目标检测定位蒸馏

郑兆晖, 叶荣光, 侯淇彬, 任冬伟, 王萍, 左旺孟, 程明明

摘要—先前针对目标检测的知识蒸馏（KD）方法主要侧重于特征模仿，而非模仿预测的logits，因为在蒸馏定位信息方面存在效率低下问题。在本文中，我们研究了logits模仿是否总是落后于特征模仿。为了实现这一目标，我们首先提出了一种新颖的定位蒸馏（LD）方法，可以有效地将教师的定位知识转移到学生模型中。其次，我们引入了有价值的定位区域的概念，可以有选择性地蒸馏特定区域的分类和定位知识。通过结合这两个新组件，我们首次表明logits模仿可以胜过特征模仿，并且缺乏定位蒸馏是logits模仿多年来表现不佳的一个关键原因。深入研究展示了logits模仿的巨大潜力，可以显著减轻定位的模糊性，学习鲁棒的特征表示，并在训练的早期阶段减轻训练困难。我们还提供了所提出的LD方法与分类KD之间的理论联系，它们具有相同的优化效果。我们的蒸馏方案简单而有效，并且可以轻松应用于密集水平物体检测器和旋转物体检测器。在MS COCO、PASCAL VOC和DOTA基准测试中进行的大量实验证明，我们的方法可以在不降低推断速度的情况下实现显著的平均精度（AP）提升。我们的源代码和预训练模型已公开发布在 https://github.com/HikariTJU/LD.

Index Terms—目标检测, 定位蒸馏, 知识蒸馏, 旋转目标检测.

1 介绍

作为一种模型压缩技术，知识蒸馏（KD） [1], [2]已成为一种有效的方法，用于学习紧凑的模型以减轻计算负担。通过将大型教师网络捕获的泛化知识传递给小型学生网络，知识蒸馏对于提升小型学生网络性能的有效性已得到广泛验证 [1], [2], [3], [4], [5], [6]。在目标检测中，关于知识蒸馏主要有三种流行的蒸馏流程，如Fig. 1所示。首先是logits模仿 [1]，也被称为分类蒸馏，最初是针对图像分类而设计的，其中蒸馏过程针对师生对的logits进行操作。其次是特征模仿，受先驱工作FitNet [2]的启发，旨在强制师生对之间的特征表示一致。最后，伪边界框回归使用来自教师的预测边界框作为对学生的边界框预测分支的额外监督。

在这些方法中，最初的logits模仿技术 [1]用于分类通常效率低下，因为它只传递分类知识，而忽视了定位知识蒸馏的重要性。因此，目标检测中现有的知识蒸馏方法主要侧重于特征模仿，并且表明蒸馏特征表示比蒸馏logits更有优势 [9], [10], [11]。我们总结了这种现象的三个关键原因：首先，logits模仿的有效性在一定程度上取决于类别数量，而不同的应用场景中类别数量可能不同 [9]。其次，logits模仿只能应用于分类头部，无法蒸馏定位信息。最后，在多任务学习的框架下，特征模仿可以传递分类和定位的混合知识，从而有利于下游的分类和定位任务。

