

课程论文题目

Oriented RCNN for Object Detection

Abstract

当前主流的双阶段检测方法在候选框提取阶段非常耗时，这会降低检测器的速度，从而成为目标检测系统的计算瓶颈。本文提出了一种有效的、简单的目标检测框架，称为 Oriented R-CNN，它是一种通用的双阶段检测器，具有优秀的精度和效率。具体来说，在第一阶段，我们提出了一个候选框提取网络（RPN），它以低时耗的方式直接生成高质量的提案选择框。第二阶段是面向 R-CNN 的分类网络，用于细化感兴趣区域并识别它们。在不使用任何技巧的情况下，在目标检测领域常用的两个数据集 DOTA（75.87%）和 HRSC2016（96.50%）上的实现最优的准确率和 15.1FPS 的优良速度。

在复现的基础上，我尝试对网络模型进行改进，包括对网络架构的改进和使用一些训练技巧，最终完成的网络模型能够在 DOTA 数据集上准确率达到 88.63%，较原论文内容提升了 7.5%。

关键词：旋转目标检测；候选框提取网络；双阶段检测器

1 引言

目标检测是计算机视觉研究的重点问题之一，是理解图像高层语义信息的重要基础。其主要任务是根据输入图像定位感兴趣的目标并给出类别信息，输出结果通常是在输入图像的基础上，以边界框（Bounding Box）的形式标识目标的具体位置并在矩形框的上方显示类别信息和置信度分数。

现有的最先进的目标检测算法大多都依赖于候选区域提取的框架 [2, 14]。候选区域提取的早期方案是旋转候选区域提取网络 (Oriented RPN)，其方案是在每个锚点位置放置 54 个不同角度和尺度的旋转候选框 [8]。这种方法对于稀疏目标的图像检测具有较好的性能，但是对于稠密目标的图像性能欠佳并且非常消耗资源。为了解决这个问题，ROI Transformer 网络 [2] 通过复杂的过程从水平的候选区域中学习得到旋转候选区域。这个思路提供了很好的旋转框提取方案，并且大大减小了旋转锚框的数量，但同时带来了昂贵的计算成本。现在，如何设计一个优雅而高效的解决方案来生成面向建议的方案是打破最先进的旋转检测器计算瓶颈的关键。

在此之前，基于区域提案的定向检测器效率一直落后于其他方法。我们实验得到阻碍提案驱动探测器速度提高的主要障碍是来自于提案生成阶段，也就是候选区域生成阶段。一个自然而直观的问题是：我们能否设计一个通用的、简单的旋转区域提案网络（简称旋转 RPN）来直接生成高质量的旋转提案？基于此问题，本文提出了一种简单的双阶段目标检测框架，即旋转 R-CNN。本文提出的方法在时间效率上可以与单阶段旋转目标检测方法持平。

具体来说，在定向 R-CNN 的第一阶段，给出了一个概念上简单的旋转 RPN。我们的定向 RPN 是一种轻量级的全卷积网络，其参数比旋转的 RPN 和 RoI Transformer 的参数量减少许多，从而降低了过拟合的风险。我们通过将 RPN 回归分支的输出参数数从 4 个改为 6 个来实现这一点。天下没有免费的午餐。旋转 RPN 的设计得益于我们提出的面向对象的提案表示方法，即中点偏移表示法。对于图像中任意旋转的对象，我们使用 6 个参数来表示它们。中点偏移表示法继承了水平旋转框的回归机

制，并为预测旋转角度的建议提供了有界约束。旋转 R-CNN 的第二阶段是旋转 R-CNN 检测头：通过旋转 RoI 对齐提取的每个面向提案的特征，并进行分类和回归。

在没有使用任何技巧的情况下，原论文在两个流行的面向对象检测基准上评估我们的旋转 CNN，即 DOTA 和 HRSC2016。我们的 ResNet-50-FPN 方法超过了所有现有最先进探测器的精度，在 DOTA 数据集上实现了 75.87% 映射，在 HRSC 数据集上达到了 96.50% 检测。因此，面向 R-CNN 在准确性和效率方面都是一个实用的目标检测框架。

这篇论文为目标检测候选区域提取提供了良好的思路，这项工作取得的成就和贡献在旋转目标检测方向可以说是里程碑性质的，所以很具有复现研究意义。

2 相关工作

目标检测领域可划分为水平目标检测和旋转目标检测两大方向。一般的水平目标检测方法不能够紧密定位图像中的旋转物体，因为一个水平框除了包含检测物体外还会包含部分背景区域或多个对象，这就导致了最终的分类置信度和定位精度之间不一致的问题。相较于水平目标检测，旋转目标检测采用不同的旋转框参数定义方式，将旋转角度融入到目标检测框中，从而使框体能够更加紧密地贴合目标物体。

