Muhammet Bastan, IUPR, August 2011

XRanT: X-Ray Annotation Tool

XRanT is an image and region-level manual annotation tool. You open a directory of images, select regions/object for each image by painting on them and the tool saves each object mask as a separate *png* image; it also saves an annotation list file, containing the list of images and the MBRs of selected objects, etc.

The tool is developed in Python using the PyQT4 GUI library and tested on Ubuntu 11.04.

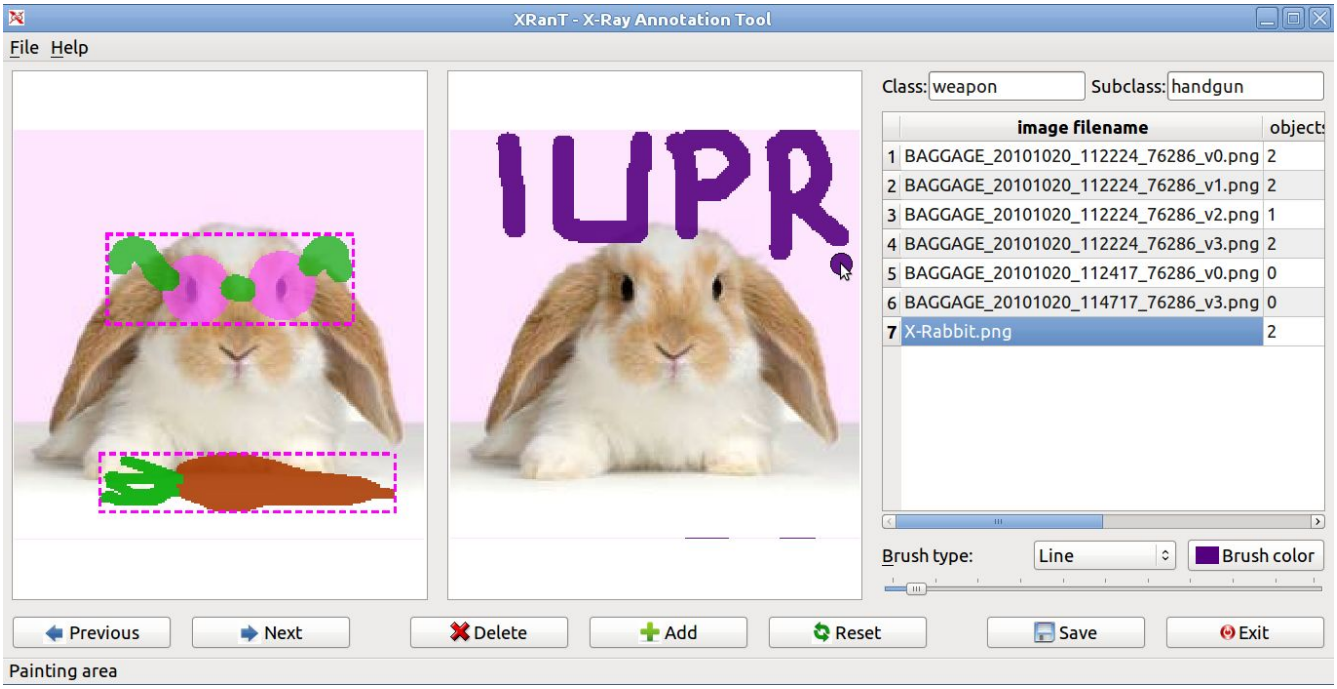


Figure 1: XRanT screenshot. The image list is shown on the right. The user paints on the right image and the selected objects are shown on the left image.

1. Installation

The program is tested only on Linux (Ubuntu 11.04), but it should (hopefully) run on other platforms as well, provided that the required programs and libs are installed.

You should have the following installed (they can be installed using the synaptic package manager in Ubuntu):

- Python (2.7)
- python numpy & scipy
- PyQT4

Then, type ***python XRanT.py*** on the command line to run the program.

2. Interface & Using the Program

To start using the program, type ***python XranT.py***, then do:

(1) ***File → Open image directory*** to load the list of images with the specified file extension (.png, .jpg, ...)