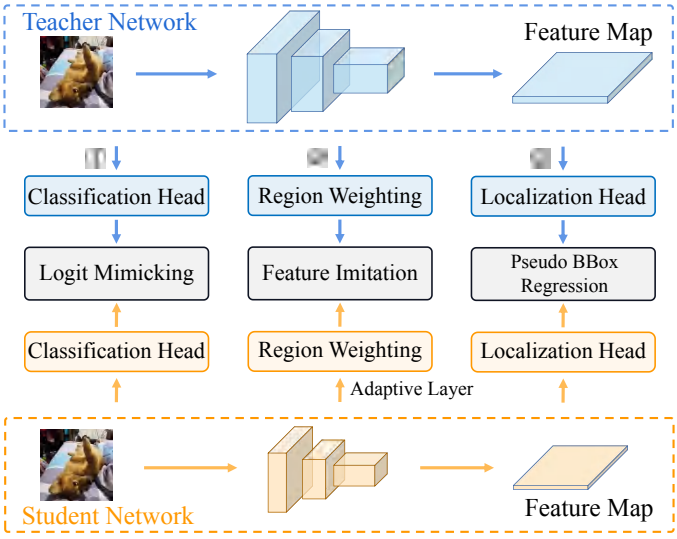


图 1. 目标检测中现有的知识蒸馏框架。① Logit Mimicking（分类蒸馏）：在[1]中提出的分类蒸馏方法。② Feature Imitation（特征模仿）：最近的流行方法，基于不同的蒸馏区域蒸馏中间特征，通常需要自适应层来对齐学生的特征图的尺寸。③ Pseudo BBox Regression（伪边界框回归）：将教师的预测边界框视为额外的回归目标 [7], [8]。

在本工作中，我们考察了目标检测知识蒸馏中先前提出的普遍观点，并检查特征模仿是否总是领先于logits模仿？为此，我们首先提出了一种简单而有效

- *Z. Zheng, Q. Hou, and M.M. Cheng are with TMCC, CS, Nankai University, Tianjin, China. E-mail: Zh_zheng@mail.nankai.edu.cn; {houqb,cmm}@nankai.edu.cn*
- *R. Ye and P. Wang are with the School of Mathematics, Tianjin University, China. E-mail: {ementon,wang_ping}@tju.edu.cn*
- *D. Ren and W. Zuo are with the School of Computer Science and Technology, Harbin Institute of Technology, China. E-mail: {rendongweihit,cswmzuo}@gmail.com*
- *Q. Hou is the corresponding author.*

Manuscript received March 1, 2022; revised August 26, 2022.

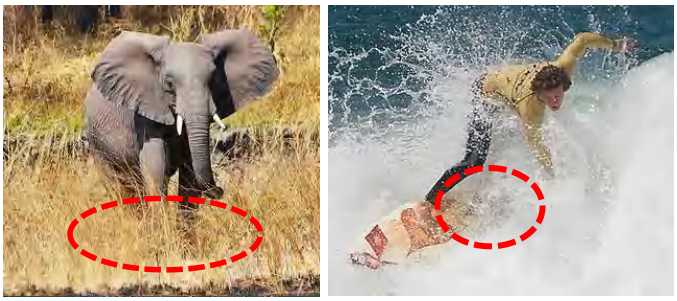


图 2. “大象”的下边缘和“冲浪板”的右边缘，定位是模糊的。

的定位蒸馏（LD）方法，受到一个有趣的观察启发：教师生成的边界框分布 [12], [13]可以作为对学生检测器的强监督。边界框分布 [12], [13]最初设计用于建模真实边界框的分布，是解决定位模糊性的高效方法，如Fig. 2所示。通过离散化的概率分布表示，定位器可以通过分布的平坦程度和锐度来反映定位的模糊性，而这在传统的边界框的狄拉克 δ 函数表示 [14], [15], [16], [17]中是不具备的。这使得我们的定位蒸馏方法能够从教师到学生更有效地传递更丰富的定位知识，而不是仅使用伪边界框回归（Fig. 1中的右部分）。

将提出的定位蒸馏（LD）方法与分类蒸馏相结合，形成了一种统一的知识蒸馏方法，基于纯logits模仿的框架，适用于分类分支和定位分支。由于logits模仿使我们能够分别蒸馏分类知识和定位知识，我们发现这两个子任务对不同的蒸馏区域有不同的偏好。受这一观察启发，我们引入了有价值的定位区域（VLR）的概念，并提出以选择性区域蒸馏的方式进行蒸馏。在实验部分，我们将展示在我们的蒸馏框架中使用VLR的优势。

此外，我们全面讨论了LD的技术细节，并详细阐述了logits模仿和特征模仿的行为。有趣的是，我们观察到logits模仿首次能够胜过特征模仿，这表明定位蒸馏的缺失实际上是logits模仿在目标检测中多年表现不佳的关键原因。另一个观察是，我们发现logits模仿有效的原因不是因为师生对之间特征表示的一致性。相反，从l_n距离和线性相关性的角度来看，学生的特征表示与教师的特征表示存在显著差异。我们还观察到，如果使用特征模仿训练学生模型，它倾向于在特征子空间中产生一个尖锐的AP得分分布，并加剧了早期训练阶段的训练难度。

