NAS-FCOS: 目标检测任务中神经网络结构的快速搜索

Ning Wang[1], Yang Gao[1], Hao Chen[2], Peng Wang[1], Zhi Tian[2], Chunhua Shen[2] and Yanning Zhang[1]

[1] 计算机学院，西北工业大学，中国
[2] 计算机学院，阿德莱德大学，澳大利亚

摘要

神经网络的成功依赖于一些重要的结构。尽管要耗费大量的计算资源，比如，几千块 *GPU* 跑一天，通过自动搜索最优结构，最近神经网络结构搜索的出现有望大大减少在神经网络结构设计中的人工工作。至今，在有挑战的视觉任务上，如：目标检测，人们对 *NAS*, 尤其是 *NAS* 的快速版本研究得还比较少。在本文中，我们提出在考虑搜索效率的情况下，搜索目标检测器的解码器的结构。更具体来说，使用一种定制的强化学习算法，我们高效地搜素特征金字塔网络 *(FPN)* 和叫做 *FCOS[24]* 的简单的无关键点的目标检测器的预测头部。通过精心设计搜索空间、搜索算法和策略，在评估网络性能时，我们可以在 *4* 天时间内用 *8* 块 *V100 GPU* 高效地搜索出一种表现最优的检测结构。搜索出的结构在 *COCO* 数据集的 *AP* 指标上超过了当前最先进的目标检测模型（比如 *Faster RCNN, RetinaNet* 和 *FCOS)1.5* 到 *3.5* 个百分点，同时该模型的计算复杂性和内存占用与当前最先进的目标检测模型相当，这显示了我们提出的用于目标检测模型的 *NAS* 的高效性。

1 简介

目标检测是计算机视觉的一项基本任务，已经被广泛地研究。最近几年，在目标检测任务上最先进的模型都是基于深度卷积神经网络的，例如 Faster R-CNN[20]，RetinaNet[11]，因为它们在该任务上有着令人惊讶的表现。通常来说，用于目标检测的网络的设计要比图像分类的网络的设计复杂得多，因为前者需要在一张图片中同时对多个目标物体进行定位和分类，而后者只需输出图像级别的标签即可。因为结构复杂，超参数众多，设计有效的目标检测网络更具有挑战性，通常需要更多的人工努力。

另一方面，神经网络结构搜索 (NAS) 方法 [4, 17, 32] 已经在大规模搜索空间中自动寻找最优神经网络结构方面）展现了很好的结果。与人工设计相比，NAS 方法是数据驱动的，而不是经验驱动的，因此需要较少的人为干预。正如在 [3] 中定义的一样，NAS 的工作流可以被分成以下三个过程：1）在一个搜索空间中按照某些搜索策略对网络结构进行采样；2）评估采样得到的结构的表现；和 3）根据表现来更新参数。

阻止 NAS 被在更多的实际应用中使用的主要原因之一就是它的搜索效率。评估过程是最耗费时间的部分，因为它包括了一个神经网络的完整训练过程。为了减少评估时间，实际中，常常使用一个代理任务作为评估过程的低耗时的替代品。在代理任务中，输入，即网络参数和训练迭代次数经常被成比例地缩小以加快评估过程。然而，对于使用代理任务和目标任务的样本来说，二者经常存在一个性能的差距，这就使得评估过程是有偏向的。对于特定问题，如何去设计代理任务，使得它既高效又准确，是一个有挑战的问题。另一个解决搜索效率的方法就是构造一个包含所有搜索空间和共享参数的待选训练结构 [15, 18] 的超网络。然而，这种方法导致了急剧增长的内存消耗，因此只能在小到中等尺寸的搜索空间中使用。

据我们所知，尽管针对目标检测网络的高效准确的 NAS 方法很重要，这方面的研究却很少。为此，针对目标检测网络，我们提出了一种快速而节省内存的 NAS 方法，它可以发现性能最佳的结构，同时极大地节省搜索时间。我们整体的检测网络结构基于 FCOS[24], 一种简单的、无关键点的、单阶段的目标检测网络，其中，

FCOS 的特征金字塔网络和预测头部是由我们提出的 NAS 方法搜索得到的。

我们的主要贡献总结如下：

• 在这篇文章中，我们提出了一种快速的、节省内存的 NAS 方法来搜索 FPN 和头部结构，同时精心设计了代理任务、搜索空间和评估准则，这样，我们的方法可以在 28 块 GPU 工作一天的情况下在 3000 个结构中找到性能最佳的。

具体来说，我们方法的高效性是通过以下的具体设计来实现的。

– 通过跳过骨干网络的微调步骤，形成一个快速的代理任务的训练策略；

– 采用渐进式的搜索策略以减小扩大的搜索空间带来的时间消耗；

– 使用更有判别性的准则对搜索到的结构进行评估；

– 使用一种高效的无关键点的、单阶段的检测网络和简单的后处理过程；

• 通过 NAS，我们探讨了 FPN 和预测头部的工作量之间的关系，证明了在预测头部参数共享的重要性。

• 我们展示了 NAS-FCOS 的总体结构是通用而灵活的，它的骨干网络可以使用多种结构，包括 MobileNetV2, ResNet-50, ResNet-101 和 ResNeXt-101，它超过了当前最先进的目标检测算法，同时计算复杂度与内存占用与之前的方法相当。更具体来说，对于上述的所有模型，我们的模型可以提升 AP 值 1.5 3.5 个百分点。

2 相关工作

2.1 目标检测

目标检测任务的深度神经网络结构可以粗略被分成两种类型：单阶段的检测器 [12] 和双阶段的检测器 [6, 20]。

双阶段的检测网络首先用区域候选网络 (RPN) 产生与类别无关的候选区域，然后使用额外的检测头部对它们进行分类和过滤。尽管达到了最佳性能，双阶段的方法有着明显的缺点：计算复杂度高，超参数众多并且需要根据特定的数据集来调整。

