Yolov4

Yolov4 相比于 Yolov3 的创新包括输入端创新、BackBone 主干网络创新，Neck 层改进以及预测改进，其中输入端创新主要包括 Mosaic 数据增强、cmBN、SAT 自对抗训练；BackBone 更新主要包括 CSPDarknet53、Mish 激活函数、Dropblock；Neck 层的改进其实就是加入了一些组合层，实验效果不错就留下了，包括 SPP 和 FPN+PAN 等；最后 Prediction 改进锚框的损失，包括 CIoU 和 DIoU。其实在 Yolov4 提出前，Yolov3 已经取得了不错的效果，Yolov4 相比于 Yolov3 的改进，更多是增加了一些训练 tricks，包括 BoF(Bag of freebies) 和 BoS(Bag of Specials)，因此这里主要介绍一下各种 tricks，虽然 YOLOv4 中因为实验结果遗弃了一些 tricks，但这只是因为不适合本文的任务，可能在别的任务里能取得不错的效果，因此这里都做了说明分析。

Bag of Freebies：指的是那些不增加模型复杂度，也不增加推理的计算量的训练方法技巧，来提高模型的准确度：

1. 数据增强：

1.1 通过调整输入图片的亮度、对比度、色调、饱和度、添加噪声以及进行随机翻转、旋转、缩放和裁剪操作进行数据增强。

1.2 通过物体遮挡增强数据，包括 Random Erase, Cutout, Hide-and-seek, Grid mask。Random Erase 称为随机擦除，随机选择一个区域，采用随机值进行覆盖，所选区域一定在图像内。cutout 随机选择一个固定大小的正方形区域，像素值全部用 0 填充即可，但为了避免 0 填充对训练数据有影响，需要对训练数据进行归一化处理，norm 到 0。Hide-and-seek 将图像切分成 S*S 网络，每个网络使用一定概率进行遮挡，从而模拟出 Random Erase 和 Cutout 的效果。Grid mask 结构化了 drop 操作，例如均匀分布地删除正方形区域，并且通过控制密度和 size 参数，来达到平衡，密度就是正方形个数，size 就是正方形大小。把物体遮挡一部分也能被正确识别分类，这样就迫使网络利用局部数据进行训练识别，增大了训练难度，一定程度提高模型的泛化能力。

1.3DropOut、DropConnect、DropBlock，通过在特征图丢失一些信息防止模型过拟合，达到数据增强的效果。DropOut 是在训练过程中随意按概率丢弃一些神经

元，丢弃的神经元输出置 0，DropConnect 是在训练过程中把神经元的输入权值按概率置 0，DropBlock 对整个局部区域进行删减丢弃，目的都是为了增强网络对数据的拟合能力，防止过拟合。

1.4 多张图进行数据增强, MixUP, CutMix, Mosaic。MixUp 使用两个图像与不同的系数比率相乘叠加，然后用这些叠加的比率调整标签。也就是就是将两张图片采用比例混合，label 也按照比例混合。CutMix 是将裁剪后的图像覆盖到其他图像的矩形区域，并根据混合区域面积的大小调整标签, 也就相当于是是 CutOut 和 Mixup 的结合，将图片的一部分区域擦除并随机填充训练集中的其他数据区域像素值。Mosaic 是 CutMix 的拓展，CutMix 是混合两张图，Mosaic 数据增强是混合了四张具有不同语义信息的图片，可以让检测器检测出超乎常规语境的目标，增强鲁棒性，并减少对大的 mini-batch 的依赖。上述三种数据增强还是取决于具体任务，改变任务时感觉会有些反效果。

1.5 Style Transfer GAN，原文认为通过 GAN 随机将物体的纹理替换，可以增强网络的学习特征能力，作者认为 CNN 训练学习到的实际是纹理特征（texture bias）而不是形状特征，这与人类的常识相反，因而通过引入风格化的 ImageNet 数据集，平衡纹理和形状偏置，提高模型的泛化能力。

