HALCON

a product of MVTec

HDevelop Label Tool

Version 1.2

MVTec Software GmbH

Building Vision for Business

About This HDevelop Label Tool

This tool is only a first prototypic version of a label tool that allows to create ground truth data for deep-learning-based object detection or semantic segmentation. The tool runs in HDevelop. As HDevelop is not built for the purpose of programming GUI-applications, unexpected behavior or failure-cases may occur.

Release Notes

V 1.2 Release notes for Version 1.2, as released in May 2019.
New functionality: The tool supports annotations for `dataset_type` *'detection_rect2'*.

V 1.1 Release notes for Version 1.1, as released in January 2019.
Bug Fixes:

- If the label of an already labeled image is modified, the corresponding sample in the DLDataset might be saved twice and in rare cases an existing sample is overwritten. This problem has been fixed.

More information about HALCON can be found at: `http://www.halcon.com/`

First Steps

Before you start labeling your own data, we encourage you to play around and discover all the functionalities of the label tool. For this you can use our small toy example, which is based on few demonstration images of the HALCON example `’pill_bags’` that are delivered together with the tool.

1. Open `mvtec_hdev_labeltool.hdev` with HDevelop.
2. Choose one of the demonstration parameter files:
    - `pill_bags/params_detection.txt` for a dataset of type *’detection’* as used in object detection with axis-aligned boxes
    - `pill_bags/params_detection_rect2.txt` for a dataset of type *’detection_rect2’* as used in object detection with oriented boxes
    - `pill_bags/params_segmentation.txt` for a dataset of type *’segmentation’* as used in semantic segmentation
3. Run the label tool, e.g., by pressing "F5". Four windows should open: the header window, the main window with the raw image, the preview window next to the main window, the instruction window on the right.

    Use the keyboard to select one of the functionalities. The key for each possible functionality is shown in the instruction window.

    Use the mouse to draw or change annotations. If you are done with drawing this annotation, use a right-click to confirm.

Functionalities

There are two different modes, depending on whether you edit a single annotation or not. In the following, for each of these modes the available options are listed. The keys mentioned in the tables correspond to the default layout, but this can be customized in the procedure `dlt_create_tool_options`.

Selection Mode

Key	Function
▶	Go to next image.
◀	Go to previous image.
J	Jump to an image with a certain index: type the index number and confirm with ENTER.
A	Add or change annotations → change to Editing mode.
P	Copy and paste all annotations from the previously edited image.
X	Unlabel the current image. This means that all annotations are removed from the image and the underlying sample is deleted from the DLDataset.
S	Stop the program. This allows you to modify the program without ending it.
D	Done. The annotations are saved and the program ends.

Table 1: Available commands in the Selection mode.

Editing Mode

Certain commands differ depending on the `dataset_type`.

Key	Function
◀	Select previous annotation.
▶	Select next annotation.
A	After pressing this key, you can select an annotation by a left mouse-click.
N	Add a new annotation. The finished drawing is confirmed with a right-click. The drawing object differs depending on the dataset type: *'detection'* and *'detection_rect2'*: draw the rectangle parallel to the axis *'segmentation'*: draw the polygon
M	Add a new annotation with free orientation. (*'detection_rect2'* only)
O	Flip the direction (but not the box) by 90°. (*'detection_rect2'* only)
P	Draw a polynom, the smallest enclosing oriented rectangle will be determined as annotation. (*'detection_rect2'* only)
E	New elliptic annotation can be added, which will be processed as a polygon. (*'segmentation'* only)
X	Remove the current annotation.
R	Change the annotation region. This differs depending on the dataset type: *'detection'* and *'detection_rect2'*: modify the rectangle *'segmentation'*: modify the polygon
C	Change the annotation class.
S	Stop the program. This allows you to modify the program without ending it.
D	Done. The annotations are registered and you go back to the Selection mode.

Table 2: Available commands in the Editing mode.

Configure Your Own Dataset

Configure the label tool parameters according to your dataset.

All possible parameters with their default values and explanation are shown in the file `labeltool_parameters_template.txt`. We recommend to copy one of the demonstration configurations as a starting point: - `pill_bags/params_detection.txt` for annotations used in object detection - `pill_bags/params_segmentation.txt` for annotations used in semantic segmentation

Please note, paths specified in this parameter-file need to be given with forward slashes. Moreover, paths can be absolute or relative to the location of mvtec_hdev_labeltool.hdev.

Parameters which should be specified:

`dataset_type`: Type of the dataset. The following options are available:

- *'detection'*: The label tool is configured to annotate axis-aligned boxes. The boxes can be overlapping and there is a separate instance for each box.
  The boxes are parametrized in the format of `rectangle1` in HALCON.

