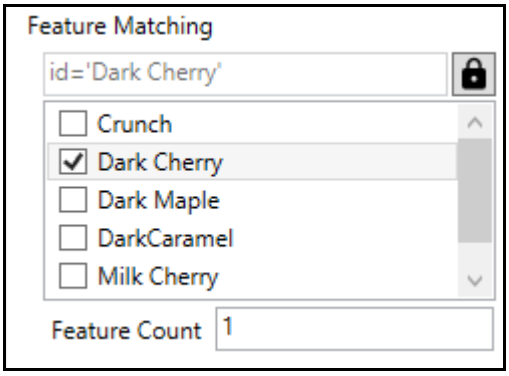


8. Click the **Add**  button and move the ROI to outline the second chocolate.

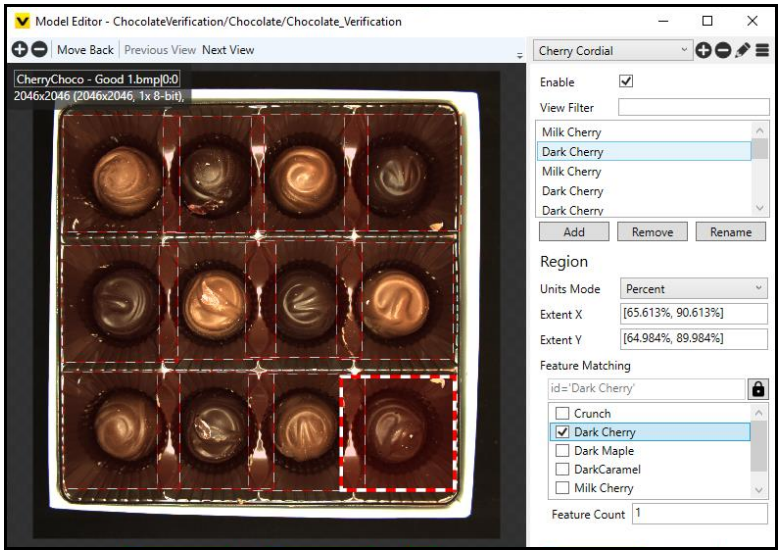


9. Click the **Rename**  button to update the name of Region 1 to *Dark Cherry* and check the *Dark Cherry* box in the Feature Matching section.



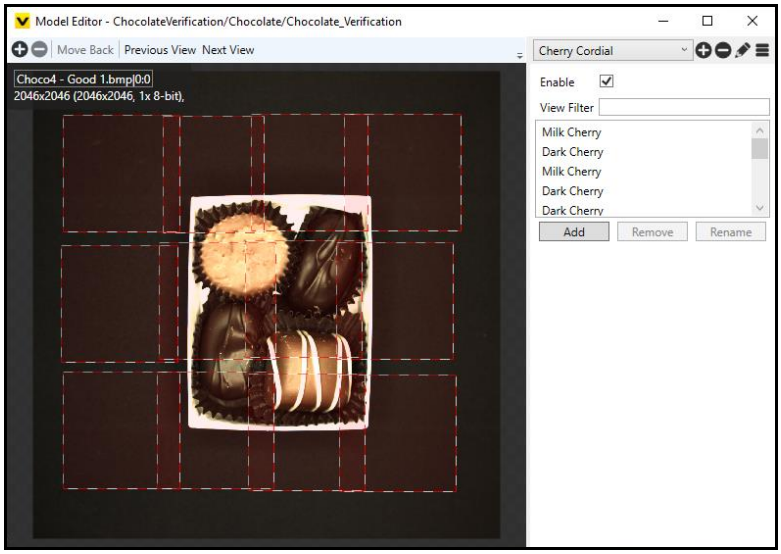


10. Repeat for each of the remaining chocolates in the assortment.

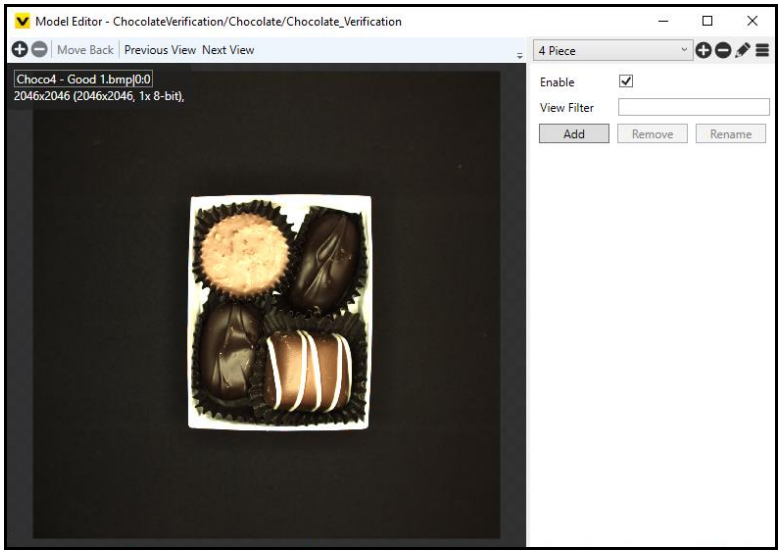


11. Close the Model Editor and move to the next chocolate assortment in the Image Library.
12. From the **Tool** menu select **Edit Models**.
13. Select **Layout Model**.

    The selected assortment displays in the Model Editor.

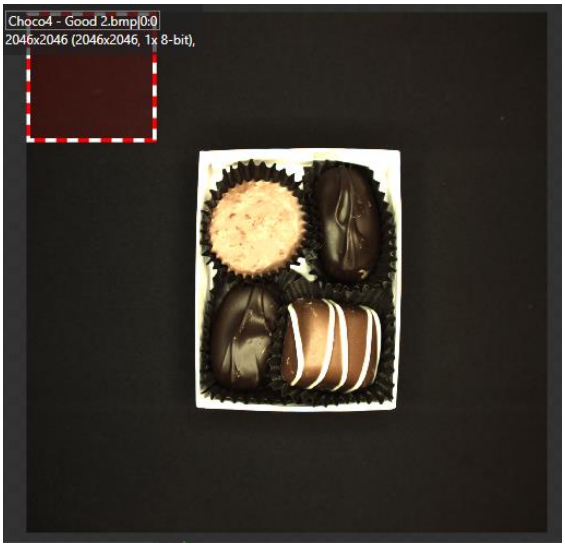


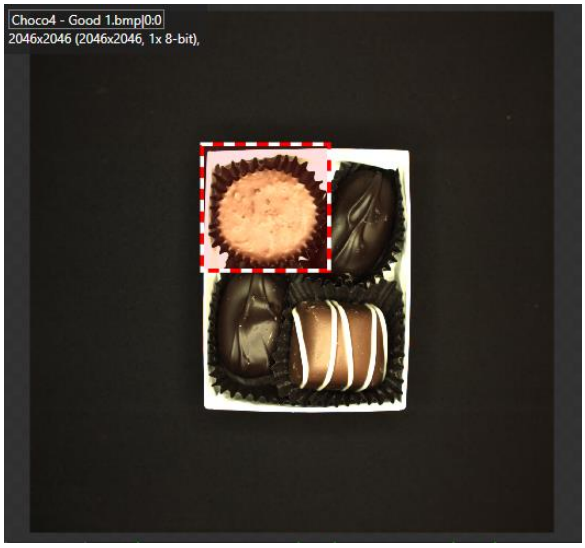
14. Click the ⊕ button and select **Layout** from the flyout.
15. Click the **pencil** icon to rename the model to *4 Piece*.

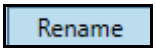
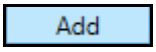


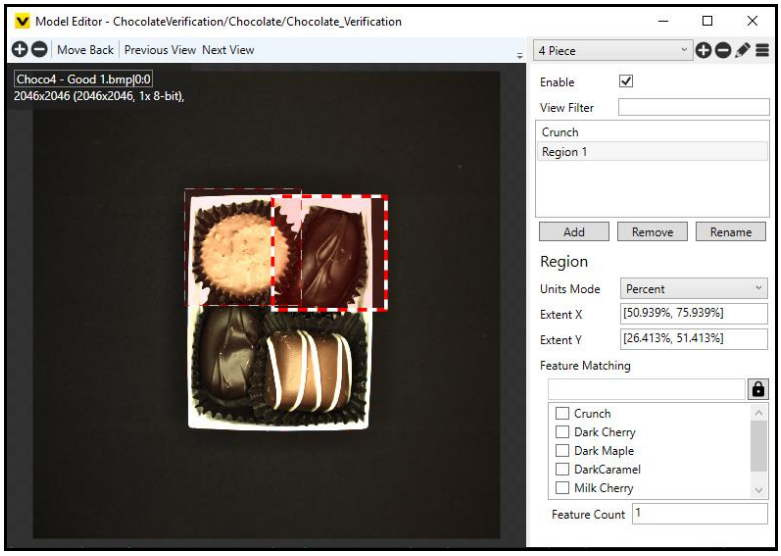
16. Click the **Add** Add button.

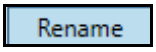
17. Move the ROI to outline the first chocolate.

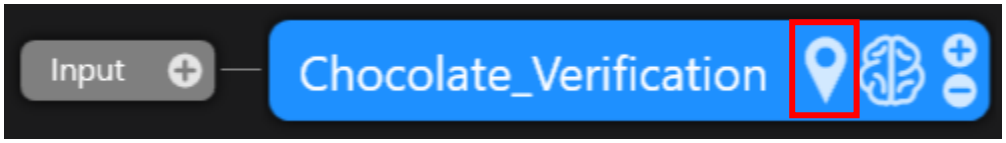




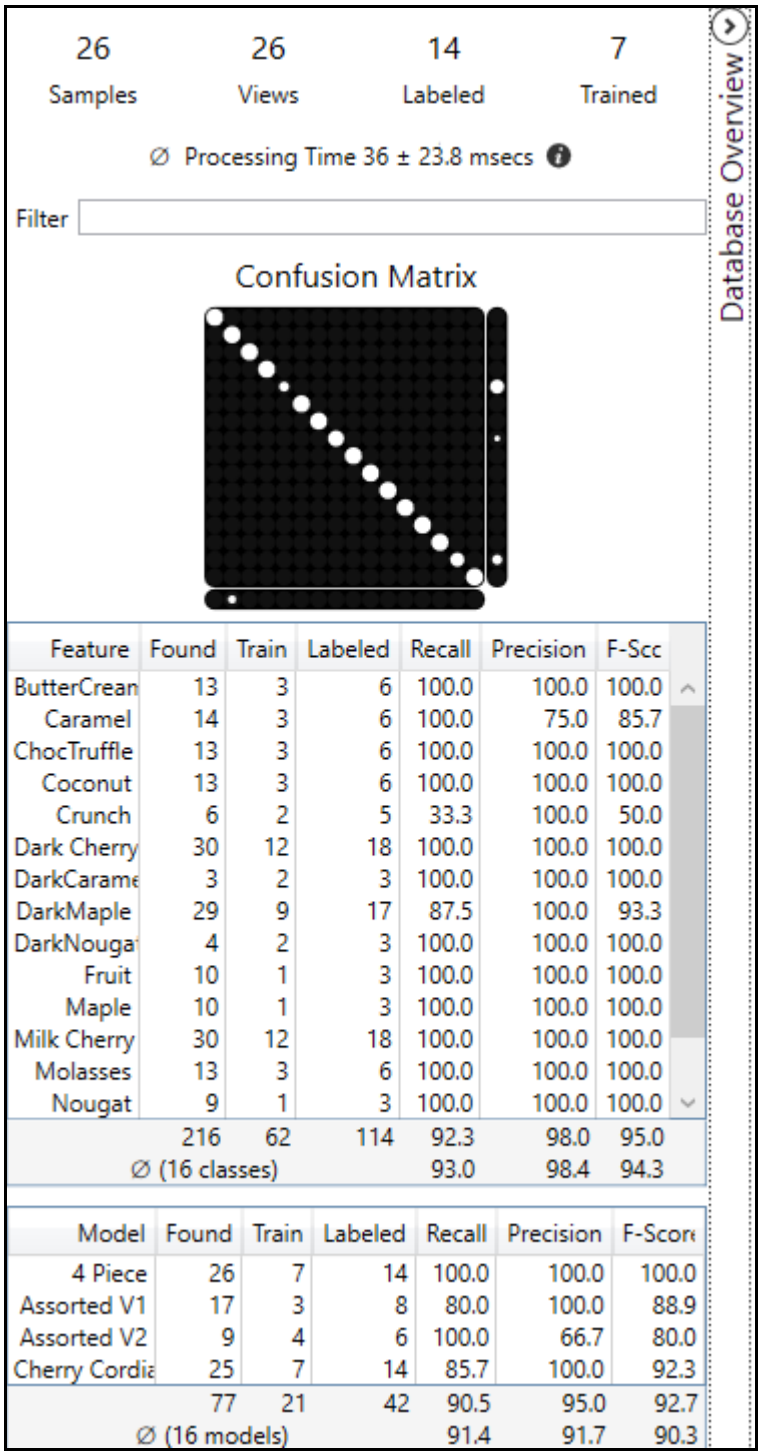
18. Click the **Rename**  button to update the name of Region 1 to *Crunch* and check the *Crunch* box in the Feature Matching section.
19. Click the **Add**  button and move the ROI to outline the second chocolate.



20. Click the Rename  button to update the name of Region 1 to *DarkMaple* and check the *DarkMaple* box in the Feature Matching section.
21. Repeat the steps to add the last two chocolates to the Model.
22. Repeat steps 12 – 21 for the remaining two chocolate models.
23. Process the models.



23. Open the **Database Overview** tab and review the results.



Feature	Found	Train	Labeled	Recall	Precision	F-Scc
ButterCrean	13	3	6	100.0	100.0	100.0
Caramel	14	3	6	100.0	75.0	85.7
ChocTruffle	13	3	6	100.0	100.0	100.0
Coconut	13	3	6	100.0	100.0	100.0
Crunch	6	2	5	33.3	100.0	50.0
Dark Cherry	30	12	18	100.0	100.0	100.0
DarkCarame	3	2	3	100.0	100.0	100.0
DarkMaple	29	9	17	87.5	100.0	93.3
DarkNougat	4	2	3	100.0	100.0	100.0
Fruit	10	1	3	100.0	100.0	100.0
Maple	10	1	3	100.0	100.0	100.0
Milk Cherry	30	12	18	100.0	100.0	100.0
Molasses	13	3	6	100.0	100.0	100.0
Nougat	9	1	3	100.0	100.0	100.0
	216	62	114	92.3	98.0	95.0
Ø (16 classes)				93.0	98.4	94.3

Model	Found	Train	Labeled	Recall	Precision	F-Score
4 Piece	26	7	14	100.0	100.0	100.0
Assorted V1	17	3	8	80.0	100.0	88.9
Assorted V2	9	4	6	100.0	66.7	80.0
Cherry Cordia	25	7	14	85.7	100.0	92.3
	77	21	42	90.5	95.0	92.7
Ø (16 models)				91.4	91.7	90.3

2. Inspection of Medical Screws – Part 1

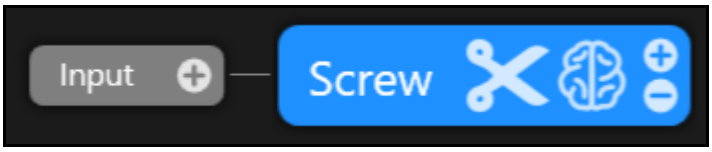
Screws (or any cylindrical object) need to be rotated during inspection. Looking not only at one specific line along the rotation axis, but at the full object over several images allows you to see the surface and defects at different angles to the camera and illuminations. For instance, some defects will show best when they are on the center line, while others will show better when they are slightly off to the sides.

Follow the steps below to complete the lab exercise:

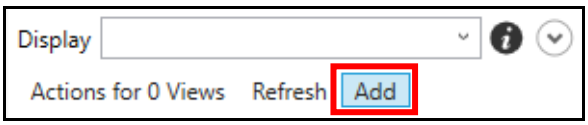
1. Click the **Add Stream** button, to add a Stream to the Workspace.
2. Click the **Rename Stream** button, rename the Stream *Screw Example* and press the **<Enter>** key.

3. Click the **Input** button to view the available ViDi tools.

4. Enter *Screw* in the Tool name field and click the **Blue Locate** tool to select.
   The **Blue Locate** tool is added to the Stream.

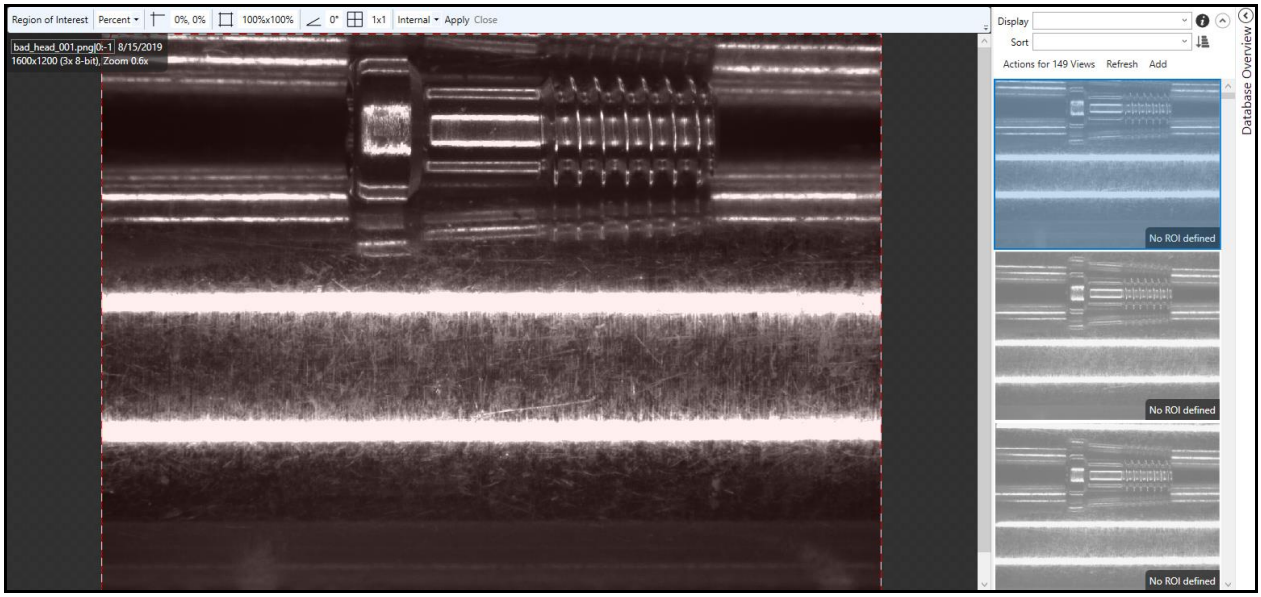


5. Click the **Add** button in the View Panel Stream.

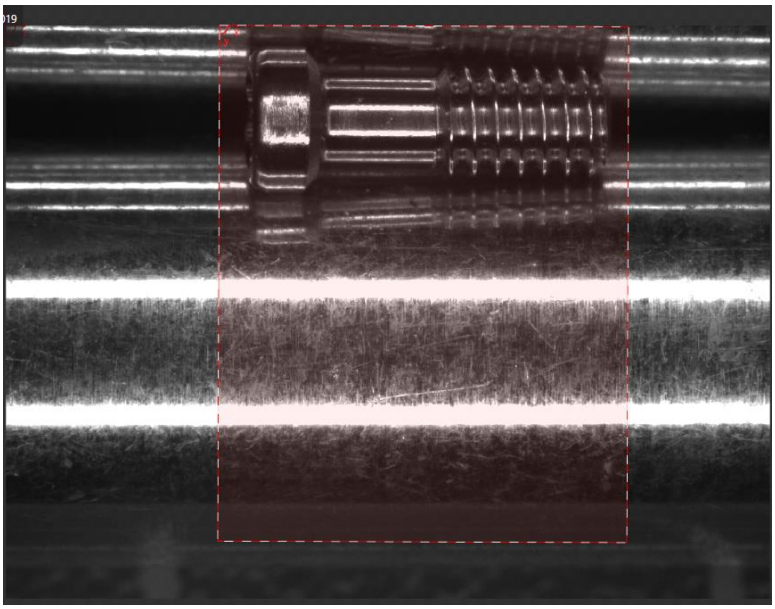


6. Browse to the folder containing the Medical Screw images.
   NOTE: *The images are in the ViDi Student Folder → Lab 4-1 & 7-1 Medical Screw.*
7. All of the images will be added to the Workspace, select the first image and press **<Ctrl + A>**, this will automatically select all of the images. Click the **Open** button.
8. Once the images have been added, they will appear in the View Panel.
9. Click an image to select.

The **Image** will display in the Image Panel.