上述观察反映了logits模仿相对于特征模仿的巨大潜力： 1）能够分别传递不同类型的知识； 2）学习更加鲁棒的特征表示； 3）减轻训练难度。我们的方法简单且易于实施，可以轻松应用于水平和旋转物体检测器中，以提高它们的性能，而无需引入任何推断开销。在MS COCO上进行的大量实验证明，我们的方法无需繁琐的设计，仅使用ResNet-50-FPN骨干网络，在强基线模型GFocal [12]的基础上，将AP得分从40.1提升到42.1，将AP_{75}得分从43.1提升到45.6。我们使用ResNeXt-101-32x4d-DCN骨干网络的最佳模型可以实现50.5 AP的单尺度测试，超越了在相同骨干、neck和测试设置下的所有现有检测器。我们的源代码和预训练模型的PyTorch [18]和Jittor [19]版本已公开在https://github.com/HikariTJU/LD上。

本文的主要贡献包括以下四个方面：

1) 我们提出了一种新颖的定位蒸馏方法，极大地提高了目标检测中logits模仿的蒸馏效率。
2) 我们对logits模仿和特征模仿的行为进行了探索性实验和分析。据我们所知，这是首次揭示logits模仿相对于特征模仿的巨大潜力的工作。
3) 我们提出了基于新引入的有价值的定位区域的选择性区域蒸馏方法，以更好地蒸馏学生检测器。
4) 我们将定位蒸馏扩展到旋转版本，使其可以应用于任意方向的目标检测。

本文是其先前会议版本 [20]的实质性扩展。具体而言，本文的扩展包括： (a) 我们为提出的定位蒸馏方法和分类蒸馏提供了理论上的联系，揭示了它们共享等效的优化效果。通过线性插值两个蒸馏效果，可以获得定位蒸馏的优化效果。 (b) 我们对logits模仿和特征模仿进行了更详细、更有洞察力的分析，包括学习到的特征表示和logits的不同特征、特征模仿的训练困难性。 (c) 我们将原始的定位蒸馏方法扩展为更通用的版本，即旋转定位蒸馏，可以对任意方向的目标检测器进行蒸馏。

2 相关工作

2.1 知识蒸馏

知识蒸馏 [1], [21], [22], [23], [24], [25]作为一个热门的研究课题，近年来得到了深入的研究。其基本思想是利用表现良好的大型教师网络将其所捕捉到的知识传递给小型的学生网络。 logits模仿，即分类蒸馏，最早由Hinton等人 [1]引入，其中学生分类器的logits输出受到教师分类器logits的监督。后来，FitNet [2]通过模仿教师模型的隐藏层中的中间提示，扩展了师生学习框架。知识蒸馏首次应用于目标检测是在 [7]中，其中hint learning、分类蒸馏和伪边界框回归同时用于多类别目标检测。然而，目标检测器不仅需要精确的分类能力，

还需要强大的定位能力。传统的知识蒸馏方法缺乏定位知识蒸馏，限制了其性能。

为了解决上述问题，已经开发了许多特征模仿方法，其中大部分方法关注的是在哪里进行蒸馏和损失函数的加权。其中，Li等人 [26]提出了在Faster R-CNN中模仿候选区域内的特征。 Wang等人 [9]模仿了接近锚框位置的细粒度特征。最近，Dai等人 [27]引入了通用实例选择模块，用于模仿师生对之间具有区别性补丁的深层特征。 DeFeat [28]在对目标区域和背景区域进行特征模仿时利用不同的损失权重。从加权模仿损失的角度来看，还有各种特征模仿方法，包括高斯掩膜加权 [8]、特征丰富度加权 [29]和预测引导的模仿损失 [30]。与前面提到的方法不同，我们的工作引入了定位蒸馏，并证明在目标检测的知识蒸馏中，logits模仿能够胜过特征模仿。

