

Pixie - User Manual

Contents

Pixie - User Manual	1
About.....	2
Introduction.....	2
Configurations.....	3
First start up	3
General configurations.....	4
Image export	4
Attributes definition.....	6
Object Attributes Definition	6
Frame/Scene Attributes Definition	7
Image/Frame annotation	8
Scene labeling	8
Bounding Box Labeling.....	8
Polygon Labeling.....	10
Semantic Segmentation.....	11
Basic usage.....	11
Advanced usages.....	13
Annexes.....	17
GUI Description.....	17
Edit Bounding Box object.....	17
Hotkeys	18
Windows DPI configuration	18

About

Pixie might be a "yet another" labeling tool, but our goal is to provide a labeling application feature rich so that the labeling work will be easier, faster and hopefully it will provide a better ground truth data. Before starting to implement Pixie, we have spent more than 6 months to study what labeling work really means, which are the similarities/differences across various industries/applications, how can we improve the labeling process and maybe most important factor, how can we reduce the human error during labeling process. These are the questions which represent the basis of our labeling application, Pixie. One other fundamental brick is the **flexibility**, Pixie must be a cross platform application and it must be easily configurable to adapt to new requirements.

Introduction

In the past years, the need for ground truth data boomed especially thanks to the progress of the Machine learning algorithms. If in the not so far past, the machine learning algorithms were mostly based on bounding box data, in the last few years the data has evolved to 3D bounding boxes, polygon data, semantic segmentation, various sensor fusion data, etc. Pixie is trying to provide data labeled using the following methods:

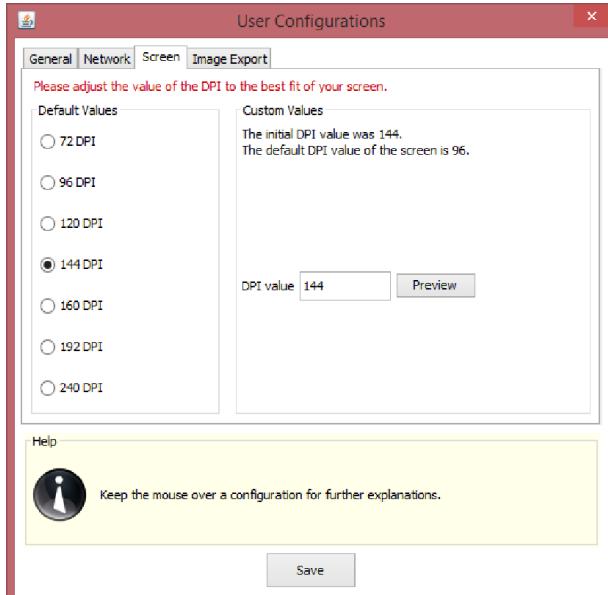
- Bounding box
- Polygon
- Semantic segmentation
- Free hand drawing - coming soon
- Scene/Frame understanding

The output data/ground truth is saved in various formats (json, binary maps, images, etc.) depending by the labeling type and it can be easily extended to match any new requirement.

Configurations

First start up

In the first start up, the application will pop-up a file chooser window in which the user can select an image. Once the image is selected, a new user configuration window will pop-up.



Using this screen the user can configure his preferred DPI size. This step is necessary for Windows (8.1, 10) and Linux operating systems which use a different DPI than the standard one. In Windows 8.1 this can be adjusted using the [slider](#).

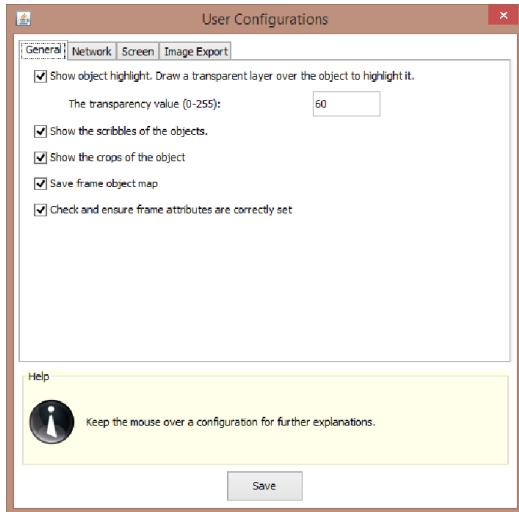
The user can select one of the predefined values, the radio buttons from the **Default Values** panel, or he can specify a different value using the **DPI value** text box. Once a selection has been made, the user can press the **Preview** button; once the button is pressed, the Pixie's GUI will be adjusted accordingly. At this step, the user shall try to choose a value which will stretch/enlarge the GUI at the maximum.

Caution: If the user chooses a greater value than the optimal one, the previously selected image will not be correctly displayed; the image will look like being cropped. In this case, the user has to choose a different DPI value.

During the first start up, we strongly encourage the user to define the [Frame and Object attributes](#).

