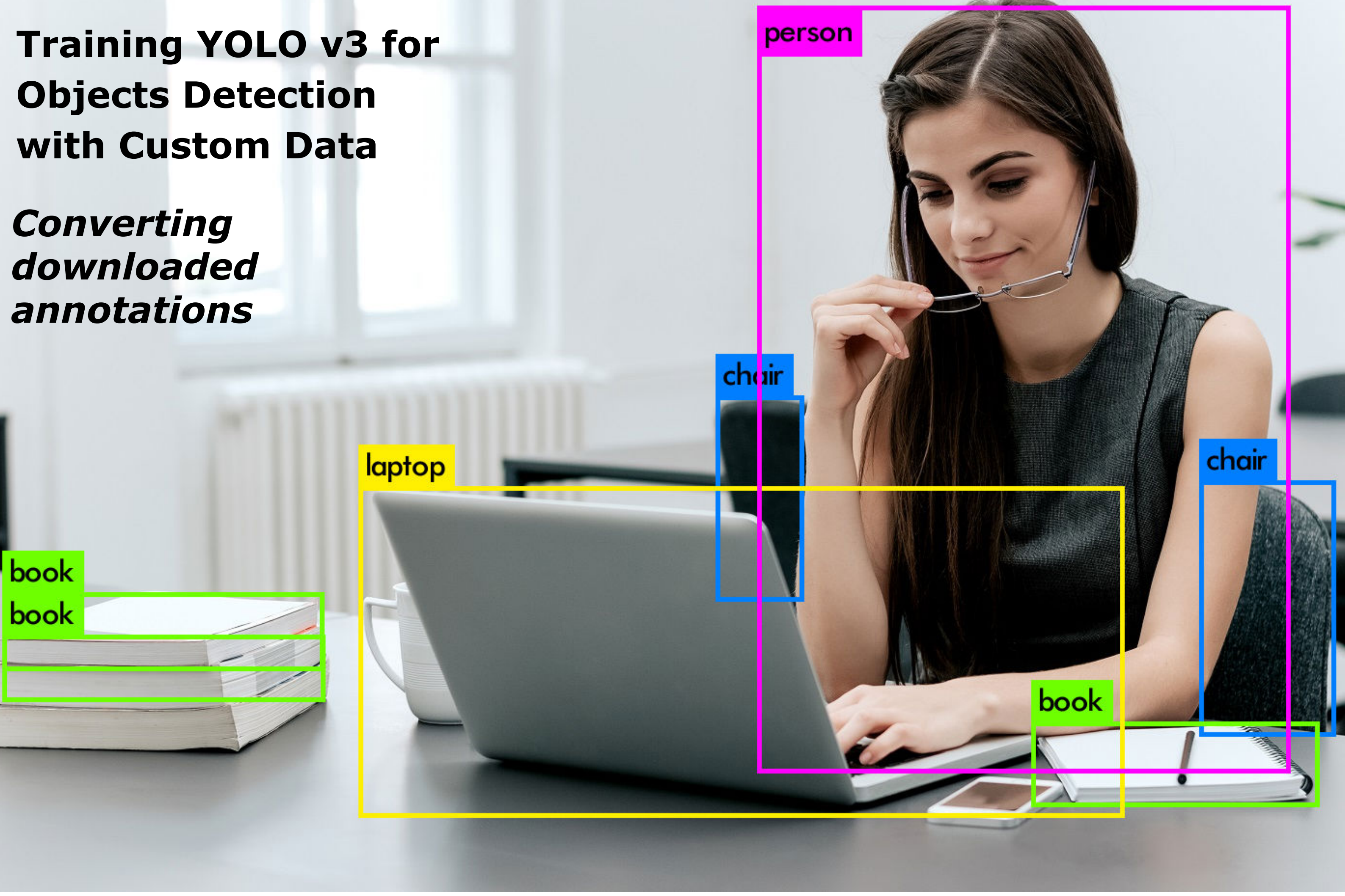
Training YOLO v3 for Objects Detection with Custom Data

Converting downloaded annotations



Converting downloaded files into YOLO format

After downloading images and annotations from *Open Images Dataset*, it is needed to convert given annotations into YOLO format. Annotations of bounding boxes' coordinates in *csv* file are as following:

XMin **XMax** **YMin** **YMax**

but YOLO needs following:

[centre in x] **[centre in y]** **[width]** **[height]**

All annotations are in one *csv* file, but YOLO needs separate *txt* file next to every image and with the same name as image file has.

