一文看尽物体检测中的各种FPN

轻墨
解密AI大骗局，欢迎关注我的微信公众号「小纸屑」。

已关注

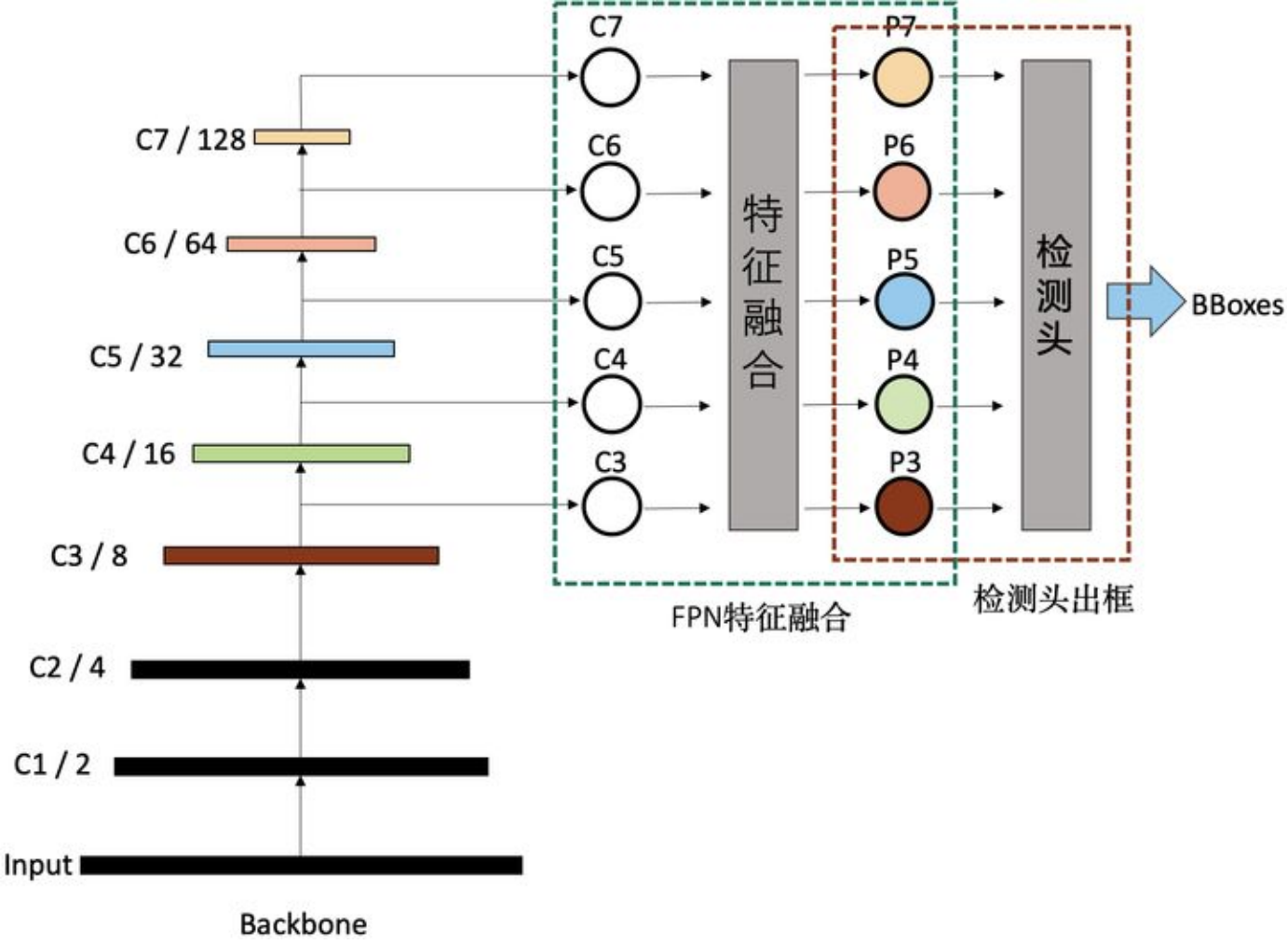
早期的物体检测算法，无论是一步式的，还是两步式的，通常都是在Backbone的最后一个stage（特征图分辨率相同的所有卷积层归类为一个stage）最后一层的特征图，直接外接检测头做物体检测。此种物体检测算法，可以称之为单stage物体检测算法。

由于单stage物体检测算法中，Backbone的最后一个stage的stride通常是32，导致输出的特征图分辨率是输入图片分辨率的1/32，太小，不利于物体检测，因此单stage的物体检测算法，一般会将最后一个stage的MaxPooling去掉或者将stride为2的conv改为stride为1的conv，以增大最后一个分辨率。

后来研究发现，单stage物体检测算法中，无法用单一stage的特征图同时有效的表征各个尺度的物体，因此，后来物体检测算法，就逐渐发展为利用不同stage的特征图，形成特征金字塔网络（feature parymid network），表征不同scale的物体，然后再基于特征金字塔做物体检测，也就是进入了FPN时代。

本文将认真梳理物体检测中常用的各种FPN。

解构物体检测各个阶段



如上图，我们常见的物体检测算法，其实可以分解为三个递进的阶段：

1）Backbone生成特征阶段

计算机视觉任务一般都是基于常用预训练的Backbone，生成抽象的语义特征，再进行特定任务微调。物体检测也是如此。

Backbone生成的特征，一般按stage划分，分别记作C1、C2、C3、C4、C5、C6、C7等，其中的数字与stage的编号相同，代表的是分辨率减半的次数，如C2代表stage2输出的特征图，分辨率为输入图片的1/4，C5代表，stage5输出的特征图，分辨率为输入图片的1/32。