General configurations

The following general configurations are available

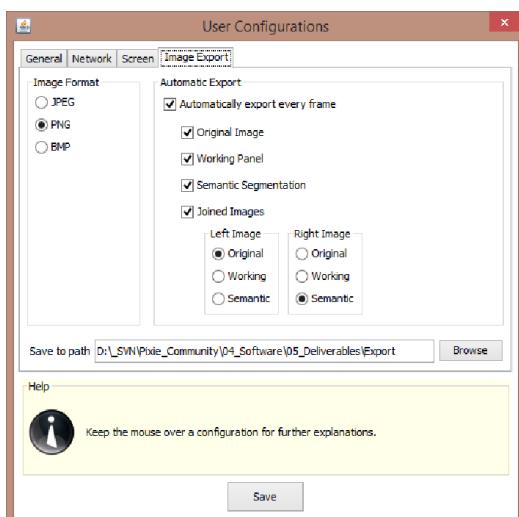


Each of these options is further explained while keeping the mouse on top of an option.

Image export

The image export panel provides the option to export the user's work:

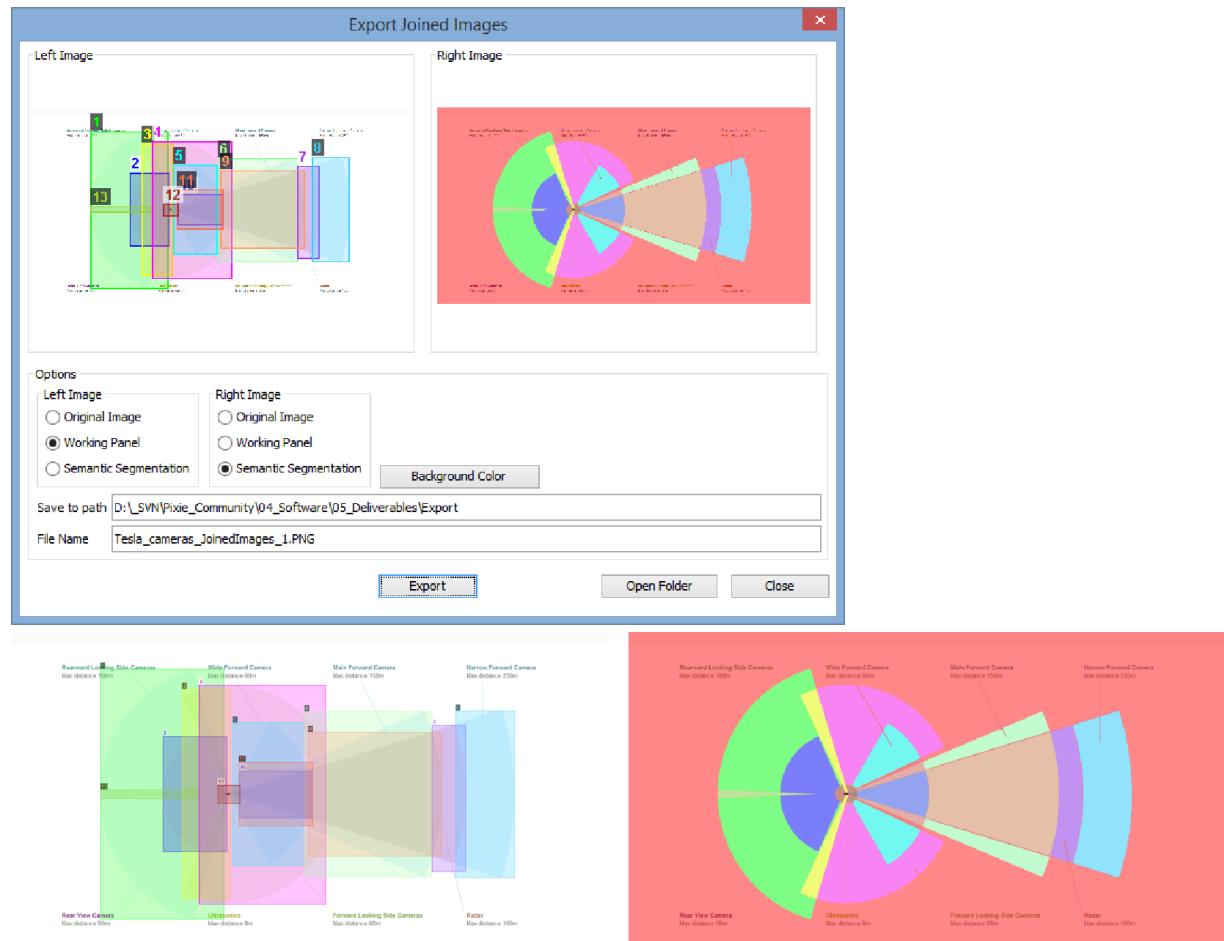
- export original image
- export working panel
- export semantic segmentation result
- export a *joined images* (original + semantic, original + working panel, working panel + semantic and so on).



