

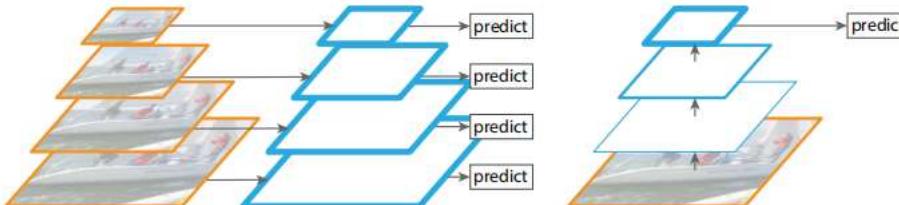
## 目标检测之FPN、AugFPN、NAS-FPN

针对小目标的检测有提升的文章。未完待续~

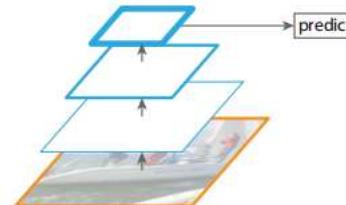
# Feature Pyramid Networks for Object Detection

FPN是一种多尺度的目标检测算法。大多数目标检测算法都是采用顶层特征来做预测的，但是我们知道：低层的特征语义信息较少，但是位置信息丰富；高层的特征语义信息丰富，位置信息粗略。虽然也有些多尺度特征融合的方法，但是一般采用融合后的特征做预测，但是FPN是在不同特征层独立进行预测的。

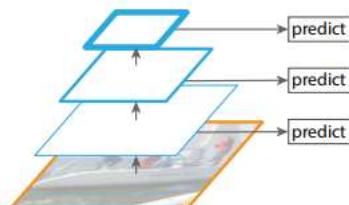
4种利用特征的形式：



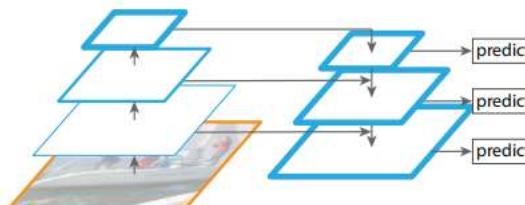
(a) Featurized image pyramid



(b) Single feature map



(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

(a) **图像金字塔**：将图像做成不同的scale，不同scale的图像生成对应的不同scale的特征。缺点在于**时间成本高**。有些算法会在**测试时候**采用图像金字塔。

(b) 像**SPP net, Fast RCNN, Faster RCNN**是采用这种方式，即仅采用网络最后一层的特征。

(c) 像**SSD (Single Shot Detector)** 采用这种多尺度特征融合的方式，没有上采样过程，即从网络不同层抽取不同尺度的特征做预测，这种方式不会的计算量。作者认为**SSD**算法中没有用到足够低层的特征（在SSD中，最低层的特征是VGG网络的conv4\_3），而在作者看来**足够低层的特征对于检测小目标帮助的**。

== (d) ==本文作者是采用这种方式，顶层特征通过上采样和低层特征做融合，而且每层都是独立预测的。

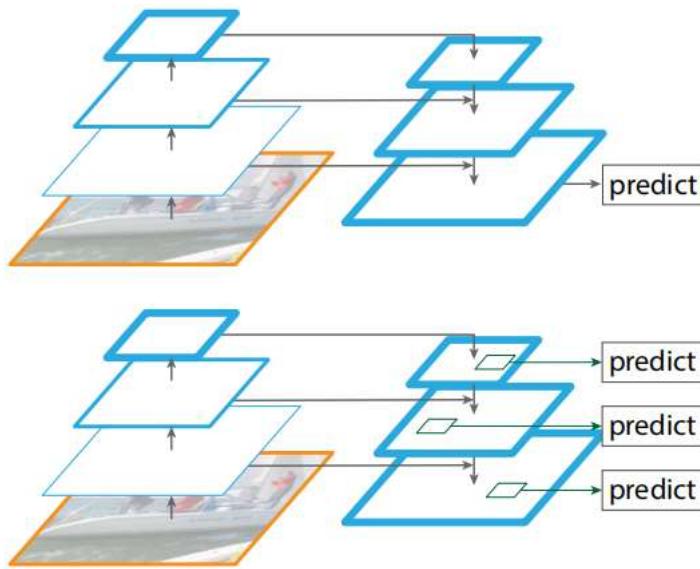


Figure 2. Top: a top-down architecture with skip connections, where predictions are made on the finest level (e.g., [28]). Bottom: our model that has a similar structure but leverages it as a *feature pyramid*, with predictions made independently at all levels. [https://blog.csdn.net/qq\\_37099552](https://blog.csdn.net/qq_37099552)

