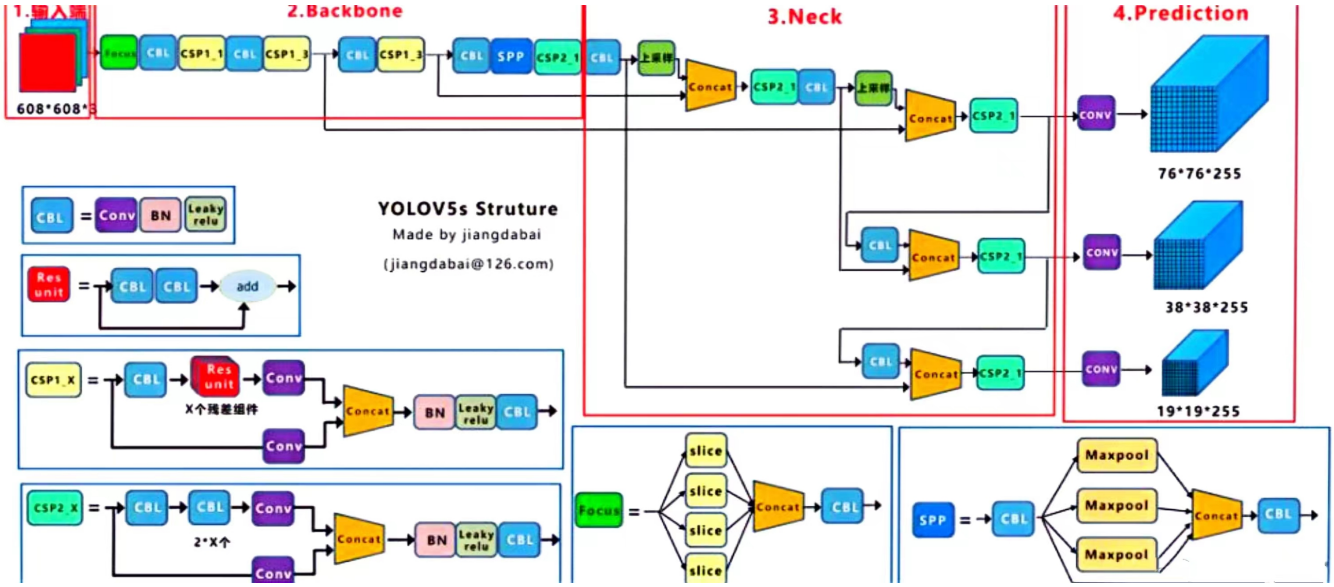
一.YOLOv5 s Structure

特点：适合于移动端部署，模型小、速度快
有smlx四个版本，不同的是depth_multiple和width_multiple两个参数
网络架构：
1.Input：Mosaic数据增强、自适应锚框计算、自适应图片缩放
2.Backbone:Focus结构、CSP结构
3.Neck：FPN+PAN结构
4.Head:CLOU_Loss

其中：
Mosaic数据增强：增加训练数据的多样性。他通过将多个图像拼接成一个大图像，原始图像的目标标签需要重新计算坐标和调整位置，生成新的训练样本，增加训练数据的多样性。
Mosaic数据增强在于它能模拟真实世界更复杂的场景-多个目标出现在同一图像中、不同目标之间的空间关系和上下文信息，泛化能力和鲁棒性。
鲁棒性：对不确定因素的适应能力。
泛化能力：对新数据的适应能力。



自适应锚框计算：锚框（Anchor Boxes）是用于目标检测任务中的一种技术，它们用于预测目标的位置和大小。自适应锚框计算指的是根据训练数据集中目标的尺寸和形状分布来动态地计算和调整锚框的大小和比例，以便更好地适应不同尺度和形状的目标。

自适应图片缩放：自适应图片缩放是根据模型的输入要求或者训练数据集的特点，动态地调整图像的尺寸和大小，使其适合输入到模型中进行训练或推断。这种预处理技术可以有效地处理不同尺寸和比例的图像数据。

卷积：提取图片特征，生成输出特征图。
池化：减少特征图的尺寸但保留下来特征图的重要特征信息。

基本组件：
Focus：一种特殊的卷积结构，用于处理输入图像。帮助模型更好捕获小目标的细节信息。
它把输入图像分成多个部分，并在每个部分执行卷积操作，从而提高模型的感受野和信息提取能力。
CSP：把网络分为两个部分，每部分包含多个密集链接的卷积层，旨在提高信息的传递和特征的重用。