If the **Automatically export every frame checkbox** is marked, then based on the user's selection the images will be automatically exported when [Next button in the navigation panel](#) is pressed. The files will be saved in the user defined **Save to path**.

Alternatively, the user can manually save the desired images using the Exports menu. The images will be saved in the **Save to path** file location.

Pressing the Exports → Joined Images, a new window will pop-up for configuring the *joined image*. The user can press click in any of the images (left or right) and they will open in a zoomed window. Once the user selects his preferred export configuration (e.g. working + semantic), he can press **Export** button and the image will be saved in the **Save to path** location.



Attributes definition

Pixie provides 2 distinct attribute types:

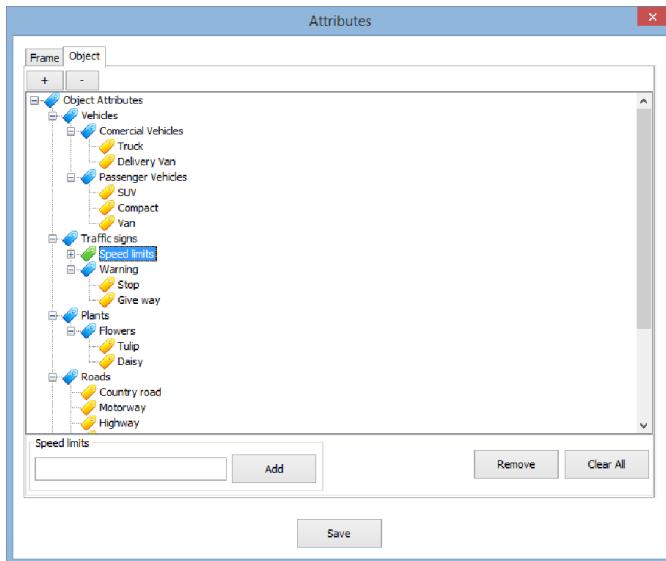
- object attributes
- frame/scene attributes

Object Attributes Definition

One of the most important things during the labeling work is the object's attributes assignation. Since Pixie tries to cover any labeling scenario (automotive, aerial, biology, etc.) we have tried to organize and simplify the object's attributes definition in 3 distinct categories: type, class and value.

Industry/Application field	Type	Class	Value
Automotive	Vehicles	Passenger cars	SUV, compact, van
Automotive	Vehicles	Public transportation	Bus, tram,
Automotive	Vehicles	Commercial vehicles	Trucks, delivery van, refrigerated truck
Automotive	Vehicles	Heavy machinery	Bulldozer, excavator, vibratory compactor
Automotive	People	Pedestrians	Adults, children, handicapped person, bicyclists
Automotive	Traffic signs	Speed limit	10, 20, 30, 40
Automotive	Traffic signs	Warning	Stop, give way, tram crossing
Automotive/Biology	Animals	Small sized	Cats, dogs, rabbits
Automotive/Biology	Animals	Medium sized	Sheep, goats, reindeer
Automotive/Biology	Animals	Large sized	Cows, horses, elks
Biology	Vegetation	Flowers	Daffodil, daisy, tulip, snow drop
Biology	Vegetation	Trees	Oak, apple, olive, palm
Biology	Vegetation	Fruits	apples, oranges, bananas

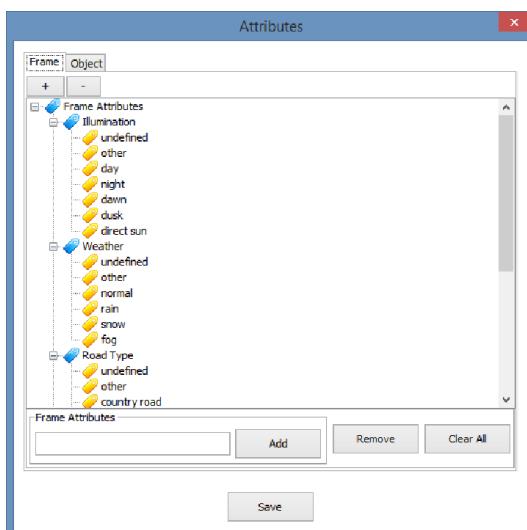
The objects attributes definition is done using the configuration window Options → Attributes Definition → Object.



For adding a new attribute, the user has to select one category and then just write the new attribute in the text area located in the lower part of the window and press **Add** button or press **Enter**. In the image above, the Speed limits has been selected, hence writing a value into the text area will add the new attribute in the Speed limits category.

Frame/Scene Attributes Definition

Currently the frame/scene attributes are implemented for automotive industry and they can be defined similarly to [object attributes](#).



Image/Frame annotation

Pixie provides the following options to label images:

- Scene labeling
- Bounding Box
- Polygon
- Semantic Segmentation
- Free hand drawing - Coming soon; hopefully in the next release version

Scene labeling

The user can assign some extra information related to the scene e.g.