Download Py files into Custom-Data

Create a folder with name *Custom-Data* to keep everything organized. Download *Py* files from *Resources* and copy them to this folder. You should have following:

- *Custom-Data/*
    - *getting-full-path.py*
    - *converting-annotations.py*

Getting full paths

Before converting annotations into YOLO format, it is needed to find *absolute* or *full path* to the *csv* files with annotations and *full path* to the *downloaded images*.

Before finding *full path* to the *downloaded images* it is needed to change name of the folder, replacing gap between words *Bicycle* and *wheel* by bottom character. In this way we eliminate future possible mistakes. Open explorer, find and rename folder `Car_Bicycle wheel_Bus` to `Car_Bicycle_wheel_Bus` replacing gap by _.

- Copy and paste *Py* file ***getting-full-path.py*** to the folder with *csv* files and inside the folder with *downloaded images*:
    - *OIDv4_Toolkit/OID/csv_folder*
    - *OIDv4_Toolkit/OID/Dataset/train/Car_Bicycle_wheel_Bus* (yours can be different if you downloaded other classes)
- Open *Terminal* (or *Anaconda Prompt*) and activate your *Python v3* environment
- Go to the directory *OIDv4_Toolkit/OID/csv_folder* and run following command:

  ```
  python3 getting-full-path.py
  ```

  or:

  ```
  python getting-full-path.py
  ```

- Go to the directory *OIDv4_Toolkit/OID/Dataset/train/Car_Bicycle_wheel_Bus* and run following command:

  ```
  python3 getting-full-path.py
  ```

  or:

  ```
  python getting-full-path.py
  ```

- You should get two full paths like following (yours should be different):
    - */home/my_name/OIDv4_Toolkit/OID/csv_folder*
    - */home/my_name/OIDv4_Toolkit/OID/Dataset/train/Car_Bicycle_wheel_Bus*
- Open *Py* file ***converting-annotations.py*** in your *Programming Environment (PyCharm or any other you use)* and assign to the following variables found full paths:
    - `full_path_to_csv = ‘’`
    - `full_path_to_images = ‘’`

Converting annotations

When full paths were found, it is time for converting:

- Open *Py* file ***converting-annotations.py*** in your *Programming Environment (PyCharm or any other you use)*
- Write in the *list* classes’ names you downloaded images for (yours can be different). Pay attention on spelling. Names have to be the same as in *csv* file:

  ```
  labels = [‘Car’, ‘Bicycle wheel’, ‘Bus’]
  ```

- Run the code
- Open folder with images and check if *txt* files were created

Verify annotations by LabelIMG

After converting annotations into YOLO format, it is possible to check that calculations for bounding boxes were made correctly.

- Open folder with images and just created *txt* files with annotations
- Create one more *txt* file with name ***classes.txt*** (use any text editor like *notepad* or other) and in every separate line write classes' names that you downloaded images for (yours can be different):
  Car
  Bicycle wheel
  Bus
- Save changes and close the file ***classes.txt***
- Open *Terminal* (or *Anaconda Prompt*) and activate *environment* in which you installed *LabelIMG* tool
- Launch *LabelIMG* by one of the following command (depending on the way you chose for installation):
  `labelImg` (if pip was used)
  `python3 labelImg.py` (in other cases)
  `python labelImg.py` (in other cases)
- Go to *File --> Reset all* (it should close *LabelIMG*)
- Launch *LabelIMG* again
- Click on button *Open Dir* and navigate to the folder with images, annotations in *txt* files and just created file *classes.txt*
- By using *Next* and *Previous*, check if bounding boxes cover regions with needed objects

Useful Links

Check out these links with official resources for *OIDv4 toolkit* as well as the link to *Open Image* dataset and link to *LabelIMG tool*:

[1] OIDv4 ToolKit – official resource with full description

[2] Open Images dataset – publicly available huge dataset with labelled images from 600 classes

[3] LabelIMG – desktop tool for creating annotations in YOLO format