**Object Detection Training — Preparing your custom dataset**

*Tutorial on* ***annotating*** *your custom image dataset in the Pascal VOC format*



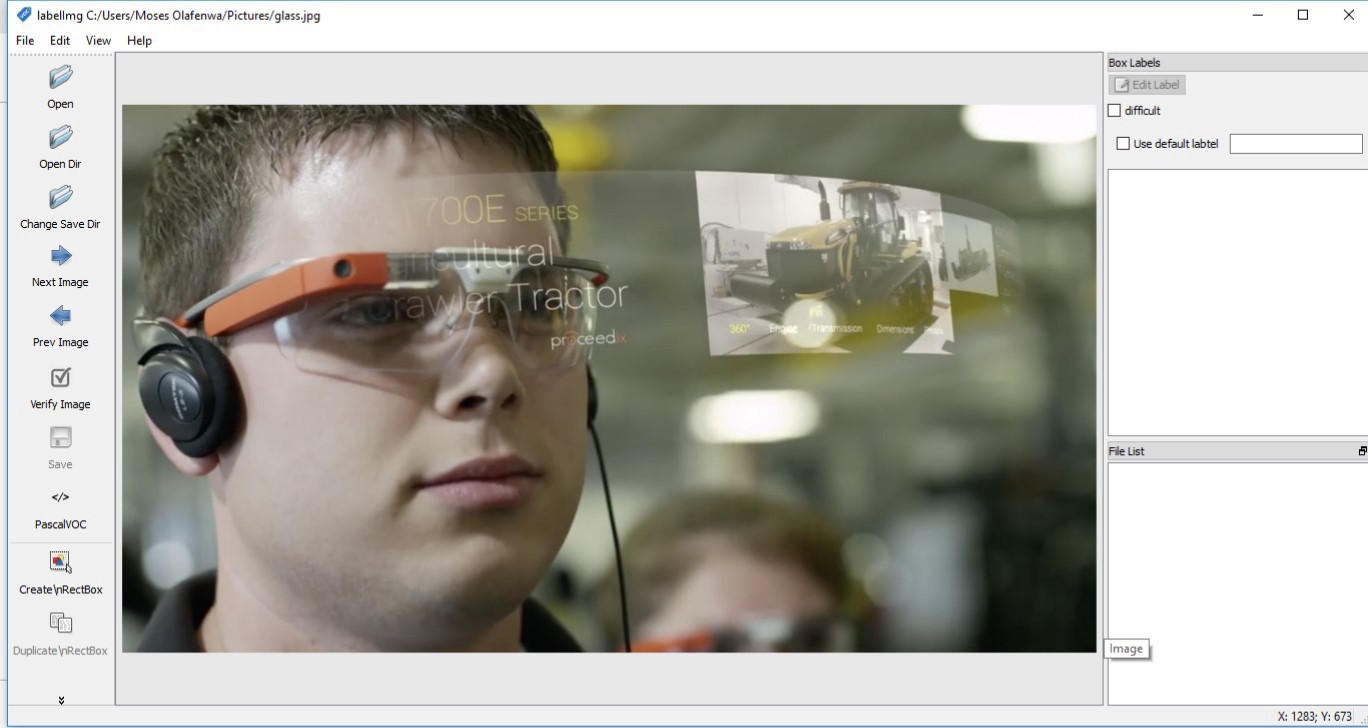
**Object detection** is one of the most fascinating aspect of **computer vision** as it allows you to **detect** each individual object in images, **locate** their position and size as well relative to the rest of the image. Today’s state-of-the-art object detection models are powered by **Deep Learning**and at the very soul of training deep learning networks is the **training dataset**.

**Training dataset** are images collected as **samples** and **annotated** for training deep neural networks. For object detection, there are many formats for preparing and annotating your dataset for training. The most popular formats for annotating your datasets are:

* **Pascal VOC**
* **Micosoft COCO**

For the purpose of this tutorial, we will be showing you how to prepare your image dataset in the Pascal VOC annotation format.

**Step 1 — Install LabelImg**



The **Pascal VOC**format uses **XML files** to store details of the objects in your individual images. To easily generate these XML files for the images, we will be using an easy to use tool by the name **LabelImg**that allows you to

* draw visual boxes around your objects in the images
* and it automatically saves the XML files for your images

Install **LabelImg** via the options available below.