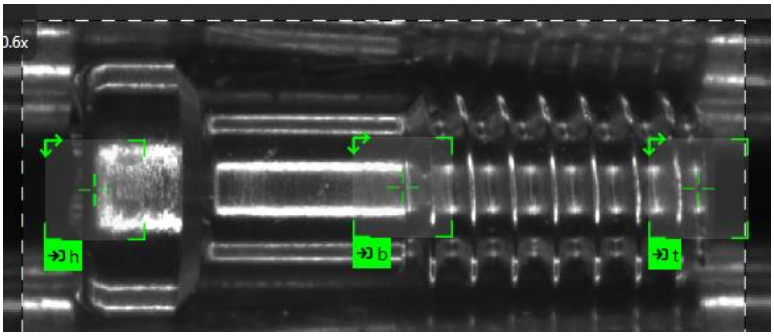
10. Define the Region of Interest (ROI) as shown below and click **Apply**.
    NOTE: *There is a significant amount of area to both the left and right of the image, but the screw moves up and down in the region.*



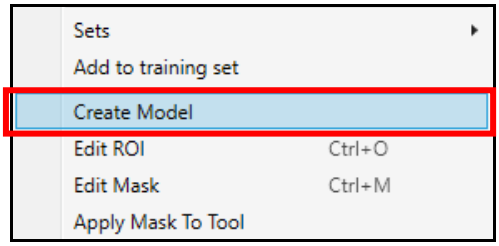
11. The **ROI** is defined.
12. Open the **Tool Parameters** and set the Feature Size to **120**.

13. To find an object we must first define the parts it's made of. In the case of the screw we will break it up into 3 separate segments, a head (labeled **h**), a body (labeled **b**), and a tail (labeled **t**).

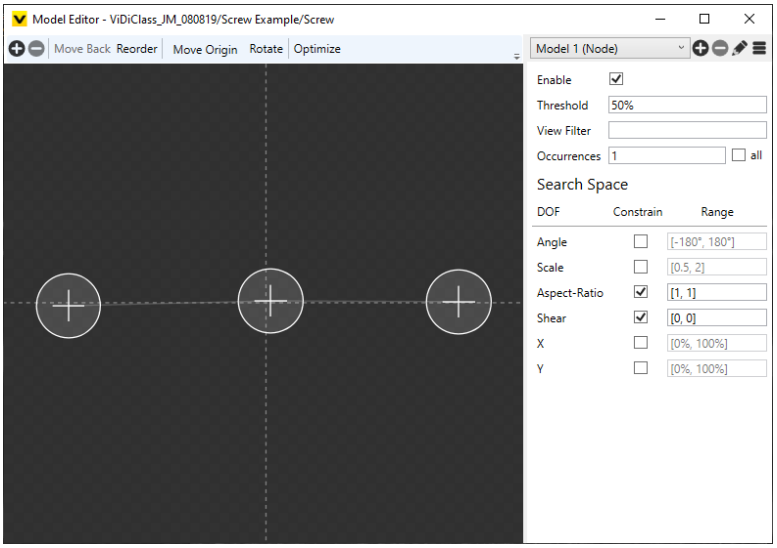
    The final product should look like the image below:



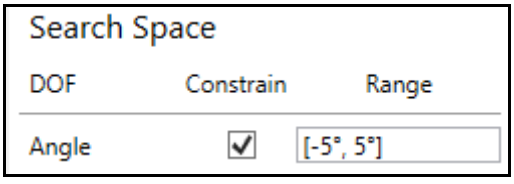
14. Right click on the image and select **Create Model** from the list.



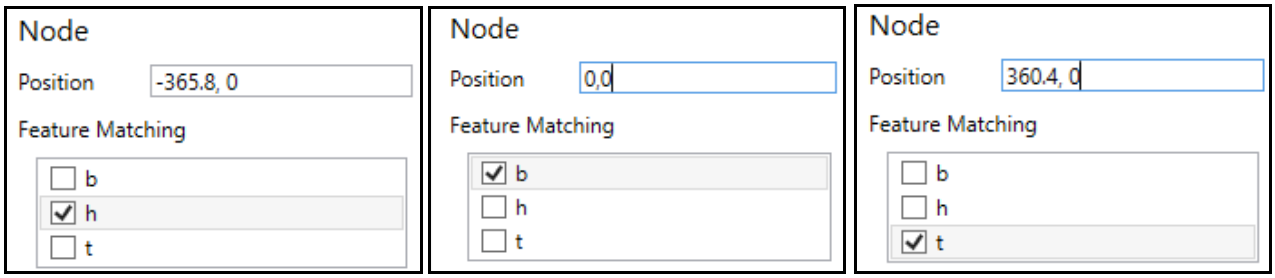
    The **Model Editor** displays.



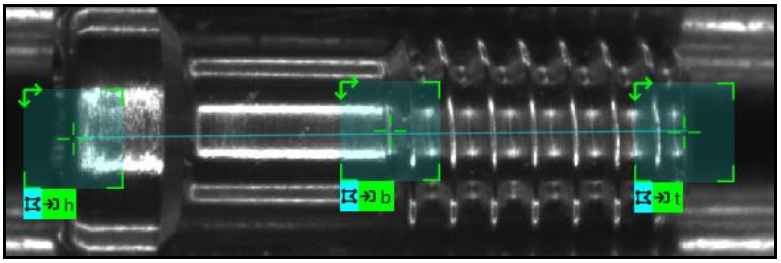
15. Click the **pencil**  icon to rename the model *Screw*.
16. Change the **Angle** to range from -5° to 5°.



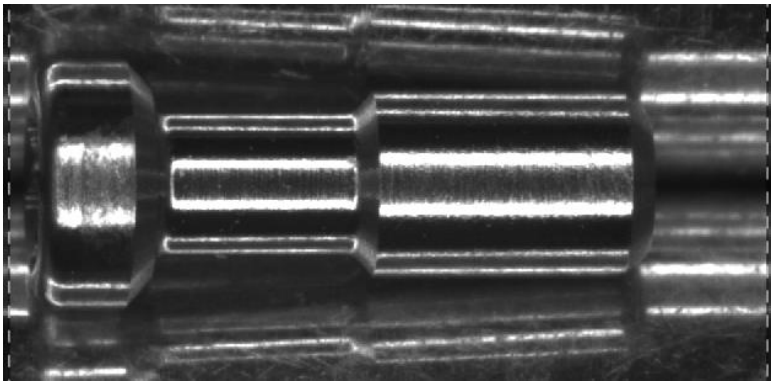
17. Position the Nodes in a straight line by changing the position of the Y axis to 0.
NOTE: *Remember the node b should be at the center (0,0) so change both of its coordinates to 0.*



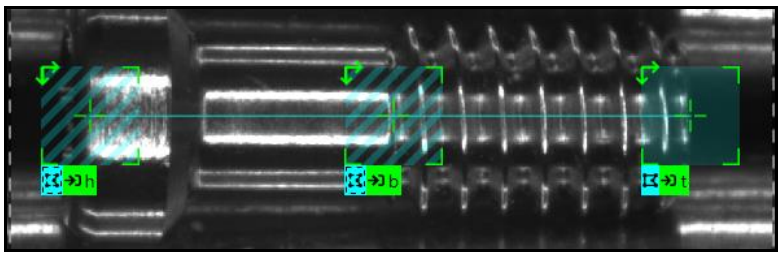
18. Close the Model Editor.
The screw positions will be labeled



19. Label approximately 20 more screws. Be sure to include some of the screws that are missing the threads. Example below – the name of the screw is 'no_thread'

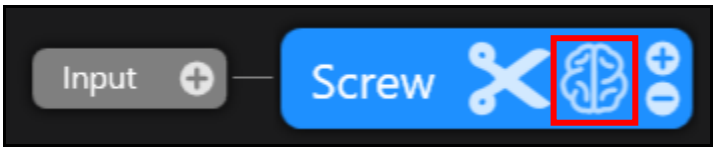


NOTE: *ViDi will remember the last node that was labeled – in this case the tail and will position the nodes as such on the image.*

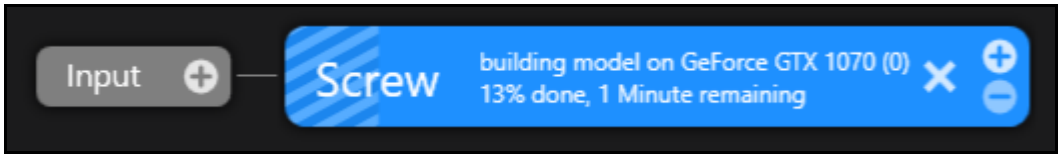


20. Type an **h** in the first node, this will move the model to the correct position on the screw.
21. Click on each node – this will remove the diagonal lines and accept the view.

22. Click the **Train** button to train and select your Training Set.



The **Analysis** begins.



23. Once the training is complete scroll through your images to view the results.
24. Confirm that there is only one Screw model in each view – to do this use the following function to search through all the views: **not count(match[Screw])=1**. This should return 0 results.
25. Confirm that all screws are correctly marked and label any screws that were not marked.
26. Add more samples to the training set. Look for views where the nodes are displayed as yellow (markings), and green (labels) these are the ones that you have already used – to turn markings into labels, right click on the image and select **Accept View**.
    Note: Add 20 to 30 more samples favoring those that had trouble detecting originally.
27. Repeat steps 22 – 26 until 100% of samples are detected correctly.

Lab Exercise 5.1 – ViDi Blue – Read Tool

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi Blue – Read tool to solve their OCR application

The Participant will utilize the following ViDi tool to successfully complete this exercise:

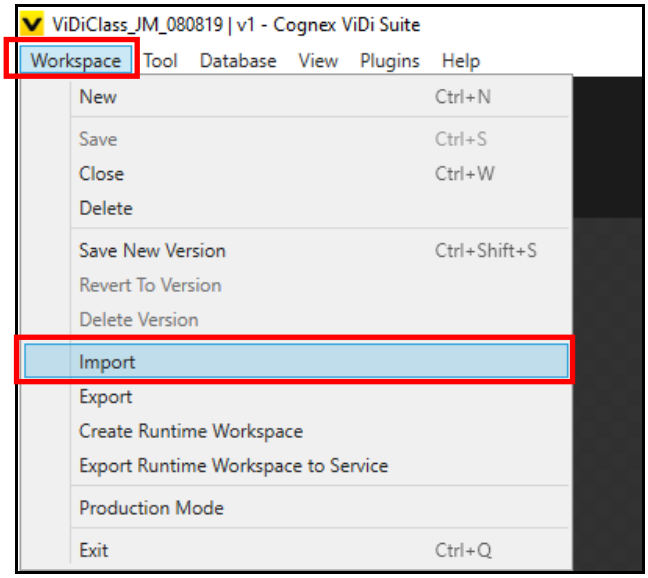
- Blue – Read Tool

OCR

Optical Character Recognition (OCR) is a common application in the industry. It can be used for validating the data read from a code matches the human readable portion, confirming the correct text is present, and many other applications.
The codes in this application come from the end of lumber pieces, some of which are damaged, or hard to read, but all of them are in the same format. For this example, we will create a model that accurately reads the codes and is able to identify the ones that have missing characters due to damage.

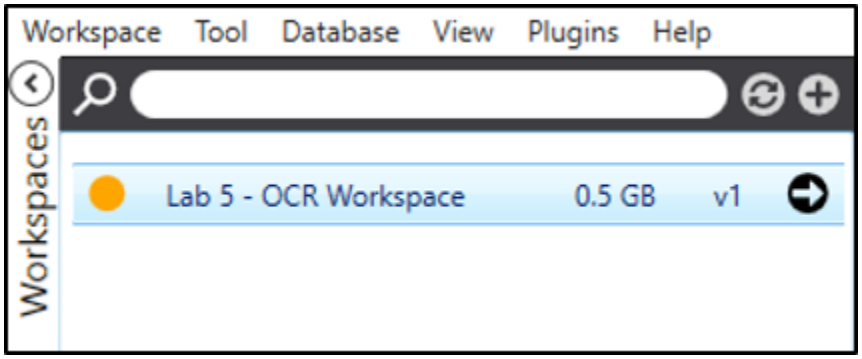
Follow the steps below to complete the lab exercise:

1. Import the OCR1 Workspace.
   To do this click **Workspace** in the top left of the ViDi window and select **Import**.

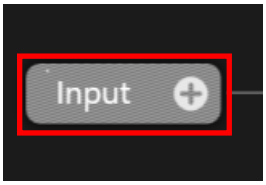


2. Browse to the folder containing the OCR1 workspace, select **OCR1** and click **Open**.
   This should be inside of the ViDi Student folder under Labs folder, in the OCR folder.

3. Navigate to the Workspaces tab and click the arrow to open the OCR Workspace that you just imported.
   NOTE: *This workspace has all the images imported already.*

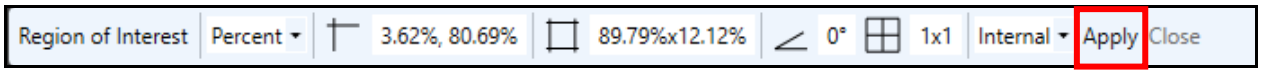


4. Add a **Blue – Read Tool** to the Stream by first clicking on the Inputs button, adding the desired name, then clicking the Blue – Read Tool button.
   The recommended name for this tool is **OCR**.



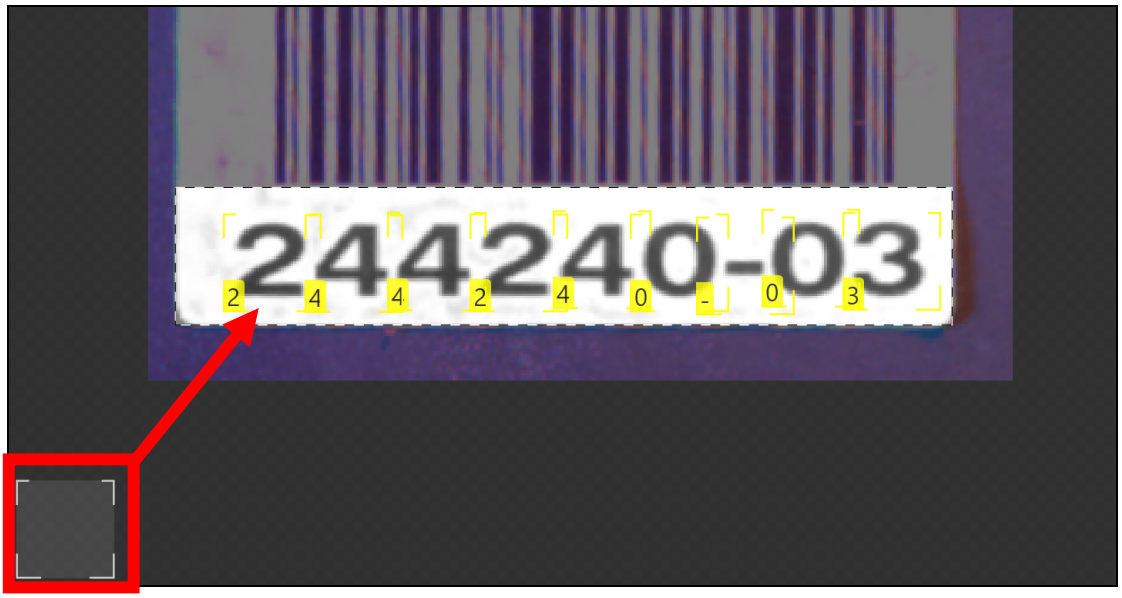
5. Click the first image to select.
6. Edit the ROI so that it only includes the text at the bottom of the code, as shown below.

7. Click **Apply** on the Region of Interest toolbar.

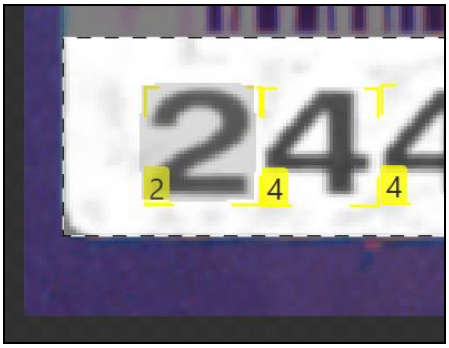


NOTE: *ViDi will automatically process the images and use its pre-trained font library to detect the characters.*

8. Adjust the **Feature Size**.
   To do so drag the feature size from the bottom left of the image panel onto the image.

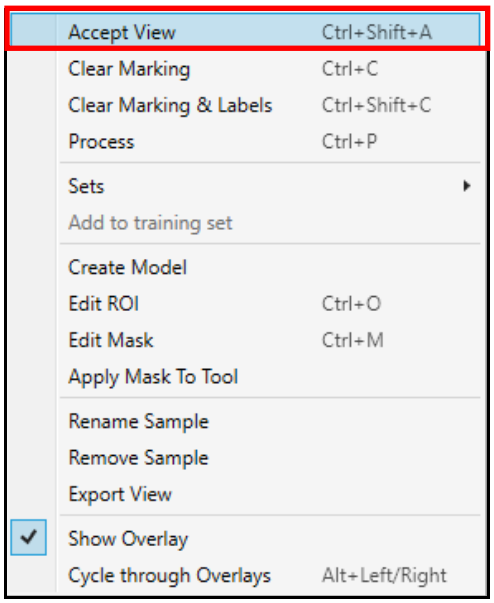


9. Adjust the Feature Size to be approximately the same size as the characters.



10. Process the Views by clicking on the Book button at the top of the screen.

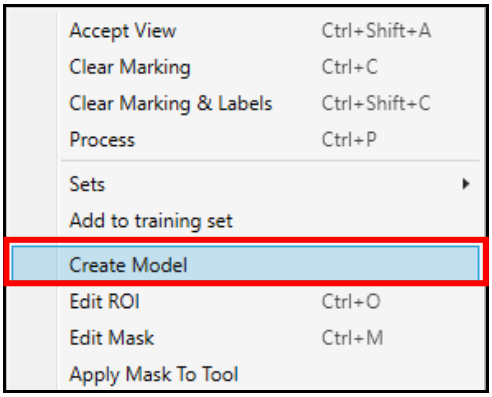
11. Right click on the image and select **Accept View**.



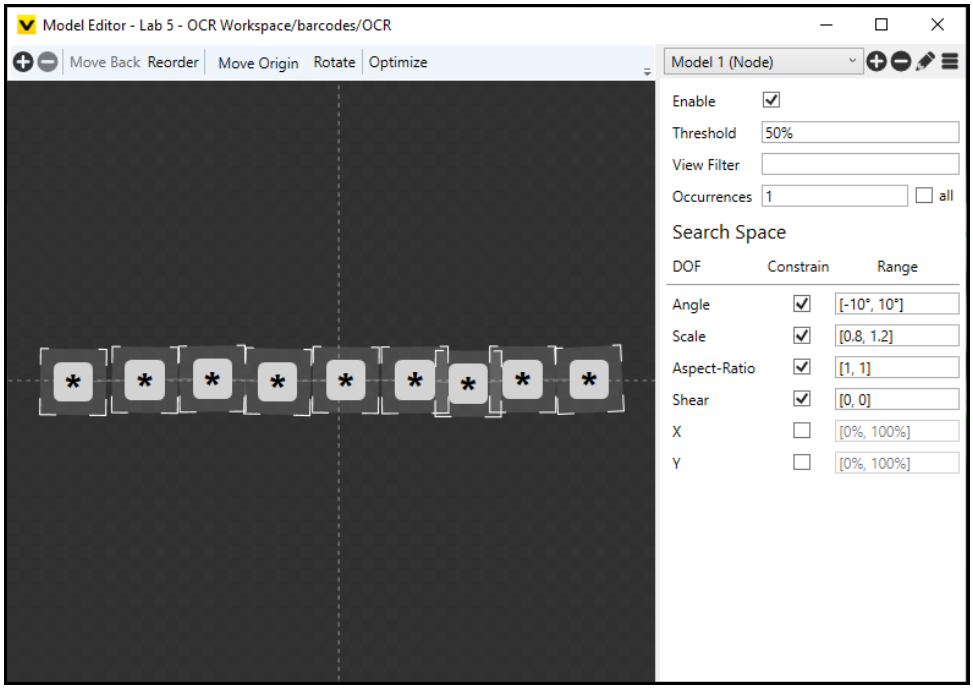
Notice that the color changes from yellow to green on the character outlines.

12. Select all Characters to create a Model.
Press the **<Ctrl>** key and click on each character to highlight.

13. Right click on the image and select **Create Model.**

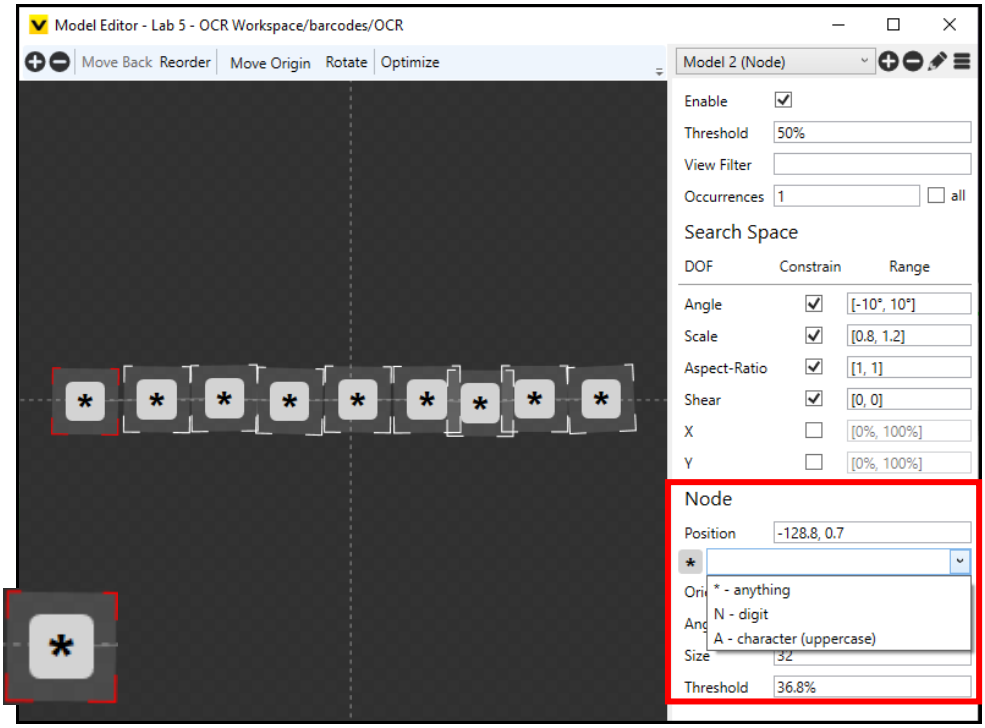


The Model Editor displays. There are 9 total nodes in the model, each with an asterisk inside of it.