(2) ***File →Load annotation list*** to load existing annotations: this enables the user to do the annotation in multiple sessions, by saving and re-loading the annotations

Image list

As you can see in Figure 1, the list of loaded images are shown in the table on the right. You can navigate through the images by (1) clicking on the *Previous/Next* buttons (2) by clicking on the image name in the table with the mouse (3) using the shortcut keys: *Space* (next), *Ctrl + Space* (previous). The *objects* column in the table shows the number of selected objects for that image.

Image panels and object selection

You use the mouse to paint on the right image, and then click the ***Add*** button (or right click → add object) to add the selected object to the list. The selected objects are shown on the left image panel.

For painting you can use one of the available brushes:

- Line: paints a line over the image, between the consecutive mouse positions as you press the left mouse button and move the mouse (drag)
- Circle, rectangle, rounded rectangle
- Polygon: you draw a polygon to outline the object, when you finish drawing, the tool inpaints the polygon so that you can further edit it using other brushes, or add it to the list:
  - select the points of the polygon as usual: left mouse click
  - the polygon gets updated as you move the mouse
  - *right click → end polygon* to finish drawing

You can switch between the brushes, hide/show the brush (right click), change the brush size and brush color while selecting a single object. You can erase all the painting by: *Reset* button or right click → *reset/clear*. You can increase/decrease the opacity/transparency of painting by right click → *increase/decrease opacity*.

Erasing the painting

You can erase the painting using the same brushes (to fine tune the painting) by: *right click on the image → Change mode to: erase*. You can switch back to painting mode the same way: *right click on the image → Change mode to: paint*.

Deleting objects

On the left image panel, showing the list of selected objects, select the objects with the mouse, then press *Delete* button (shotcut: Del key), or right click → *delete object*. To delete all objects, right click on the image, outside the object MBRs, then *delete all objects*.

Zooming

When you load the images they are scaled to fit to the provided image panels. You can zoom in/out of both images using *Ctrl + mouse wheel*, to see the details and paint more easily. If you double click on the image, the image size toggles between the actual size and fit to the view.

Whole image labeling

In addition to object-level labeling, you can also do image-level labeling. Right click on the image name in the image list table and select *Positive/Negative/Skip*. All images are labeled as *Skip* unless you label them otherwise.

Saving

We have two different data to save: (1) object mask images, (2) annotation list

Saving object mask images

The tool automatically saves the object mask images as *png* files as you navigate to a new image. You can click the *Save* button to force a save (both masks and annotation list) at any time. See below for details.

Warning: The tool saves the object masks automatically as you go, but it does not save the annotation list file, therefore, press the Save button from time to time to avoid the risk of losing data.

[We can set an auto-save for the annotation list as well, e.g., save after annotating 3-5 images]

Class names

You can specify a class name and a subclass name for the annotation using the text fields on the top right, e.g., class name: weapon, subclass name: handgun

Output format

If you specify the class/subclass names, the output will be saved in a directory below the currently loaded image directory. If you load images in /home/user/xray/images/, then the output will be saved in /home/user/xray/images/annotation/subclassname/, or /home/user/xray/images/annotation/classname/, or /home/user/xray/images/annotation/, depending on which one of the class/subclass names you specified.

[This can be changed to /home/user/xray/images/annotation/classname/subclassname/]

You can change the default directory to the one you want by *File → Change ouput directory*.

The file name for each object: ImageName.ID.png, where ID is the ID of the object in the image, ranging from 0 to numberOfObjects – 1.

The object mask images are single channel png images, with bakground = 0, and foreground/object = 255.

Example:

BAGGAGE_20101020_112224_76286_v0.0.png

BAGGAGE_20101020_112224_76286_v0.1.png

BAGGAGE_20101020_112224_76286_v3.0.png

…

[This can also be changed easily depending on the needs.]

When you click on the *Save* button (*or Ctrl+S*, or *File → Save all*), the annotation list file is saved to a file in the same directory as the object masks with the (default) filename: ImageDir.ClassName.SubClassName.txt (example: carry_on_weapon.weapon.handgun.txt). You can save a copy to another location with a new file name: *File → Save a copy of annotation list as*.

The file has the following format:

line 1: className subClassname

line 2: image_directory folder_name

line 3: output_directory folder_name

line 4: number_of_images

line i : set level image_label number_of_objects *image_file_name* 'view label object_MBRs'

set: training/test/skip set [1, -1, 0]

level: difficulty level of image [0, 1, 2, 3, 4, 5]

image_label: positive (1), negative (-1), skip (0)

number_of_objects: number of objects you selected in the image

image_file_name: as the name suggests

view: object view [0, 1, 2, 3]

label: object label, positive (1), negative (-1), skip (0)

object_MBRs: minimum bounding rectangles (MBR – x1, y1, w, h) of the selected objects for each image, in the same order as the object mask png images, i.e., the first MBR corresponds to *.0.png image.