**FOR WINDOWS:**

* Install **Python3**via <https://www.python.org/downloads/release/python-357/>
* Install **PyQt5**and **lxml** via **PIP**

pip3 install pyqt5 lxml

* Download, unzip and run the **LabelImg.exe**file via <https://www.dropbox.com/s/kqoxr10l3rkstqd/windows_v1.8.0.zip?dl=1>

**FOR LINUX:**

* Install **LabelImg**via **PIP**

pip3 install labelimg

* Launch **LabelImg**via the command below

python3 labelimg

**FOR MACOS:**

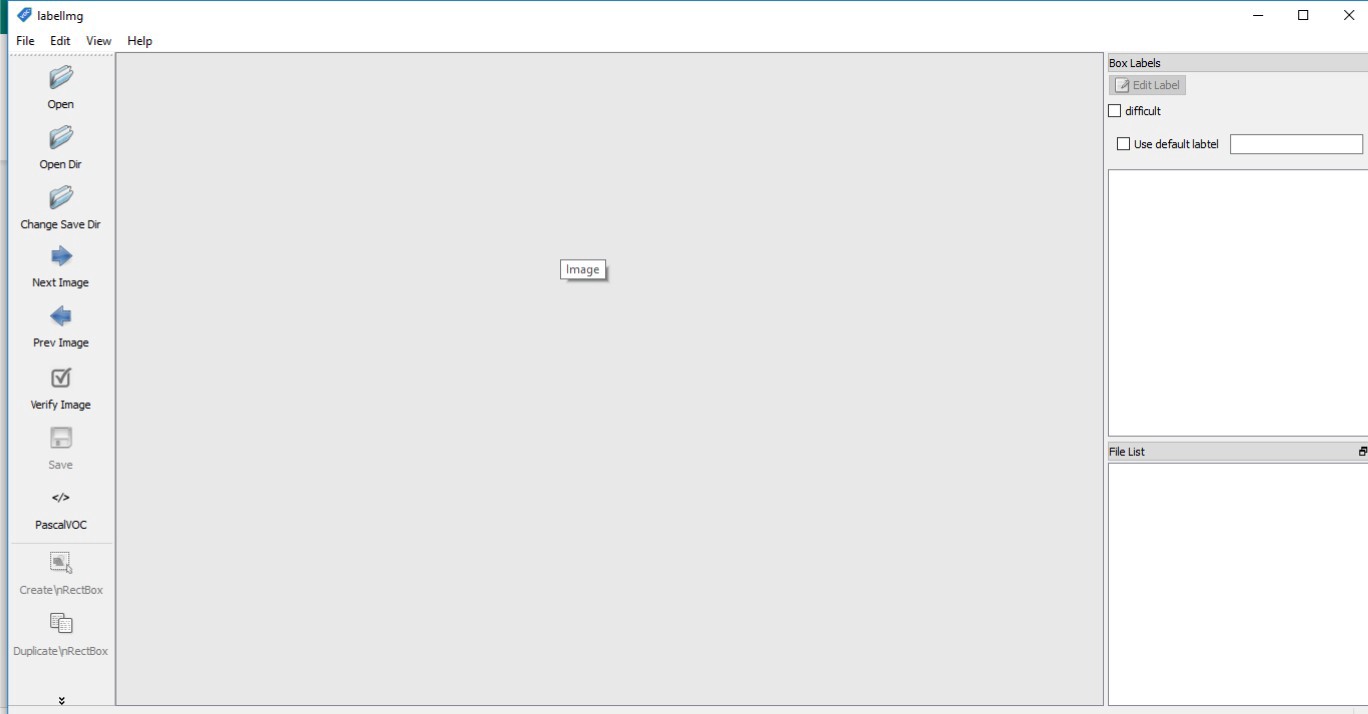
* Clone the **LabelImg**repository

git clone <https://github.com/tzutalin/labelImg.git>

* Then run the commands below in the repository folder

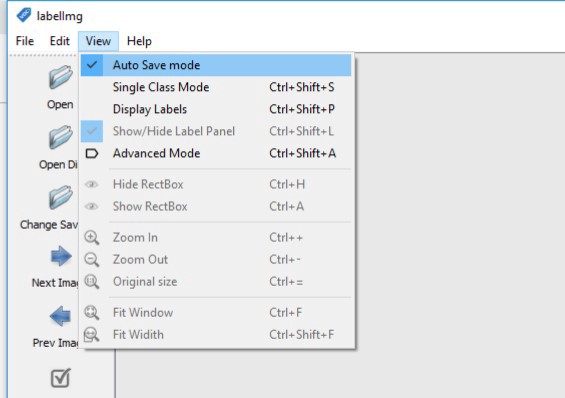
pip3 install pyqt5 lxml # Install qt and lxml by pip  
  