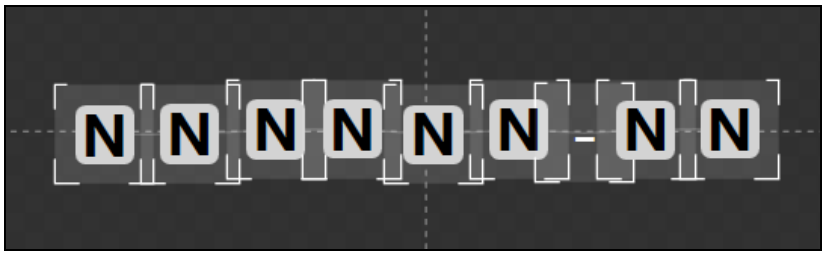


14. Click the **pencil** button and rename the Model *Code*.
15. Click the first node to highlight.

    The first node is outlined in red and the Node parameters display.



16. Select **N** from the Feature ID * drop down list and press the **<Enter>** key.



17. Repeat for each node – in this model there should be 6 numbers dash 2 numbers.

    NOTE: *Since node 7 is always a dash, you can manually enter a dash (-) in the Feature ID box*.

    The text should now be included in a green border with the characters read displayed below.

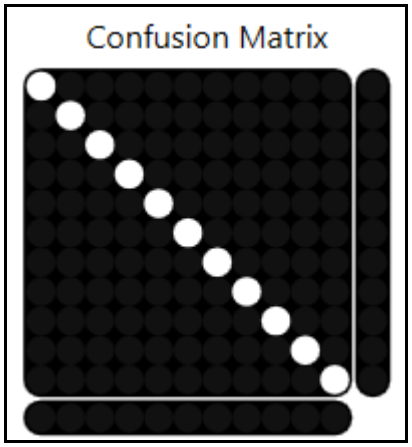
18. Process the Views utilizing the newly created model by clicking on the Book button at the top of the screen.

19. Accept more views, correcting any mistaken detections made by the tool.

    NOTE: *Make sure to label at least 5 of each character. The number of each character labeled can be viewed in the Database Overview panel on the right of the screen.*

20. Once enough of each character has been labeled hit the **Train** button.

21. Review the results and make the appropriate changes to the tool.
In this example close to 100% character detection is possible, even on the damaged or bent codes.



Feature	Found	Train	Labeled	Recall	Precision	F-Score
-	390	26	46	100.0	100.0	100.0
0	583	38	70	100.0	100.0	100.0
1	352	21	33	100.0	100.0	100.0
2	357	25	44	100.0	100.0	100.0
3	665	46	79	100.0	100.0	100.0
4	365	25	49	100.0	100.0	100.0
5	115	8	13	100.0	100.0	100.0
6	133	8	17	100.0	100.0	100.0
7	75	4	7	100.0	100.0	100.0
8	228	17	28	100.0	100.0	100.0
9	226	14	26	100.0	100.0	100.0
	3489	232	412	100.0	100.0	100.0
	Ø (11 classes)			100.0	100.0	100.0

Model	Found	Train	Labeled	Recall	Precision	F-Score
Code	390	26	46	100.0	100.0	100.0

Lab Exercise 6.1 – ViDi Green – Classify Tool

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi Green – Classify tool to solve their application

The Participant will utilize the following ViDi tools to successfully complete this exercise:

- Green – Classify Tool

Classify Objects
Green – Classify Tool

Distinguishing different objects under various angles and lighting conditions is a common, yet difficult task. In this lab exercise we will use the Green – Classify tool's simple and intuitive interface to work on an image classification database that contains 150 different objects. There are 24 images from different angles for each object.

The goal of this lab exercise is to use a subset of the images from each class to train and test the resulting performance on the remaining untrained images.

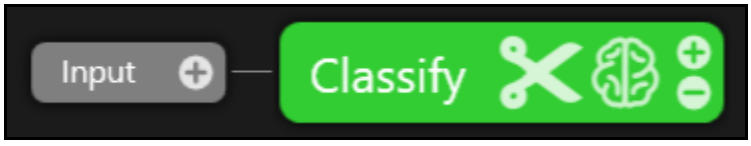
Follow the steps below to complete the lab exercise:

1. Click the **Add Stream** button, to add a Stream to the Workspace.
2. Click the **Rename Stream** button, rename the Stream *Classify Example* and press the **<Enter>** key.

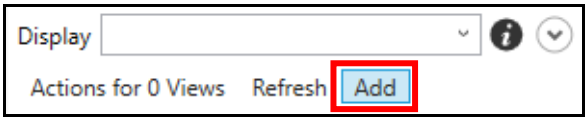
3. Click the **Input** button to view the available ViDi tools.

4. Enter *Classify* in the Tool name field and click the **Green Classify** tool to select.

   The **Green Classify** tool is added to the Stream.

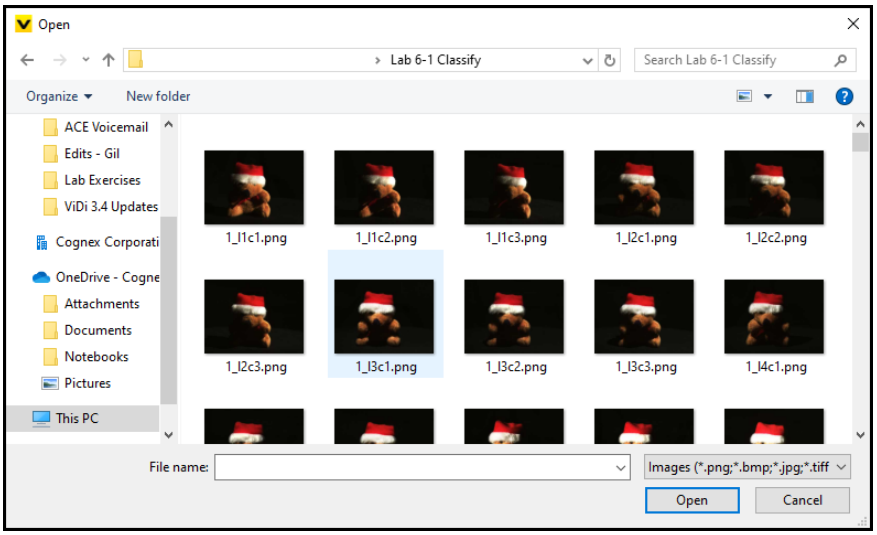


5. Click the **Add** button in the View Panel Stream.



6. Navigate to the folder containing the Classification images.

   NOTE: *The images are in the ViDi Student Folder → Lab 6.1 Classify.*



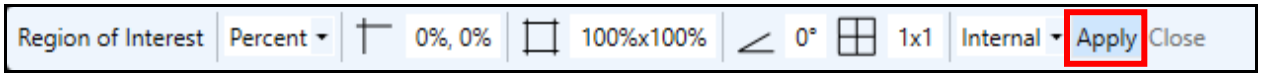
7. All images will be added to the Workspace, select the first image and press **<Ctrl+A>,** this will automatically select all the images. Click the **Open** button.

   Once the images have been added, they will appear in the View Panel.

8. Click an image to select.

   The image will display in the Image Panel.

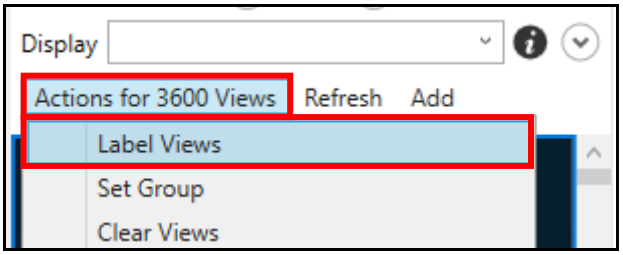
9. Define the Region of Interest (ROI). In this example we will be using the entire image, so click **Apply**.



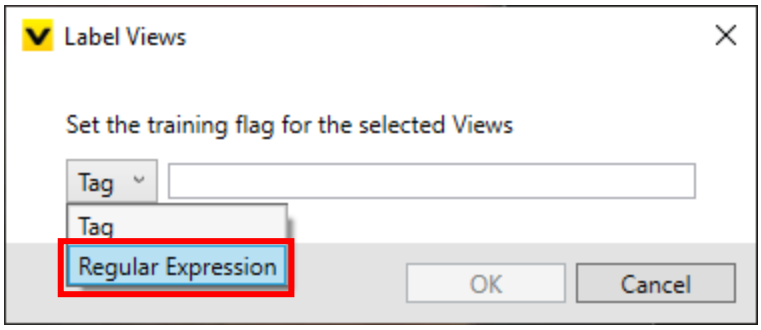
   The **ROI** is defined.

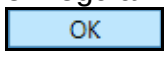
10. Open the **Tool Parameters**.
    Feature Size = 80 pixels so that the part can be properly detected.
    Color = 3 since the image is in color.
11. Next, label the images.

    NOTE: *There are two options to label the images.*
12. Select **Actions for 3600 Views** and select **Label Views**.

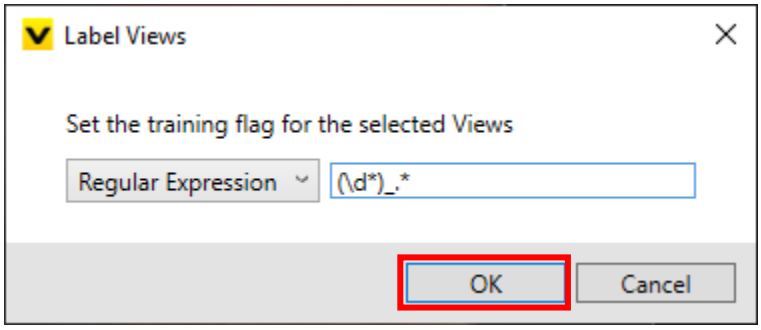


    The Label Views dialog displays.
13. Select **Regular Expression** from the drop-down list.



    NOTE: *The images follow the naming convention **n**_|**x**c**y**.png where **n** is the object number, ranging from 1 to 150 **x** and **y** are numbers describing light incidence. We want to tag them according to the object number.*
14. Type the Regular Expression in the field. For this example enter (\d*)_.* and click the **OK** 

button.

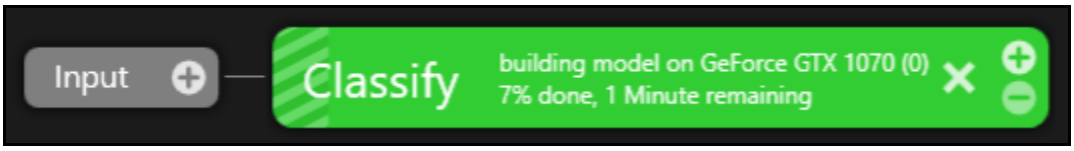


    NOTE: *The Regular Expression follows:*
    ***Literal** – literal characters to be matched*
    ***^** - matches the beginning of the string*
    *(**pattern**) – capturing group: pattern is used as a label for the image*
    ***.** – matches any single character*
    ***** - match 0 or more of the preceding token*

The images are now numbered in the lower left hand corner.

15. Click the **Train** button to train.

    The analysis begins.



16. Review the results in the Statistics Panel.

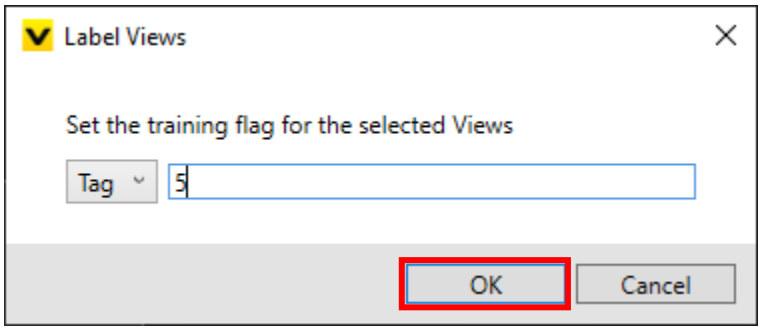
    NOTE: *You are looking for items that have a low recall or a high confusion or precision rate.*

17. Object 5 has lower statistics, double click on the line to open the views whose best tag is 5.
18. Object 5 results display – scroll through and notice that object 5 becomes confused with object 50.

19. Review the results for object 50 by navigating to line 50 and double clicking the line.

    Notice that object 5 and object 50 are both walnuts and are indeed the same object.

20. Select **Actions for 24 views** and use the **Tag** parameter to retag all of object 50 as object 5 and click the **OK** button.



There may be issues with other numbers – for problematic parts, right click on the image and add them to your training set.

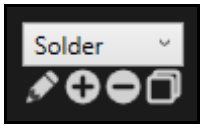
21. Repeat steps 18 – 21 until all parts receive a 100% F-Score.

Solder Identification – Part 1
Green – Classify Tool

This lab exercise will create a processing tool-chain with segmentation based on the classification results. We will experiment with selection filters and sorting criteria using advanced techniques.

Follow the steps below to complete the lab exercise:

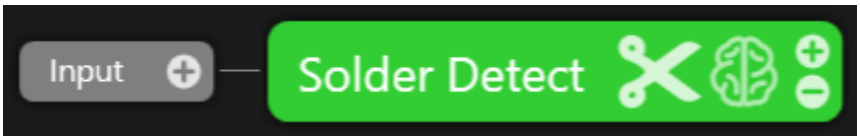
1. Click the **Add Stream** button, to add a Stream to the Workspace.
2. Click the **Rename Stream** button, rename the Stream *Classify Example* and press the **<Enter>** key.



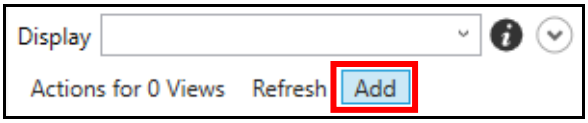
3. Click the **Input** button to view the available ViDi tools.

4. Enter *Solder Detect* in the Tool name field and click the **Green Classify** tool to select.

The **Green Classify** tool is added to the Stream.

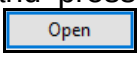


5. Click the **Add**  button in the View Panel Stream.



6. Navigate to the folder containing the Classification images.

   NOTE: *The images are in the ViDi Student Folder → Lab 6.2 & 7.2 Solder Problem.*

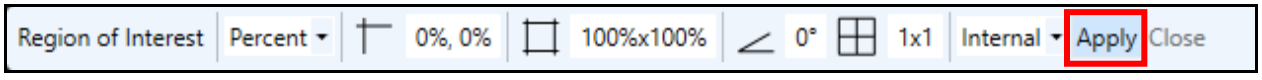
7. All images will be added to the Workspace, select the first image and press **<Ctrl+A>,** this will automatically select all the images. Click the **Open**  button.

   Once the images have been added, they will appear in the View Panel.

8. Click an image to select.

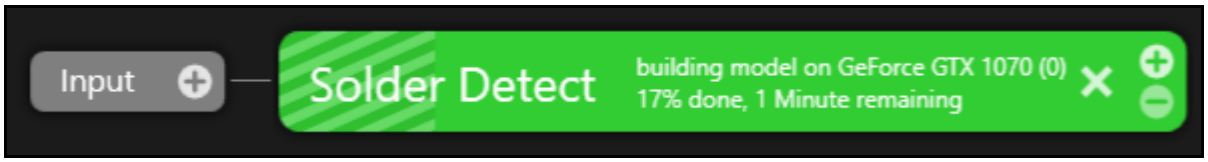
   The image will display in the Image Panel.

9. Define the Region of Interest (ROI). In this example we will be using the entire image, so click **Apply**.



The **ROI** is defined.

10. Open the **Tool Parameters**.
    Feature Size = 180 to 220 pixels so that the part can be properly detected.
11. Label the Views.
    NOTE: *This can be done by searching for 'ok' and labeling them as ok, and 'bad' and labeling them as bad. Some of the images have a file name of 'test' – these should not be labeled.*
12. Set your Training Selection to **50%**.
13. Click the **Train** button to train.
    The analysis begins.



14. Review the results.
    Flip through the images, notice some of them are not ok or bad and are labeled as test. These are specifically so that not all images are properly labeled.

Lab Exercise 7.1 – ViDi Tool Chaining

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi tools to solve their application

The Participant will utilize the following ViDi tools to successfully complete this lab exercise:

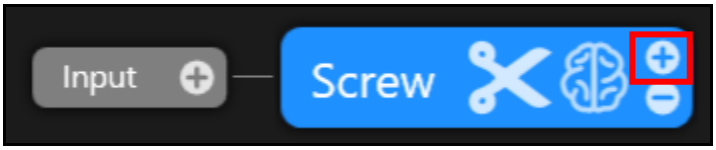
- Blue – Locate Tool
- Red – Analyze Tool
  - Unsupervised Mode
  - Supervised Mode
- Green – Classify Tool

Medical Screw – Part 2
Fixturing

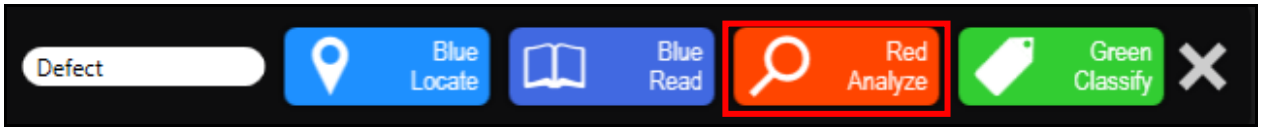
This lab exercise will continue with the medical screw example from lab exercise #4. After detecting the medical screw, we will now be detecting defects on the screw. This will be done by adding a Red – Analyze tool in Unsupervised mode.

Follow the steps below to complete the lab exercise:

1. Access the **Medical Screw** stream.
2. Click the **Input** button on the Blue – Locate tool to view the available ViDi tools.

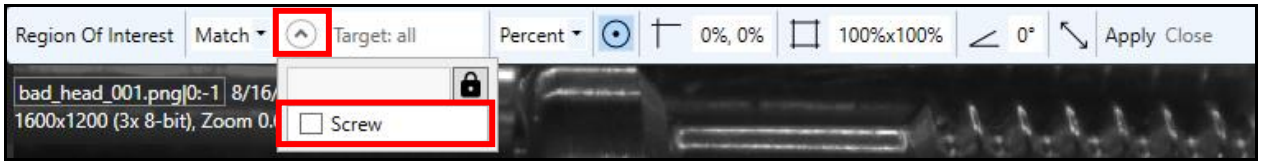


3. Enter *Defect* in the Tool name field and click the **Red Analyze** tool to select.

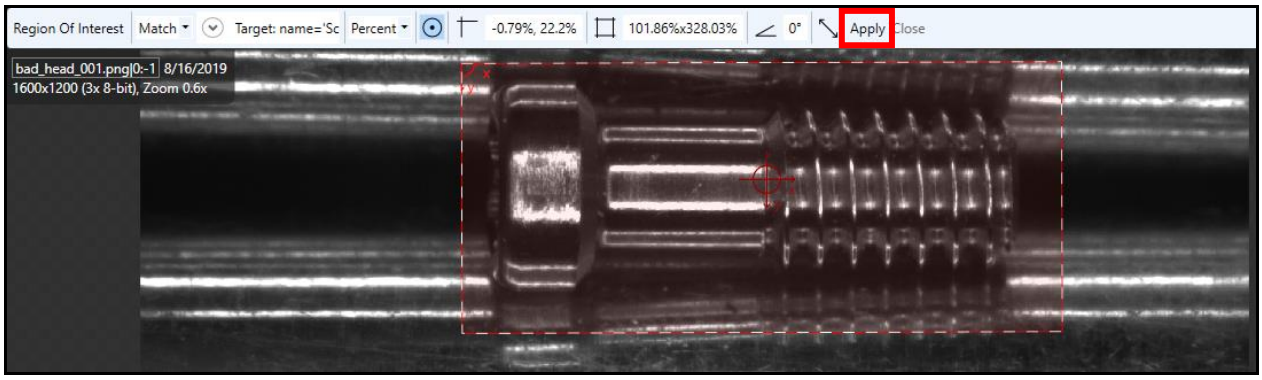


The **Red Analyze** tool is added to the Stream.

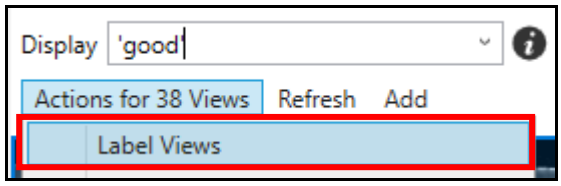
4. Define the ROI – notice that the Region of Interest toolbar has additional fields – open the arrow and select the **Screw** model (or the name that you gave your model from the Medical Screw Example in lab #4).