对于旋转目标检测地研究，科研学者们进行了大量的实验 [1, 11, 17]。例如，Xia 等人 [12] 构建了一个具有旋转标注的大型对象检测基准，命名为 DOTA。许多现有的面向对象检测器 [2, 13, 4] 主要是基于基于候选框提案的目标检测框架。旋转物体识别的一个自然解决方案是设置旋转的锚点 [8, 6]，如旋转的 RPN[8]，其中具有不同角度、比例和纵横比的锚点被放置在每个位置上。这些密集旋转的锚点会导致大量的计算和内存占用。

为了减少大量旋转锚的数量，减少特征与物体之间的不匹配，Ding 等 [2] 提出了 RoI 变压器，从 RPN 产生的水平 RoI 中学习旋转 RoI。这种方式大大提高了定向目标的检测精度。然而，在学习旋转 RoI 的过程中，它涉及到全连接层和 RoI 对齐操作，这使得网络变得沉重和复杂。为了解决旋转目标检测任务中小目标识别的挑战，Yang 等 [16] 在 Fast R-CNN 的通用目标检测框架上建立了一种旋转目标检测的方法。Xu 等人 [13] 提出了一种新的旋转目标框表示方法，称为滑动顶点。它通过在 Faster R-CNN 检测头的回归分支上学习四个顶点滑动偏移量来实现定向目标检测。然而，这两种方法都采用水平 roi 来进行分类和旋转边界盒回归，仍然存在分类置信度和定位精度失调的问题。

此外，[3, 10, 15, 9] 的一些工作还探索了单阶段或无锚点的面向对象检测框架：输出对象类和面向边界框，而不需要区域建议生成和 RoI 对齐操作。例如，Yang 等人 [15] 提出了一种改进的面向一期对象检测器，它涉及两个关键的改进，包括特征细化和渐进回归，以解决特征错位的问题。Ming 等 [9] 设计了一种基于视网膜网络 [5] 的面向一期目标检测的标签分配策略。它通过一种新的匹配策略动态地分配正锚或负锚。Han 等人 [3] 提出了一种用于定向目标检测的单镜头对齐网络 (S2ANet)。S2ANet 旨在通过深度特征对齐来缓解分类评分和位置精度之间的不一致性。Pan 等人 [10] 设计了一种基于无锚目标检测方法 CenterNet 的面向目标检测的动态细化网络 (DRN)。

与上述方法相比，复现论文的工作属于基于提案的旋转提案的目标检测方法。它着重设计一个高效的 RPN，以解决生成旋转建议候选区域的计算瓶颈。

3 本文方法

3.1 本文方法概述

论文中提出的目标检测方法，称为旋转 R-CNN，由一个旋转 RPN 和一个旋转 RCNN 头组成（见图 1）。它是一个两阶段的检测器，第一阶段以几乎免费的方式生成高质量的旋转候选框，第二阶段是面向 RCNN 头进行提案分类和回归。我们的 FPN 主干遵循 [7]，它产生了 5 个层次的特征 P2、P3、P4、P5、P6。接下来，主要对旋转 RPN 进行详细描述。