Example: carry_on_weapons.weapon.handgun.txt

weapon handgun

/home/bastan/research/sicura/data/carry_on_weapons/ carry_on_weapons

//home/bastan/research/sicura/data/carry_on_weapons/annotation/handgun/ handgun

3

0 0 -1 0 20101020_111645_76286_v3.png

0 0 1 2 20101020_112224_76286_v0.png 1 0 414 312 238 249 1 0 400 470 205 100

0 0 1 1 20101020_112224_76286_v1.png 1 0 596 308 111 228

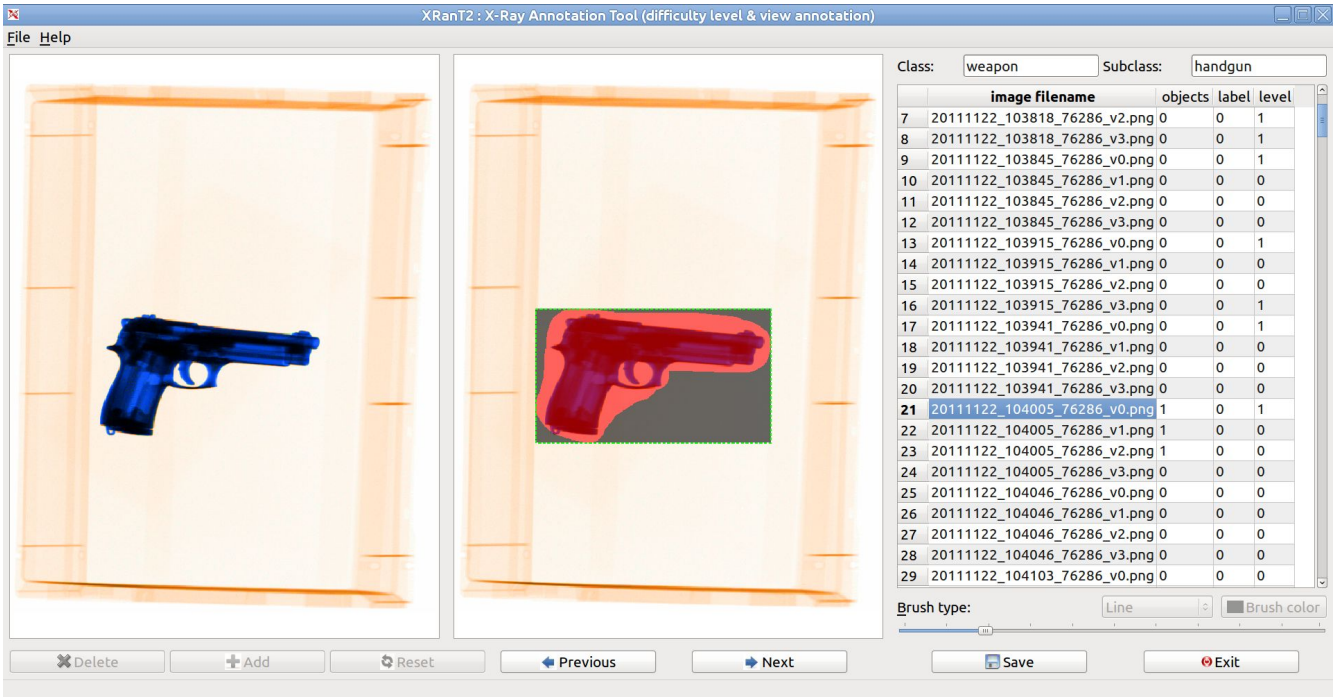
------ from the Annotation.py file --------

```
# whole image annotation labels
LPOS, LNEG, LSKIP = 1, -1, 0

# difficulty level:
# skip: no label (default)
# simple L1: one object on clean background
# simple L2: multiple objects on clean background
# medium L3: single/multiple objects, but not cluttered
# difficult L4: cluttered
# very difficult L5: very hard or impossible to identify the contents
L0, L1, L2, L3, L4, L5 = 0, 1, 2, 3, 4, 5

# view type:
# skip V0: no label
# good view V1: a typical, easy to recognize view @ zero (or close to zero) angle
# moderate V2: at some angle, but clearly visible
# side view V3: side, hard to recognize view
V0, V1, V2, V3 = 0, 1, 2, 3

# set
# S0: skip this image, do not use it
# STR: in the training set
# STS: in the test set
S0, STR, STS = 0, 1, -1
```