二.输入端

（1）MOSAIC数据增强：

MOSAIC数据增强算法将四张图片按照一定的比例组合成一张图片，使模型在更小的范围内识别目标。

步骤：

1.随机选取图片拼接基准点坐标，另随机选取四张图片；

2.四张图片根据基准点，分别经过尺寸调整和比例缩放后，放置在指定尺寸的大图的左上、右上、左下、右下位置；

3.根据每张图片的尺寸变换方式，将映射关系对应到图片标签上；

4.依据指定的横纵坐标，对大图进行拼接。处理超过边界的检验框坐标。

优点：

1.丰富数据集：随机使用4张图像，随机缩放后随机拼接，增加了很多小目标，大大增加了数据多样性。

2.增强模型鲁棒性：混合四张具有不同语义信息的图片，可以让模型检测超出常规语境的目标

3.加强批归一化层（Batch Mormalization）的效果：当模型设置BN操作后，训练时会尽可能增大批样本总量（BatchSize）。因为BN原理是计算每一个特征层的均值和方差，如果批样本总量越大，那BN计算的均值和方差就越接近整个数据集的均值和方差，效果越好。

（2）自适应锚框计算：

YOLOv3、YOLOv4，对于不同的数据集，都会计算先验框anchor，然后在训练时网络会在anchor的基础上进行预测，输出预测框，再和标签框进行比较，最后进行反向传播。

在YOLOv3、YOLOv4中训练不同数据集时，是使用单独的脚本进行初始锚框的计算，在YOLOv5中，则是将此功能嵌入到整个训练代码里。所以每次训练开始前，它都会根据不同的数据集来自适应计算anchor。

如果觉得计算的锚框效果不好，也可以在代码中将这个功能关闭。<是关闭自适应锚框自己手动调整锚框的尺寸>

自适应计算步骤：

①获取数据集中所有目标的高和宽；

②将每张图片中按照等比例缩放到resize指定大小，这里保证宽高中的最大值符合指定大小；

③将bboxes从相对坐标改成绝对坐标，这里乘以的是缩放后的宽和高；

④筛选bboxes，保留宽高都大于等于两个像素的bboxes。

⑤使用k-means聚类算法得到n个anchors

⑥使用遗传算法随机对anchors的宽高进行变异。倘若变异效果好，就将变异后的结果赋值给anchors。如果变异后效果变差就跳过，默认变异1000次，这里使用的是anchor_fitness方法计算得到的适应度fitness，然后再进行评估。

（3）自适应图片缩放

步骤：

①根据原始图片大小以及输入到网络的图片大小计算缩放比例

②根据原始图片大小与缩放比例计算缩放后的图片大小

③计算黑边填充数值

tips：

->YOLOv5中填充的是灰色，即（114，114，114）

->训练时没有采用缩减黑边的方式，还是采用传统填充的方式，即缩放到416 * 416大小。只是在测试使用模型推理时，才采用缩减黑边的方式，提高目标检测，推理的速度。

->为什么np.mod函数后面用32？

因为YOLOv5网络经过5次降采样，2的5次方=32.所以至少要去掉32倍，再进行取余。以免产生尺度太小走不完stride（filter在原图上扫描时需要跳跃的格数）的问题，再进行取余。

三.Backbone

1.Focus结构

Focus模块是YOLOv5进入**Backbone**前，对图片进行切片操作，具体操作是在一张图片中每隔一个像素拿到一个值，类似于临近降采样，这样就拿到了4张图片，4张图片互补，得到和原来大小差不多的图片但是没有信息丢失。这样一来，将W、H信息就集中到了通道空间，输入通道扩充了4倍，即拼接起来的图片相对于原先的RGB三通道模式就变成了12个通道，最后将得到的新图片再经过卷积操作，最终得到了没有丢失信息情况下的二倍降采样特征图。