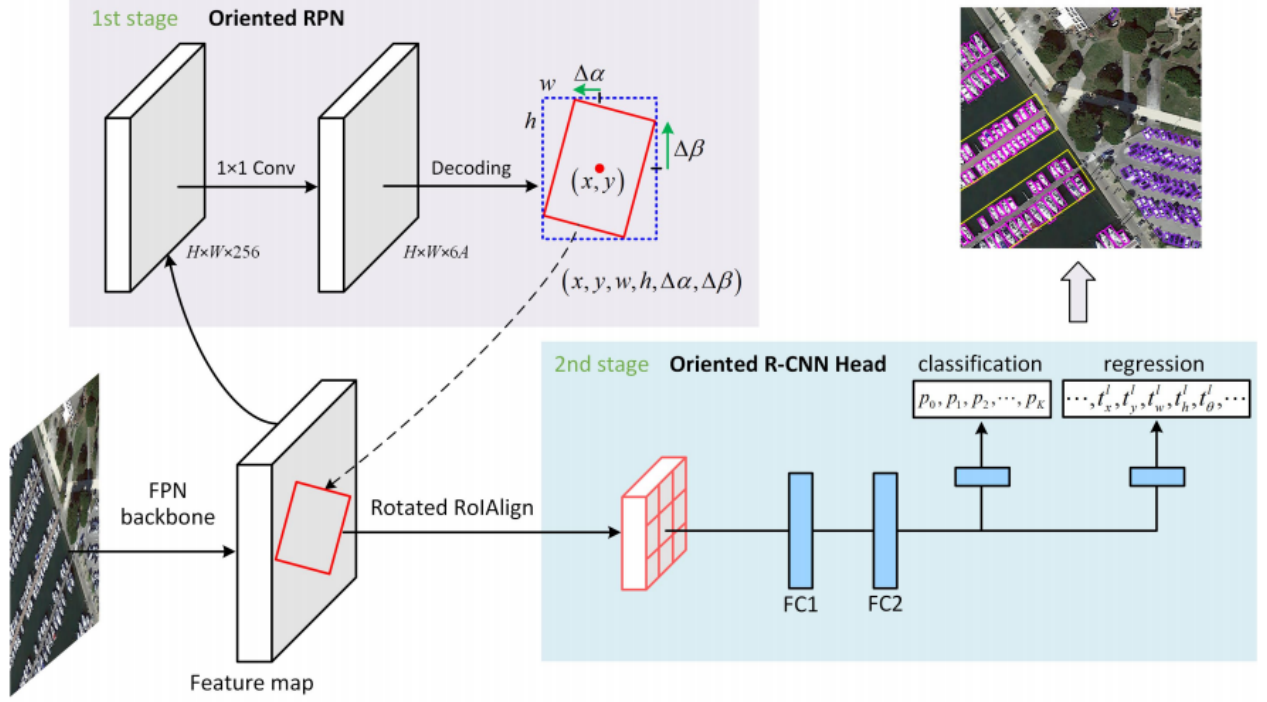


Figure 1: 主干网络图示

3.2 候选区域提取网络 (RPN)

给定任何大小的输入图像，面向定向 RPN 输出一组稀疏的定向建议。整个过程都可以用轻量级的全卷积网络来建模。

具体来说，它将 FPN 的五个层特征 P2、P3、P4、P5、P6 作为输入，并将一个相同设计的头部（一个 3×3 卷积层和两个 1×1 卷积层）附加到每个特征层上。我们分配了三个水平锚和三个比例 1: 2, 1: 1, 2: 1 到所有层次特征的每个空间位置。锚点在 P2、P3、P4、P5、P6 上分别有 32², 64², 128², 256², 512² 的像素区域。每个锚框 \mathbf{a} 用一个四维向量 $\mathbf{a} = (a_x, a_y, a_w, a_h)$ 表示，其中 (a_x, a_y) 是锚的中心坐标， a_w 和 a_h 表示锚的宽度和高度。两个 1×1 卷积层中有一个是回归分支：输出相对于锚的偏移 $\delta = (\delta_x, \delta_y, \delta_w, \delta_h, \delta_\alpha, \delta_\beta)$ 。在特征图的每个位置，我们生成 A 个候选框（A 是每个位置的锚点数量，在本工作中等于 3），因此回归分支有 A 个长度为 6 的输出。通过对回归输出的解码，我们可以得到面向方向的方案。解码的过程描述如下：

$$\begin{cases} \Delta\alpha = \delta_\alpha \cdot w, & \Delta\beta = \delta_\beta \cdot h \\ w = a_w \cdot e^{\delta_w}, & h = a_h \cdot e^{\delta_h} \\ x = \delta_x \cdot a_w + a_x, & y = \delta_y \cdot a_h + a_y \end{cases} \quad (1)$$

3.3 中点偏移表示

我们提出了一种新的旋转物体表示方案，称为中点偏移表示，如图 2所示。黑点是水平框每边的中点，这是旋转边界框 O 的外部矩形。橙色的点代表旋转边界框 O 的顶点。

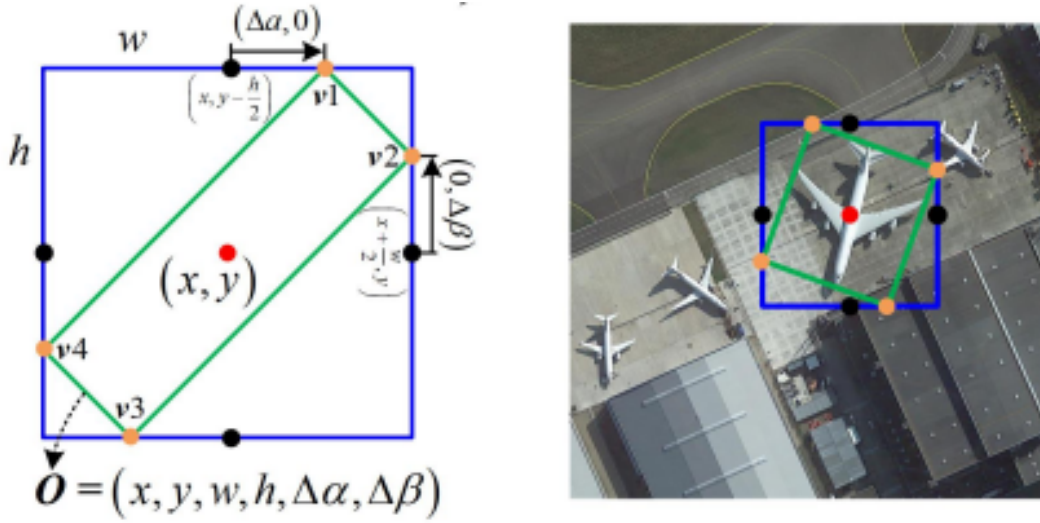


Figure 2: 中心点偏移

具体来说，我们使用一个带有六个参数 $O = (x, y, w, h, \Delta\alpha, \Delta\beta)$ 的旋转边界框 O 来表示一个由式 (1) 计算的物体。通过这 6 个参数，我们可以得到每个方案的四个顶点的坐标集 $v = (v1, v2, v3, v4)$ 。在这里， $\Delta\alpha$ 是 $v1$ 相对于水平盒顶部一侧的中点 $(x, y - h/2)$ 的偏移量。根据对称性， $-\Delta\alpha$ 表示 $v3$ 相对于底部中点 $(x, y + h/2)$ 的偏移量。 $\Delta\beta$ 表示 $v2$ 相对于右中点的偏移量 $(x + w/2, y)$ ， $-\Delta\beta$ 表示 $v4$ 相对于左中点的偏移量 $(x - w/2, y)$ 。因此，四个顶点的坐标可以表示如下。