2. 数据分布不平衡：

2.1 hard negative example mining: 一般用在 fastrcnn 等 two-stage 网络中，主要是困难负样本难挖掘，在目标检测中我们会事先标记好 ground_truth，接下来在图片中随机提取一系列 sample，与 ground_truth 重叠率 IoU 超过一定阈值的(比如 0.5)，则认为它是 positive sample，否则为 negative sample，考虑到实际负样本数远多于正样本数，我们为了避免 network 的预测值少数服从多数而向负样本靠拢，取正负样本大约 1:3，显而易见，用来训练网络的负样本为提取的负样本的子集，因而当然选择负样本中容易被分错类的困难负样本来训练。之后定义困难负样本：困难负样本是指那些容易被网络预测为正样本的 proposal，即假阳性(false positive)，如 roi 里有二分之一个目标时，虽然它仍是负样本，却容易被判断为正样本，这块 roi 即为 hard negative，训练 hard negative 对提升网络的分类性能具有极大帮助，因为它相当于一个错题集。为了确定负样本，通常先用初始样本集(即第一帧随机选择的正负样本)去训练网络，再用训练好的网络去预测负样本集

中剩余的负样本，选择其中得分最高，即最容易被判断为正样本的负样本为困难样本，加入负样本集中，重新训练网络，循环往复。

2.2 online hard example mining: 与困难负样本挖掘相似，一般也用在 two-stage 网络中，在网络前向训练时全部的 ROI 通过网络，根据 loss 排序，之后反向回传会根据排序，选择 B/N 个 loss 值最大的（worst）样本来后向传播更新 model 的 weights

2.3 Focal Loss:主要是为了解决 one-stage 目标检测中正负样本比例严重失衡的问题。该损失函数降低了大量简单负样本在训练中所占的权重，也可理解为一种困难样本挖掘。将分类的交叉熵损失函数替换为：

$$\mathrm{L}_{fl}=\begin{cases}-\alpha\left(1-y^{'}\right)^{\gamma}\log y^{'}, & y=1\\ -(1-\alpha)y^{'\gamma}\log\left(1-y^{'}\right), & y=0\end{cases}$$

其中：$\gamma>0$ 使得减少易分类样本的损失。使得更关注于困难的、错分的样本。

例如 γ 为 2，对于正类样本而言，预测结果为 0.95 肯定是简单样本，所以（1-0.95）的 γ 次方就会很小，这时损失函数值就变得更小。而预测概率为 0.3 的样本其损失相对很大。对于负类样本而言同样，预测 0.1 的结果应当远比预测 0.7 的样本损失值要小得多。对于预测概率为 0.5 时，损失只减少了 0.25 倍，所以更加关注于这种难以区分的样本。

3. one-hot 类别之间没有关联：

3.1 Label Smoothing: 思想就是对 one-hot 的 label 进行 soft 操作，比如[0 1]的 one-hot 编码，经过 label smooth 变成[0.05 0.95]。label smooth 核心就是对 label 进行 soft 操作，不要给 0 或者 1 的标签，而是有一个偏移，相当于在原 label 上增加噪声，让模型的预测值不要过度集中于概率较高的类别，把一些概率放在概率较低的类别。

3.2 知识蒸馏: 用于获得更好的 soft label。

4. BBox 回归损失：

4.1 IoU Loss: 考虑了预测 BBox 区域和地面真实 BBox 区域的覆盖范围。由于 IoU 是尺度不变的表示，可以解决传统方法在计算{x, y, w, h}的 L1loss 或者 L2loss 时，损失会随着尺度的增大而增大的问题。

$$IoULoss = -ln\frac{Intersection(bbox_1, bbox_2)}{Union(bbox_1, bbox_2)}$$

4.2 GIoU Loss: GIoU 损失是在覆盖区域之外还包括对象的形状和方向。他们提出寻找能够同时覆盖预测的 BBox 和 ground truth BBox 的最小面积 BBox，并用这个 BBox 作为分母来代替原来 IoU loss 中使用的分母。