作用：

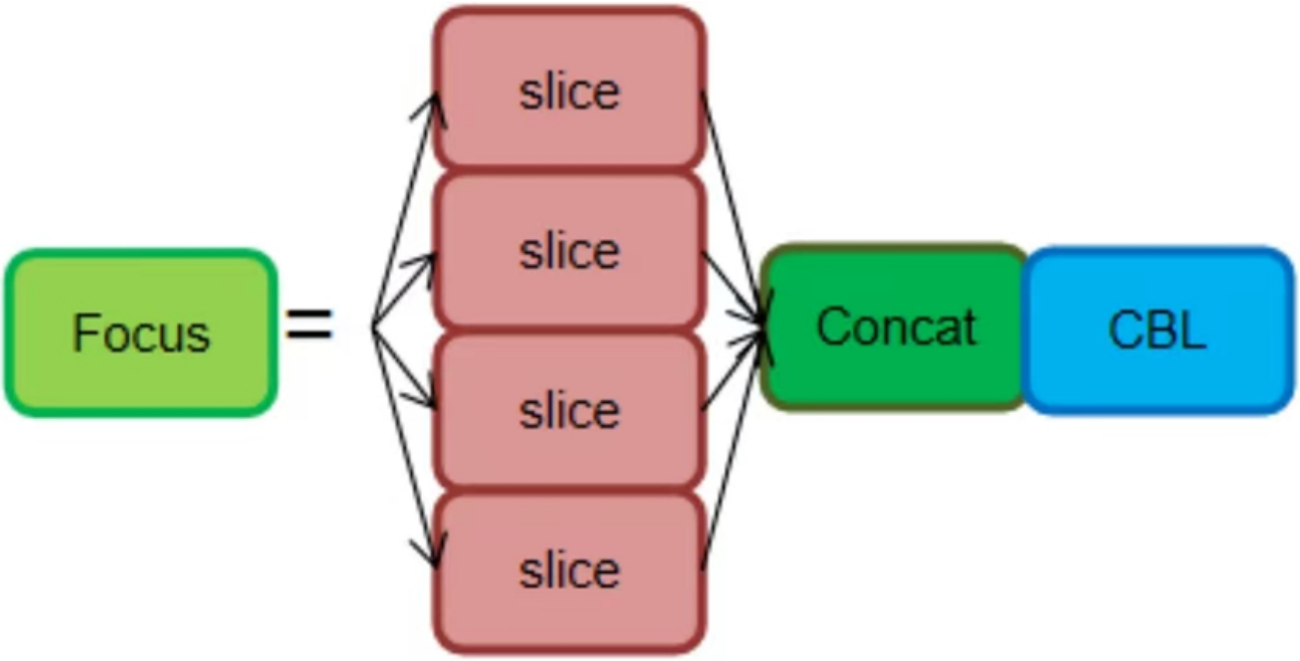
可以使信息不丢失情况下提高算力；

有助于增加网络的感受野（receptive field），从而更好的捕抓图像的细节和特征。它可以帮助网络更好的识别和定位目标，尤其是小目标或者密集目标的检测任务，有着显著的改善效果。 **为什么信息不会丢失**

因为采样间隔固定，保证了信息的连续性和一致性。

不足：

Focus对某些设备不支持且不友好，开销很大，另外切片对不齐的话模型就崩了。



后期改进： 在新版中，YOLOv5 将Focus 模块替换成了一个 6 x 6 的卷积层。两者的计算量是等价的，但是对于一些 GPU 设备，使用 6 x 6 的卷积会更加高效。

(2)CSP结构

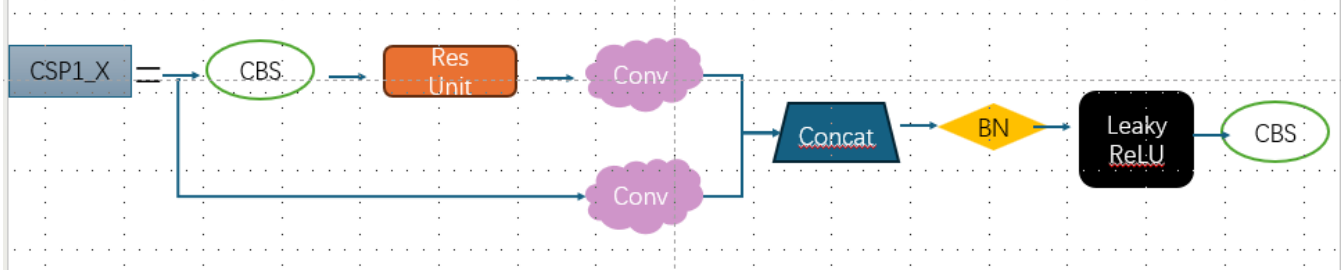
YOLOv5的Backbone网络中运用了CSP1_X结构,在Neck网络中运用了CSP2_X结构。

一些术语补充:

这些三个术语都是针对特征图来说 上采样upsampling：增加信号或数据的采样率，从而增加空间分辨率；通常用来实现特征图的尺寸扩大，以便与其他层的特征图尺寸相匹配，或者用于生成高分辨率的输出。<插值、转置卷积>

下采样downsampling：减少信号或数据的采样率，从而减少空间的分辨率；通常用来实现减少特征图尺寸，以减少模型参数和计算量，并增加网络的receptive field.<池化、步长大于1的卷积>