$$\begin{cases} v1 = (x, y - h/2) + (\Delta\alpha, 0) \\ v2 = (x + w/2, y) + (0, \Delta\beta) \\ v3 = (x, y + h/2) + (-\Delta\alpha, 0) \\ v4 = (x - w/2, y) + (0, -\Delta\beta) \end{cases} \quad (2)$$

通过这种表示方式，我们通过预测其外部矩形的参数 (x, y, w, h) 和推断其中点偏移量的参数 $(\Delta\alpha, \Delta\beta)$ 来实现每个定向方案的回归。

3.4 损失函数定义

为了训练定向的 RPN，正样本和负样本的定义如下。首先，我们为每个锚点分配一个二进制标签。这里，0 和 1 表示锚定点属于正样本和负样本。具体来说，在以下两种情况下，我们认为锚作为正样本：(i) 锚框与任何地面真实盒的重叠程度高于 0.7；(ii) 锚框与地面真实盒的重叠程度最高，且 IoU 高于 0.3。当锚的 Iou 低于 0.3 时，锚框被标记为负样本。既不正也不负锚被认为是无效样本，在训练过程中输入。值得注意的是，上述地面真值框指的是旋转边界框的外部矩形。

损失函数的定义如下：

$$L_1 = \frac{1}{N} \sum_{i=1}^N F_{cls}(p_i, p_i^*) + \frac{1}{N} p_i^* \sum_{i=1}^N F_{reg}(\delta_i, t_i^*) \quad (3)$$

这里， i 是锚点的索引， N （默认为 $N=256$ ）是一个小批量中的样本总数。 p_i^* 是第 i 个锚的地面真实标签。 p_i 是面向 RPN 的分类分支的输出，表示建议框属于前景的概率。 t^* 是相对于地面真实盒的偏移量。 F_{cls} 是交叉熵损失。 F_{reg} 是光滑的 L1 损失。对于 box 回归，我们采用仿射变换，其公式

如下：

$$\left\{ \begin{array}{ll} \delta_{\alpha} = \Delta\alpha/w, & \delta_{\beta} = \Delta\beta/h \\ \delta_w = \log(w/w_a), & \delta_h = \log(h/h_a) \\ \delta_x = (x - x_a)/w_a, & \delta_y = (y - y_a)/h_a \\ t_{\alpha}^* = \Delta\alpha_g/w_g, & t_{\beta}^* = \Delta\beta_g/h_g \\ t_w^* = \log(w_g/w_a), & t_h^* = \log(h_g/h_a) \\ t_x^* = (x_g - x_a)/w_a, & t_y^* = (y_g - y_a)/h_a \end{array} \right. \quad (4)$$

式中 (x_g, y_g) 、 w_g 和 h_g 分别为外部矩形的中心坐标、宽度和高度。 $\Delta\alpha_g$ 和 $\Delta\beta_g$ 是上顶点和右顶点相对于上边和左边的中点的偏移量。

4 复现细节

4.1 与已有开源代码对比

本次复现工作是在现有代码的基础上进行的，主要是参考了现有代码的整体架构和部分网络模型，一方面是为了能够更快速地完成复现工作，另一方面是为了能够更准确地实现复现效果。在复现的基础上，我对实验进行了一些改进与创新，创新的最突出的结果是在原论文的基准上将模型 mAP 提高了 7.5%，使网络模型在 Dota1.0 数据集上的准确率达到 88.63%。主要创新包括：数据集预处理、特征提取网络创新、特征融合网络创新、训练策略创新。复现过程参考已有代码的地址为：<https://github.com/jbwang1997/OBBDetection>。

4.2 实验环境搭建

实验平台环境为 Linux 系统，GPU 为 RTX 3090 24GB 显存，cpu 为 16GB 8 核处理器。实验语言使用 python 3.8 版本，编码框架使用 Jittor1.3.6 版本，cuda 版本为 11.3。对于实验所必须的一些库文件，我将这些必备库都写到了 requirements.txt 文件，只需要运行 `python -m pip install -r requirements.txt` 指令即可完成对相关库的安装。