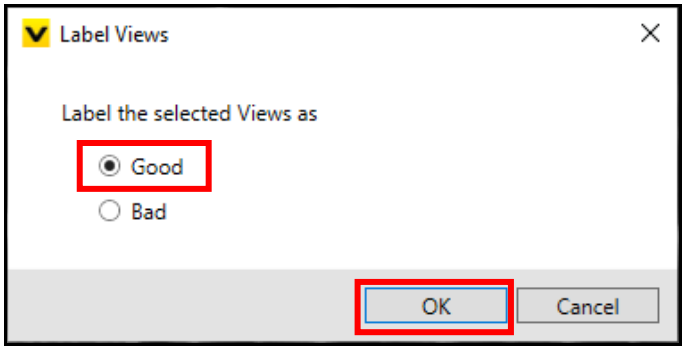


5. Position the ROI to include the screw, as shown below, and click **Apply**.



6. Label the images as follows:
   '*good'* should be labeled as good
   '*no'* should be labeled as bad
   '*bad'* should be labeled as bad





6. Click the **Train** button to select your Training Set and train the tool.

The Analysis begins.

7. Review the results.
8. Use the knowledge you gained in section 3 to identify and adjust the parameters of the Red Analyze tool to appropriately detect the good screws vs. the bad screws.

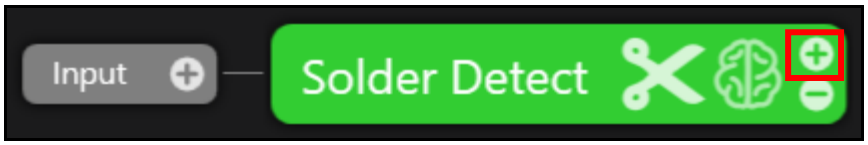
Solder Problem – Part 2
Selective Processing

This lab exercise will create a processing tool-chain with the inspection of an image based on the classification of a previous tool.

To do this we will utilize the Green Classify tool to first filter out only the results that have solder present. We will then segment this solder for future inspections.

Follow the steps below to complete the lab exercise:

1. Access the **Solder** stream.
2. Click the **Input** button on the Green – Classify tool to view the available ViDi tools.

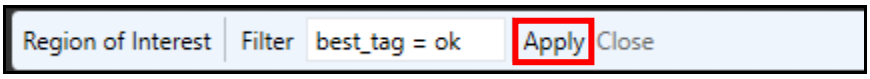


3. Enter *Segment* in the Tool name field and click the **Red Analyze** tool to select.

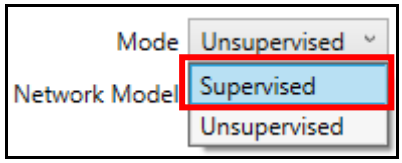
The **Red Analyze** tool is added to the Stream.

4. Enter *best_tag = 'ok'* and click **Apply** to define the Region of Interest.

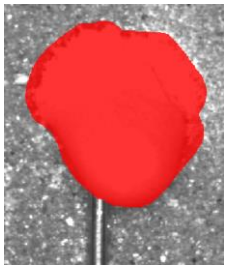
   NOTE: *This states that the only FOVs that will be processed are those that have been tagged as 'ok' by the classify tool.*



5. Open the Tool Parameters and change the Mode from Unsupervised to **Supervised**.



6. Use the line tool to label the Solder (20 – 30 images) as shown below:



5. Open the Tool Parameters and set the following:
   Flip = Horizontal
   Luminance = 5 – 10%
   Simple Regions = On (box checked)
6. Click the **Train** button to set your Training Set and train the tool.

The Analysis begins.

7. Review the results.
8. Use the knowledge you gained in section 3 to identify and adjust the parameters of the Red Analyze tool until all solder joints are properly segmented.

Battery Problem – Part 2
Dynamic Masking

This example will use the result from a Red – Analyze tool in Supervised mode to generate a dynamic mask.

In this case we are seeking to classify the batteries into their different models, but given the background of similar color, and varying shape we utilized the red tool to filter out just the battery to be classified. From there we will utilize a Green – Classify tool to classify the batteries into different models.

Follow the steps below to complete the lab exercise:

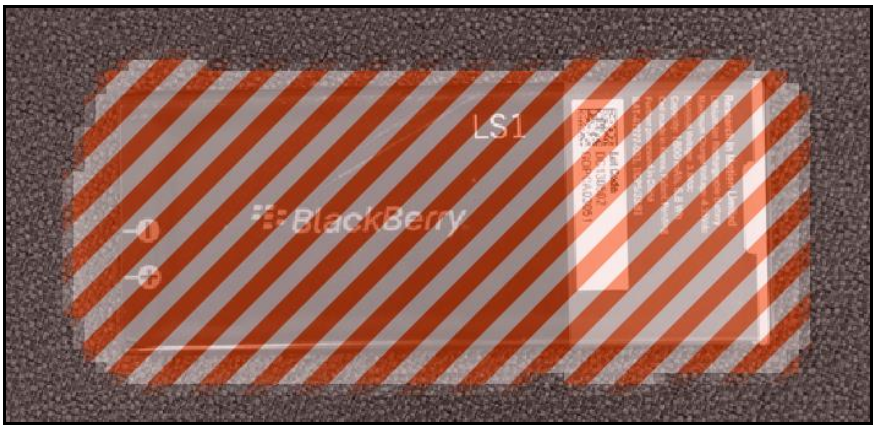
1. Access the **Battery Example** stream.
2. Click the **Input** button on the Red – Analyze tool to view the available ViDi tools.

3. Enter *Model* in the Tool name field and click the **Green Classify** tool to select.

The **Green Classify** tool is added to the Stream.

4. Define the Region of Interest – click the **Masks the Regions inside the ROIs** button.

The battery will be masked as shown below.

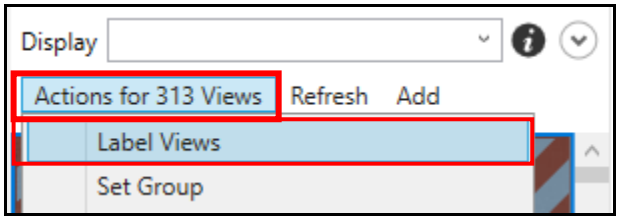


5. Click the **Invert Mask** button.

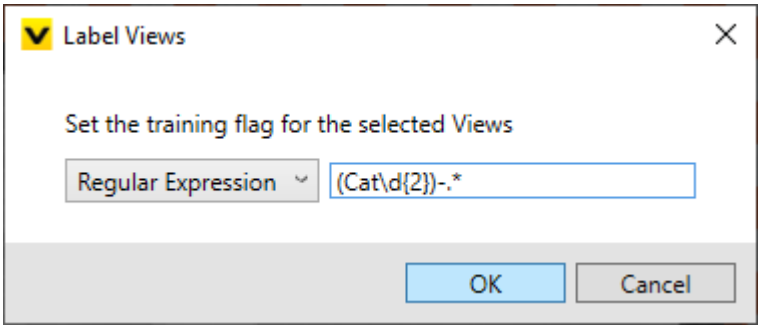
The mask inverts – this will allow only the area of the battery to be considered when training the tool, all other areas will be disregarded due to the overlaying mask.

6. Click **Apply**.

7. Click **Actions for 313 Views** and select **Label Views**.



8. Select **Regular Expression** from the drop-down list and enter *(Cat\d{2})-.** in the field.



9. Click the **Train** button to train the tool.

The analysis begins.

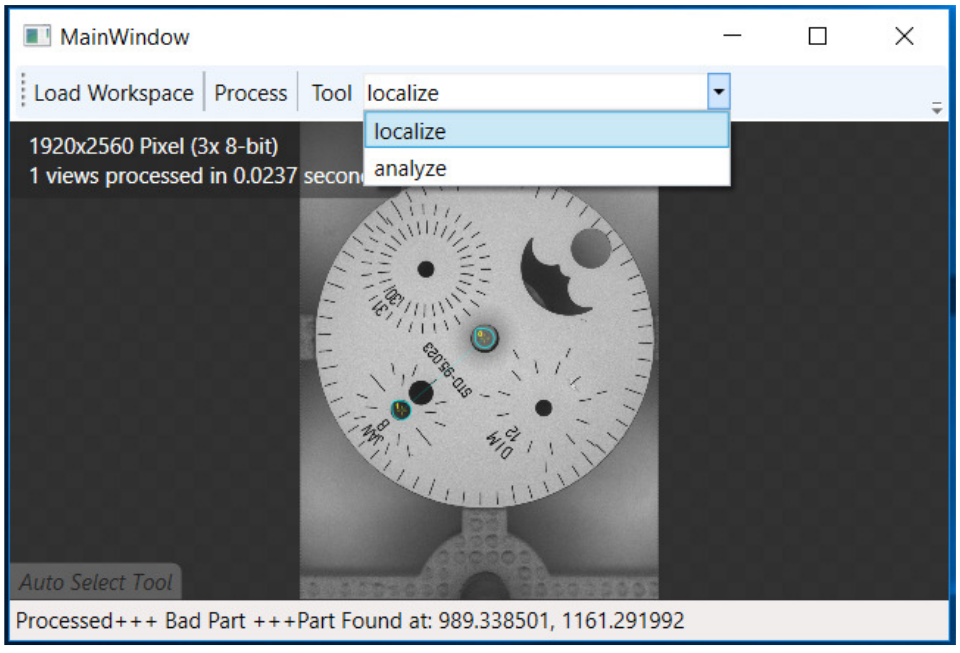
10. Review the results and modify the training set.

NOTE: *If the same operation was completed without the Red – Analyze tool the results would be lower. To compare results, add another Green – Classify tool from the input and attempt to resolve the problem using the full ROI.*

Lab Exercise

COGNEX

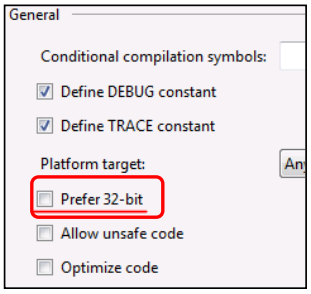
Building an App



COGNEX

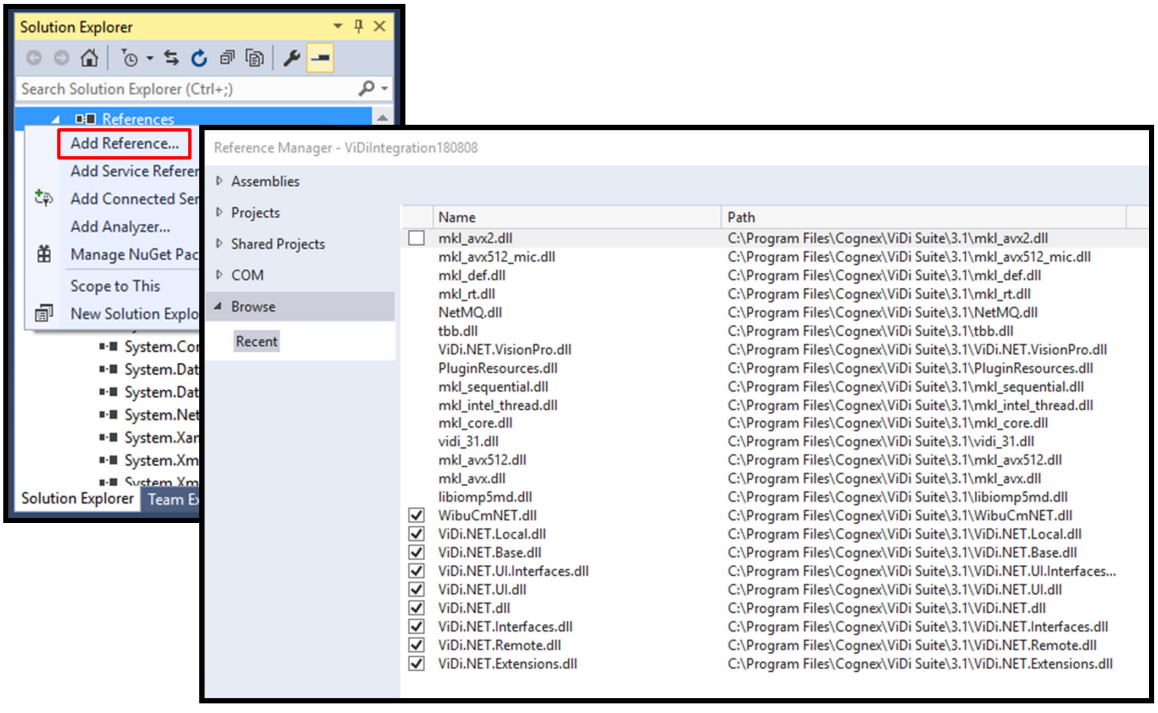
Runtime API C# / Getting Started

1. Create a new C# WPF project in visual studio.
2. In the project properties, uncheck *Prefer 32-bit*



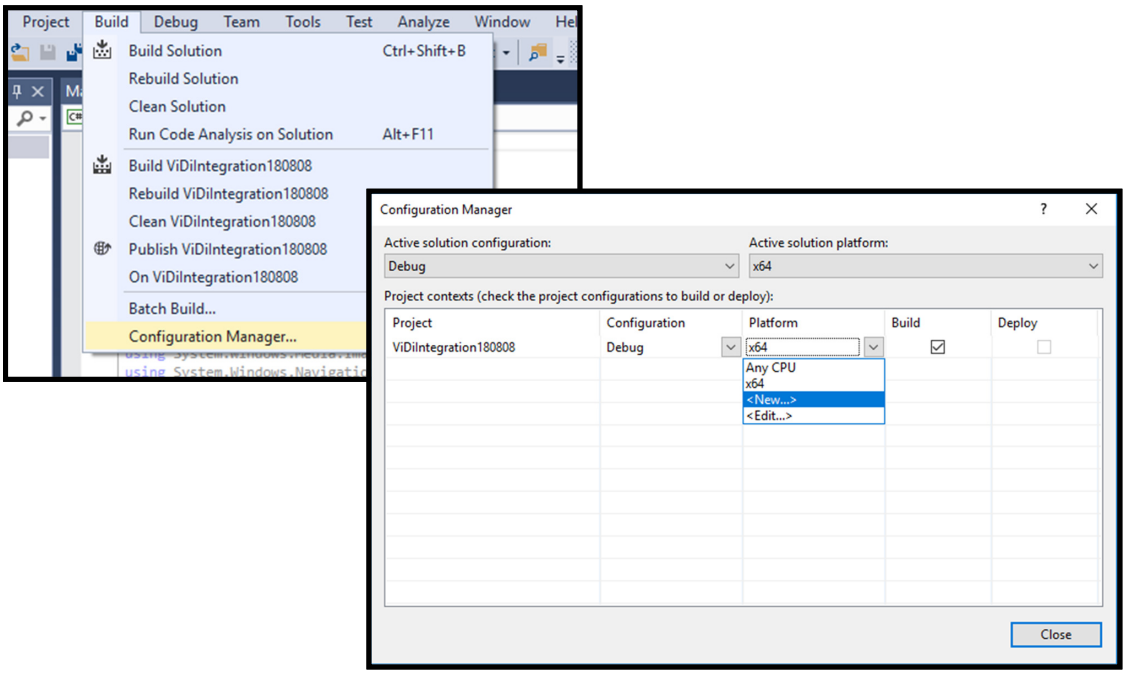
COGNEX

Adding References



COGNEX

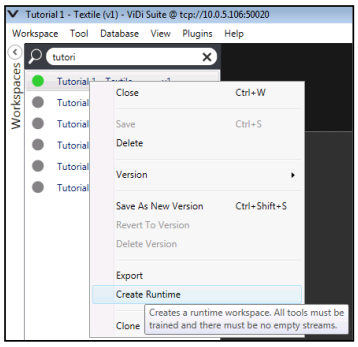
Build Configuration



COGNEX

Runtime API / Getting Started

1. Train all the tools in a workspace using the GUI.
2. Export a runtime workspace.
3. Use this runtime workspace in your application.



COGNEX

Runtime API C# / Getting Started

MainWindow.xaml.cs
Initializes a ViDi local control when loading the main window

```
using ViDi2.Local;
using ViDi2.Runtime;
using ViDi2;

namespace ViDiExample
{
   public partial class MainWindow : Window
   {
      private ViDi2.Runtime.IControl control = null;
      public MainWindow()
      {
         control = new ViDi2.Runtime.Local.Control();
         InitializeComponent();
      }
   }
}
```


Load a Runtime Workspace

MainWindow.xaml
Adding a Tool Bar and Button to the window:

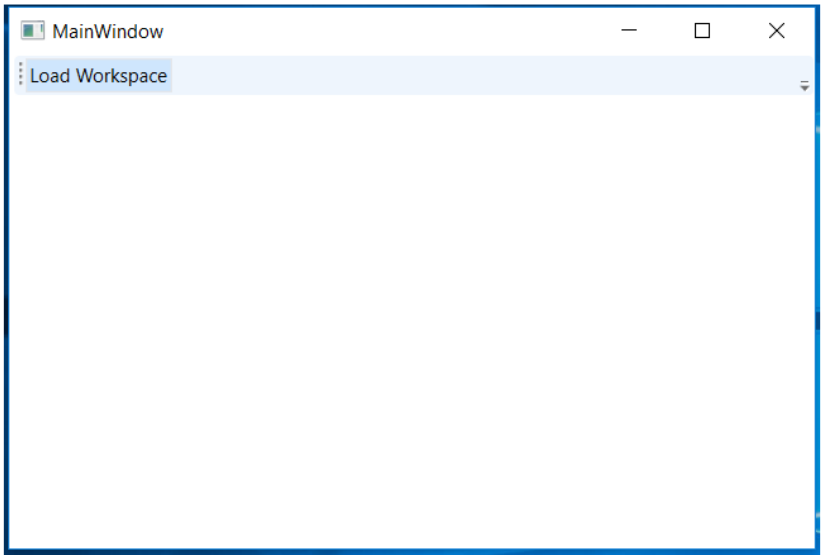
```
<DockPanel >
      <ToolBar DockPanel.Dock="Top">
            <Button Content="Load Workspace" Click="load"
           VerticalAlignment="Stretch"/>
      </ToolBar>
      <Grid>
      </Grid>
   </DockPanel>
</Window>
```

Load a Runtime Workspace

MainWindow.xaml.cs
Load a Runtime Workspace

```
public partial class MainWindow : Window
   {
       private ViDi2.Runtime.IControl control = null;
       string path = @"C:\Users\gdavitor\Desktop\ViDi Student Folder\IntegrationExample\";
       ViDi2.Runtime.IWorkspace workspace = null;
       public MainWindow()
       {
           control = new ViDi2.Runtime.Local.Control();
           InitializeComponent();
       }

       private void load(object sender, RoutedEventArgs e)
       {
           workspace = control.Workspaces.Add("workspace", path + "workspace.vrws");
       }

   }
```


IT'S ALIVE!



Processing an Image

MainWindow.xaml
Adding a Button and Status Bar with Text

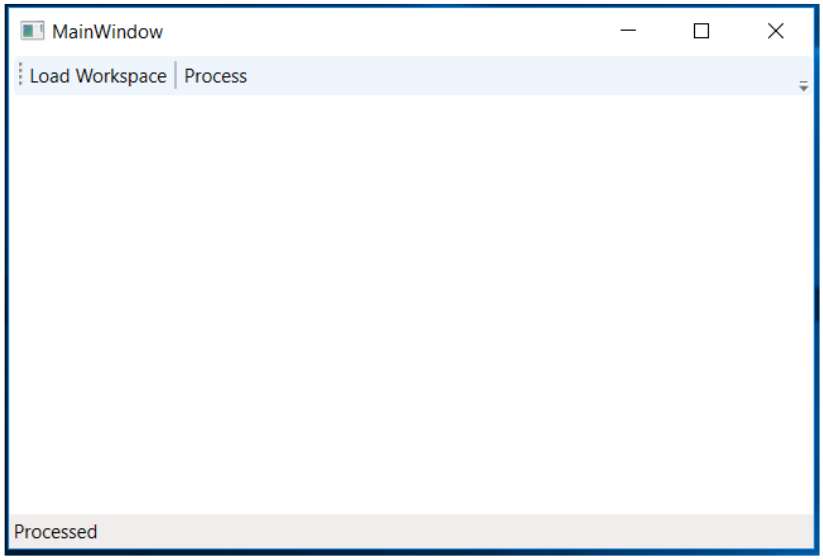
```
<DockPanel>
        <ToolBar DockPanel.Dock="Top">
            <Button Content="Load Workspace" Click="load" VerticalAlignment="Stretch"/>
            <Separator/>
            <Button Content="Process" Click="process" VerticalAlignment="Stretch"/>
        </ToolBar>
        <StatusBar DockPanel.Dock="Bottom" VerticalAlignment="Stretch">
            <TextBlock Text="Started" Name="slbStatusLabel"/>
        </StatusBar>
        <Grid>
        </Grid>
    </DockPanel>
```


Processing an Image

MainWindow.xaml.cs
Utilizing the Stream in the Workspace to Process an image

```
private void load(object sender, RoutedEventArgs e)
{
    //loads the workspace
    workspace = control.Workspaces.Add("workspace", path + "workspace.vrws");
    //Changes the status bar text to reflect the Workspace Loaded Action
    slbStatusLabel.Text = "Loaded";
}

ViDi2.IStream stream = null;
ViDi2.ISample sample = null;

private void process(object sender, RoutedEventArgs e)
{
    //gets the default stream
    stream = workspace.Streams["default"];
    //load an image from disk
    ViDi2.IImage image = new ViDi2.WpfImage(path + "image.png");
    //process this image with all the tools in the stream
    sample = stream.Process(image);
    //Changes the status bar text to reflect the Processed Action
    slbStatusLabel.Text = "Processed";
}
```

IT WORKS!



