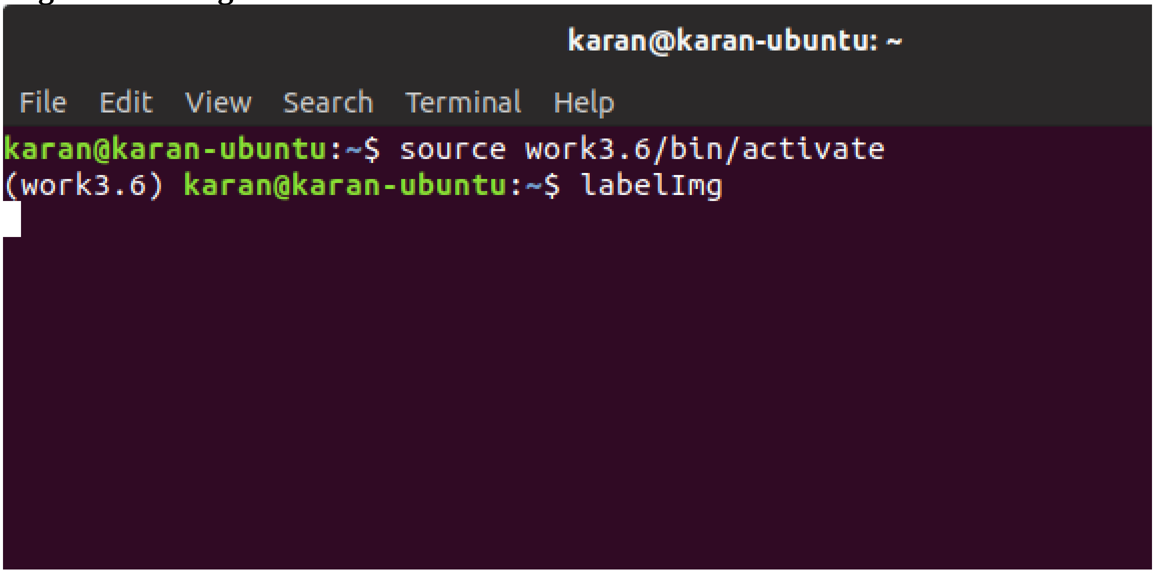
Installing package for Labeling/Annotating images:
pip install labelImg

Official GitHub Link: https://github.com/tzutalin/labelImg