4.3 创新点

4.3.1 数据集预处理

为了能够对模型更好的训练，使网络模型能够更快地收敛，我对 Dota1.0 数据集进行了预处理操作，主要包括：样本均衡、图像归一化、随即裁剪、随即翻转、随机旋转、多尺度缩放。在预处理操作之前，我先对数据集进行了分析，主要是对数据集中的检测目标的类别和尺度进行分析，分别统计出数据集中样本尺度分布和样本类别分布，依次为依据进行数据预处理操作。

4.3.2 特征提取网络

原论文中使用的特征提取网络使 ResNet50 和 ResNet101，考虑到基本的残差块对小目标物的提取效果不佳，这里我使用特征提取粒度更小、网络结构更加复杂的 Res2Net50 网络进行特征提取。Res2Net 网络与 ResNet 相比，在特征提取的同时融合了对尺度信息的提取，其使用较小的滤波器组以类似于残差的分层方式连接，以增加输出特征可以表示的尺度数量，使网络模型可以提取细节特征和全局特

征。

图 3 中 (a) 左侧所示的残差结构是原论文中特征提取模块的基本构建块。Res2Net 不像在残差块中那样使用一组 3×3 过滤器来提取特征，而是寻找具有更强的多尺度特征提取能力的替代架构，同时保持类似的计算负载。具体来说，是用一组更小的过滤器替换一组 3×3 过滤器，同时以类似残差的分层方式连接不同的过滤器组。

4.3.3 特征融合网络

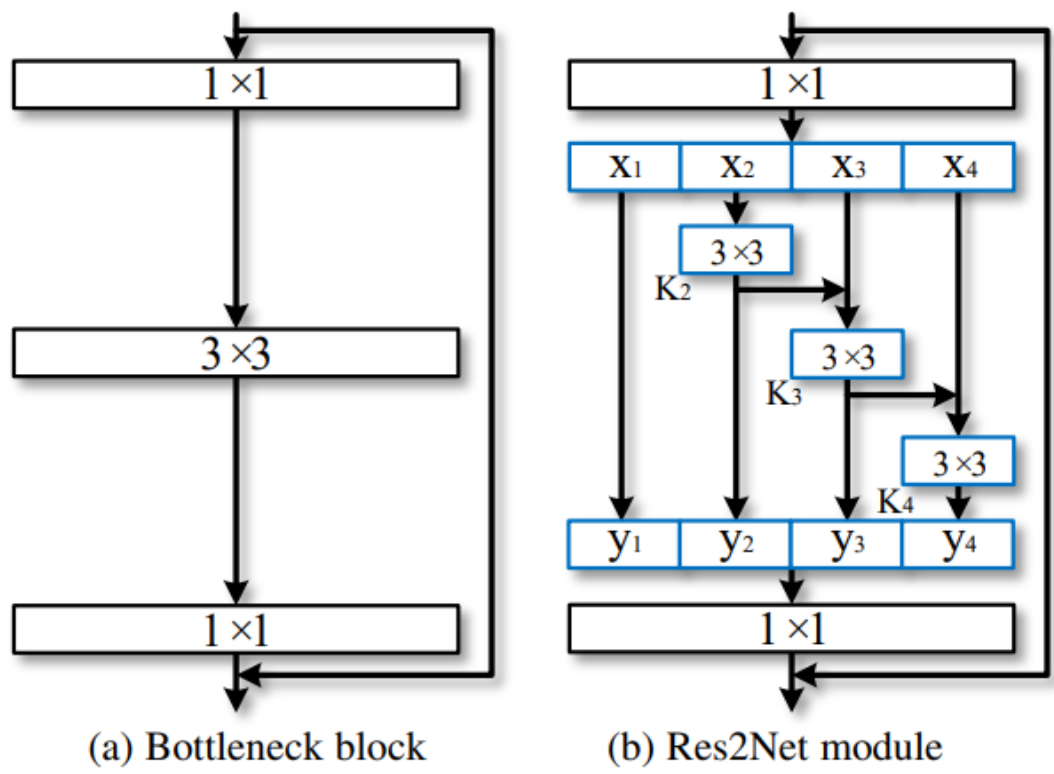


Figure 3: ResNet(a) 与 Res2Net(b) 的基础块

特征融合网络主流模型是考虑到对多尺度信息的提取与融合，原论文中使用的网络模型结果是 FPN 网络结构。FPN 网络能够融合特征提取网络中多个阶段提取到的不同尺度的特征图，这些不同尺度的特征图代表着不同的语义信息，将这些语义信息进行融合可以使提取到的更加丰富的特征信息。但是，对 FPN 的网络结构进行观察可以发现，FPN 网络的特征融合阶段只有从上到下的单方向的特征融合，这么做会使顶部的特征图无法共享到底部的特征信息，而顶部特征图代表着更全局的信息，底部特征图代表更细节的信息，所以这么做不利于对细节和纹理信息的提取。基于这个原因，我尝试使用 PAFPN 网络和 ASFF 网络进行特征融合，二者都是在 FPN 的基础上进行创新的。