2）特征融合阶段

这个是FPN特有的阶段，FPN一般将上一步生成的不同分辨率特征作为输入，输出经过融合后的特征。输出的特征一般以P作为编号标记。如FPN的输入是，C2、C3、C4、C5、C6，经过融合后，输出为P2、P3、P4、P5、P6。这个过程可以用数学公式表达：

$$P_i、\ P_{i+1}、\ \ldots、\ P_{i+n} = f(C_i、\ C_{i+1}、\ \ldots、\ C_{i+n})$$

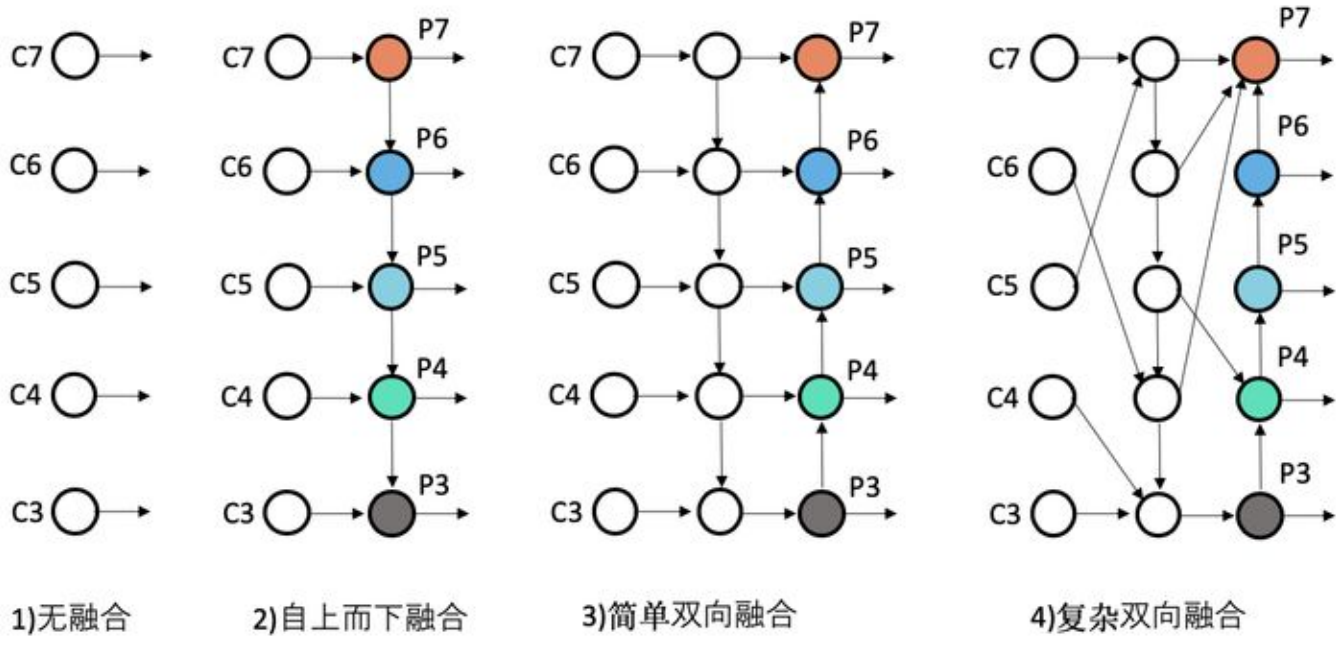
3）检测头输出bounding box

FPN输出融合后的特征后，就可以输入到检测头做具体的物体检测。

FPN的演进

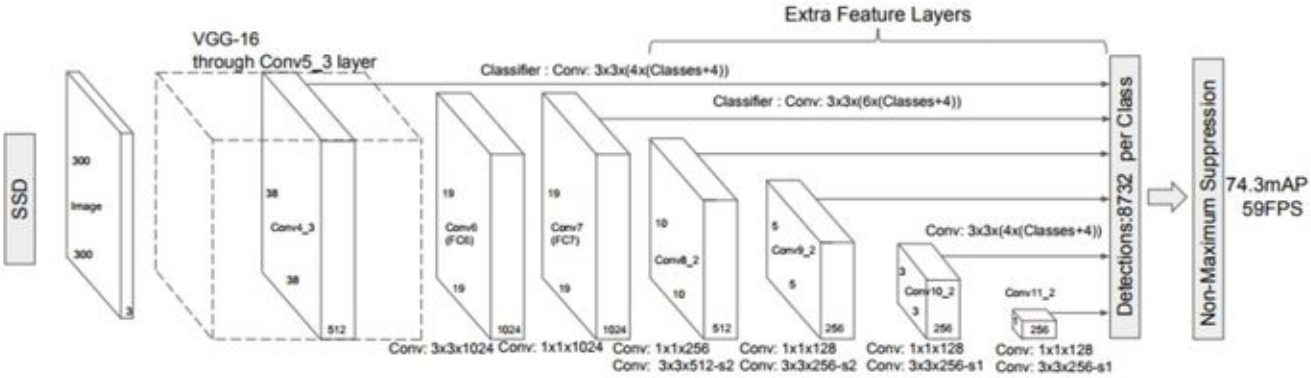
物体检测性能提升，一般主要通过数据增强、改进Backbone、改进FPN、改进检测头、改进loss、改进后处理等6个常用手段。