Scene attributes	Description
Illumination	Sunny, Cloudy, Day, Night, Dawn, etc
Weather	Rainy, Snowy, Foggy/Misty
Country	list of countries
Lens status	Clean/Dirty
Image status	Distorted(Fish eye)

This extra information is purely optional but it might be helpful for some particular labeling scenarios.

Bounding Box Labeling

By default, Pixie starts preselecting the Bounding Box mode. In case the Bounding Box mode is not selected, the user can manually choose it from the **Label Type** panel.



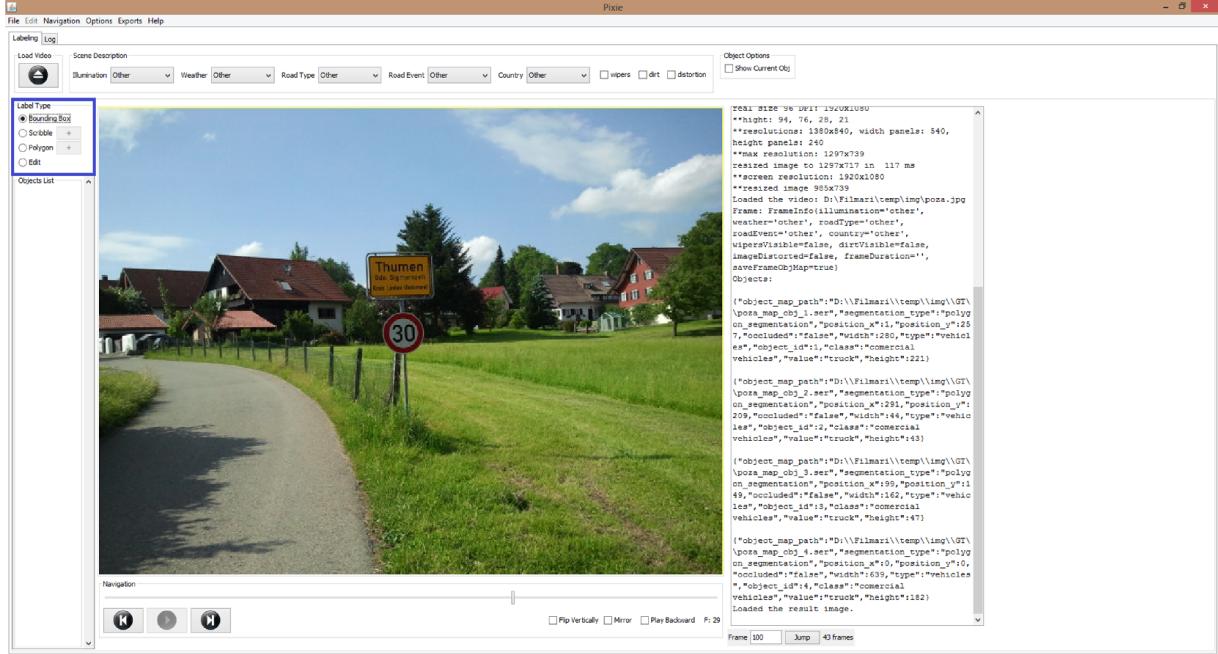
Once the Bounding Box mode is selected, the user can start drawing Bounding Boxes on the [Working panel](#). After a Bounding Box has been drawn, a new window will pop-up for further object edits. The following actions are available in the [Edit Object window](#):

Action	Description
Assign object's attributes	Assign the Type, Class, Value and occlusion properties for the current object
Edit Object	<p>Using the Edit menu, the user can further improve the BBox:</p> <ul style="list-style-type: none"> ▪ resize the BBox ▪ move the BBox ▪ delete BBox - this action will delete the current BBox and it will close the edit window. The deletion of an object is an irreversible operation.
Obj. Color	It will open a color palette for choosing a new color for the current object
Highlight	It will highlight the original image. This operation doesn't alter the original image.
Restore Pos	It reverts all the resize/move actions.

The [Edit Object window](#) opens in max zoomed mode for letting the user to perform a first quality check of BBox's position. At max zoomed size, it shall be easily visible if the BBox is drawn at pixel level precision. The user can zoom in/out using the mouse scroll. Once a new zoom in/out level has been set it will be saved as object preference and next time when the [Edit Object window](#) is opened for that object it will use the user's zooming level. The BBox can be adjusted using the options from Edit menu or the [keyboard shortcuts](#). Once the BBox position has been properly set, the user can go on by setting the object attributes and finally save the object.

Polygon Labeling

The Polygon labeling is initiated by selecting the **Polygon** followed by pressing the + symbol in **Label Type** panel.