相比之下，单阶段的检测器的网络结构就会简单许多。单阶段的检测器直接在一个 CNN 骨干网络产生的特征图上的每一个位置预测目标物体的类别和边界框的位置。

我们注意到绝大部分的性能最优的检测器 (包括单阶段的检测器 [12, 16, 19] 和双阶段的检测器 [20]) 都是基于特征图上每一个位置处不同大小和不同长宽比的锚盒来做出预测的。然而，使用锚盒可能会导致包含目标物体和不包含目标物体的样本之间极度不平衡，同时需要引入额外的超参数。最近，无关键点、单阶段的检测器 [9, 10, 24, 29, 30] 吸引了很多人的研究兴趣，因为它们的网络结构简单，消耗的计算资源少。

2.2 神经网络结构搜索

NAS 通常很耗时。从 24,000GPU· 天 [32] 到 0.2GPU· 天 [28]，我们已经实现了很大的改进。方法是首先构建一个包含整个搜索空间的超网络，并且使用双层优化和高效的参数共享 [13, 15] 方法将候选结构一次训练完成。

最近，研究者们 [1, 5, 23] 提出使用单路径的训练方法去减小超网络的近似和模型简化所引入的偏差。DetNAS 根据这个想法去搜索高效的目标检测网络结构。单路径方法的一点不足是它的搜索空间被限制成顺序的网络结构。单路径采样和直接估计权值梯度为优化过程引入了很大的偏差，同时在这个框架下不利于搜索出更加复杂的结构。在这个简单的搜索空间下，NAS 算法只为人工设计的模型做一些不重要的决策，如设计卷积核的尺寸大小。

目标检测模型与单路径的图像分类网络不同，前者需要合并多层级的特征，并把检测任务分配给多个并行的预测头部。特征金字塔网络 (FPN)[4, 8, 11, 14, 27] 就是被设计来完成这项任务的，它在一些流行的目标检测模型中起着重要的作用。NAS-FPN 的目的是在基于单阶段的网络 RetinaNet[12] 的情况下，寻找一种可以替代 FPN 的结构。特征金字塔结构是用一个循环神经网络 (RNN) 控制器来采样的。RNN 控制器是用强化学习 (RL) 的方法训练的。然而，尽管一个用 ResNet-10 作为骨干网络的代理任务被训练，用来评估每个结构，搜索过程仍然十分耗时。

既然这三种搜索方法 ([2, 4] 和我们的方法) 都是针对目标检测网络，我们阐述了它们之间的不同：*DetNAS[2] 的目标是搜索设计得更好的骨干网络，而 NASFPN[4] 是搜索 FPN 结构，在我们的方法中，搜索空间包括 FPN 和预测头部结构。*

为了加快基于强化学习的 NAS 的奖励评估过程，[17] 的工作提出使用渐进式任务和其他训练加速方法。通过缓存编码器特征，上述方法可以高效地使用非常大的批次去训练语义分割的解码器。在这篇文章的后续部分，我们把这种技术叫做快速解码器适应。然而，直接把这种方法应用到目标检测任务并不能得到相似的加速效果，因为目标检测任务既没有使用全卷积的模型 [11]，也不需要不能随批量大小扩展的复杂后处理过程 [12]。

为了减小后处理过程的开销，我们采用了一种最近提出的无关键点、单阶段的网络，叫做 FCOS[24], 它通过去掉了 RetinaNet 中锚盒匹配的处理时间，极大地提升了搜索效率。

与基于关键点的同类模型相比，FCOS 显著地减小了训练时的内存使用量，同时又提升了性能。

3 我们的方法

在我们的工作中，我们寻找具有快速解码器适应性的无关键点全卷积检测模型。因此，NAS 方法可以被轻松应用。

3.1 问题提出

我们在一种单阶段的网络 FCOS 上进行搜索算法研究，因为 FCOS 比较简单。我们的训练数据对 $\{(\mathrm{x}, Y)\}$ 是由尺寸为 $(3\times H\times W)$ 的输入图像张量 x 和以金字塔形式表达的 FCOS 目标输出 Y 组成，Y 由一组向量 y_l 组成，每个向量的尺寸是 $((K+4+1)\times H_l\times W_l)$，其中，$H_l\times W_l$ 是金字塔结构第 p 层特征图的尺寸大小。$(K+4+1)$ 是 FCOS 的输出通道数，三项分别是长度为 K 的独热码分类标签，4 个边界框的回归目标值和一个中心因子。

网络 g：$\mathrm{x}\rightarrow\hat{Y}$ 在起初的 FCOS 结构中是由三部分组成的，一个骨干网络 b，FPN f 和多层的子网络，在这篇文章中我们把它叫做预测头部 h。起初，骨干网络 b：$\mathrm{x}\rightarrow C$ 将输入向量映射到一组中级特征 $C=\{c_3,c_4,c_5\}$, 它们的分辨率是 $(H_i\times W_i)=(H/2^i\times W/2^i)$。之后 FPN f：$C\rightarrow P$ 将这些特征映射到特征金字塔 $P=\{p_3,p_4,p_5,p_6,p_7\}$。然后预测头部 h：$\mathrm{p}\rightarrow\mathrm{y}$ 被应用到 P 的每一层，得到的结果就组成了最后的预测结果。为了避免过度拟合，P 中所有的个体通常使用相同的 h。

由于不同大小的目标物体需要不同的有效感受野，选择和合并中级特征 C 的机制在目标检测网络的设计中十分重要。因此，大部分的研究 [16, 20] 都关注于如何设计 f 和 h，同时使用广泛应用的骨干网络结构，比如 ResNet[7]。顺着这个想法，我们的研究目标是去决定何时选择特征，从 C 中选择哪些特征，并且如何融合这些特征。