其中FPN自从被提出来，先后迭代了不少版本。大致迭代路径如下图：



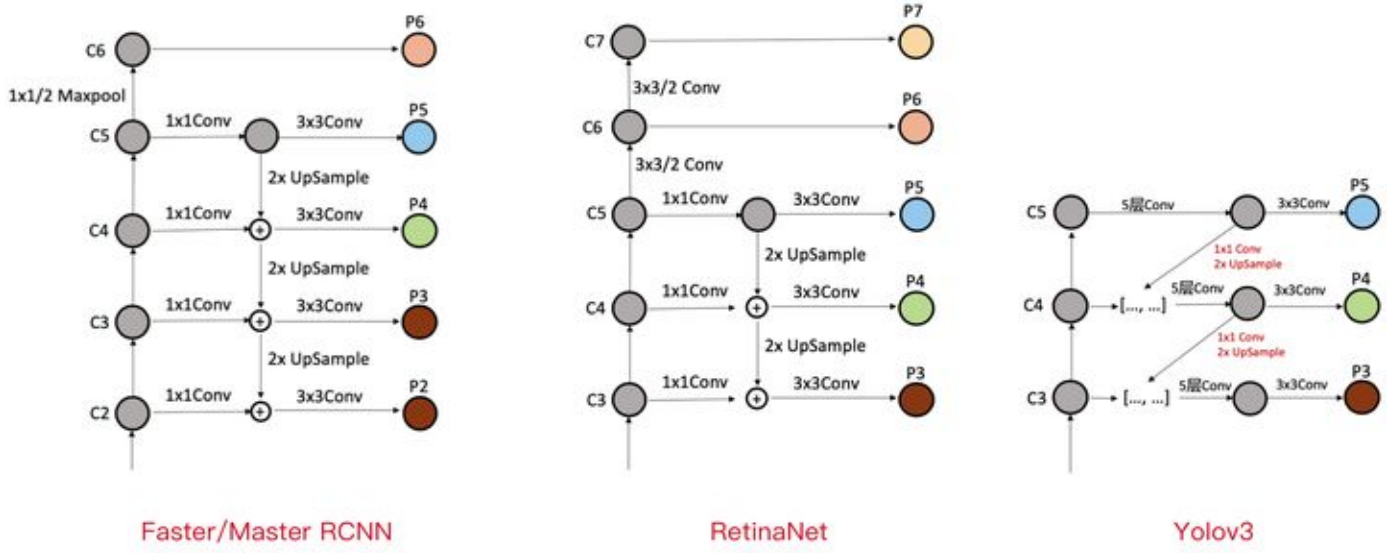
1）无融合

无融合，又利用多尺度特征的典型代表就是2016年日出的鼎鼎有名的SSD，它直接利用不同stage的特征图分别负责不同scale大小物体的检测。



2）自上而下单向融合

自上而下单向融合的FPN，事实上仍然是当前物体检测模型的主流融合模式。如我们常见的Faster RCNN、Mask RCNN、Yolov3、RetinaNet、Cascade RCNN等，具体各个FPN的内部细节如下图。



a）Faster/Master/Cascade RCNN中的FPN

Faster/Master/Cascade RCNN中的FPN，利用了C2-C6五个stage的特征，其中C6是从C5直接施加1x1/2的MaxPooling操作得到。FPN融合后得到P2-P6，其中P6直接等于C6，P5是先经过1x1Conv，再经过3x3Conv得到，P2-P4均是先经过1x1Conv，再融合上一层2xUpsample的特征，再经过3x3Conv得到。具体过程可以看上图。

b）RetinaNet中的FPN

RetinaNet中的FPN，利用了C3-C7五个stage的特征，其中C6是从C5直接施加3x3/2的Conv操作得到，C7是从C6直接施加3x3/2的Conv操作得到。FPN融合后得到P3-P7，其中P6、P7直接等于C6、C7，P5是先经过1x1Conv，再经过3x3Conv得到，P3-P4均是先经过1x1Conv，再融合上一层2xUpsample的特征，再经过3x3Conv得到。具体过程可以看上图。

可以看出，RetinaNet基本与Faster/Master/Cascade RCNN中的FPN一脉相承。只是利用的stage的特征略有差别，Faster/Master/Cascade RCNN利用了高分辨率低语义的C2，RetinaNet利用了更低分辨率更高语义的C7。其他都是细微的差别。