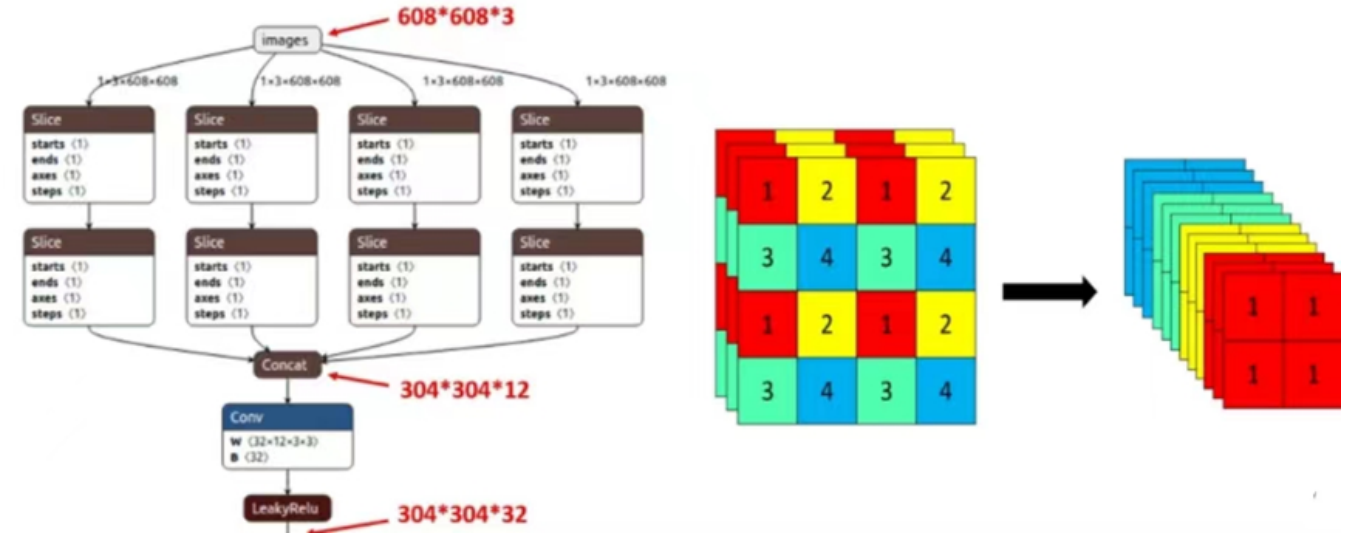
降采样subsampling：描述对信号或数据进行采样率的降低，无论是在时空域还是频率域。



①CSP1_X：

将输入分为两个分支，一个分支先通过CBS，再经过多个残差结构，再进行一次卷积；另一个分支直接进行卷积；然后两个分支进行concatenate，再经过BN（正态分布），再来一次激活函数（之前的版本是Leaky，后期变成SiLU），最后进行一个CBS。

CSP1_X应用于backbone主干网络部分，backbone是较深的网络，增加残差结构可以增加层与层之间之间反向传播的梯度值，避免因为加深而带来的的梯度消失，从而可以提取到更细颗粒度的特征并且不用担心网络退化。



· Cues

· Notes

将输入分为两个分支

Resunit 是 X 个残差组件

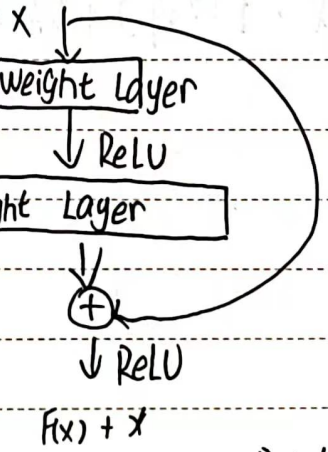
Resnet

增加网络层数 ⟶ 复杂的特征提取

(堆叠网络)

↳ 网络准确度出现饱和
不是因为 overfitness
而是堆叠过多网络难以收敛，且
梯度爆炸(消失)
训练、测试 error ↑

feature map: 特征映射



Rectified Linear Unit

非线性修正函数

tanh、Sigmoid、ReLU☆

Conv 线性 解决问题有限

↳ 对每个像素点进行点乘，并用 0 来替换负值像素点.

Residual Learnig:

a building block

· Summary

F(x) 和 X 联合获取特征，取极端 F(x)=0 (堆积结构学习无效果)，这里指不来堆积层仅做恒等映射，至少网络性能不会下降

residual: $F(x) = H(x) - X$

new feature: $H(x) = F(x) + X$

· Cues

设计规则：

· Notes

stride 步伐

Resnet 中每两层之间增加了短路机制，形成残差学习.
实线-快捷连接 虚线-feature map 数量发生改变.
~~(1) 输入和输出~~

(1) Input, output : dimension same

→ Identity Quick Connection $y = F(x, \{W_i\}) + x$

(2) Dimension increase :

→ Identity Quick connection Add additional zero inputs to increase dimensions.

