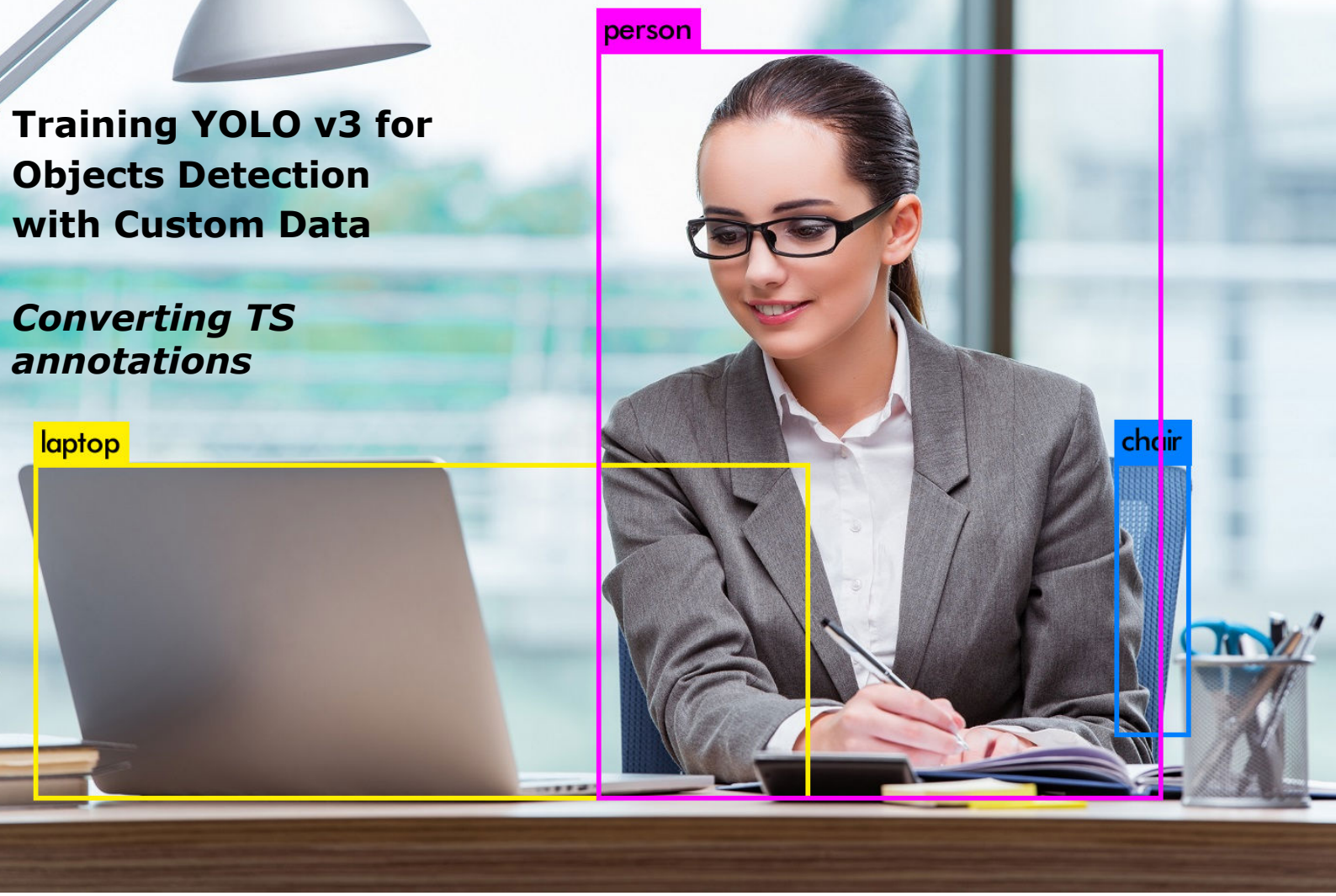

Training YOLO v3 for Objects Detection with Custom Data

Converting TS annotations

Converting Traffic Signs dataset into YOLO format

After downloading images of Traffic Signs and annotations, it is needed to convert given annotations into YOLO format. Annotations of bounding boxes' coordinates in *txt* file are as following:

	XMin	YMin	XMax	YMax

but YOLO needs following:

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

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

Download Py files into Traffic-Signs-Data

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

- *Traffic-Signs-Data/*
  - *getting-full-path.py*
  - *converting-ts-annotations.py*

Getting full path

Before converting annotations into YOLO format, it is needed to find *absolute* or *full path* to the *txt* file with annotation and *downloaded images* of Traffic Signs:

- Copy and paste *Py* file ***getting-full-path.py*** to the folder with *txt* file and *downloaded images*:
  - *Downloads/ts*
    (yours can be different)
- Open *Terminal* (or *Anaconda Prompt*) and activate your *Python v3* environment
- Go to the directory *Downloads/ts* and run following command:
  `python3 getting-full-path.py`
  or:
  `python getting-full-path.py`
- You should get full path like following (yours should be different):
  - */home/my_name/Downloads/ts*
- Open *Py* file ***converting-ts-annotations.py*** in your *Programming Environment (PyCharm or any other you use)* and assign to the following variable found full path:
  - `full_path_to_ts_dataset = ‘’`

Converting annotations

When full path was found, it is time for converting:

- Open *Py* file ***converting-ts-annotations.py*** in your *Programming Environment (PyCharm or any other you use)*
- 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 categories’ names that we used for Traffic Signs:
  prohibitory
  danger
  mandatory
  other
- 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 *German Traffic Sign Detection Benchmark*:

[1] Traffic Signs dataset – official resource with full description

[2] Archive of Traffic Signs – publicly available dataset with images of Traffic Signs