PAFPN 的网络架构如图 4 所示。图中，红色虚线表示原始图像经过数十甚至近百层的卷积得到的特征图 P5，P5 已经过滤了大多数的细节信息。PAFPN 相比 FPN 增加了 (b) 部分的 down-top 连接，将底层细节通过较短的路径融合到上层，即绿色虚线所示的路径 (P2->N2->N3->N4->N5)。图中的 top-down 连接是通过核大小为 3×3 ，步长为 1 的卷积层卷积后再做相加操作来实现的而不是简单进行元素相加，让网络自己去学习如何进行特征融合。通过绿色虚线路径的特征只需要数层卷积层就能达到 N5 处，这样两个部分 (top-down 和 down-top) 就很好地解决了利用顶层特征图进行检测时缺少细节和纹理的问题。

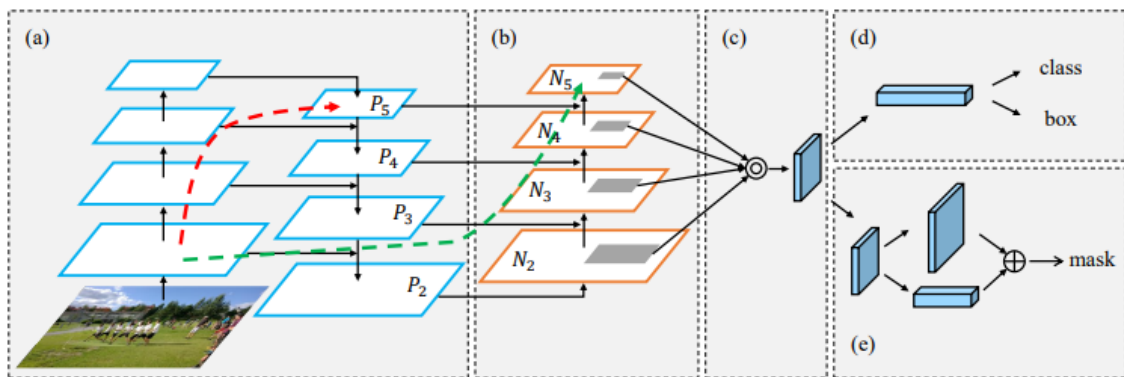


Figure 4: PAFPN 网络结构

ASFF 的网络架构如图 5所示。与 PAFPN 相比，ASFF 网络结构更加复杂，引入了注意力机制对不同特征进行融合。其主要思想是对原来的特征拼接的相加操作引入了权重因子参数，在梯度反向传播的过程中会自动更新这些权重因子，从而使网络输出的特征是最有利于正确进行目标检测的特征。在本算法中，会产生四个 ASFF 块，分别对应输出特征图中的 Layer1、Layer2、Layer3、Layer4，不同的 ASFF 块会根据权重向量融合出不同的特征通道。

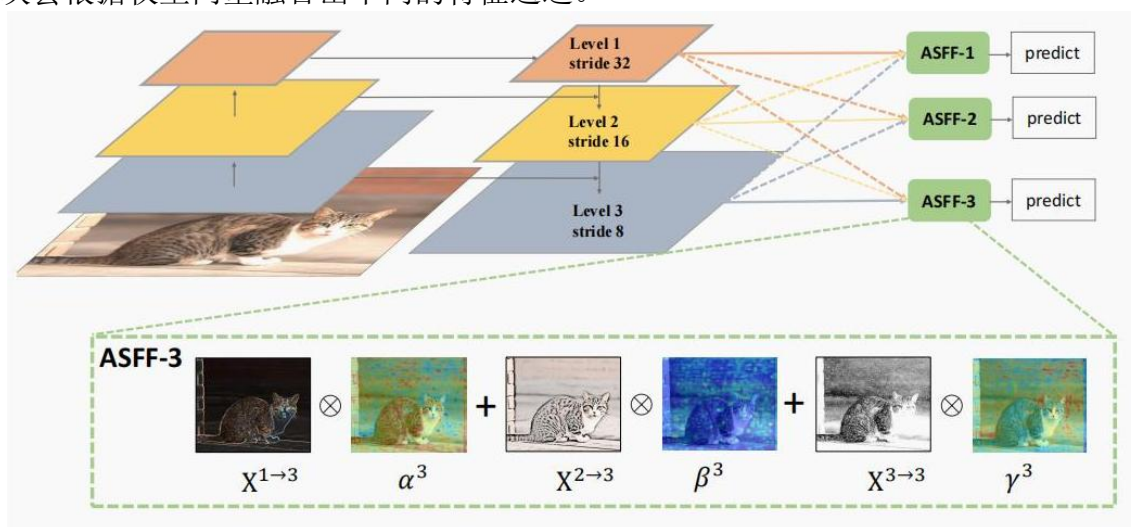


Figure 5: ASFF 网络结构

4.3.4 训练策略

原论文中没有使用任何的训练技巧，在复现论文进行训练的时候我采取了一些训练技巧来使模型更快地收敛，同样达到更好的效果。具体的训练策略包括：权重衰减、stepLR 学习率调整策略、SWA 权重调整策略，以及为了达到更高的 mAP 指标，对模型超参数进行优化与验证。