$y = F(x, \{W_i\}) + W_s x$.

mmary

SHANBAY

在目标识别中通常使用一个模块化的结构，包括backbone、neck和head三个主要部分。

Backbone：模型的基础，**负责从输入图像中提取特征**。<卷积层、池化层、归一化层组成，用于逐渐减少图像的空间尺寸并增加特征的深度><不同的主干网络设计影响这模型对图像特征的抽取能力和表达能力，常见的主干网络有ResNet、VGG.....>

Neck:用于进一步处理和整合从主干网络中提取的特征。它常常由<卷积层、池化层和特征融合>组成。它能增强特征的表达能力，并为头部网络提供更丰富和抽象的特征表示，以便于后续的任务处理。

Head：负责执行具体的任务，如分类、检测或分割等。<常由全连接层、卷积层和激活函数组成>

四.Neck

YOLOv5的neck和YOLOv4一样都采用FPN+PAN的结构。但在它的基础上做了一些改进：YOLOv4的neck结构中采用的都是普通的卷积操作，而YOLOv5的neck中采用的是CSPNet设计的CSP2结构，从而加强了网络特征融合能力。

FPN自顶向下传达语义特征，PAN塔自底向上传达定位特征。

处理过程：

1.FPN处理：输入的特征图首先通过FPN，这个过程中会生成多尺度的特征金字塔，即在不同的层级上提取特征，以便网络可以在多个尺度上进行目标检测。

2.PAN处理：接下来FPN生成的多尺度特征图会被送入PAN，PAN负责在不同层级之间传递和聚合特征。这样可以使得网络更好的理解图像的语义信息，提高目标检测的精度。

3.CSP2_X处理：最后CSP2_X会进一步增强特征融合能力，它通过将输入特征图分为两部分，一部分进行卷积操作，另一部分保持不变。然后将卷积后的部分与保持不变的部分进行拼接，这样可以增加网络的非线性化程度，提高特征融合的效果。

CSPNet(Cross Stage Partial Network)用来解决因信息瓶颈导致特征图的分辨率逐渐降低，特征表达能力受限等问题。

导致信息瓶颈的可能原因：

①特征图尺寸减少：当特征图经历多次步幅较大的卷积和池化操作，这会导致尺寸减小导致信息丢失，因为较小的特征图可能无法有效的表达图像的细节和语义信息。

②特征的压缩和抽象：在网络的深层结构中，特征通常会经历多次卷积和池化操作，这会导致特征信息被压缩和抽象化。虽然这种抽象能力有助于网络学习更高级别的特征，但也可能导致部分图像信息的丢失或模糊化。

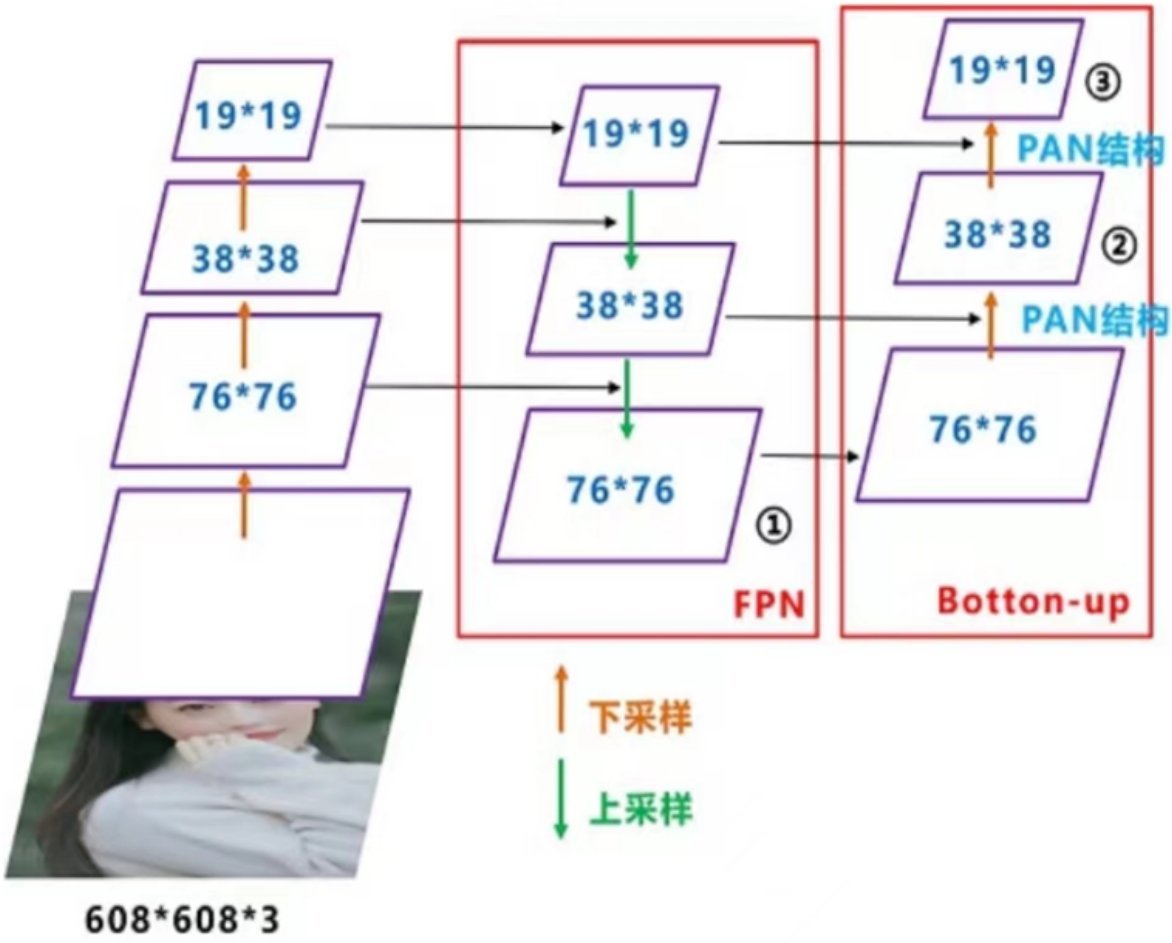
③梯度消失或梯度爆炸：在深层网络中，梯度消失或梯度爆炸是常见的问题。梯度消失指的是在反向传播过程中，梯度值变得非常小，导致网络参数无法有效更新。梯度爆炸则是梯度值变得非常大，使得参数更新过程不稳定。这些问题可能会导致网络无法充分学习和表达复杂的特征。