为了提高效率，我们重复使用了在特定数据集上预训练好的 b 的参数，之后去搜索最优结构。为了下面陈述的方便，我们把要搜索的网络组成部分，即 f 和 h，合称目标检测网络的解码器结构。

f 和 h 在目标检测工作中起不同的作用。f 提取针对不同对象尺度的特征，以金字塔形式 P 来表征，而 h 是应用于 P 中每个特征的统一映射，以避免过拟合。实际中，人们很少讨论使用一种更加多样化的 f 在不同层级提取特征的可能性，或者在 h 中，层级间有多少层网络需要共享参数。在这篇文章中，我们使用一种自动化的方法 NAS 来检验这些可能性。

3.2 搜索空间

考虑到 f 和 h 是不同的映射，我们使用了两个搜索空间。鉴于 FPN 结构的特殊性，一个带有新的整体连接的基本块和 f 的输出设计已经完成了。为了简单起见，h 这一部分使用了顺序的搜索空间。

我们用原子操作代替单元结构，以提供更多的灵活性。为了构建一个基本块，首先我们从采样池 X 中选取编号为 id1，id2 的两层网络 x1，x2，然后对它们都执行两个操作 op1 和 op2，之后使用一种聚合操作 agg，将两个输出融合成一个特征。为了构建深层的解码器，我们使用了多个基本块，并将其输出添加到采样池中。我们的基本块 $bb_t: X_{t-1}\rightarrow X_t$ 在第 t 步将采样池中的 X_{t-1} 转换成 $X_t=X_{t-1}\cup\{\mathrm{x}_t\}$，其中 x_t 是 bb_t 的输出。

ID	具体描述
0	可分离的 3×3 卷积
1	扩张率为 3 的可分离 3×3 卷积
2	扩张率为 6 的可分离 5×5 卷积
3	残差连接
4	可变形的 3×3 卷积

表 1- 搜索过程中使用到的一元操作。

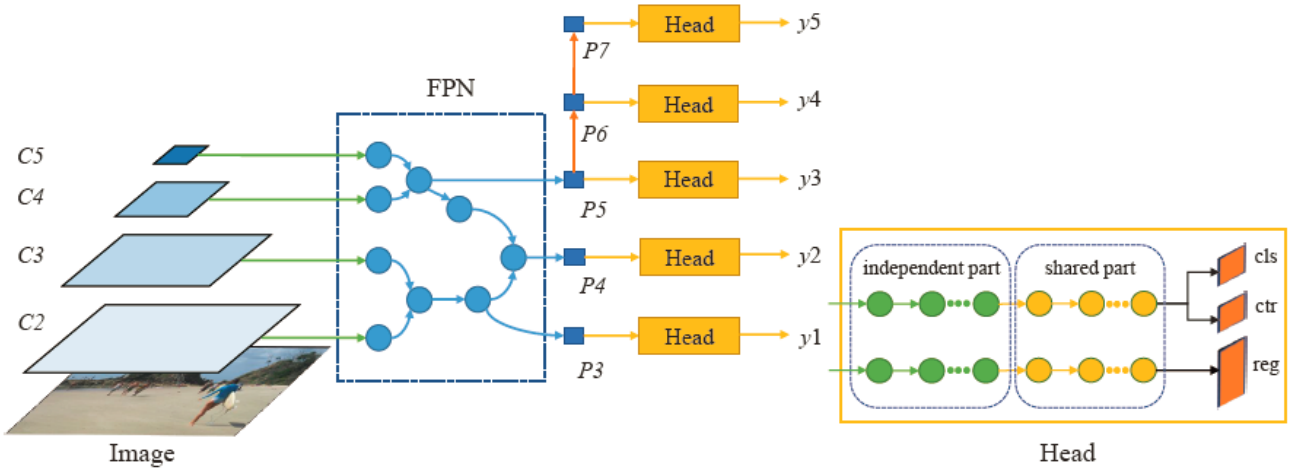

图 1- 这是我们的 NAS-FCOS 解码器的概念性例子。它由两个子网络构成，FPN f 和一组有共享结构的预测头部 h。与其他基于 FPN 的单阶段检测器的一个明显不同是，我们的预测头部存在局部共享的权值。只有预测头部的最后几层网络 (标记为黄色) 通过它们的权重联系起来。需要共享权重的层数是通过搜索算法自动确定的。注意 FPN 和预测头部都在我们实际的搜索空间中；它们的网络层数都比图中展示的更多。这里的图片只是为了解释说明。

候选操作列在了表 1中。为了使解码器更加高效，我们只使用了可分离 (深度方向) 的卷积。为了能够使解码器在不规则网格上使用卷积核，这里我们还使用了 3×3 的不规则卷积 [31]。对于聚合操作，我们使用了逐元素相加、连接，之后是 1×1 卷积操作。

解码器的组成可以用一个由三个部分组成的序列表示，即 FPN，预测头部和权值共享部分。我们在接下来的章节中对它们都做了详细的描述。解码器结构的完整图像在图 1中画出。

3.2.1 FPN 搜索空间

正如在上文中提到的，FPN f 把卷积特征 C 映射到 P。起初，我们初始化采样池 $X_0 = C$。我们的 FPN 是通过在采样池中应用 7 次基本块来定义，$f := bb_1^f \circ bb_2^f \circ \cdots \circ bb_7^f$。为了生成金字塔特征 P，我们把最后三个基本块的输出 $\{x_5,x_6,x_7\}$ 作为金字塔特征 $\{p_3,p_4,p_5\}$。