如图一个带有**skip connection**的网络结构在预测的时候是在finest level (自顶向下的最后一层) 进行的，简单讲就是经过多次上采样并融合特征到最后一步生成的特征做预测。

下面一个网络结构和上面的类似，**区别在于预测是在每一层中独立进行的**。

作者的主网络采用**ResNet**。

算法结构分为三部分：

1. 自底向上的线路
2. 自顶向下的线路
3. 横向连接 (lateral connection)

图中放大的区域就是横向连接，这里 $1 \times 1$ 的卷积核的主要作用是减少卷积核的个数，也就是减少了**feature map**的个数，并不改变**feature map**的尺寸。

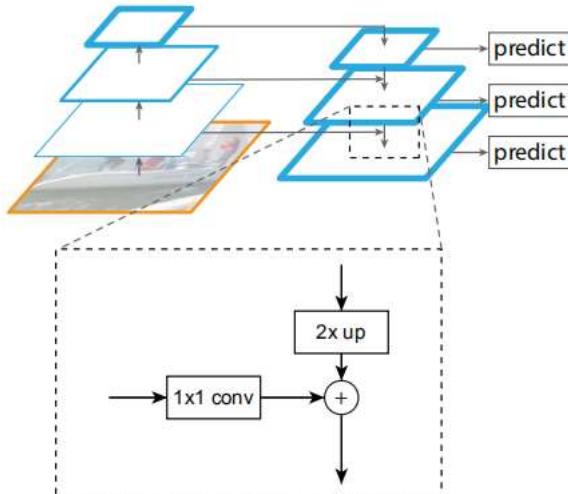


Figure 3. A building block illustrating the lateral connection and the top-down pathway, merged by addition. [https://blog.csdn.net/qq\\_37099552](https://blog.csdn.net/qq_37099552)

**自底向上**：网络的前向过程。在前向过程中，feature map的大小在经过某些层后会改变，而在经过其他一些层的时候不会改变，作者将不改变**feature map**大小的层归为一个**stage**，因此每次抽取的特征都是每个**stage**的最后一个层输出，这样就能构成特征金字塔。

**自顶向下**: 采用上采样 (upsampling) 进行, 而横向连接则是将上采样的结果和自底向上生成的相同大小的**feature map**进行融合 (merge)。在融合时采用 $3 \times 3$ 的卷积核对每个融合结果进行卷积, 目的是消除上采样的混叠效应 (aliasing effect)。并假设生成的feature map结果是P2, P3, P4, P5, 和自向上的卷积结果C2, C3, C4, C5——对应。

**作者一方面将FPN放在RPN网络中用于生成proposal**, 原来的RPN网络是以主网络的某个卷积层输出的feature map作为输入, 简单讲就是只用这个一个尺寸的**feature map**。但是现在要将FPN嵌在RPN网络中, 生成不同尺度特征并融合作为RPN网络的输入。在每一个scale层, 都定义了不同大小的anchor, 对于P3, P4, P5, P6这些层, 定义anchor的大小为 $32^2, 64^2, 128^2, 256^2, 512^2$ , 另外每个scale层都有3个长宽对比度: 1:2, 1:1, 2:1。所以整个特征金字塔有5种anchor。

**正负样本的界定和Faster RCNN差不多**: 如果某个anchor和一个给定的ground truth有最高的IOU或者和任意一个Ground truth的IOU都大于0.7, 则是正样本; 如果一个anchor和任意一个ground truth的IOU都小于0.3, 则为负样本。