2.2 目标定位

目标定位是目标检测中的一个基本问题 [31], [32], [33], [34], [35], [36], [37], [38], [39], [40]。到目前为止，边界框回归是目标检测中定位的最常用方法 [14], [15], [16], [41], [42]，多年来一直使用狄拉克 δ 函数分布表示。 R-CNN系列 [16], [43], [44], [45]采用多个回归阶段来优化检测结果，而YOLO系列 [14], [46], [47], [48]、SSD系列 [15], [49], [50]和FCOS系列 [12], [17]采用单阶段回归。在 [51], [52], [53], [54]中，提出了基于IoU的损失函数来提高边界框的定位质量。最近，边界框表示从狄拉克 δ 函数分布 [14], [15], [16]发展到高斯分布 [55], [56]，进一步发展到概率分布 [12], [13]。边界框的概率分布对于描述边界框的不确定性更加全面，目前已被证实是最先进的边界框表示方法。

2.3 定位质量估计

如其名称所示，定位质量估计（Localization Quality Estimation，简称LQE）预测一个分数来衡量检测器预测的边界框的定位质量。 LQE通常在训练期间与分类任务协同使用 [57]，即增强分类和定位之间的一致性。它还可以在后处理的联合决策中应用 [14], [17], [58]，在执行NMS时考虑分类分数和LQE。早期的研究可以追溯到YOLOv1 [14]，其中使用预测的对象置信度对分类分数进行惩罚。然后，引入了边界框/掩模IoU [58], [59]和边界框/极坐标中心度 [17], [60]，分别用于建模目标检测和实例分割中的检测不确定性。对于边界框表示，Softer-NMS [55]和Gaussian YOLOv3 [56]预测了边界框各边的方差。 LQE是模拟定位模糊性的一种初步方法。

2.4 任意方向目标检测

在目标检测取得成功的推动下，近年来旋转目标检测已成为计算机视觉中的热门研究课题 [61]。主流的旋转目标检测器，如RRPN [62]，基于Faster R-CNN [16]生成旋转提议，而Rotated-RetinaNet [63]则直接基于RetinaNet预测额外的旋转角度。为了解决边界不连续和类似正方形的问题，SCRDet [37]和RSDet [64]分别提出了IoU平滑L1损失和调制损失，以实现更平滑的边界损失，而CSL [65]提出使用角度分类而不是角度回归。

与可以轻松利用基于IoU的损失函数（如GIoU [52]、 DIoU [53]和CIoU [54]）以增强定位能力的水平边界框回归不同，由于现有深度学习库（如tensorflow、 PyTorch、 Jittor等）中的反向传播的复杂性，用于旋转边界框回归的Skew IoU损失非常难以实现。 PIoU损失 [66]通过累积两个旋转边界框的交集和并集的像素来近似Skew IoU。 GWD [38]和KLD [39]通过2D高斯分布表示建模旋转边界框，并分别提出使用高斯Wasserstein距离和KL散度来模拟Skew IoU损失。最近，基于旋转边界框的2D高斯分布表示，Yang等人 [67]通过利用卡尔曼滤波器的形式化来模仿趋势水平上的Skew IoU，提出了KFIoU损失。总之，基于旋转回归的检测器仍然在这个任务中占据主导地位，因为它们简单且性能强大。

3 方法

首先，我们回顾知识蒸馏的背景，包括logits模仿和特征模仿。接下来，我们描述了我们简单而有效的定位蒸馏（LD）方法，并解释了如何将LD应用于旋转目标检测。然后，我们分析了所提出的LD损失的特性，特别是与分类蒸馏之间的理论连接。此外，我们还介绍了有价值的定位区域的概念，以更好地在我们的框架中蒸馏定位知识。最后，我们描述了基于新引入的有价值定位区域的选择性区域蒸馏方法，并给出了优化目标。

3.1 预备知识

在目标检测的知识蒸馏流程中，输入图像被输入到两个目标检测器中，即学生检测器和冻结的教师检测器。蒸馏过程要求学生的输出模仿教师的输出。目标检测中有两种主流的知识蒸馏方法范式。