为了使共享信息能够跨越所有的网络层，我们使用一个简单的方法去创建全局特征。如果存在一个悬空的层 x_t，它既没有被之后的基本块 $\{bb_i^f|\text{i>t}\}$ 采样到，也不属于当 t<5 时的最后三层，那么我们就用逐元素相加的操作将它融合到所有的输出特征中去

$$\mathbf{p}_i^* = \mathbf{p}_i + \mathbf{x}_t, i \in \{3,4,5\} \tag{1}$$

与聚类操作相同，如果特征的分辨率不同，则用双线性插值对较小的特征进行上采样。

为了和 FCOS 保持一致，p_6 和 p_7 分别通过在 p_5 和 p_6 上进行步长为 2 的 3×3 卷积得到。

3.2.2 预测头部搜索空间

预测头部 h 将金字塔 P 中的每一个特征映射到对应的输出 y，在 FCOS 和 RetinaNet 中，预测头部是由 4 个 3×3 的卷积层组成。为了研究预测头部的潜在性能，我们确定了一个顺序的搜索空间来生产它。具体来说，我们把预测头部定义为 6 个基本运算的序列。与 FPN 结构中的候选运算相比，预测头部的搜索空间有两处轻微的不同。首先，为了更好地进行比较，我们把标准卷积模块 (包括 1×1 卷积和 3×3 卷积) 加入到预测头部的采样池中。其次，我们依照 FCOS 的设计，在预测头部的运算采样池中把批标准化 (BN) 层替换为组标准化 (GN)[25]，因为考虑到预测头部需要在不同层级间共享权重，而这会导致 BN 失效。预测头部最终的输出就是最后一层 (第六层) 的输出。

3.2.3 关于预测头部权重共享的搜索

为了增加更多的灵活性和理解在预测头部中权重共享的作用，我们后来又在模型中增加了一个索引 i, 即预测头部开始权重共享的位置。对于 i 之前的每一层，预测头部会为 FPN 的每一个输出层产生一组独立的权重，对于 i 之后的层，预测头部会使用全局的权重以达到共享的目的。

把预测头部的独立部分看作是延伸的 FPN 分支，把共享部分看作是有自适应长度的预测头部，我们可以

进一步平衡工作负荷，为每一个单独的 FPN 分支提取特定级别的特征和所有级别共享的预测头部。

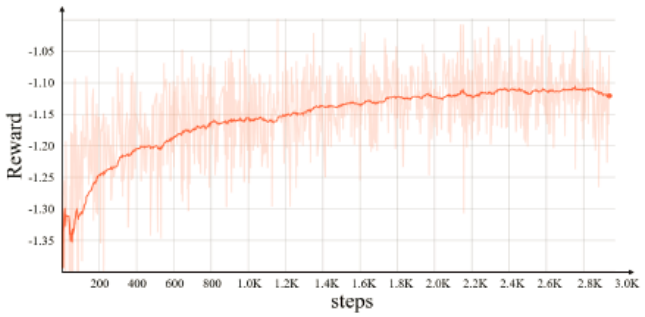

图 2- 在代理任务训练的过程中，激励值一直在上升，这说明强化学习模型是起作用的。

3.3 搜索策略

在搜索过程中我们应用到了基于 RL 的策略。我们依靠一个基于 LSTM 的控制器来预测全部配置。我们考虑使用一种渐进式的搜索策略而不是共同搜索 FPN 结构和预测头部，因为前者花费的计算资源和时间开销更小。训练数据集被随机分成两部分：元训练集 D_t 和元验证集 D_v。为了加速训练，我们固定了骨干网络，缓存了提前计算好的骨干网络的输出 C。这样就使得我们模型训练的花销与骨干网络的深度无关。利用这个优势，我们可以使用更加复杂的骨干网络，获得高质量的多层级特征，作为解码器的输入。我们发现，如果缓存的特征足够有效，可以跳过骨干网络微调这一过程。一些加速技巧，如 Polyak 权值平均在训练过程中也有被用到。

应用最广泛的检测任务的评估准则是平均精度 (AP)。然而，由于检测任务的困难，早期 AP 值太低了，不足以分辨出表现好的结构与不好的结构，这就使得控制器要花费更多的时间去收敛。为了在训练前期，使结构的评估过程容易一些，我们使用了负的损失总和作为评估函数，而不是平均精度：

$$R(a) = -\sum_{(x,Y)\in D_v} (L_{cls}(x, Y \mid a) + L_{reg}(x, Y \mid a) + L_{ctr}(x, Y \mid a)) \quad (2)$$