$$GIoU = IoU - \frac{A^c - u}{A^c}$$

上式中 A^c 表示两个 bbox 框住的大 BBox 的区域，u 表示两个 bbox 的并集。

4.3 DIoU Loss: 好的目标框回归损失应该考虑三个重要的几何因素：重叠面积，中心点距离，长宽比。DIoU Loss,相对于 GIoU Loss 收敛速度更快，该 Loss 考虑了重叠面积和中心点距离，但没有考虑到长宽比。

$$DIoU = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2}$$

其中，b 和 b^{gt} 分别表示两个 bbox 的中心点，$\rho()$ 表示欧式距离, c 表示两个 bbox 的最小外界矩形的对角线距离。

4.4 CIoU Loss: 同时考虑了重叠面积、中心点之间的距离和长宽比。CIoU 可以在 BBox 回归问题上获得更好的收敛速度和精度。CIou Loss 改进了 DIOU Loss，把重叠面积，中心点距离，长宽比都考虑进来了。

$$L_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2$$

Bag of Specials：指的是那些增加少许模型复杂度或计算量的训练技巧，但可以显著提高模型的准确度：

1. 增强感受野：目标检测网络的精度和网络的感受野大小密切相关，感受野过小可能无法识别大尺度物体，从而导致目标检测网络精度下降。

1.1 SPPNet，空间金字塔池化网络，由于卷积层在面对不同尺度的输入图像时，

会生成大小不一样的特征图，会对网络的训练带来较大的麻烦，因而 SPPNet 就通过将卷积层最后一层的池化层替换为金字塔池化层，固定输入的特征向量的大小。如下图所示，金字塔池化层就是固定了池化层的大小和数量，在面对不同尺度的特征图时，也能通过金字塔池化层来固定输出特征向量的大小（图中使用了 3 个不同大小的池化层，分别是 4×4，2×2 和 1×1，这样每张特征图提取的特征向量长度就是 4×4+2×2+1×1=21）

