**Cfg/yolov7.yaml**

nc: 80 # number of classes

Scaleing the network:

depth\_multiple: 1.0 # model depth multiple (more layer)

width\_multiple: 1.0 # layer channel multiple (more channel in a layer)

# anchors

anchors:

- [12,16, 19,36, 40,28] # P3/8

- [36,75, 76,55, 72,146] # P4/16

- [142,110, 192,243, 459,401] # P5/32

Anchor (Anchor boxes):

Anchor boxes are a set of predefined bounding boxes of a certain height and width. These boxes are defined to capture the scale and aspect ratio of specific object classes you want to detect and are typically chosen based on object sizes in your training datasets. (https://www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html)

**hyp.scratch.custom.yaml**

lr0: 0.01 # initial learning rate (SGD=1E-2, Adam=1E-3)

lrf: 0.1 # final OneCycleLR learning rate (lr0 \* lrf) /varying learning rate

momentum: 0.937 # SGD momentum/Adam beta1

weight\_decay: 0.0005 # optimizer weight decay 5e-4 /Weight decay is a regularization technique by adding a small penalty, usually the L1 or L2 norm of the weights/

warmup\_epochs: 3.0 # warmup epochs (fractions ok)

warmup\_momentum: 0.8 # warmup initial momentum

warmup\_bias\_lr: 0.1 # warmup initial bias lr

box: 0.05 # box loss gain

cls: 0.3 # cls loss gain

cls\_pw: 1.0 # cls BCELoss positive\_weight / weighted binary cross entropy/

obj: 0.7 # obj loss gain (scale with pixels)

obj\_pw: 1.0 # obj BCELoss positive\_weight

iou\_t: 0.20 # IoU training threshold

anchor\_t: 4.0 # anchor-multiple threshold

/ 1/anchor\_t < (size of gt\_box / size of anchor\_box) < anchor\_t/

# anchors: 3 # anchors per output layer (0 to ignore)

fl\_gamma: 0.0 # focal loss gamma (efficientDet default gamma=1.5)

/ <https://arxiv.org/abs/1708.02002/> helps for smaller object/

hsv\_h: 0.015 # image HSV-Hue augmentation (fraction)

hsv\_s: 0.7 # image HSV-Saturation augmentation (fraction)

hsv\_v: 0.4 # image HSV-Value augmentation (fraction)

degrees: 0.0 # image rotation (+/- deg)

translate: 0.2 # image translation (+/- fraction)

scale: 0.5 # image scale (+/- gain)

shear: 0.0 # image shear (+/- deg) //image distortion along an axis

perspective: 0.0 # image perspective (+/- fraction), range 0-0.001

flipud: 0.0 # image flip up-down (probability)

fliplr: 0.5 # image flip left-right (probability)

mosaic: 1.0 # image mosaic (probability)

mixup: 0.0 # image mixup (probability)

copy\_paste: 0.0 # image copy paste (probability)

paste\_in: 0.0 # image copy paste (probability), use 0 for faster training

loss\_ota: 1 # use ComputeLossOTA, use 0 for faster training

/OTA: Optimal Transport Assignment (https://arxiv.org/abs/2103.14259) for more accurate label assignment/