看看加入FPN的RPN网络的有效性, 如下表Table1。网络这些结果都是基于ResNet-50。评价标准采用AR, AR表示Average Recall, AR右上角的100表示在所有100个anchor中召回率, AR的右下角s, m, l表示COCO数据集中object的大小分别是小, 中, 大。feature列的大括号{}表示每层独立预测。

| RPN                                 | feature   | # anchors | lateral? | top-down? | AR <sup>100</sup> | AR <sup>1k</sup> | AR <sub>s</sub> <sup>1k</sup> | AR <sub>m</sub> <sup>1k</sup> | AR <sub>l</sub> <sup>1k</sup> |
|-------------------------------------|-----------|-----------|----------|-----------|-------------------|------------------|-------------------------------|-------------------------------|-------------------------------|
| (a) baseline on conv4               | $C_4$     | 47k       |          |           | 36.1              | 48.3             | 32.0                          | 58.7                          | 62.2                          |
| (b) baseline on conv5               | $C_5$     | 12k       |          |           | 36.3              | 44.9             | 25.3                          | 55.5                          | 64.2                          |
| (c) FPN                             | $\{P_k\}$ | 200k      | ✓        | ✓         | 44.0              | 56.3             | 44.9                          | 63.4                          | 66.2                          |
| <i>Ablation experiments follow:</i> |           |           |          |           |                   |                  |                               |                               |                               |
| (d) bottom-up pyramid               | $\{P_k\}$ | 200k      | ✓        |           | 37.4              | 49.5             | 30.5                          | 59.9                          | 68.0                          |
| (e) top-down pyramid, w/o lateral   | $\{P_k\}$ | 200k      |          | ✓         | 34.5              | 46.1             | 26.5                          | 57.4                          | 64.7                          |
| (f) only finest level               | $P_2$     | 750k      | ✓        | ✓         | 38.4              | 51.3             | 35.1                          | 59.7                          | 67.6                          |

1. 从 (a) (b) (c) 的对比可以看出FRN的作用确实很明显。另外 (a) 和 (b) 的对比可以看出高层特征并非比低一层的特征有效。
2. (d) 表示只有横向连接, 而没有自顶向下的过程, 也就是仅仅对自底向上 (**bottom-up**) 的每一层结果做一个**11的横向连接和33的卷积**得到最终的点像Fig1的 (b)。从feature列可以看出预测还是分层独立的。作者推测 (d) 的结果并不好的原因在于在自底向上的不同层之间的semantic gaps比做横向连接不好的原因)。
3. (e) 表示只有自顶向下的过程, 但是没有横向连接, 即向下过程没有融合原来的特征。这样效果不好的原因在于目标的**location**特征在经过多次上采样过程后变得更加不准确。
4. (f) 采用finest level层做预测 (参考Fig2的上面那个结构), 即经过多次特征上采样和融合到最后一步生成的特征用于预测, 主要是证明金字塔分层的表达能力。显然finest level的效果不如FPN好, 原因在于**RPN**网络是一个窗口大小固定的滑动窗口检测器, 因此在**金字塔的不同层滑动可以增加其化的鲁棒性**。另外 (f) 有更多的anchor, 说明增加**anchor**的数量并不能有效提高准确率。

另一方面将FPN用于Fast R-CNN的检测部分。除了 (a) 以外, 分类层和卷积层之前添加了2个1024维的全连接层。细节地方可以等代码出来后再研究。

实验结果如下表Table2, 这里是测试Fast R-CNN的检测效果, 所以proposal是固定的 (采用Table1 (c) 的做法)。与Table1的比较类似,

- (a) (b) (c) 的对比证明在基于区域的目标卷积问题中, 特征金字塔比单尺度特征更有效。
- (c) (f) 的差距很小, 作者认为原因是**ROI pooling**对于**region**的尺度并不敏感。因此并不能一概认为 (f) 这种特征融合的方式不好。