make qt5py3  
python3 labelImg.py

**Step 2 — Organize your image dataset**

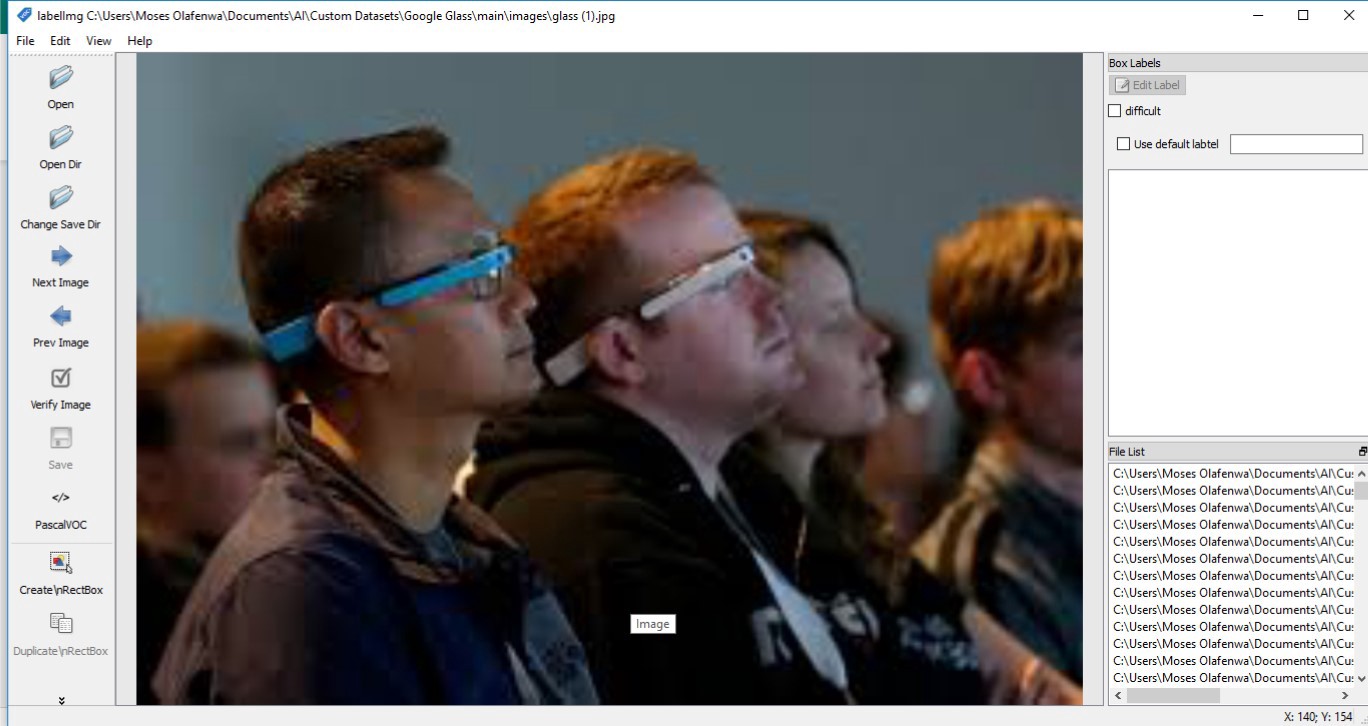


Now that LabelImg annotation tool is opened follow the instructions below:

* create a folder “**images”**and put all your images in it
* create another folder “**annotations**”
* Then go to your LabelImg menu, select “**View”**and make sure “**Auto Save Mode”**is checked.



* Click on “**Open Dir**” on the top-left and select your “**images**” directory where your images are kept. The first image in your folder will be shown as seen in the example below.



* Click on the “**Change Save Dir**” on the top-left and select your “**annotations**” folder. This is where the generated XML file containing the annotation for your images will be stored.