Muhammet Bastan, IUPR, August 2011

XRanT2

The tool is a simpler version of XRanT for annotating the images with 5 difficulty levels & annotating the selected objects with 3 view labels. [Date 07.12.2011]



Difficulty level annotation

Right click on the image name in the table on the right or on the right image (but not on the selected object, outside --- right clicking on the object is for view annotation) and select one of the options (0--5). The table column “level” will be updated with the new label (0—5). You should click on the image if you want to label the current image. You can set the label of ALL the images by right-clicking on the table on the right.

View annotation of the objects

Right click on the object (if there is any in the current image) and select one of the options (0—3), and the color of the MBR of the object will be updated accordingly (magenta, green, blue, red --- see below)

Difficulty Levels

Here we define the difficulty levels that will be assigned to the images using the XRanT2 tool. When you right click on the image (or the table as described above), you see the following options:

level 1: simple -- one object in the bin

level 2: simple -- mult. objects in the bin,

level 3: medium -- baggage un-cluttered

level 4: difficult -- baggage cluttered

level 5: too difficult -- hard to see

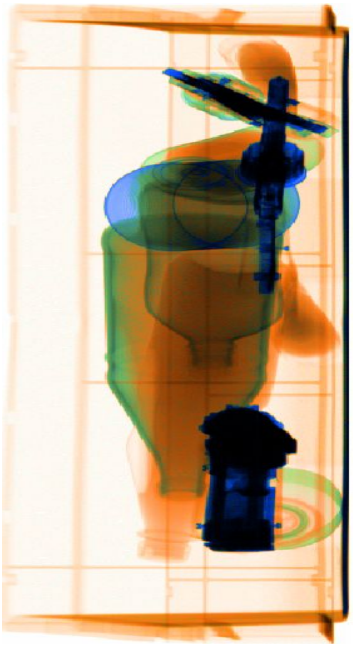
skip 0: not labeled

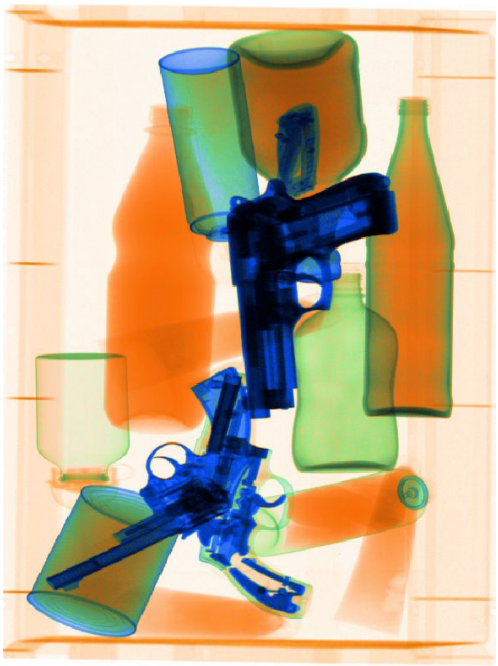
Below each difficulty level will be illustrated with images. “Skip 0” means there is no label provided for the image and this is the initial/default value.

level 1: simple -- one object in the bin: there is only a single object in the **bin**; the object can be anything. The object should in the **bin** (hence the background is very clean), not in a baggage (in that case there is usually some background clutter). There images are recorded to serve as models.