③梯度消失或梯度爆炸： 在深层网络中，梯度消失或梯度爆炸是常见的问题。梯度消失指的是在反向传播过程中，梯度值变得非常小，导致网络参数无法有效更新。梯度爆炸则是梯度值变得非常大，使得参数更新过程不稳定。这些问题可能会导致网络无法充分学习和表达复杂的特征。

梯度消失：当网络深度较大时梯度在反向传播的过程中被多次相乘导致梯度值指数级的减小。当梯度接近0时，网络参数几乎得不到更新，导致网络无法学习到正确的特征表示。

梯度爆炸：因为网络中存在参数之间的相互依赖关系，导致梯度在反向传播的过程中指数级增长，导致参数更新过程中不稳定，甚至使网络的参数值设置不合理。

梯度问题和网络结构、激活函数的选择、参数初始化有关。



五.Head

（1）Bouding box损失函数

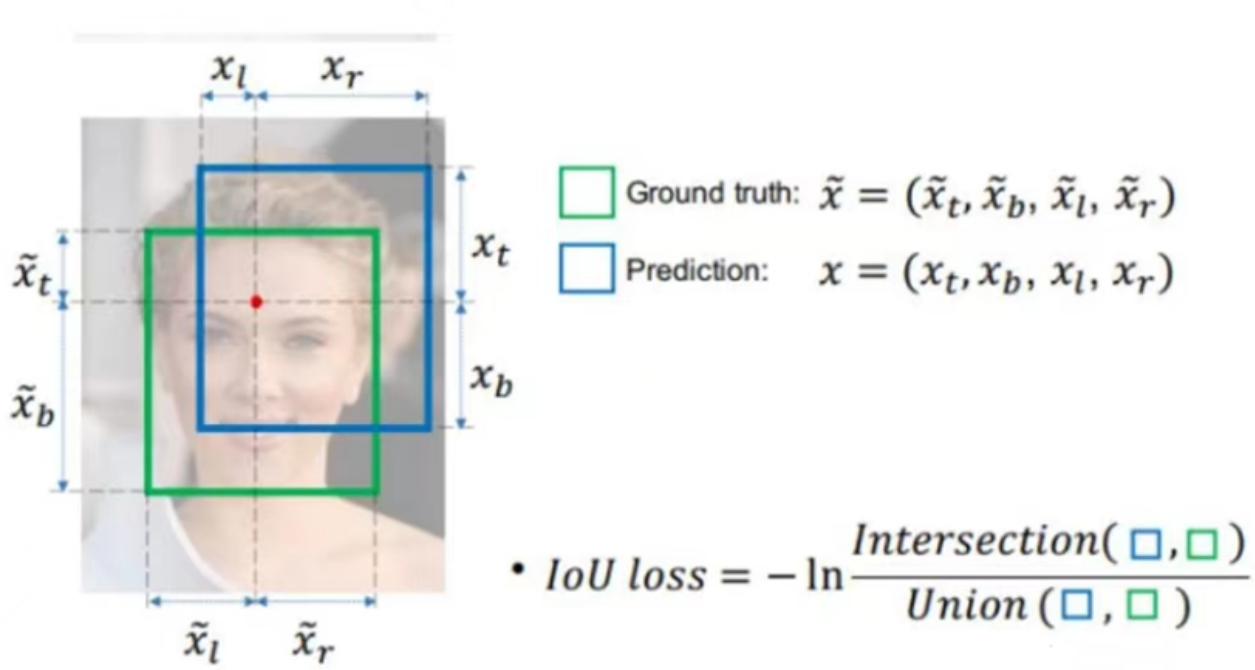
YOLOv5采用**CLOU_LOSS**作为bounding box损失函数

CIOU（Complete IoU）损失：

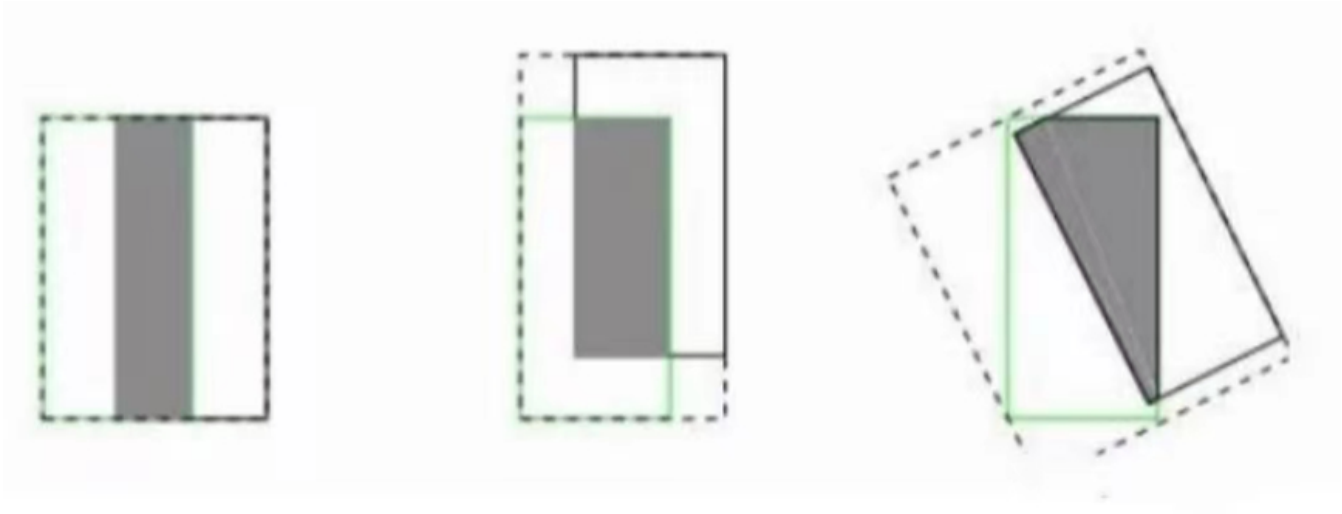
在学习CLOU_LOSS前先让我们学习一些先导知识：

经典IOU loss：

大部分的检测算法都是用它



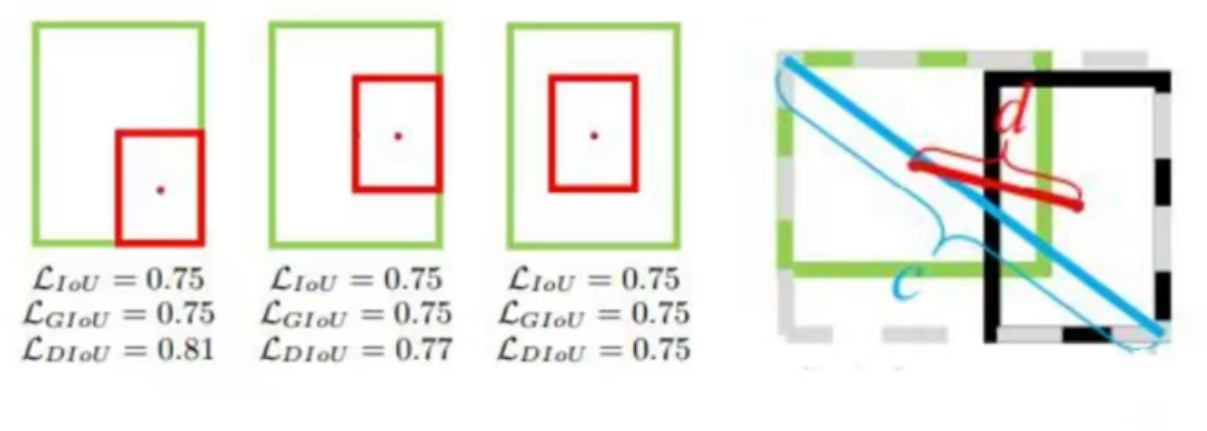
不足：没有相交则IOU=0无法梯度计算，相同的IOU却无法实际的情况。



GIOU（generalized iou）损失：
当检验框和真实框没有出现重叠的时候iou的loss都是一样的
因此Giou就引进了最小封闭形状C（C可以把A,B包含在内），在不重叠情况下能让预测框尽可能朝着真实框前进，这样就能解决检验框和真实框没有重叠问题。

不足：在两个预测框完全重叠的情况下，不能反映出实际情况。

DIOU（Distance IOU）损失：