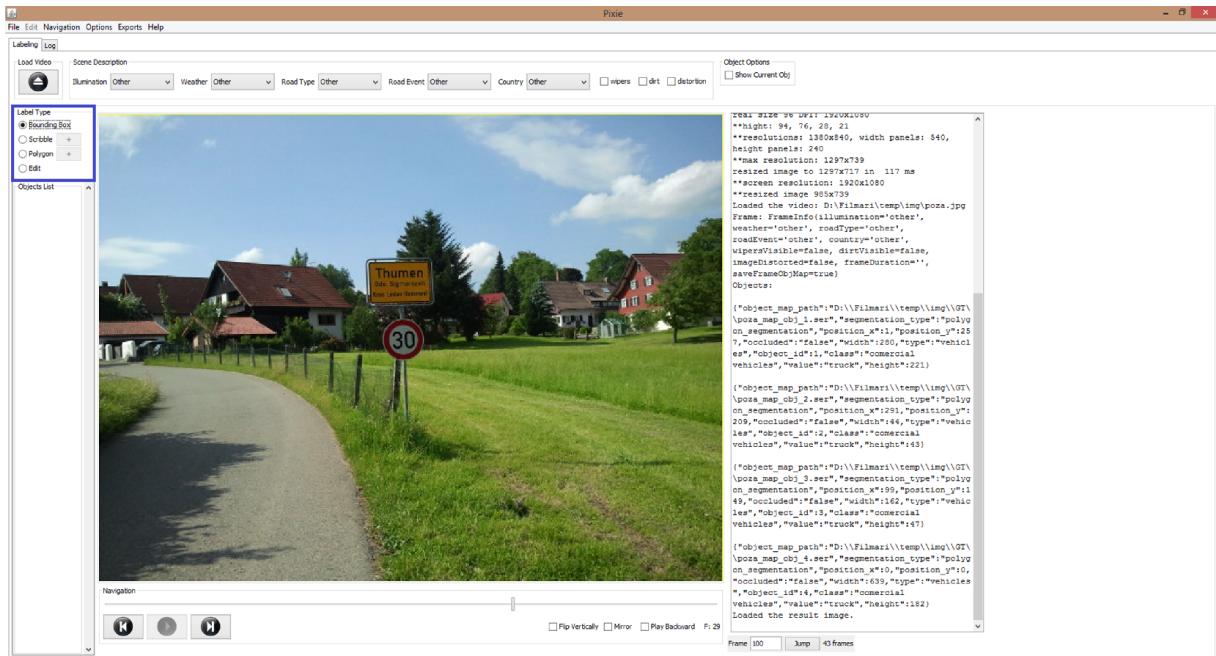
After the previously mentioned actions have been performed, the user can start drawing polygons in the [Working panel](#). Once the user has finished drawing the polygon, he has to save it by pressing the S button in **Label Type** panel. At this point the [Edit Object window](#) will pop-up letting the user to set the object attributes.

Important: At the date of writing the document, 10.10.2017, the polygon's position/coordinates cannot be adjusted; this feature will be implemented in the next release

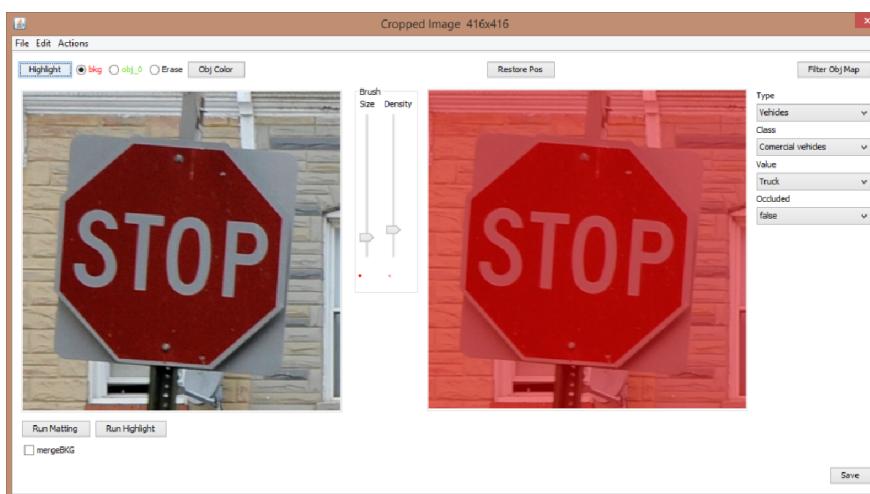
Semantic Segmentation

Basic usage

The Semantic Segmentation labeling is initiated by selecting the **Scribble** followed by pressing the + symbol in **Label Type** panel.