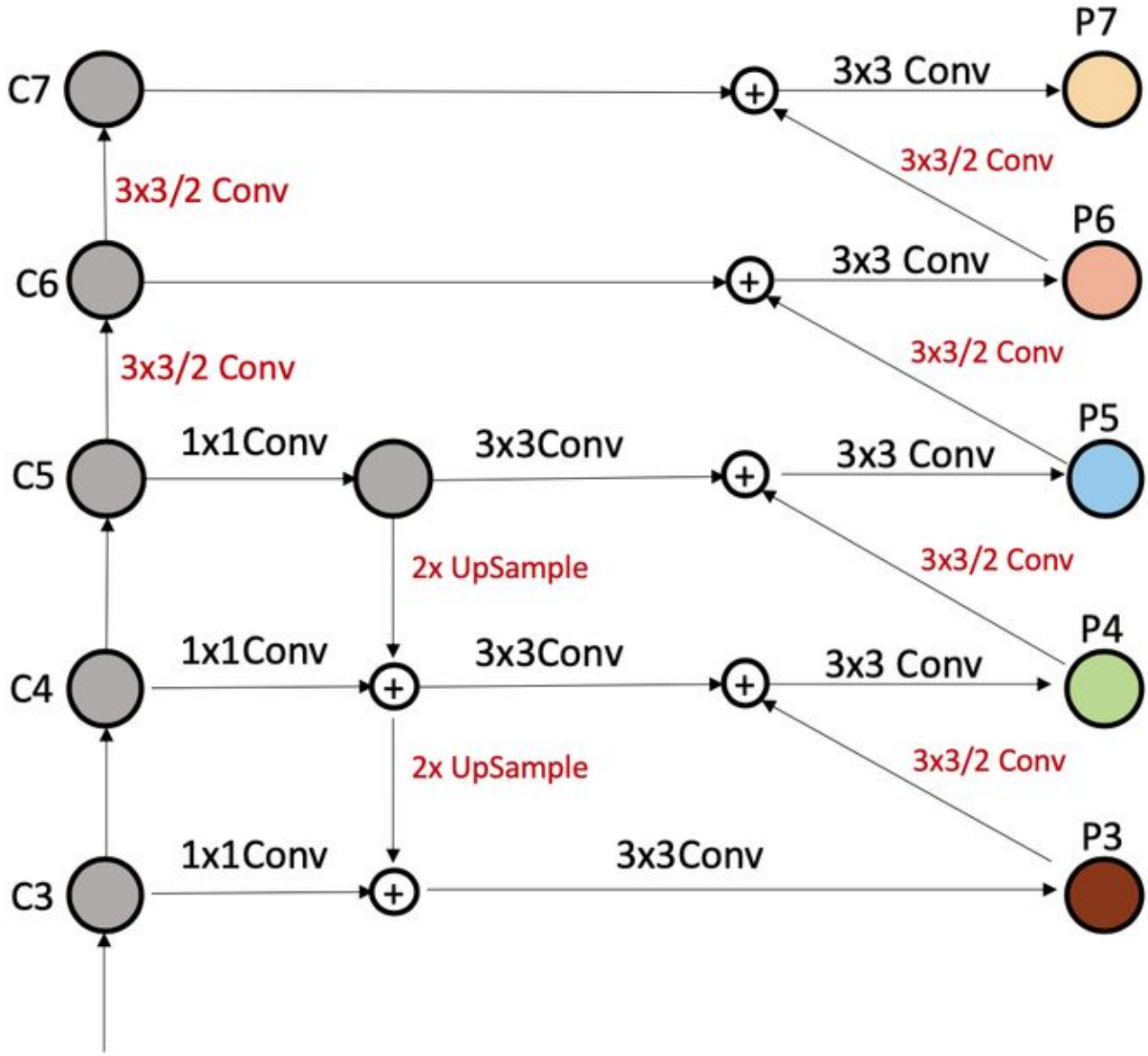
c）Yolov3中的FPN

Yolov3中的FPN与上述两个有比较大的区别。首先，Yolov3中的FPN只利用到了C3-C5三个stage的特征；其次，从C5征到P5特征，会先经过5层Conv，然后再经过一层3x3Conv；最后，C3-C4到P3-P4特征，上一层特征会先经过1x1Conv+2xUpsample，然后先与本层特征concatenate，再经过5层Conv，之后经过一层3x3Conv。看图最清楚。

可以看图仔细对比Yolov3与Faster/Master/Cascade RCNN以及RetinaNet细节上的区别。

3）简单双向融合

FPN自从提出来以后，均是只有从上向下的融合，PANet是第一个提出从下向上二次融合的模型，并且PANet就是在Faster/Master/Cascade RCNN中的FPN的基础上，简单增了从下而上的融合路径。看下图。

