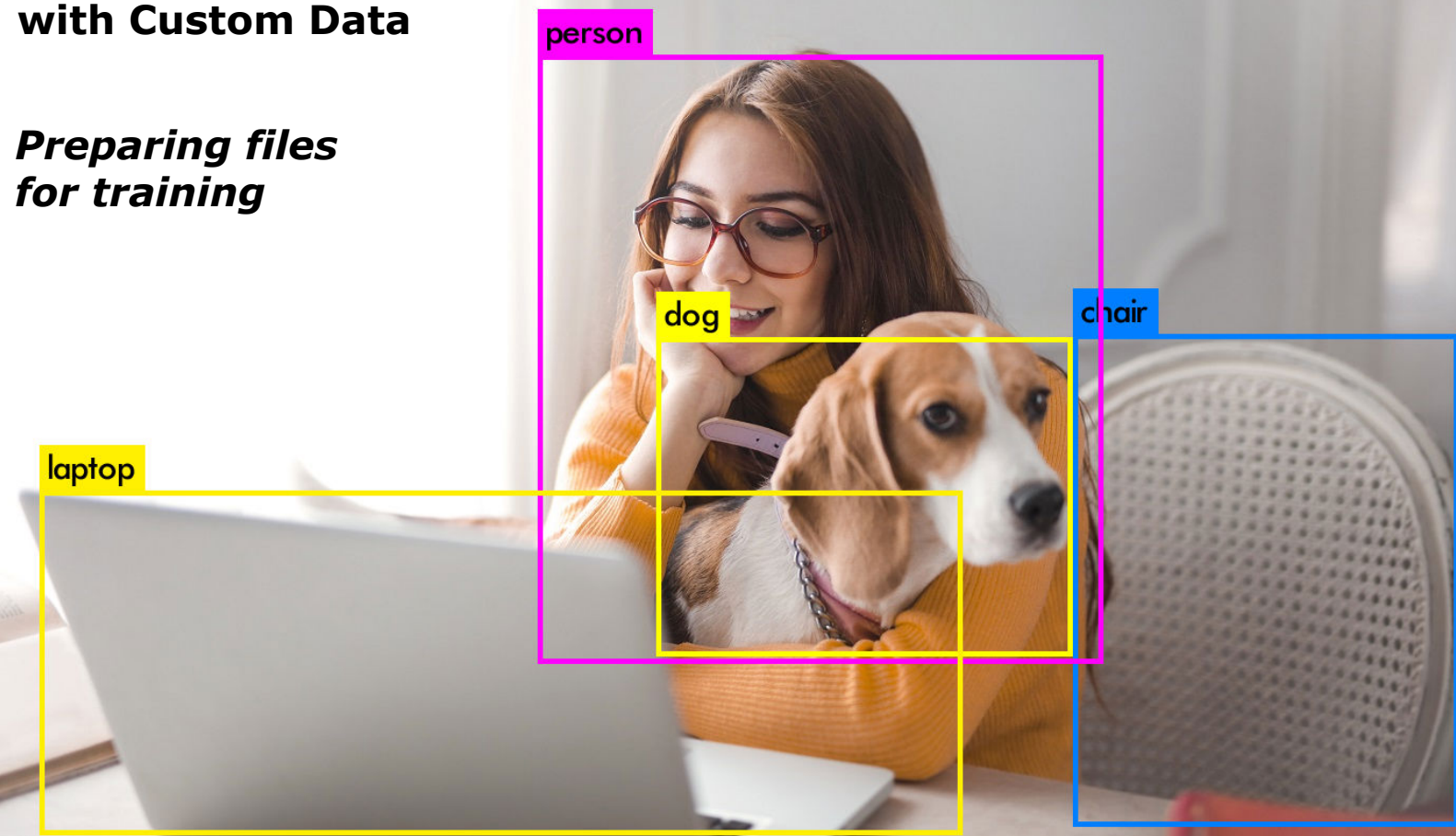


Training YOLO v3 for Objects Detection with Custom Data

Preparing files for training



Preparing files for training

After all images were downloaded and annotations were converted, it's time to prepare other files needed for training in *Darknet framework*.

These files are:

- custom_data.data
- classes.names
- train.txt
- test.txt

Five lines inside *custom_data.data* are:

- classes = 3
- train = /home/my_name/**train.txt**
- valid = /home/my_name/**test.txt**
- names = /home/my_name/**classes.names**
- backup = backup

First line specifies number of classes, namely, number of downloaded objects that *YOLO v3* will be trained on, and that will be used for detection after training.

Second line specifies full path to the file *train.txt* that in turn consists of full paths to the images for training. The same is true for **third line** with difference that images are used for validation during training.

Fourth line specifies full path to the file *classes.names* that has names of downloaded objects.

Fifth line specifies folder where trained weights will be saved.

Files *train.txt* and *test.txt* look like following (every path is in a new line):

- /home/my_name/downloaded-images/image001.jpg
- /home/my_name/downloaded-images/image002.jpg
- /home/my_name/downloaded-images/image003.jpg
- ...
- /home/my_name/downloaded-images/image799.jpg
- /home/my_name/downloaded-images/image800.jpg

File *classes.names* looks like following (classes' names and their number can be different):

- Car
- Bicycle wheel
- Bus

Download Py files into Custom-Data

In previous lecture we already created folder *Custom-Data*. Download *Py* files from *Resources* and copy them to this folder. You should have two new files appeared:

- *Custom-Data/*
 - *getting-full-path.py*
 - *converting-annotations.py*
 - *creating-train-and-test-txt-files.py*
 - *creating-files-data-and-names.py*

Getting full path

Before creating needed files to train in *Darknet framework*, it is needed to find *absolute* or *full path* to the directory with *downloaded images*:

- Copy and paste *Py* file **getting-full-path.py** to the folder with *downloaded images*
- Open *Terminal* (or *Anaconda Prompt*) and activate your *Python v3* environment and go to the directory with *downloaded images*. You can list all available sub-directories in the current directory by using following command in *Terminal* (or *Anaconda Prompt*):

```
dir
```

It will show all sub-directories you can go in. Go inside needed directory by using following command in *Terminal* (or *Anaconda Prompt*):

```
cd Downloads/downloaded_images
```

(yours should be different)

- Run following command in *Terminal* (or *Anaconda Prompt*):

```
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/OIDv4_Toolkit/OID/Dataset/train/Car_Bicycle_wheel_Bus`
- Open Py file ***creating-train-and-test-txt-files.py*** and Py file ***creating-files-data-and-name.py*** in your *Programming Environment* (PyCharm or any other you use) and assign to the following variable found full path:
 - `full_path_to_images = ''`

Creating files *train.txt* and *test.txt*

When full path was found, it is time for creating files *train.txt* and *test.txt*:

- Open Py file ***creating-train-and-test-txt-files.py*** in your *Programming Environment* (PyCharm or any other you use)
- Run the code
- Open folder with *downloaded images* and check if *txt* files were created

Creating files *custom_data.data* and *classes.names*

Next, it is time for creating files *custom_data.data* and *classes.names*:

- Open Py file ***creating-files-data-and-name.py*** in your *Programming Environment* (PyCharm or any other you use)
- Run the code
- Open folder with *downloaded images* and check if files were created