其中 L_{cls},L_{reg},L_{ctr} 是 FCOS 中的三个损失项。控制器的梯度是通过近端策略优化 (PPO)[22] 来估计的。

4 实验

4.1 实现细节

4.1.1 搜索阶段

我们设计了一个快速的代理任务，用于评估在搜索阶段采样的解码器架构。PASCAL VOC 被选做代理任务的

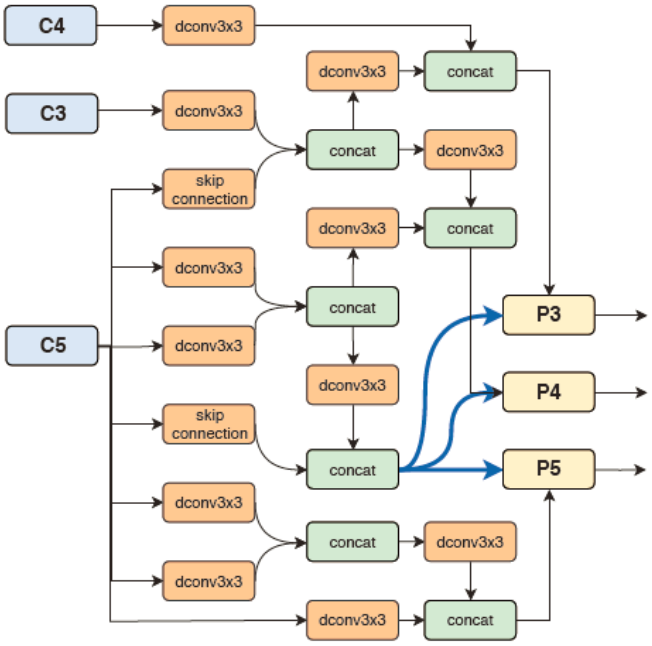

图 3- 这是我们发现的 FPN 结构。C_2 在这张图中被省略了，因为在搜索过程中它不是被此结构选择出来的。

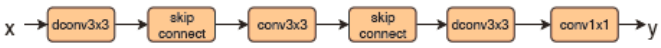

图 4- 我们发现的头部结构

数据集，它包含 5715 张训练图片和 20 个类别的目标物体边界框的标注。模型的迁移能力可以得到证实，因为搜索和完整的训练阶段使用了不同的数据集。VOC 训练集被随机分成两部分，有 4000 张图片的元训练集和有 1715 张图片的元验证集。对于每一个采样到的结构，我们在元训练集上训练，在元验证集上计算它的激励值 (2)。输入图像的大小被调整为短边长为 384，之后被随机裁剪为 384×384 大小，并对感兴趣的目标对象尺寸进行相应的缩放。我们使用 Adam 优化器，学习率设置为 8e-4，批大小设置为 200。同时采用了 Polyak 平均法，衰减率为 0.9。解码器在 300 次迭代后进行评估。由于我们采用了快速解码器自适应方法，在搜索阶段骨干网

解码器	骨干网络	浮点运算次数 (G)	参数量 (M)	平均精度
FPN-RetinaNet @256	MobileNetV2	133.4	11.3	30.8
FPN-FCOS @256	MobileNetV2	105.4	9.8	31.2
NAS-FCOS (我们的) @128	MobileNetV2	39.3	5.9	32.0
NAS-FCOS (我们的) @128-256	MobileNetV2	95.6	9.9	33.8
NAS-FCOS (我们的) @256	MobileNetV2	121.8	16.1	34.7
FPN-RetinaNet @256	R-50	198.0	33.6	36.1
FPN-FCOS @256	R-50	169.9	32.0	37.4
NAS-FCOS (我们的) @128	R-50	104.0	27.8	37.9
NAS-FCOS (我们的) @128-256	R-50	160.4	31.8	39.1
NAS-FCOS (我们的) @256	R-50	189.6	38.4	39.8
FPN-RetinaNet @256	R-101	262.4	52.5	37.8
FPN-FCOS @256	R-101	234.3	50.9	41.5
NAS-FCOS (我们的) @256	R-101	254.0	57.3	43.0
FPN-FCOS @256	X-64x4d-101	371.2	89.6	43.2
NAS-FCOS (我们的) @128-256	X-64x4d-101	361.6	89.4	44.5
FPN-FCOS @256 w/improvements	X-64x4d-101	371.2	89.6	44.7
NAS-FCOS (我们的) @128-256 w/improvements	X-64x4d-101	361.6	89.4	46.1

表 2- 这是完整训练之后在 MS COCO test-dev 集上的结果。R-50 和 R-101 表示 ResNet 骨干网络，X-64x4d-101 表示 ResNeXt-101(64×4d)。所有的网络的输入图像的分辨率是相同的。浮点运算次数和参数量是在 1088×800 大小的输入图像上测量的，这在 COCO 数据集中属于中等大小。对于 RetinaNet 和 FCOS，我们使用作者提供的官方模型。对于我们的 NAS-FCOS，@128 和 @256 分别意味着解码器的通道数是 128 和 256。@128-256 即解码器中，FPN 部分的通道数是 128，预测头部的通道数是 256。为了公平比较，在最新的 FCOS 版本中使用的提升性能的技巧，在我们的模型中也被使用了。