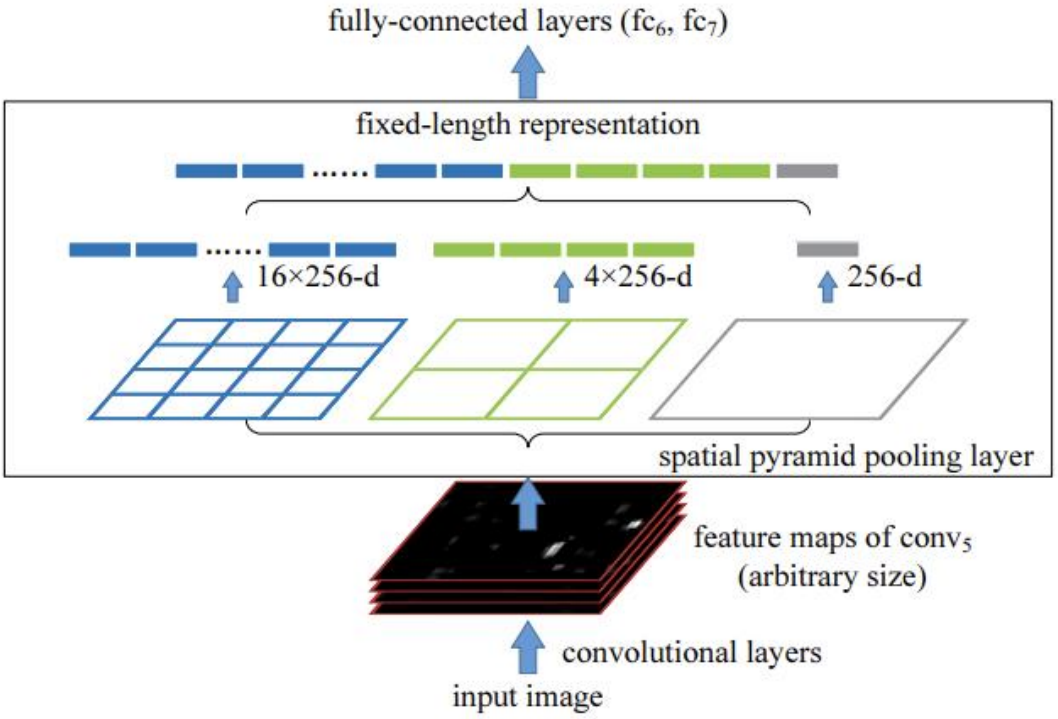

图 1. SPPNet

1.2 ASPP: 空洞空间金字塔池化网络，其实就是把特征图通过以不同采样率的空洞卷积并行采样，对每个采样率提取到的特征在不同的分支中进一步处理后融合从而产生最后的结果，如下图所示：

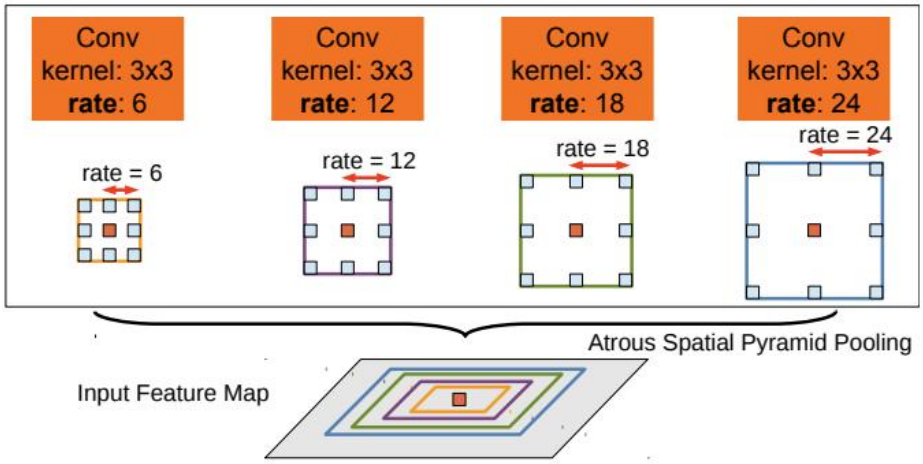

Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

图 2. ASPP

1.3 RFB: RFB 的每个分支上使用不同尺度的常规卷积 + 空洞卷积，通过常规卷积的不同卷积核尺度来模拟 pRFs 中的不同感受野，各个分支上通过各自 dilated conv 所得到的离心率，来模拟 pRF 的尺度与离心率的比例，其实就是为了模拟人类的视觉感知模式。由图可知，RFB 其实就是先使用 1×1 卷积降低特征图通道数，并行使用 3 个不同大小的正常卷积提取特征图的特征信息，然后再接与正常卷积大小相同的 dilated rate 的 3×3 空洞卷积，最后再拼接起来，使用 1×1 卷积恢复通道数。（类似 bottleneck 结构，只不过增加了空洞卷积）

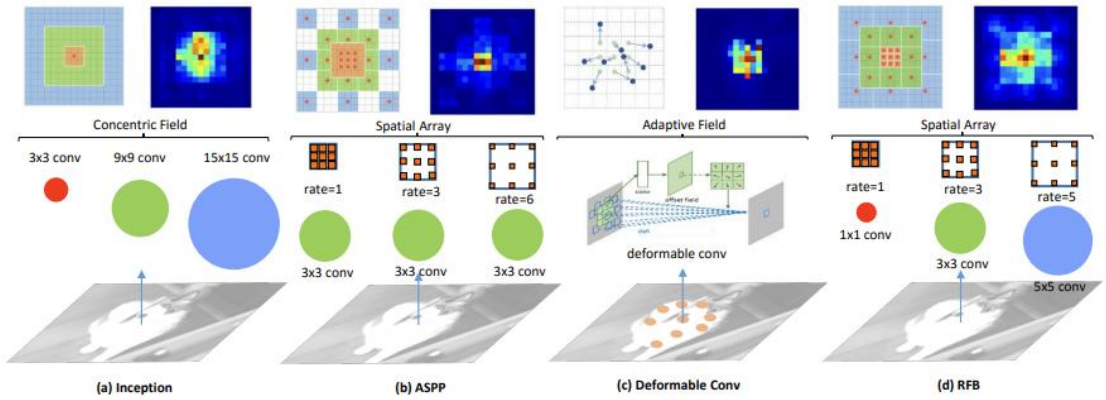

图 3. RFB

与上面提到的 ASPP 的区别就是，ASPP 在对特征图进行提取特征时，使用的是同一大小的 3×3 正常卷积，然后再接 dilated rate 不同的空洞卷积，提取的特征可判别性不高。而 RFB 在对特征图提取特征时，是使用了 3 个不同大小的正常卷积，再接空洞卷积，提取的特征符合人类视觉感知。