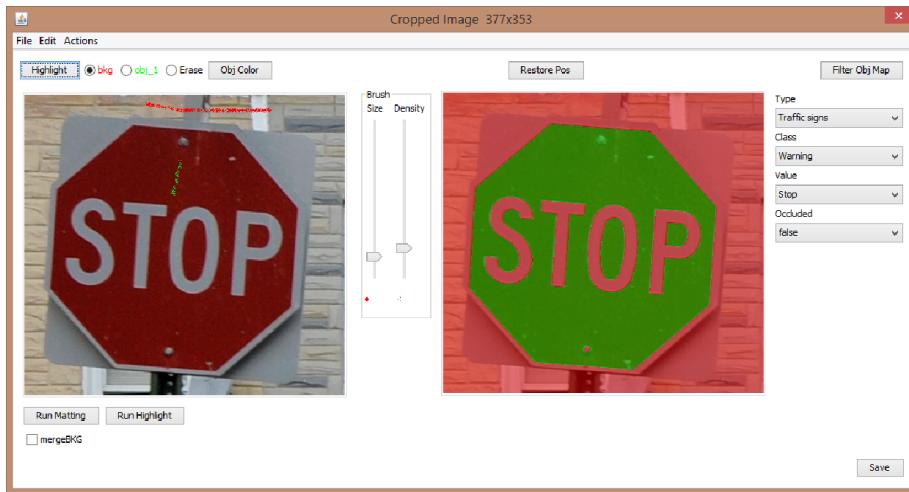


After pressing the + button, the user can start drawing a first BBox on the [Working panel](#). This BBox crops the area marked by the user and opens a new window for performing the semantic segmentation.

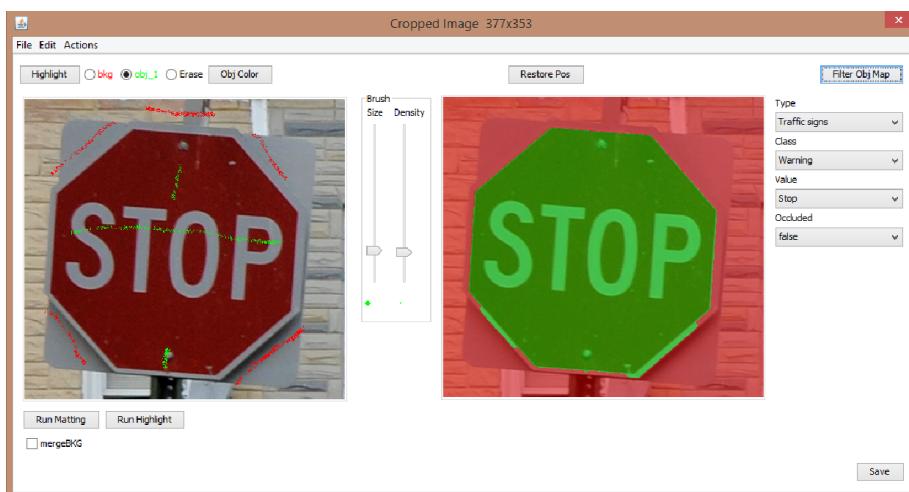


The left image represents the user's cropped area while the right side image shows the output of the semantic segmentation algorithm. The cropped image can be adjusted (increase/decrease size, move the crop) using the options from Edit menu or using the [keyboard shortcuts](#).

The semantic segmentation starts by drawing some scribbles on the left side image. By default, the red radio button **bkg** is selected; this is used for marking the background information (not relevant for object) while the green radio button **obj** is used for marking the object.



As can be seen in above image, just drawing 2 scribbles for delimiting each object we have generated a first semantic segmentation result. If the quality is not what the user expected (which is not in this case), he can continue drawing some further scribbles in order to "teach" the semantic segmentation algorithms that it is not ready yet.



In the second step, some further scribbles have been drawn for background but also for object in order to create a better delimitation of them. As can be seen in above picture, the stop traffic sign is almost perfect. If the user is not satisfied with the current output, he can continue drawing further scribbles for performing even a better delimitation. In case the result's quality is good

enough, the user can continue by selecting the object's attributes and then save the object. After the object is saved, the application will return to main window. In the [semantic preview panel](#), the user can visualize all the segmented objects.

Important: At the date of writing the document, 10.10.2017, the zooming is not implemented for this type of labeling; this feature will be implemented in the next release

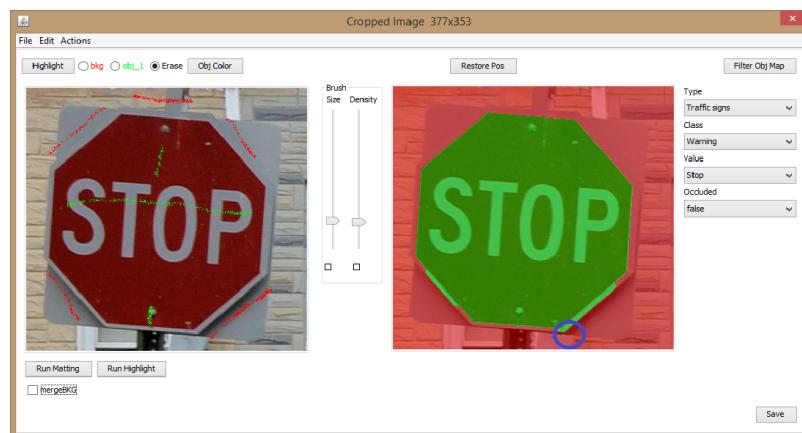
Hint: while in cropped image window, pressing a click on the right side image, a new window will pop-up in which the semantic segmentation result is max zoomed out. This shall help the user to decide if the generated ground truth fulfils the requirements.