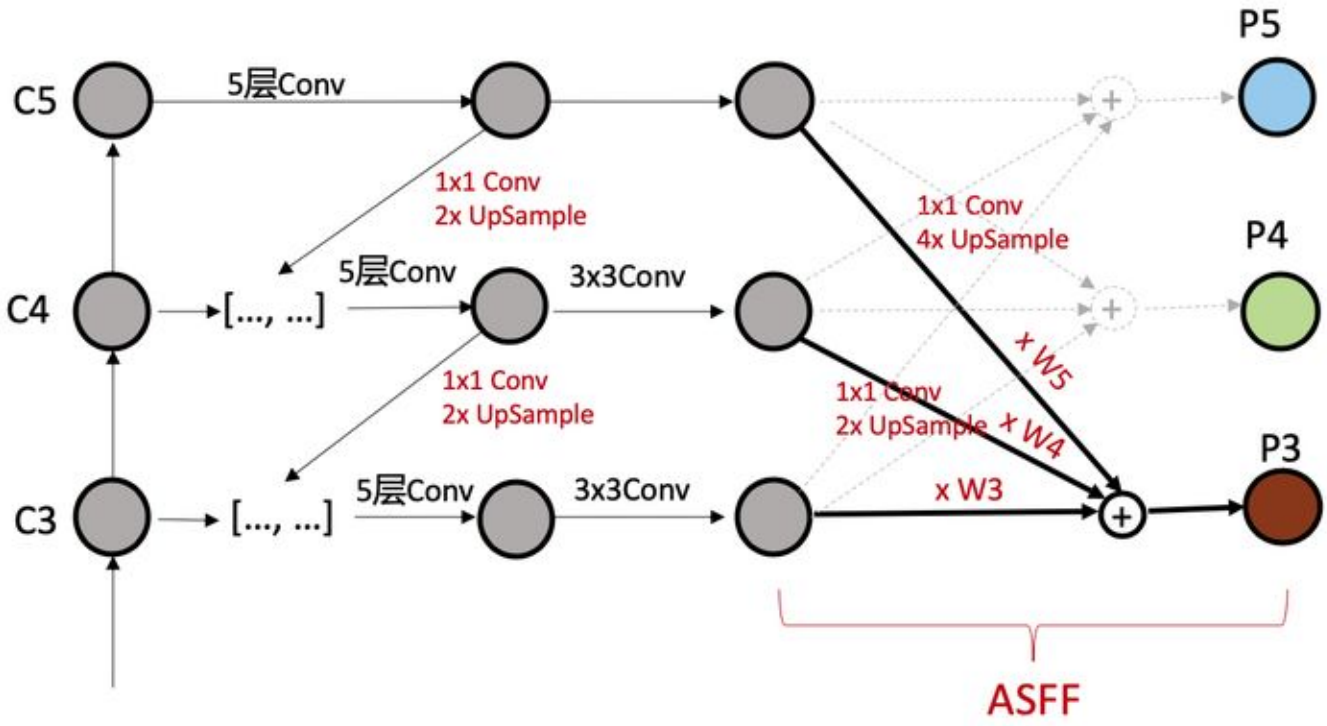


4）复杂的双向融合

PANet的提出证明了双向融合的有效性，而PANet的双向融合较为简单，因此不少文章在FPN的方向上更进一步，尝试了更复杂的双向融合，如ASFF、NAS-FPN和BiFPN。

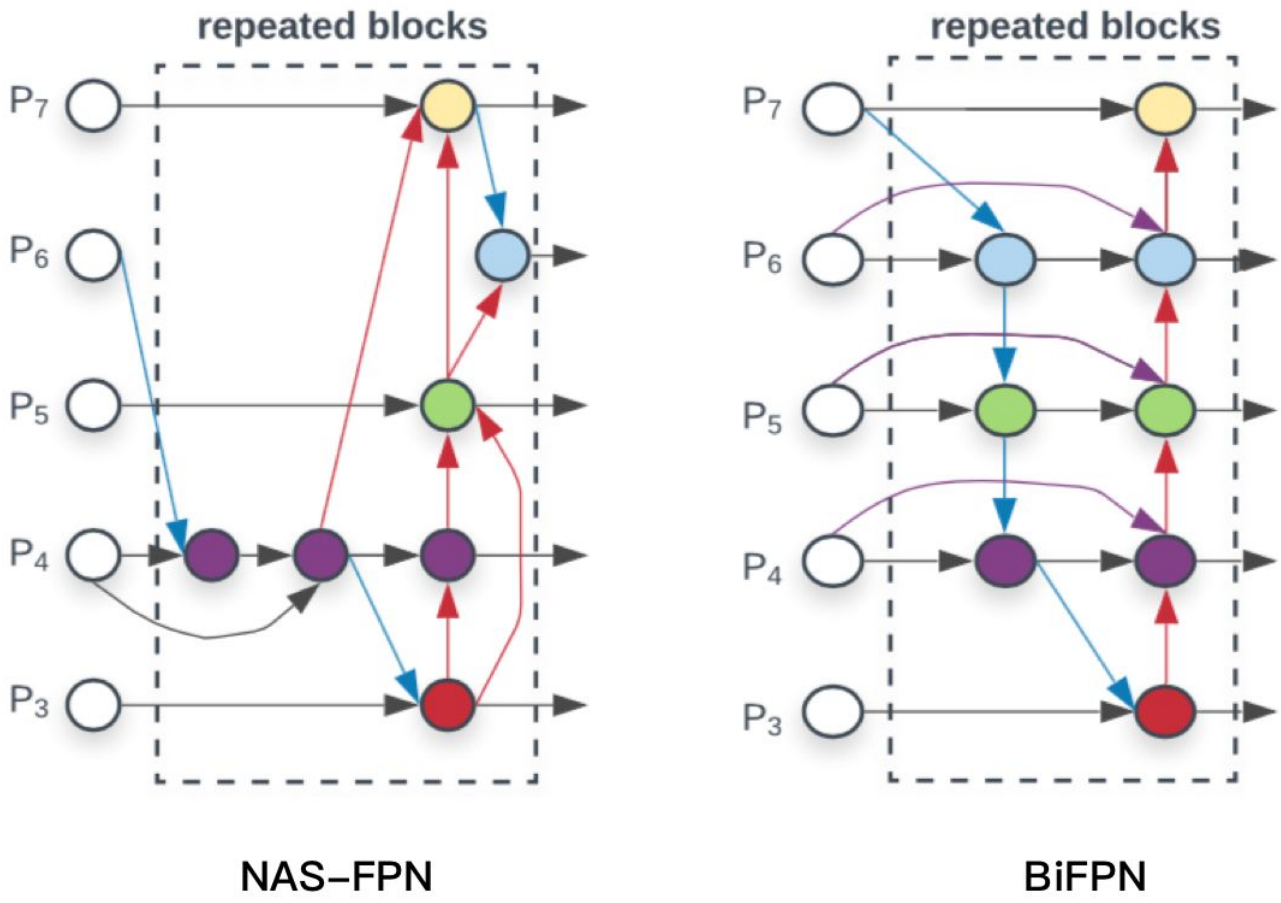
ASFF

ASFF（论文：Learning Spatial Fusion for Single-Shot Object Detection）作者在YOLOV3的FPN的基础上，研究了每一个stage再次融合三个stage特征的效果。如下图。其中不同stage特征的融合，采用了注意力机制，这样就可以控制其他stage对本stage特征的贡献度。