2.注意力机制：

2.1 SENet：对每张特征图添加权重，并让网络自动学习调整权重信息。对 C 张大小为 H × W 的特征图（总维度就是 C × H × W）进行全局平均池化，也就是上图的 Global pooling 操作，然后输出就是 C 张大小为 1 × 1 的特征图（总维度是 C × 1 × 1），跟着全连接层，全连接层主要是为了减少通道数量，r 的大小关系到通道压缩比例，文中 r 取了 16，就是把通道数缩小到原来的 1/16，然后进行 ReLU 激活，再通过全连接层将通道数恢复到 C，经过 Sigmoid 激活后（维度为 C × 1 × 1）直接与 C 张大小为 H×W 的特征图相乘，这样就可以理解为每张特征图都被添加了权重信息，如下图所示：

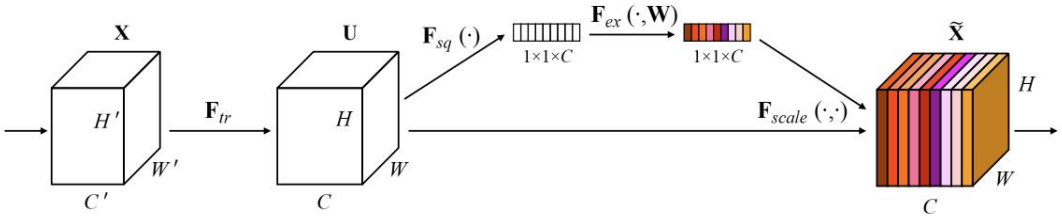

图 4. SENet

2.2 SAM: 空间注意力模块用来对特征图内部的空间位置添加注意力机制的模块，假定输入的特征图还是 C×H×W（也就是 C 张大小为 H×W 的特征图），这次我们对特征图的每个点（H×W 内）进行通道数为 C 的最大值池化，这样最大值池化输出的特征图大小就是 1×H×W，同时也进行通道数为 C 的平均值池化，输出的特征图大小也是 1×H×W，将最大值池化输出的特征图和平均值池化输出的特征图进行拼接形成 2×H×W 的拼接特征图，然后通过 1×1 卷积进行通道降维成 1×H×W 的输出特征图，再经过 Sigmoid 激活形成空间注意力权重，然后和原来的 C×H×W 的特征图进行相乘。这样相当于给每张 H×W 的特征图乘于一个 H×W 的空间权重，从而形成空间注意力模块。