博主个人认为要针对具体问题来看待, 像上面在RPN网络中, 可能 (f) 这种方式不太好, 但是在Fast RCNN中就没那么明显。

在这里插入图片描述

| Fast R-CNN                          | proposals      | feature   | head  | lateral? | top-down? | AP@0.5 | AP   | AP <sub>s</sub> | AP <sub>m</sub> | AP <sub>l</sub> |
|-------------------------------------|----------------|-----------|-------|----------|-----------|--------|------|-----------------|-----------------|-----------------|
| (a) baseline on conv4               | RPN, $\{P_k\}$ | $C_4$     | conv5 |          |           | 54.7   | 31.9 | 15.7            | 36.5            | 45.5            |
| (b) baseline on conv5               | RPN, $\{P_k\}$ | $C_5$     | 2fc   |          |           | 52.9   | 28.8 | 11.9            | 32.4            | 43.4            |
| (c) FPN                             | RPN, $\{P_k\}$ | $\{P_k\}$ | 2fc   | ✓        | ✓         | 56.9   | 33.9 | 17.8            | 37.7            | 45.8            |
| <i>Ablation experiments follow:</i> |                |           |       |          |           |        |      |                 |                 |                 |
| (d) bottom-up pyramid               | RPN, $\{P_k\}$ | $\{P_k\}$ | 2fc   | ✓        |           | 44.9   | 24.9 | 10.9            | 24.4            | 38.5            |
| (e) top-down pyramid, w/o lateral   | RPN, $\{P_k\}$ | $\{P_k\}$ | 2fc   |          | ✓         | 54.0   | 31.3 | 13.3            | 35.2            | 45.3            |
| (f) only finest level               | RPN, $\{P_k\}$ | $P_2$     | 2fc   | ✓        | ✓         | 56.3   | 33.4 | 17.3            | 37.3            | 45.6            |

同理, 将FPN用于Faster RCNN的实验结果如下表Table3。

| Faster R-CNN                                 | proposals      | feature   | head  | lateral? | top-down? | AP@0.5 | AP   | AP <sub>s</sub> | AP <sub>m</sub> | AP <sub>l</sub> |
|--|----------------|-----------|-------|----------|-----------|--------|------|-----------------|-----------------|-----------------|
| (* baseline from He et al. [10] <sup>†</sup> | RPN, $C_4$     | $C_4$     | conv5 |          |           | 47.3   | 26.3 | -               | -               | -               |
| (a) baseline on conv4                        | RPN, $C_4$     | $C_4$     | conv5 |          |           | 53.1   | 31.6 | 13.2            | 35.6            | 47.1            |
| (b) baseline on conv5                        | RPN, $C_5$     | $C_5$     | 2fc   |          |           | 51.7   | 28.0 | 9.6             | 31.9            | 43.1            |
| (c) FPN                                      | RPN, $\{P_k\}$ | $\{P_k\}$ | 2fc   | ✓        | ✓         | 56.9   | 33.9 | 17.8            | 37.7            | 45.8            |

Table4是和近几年在COCO比赛上排名靠前的算法的对比。注意到算法在小物体检测上的提升是比较明显的。

| method                    | backbone   | competition | image pyramid | test-dev          |      |                 |                 | test-std        |                   |      |                 |      |      |
|---------------------------|------------|-------------|---------------|-------------------|------|-----------------|-----------------|-----------------|-------------------|------|-----------------|------|------|
|                           |            |             |               | AP <sub>0.5</sub> | AP   | AP <sub>s</sub> | AP <sub>m</sub> | AP <sub>I</sub> | AP <sub>0.5</sub> | AP   | AP <sub>s</sub> |      |      |
| ours, Faster R-CNN on FPN | ResNet-101 | -           | 59.1          | 36.2              | 18.2 | 39.0            | 48.2            | -               | 58.5              | 35.8 | 17.5            | 38.7 | 47.8 |

Competition-winning single-model results follow:

|                                |                                  |      |   |      |      |      |      |      |      |      |      |      |      |
|--------------------------------|----------------------------------|------|---|------|------|------|------|------|------|------|------|------|------|
| G-RM <sup>†</sup>              | Inception-ResNet                 | 2016 | - | 34.7 | -    | -    | -    | -    | -    | -    | -    |      |      |
| AttnetionNet <sup>‡</sup> [10] | VGG16 + Wide ResNet <sup>§</sup> | 2016 | ✓ | 53.4 | 35.7 | 15.6 | 38.0 | 52.7 | 52.9 | 35.3 | 14.7 | 37.6 | 51.9 |
| Faster R-CNN +++ [16]          | ResNet-101                       | 2015 | ✓ | 55.7 | 34.9 | 15.6 | 38.7 | 50.9 | -    | -    | -    | -    | -    |
| Multipath [40] (minival)       | VGG-16                           | 2015 | - | 49.6 | 31.5 | -    | -    | -    | -    | -    | -    | -    | -    |
| ION <sup>‡</sup> [2]           | VGG-16                           | 2015 | - | 53.4 | 31.2 | 12.8 | 32.9 | 45.2 | 52.9 | 30.7 | 11.8 | 32.8 | 44.8 |

另外作者强调这些实验并没有采用其他的提升方法（比如增加数据集，迭代回归，hard negative mining），因此能达到这样的结果实属不易。

## 总结

作者提出的FPN (Feature Pyramid Network) 算法同时利用低层特征高分辨率和高层特征的高语义信息，通过融合这些不同层的特征达到预测的效果。并在每个融合后的特征层上单独进行的，这和常规的特征融合方式不同。