- *'detection_rect2'*: The label tool is configured to annotate oriented boxes. The boxes can be overlapping and there is a separate instance for each box.
  The boxes are parametrized in the format of `rectangle2` in HALCON.
- *'segmentation'*: The label tool is configured to annotate pixel-precise regions by annotating polygons.
  During the label-process, you annotate different instances but for each class they are summarized (union) to a single output region.
  All pixels that have not been labeled are set to the background class.

`image_dir`**:** Path where the images that should be labeled are placed.

Please note, you should not change the images inside any subdirectory of `image_dir` during the annotation process. When you add new images, you need to restart the label tool in order to load the new images. If images get replaced, annotations for these images will most likely not fit anymore.

`class_names` **or** `class_names_file`**:** Either specify a tuple containing all class names for `class_names` or a .txt-file that contains all class names (one per row). If both are specified, the names given inside the `class_names_file` are used. In case of segmentation, please note that the background class name should not be part of the list, as it is set automatically. Instead, the background class name can be set using the parameter `background_class_name`, see below.

`default_class_name`**:** The class that is used, if you draw a new region. This class has to be one of the classes that are specified by `class_names` or inside `class_names_file`.

`dl_dataset_file`**:** The output file where the annotations should be stored in the DLDataset format (.hdict).

`background_class_name`**:** (for segmentation only) Class name of the background pixels. When the dataset is saved, all pixels that are not assigned to another class will be set to this class. This class has to be different from the classes that are specified by `class_names` or inside the `class_names_file`.

Optional parameters for segmentation only:

`polygon_tolerance`**:** Set the maximal tolerance/distance the polygon is allowed to have from the region contour when the polygon is created to approximate the run length encoding in order to used the draw-functionalities. This is important if, e.g., an elliptic region is approximated by a polygon drawing object when applying "change the annotation region" (pressing "r"). Please note, if you use a `polygon_tolerance` > 0 and apply "change the annotation region" (pressing "r") without modifications, the region might get deformed due to the internal approximations.

`segmentation_dir`**:** Path where the segmentation images will be placed.

`set_background_id`**:** Class ID, which all unlabeled pixels will obtain (background).

Optional parameters for the visualization of all types:

`zoom_factor`**:** Changes the resolution when creating the sampling points such that fine regions can be drawn more easily. A higher `zoom_factor` leads to more sampling points.

`image_width`**:** Maximal width of the displayed images. The displayed images will keep the ratio of the raw image.

`image_height`**:** Maximal height of the displayed images. The displayed images will keep the ratio of the raw image.

`image_window_size_factor`**:** Set the Window size factor depending on your screen-resolution.

Configure the Label Tool with an Existing `DLDataset`

In case the dataset file `dl_dataset_file` already exists, the following parameters are taken from the dictionary `DLDataset` and are ignored in the parameter-file (and in the .txt-file containing the class names):

- `dataset_type`
- `class_names` / `class_name_file`
- `set_background_id` (segmentation)

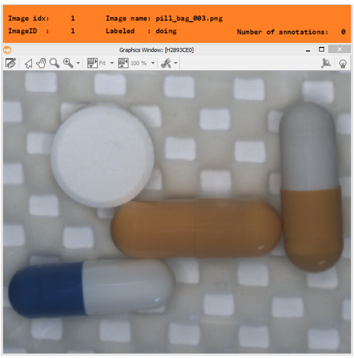
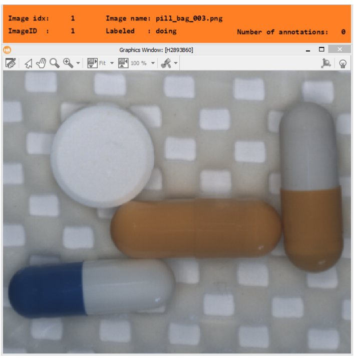
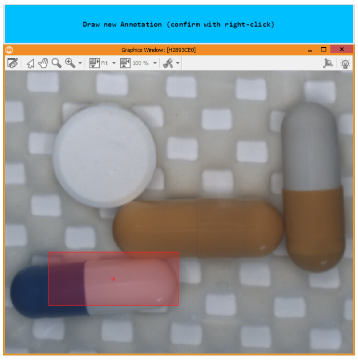
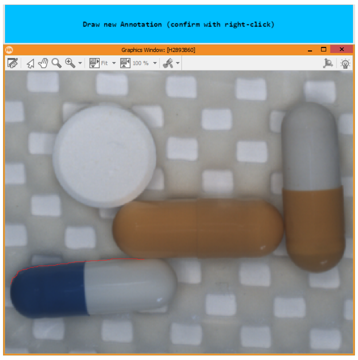
Instead, `class_names` and the class IDs are obtained from the existing `DLDataset`. In case of segmentation, the background ID is the class ID of the background class name, therefore, it is crucial that the parameter `background_class_name` is part of the list of class names in `DLDataset`.