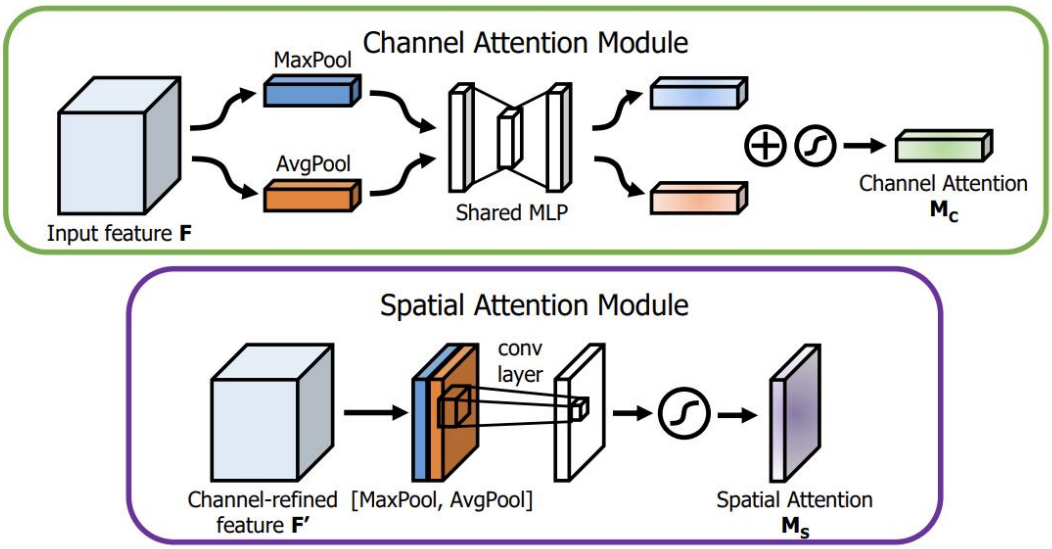

图 4. SAM

2.3 Modified SAM: 这是本文提出的一个创新点，称为像素注意力机制，它的思路也非常简单，就是把 SAM 模块的池化层全部去除，对 C×H×W 的特征图进行 1×1 卷积（既没有降通道也没有升通道），得到 C×H×W 的输出特征图，然后使用 Sigmoid 激活，再与原来的 C×H×W 进行像素点相乘。

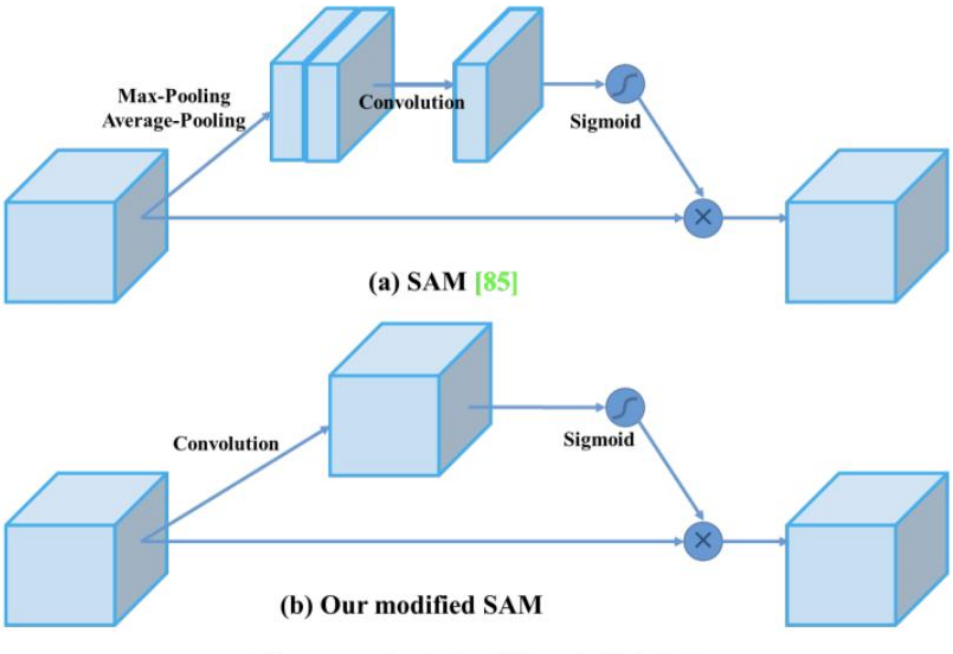

图 5. Modified SAM

3. 特征融合：卷积神经网络有一个很大的特点就是，高层的特征图包含丰富的语义信息，但位置信息由于卷积和池化的操作会有较大的损失，而低层的特征图虽然语义信息较少，但是由于卷积和池化的操作次数较少，保留了较为完整的位

置信息，目标检测既需要丰富的语义信息来判别物体的类别，也需要精确的位置信息来定位物体的位置，因而如何把低层的特征图和高层的特征图信息相互融合，就是特征融合所需要解决的问题。

3.1 Skip Connection: 类似于 resnet 残差块。

3.2 Hyper Column: 取得一个检测框（可能是正方形），resize 到分类网络的固定输入大小（分类网络使用了全连接层），resized 图像分块进入分类网络，把各个层的特征图进行 bilinear 插值，恢复到图像分块大小，取得各个层的对应位置的超列（级联多通道激活值）.