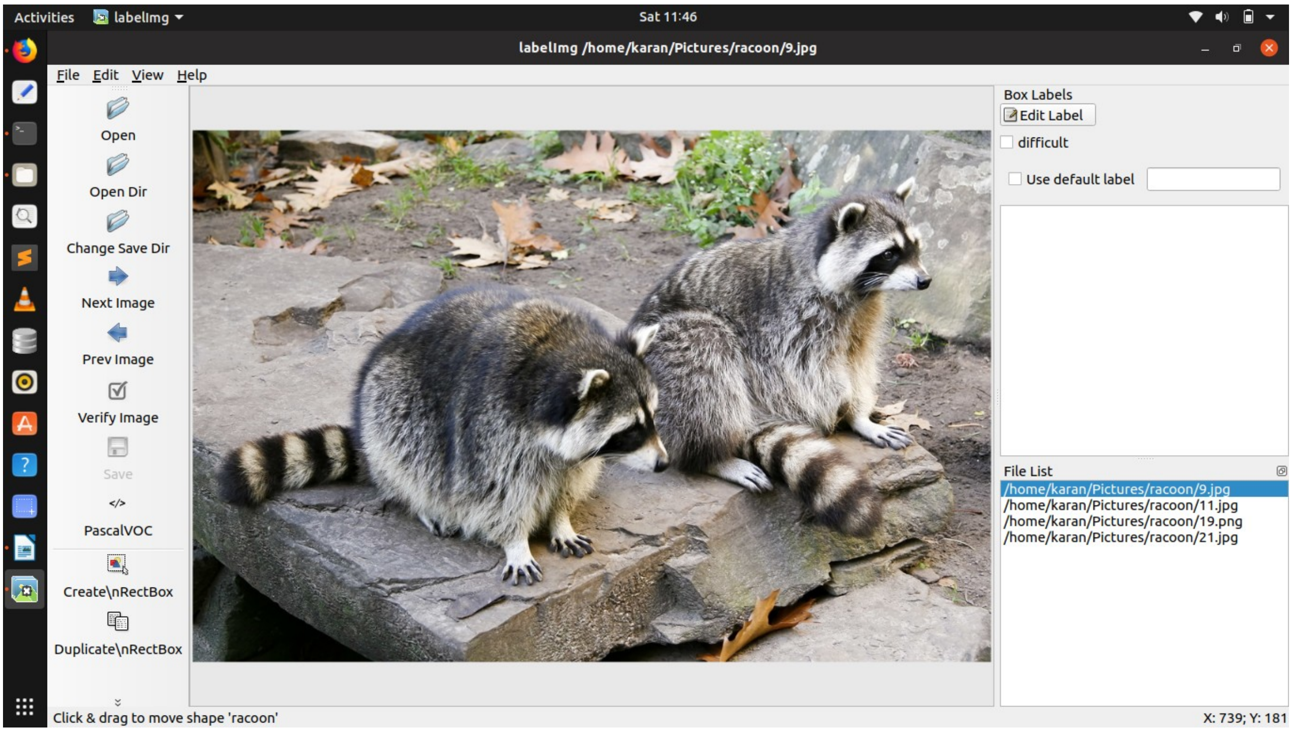
Running the labelImg from terminal:



```
karan@karan-ubuntu:~$ source work3.6/bin/activate
(work3.6) karan@karan-ubuntu:~$ labelImg
```