In case you need to add a class to a `DLDataset`, you have already started labeling, you need to do this in the dictionary `DLDataset` (saved under `dl_dataset_file`). Add the class name to the key *'class_names'* and a corresponding class id to *'class_ids'* using the operators `get_dict_tuple` and `set_dict_tuple` and save the changed dictionary to the path given through `dl_dataset_file`. Please note, that it is not recommended to remove entries from `class_names` or `class_ids` once you have started labeling.

Important Note for Linux Users

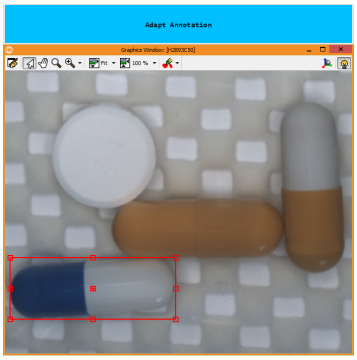
In order to use the keyboard commands, you have to place your mouse over the main window.

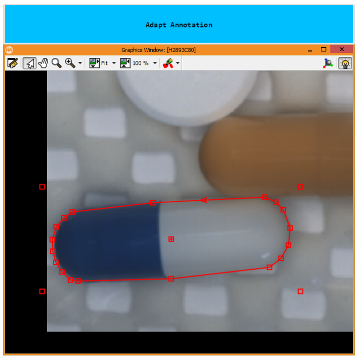
Walk Through First Annotations

Key	Object detection	Semantic Segmentation
Enter Editing mode:		
A	 Image idx: 1 Image name: pill_bag_003.png ImageID : 1 Labeled : doing Number of annotations: 0 Graphics Window: [H2893CE0]	 Image idx: 1 Image name: pill_bag_003.png ImageID : 1 Labeled : doing Number of annotations: 0 Graphics Window: [H2893B60]
Add a new annotation:		
N	 Draw new Annotation (confirm with right-click) Graphics Window: [H2893CE0]	 Draw new Annotation (confirm with right-click) Graphics Window: [H2893B60]

Modify the region of your annotation:

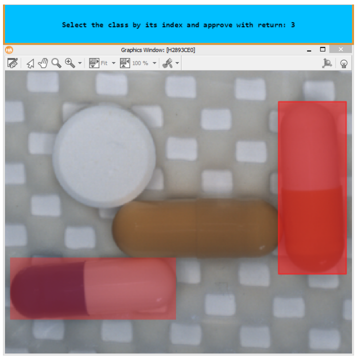
R

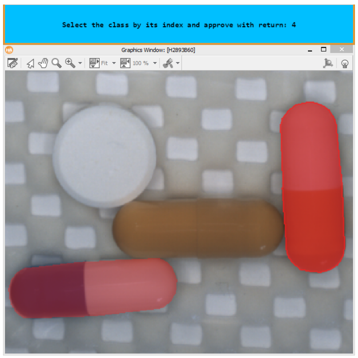

Adapt Annotation


Adapt Annotation

Change the class of your annotation:

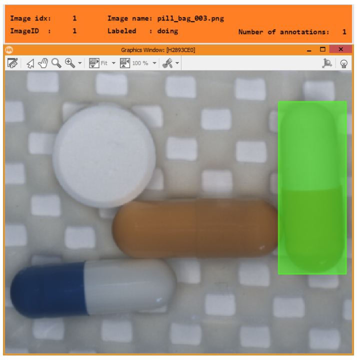
C

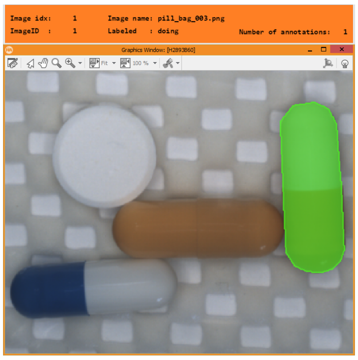

Select the class by its index and approve with return: 3


Select the class by its index and approve with return: 4

Delete the selected annotation:

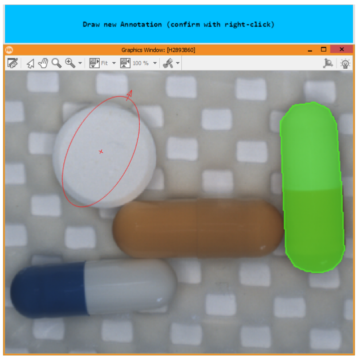
X


```
Image idx:     1      Image name: pill_bag_003.png
ImageID  :     1      Labeled    : doing          Number of annotations:   1
```


```
Image idx:     1      Image name: pill_bag_003.png
ImageID  :     1      Labeled    : doing          Number of annotations:   1
```