3.3 FPN: 特征金字塔网络，将低分辨率、强语义特征与高分辨率、弱语义特征通过自顶向下的路径和横向连接相结合，从而进行特征融合。

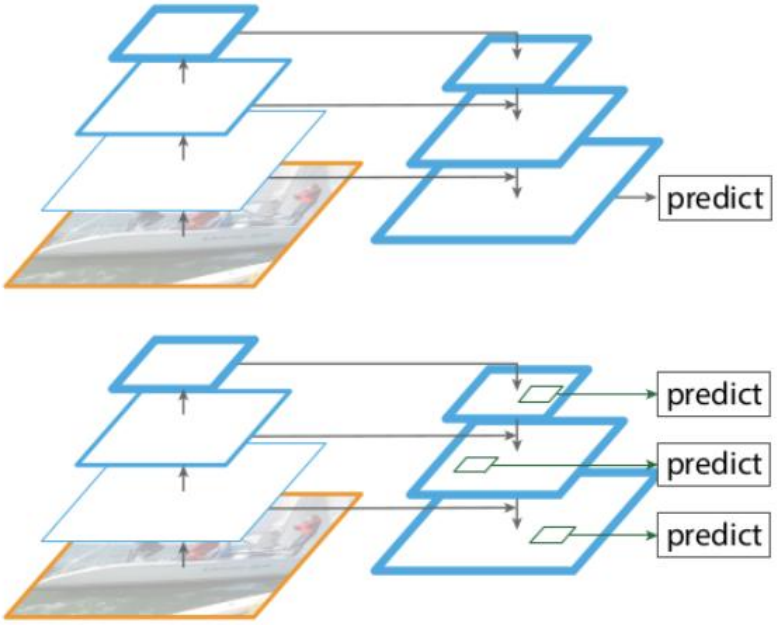

图 6. FPN

3.4 SFAM: 主要思想是利用 SE 模块对多尺度级联特征映射进行通道级加权，SFAM 是将 TUM 产生的多层次特征图聚合为特征金字塔，第一阶段是将同一分辨率特征进行拼接，注意是同一分辨率的，TUM 产生的有 6 个尺度不一样的特征图，论文中一共有 8 个 TUM 模块，因而每个 TUM 模块产生的特征图均是 128×S×S（S 为 1，3，5，10，20，40），因而通道拼接后大小为 128×8=1024，与图示相同。第二阶段是对通道添加权重，也就是 SE 模块。在检测阶段，在 6 层特征金字塔后面接两个卷积层来进行位置回归和分类，六个特征图的默认框的检测范围遵循原始 SSD 的设置。在特征图的每个点，设置三种不同比例总共六

个锚点框。之后，使用 0.05 的概率分数作为阈值来过滤掉分数较低的大多数锚点。使用 soft-NMS 作为后处理过程。

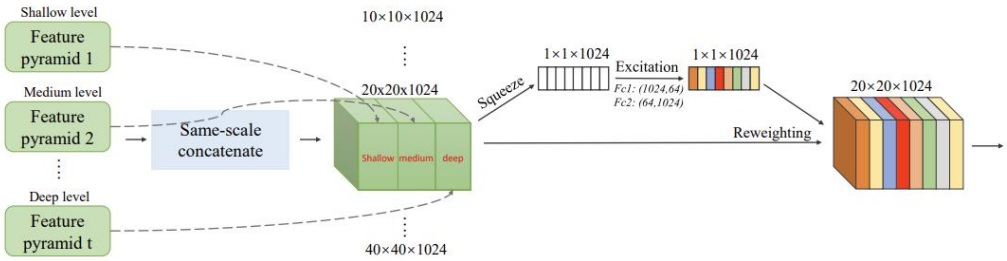

图 7. SFAM

3.5 ASFF：自适应空间特征融合，通过学习权重参数的方式将不同层的特征融合到一起，就是为了改进 FPN 进行特征融合时只是暴力地将高层特征图上采样然后进行像素级的相加操作。ASFF 通过对每张融合的特征图设置自学习的权重来进行加权融合。如下图中有三张不同尺度的特征图，称为 Level1、Level2、Level3，分别α3，β3 和γ3 相乘在相加，就构成了 ASFF-3：