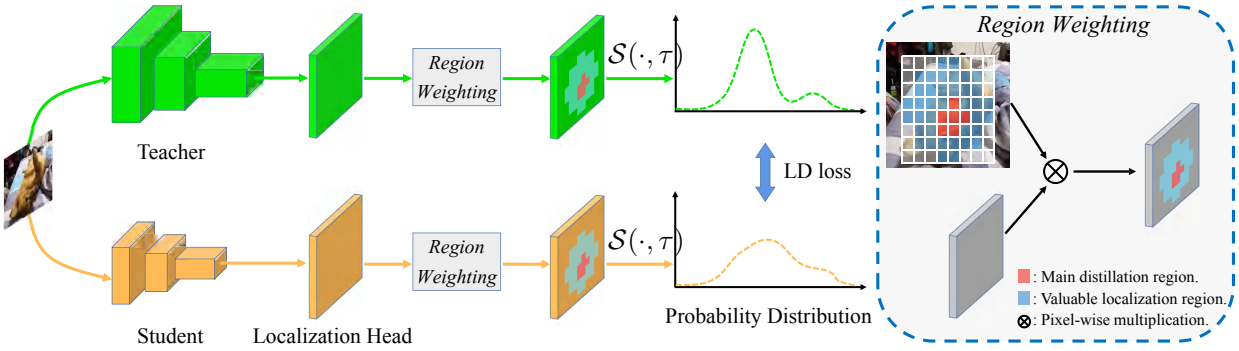


图 3. 对边缘$e \in \mathcal{B}$进行定位蒸馏（LD）的示意图。这里只显示了定位分支。$\mathcal{S}(\cdot,\tau)$是带有温度τ的广义SoftMax函数。对于给定的检测器，我们首先将边界框表示转换为概率分布。然后，通过主要蒸馏区域和有价值的定位区域上的区域加权确定蒸馏位置。最后，计算教师和学生预测的两个概率分布之间的LD损失。

Logits模仿。 Logits模仿（Logit Mimicking，LM）最初用于图像分类 [1]，其中学生模型可以通过模仿教师分类器的软输出来得到改进。令$\boldsymbol{z}_S, \boldsymbol{z}_T \in \mathbb{R}^{W\times H\times C}$为学生和教师预测的logits，其中$W$和$H$表示logit图的输出尺寸，$C$表示类别数量。然后，通过使用广义SoftMax函数将这些logits转换为概率分布$\boldsymbol{p}_\tau$和$\boldsymbol{q}_\tau$。我们可以通过最小化损失来训练网络：

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{KD}} \tag{1}$$

$$= \mathcal{H}(\boldsymbol{p}, \boldsymbol{g}) + \lambda\mathcal{H}(\boldsymbol{p}_\tau, \boldsymbol{q}_\tau), \tag{2}$$

其中，$\boldsymbol{p}$是预测的概率向量，$\boldsymbol{g} = 0, 1^n$是真实标签的one-hot向量，$\mathcal{H}$表示交叉熵损失，λ平衡了两个损失项。对于目标检测，可以在一些预定义的蒸馏区域$\mathcal{R}$上进行蒸馏。

特征模仿。 最近研究发现，特征模仿（Feature Imitation，FI）比分类蒸馏（Classification KD）更有效，特征模仿旨在通过模仿教师和学生之间的深层特征来传递知识 [2], [9]。数学上，特征模仿过程可以表示为：

$$\mathcal{L}_{\text{FI}} = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} ||\tilde{M}_S(r) - M_T(r)||_2, \tag{3}$$

其中，$\mathcal{R}$表示模仿的区域，$|\cdot|$表示区域的基数注意需要使用自适应层将学生的特征图M_S的大小转换为与教师的特征图M_T相同的大小，即$\tilde{M}_S, M_T \in \mathbb{R}^{W\times H\times D}$。

边界框表示。 对于给定的边界框$\mathcal{B}$，常见的表示有两种形式，即 $\delta_x, \delta_y, \delta_w, \delta_h$（编码从锚框到真实框的中心点、宽度和高度的坐标映射） [14], [15], [16] 和t, b, l, r（从采样点到顶部、底部、左侧和右侧边缘的距离） [17]。实际上，这两种形式都遵循了狄拉克 δ 分布，只关注真实位置，无法模拟边界框的不确定性，如Fig. 2所示。这在一些先前的工作中也得到了明确的证明 [12], [55]。