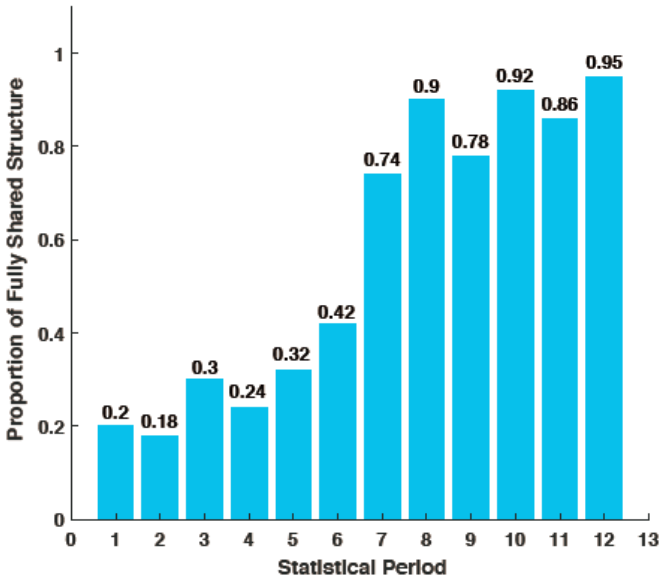

图 5- 搜索过程中头部权值共享的趋势图。横轴坐标表示统计阶段的个数。一个阶段由 50 个头部结构组成。纵轴表示在 50 个结构中完全共享参数的头部的比例。

络得到的特征被固定下来并缓存好。为了增强骨干网络得到的缓存特征，我们最初使用开源的 FCOS 实现提供的预训练权重来初始化它们，之后使用 FCOS 的训

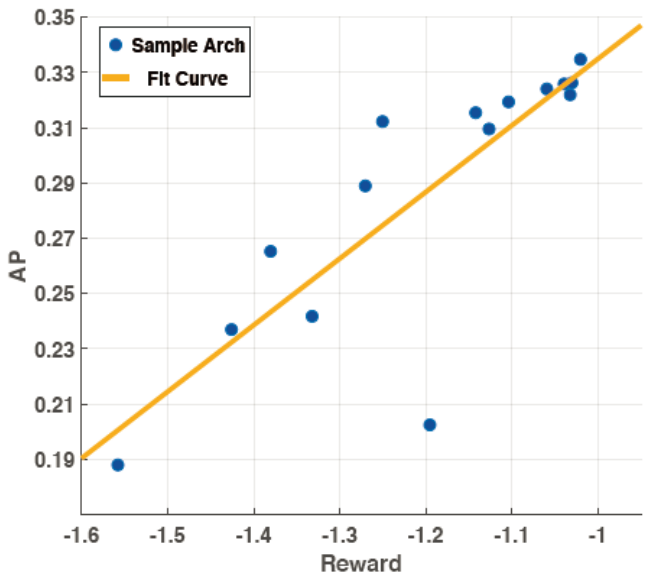

图 6- 从 VOC 元验证集上得到的搜索激励值和从 COCO 验证集上评估得到的 AP 之间的关联。

结构	浮点运算次数 (G)	搜索代价 (GPU-天)	搜索到的结构	平均精度
NAS-FPN @256 R-50	>325.0	333×#TPUs	17000	<38.0
NAS-FPN 7@256 R-50	1125.5	333×#TPUs	17000	44.8
DetNAS-FPN-Faster	-	44	2200	40.2
DetNAS-RetinaNet	-	44	2200	33.3
NAS-FCOS(我们的)@256 R-50	189.6	28	3000	39:8
NAS-FCOS(我们的) @128-256 X-64x4d-101	361.6	28	3000	46.1

表 3- 与其他 NAS 方法的比较。对于 NAS-FPN，输入图像的尺寸是 1280×1280，搜索代价应该通过用来训练每个结构的 TPU 的数量来计算。注意到这里 NAS-FPN @256 的浮点运算次数和平均精度来自 NAS-FPN[4] 中的图 11，NAS-FPN 7@256 将搜索到的 FPN 结构叠加了 7 次。在 DetNASNet[2] 和我们的模型中，输入图像被重新调整大小，以保证短边长为 800 像素。

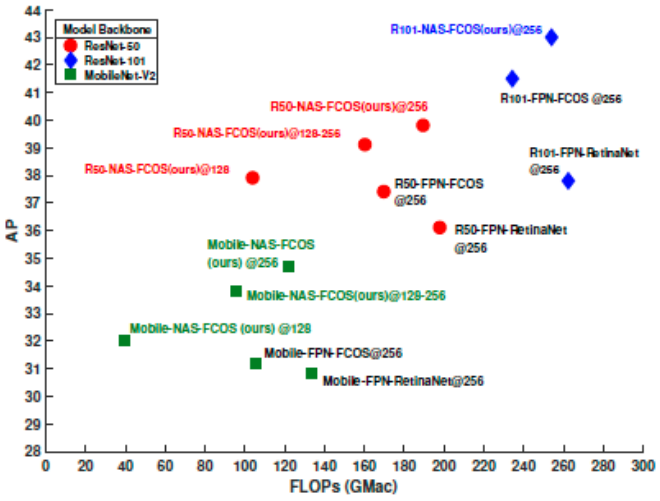

图 7- 不同骨干网络，浮点运算次数和平均精度之间的关系图。不同形状的点代表不同的骨干网络。NAS-FCOS@128 在精度方面有轻微的提升，同时在计算次数上也有一定优势。通道数为 256 的 NAS-FCOS 精度最高，计算复杂性也更高。FPN 通道为 128，预测头部通道为 256(@128-256) 的 NAS-FCOS 在二者之间达成平衡。

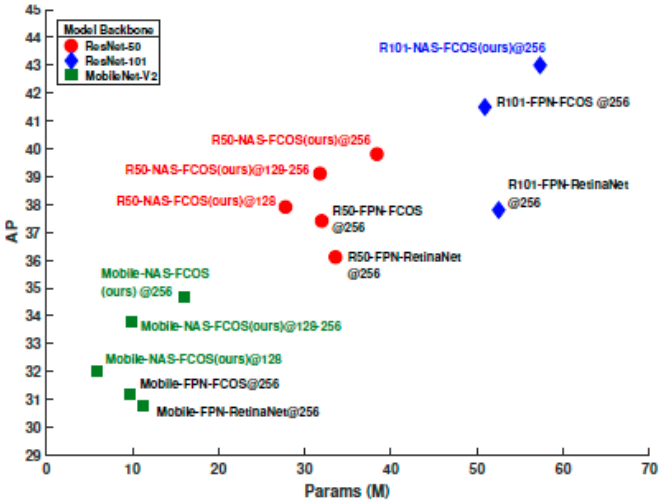

图 8- 不同骨干网络，参数量和平均精度之间的关系图。调整 FPN 和头部中的通道数可以帮助我们在参数量和平均精度之间达成平衡。

练策略在 VOC 数据集上对它们进行微调。注意到上面提到的微调过程仅仅在搜索过程的开始阶段执行一次。

在搜索 f 和 h 的过程中我们使用了渐进式的策略。我们首先搜索 FPN 部分，保持原始的预测头部不变。所有在 FPN 结构中的操作都有 64 个输出通道。解码器的输入 C 的大小通过 1×1 卷积被调整到与 FPN 的输出通道的宽度相符合。这步之后，搜索到的 FPN 结构就被固定了，第二阶段搜索预测头部的工作在此基础上开始。搜索预测头部过程中的大部分参数与搜索 FPN 过程中的一样，除了为了表达更多的信息，将输出通道的宽度从 64 调整到 128。

对于 FPN 搜索过程，控制器模型针对代理任务在搜索了超过 2800 个结构后几乎收敛了，如图 2所示。然后，在代理任务上表现最好的 20 个结构被选择保留下来，用于接下来的完整训练阶段。对于预测头部的搜索过程，我们从表现最好的前 20 个结构中挑出最好的 FPN 结构，提前获取它输出的特征。控制器达到收敛大概需要 600 轮，与搜索 FPN 结构相比，快了很多。之后，我们为后面完整的训练阶段选择出 10 个在代理任务上表现最好的预测头部结构。总之，全部的搜索过程可以在 4 天内用 8 块 V100 GPU 来完成。