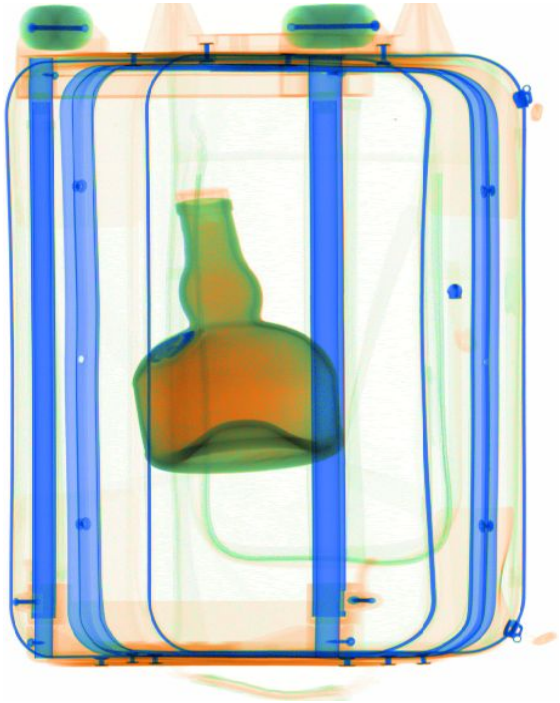
***level 2*: *simple -- multiple objects in the bin*:** there are multiple objects in the bin; the objects can be the same or different. The objects should in the bin (hence the background is very clean), not in a baggage (in that case there is usually some background clutter). There can be some overlap and clutter between the objects but they should be easily visible and recognizable.

If the overlap and clutter is too much that it is hard to clearly see and recognize the objects then we can consider labeling the image as "**level 4: difficult**", like the following images:



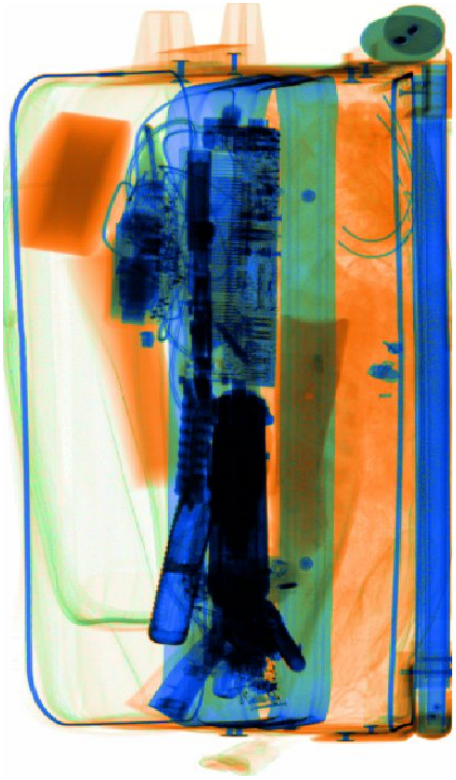


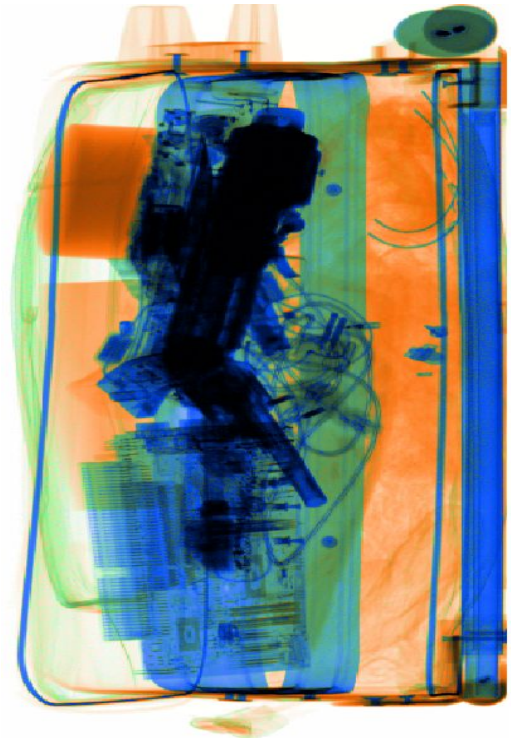
level 3: medium -- baggage un-cluttered: single or multiple objects in a baggage, without clutter. There is a little clutter due to baggage (zips, wheels, etc.), but the objects are clearly visible and recognizable. The objects can be anything.