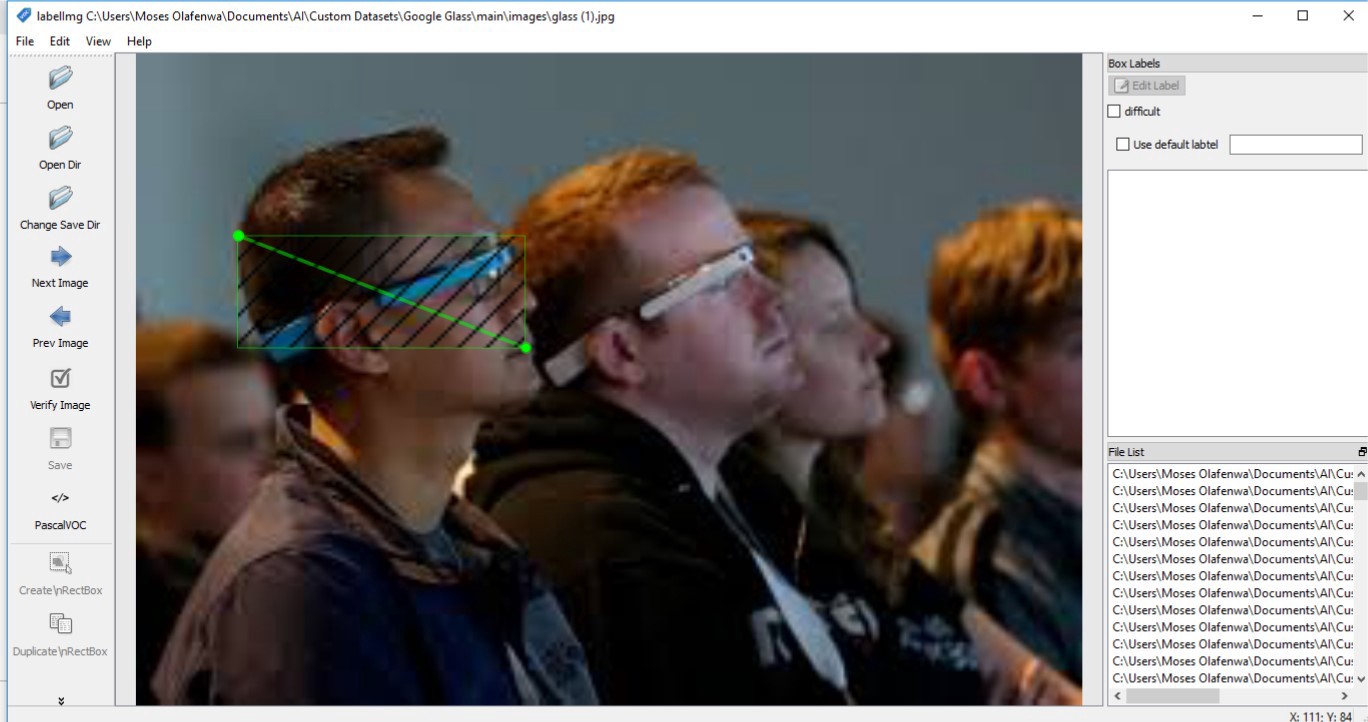
**Step 4 — Annotating your dataset**

Now that you have loaded your images and set the save folder for the annotations. In this example, we are using an image dataset on **Google Glass.**