DIOU考虑到GIOU的缺点，也是增加了C检测框，将真实框和预测框都包含了进来，但DIOU的计算不是框之间的交并，而是计算每个检验框的欧式距离。



好啦接下来我们来看CIOU（complete iou）损失

·CIoU就是在Diou的基础上**增加了检验框尺度的loss、增加了长和宽的loss**,这样的话预测框就会更加复合真实框。

loss function：必须考虑三个几何因素重叠面积、中心点距离、长宽比。

$$CIOU = IoU - \frac{p^2}{c^2} - \alpha \cdot v$$

其中，

- IoU是传统的交并比，表示预测边界框与真实边界框的重叠面积与并集面积之比。
- p^2 表示边界框之间中心点距离的平方。
- c^2 表示边界框之间最小闭合框的对角线长度的平方。
- v 表示边界框之间宽高比之差的平方。
- α 是一个平衡系数。

总结：

·IOU_LOSS:主要考虑检测框和目标框重叠面积。

·GIOU_LOSS：在IOU的基础上解决边界框不重合时的问题。

·DIOU_LOSS：在GIOU的基础上，考虑边界框中心点距离的信息|转为 欧式距离问题 ·CIOU_LOSS:在DIOU的基础上，考虑边界框宽高比的尺度信息。

（2）NMS非极大值抑制

NMS的本质是搜索局部极大值，抑制非极大值元素。在目标检测中检测器可能会在同一个目标周围生成多个相似的边界框。为了避免这种情况，NMS被用来选择最合适的边界框，以确保每个目标只被标记一次，消除的冗余边界框。

算法流程：

1.对所有预测框的置信度降序排序

2.选出置信度最高的预测框，确认其为正确预测，并计算它与其他预测框的iou

3.根据步骤2中计算的IOU去除重叠度高的，IOU>threshold阈值就直接删除

4.剩下的预测框返回第一步，直到没有剩下的为止。