COGNEX

Adding Red – Analyze Score

MainWindow.xaml.cs inside of the process method

```
//Adding Results
//Removing Markings from Sample
IMarking marking = sample.Markings["analyze"];
//Removing View from markings
IView view = marking.Views[0];
//Creating a redView Object from the original View
IRedView redView = (IRedView)view;
if(redView.Score<redView.Threshold.Lower)
{
    //Checks for Good Part via Scoare and Threshold
    slbStatusLabel.Text += "--- Good Part ---";
}
else if(redView.Score>redView.Threshold.Upper)
{
    //Checks for Bad Part via Score and Threshold
    slbStatusLabel.Text += "+++ Bad Part +++";
}
else
{
    //If its in between the part is unclassified
    slbStatusLabel.Text += "*** Unclassified ***";
}
```


COGNEX

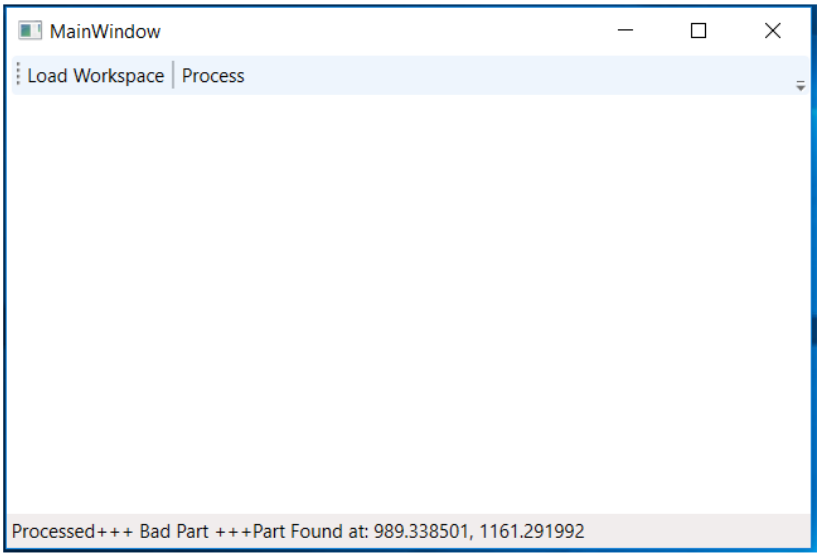
Adding Blue – Locate Tool Position

MainWindow.xaml.cs inside of the process method

```
//Removing the markings of the blue tool
IBlueView blueView = sample.Markings["localize"].Views.First() as IBlueView;
Point position = blueView.Matches.First().Position; //position of the best score model
//Adding position of the part to the Status Bar
slbStatusLabel.Text += "Part Found at: " + position.X + ", " + position.Y;
```


COGNEX

RESULTS!



COGNEX

Adding a Display!

MainWindow.xaml

We need to add refernces and create the SampleViewer in the xaml code

```
xmlns:v="clr-namespace:ViDi2.UI;assembly=ViDi.NET.UI"
        Title="MainWindow" Height="350" Width="525">
    <DockPanel>
        <ToolBar DockPanel.Dock="Top">
            <Button Content="Load Workspace" Click="load" VerticalAlignment="Stretch"/>
            <Separator/>
            <Button Content="Process" Click="process" VerticalAlignment="Stretch"/>
        </ToolBar>
        <StatusBar DockPanel.Dock="Bottom" VerticalAlignment="Stretch">
            <TextBlock Text="Started" Name="slbStatusLabel"/>
        </StatusBar>
        <Grid>
            <v:SampleViewer Name="sampleViewer" AllowDrop="True"/>
        </Grid>
    </DockPanel>
```


Adding a Display!

App.xaml

Letting the display know how to mark images

```
<Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary

          Source="pack://application:,,,/ViDi.NET.UI;component/Styles/MarkingStyle.xaml“
                    />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>
```

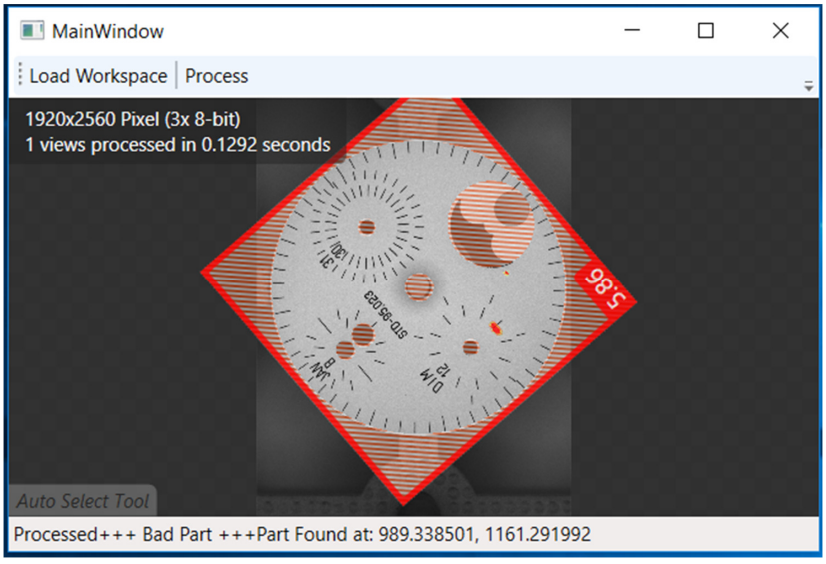
Adding a Display!

MainWindow.xaml.cs inside of the process function

```
//send the sample to the sampleViewer
sampleViewer.Sample = sample;
```


COGNEX

IT SHOWS THINGS!



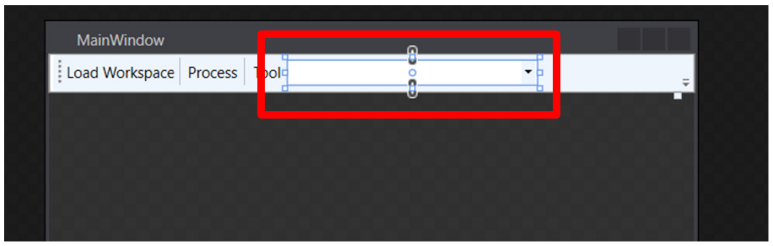
COGNEX

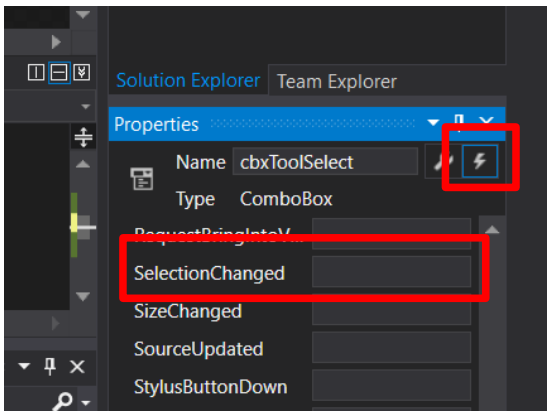
Changing Views

MainWindow.xaml
Adding a label and drop down box

```
<ToolBar DockPanel.Dock="Top">
     <Button Content="Load Workspace" Click="load" VerticalAlignment="Stretch"/>
     <Separator/>
     <Button Content="Process" Click="process" VerticalAlignment="Stretch"/>
     <Separator/>
     <Label Content="Tool" VerticalAlignment="Stretch"/>
     <ComboBox Name="cbxToolSelect" Width="200"/>
 </ToolBar>
```


Changing Views





Changing Views

MainWindow.xaml.cs

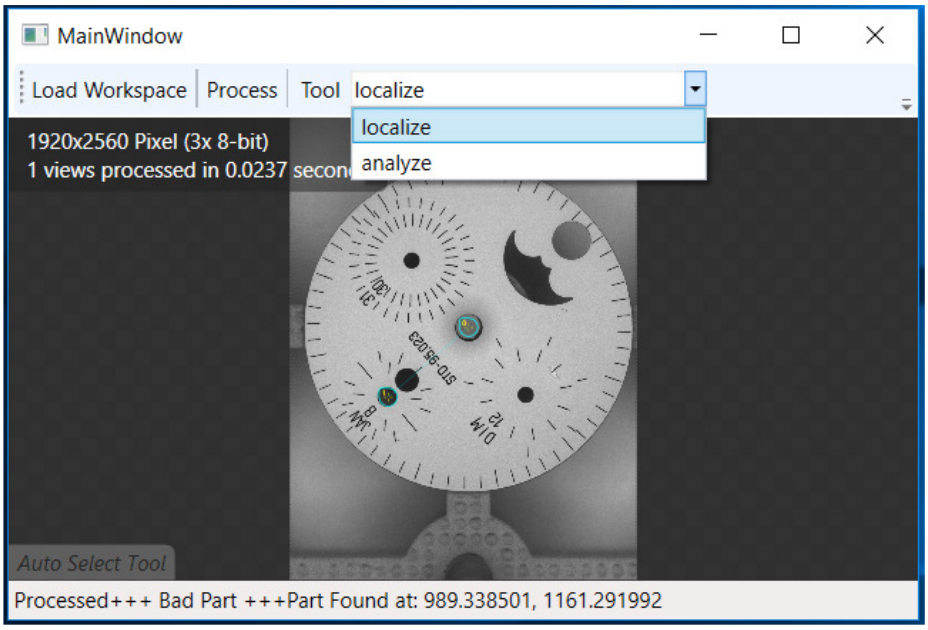
Inside of process function

```
//Sets the ComboBox source equal to the List of Tool Names
cbxToolSelect.ItemsSource = sample.Tools.Descendants.Select(x => x.Name).ToList();
```

System Generated Event

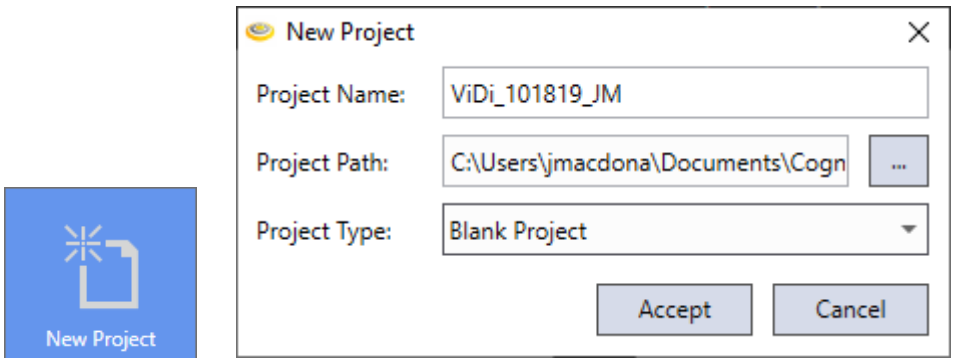
```
private void cbxToolSelect_SelectionChanged(object sender, SelectionChangedEventArgs e)
{

      //Set the sampleViewer image equal to the select tool
      sampleViewer.ToolName = cbxToolSelect.SelectedValue.ToString();
}
```


You now have an App!

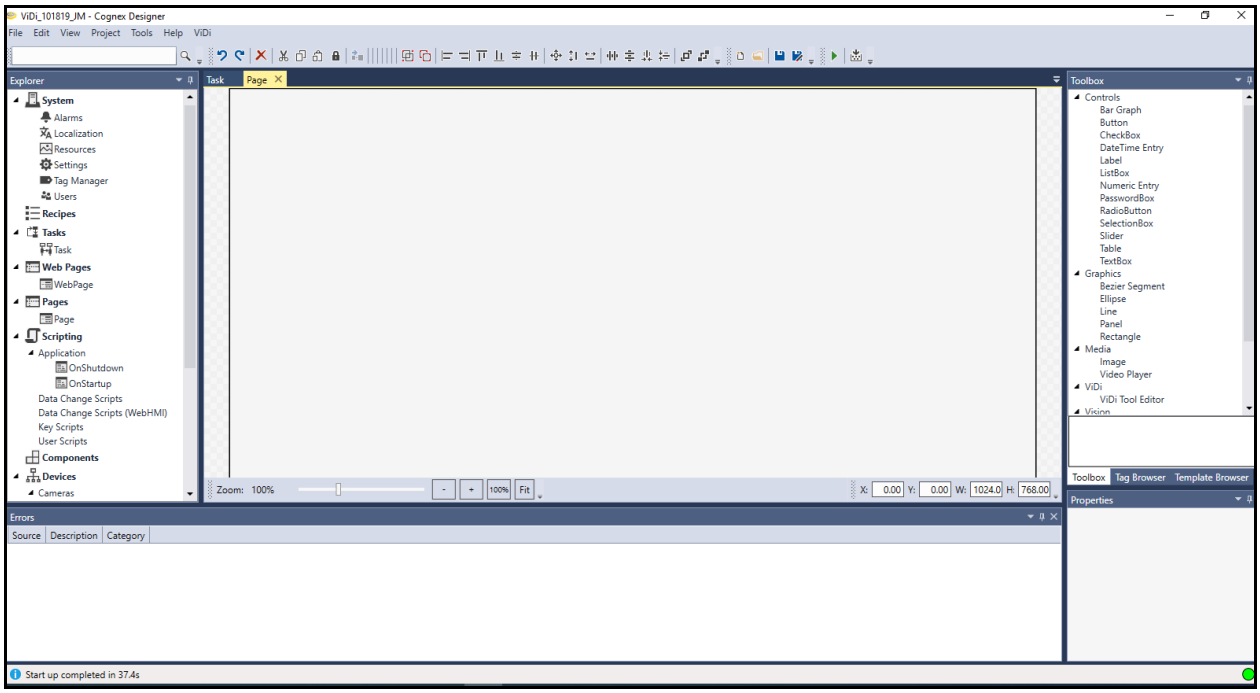


2. Click the **New Project** button, name your new project and click the **Accept** button.

   NOTE: *It is recommended that you name the project **ViDi_MMDDYY_Initials**. Where MM is the month, DD is the day, and YY is the year. The Initials are of your first and last names.*

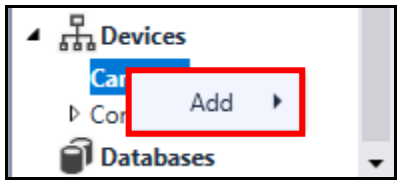


   The **Cognex Designer** application launches.

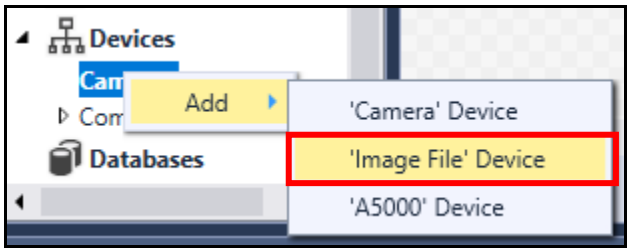


3. Right-click the **Cameras** link under **Devices** in the Cognex Designer Explorer window.

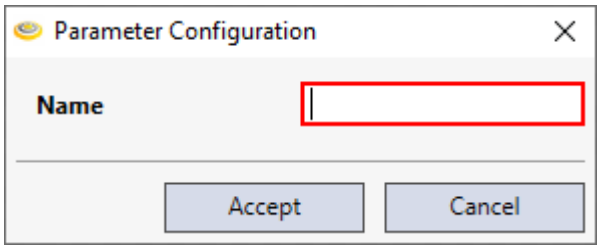
   The **Add** Menu displays.

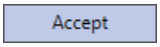


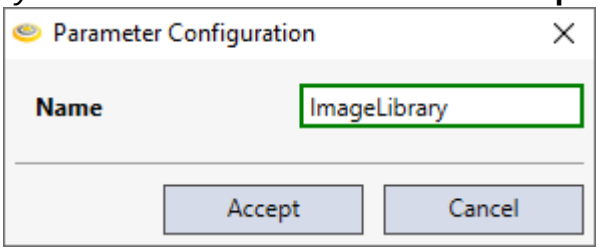
4. Hover your mouse over the word **Add** and select 'Image File' Device from the fly out.



The **Parameter Configuration** window displays.

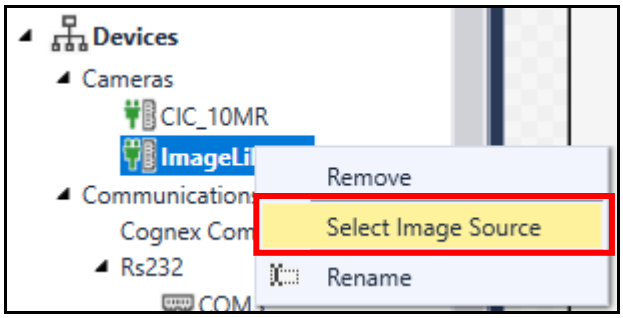


5. Enter *ImageLibrary* in the **Name** field and click the **Accept**  button.

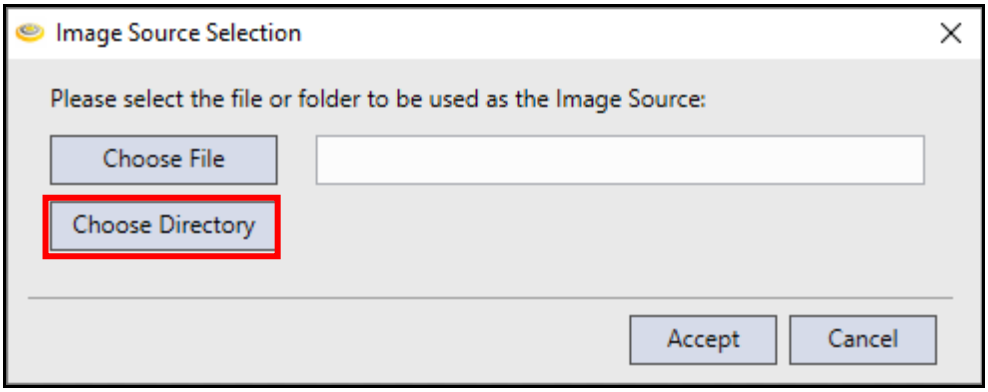


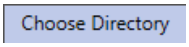
ImageLibrary is added under Cameras.

6. Right click on **ImageLibrary** and choose **Select Image Source** from the fly-out.



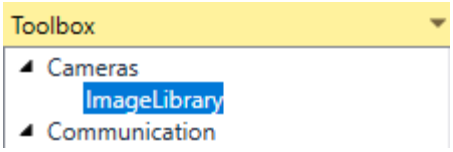
The **Image Source Selection** dialog displays.



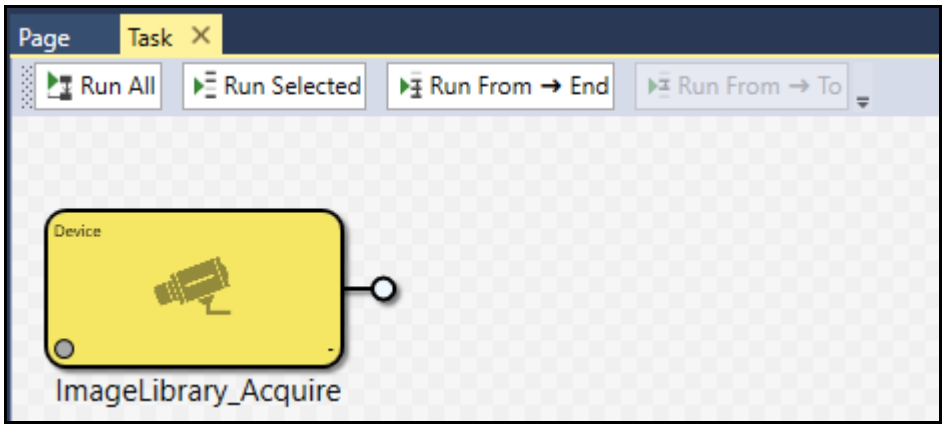
7. Click the **Choose Directory**  button and navigate to the location of your image database.

NOTE: *The watch face images can be found in the ViDi Student Folder on the desktop.*
8. Click the **Accept**  button.

ImageLibrary has been added under Cameras in the Cognex Designer toolbox.