## AugFPN: Improving Multi-scale Feature Learning for Object Detection

### 论文地址

顾名思义，AugFPN就是在经典FPN结构的基础上做了改进。FPN网络的最大贡献就是把多尺度特征金字塔网络引入到了Faster RCNN等检测网络中，从而改善了网络的检测性能

### Abstract

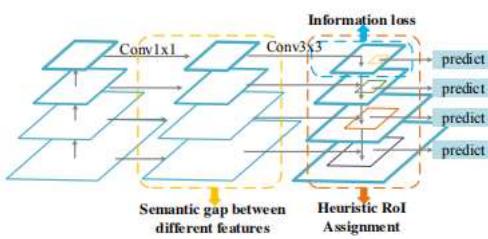
经典的FPN网络通过利用多尺度特征学\*\*\*\*提升了检测网络的检测性能，然而这种设计本身背后就存在一些缺陷，使得多尺度的特征不能被网络完全的利用。本文通过提出一种新的特征金字塔网络结构AugFPN，可以有效的克服这些缺陷。

AugFPN的创新之处在于三个组成成分，分别是：

- **Consistent Supervision (一致性监督)**：缩小了特征融合前不同尺度特征之间的语义差距
- **Residual Feature Augmentation (残差特征增强)**：通过残差特征增强提取比率不变的上下文信息，减少特征映射在最高金字塔层次的信息丢失
- **Soft ROI Selection (软RoI选择)**：自适应地学习更好的RoI特征

至于这三个模块是如何克服FPN的缺陷的，下文会有详细的分析。最后就是使用了AugFPN，无论是在大的backbone，还是小的backbone，是one-stage还是two-stage检测网络上，都在coco数据集上取得了明显的涨点。

### FPN结构存在的缺陷



FPN从结构上可以划分为三个部分，即分别是：

1. before feature fusion
2. top- down feature fusion
3. after feature fusion

FPN的三个缺陷也正好对应着这三个阶段：

### 1. 特征混合之前

我们知道，FPN在特征混合之前，backbone的不同stage学习到的特征需要先经过11卷积进行一个降维，降到通道数相同才能进行特征相加。然而不同stage的特征感受野是不一样的，包含的语义信息也不同。把两个语义信息差距较大的特征直接相加，势必会减弱多尺度特征的表达能力

简言之：特征求和前不同层次特征之间的语义差异

### 2. 自顶向下的特征融合

自顶向下的特征融合过程就是将顶层的高级特征和底层的低级特征相融合的过程，这样虽然底层的特征得到了来自顶层的高级特征的加强，然而由于\*\*顶层过11降维，势必会造成信息损失\*\*。

简言之：金字塔顶层层次特征的信息丢失

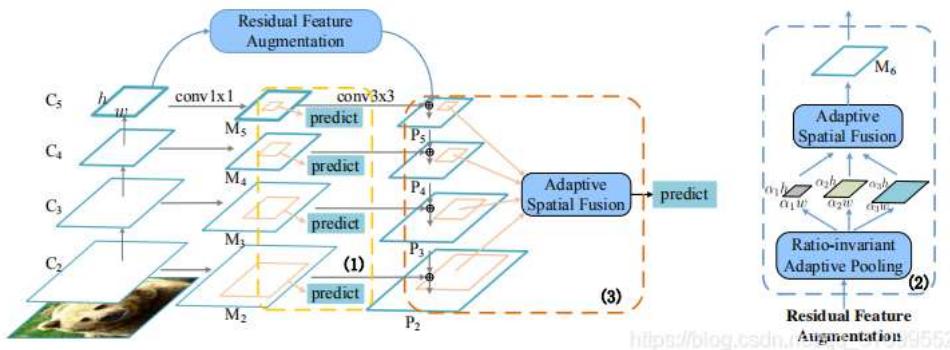
### 3. 特征混合之后

我们知道，FPN中每个候选区域(ROI)的特征都是根据proposal的尺度来决定相应的特征图从哪一层选择，然而那些忽略的层也是包含着丰富信息的，它们对分类和回归结果也有着影响。直接忽略了其它层的特征势必会影响到最终的检测结果。