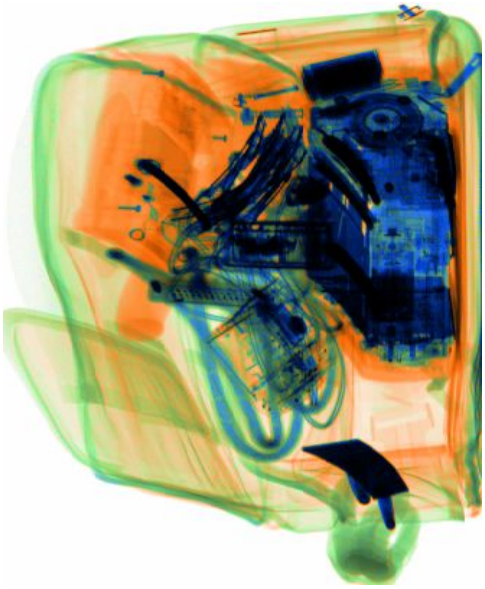


level 4: difficult -- baggage cluttered: single or multiple objects in a baggage, with overlap and/or clutter. The clutter may be either due to baggage or the objects. The objects can be anything. See also ***level 2*** above.

level 5: too difficult -- hard to see: it very hard if not impossible to see and identify the objects. This may be due to too much overlap/clutter or due to a bad view (e.g., usually side view).







Muhammet Bastan, IUPR, August 2011

View annotation

Here we define the view annotation labels that will be assigned to the selected objects in images using the XRanT2 tool. When you right click on an already selected object in an image, you see the following options:

view 1: clear, visible (green mbr)

view 2: at some angle (blue mbr)

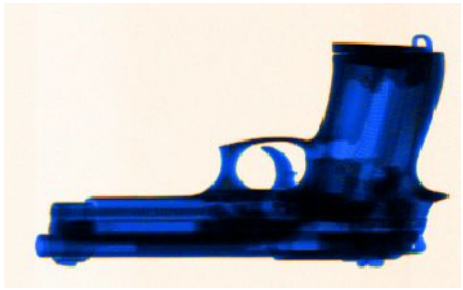
view 3: side, not clearly visible (red mbr)

skip 0: no view label (magenta mbr)

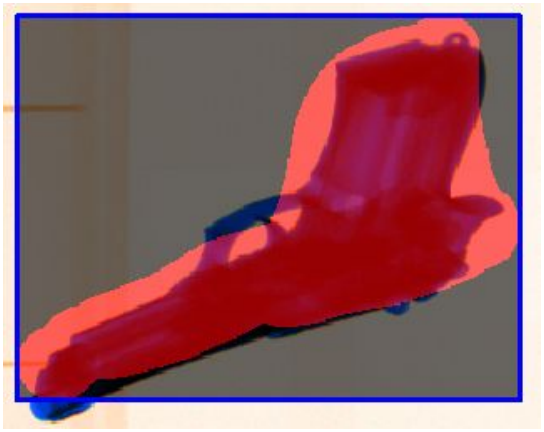
Below each view level will be illustrated with images. The MBR of each object has a color corresponding to its view label: green (1), blue (2), red (3), magenta (0—skip).

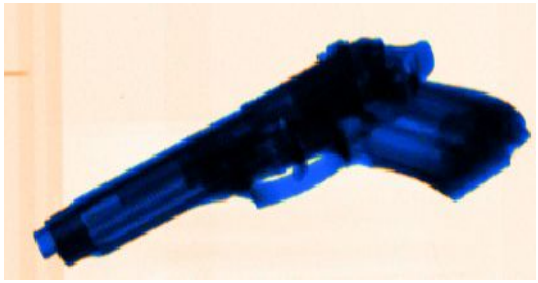
"Skip 0" means there is no view label provided for the object and this is the initial/default value.

view 1: clear, visible: this corresponds to the standard and most easily recognizable view of an object. The MBR is drawn in green.



view 2: at some angle: the object is imaged at some angle (due to different viewing angles) and is not in the most standard form, but it is still easily recognizable. The MBR is drawn in blue.





view 3: side, not clearly visible: the object is imaged from a side view and not easily recognizable. The MBR is drawn in red.