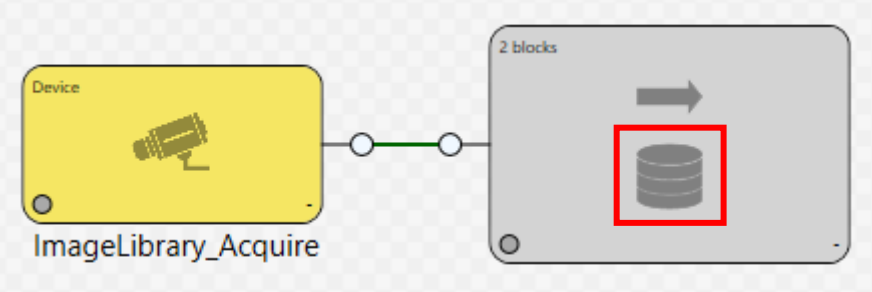


9. Drag and drop the *ImageLibrary* Block into the Cognex Designer Task.

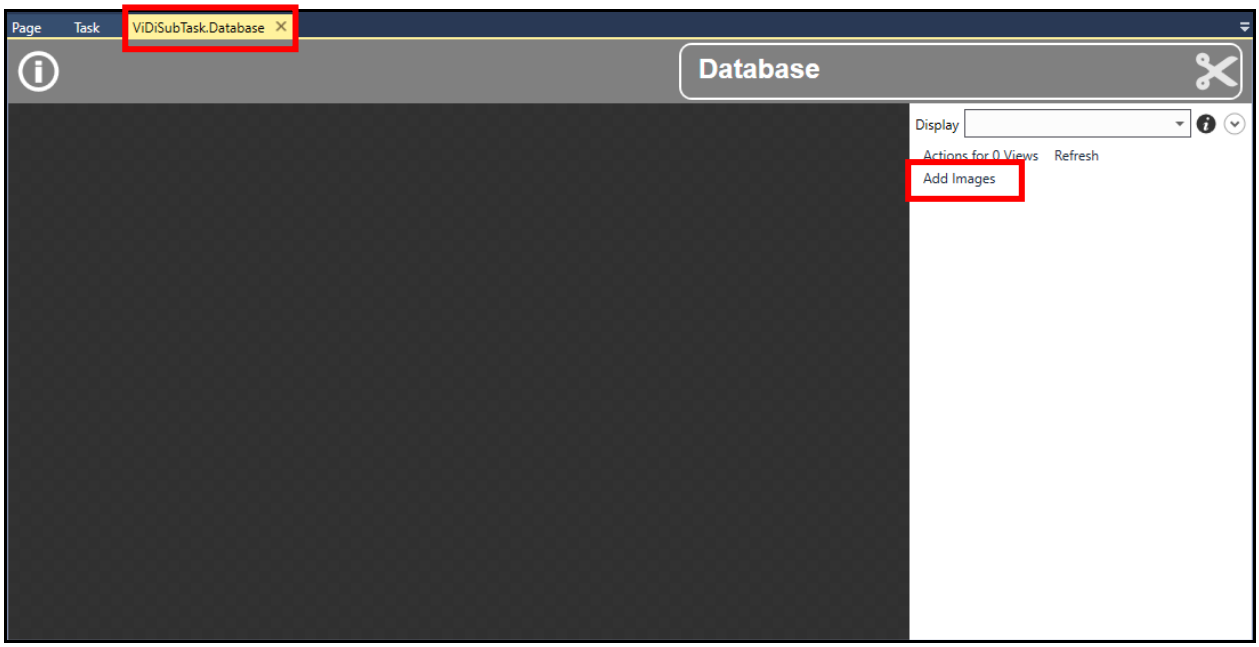


10. Add a ViDi Sub Task to the Cognex Designer Task.
NOTE: *The ViDi Sub Task is in the ViDi Suite section of the Toolbox*.

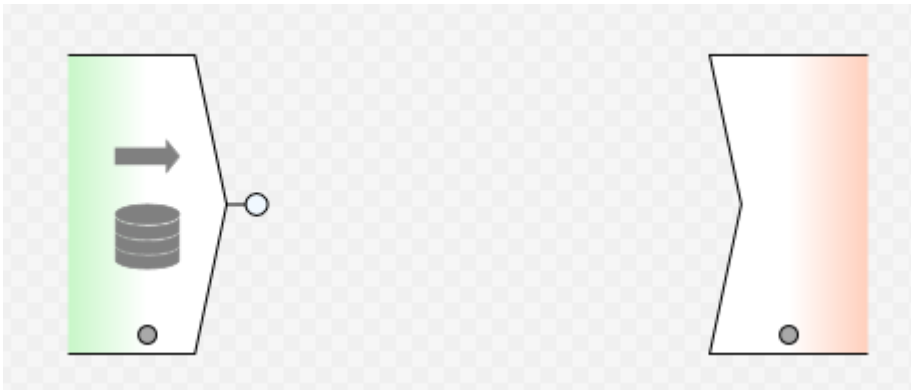
11. Connect the Output of the Image source to the input of the ViDi Sub Task.



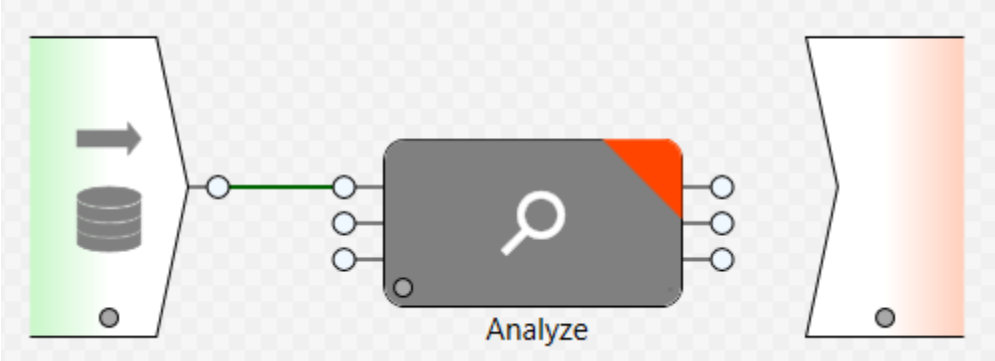
12. Double click the **Image Database** icon on the ViDi Sub Task.
    The **Image Database** opens in a new tab.



13. Click the **Add images** link, browse to your image folder, select your images and click the **Open** button.
    The images display.
14. Return to the **Task** tab and double-click the grey background of the ViDi Sub Task.
    The **ViDi Sub Task** displays.



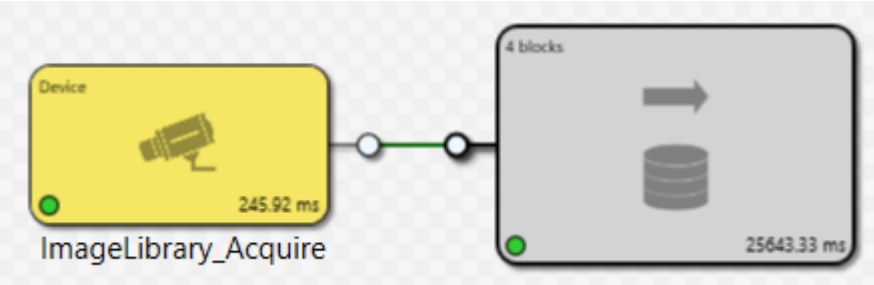
15. Add an **Analyze** tool to the ViDi Sub Task.
    NOTE: *The Analyze tool is found in the Toolbox under ViDi Suite → ViDi Tools → Analyze.*
    The **Analyze tool** is added.



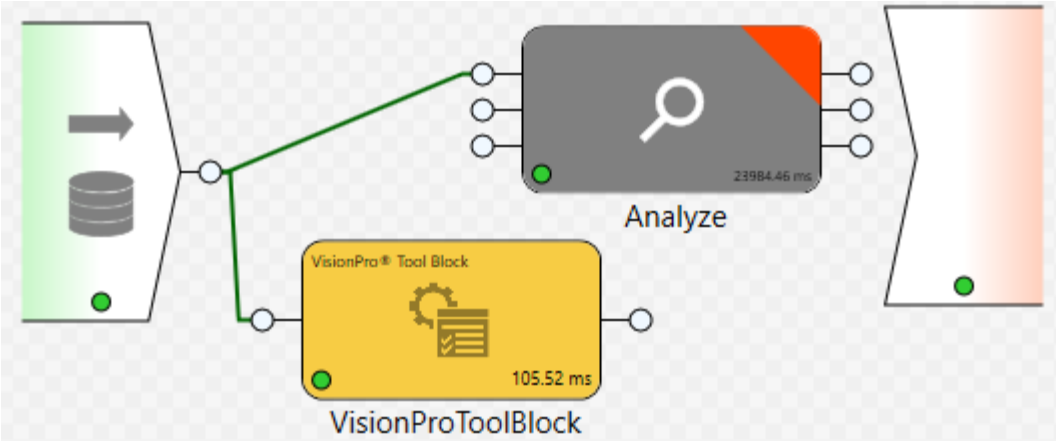
NOTE: *The tool will auto link the Image input of any ViDi tool to the image output of the Image Database.*

Fixturing the Image

16. Return to the **Task** tab and click the **Run All**  button.
    The image is brought through the Task.

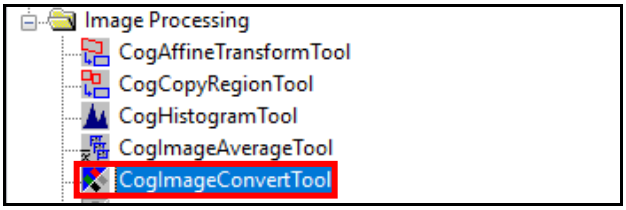


17. Return to the ViDiSubTask tab and add a **VisionPro® Tool Block** to the ViDi Sub Task and link the Image Output from the database to the Image Input of the VisionPro Tool Block.

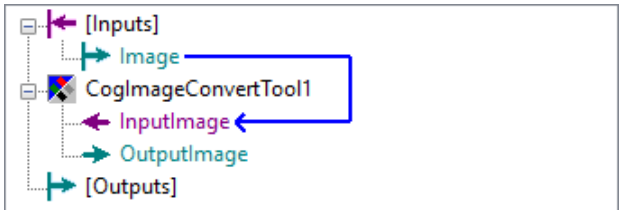


18. Double-click the VisionPro Tool Block to open.
19. Add a **CogImageConvertTool.**

NOTE: *The CogImageConvertTool is found in the Toolbox under Image Processing.*

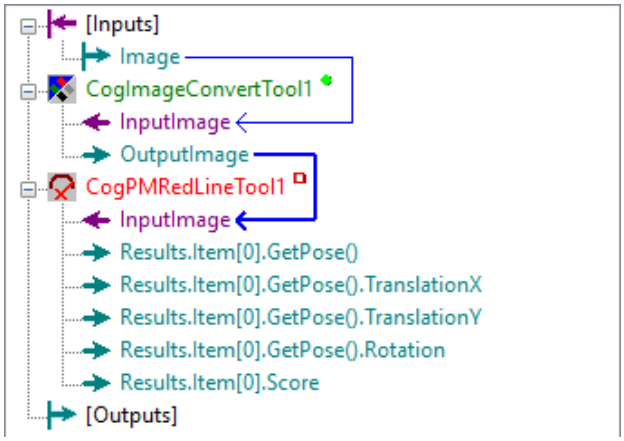


20. Link the [Inputs].Image to the InputImage of the CogImageConvertTool.



NOTE: *This tool is being added because the images are not in a format that is accepted by the PatMax RedLine tool (which will be added next).*

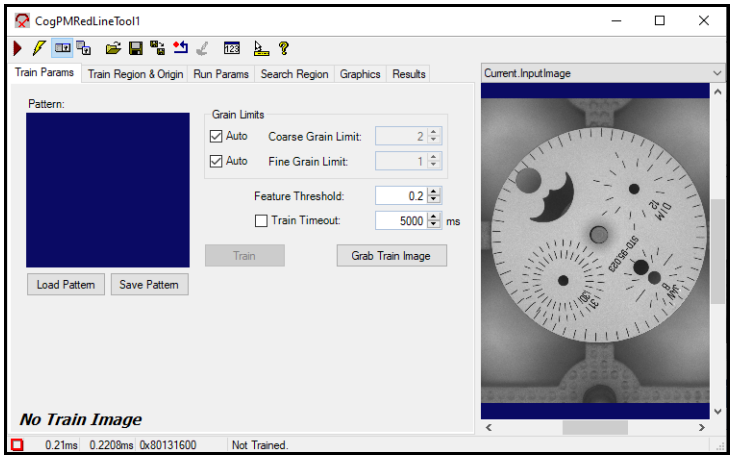
21. Add a **CogPMRedLine** tool to the VisionPro Tool Block and link the Output Image from the CogImageConvertTool to the Input Image of the CogPMRedLine tool.
22. Click the **Run Tools** button.

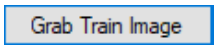


NOTE: *The CogPMRedLine tool fails because it does not have a pattern trained.*

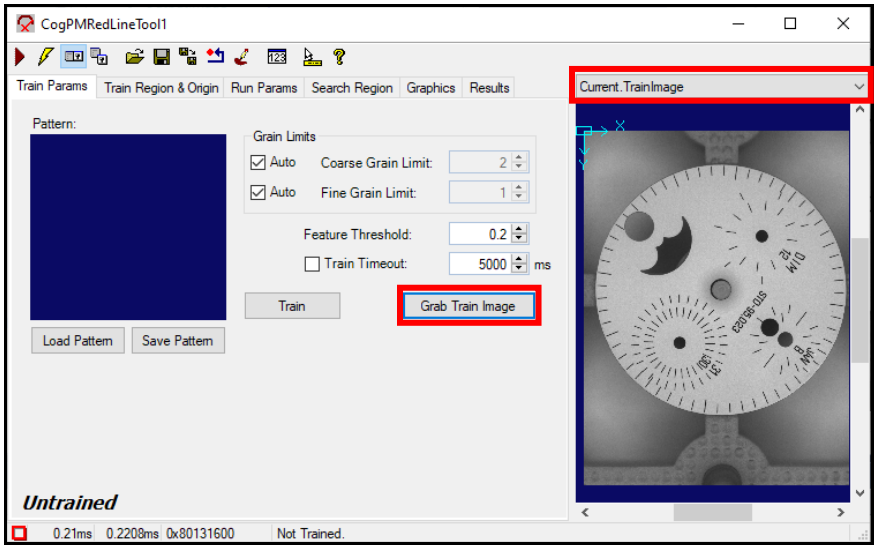
23. Double-click the CogPMRedLineTool to access the tool settings.

    The **Tool Settings** dialog displays.

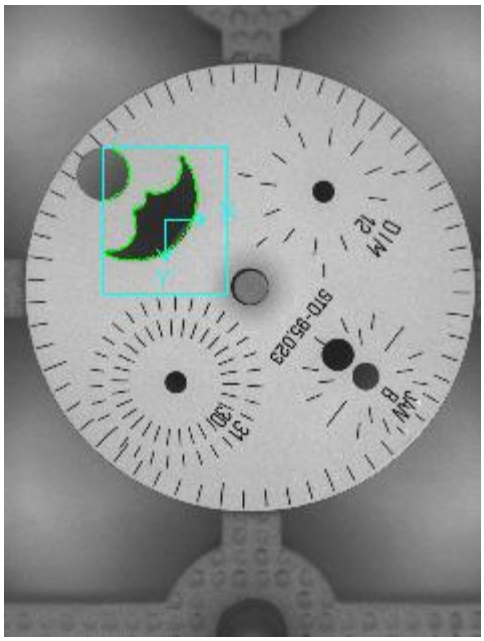


24. Change the selected image buffer to **Current.TrainImage** and click the **Grab Train Image** 

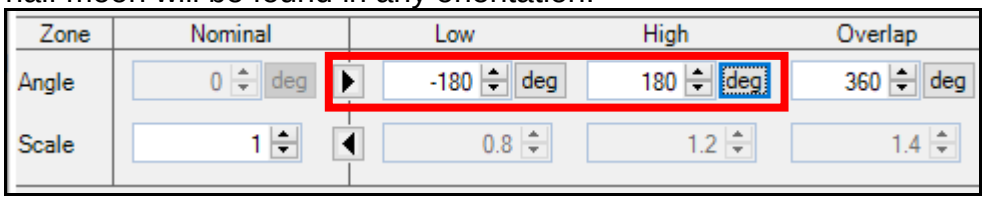
button.



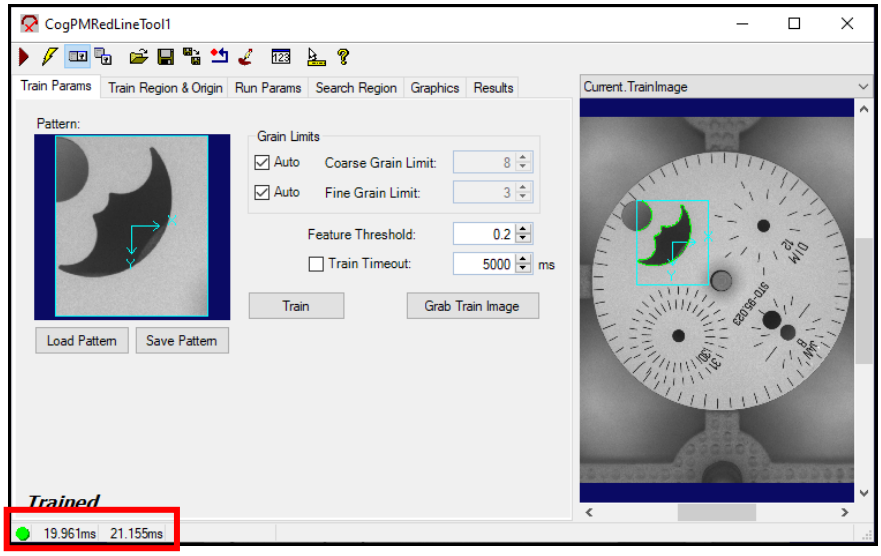
25. Click the **Train Region & Origin** tab and select **CogRectangleAffine** as the Region Shape. Position the Region as shown below, click the **Center Origin** button and train the tool.



26. Click the **Run Params** tab and click the left arrow to adjust the Angle setting so that the half moon will be found in any orientation.



27. Click the CogPMRedLine tool's **Run** button.

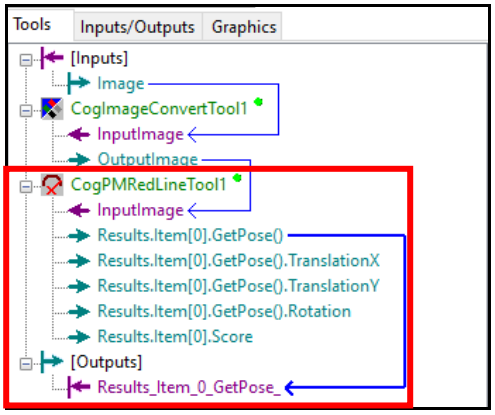


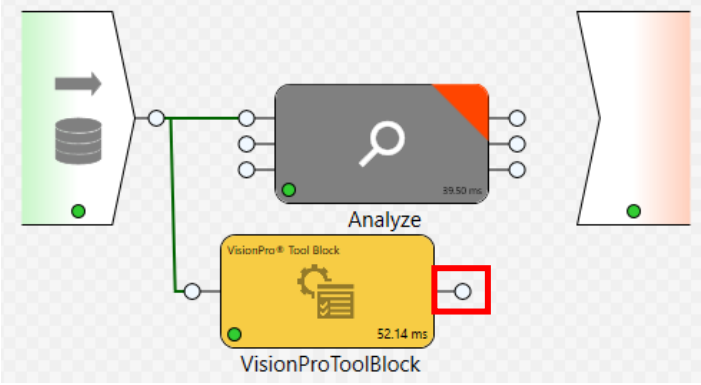
28. If the LED shows as green, close the window, if not consult with the instructor.

    NOTE: *This tool outputs a Pose, this contains X and Y position data, as well as rotational data of the part. We want this available in the ViDi Sub Task so that the Red-Analyze tool can be fixtured.*

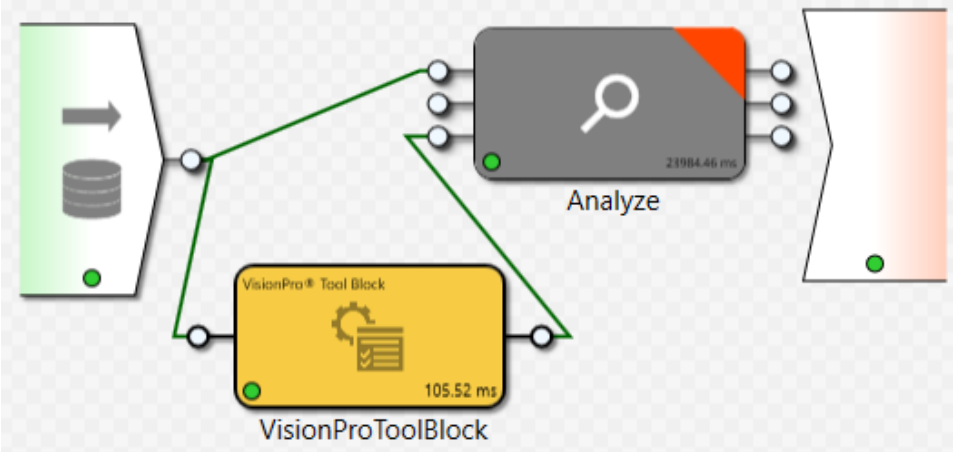
29. Drag the Pose pin from the CogPMRedLine tool to the Outputs section of the tool block.

    This generates an output for the pose inside the of the tool block as well as on the VisionPro ToolBlock on the ViDi Sub Task.