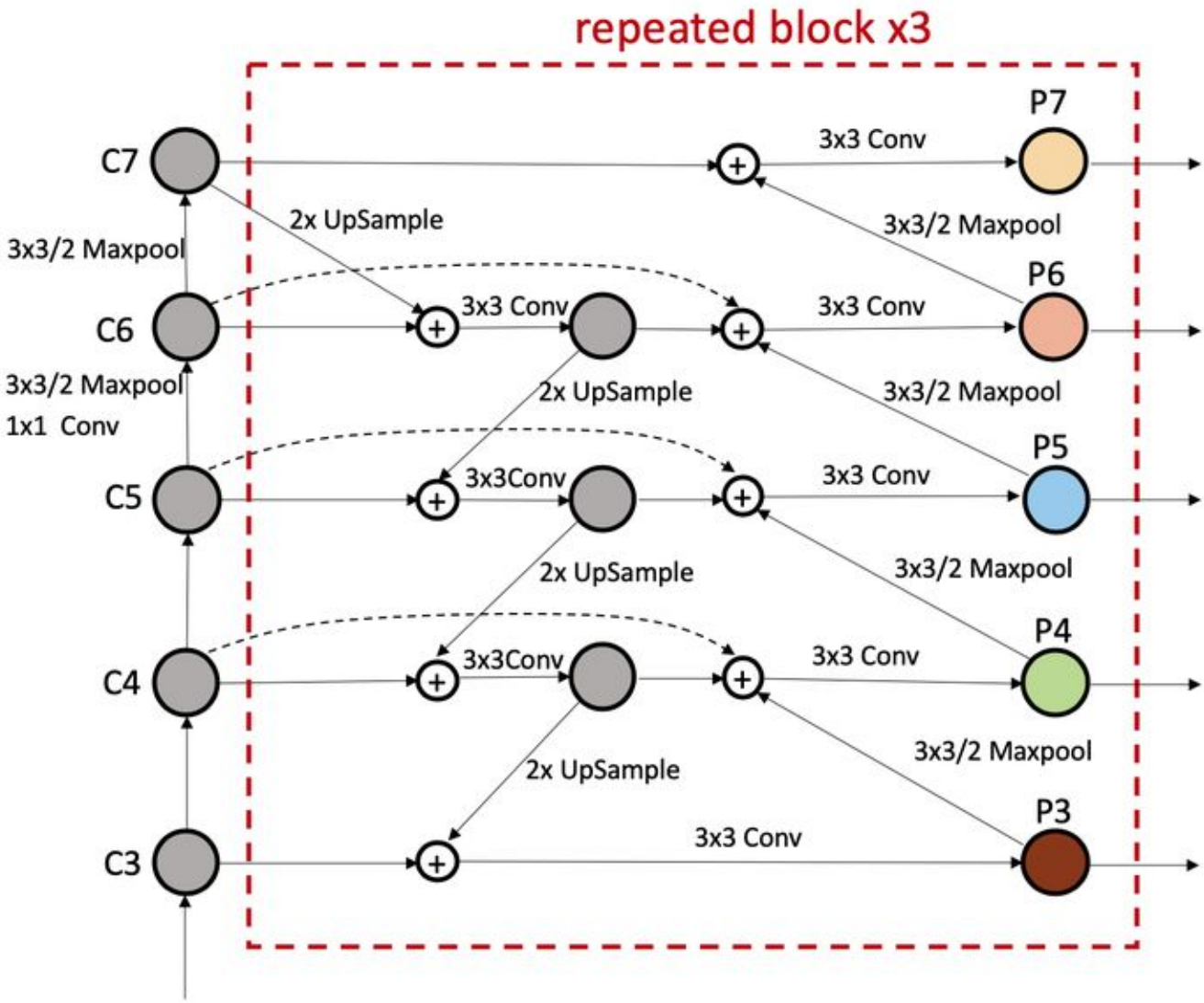


NAS-FPN和BiFPN

NAS-FPN和BiFPN，都是google出品，思路也一脉相承，都是在FPN中寻找一个有效的block，然后重复叠加，这样就可以弹性的控制FPN的大小。



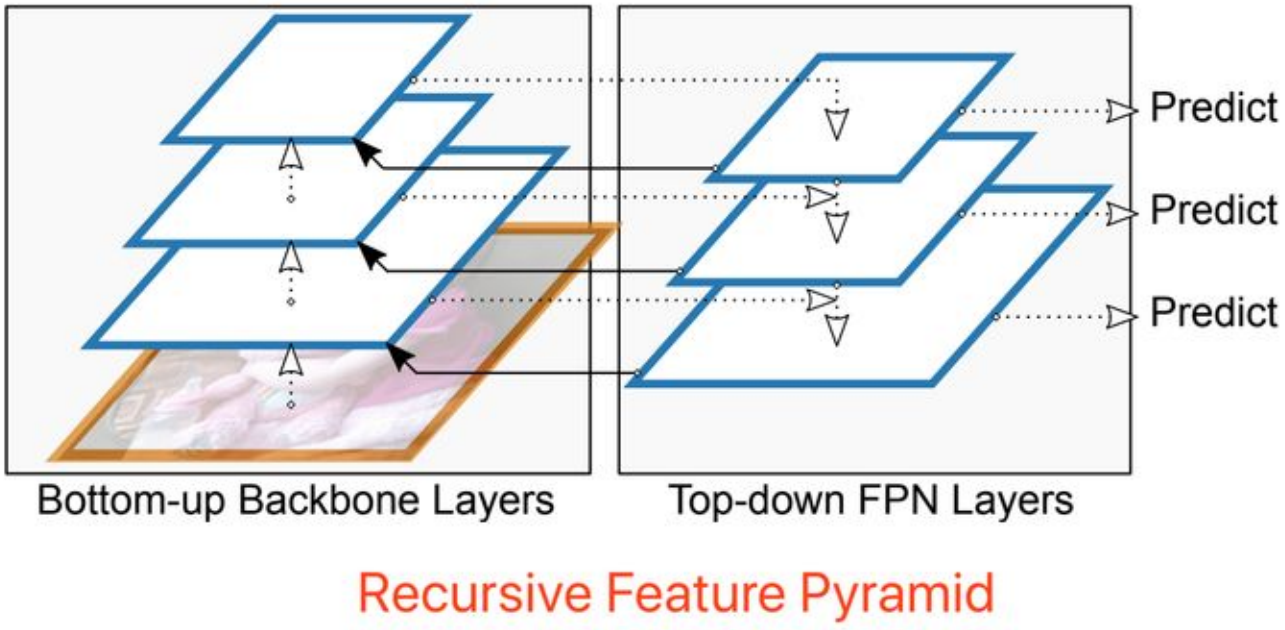
其中BiFPN的具体细节如下图。



Recursive-FPN

递归FPN是此文写作之时前两周刚刚新出炉的（原论文是DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution），效果之好令人惊讶，使用递归FPN的DetectoRS是目前物体检测（COCO mAP 54.7）、实体分割和全景分割的SOTA，太强悍了。

递归FPN理解起来很容易，就是将传统FPN的融合后的输出，再输入给Backbone，进行二次循环，如下图。



下图给出了FPN与Recursive-FPN的区别，并且把一个2层的递归FPN展开了，非常简单明了，不做过多介绍。