3.2 定位蒸馏

在本小节中，我们介绍定位蒸馏（Localization Distillation，LD），这是一种提高目标检测中蒸馏效率的新方法。我们的LD是从边界框的概率分布表示（基于锚点和无锚点）的观点出发发展而来，这种表示最初设计用于通用目标检测，并包含丰富的定位信息。我们的LD的工作原理如Fig. 3所示。该过程适用于基于锚点和无锚点的检测器。

对于给定的目标检测器，我们按照 [12], [13]的方法将边界框表示从四元表示转换为概率分布。设$e \in \mathcal{B}$为边界框的一个回归变量，其回归范围为$[e_{\min}, e_{\max}]$。边界框分布将连续的回归范围量化为具有n个子区间的均匀离散变量$\boldsymbol{e} = [e_0, e_1, \cdots, e_n] \in \mathbb{R}^{n+1}$，其中$e_0 = e_{\min}$和$e_n = e_{\max}$。定位头部预测了$n+1$个logits，$\boldsymbol{z} = z_0, z_1, \cdots, z_n$，对应于子区间$e_0, e_1, \cdots, e_n$的端点。通过使用SoftMax函数，可以将给定边界框的每个边表示为概率分布。对于子区间的数量n，我们遵循GFocal [12]的设置，推荐选择的n值为$8 \sim 16$。与 [12], [13]不同，我们使用广义SoftMax函数$\mathcal{S}(\cdot,\tau)$将$\boldsymbol{z}S$和$\boldsymbol{z}T$转换为概率分布$\boldsymbol{p}\tau$和$\boldsymbol{q}\tau$。请注意，当$\tau = 1$时，它等价于原始的SoftMax函数。当$\tau \to 0$时，它趋向于狄拉克 δ 分布。当$\tau \to \infty$时，它将成为均匀分布。经验上，设置$\tau > 1$可以使分布变得平滑，使边界框分布携带更多信息。对于边界框表示$\boldsymbol{e}$的两个概率向量$\boldsymbol{p}\tau, \boldsymbol{q}\tau \in \mathbb{R}^n$，我们通过以下方式进行定位蒸馏以衡量它们的相似性：

$$\mathcal{L}_{\text{LD}}^{\boldsymbol{e}} = \mathcal{H}(\boldsymbol{p}_\tau, \boldsymbol{q}_\tau) \tag{4}$$

$$= \mathcal{H}(\mathcal{S}(\boldsymbol{z}_S, \tau), \mathcal{S}(\boldsymbol{z}_T, \tau)). \tag{5}$$

然后，对于边界框$\mathcal{B}$的所有四个边，LD可以被表示为：

$$\mathcal{L}_{\text{LD}}(\mathcal{B}_S, \mathcal{B}_T) = \sum_{\boldsymbol{e}\in\mathcal{B}} \mathcal{L}_{\text{LD}}^{\boldsymbol{e}}, \tag{6}$$

其中，$\mathcal{B}_S, \mathcal{B}_T$分别是学生模型和教师模型预测的边界框。

3.3 旋转LD

我们的LD方法也可以灵活地用于蒸馏旋转边界框检测器。参数回归是经典的基于密集回归的旋转目标检测中最常用的方式 [37], [38], [39], [68]。常用的表示旋转边界框的方法是$\mathcal{B} = \delta_x, \delta_y, \delta_w, \delta_h, \delta_\theta$，其中$\delta_\theta$表示编码的旋转角度。为了进行旋转定位蒸馏，我们首先生成回归范围$[e_{\min}, e_{\max}]$的下限和上限，其中$e \in \mathcal{B}$。

需要注意的是，旋转角度预测δ_θ通常具有与$\delta_x, \delta_y, \delta_w, \delta_h$不同的回归范围。因此，为它们设置不同的回归范围的下限和上限。实际上，$[e_{min}, e_{max}] \subset [-5, 5]$是一个可以接受的选择。

然后，我们将旋转边界框转换为旋转边界框分布，就像Sec. 3.2所描述的那样。最后，根据Eq. (6)计算旋转边界框分布的LD损失。

3.4 LD的特性

我们可以看到，我们的LD保持了标准logit模仿的形式。一个可能会问的问题是：LD是否也继承了分类KD的特性，特别是优化过程中的特性？与分类任务中唯一的整数被视为真实标签不同，定位任务的真实标签是一个浮点数e^*，其值例如在区间$[e_i, e_{i+1}]$内。这意味着在定位蒸馏过程中，我们需要处理连续值的回归目标，而不是离散的类别标签。在接下来，我们展示了LD的一个重要特性，证明它可以继承分类KD所具有的优化效果。