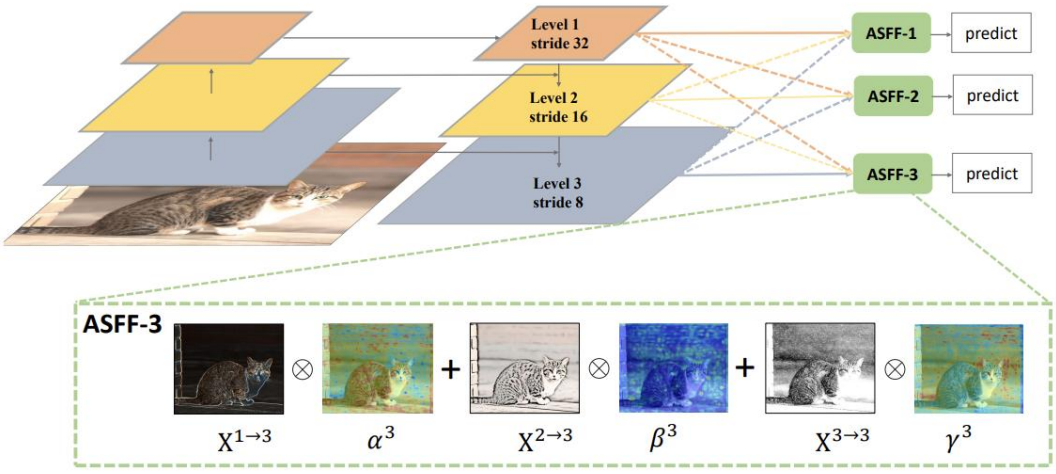

图 8. ASFF

3.6 BiFPN：经过作者观察，PAN 性能比 FPN 和 NAS-FPN 要好，但是需要更多参数和计算成本，于是从 PAN 出发，移除了只有一个输入的节点，并假定这个只有一个输入的节点对特征融合贡献不大，在相同 level 层添加一个 skip connection，为了能融合更多的特征信息。以一个自顶向下和一个自低向上的连

法作为一个基础层，不断重复这个模块，就形成了 BiFPN 特征融合模块，如下图：

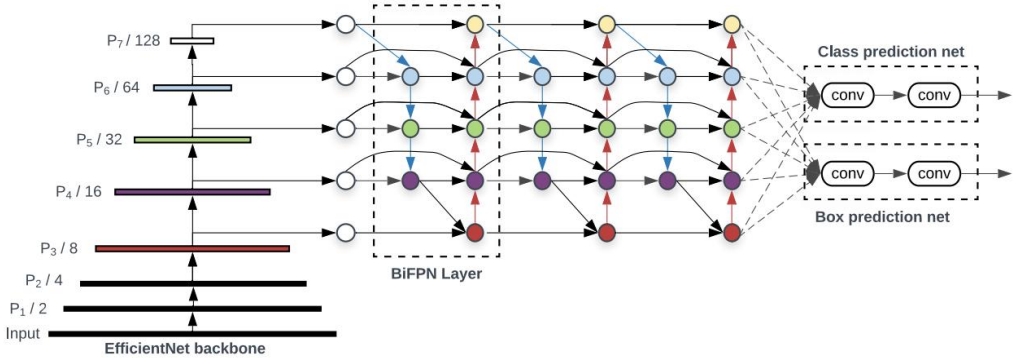

图 8. BiFPN

4. 激活函数：

4.1 Swish:

$$swish[x] = x \bullet \sigma(x)$$

4.2 hard-Swish:

$$h-swish[x] = x\frac{ReLU6(x+3)}{6}$$

swish 非线性激活函数作为 ReLU 的替代，可以可以显着提高神经网络的准确性。可以尝试使用。

4.3 Mish:

$$mish[x] = x \bullet tanh(softplus(x))$$

Mish 是一个处处可微的升级版，很好用。