Hint: pressing a click on [semantic preview panel](#), a zoomed window will pop-up for visualizing the semantic segmentation result.

Advanced usages

Filtering

Sometimes the algorithm provides in its output some sparse pixels which do not belong the object (they sparse pixels are marked with **blue circle** below)

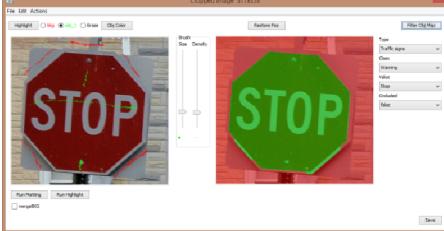
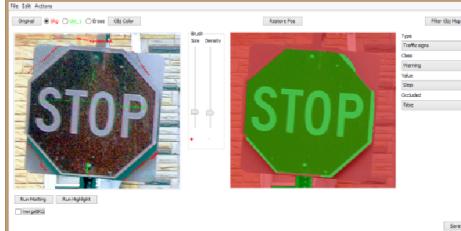
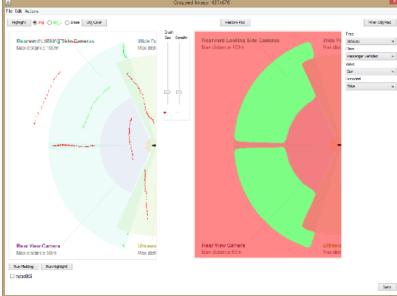
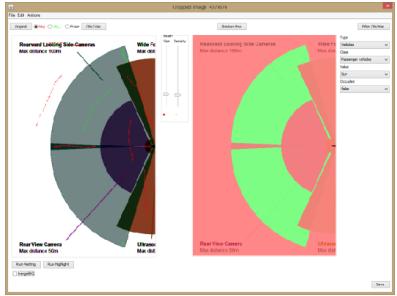
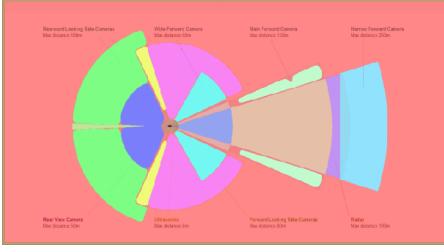
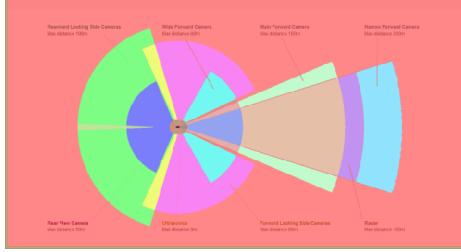


Using the **Filter Obj Map** button, these pixels are removed. This button can be pressed several times till all the undesired pixels are removed. The filtering functionality is provided in the semantic segmentation result zoomed window too.



Highlight

Running the pixel segmentation algorithm on enhanced image (Run Highlight) is especially useful for digital images (computer generated images). For comparisons reasons here are some results on real world image and computer generated image:

Image type	Original image	Highlighted image
Real world image		
Digitally generated image (object level)		
Digitally generated image (image level)		

Brush properties

The brush is the main tool for drawing scribbles; hence the size and the point's density are their main properties. The brush can have a size between 1x1 and 32x32 pixels and can be set by dragging the **Size slider** located in the **Brush panel**. The pixel density slider is used for generating the amount of points which have to be drawn and it is defined as an interval between 0.1 to 1.0, where 1.0 is generating/drawing all the pixels.

Very important: the semantic segmentation algorithm's runtime is highly dependent by the user crop's resolution and number of drawn points. We recommend using a low pixel density and a small brush size.

Semantic segmentation using graphical tablet

The semantic segmentation can be done using a graphical tablet too. Pixie was tested/validated using the **UGEE M708** graphical tablet but some other graphical tablets might work too.

Pros and Cons graphical tablet vs mouse

Pros	Cons
better drawing accuracy	for new users, the graphical tablet might not be easy to use
the brush's size can be dynamically changed based on pressure levels of the pen(if the graphical tablet/pen supports/provides pressure level information)	due to limited resources, we couldn't validate a large broad of graphical tablets, hence some tablets might not be supported.
for experienced users, the semantic segmentation labeling using a graphical tablet is done much faster	
if the pen/tablet has buttons, they can be mapped to the defined hotkeys so that the semantic segmentation labeling can be performed faster	

Multi crop object segmentation

Sometimes it might happen that an object cannot be segmented using only one crop. In this case, the segmentation can be done in several steps. The multi crop semantic segmentation can be done as follows:

- select the semantic segmentation labeling mode
- press the + button
- draw a crop, draw the scribbles, run the semantic segmentation algorithm, add object attributes and save the crop
- draw one more crop, draw the scribbles, run the semantic segmentation algorithm and save the crop
- ...
- once you finished to draw crops press the S button to save the object.

