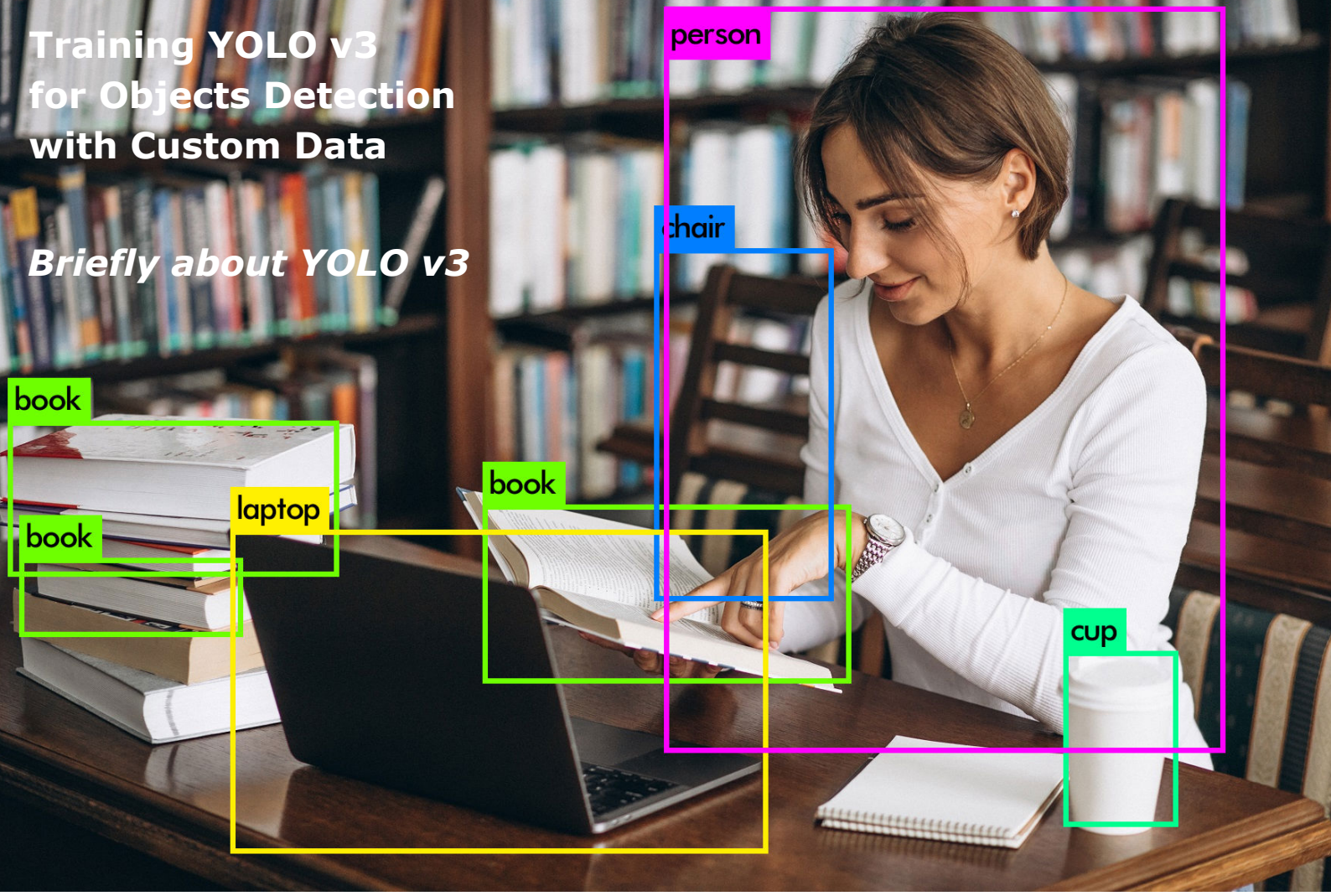

How YOLO works in few words

YOLO stands for **"You Only Look Once"** and uses convolutional neural networks (CNN) for object detection.

When **YOLO works** it **predicts classes' labels** and **detects locations of objects** at the same time. That is why, YOLO can detect multiple objects in one image. The name of the algorithm means that a single network just once is applied to whole image. YOLO **divides image into regions**, **predicts bounding boxes** and **probabilities** for every such region. YOLO also predicts **confidence for every bounding box** showing information that this particular bounding box actually includes object, and **probability of included object** in bounding box being a particular class. Then, bounding boxes are filtered with technique called **non-maximum suppression** that excludes some of them if confidence is low or there is another bounding box for this region with higher confidence.

YOLO-3 is the latest version that uses successive *3x3* and *1x1* convolutional layers. In total it has **53** convolutional layers with architecture as shown on the Fig.1 below. Every layer is followed by batch normalization and *Leaky ReLU* activation.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 1. Architecture of YOLO-3

Read more inside **original easy to understand presentation** and in **original papers** from official website with a lot of details and pictures. Links are in **Useful Links** section below.

What is inside configuration file?

Inside *yolov3.cfg* we have parameters that are used for training and testing. Some of them are described below. Read full description in **Useful Links** section below.

[net] section:

- `batch=64` – number of samples that will be processed in one batch
- `subdivisions=16` – number of *mini batches* in one batch; GPU processes *mini batch samples at once*; the weights will be updated for batch samples, that is 1 iteration processes batch images
- `width=608` – every image will be resized during training and testing to this number

- `height=608` – every image will be resized during training and testing to this number
- `channels=3` – every image will be converted during training and testing to this number

Optimization:

- `momentum=0.9` – hyperparameter for optimizer that defines how much history will influence further updating of weights
- `decay=0.0005` – decay the learning rate over the period of the training
- `learning_rate=0.001` – initial learning rate for training

Training:

- `angle=0` – parameter that randomly *rotates* images during training
- `saturation=1.5` – parameter that randomly *changes saturation* of images during training
- `exposure=1.5` – parameter that randomly *changes brightness* of images during training
- `hue=.1` – parameter that randomly *changes hue* of images during training

Useful Links

Check out additional links with detailed explanation of algorithm and other useful information for further reading:

[1] Author of the algorithm – **official website** with videos, papers and slides (go to section *Publications*)

[2] YOLO-1 explained – original PDF **paper** with details on how YOLO-1 works

[3] YOLO-1 explained – original and **easy to understand presentation** with a lot of figures and explanation how algorithm works

[4] YOLO-3 explained – original PDF **paper** explains the latest algorithm and its improvements

[5] yolov3.cfg – configuration file for YOLOv3-608 model

[6] yolov3.weights – pretrained weights with COCO dataset

[7] Config inside [net] section – more about used parameters inside **[net] section** of *yolov3.cfg*

[8] Config inside layers – more about used parameters inside **layers** of *yolov3.cfg*