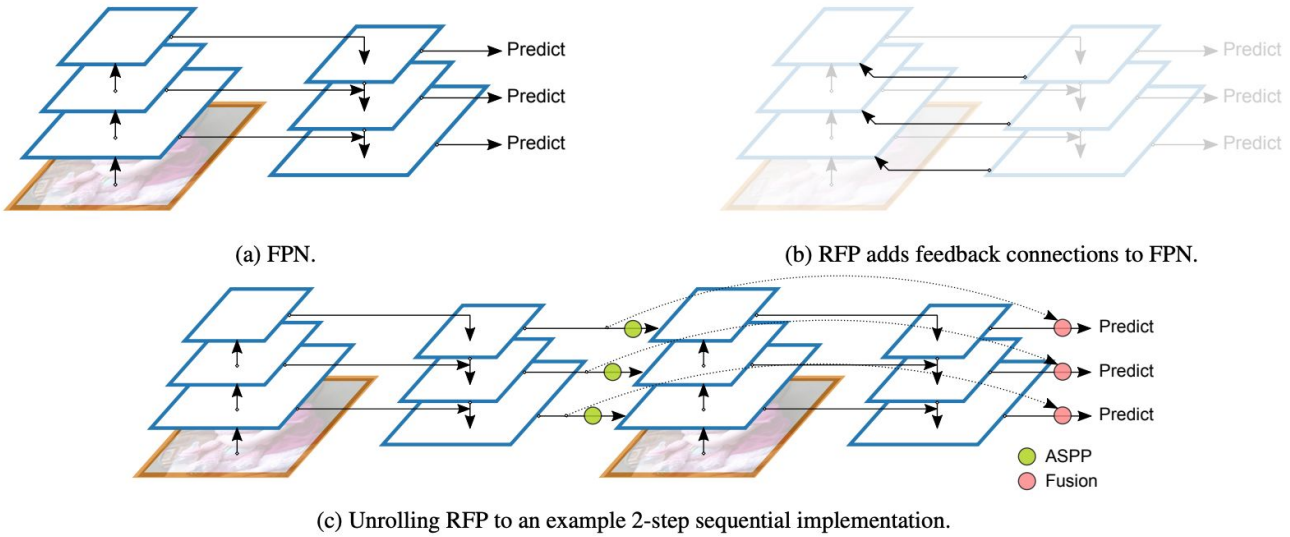


Figure 2: The architecture of Recursive Feature Pyramid (RFP). (a) Feature Pyramid Networks (FPN). (b) Our RFP incorporates feedback connections into FPN. (c) RFP unrolled to a 2-step sequential network.

5）M2det中的SFAM

M2det中的SFAM，比较复杂，它是先把C3与C5两个stage的特征融合成一个与C3分辨率相同的特征图（下图中的FFM1模块），然后再在此特征图上叠加多个UNet（下图中的TUM模块），最后将每个UNet生成的多个分辨率中相同分辨率特征一起融合（下图中的SFAM模块），从而生成最终的P3、P4、P5、P6特征，以供检测头使用。具体如下图。