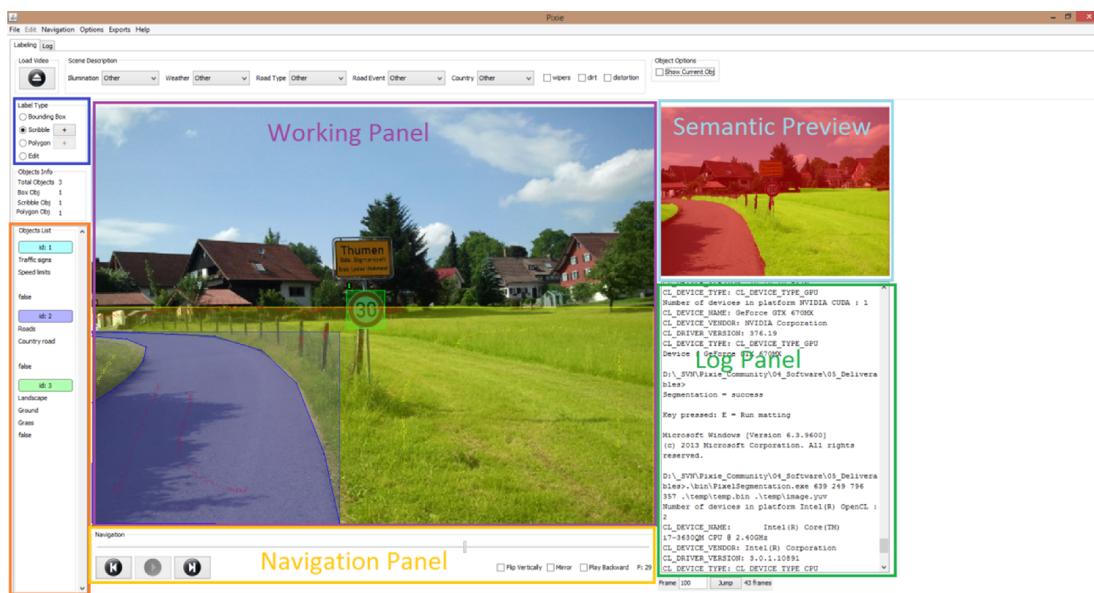
For demonstrating this functionality, we'll label the [Stop traffic sign](#) using 2 crops



In multi crop semantic segmentation labeling, the user has to select the object attributes only for the first crop. All the other crops will inherit them. If the user changes the object attributes in second crop (or later), they will be applied recursively to all crops and to final object too.

Annexes

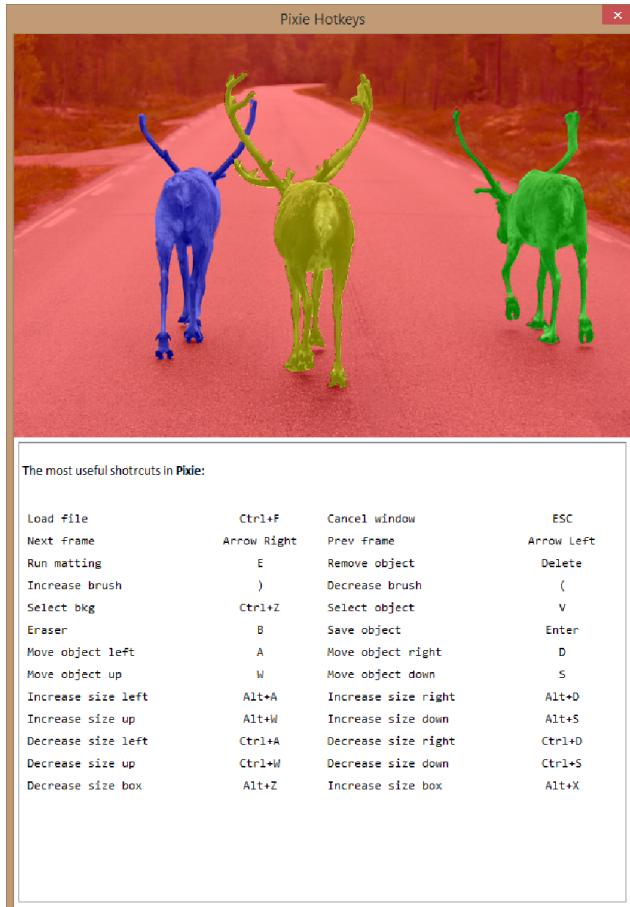
GUI Description



Edit Bounding Box object



Hotkeys



Windows DPI configuration