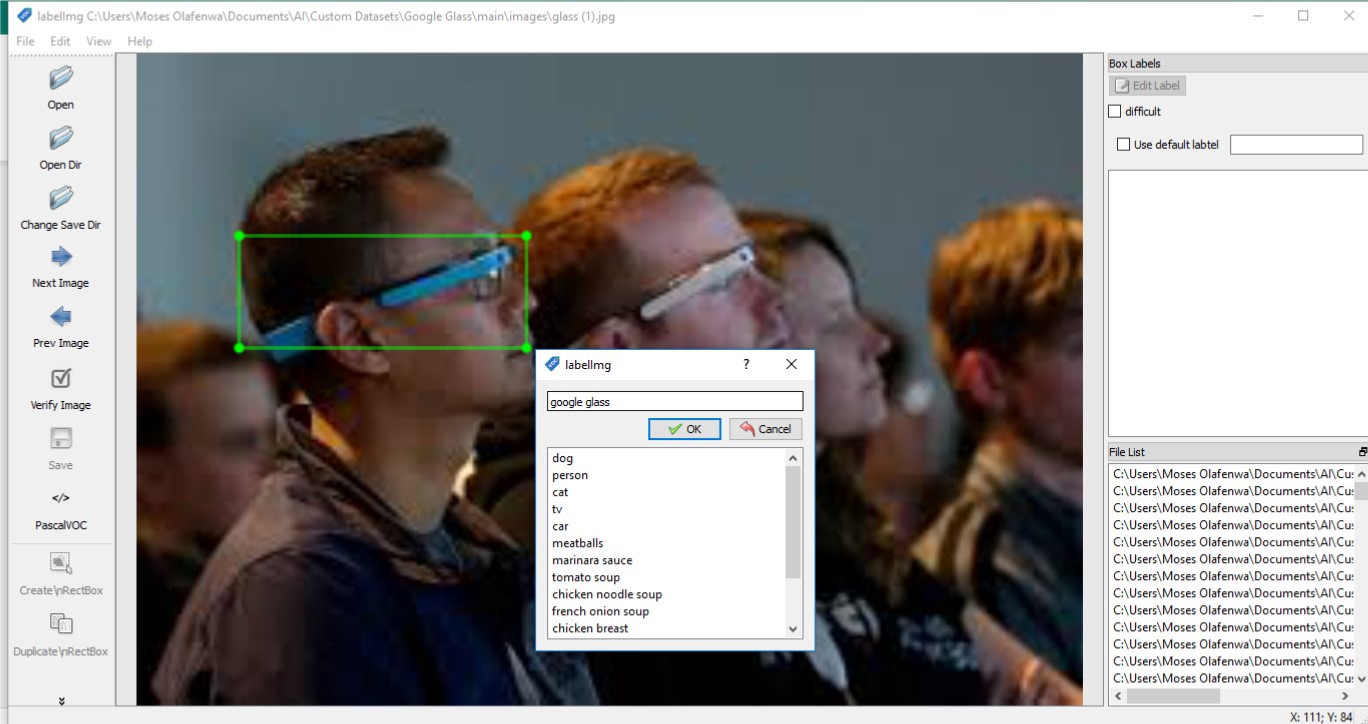


Start annotating your images by:

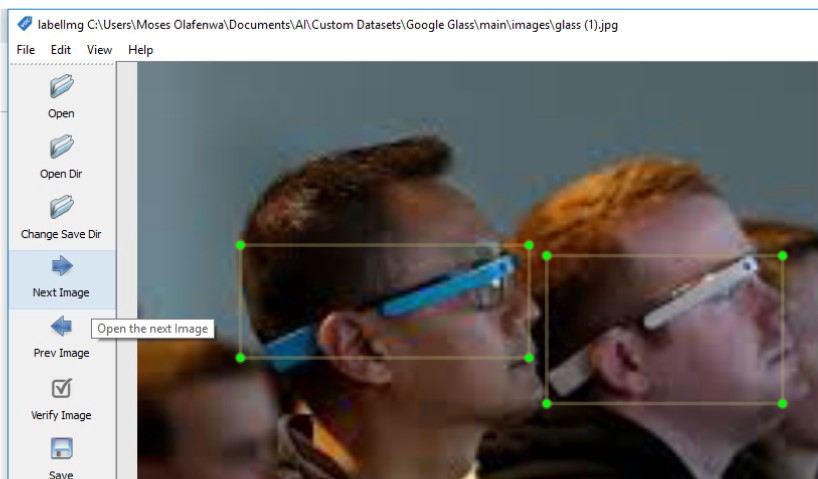
* Click on the “**Create \nRectBox**” button on the left-bottom and draw a box around the objects you want to annotate as seen in the images below.