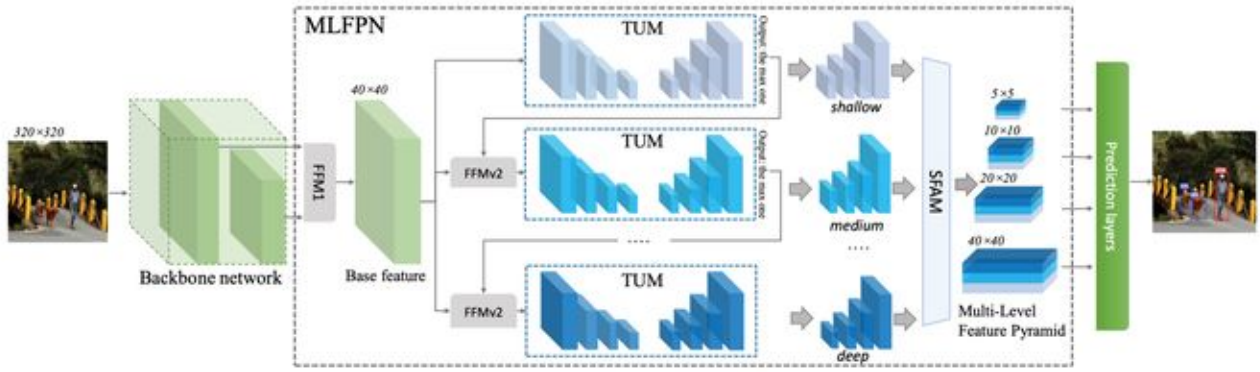


Figure 2: An overview of the proposed M2Det(320 × 320). M2Det utilizes the backbone and the Multi-level Feature Pyramid Network (MLFPN) to extract features from the input image, and then produces dense bounding boxes and category scores. In MLFPN, FFMv1 fuses feature maps of the backbone to generate the base feature. Each TUM generates a group of multi-sc features, and then the alternating joint TUMs and FFMv2s extract multi-level multi-scale features. Finally, SFAM aggregates the features into a multi-level feature pyramid. In practice, we use 6 scales and 8 levels.

每一个模块的详细细节如下图。

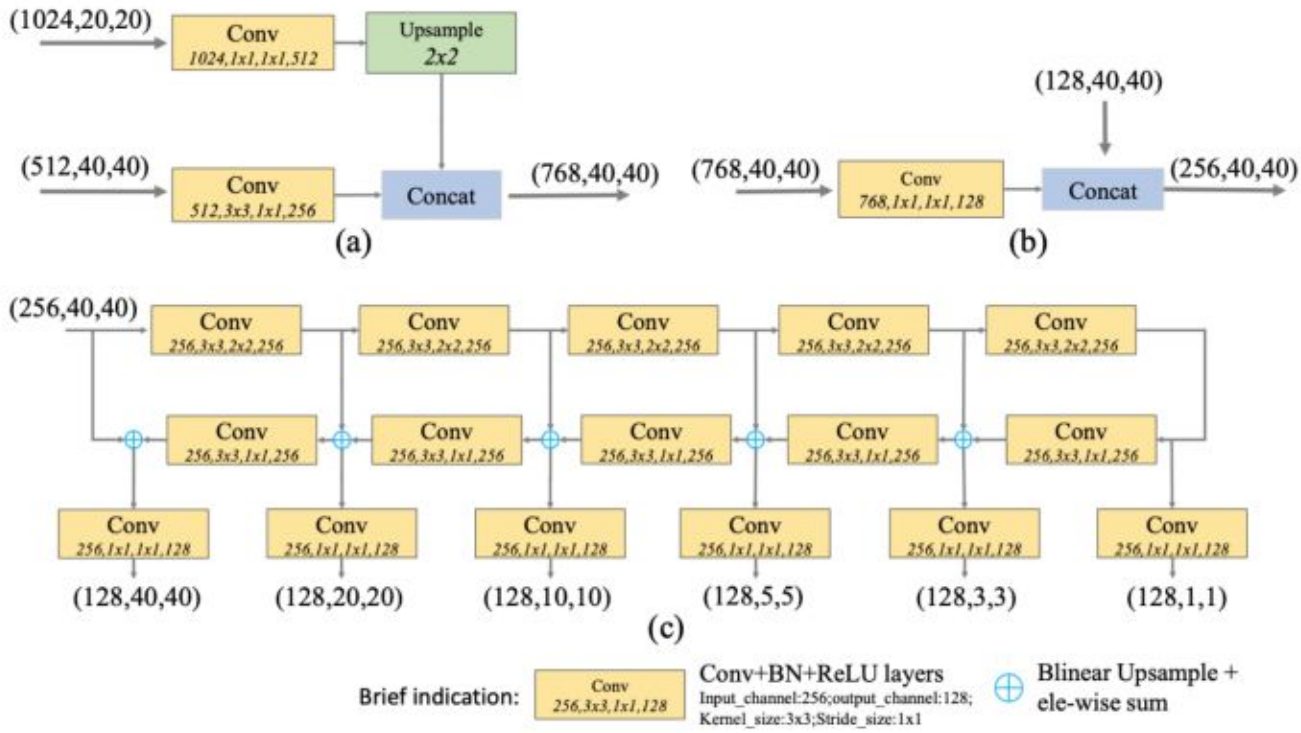


Figure 4: Structural details of some modules. (a) FFMv1, (b) FFMv2, (c) TUM. The inside numbers of each block denote: input channels, Conv kernel size, stride size, output channels.

思考

FPN的优化会显著带来物体检测的性能提升，当前最好的FPN是递归FPN，期待将来更有效的FPN出现。

最近Facebook出了一篇文章object detection by transformer，如果transformer与各种强大的FPN结合，效果如何还是值得期待。

欢迎扫码关注同名微信公众号「小纸屑」，解密AI大骗局。

1. SSD: Single Shot Multibox Detector
2. Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks
3. Mask RCNN
4. Yolov3: An Incremental Improvement
5. RetinaNet: Focal Loss for Dense Object Detection
6. Cascade RCNN: Delving into High Quality Object Detection
7. PANet: Path Aggregation Network for Instance Segmentation
8. ASFF: Learning Spatial Fusion for Single-Shot Object Detection
9. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection
10. BiFPN: (EfficientDet: Scalable and Efficient Object Detection)
11. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution
12. SFAM（M2det: A single-shot object detector based on multi-level feature pyramid network）