权重衰减策略是通过限制权重参数的大小从而减少过拟合，使模型可以更好地泛化到新的训练数据。StepLR 学习率调整策略是使学习率每间隔一定的 epoch 将学习率乘以 $\gamma (0 < \gamma < 1)$ ，因此学习率会随着训练时间的推移而衰减，从而避免了由于学习率过大而导致的模型训练不稳定的问题，同时也加快了收敛速度。

SWA 策略的基本原理如图 6所示，是在训练期间，除了每次训练迭代时使用的最新参数之外，还会记录训练期间的每次迭代的参数，并将其累加，然后将累加的参数除以迭代次数，从而获得最终的模型参数。SWA 用于提高模型的收敛性和最终性能，使用 SWA 策略最终网络模型的权重不仅仅是使用最后一个权重，而是使用若干权重的平均值作为最终的模型参数。这样做的好处是，可以减少模型

的方差，增强模型的稳定性，即使是在训练过程中可能出现的极端梯度下降情况也不会导致模型过拟合，有效地降低了泛化误差。同时，SWA 还可以改善模型的性能，降低训练时间，并减少模型参数的存储和传输。

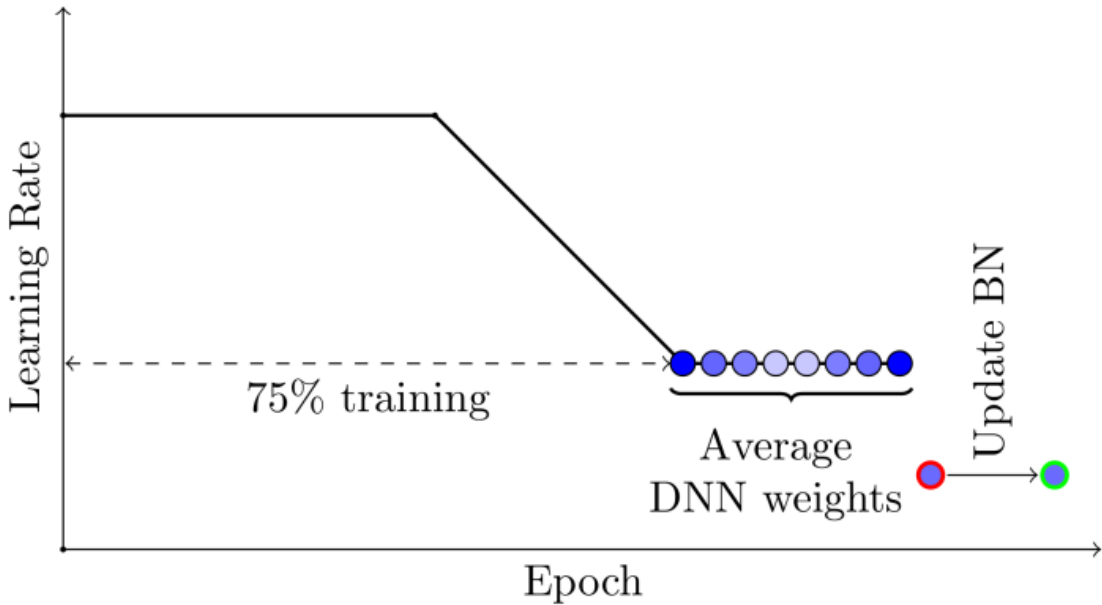


Figure 6: SWA 训练策略原理图

5 实验结果分析

为了验证对模型改进的可行性，我进行了一些对比试验，实验所参照的数据集为 Dota1.0，训练超参数遵循论文中的超参数设定，但是由于机器性能原因，批处理大小设为 4。

序号	Preprocessed	BackBone	Neck	param	mAP(%)
1	ms=[1.0]	ResNet50	FPN	ar=[0.5,1,2] bc=False	76.31
2	ms=[0.5, 1.0, 1.5]	ResNet50	FPN	ar=[0.5,1,2] bc=False	81.13
3	ms=[0.5, 1.0, 1.5]	ResNet50	FPN	ar=[0.25,0.5,1,2,4] bc=True	82.14
4	ms=[0.5, 1.0, 1.5]	ResNet50	PAFPN	ar=[0.25,0.5,1,2,4] bc=True	85.99
5	ms=[0.5, 1.0, 1.5]	res2net50_26w_6s	PAFPN	ar=[0.25,0.5,1,2,4] bc=True SWA	87.17
6	ms=[0.5, 1.0, 1.5]	res2net50_26w_6s	PAFPN	ar=[0.25,0.5,1,2,4] bc=True SWA	88.63
7	ms=[0.5, 1.0, 1.5]	res2net50_26w_6s	ASFF	ar=[0.25,0.5,1,2,4] bc=True SWA	87.63