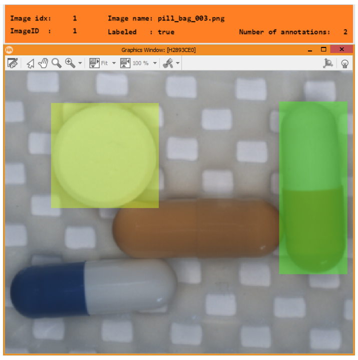
Add elliptic annotation (only available for semantic segmentation):

E

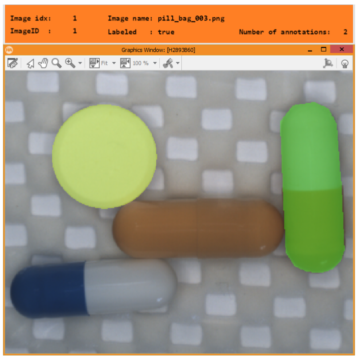

Draw new Annotation (confirm with right-click)

Leave Editing mode and go back to Selection mode:

D



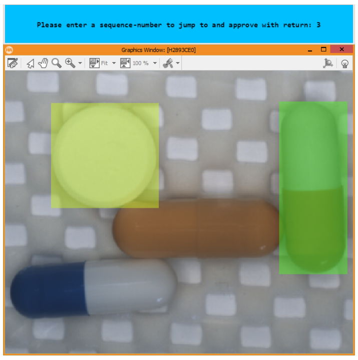
```
Image idx:     1       Image name: pill_bag_003.png
ImageID  :     1       Labeled    : true                Number of annotations:   2
```



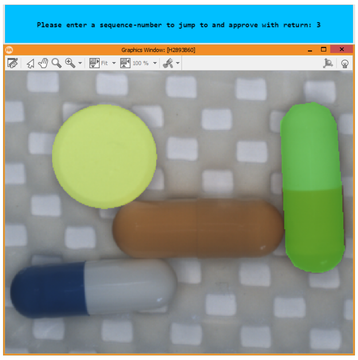
```
Image idx:     1       Image name: pill_bag_003.png
ImageID  :     1       Labeled    : true                Number of annotations:   2
```

Jump to image with specific index:

J



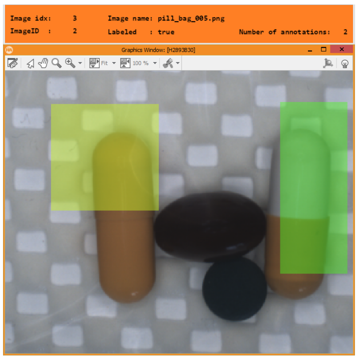
```
Please enter a sequence-number to jump to and approve with return: 3
```



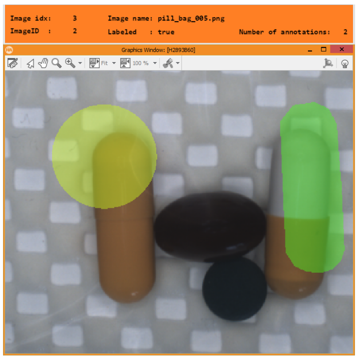
```
Please enter a sequence-number to jump to and approve with return: 3
```

Paste latest annotations into selected image:

P



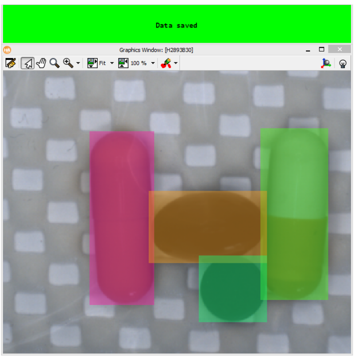
```
Image idx:     3       Image name: pill_bag_005.png
ImageID  :     2       Labeled    : true                Number of annotations:   2
```



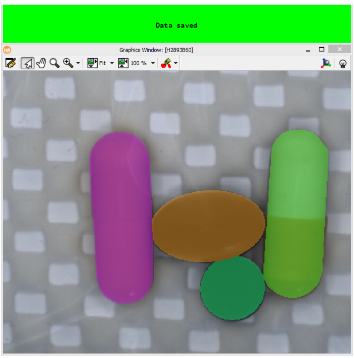
```
Image idx:     3       Image name: pill_bag_005.png
ImageID  :     2       Labeled    : true                Number of annotations:   2
```

Save your data after you finished your annotations:

D



```
Data saved
```



```
Data saved
```