labelImg window looks like this:

- Set the **Open Dir** option
- Set the **Change Save Dir** option

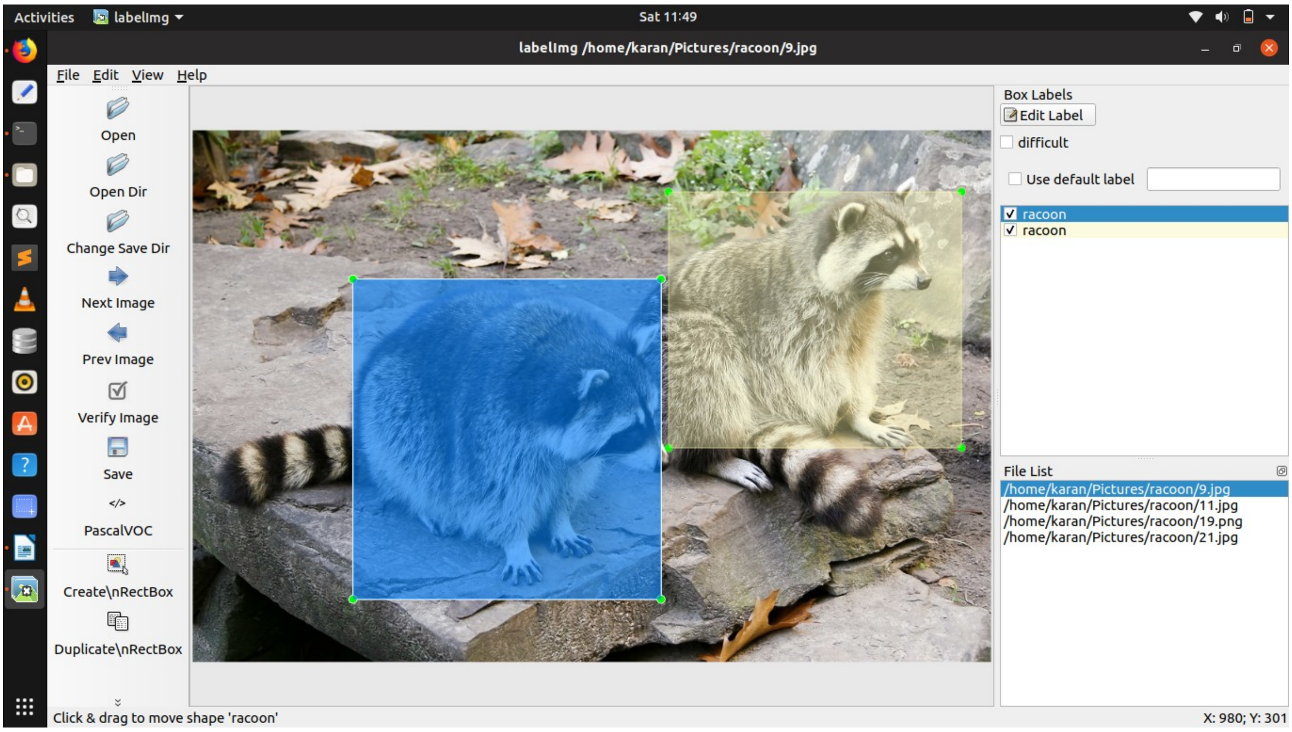


/home/karan/Pictures/racoon/9.jpg
/home/karan/Pictures/racoon/11.jpg
/home/karan/Pictures/racoon/19.png
/home/karan/Pictures/racoon/21.jpg

Click & drag to move shape 'racoon'

X: 739; Y: 181

Annotating the images as below:



☑ racoon
☑ racoon

/home/karan/Pictures/racoon/9.jpg
/home/karan/Pictures/racoon/11.jpg
/home/karan/Pictures/racoon/19.png
/home/karan/Pictures/racoon/21.jpg

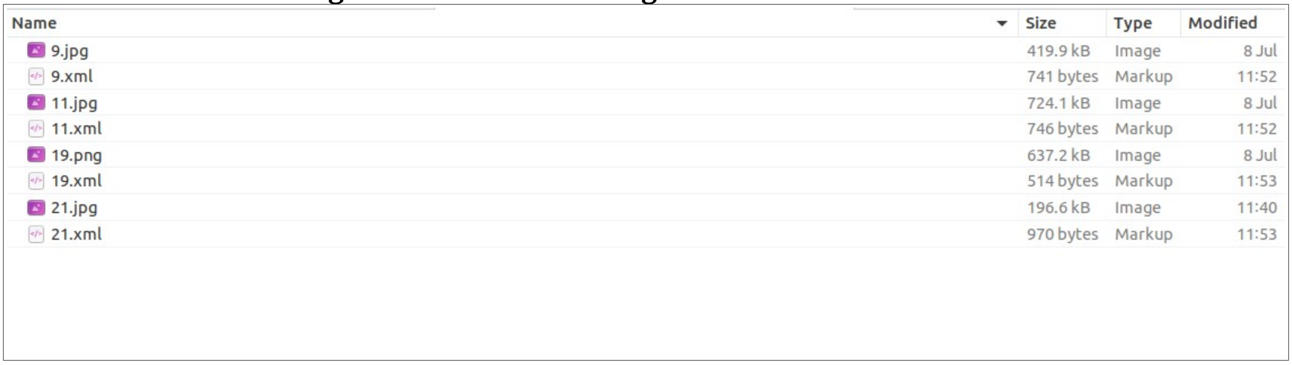
Click & drag to move shape 'racoon'

X: 980; Y: 301

Click CTRL + S to save the JSON form of the Labeled box:

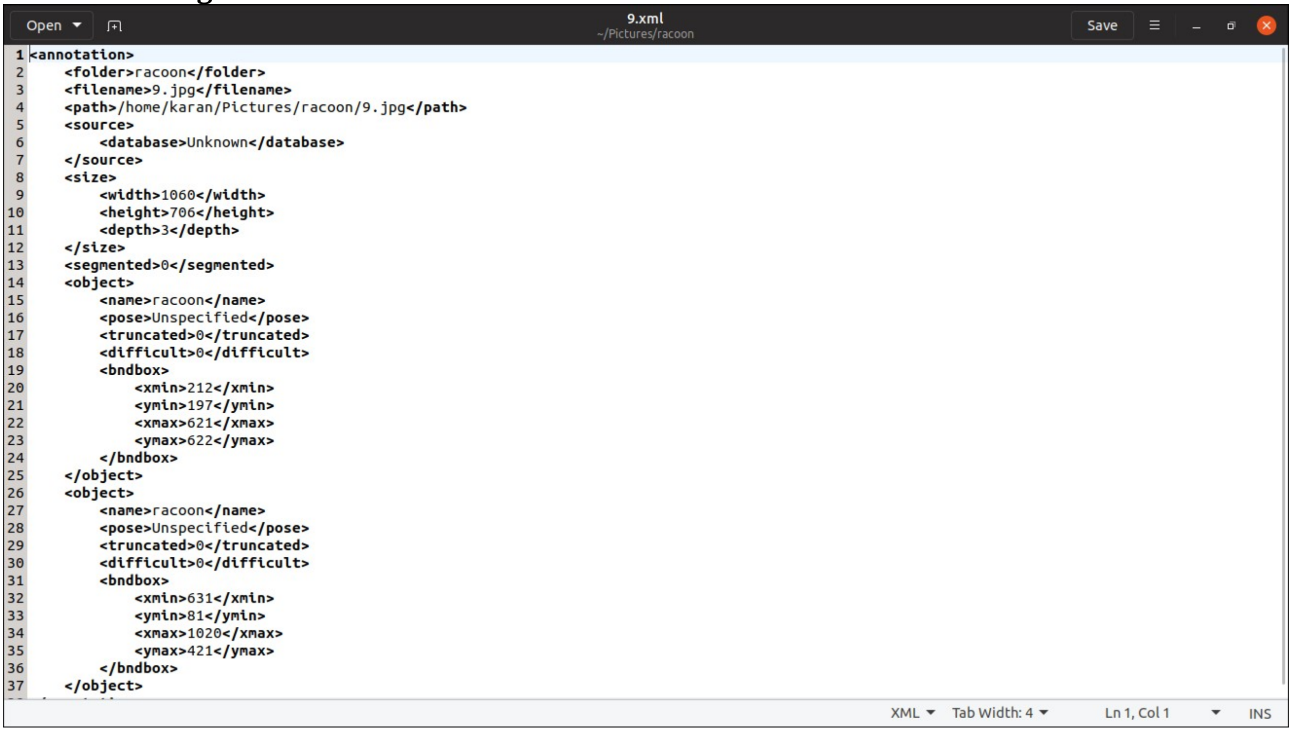
Repeat this for all the images you want to annotate:

- **.xml files will be generated for each image**



Name	Size	Type	Modified
9.jpg	419.9 kB	Image	8 Jul
9.xml	741 bytes	Markup	11:52
11.jpg	724.1 kB	Image	8 Jul
11.xml	746 bytes	Markup	11:52
19.png	637.2 kB	Image	8 Jul
19.xml	514 bytes	Markup	11:53
21.jpg	196.6 kB	Image	11:40
21.xml	970 bytes	Markup	11:53

The XML file generated looks like this:

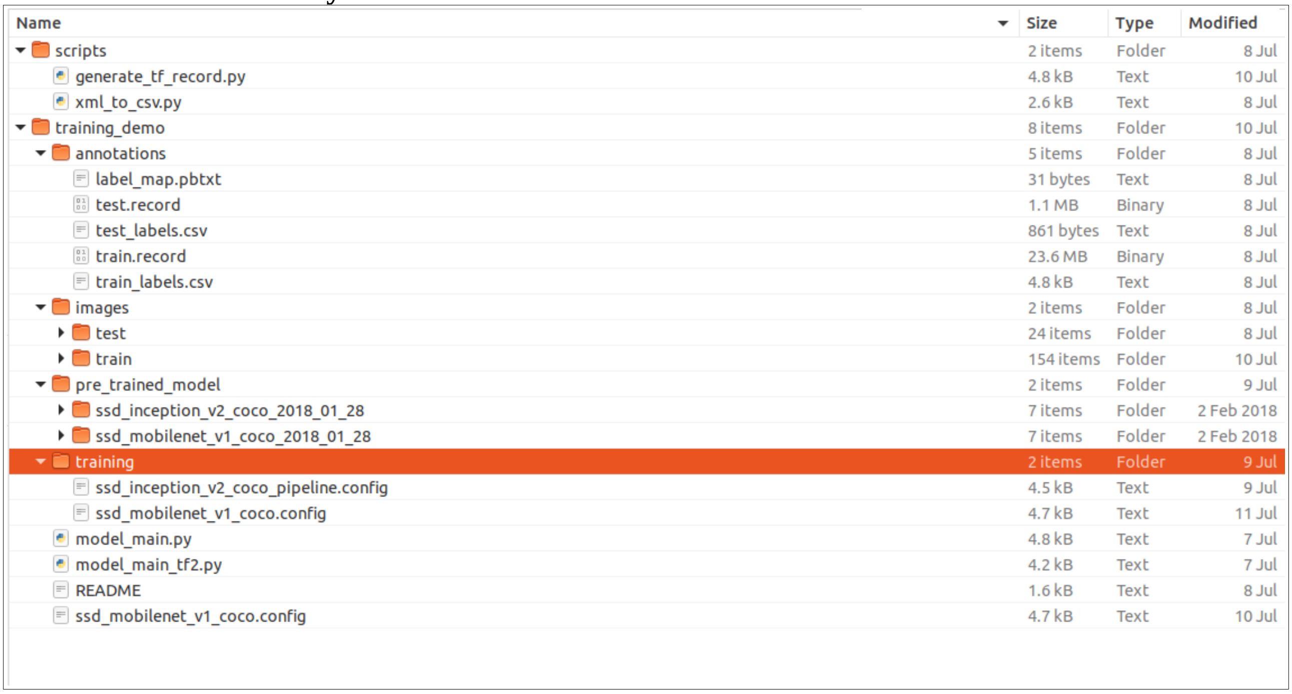


```xml
<annotation>
    <folder>racoon</folder>
    <filename>9.jpg</filename>
    <path>/home/karan/Pictures/racoon/9.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>1060</width>
        <height>706</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>racoon</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>212</xmin>
            <ymin>197</ymin>
            <xmax>621</xmax>
            <ymax>622</ymax>
        </bndbox>
    </object>
    <object>
        <name>racoon</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>631</xmin>
            <ymin>81</ymin>
            <xmax>1020</xmax>
            <ymax>421</ymax>
        </bndbox>
    </object>
```

Here, the annotation process completes.
After this, the following steps need to be done:

1. **Convert all the** .xml files **to a** single .csv file.
2. **Create a file named** label_map.pbtxt contatining names of all the labels that your dataset contains.
3. Generate TF-Record file for training and testing images.
4. Use this TF-Record file for custom dataset object detection.

This is how the directory structure looks like:



Name	Size	Type	Modified
scripts	2 items	Folder	8 Jul
generate_tf_record.py	4.8 kB	Text	10 Jul
xml_to_csv.py	2.6 kB	Text	8 Jul
training_demo	8 items	Folder	10 Jul
annotations	5 items	Folder	8 Jul
label_map.pbtxt	31 bytes	Text	8 Jul
test.record	1.1 MB	Binary	8 Jul
test_labels.csv	861 bytes	Text	8 Jul
train.record	23.6 MB	Binary	8 Jul
train_labels.csv	4.8 kB	Text	8 Jul
images	2 items	Folder	8 Jul
test	24 items	Folder	8 Jul
train	154 items	Folder	10 Jul
pre_trained_model	2 items	Folder	9 Jul
ssd_inception_v2_coco_2018_01_28	7 items	Folder	2 Feb 2018
ssd_mobilenet_v1_coco_2018_01_28	7 items	Folder	2 Feb 2018
training	2 items	Folder	9 Jul
ssd_inception_v2_coco_pipeline.config	4.5 kB	Text	9 Jul
ssd_mobilenet_v1_coco.config	4.7 kB	Text	11 Jul
model_main.py	4.8 kB	Text	7 Jul
model_main_tf2.py	4.2 kB	Text	7 Jul
README	1.6 kB	Text	8 Jul
ssd_mobilenet_v1_coco.config	4.7 kB	Text	10 Jul