4.1.2 完整训练阶段

在这一阶段，我们在 MS COCO 训练集上完整地训练搜索到的模型，并通过在 MS COCO 验证图片上评估它们，挑选出最佳模型。要注意到为了公平比较，我们的训练参数和 FCOS 中的一模一样。输入图片的短边被设置成 800，最大的长边设为 1333。整个模型使用 4 块 V100 GPU 训练，批大小为 16，进行了 90K 次迭

代。初始学习率是 0.01，分别在 60K 和 80K 次迭代时，减小到 0.001 和 0.0001。改进的技巧只应用于最终的模型 (w/improv) 上。

4.2 搜索结果

最佳的 FPN 结构展示在图 3中。控制器发现可变形卷积和连接操作分别是表现最佳的一元操作和聚合操作。从图 4中我们可以看出控制器选择使用 4 个操作 (还有两个跳跃连接)，而不是最多允许的 6 个操作。注意到模型搜索到的“可变形卷积 +1×1 卷积”结构在准确率和浮点运算次数之间达到了很好的平衡。与原始的预测头部相比，我们搜索到的结构有更少的浮点运算次数和参数 (浮点运算次数 79.24G vs 89.16G, 参数 3.41M vs 4.92M)，并且极大地提升了性能 (平均精度 38.7 vs 37.4)。

我们使用搜索到的解码器，搭配要么轻量级的骨干网络，如 MobileNet-V2[21]，要么是比较强大的骨干网络，比如 ResNet-101[7] 和 ResNeXt-101[26]。为了平衡精度与效率，我们实现了三种计算代价不同的解码器：一个的特征维度是 128(@128)，一个的特征维度是 256(@256)，另一个的 FPN 通道的宽度是 128，预测头部是 256(@128-256)。在 COCO 测试集中的验证结果 (短边长为 800) 显示在表 2中。在不同的骨干网络下，搜索到的特征维度是 256 的解码器在平均精度中超过了 FCOS 中的对应结构 1.5 到 3.5 个百分点。特征维度是 128 的解码器 (@128) 极大地减少了参数和计算量，这使得它更适合在对资源要求有限制的环境中使用。特别地，我们搜索到的 128 个通道的模型和 MobileNetV2 骨干网络超过了采用相同骨干网络的原始的 FCOS 模型平均精度 0.8 个百分点，却仅仅使用了 FCOS 1/3 的计算量。第三种解码器 (@128-256) 在精度和参数上达到了一个很好的平衡。我们搜索到的模型超过了表现最好的 FCOS 模型的变体平均精度 1.4 个百分点 (46.1 vs. 44.7), 运算量和参数量略有减少。和其他模型在运算量和参数量上的对比，分别显示在图 7和图 8中。

为了理解在预测头部中进行权重共享的重要性，我们把共享参数的层的数量作为搜索的一个目标。图 5展示了搜索过程中头部结构权值共享的趋势图。我们设定 50 个结构为一个统计周期。随着搜索过程深入，参数完全共享的结构出现的比例增加，说明在多尺度检测模型中，头部的权值共享是必需的。

在表 3中，我们也展示了我们的模型与其他针对目标检测任务的 NAS 方法的对比。我们的方法每个 GPU-天可以搜索到比 DetNAS[2] 的两倍还多的结构。要注意 NAS-FPN[4] 的平均精度是通过叠加 7 次搜索到的 FPN 结构来得到的，然而我们并没有叠加搜索到的 FPN 结构。我们以 ResNeXt-101(64x4d) 作为骨干网络的模型超过了 NAS-FPN 平均精度 1.3 个百分点，却仅仅使用了它 1/3 的浮点运算次数和更少的计算代价。

我们还进一步衡量了搜索过程中在代理数据集上得到的激励值与采用相同的结构在 COCO 数据集上训练后得到的平均精度之间的关联。特别地，我们随机从搜索到的所有结构中选取了 15 个结构，它们都是在 COCO 数据集上训练好的，批大小为 16。由于在 COCO 数据集上完整地训练很消耗时间，我们把迭代次数减少到 60K 次。然后在 COCO 2017 验证集上评估模型。从图 6中可以看出，搜索过程中的激励值与在 COCO 数据集上获得的平均精度之间有很强的关联。表现差和表现好的结构可以很好地通过它们在代理任务中的激励值来区分。

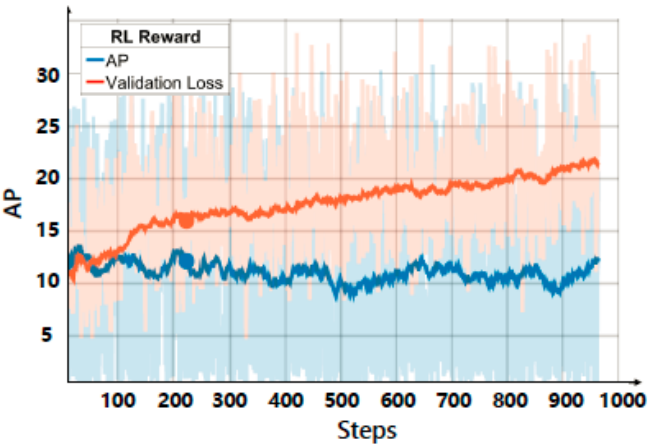

图 9- 两个设计不同的强化学习激励值之间的比较。纵轴表示在代理任务的验证集上得到的平均精度。

4.3 消融实验

4.3.1 强化学习激励值的设计

如上文所述，在搜索过程中对于特定任务使用被广泛认可的指标作为激励是很常见的，例如分割任务中使用 mIOU，目标检测任务中使用平均精度 AP。然而，我们发现在短期搜索轮次中，使用平均精度 AP 作为激励并没有带来明显的上升趋势 (图 9中的蓝色曲线)。我们进一步分析可能的原因：控制器尝试去学习从解码器到激励值的一种映射，然而 AP 本身计算太过复杂，使得控制器很难在有限的迭代次数中学会这种映射。相比较