Table 1: 实验结果（ms: 多尺度 ar: 锚框长宽比 bc: 样本均衡）

如表中内容所示，我首先对论文中的实验进行复现，在没有使用任何技巧的情况下，使用 ResNet50 作为特征提取网络，对单一尺度的数据集进行训练最终网络的 mAP 为 76.31%，使用多尺度数据集进行训练其余条件相同的情况下网络的 mAP 为 81.13%。这个结果和原论文中的结果存在一定的误差，但误差范围在 0.5% 内。这也证明了数据集预处理过程中多尺度对模型训练的影响比较大，能稳定提

升 5% 左右。

之后在论文配置的基础上进行改进，选取序号 2 的结果 81.13% 为实验基准，相续进行了一些对比实验。进行样本均衡后模型 mAP 提高到 82.14%，从对数据的分析来看，数据集中确实存在样本分布不均衡的现象，复现中采用的样本均衡策略是使用硬均衡，即固定调整训练样本中每一类的数量。在此基础上，对特征融合网络进行更改，对特征提取网络不变，使用 PAFPN 作为特征融合网络时，模型 mAP 提高到 85.99%，这对于网络本身的性能来说是一个不小的提升，说明特征融合 PAFPN 与 FPN 相比，确实能够解决对纹理和细节识别不到的问题，进而提升网络模型结果。与此作为对比，将特征提取网络更换为 Res2Net 网络，采用更细粒度的特征提取网络，实验结果表明模型 mAP 进一步提升，达到了 87.17%，也证明了更细粒度的特征提取网络能够同时提取细节特征和全局特征，将网络模型性能进一步拔高。最后将实验 5 的模型权重采用 SWA 模型优化策略，模型的 mAP 提升了接近 1.5%，这是一个不小的性能提升，实验中我是进行了 12epoch 的模型训练，选取最后的 4epoch 的权重参数进行平均作为最终的模型参数。最后，与实验 6 做对比试验，将特征融合网络更换为 ASFF，其余不变的情况下，网络模型 mAP 下降 1% 左右，但总体相差不大，并没有显著降低网络模型性能，由于 ASFF 特征融合网络更加复杂，对 ASFF 中的模型权重的学习会更加缓慢，而且注意力机制的引入需要学习不同 ASFF 层的权重向量，会使模型收敛速度降低。

6 总结与展望

论文中提出了一种实用的双阶段检测器，即 Oriented R-CNN，用于图像中的旋转目标检测。在两个具有挑战性的旋转目标检测数据集上进行了广泛的实验。实验结果表明，该方法与现有先进的双阶段模型相比具有相当的精度，同时保持了与单级模型相当的效率。

本次复现在原有论文模型的基础上更改了网络的特征提取和特征融合网络，并通过对 Dota1.0 数据集的分析，对数据集进行预处理，提升模型对小目标物体的识别精度，最终使模型的 mAP 指数能够提高 7.5%。所以，特征提取和特征融合网络的创新很大程度上决定了网络模型的上限。在将来对网络模型进一步创新的时候可以考虑对提高图像特征的提取质量来提升模型精度。

参考文献

- [1] Zhaowei Cai and Nuno Vasconcelos. “Cascade r-cnn: Delving into high quality object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6154–6162.
- [2] Jian Ding et al. “Learning RoI transformer for oriented object detection in aerial images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2849–2858.
- [3] Jiaming Han et al. “Align deep features for oriented object detection”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–11.
- [4] Chengzheng Li et al. “Feature-attentioned object detection in remote sensing imagery”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 3886–3890.
- [5] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

- [6] Zikun Liu et al. “Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds”. In: *IEEE Geoscience and Remote Sensing Letters* 13.8 (2016), pp. 1074–1078.
- [7] J. Ma and B. Chen. “Dual Refinement Feature Pyramid Networks for Object Detection”. In: (2020).
- [8] Jianqi Ma et al. “Arbitrary-oriented scene text detection via rotation proposals”. In: *IEEE Transactions on Multimedia* 20.11 (2018), pp. 3111–3122.
- [9] Qi Ming et al. “Dynamic anchor learning for arbitrary-oriented object detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 3. 2021, pp. 2355–2363.
- [10] Xingjia Pan et al. “Dynamic refinement network for oriented and densely packed object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11207–11216.
- [11] Zhi Tian et al. “Fcos: Fully convolutional one-stage object detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9627–9636.
- [12] Gui-Song Xia et al. “DOTA: A large-scale dataset for object detection in aerial images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3974–3983.
- [13] Yongchao Xu et al. “Gliding vertex on the horizontal bounding box for multi-oriented object detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.4 (2020), pp. 1452–1459.
- [14] X Yang et al. “Towards more robust detection for small, cluttered and rotated objects”. In: *Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea*. Vol. 27. 2019, pp. 8232–8241.
- [15] Xue Yang et al. “R3det: Refined single-stage detector with feature refinement for rotating object”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 4. 2021, pp. 3163–3171.
- [16] Xue Yang et al. “Scrddet: Towards more robust detection for small, cluttered and rotated objects”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 8232–8241.
- [17] Ze Yang et al. “Reppoints: Point set representation for object detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9657–9666.