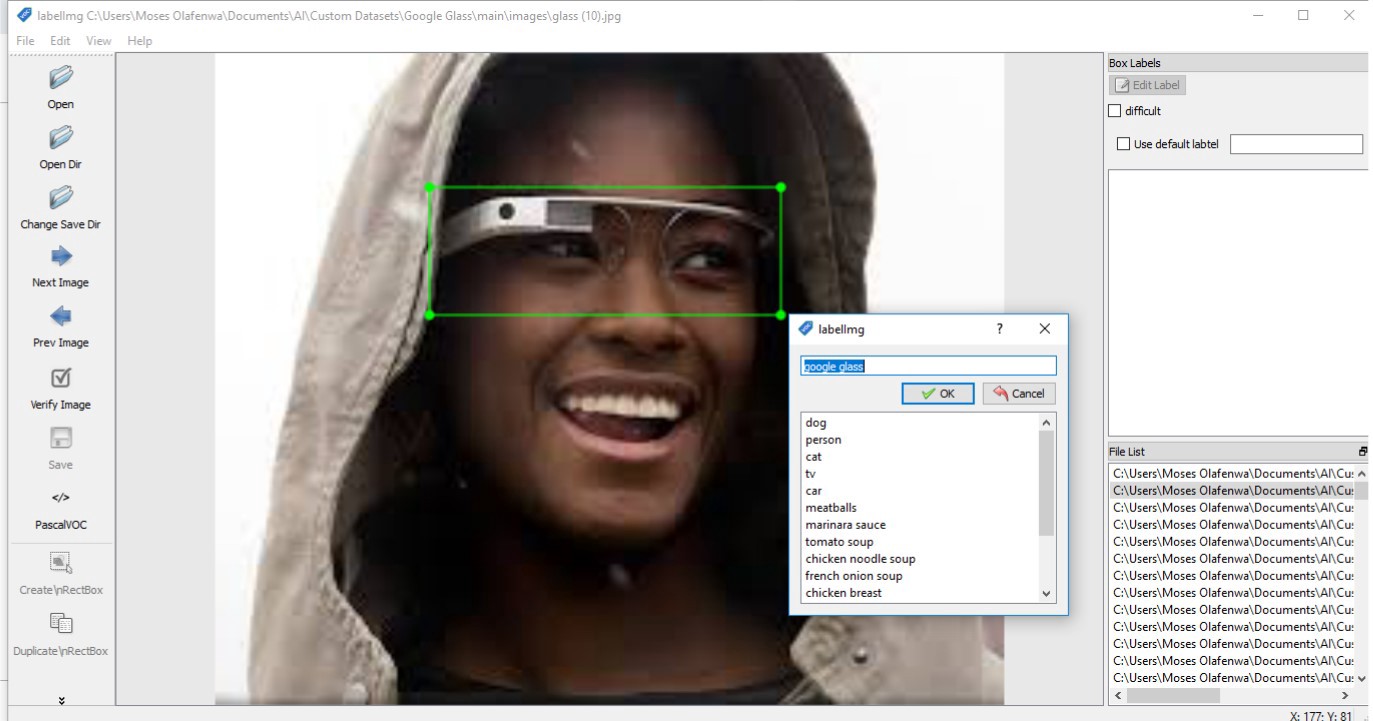


* Then once drawn, enter the name of the object in the **pop-up** box and select the “**OK**” button as seen in the example below



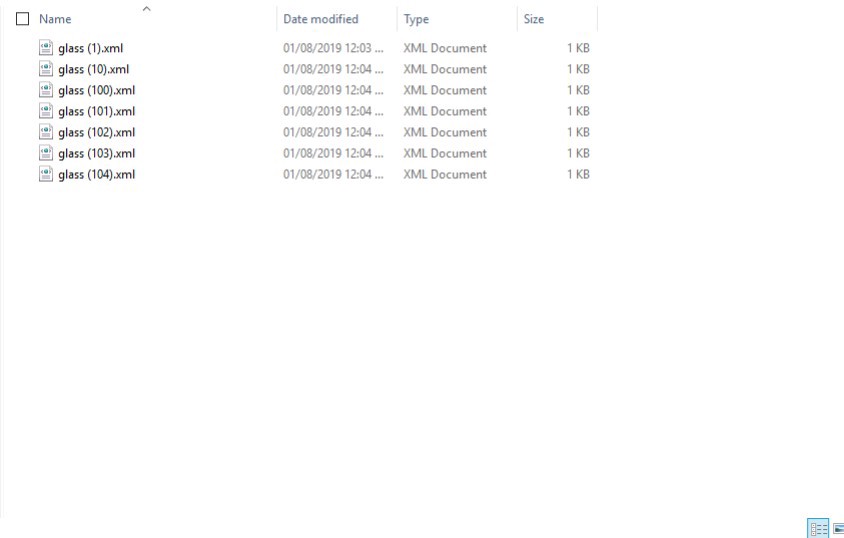
* Click on the “**Create \nRectBox**” button again and annotate all the objects in the image.
* Once you are done, click the “**Next Image**” button on the middle-left to annotate the another image.





* Continue this process until you are done annotating all your images.

As you are annotating your images, the XML file containing your box annotations are saved for each image in the “**annotations**” folder. See the picture below for the corresponding annotations XML files generated.



**N.B:**Take note that the annotation XML file for each image is saved using the name of the image file. For example:

* you have images **image\_1.jpg**, **image\_2.jpg** …… **image\_z.jpg**
* the XML annotations file will be saved as **image\_1.xml**, **image\_2.xml,**…. **image\_z.xml**

Once you are done annotating your image dataset in the **Pascal VOC**format, you can use **ImageAI’**s custom detection training code to train a new detection model on your datasets, using just **6-lines**of Python code.