·**SoftNMS：**

当两个目标靠得非常紧时，置信度低的会被置信度高的框所抑制，那么当两个物体靠得非常近时就只会识别出一个bbox。为了解决这个问题，采用softNMS。Soft-NMS不是直接将重叠的边界框抑制掉，而是通过降低重叠边界框的得分来保留它们，从而在一定程度上保留了所有重叠的边界框。这样可以更好地处理靠得很近的目标，减少了对边界框的过度抑制，从而提高了目标检测的准确性和鲁棒性。

六.训练策略

1.多尺度训练（Multi-scale training）。 如果网络的输入是416 *416，那么选练的时候就会从0.5* 416到1.5 * 416中任意取值，但所取得值都是32的整数倍。
帮助模型适应不同大小的目标，这样有助于提高模型的泛化能力和鲁棒性。

2.训练开始前使用warmup进行训练。
在训练开始阶段，使用较小的学习率进行训练，然后逐渐增加学习率，这样可以帮助模型更快的收敛到合适的参数空间，加速训练过程。
3.使用cosine学习率下降策略（Cosine LR schduler）
使用余弦函数调整学习率，这种策略在训练后期可以更加平缓地降低学习率，有助于模型更稳定地收敛到最优解。
4.采用EMA更新权重（Exponential Moving Average） 相当于训练时给参数赋予了一个动量，这样更新起来就会更加平滑。
5.使用了amp进行混合精度训练（Mixed precision）。 能够减少显存的占用并且加快训练速度，但需要GPU支持。

YOLOv5与前YOLO系列相比的改进：
·（1）增加了正样本：方法是领域的正样本anchor匹配策略。
·（2）通过灵活的配置参数，可以得到不同复杂度的模型。
·（3）通过一些内置的超参优化策略，提升整体性能
·（4）采用了mosaic增强，提升了对小物体的检测性能。