Muhammet Bastan, IUPR, August 2011

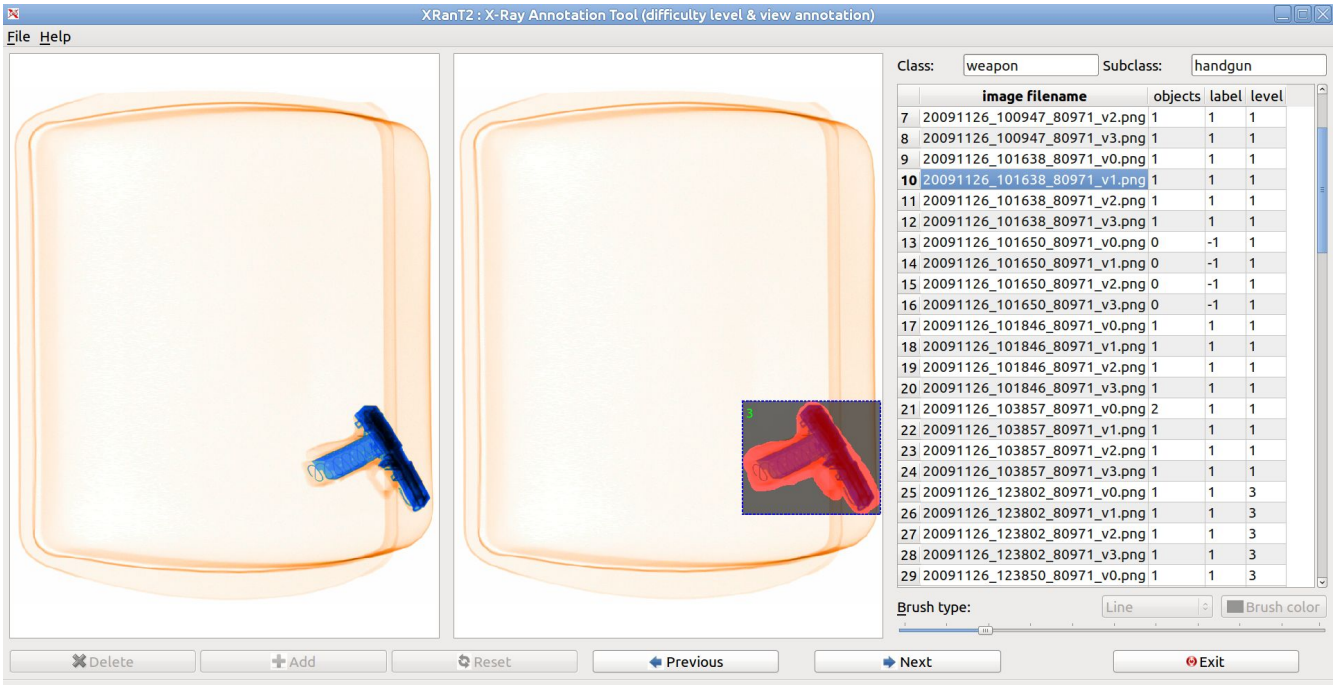
Object ID annotation

Date: July 5, 2012

Use XRanT3.py, which in turn uses Annotation23.py, for annotating the objects with IDs. You should have created the object library and decided the object IDs!

First, adjust the ID values in XranT3.py, in line 32, **idvaluesstrs** list. The list contains the Ids and and associated string description for each ID (if any). Currently the list has [0,1,..., 20], but for ease of use, you can change the list to make it shorter (or longer). Suppose you will annotate a folder which has objects with IDs [3,4,5], then you can update the list accordingly so that in the GUI you will have a much shorter list (just 3,4,5) to select from. You can also add some descriptions to the string representations, e.g., ("1: atuto gun", 1).

Notice: leave ID **0** for cases where you cannot identify the object (hence skip).



Notice that the ID of each object in the image is show on the top left corner of the MBR of the object in green (in the image 3).

Procedure:

1. run XRanT3.py
2. File – Load annotation list: to load existing annotations from file which does not have object Ids (basically obtained with XranT.py or XranT2.py)
3. Go over the images, for each object shown on the right image panel, right click and select the ID from the list.

4. When you are done, save your updated annotations: File – Save annotation list with object IDs as
5. Exit

You can do the annotations in multiple sessions. Just load the updated annotations (files with object Ids): File – Load annotation list with object IDs

The output format:

0 1 1 1 20091126_100823_80971_v0.png 1 0 2 165 452 268 202

from Annotation23.py:

str(self.set) + ' ' + str(self.level) + ' ' + str(self.label) + ' ' + str(self.numObjects()) + ' ' + self.fname

for each object:

str(obj.view) + ' ' + str(obj.label) + ' ' + str(obj.mid) + ' ' + str(obj.x1) + ' ' + str(obj.y1) + ' ' + str(obj.w) + ' ' + str(obj.h)