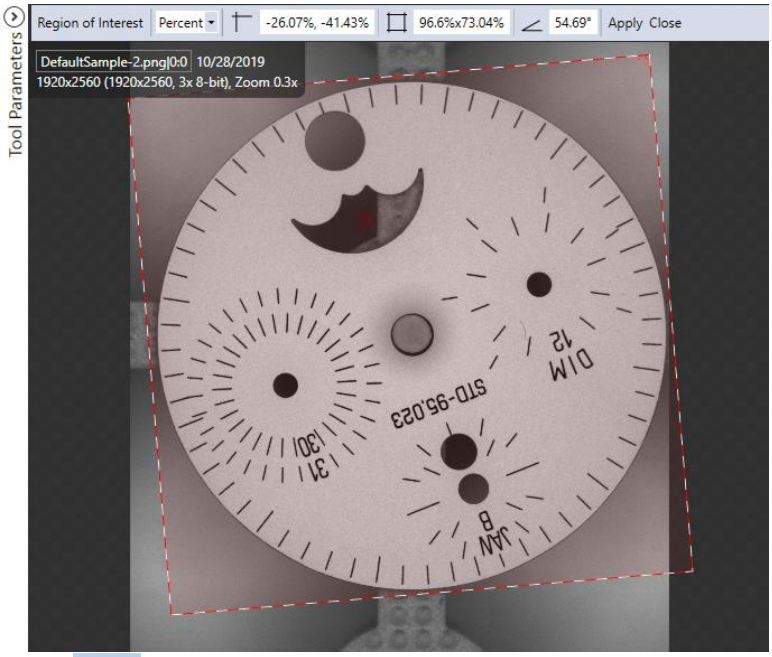


30. Link the Pose pin from the VisionPro ToolBlock to the Pose pin of the Red-Analyze tool.



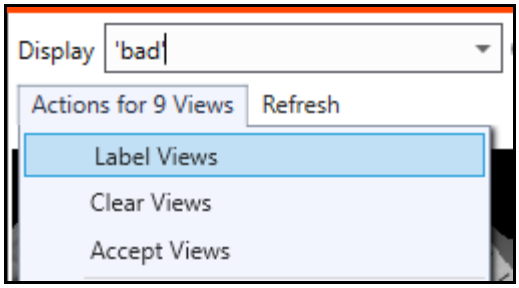
31. Return to the **Image Database** tab and click the **Process** button (Scissors) at the top right hand side of the window.

32. Return to the ViDi Sub Task tab and double-click the Red-Analyze tool to open. Select the first image and adjust the ROI so that it is a square encompassing the entire watch face, as shown below.

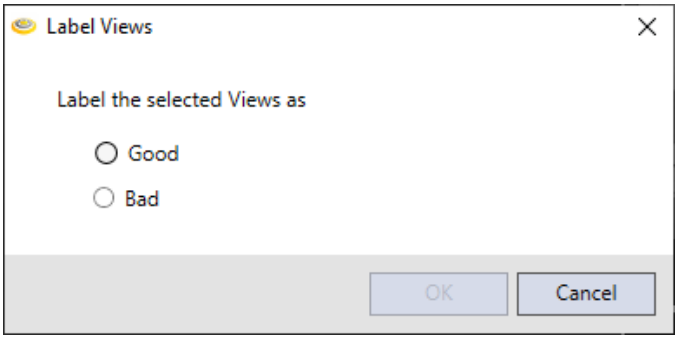


33. Click the **Apply** Apply button to apply the ROI to all images.
34. Confirm that all the images are now facing the same way.
35. Next, we will label the views.

36. Enter **'bad'** in the display field and press the **<Enter>** key.

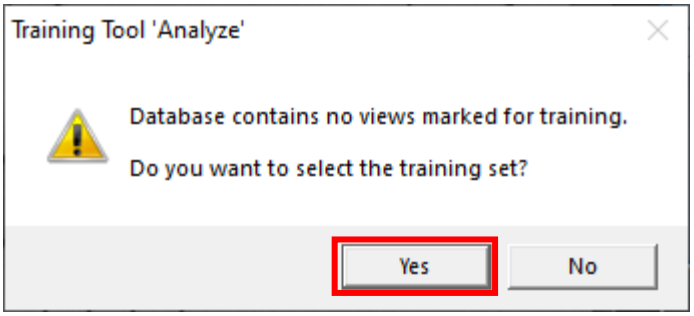


37. Select **Label Views**.
    The **Label Views** dialog displays.

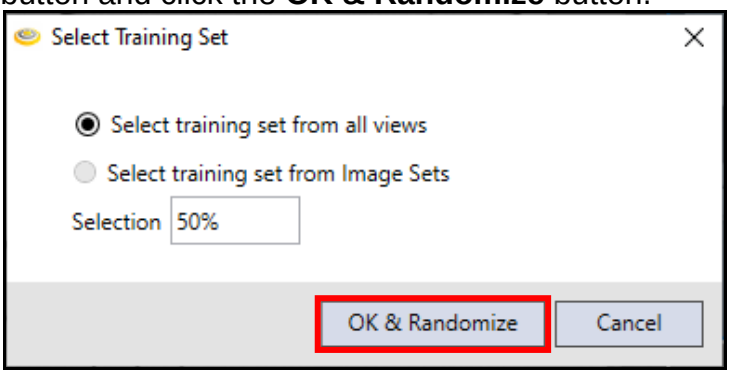


38. Click the **Bad** radio button and click the **OK** button.
39. Label the remaining views as **Good**.
    NOTE: *This can be done by selecting Not Labeled from the Display drop-down list and selecting the Good radio button on the Label Views dialog.*
40. Confirm the **Mode** is set to *Unsupervised* in the Red Tool Parameters.
41. Train the tool by clicking the **Train the Tool** button.

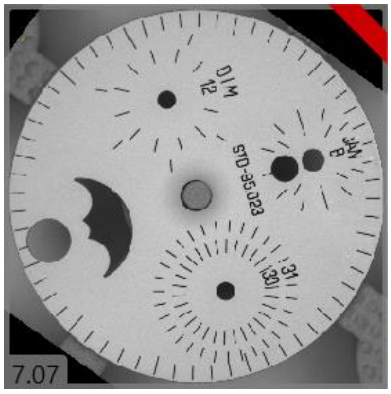
42. The **Training Tool 'Analyze'** dialog displays. Click the **Yes** button to continue.



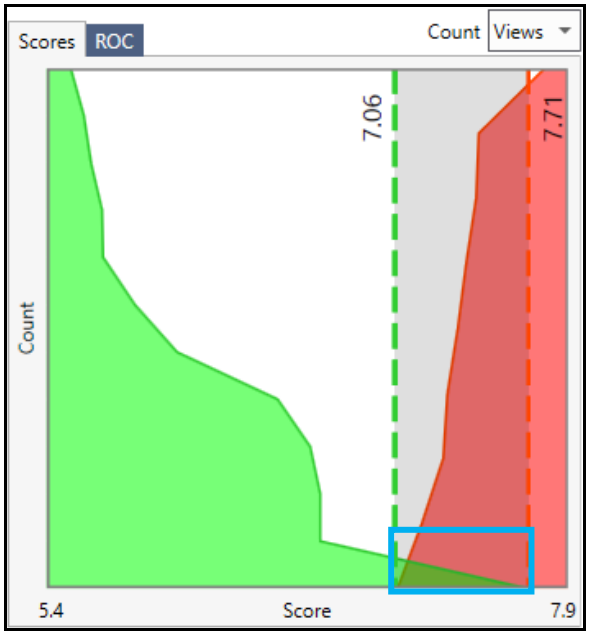
43. The **Select Training Set** dialog displays. Click the **Select training set from all views** radio button and click the **OK & Randomize** button.



44. At this point we should have images and statistics results. We will analyze these to see which actions need to be taken to get perfect, or near perfect results.
Below are some examples of Good and Bad results that have *not* been classified, this is indicated by the grey border.



This is because there is an area where a score is between Good and Bad and the tool is unable to detect which is which. This can be found in the Scores graph which is found in the Database Overview fly out.

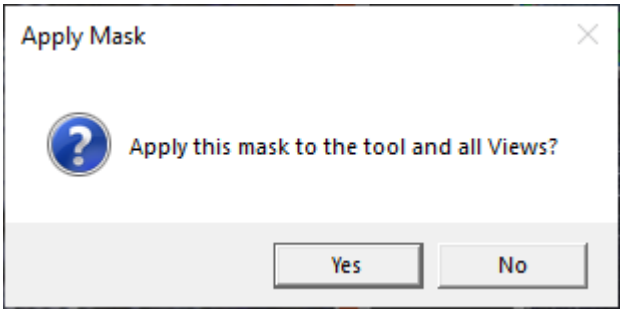


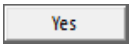
45. Note, that on the unclassified good image the heatmap shows defects around the watch face.
46. Right-click on the image in the main image window and select **Edit Mask.**

    The Mask toolbar displays.
47. Click the **Add Border** button

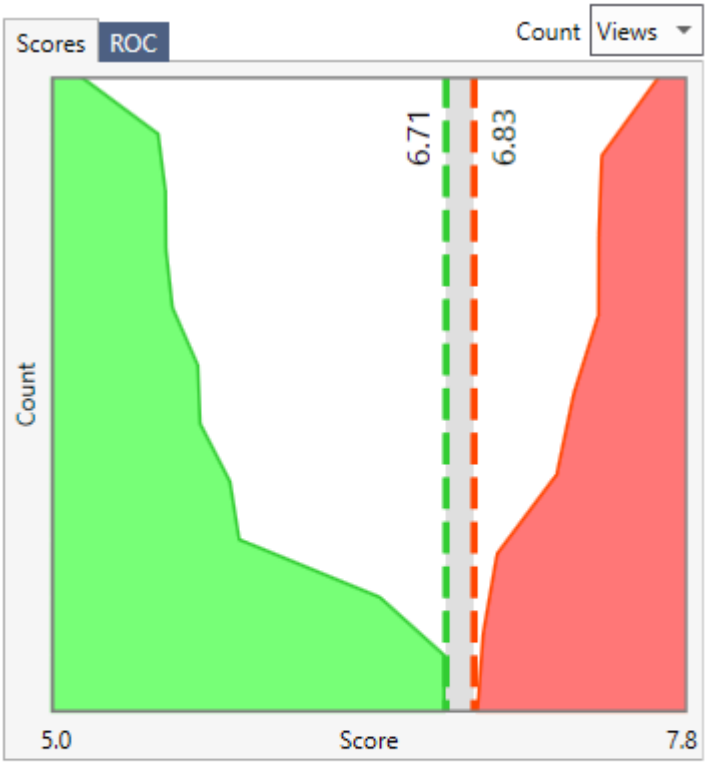
48. Position the Mask to display only the Watch Face.
49. Click **Apply** on the Mask Tool bar.

The Apply Mask dialog displays.



50. Click the **Yes**  button to **Apply this mask to the tall and all Views?**
51. Retrain by clicking the **Train the Tool** button.

Your results should look similar to the one below. All Good images should now be maked Good and all Bad images should be marked as Bad.



Confusion Matrix

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	12	0	0	12
	Bad	0	0	9	9

Appendix C – ViDi Designer Integration

At the end of this lab exercise, Participants will be able to:

- Utilize the ViDi Plug In to display results and tools on the HMI

The Participant will utilize the following ViDi tools to successfully complete this exercise:

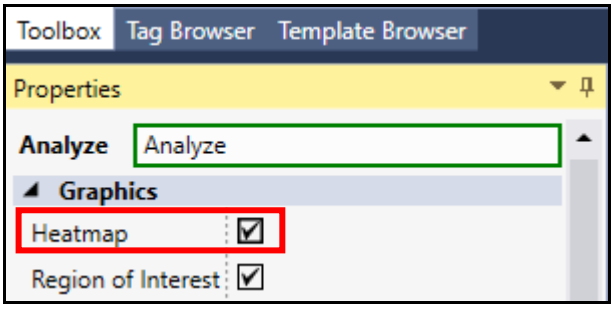
- ViDi View Record Configurator
- VisionPro Display
- ViDi Tool Editor

Watch Faces – Part 2
Displaying Results

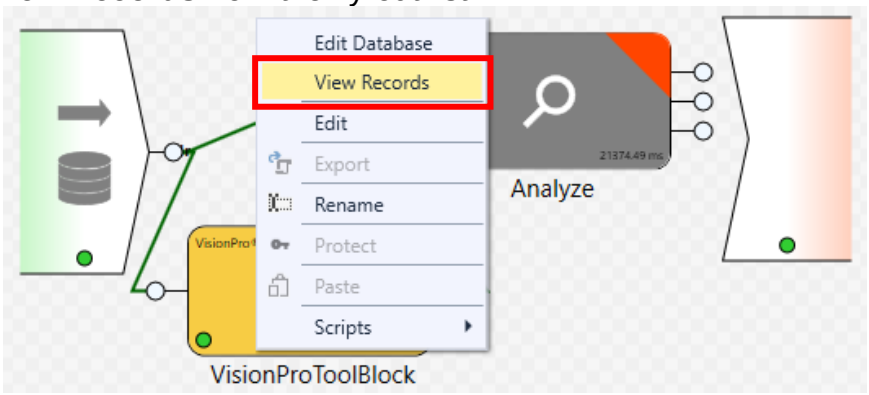
This lab exercise will continue with the watch faces exercises from the previous section. The Designer project will be built on, adding a display for the result image showing the defect on each part. This will be done utilizing the ViDi Record Editor and a VisionPro Display.

Follow the steps below to complete the lab exercise:

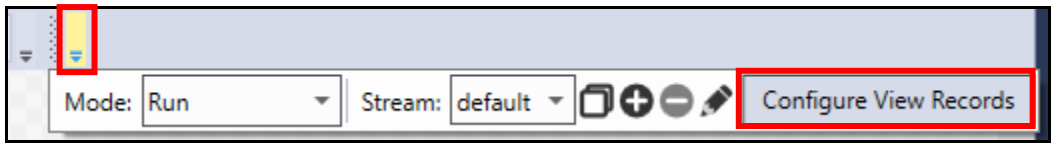
1. Open the Designer project from the previous lab exercise if not open. Navigate to the **ViDi Sub Task** tab.
2. Click the Red Analyze tool and check the **Heatmap** checkbox in the Toolbox.



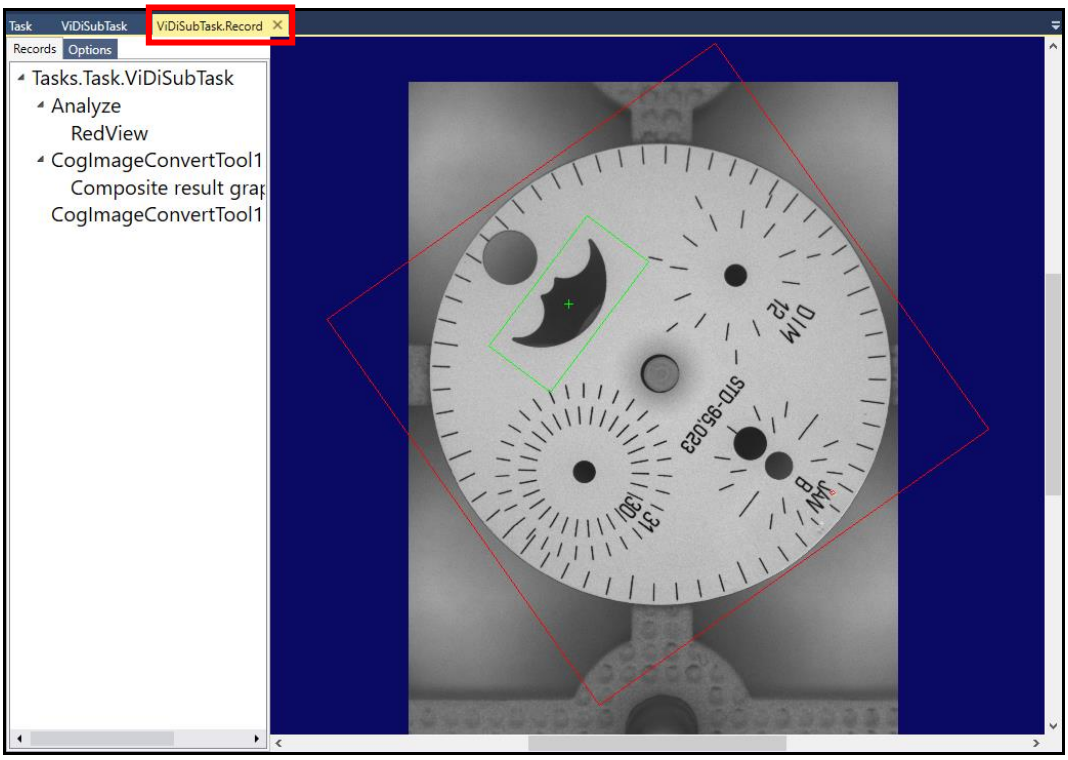
3. To access the View Records Configurator in the ViDi Sub Task, right-click and select **View Records** from the fly-out list.



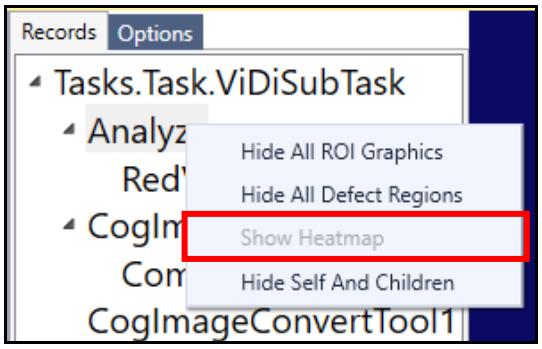
NOTE: *This can also be accessed by clicking the down arrow under the buttons in the ViDi Sub Task and clicking the* ***Configure View Records*** *button.*



The **View Record Configurator** tab opens.



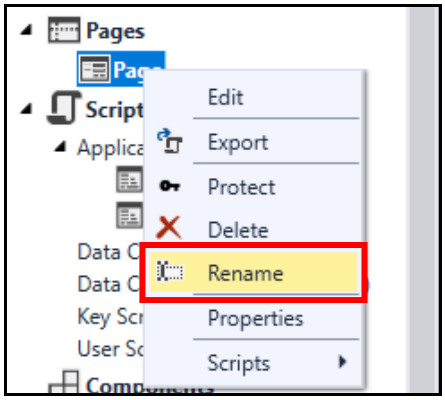
4. Right-click the Analyze tool inside the Record Editor window and enable the Heatmap Record.
   NOTE: *If the result does not have a heatmap, the part has no defects, it is not possible to enable the heatmap record, as shown below.*

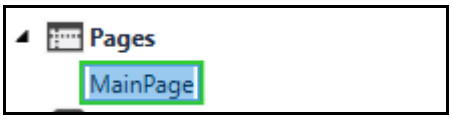


If the heatmap is enabled red dots will appear where the defects are present.

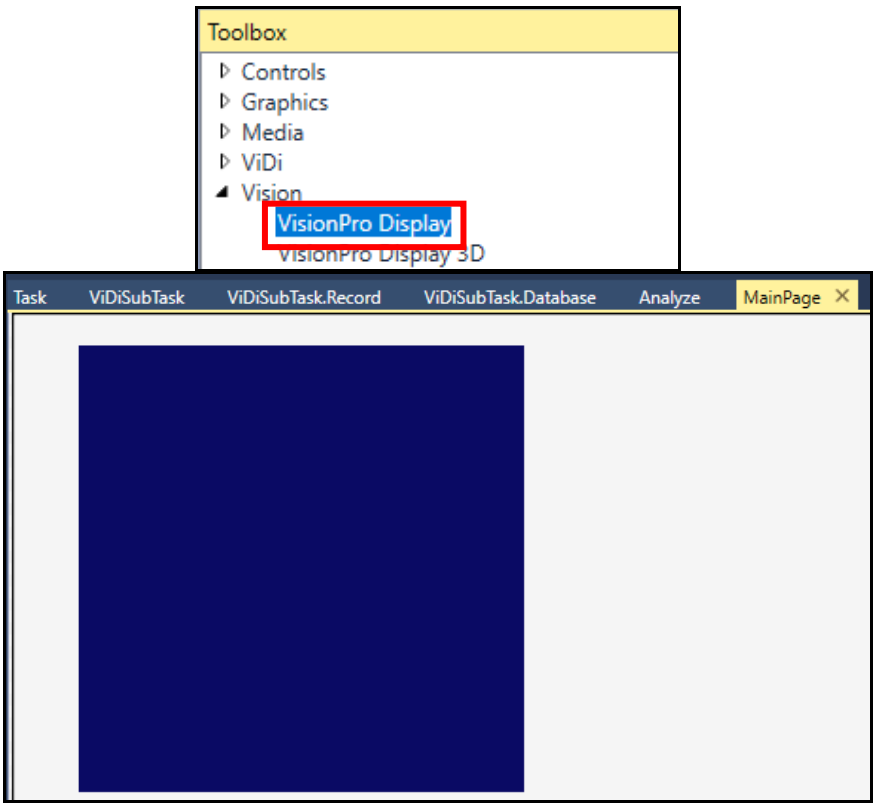
Displaying the Result Image

6. Right-click the default Page and Rename to *MainPage* and press the **<Enter>** key.

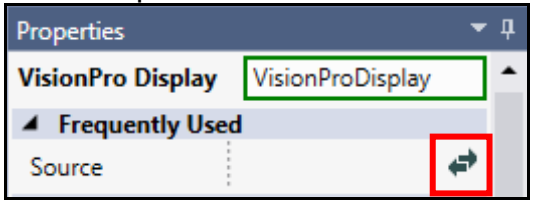




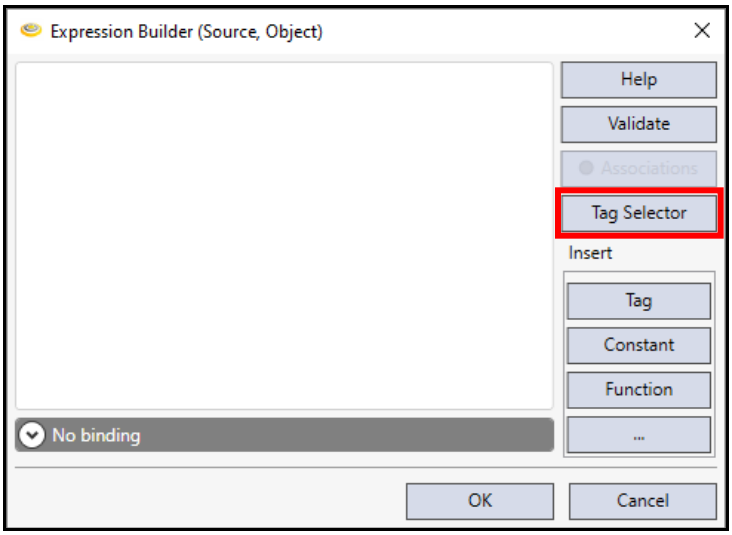
7. Double-click MainPage to open.
8. Add a **VisionPro Display** to the MainPage.