简言之：RoIs的启发式分配策略

#### AugFPN对应结构

针对FPN中存在的这三个缺陷，这篇论文依次提出三个结构用于解决这三个缺陷。

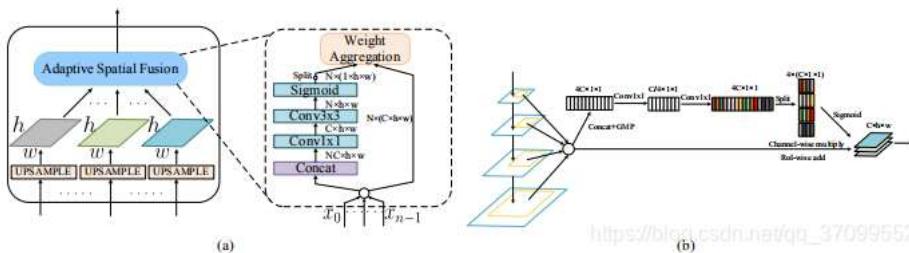


#### • Consistent Supervision

作者对混合之后的特征图M2~M5做了一个监督。具体做法是，对RPN网络得到的每一个候选区域都分别映射到M2~M5上得到相应的feature map，然这些feature map做分类和回归，可以得到一个损失函数。将这一损失和网络本身的损失做一个加权求和。由于在测试阶段，这个分支是不参与计算的Consistent Supervision对模型最终的参数和计算量没有影响。

#### • Residual Feature Augmentation

由于FPN特征融合之前需要先进行1x1的特征降维，这会导致M5的信息损失。为了避免这种信息损失，作者提出了一种很巧妙的办法。因为C5的信息是的，如果能够将C5的信息融合到M5上，就可以有效减少这种信息损失了。为此作者提出了ASF的结构，如Figure 3(a)所示。首先按照比例不变，对C5进行池化操作，然后对每个尺度的特征图进行1\*1降维，然后对降维后的特征进行上采样，对上采样之后的特征按照学到的权重进行相加，把相加完之后称之为M6，再将M6和M5进行相加作为最终的没有损失的M5。



#### • Soft ROI Selection

FPN中ROI对应的特征是根据ROI的尺度选择对应提取的层，一般小的ROI对应低级特征，大的ROI对应高级特征(我的理解，高级感受野更大，负责目标)。然而事实上，这并不是一种最优的办法。因为有时候两个相同大小的ROI可能会被分到不同层的特征，即使某个ROI被分到了某一层，但其实其它也包含着描述这个ROI的信息。因此，论文提出了一种Soft ROI Selection的办法。具体做法是，对于任意一个ROI，我们都会提出它在不同层的特征图的特征，然后利用网络本身学习权重参数，将这些不同层上的特征进行求和作为这个ROI最终的特征，这一结构和Residual Feature Augmentation中适应求和操作是一样的。