解码器	搜索空间	平均精度
FPN-FCOS @256	-	37.4
NAS-FCOS @256	只有 h	38.7
NAS-FCOS @256	只有 f	38.9
NAS-FCOS @256	$f+h$	39.8

表 4- 用 ResNet-50 作为骨干网络，在不同的搜索空间下获得的平均精度之间的比较。

而言，当用验证损失作为强化学习的激励时，我们可以清楚地看到 AP 的增长 (图 9中的红色曲线)。

4.3.2 搜索空间的有效性

为了进一步讨论搜索空间 f 和 h 的影响，我们设计了 3 个实验来验证。一个实验是在固定原始的头部的情况下搜索 f, 一个是固定原始的 FPN 不变后搜索 h，另一个实验搜索整个解码器 (f+h)。从表 4中可以看出，仅仅搜索 f 比仅仅搜索 h 的效果稍微好一些。并且我们的渐进式搜索对 f 和 h 都进行了搜索，取得了更好的结果。

4.3.3 可变形卷积的影响

如前所述，在搜索 f 和 h 的过程中，可变形卷积都被包括在候选操作集中，它能够适应目标物体的几何变化。为了公平比较，我们把原始 FCOS 中的 FPN 结构里的标准 3×3 卷积替换为可变形 3×3 卷积，并把它们叠加了两次，这样做是为了保证新模型和我们搜索到的结构的浮点运算次数和参数量几乎相等。因此新模型被叫做 DeformFPN-FCOS。在这种情况下，我们的 NAS-FCOS 模型 (只搜索 FPN 时 AP=38.9，FPN 和头部都进行搜索时 AP=39.8) 仍然比 DeformFPN-FCOS 模型 (AP=38.4) 取得了更好的性能。

5 结论

在这篇论文中，我们提出使用神经网络结构搜索进一步优化目标检测网络的设计过程。我们的工作表明：在精心设计代理任务、搜索策略和模型评估准则的情况下，可以高效地搜索到性能最佳的检测器。在 COCO 数据集上的实验证实了我们搜索到的模型具有高效性和灵活性，它可以使用不同的骨干网络结构。

参考文献

[1] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: 在特定任务与硬件上进行神经网络结构搜索. arXiv preprint arXiv:1812.00332, 2018.

[2] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Chunhong Pan, and Jian Sun. DetNAS: 在目标检测任务上的神经网络结构搜索. arXiv preprint arXiv:1903.10979, 2019.

[3] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 关于神经网络结构搜索的一项调查. arXiv preprint arXiv:1808.05377, 2018.

[4] Golnaz Ghiasi, Tsung-Yi Lin, Ruoming Pang, and Quoc V.Le. NAS-FPN: 在目标检测任务中学习可扩展的特征金字塔结构. Proc. IEEE Conf. Comp. Vis. Patt.Recogn., 2019.

[5] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. 单路一次通过均匀采样的神经结构搜索. arXiv preprint arXiv:1904.00420, 2019.

[6] Kaiming He, Georgia Gkioxari, Piotr Doll´ar, and Ross Girshick.Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 深度残差网络中的特征映射. In European conference on computer vision, pages 630–645. Springer, 2016.

[8] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollr. 全景特征金字塔网络. arXiv: Computer Vision and Pattern Recognition, 2019.

[9] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: 优于基于关键点的目标检测器. arXiv preprint arXiv:1904.03797, 2019.

[10] Hei Law and Jia Deng. Cornernet: 以成对的关键点来检测目标物体. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750, 2018.

[11] Tsung-Yi Lin, Piotr Doll´ar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 用于目标检测的特征金字塔网络. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 2117–2125, 2017.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Doll´ar. 密集物体检测中的焦点损失函数. In Proc.IEEE Conf. Comp. Vis. Patt. Recogn., pages 2980–2988, 2017.

[13] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. Auto-deeplab: 用于图像语义分割的分层神经网络结构搜索. Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2019.

[14] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. 一种用于全景分割的端到端网络. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2019.

[15] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: 可微分的结构搜索方法. arXiv preprint arXiv:1806.09055, 2018.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander Berg. SSD: Single shot multibox detector. In Proc. Eur. Conf. Comp. Vis., pages 21–37. Springer, 2016.

[17] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. 通过辅助单元快速搜索紧凑语义分割模型的神经架构. Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2019.

[18] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. 通过参数共享的高效神经网络结构搜索方法. In ICML, 2018.

[19] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv, 2018.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: 带有区域提取网络的实时目标检测器. In Proc. Advances in Neural Inf. Process. Syst., pages 91–99, 2015.

[21] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 4510–4520, 2018.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv: Comp. Res. Repository, 2017.

[23] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-path NAS: Designing hardware-efficient convnets in less than 4 hours. arXiv preprint arXiv:1904.02877, 2019.

[24] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: 单阶段全卷积目标检测. arXiv preprint arXiv:1904.01355, 2019.

[25] Yuxin Wu and Kaiming He. Group normalization. In Proceedings of the European Conference on Computer Vision(ECCV), pages 3–19, 2018.

[26] Saining Xie, Ross Girshick, Piotr Dollr, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. arXiv preprint arXiv:1611.05431, 2016.

[27] Ting Zhao and XiangqianWu. Pyramid feature attention network for saliency detection. arXiv: Computer Vision and Pattern Recognition, 2019.

[28] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. BayesNAS: A bayesian approach for neural architecture search. arXiv preprint arXiv:1905.04919, 2019.

[29] Xingyi Zhou, Dequan Wang, and Philipp Kr¨ahenb¨uhl. Objects as points. In arXiv preprint arXiv:1904.07850, 2019.

[30] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. arXiv preprint arXiv:1903.00621, 2019.

[31] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. arXiv preprint arXiv:1811.11168, 2018.

[32] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 2016.