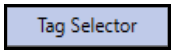
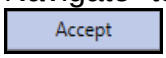


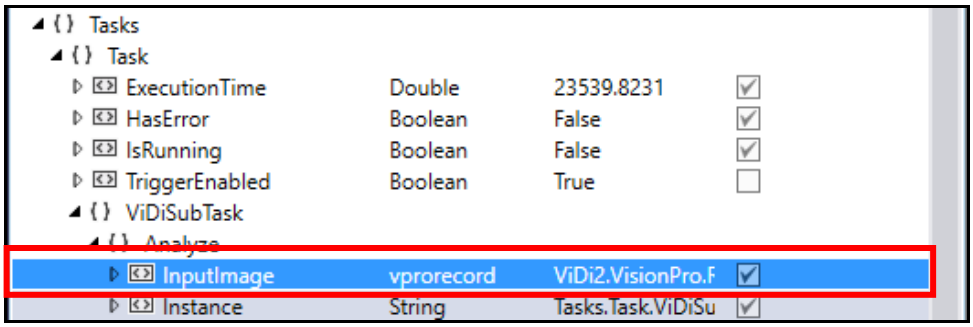
9. Click on the Display to show the **Properties**. The Properties are located on the right-hand side of the Development Environment.



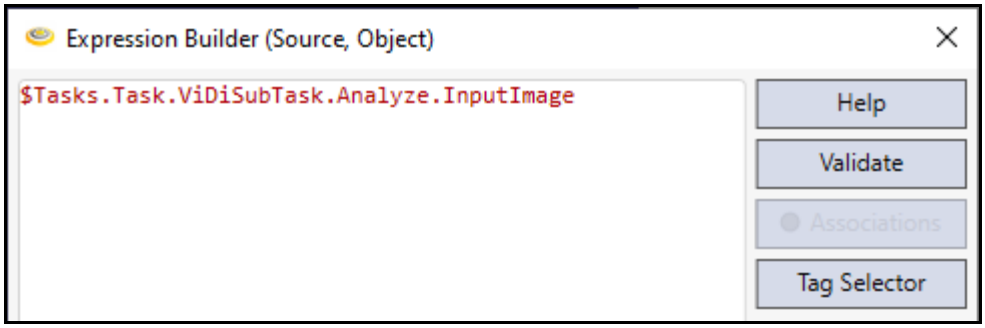
10. Click the double arrows button to set the **Source** property.
    The **Expression Builder** dialog displays.

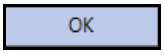


11. Click the **Tag Selector**  button.
    The **Tag Selector** dialog displays.
12. Navigate to *Tasks.Task.ViDiSubTask.Analyze.InputImage* and click the **Accept**  button.

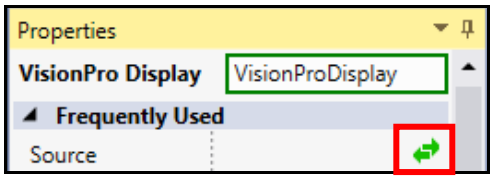


    The **Expression** has been built.



13. Click the **OK**  button.

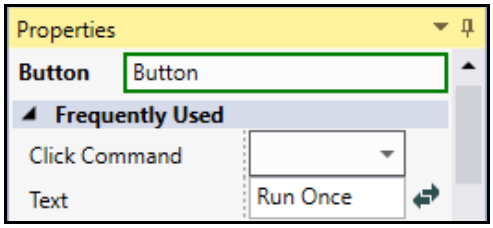
The **Source** icon has changed color from black to green indicating that there is a reference enabled.



14. Add a **Button** to the Page.

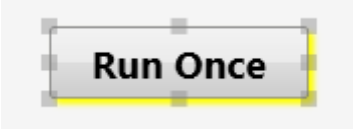
    NOTE: *The Button is found in the Toolbox under Controls*.

15. In the Button properties change the Text to *Run Once*.

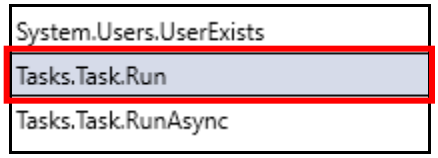


16. In the Button Appearance check the **Bold** checkbox, change the Font Size to **24**, select **DropShadow** as the effect and set the Shadow Color to **Yellow**.

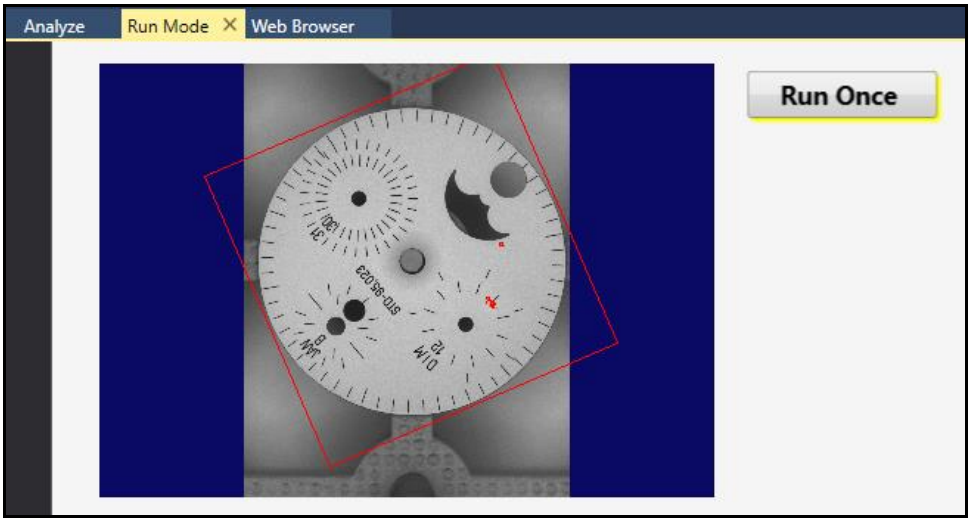
    NOTE: *You may need to increase the size of the text box to see the complete name.*



17. Under the Button Properties set the **Click Command** to **Tasks.Task.Run**.



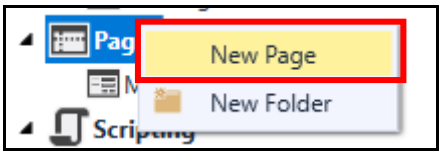
18. Click the **Run** button in the Task bar and click the **Run Mode** tab.



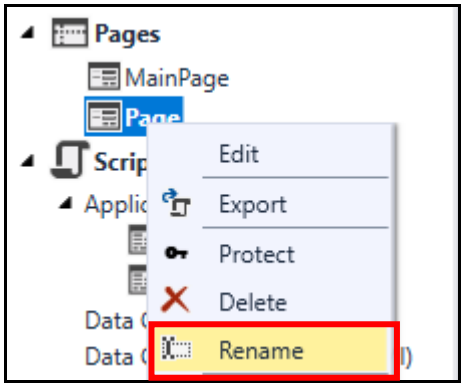
19. Click the **Stop** button in the Task bar.
20. Click the **Save** button in the Designer toolbar to save your job.

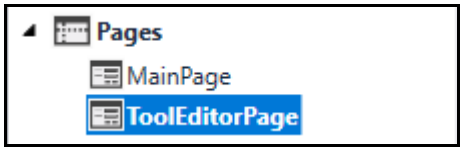
Watch Face – Part 3
Editing the Tool at Run Time

1. Add a New Page by right-clicking on the Pages and selecting **New Page** from the fly-out list.



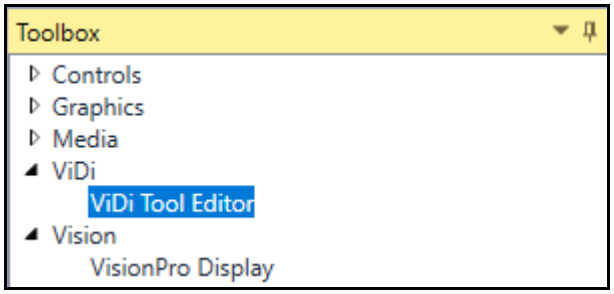
2. Right-click the New Page and Rename to *ToolEditorPage* and press the **<Enter>** key.



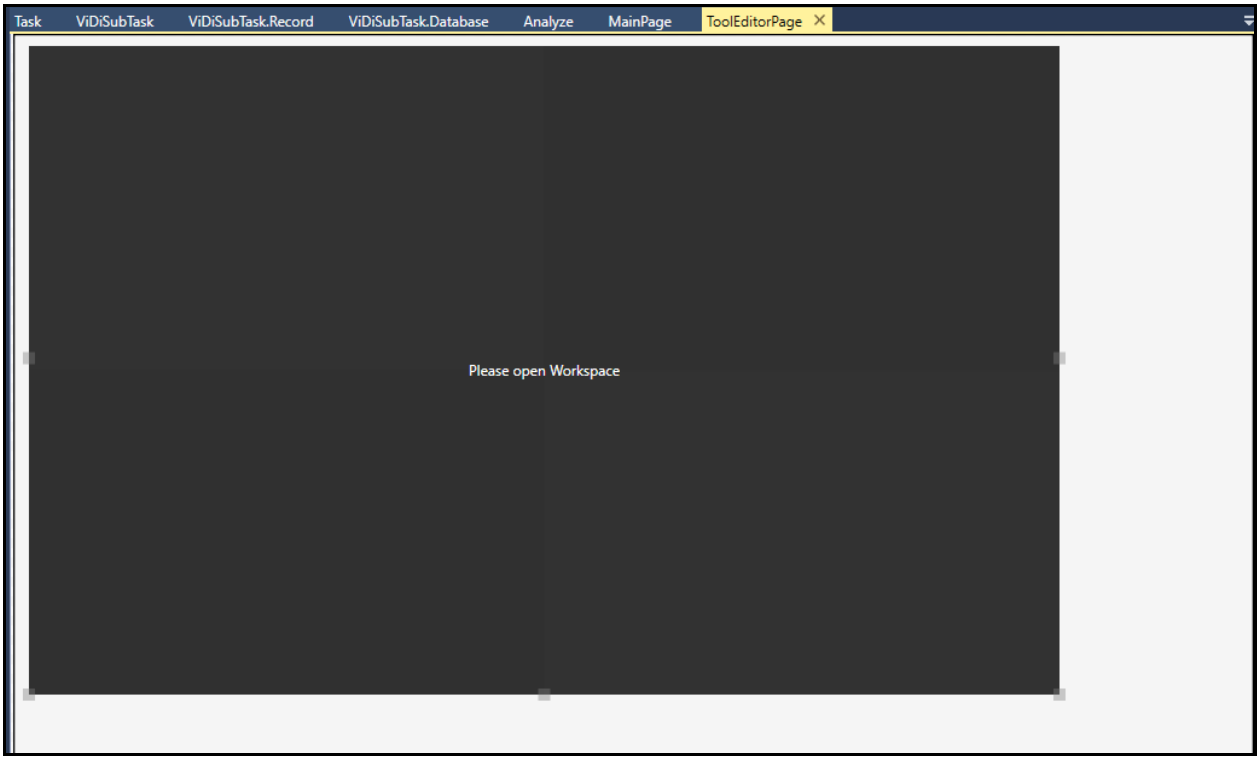


3. Double-click the ToolEditorPage to open.
4. Add a **ViDi Tool Editor** to the ToolEditorPage.

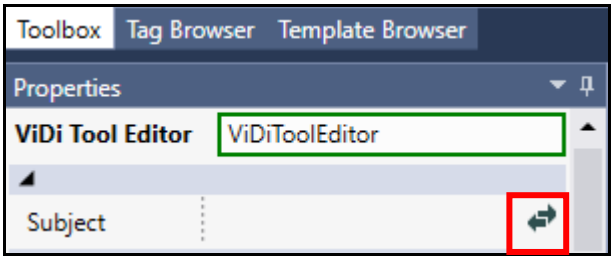
   NOTE: *The ViDi Tool Editor is found in the Toolbox under ViDi.*



5. Increase the size of the ViDi Tool Editor to fill the Page, as shown below.

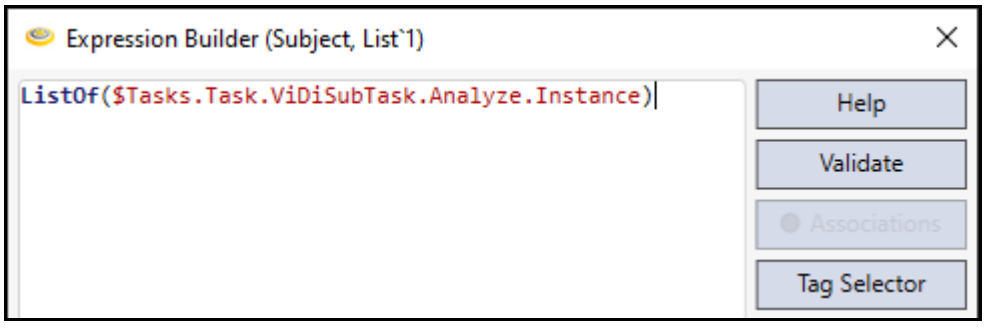


6. Click on the ViDi Tool Editor to show the **Properties**. The Properties are located on the right-hand side of the Development Environment.



The **Expression Builder** dialog displays.

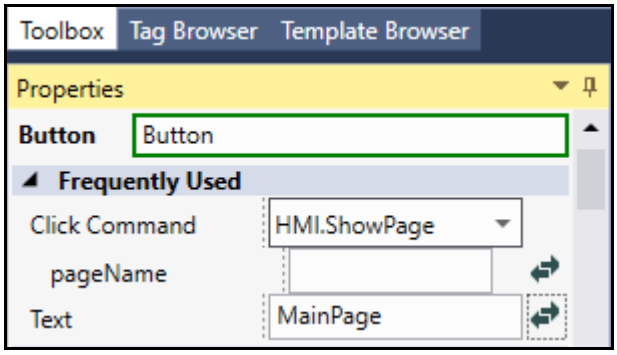
7. For this binding we will utilize a Function (ListOf) and the Instance of the Analyze tool and click the **OK** button.



Navigate Between Pages

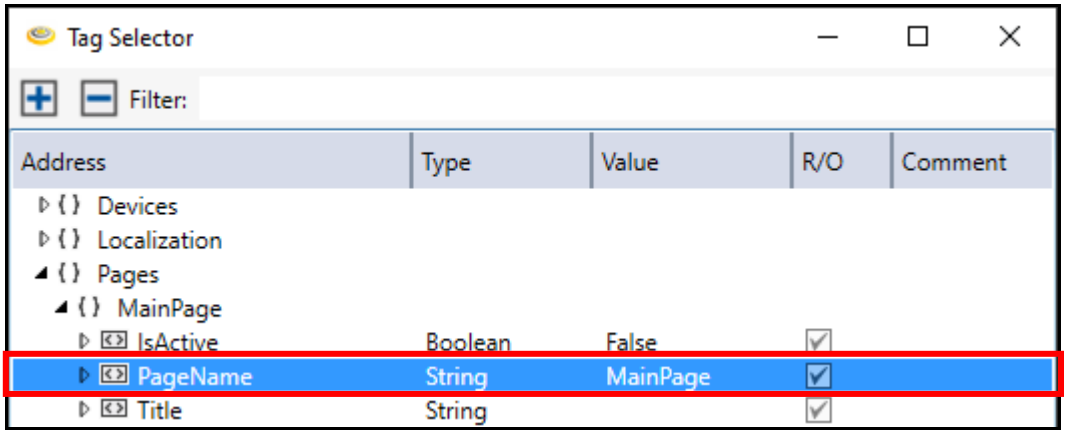
8. Add a **Button** to the ToolEditorPage and change the Text to *MainPage.*
9. Select **HMI.ShowPage** from the Click Command drop-down list.

   A new field named **pageName** displays.

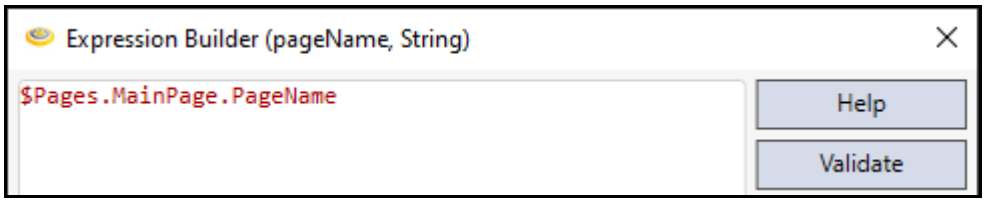


10. Click the double arrow button to set the page binding.

    The **Expression Builder** dialog displays.
11. Browse through the Tags until you find the tag for the page name, as shown below, as click the **Accept** button.

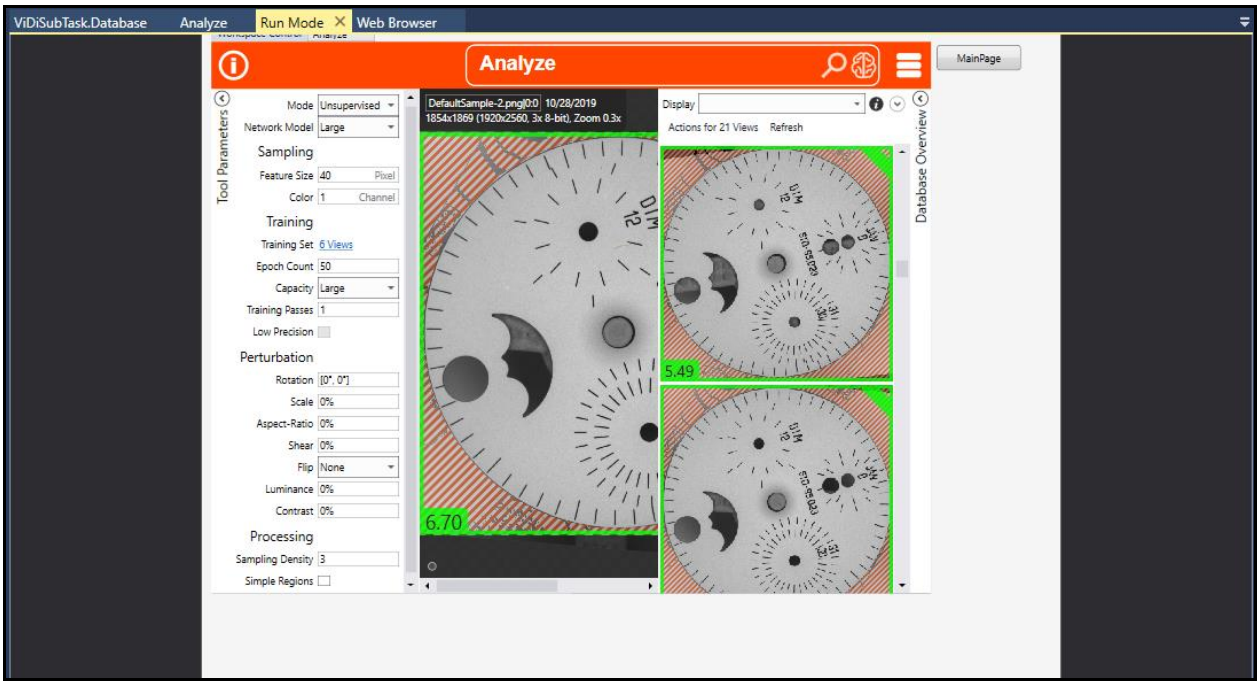


12. The **Expression** is built. Click the **OK** button to accept.



13. Repeat the steps above to create a button on the Main Page that displays the ToolEditorPage when clicked.

14. Click the **Run** button in the Task bar and click the Run Mode tab.



15. Edit the Tool Parameters and retrain the Red-Analyze tool while the project is running.
16. Click the **Stop** button in the Task bar.
17. Click the **Save** button in the Designer toolbar to save your job.