#### 总结

本文先分析了FPN的三点缺陷，然后针对这三点缺陷依次提出三种解决策略。

## NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection

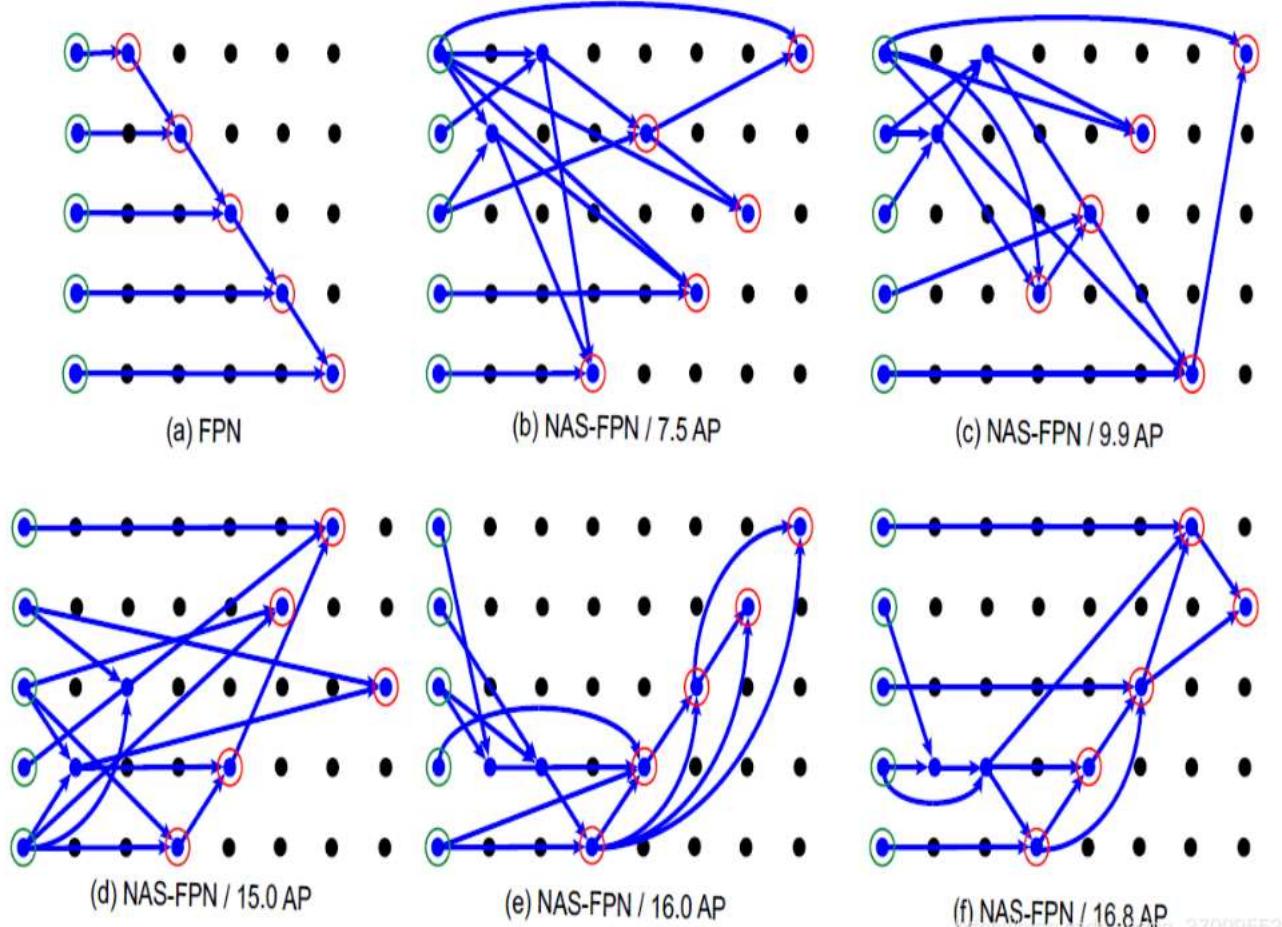
#### Summary

目前最先进的卷积结构用于目标检测是手工设计的。

在这里，我们的目标是一个更好的学习可扩展特征金字塔结构，用于目标检测。在一个覆盖所有交叉尺度连接的可扩展搜索空间中，采用神经网络结构搜索，种新的特征金字塔结构。架构名为NAS-FPN，由自顶向下和自下而上的连接组合而成，可以跨范围地融合特性。

为了发现一个更好的FPN架构，作者充分利用了神经网络搜索技术[Neural architecture search with reinforcement learning.], 使用强化学习训练了-来在给定的搜索空间中选择最好的模型结构。控制器使用在搜索空间内子模型的准确率来作为更新参数的反馈信号(reward signal)。因此，通过这样的试错会学到越来越好的结构，搜索空间在架构成功搜索的过程中起到了很重要的作用。对于FPN的可拓展性，在搜索的过程中，作者强制让FPN重复N次然后concatenation到一起形成一个大的架构。

一句话解释：FPN就是用来特征融合的层，之前都是手工设计，现在尝试神经网络搜索设计！其实就是优化FPN。



NAS-FPN使用强化学习的方式对FPN网络自动搜索，在coco数据集上得到了最好的FPN结果。可以理解为连连看，然后连出一个最好的。如下图：a图为原FPN，好看整齐。然后在一步一步搜索的reward下，结构组件变化。由b→f变为最终的结果。虽然不好看。

免责声明：本内容来自平台创作者，博客园系信息发布平台，仅提供信息存储空间服务。



0

升级成为会员

« 上一篇：单阶段目标检测  
» 下一篇：交叉熵、Focal loss、L1, L2, smooth L1损失函数、IOU Loss、GIoU、DIoU和CIoU

posted @ 2020-02-23 18:34 旅人\_Eric 阅读(2419) 评论(0)