Proposition 1. 设$\boldsymbol{s}$为学生的预测概率向量，u_1和u_2是两个常数，满足$u_1 + u_2 = 1$。我们有：

1) 若 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 是两个分类概率向量，LD对线性组合 $\boldsymbol{l} = u_1\boldsymbol{p} + u_2\boldsymbol{q}$ 的效应等于 KD 对 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 效应的线性组合；
2) 若 $\boldsymbol{l}$ 是一个定位概率向量， LD对 $\boldsymbol{l}$ 的效应等于作用在其分解 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 上的两个 KD的效应的线性组合。

上述两个观点具有相同的表达式，

$$\partial LD_i^{\boldsymbol{l}} = u_1 \partial KD_i^{\boldsymbol{p}} + u_2 \partial KD_i^{\boldsymbol{q}}, \tag{7}$$

其中 $\partial KD_i^{\boldsymbol{p}}$ 表示针对给定的逻辑值 z_i，两个概率 $\boldsymbol{s}, \boldsymbol{p}$ 的KD损失的导数， $\partial LD_i^{\boldsymbol{p}}$ 同样表示针对给定的逻辑值 z_i，概率 $\boldsymbol{p}$ 的LD损失的导数。

Algorithm 1 有价值的定位区域

Require: 一组锚框 $\boldsymbol{B}^a = \mathcal{B}_i^a$ 和一组真实边界框 $\boldsymbol{B}^{gt} = \mathcal{B}^{gt}j$，其中 $1 \leqslant i \leqslant I$，$1 \leqslant j \leqslant J$。标签分配的正样本阈值为 αpos。

Ensure: $\boldsymbol{V} = v_{ij} I \times J$，其中 $vij \in 0, 1$，编码了VLR的最终位置，其中 1 表示VLR，0 表示忽略。

1: 计算DIoU矩阵 $\boldsymbol{X} = \{x_{ij}\}_{I \times J}$，其中 $x_{ij} = DIoU(\mathcal{B}_i^a, \mathcal{B}_j^{gt})$.
2: $\alpha_{vl} = \gamma \alpha_{pos}$.
3: 使用 $\boldsymbol{V} = \{\alpha_{vl} \leqslant \boldsymbol{X} \leqslant \alpha_{pos}\}$计算定位.
4: **返回** $\boldsymbol{V}$

证明可以在补充材料中找到。命题1 提供了LD与分类KD之间的理论联系。它表明，对于一个浮点数定位问题，LD的优化效果在功能上等同于作用在整数位置分类问题上的两个KD效果。因此，作为 [69]的直接推论，对于两个近邻位置上的相对预测置信度，LD对于分布聚焦损失（DFL） [12]保持了梯度重新缩放。具体细节请参阅补充材料。

3.5 有价值的定位区域

之前的研究大多通过最小化l_2损失来使得学生网络的深层特征模仿教师网络的特征。然而，一个直接的问题出现了：我们是否应该毫无区别地使用整个模仿区域来蒸馏混合知识？根据我们的观察，答案是否定的。在本小节中，我们描述了有价值的定位区域（VLR），以进一步提高蒸馏效率。我们相信这将是训练更好的学生检测器的一种有希望的方法。

具体来说，蒸馏区域被分为两部分，主要蒸馏区域和有价值的定位区域。主要蒸馏区域由标签分配直观确定，即检测头的正位置。有价值的定位区域可以通过算法 1 获得。首先，我们计算所有锚框 $\boldsymbol{B}^a$ 与真实边界框 $\boldsymbol{B}^{gt}$ 之间的DIoU [53] 矩阵 $\boldsymbol{X}$。然后，我们将DIoU的下界设定为 $\alpha_{vl} = \gamma \alpha_{pos}$，其中 α_{pos} 是标签分配的正样本IoU阈值。 VLR可以定义为 $\boldsymbol{V} = \alpha_{vl} \leqslant \boldsymbol{X} \leqslant \alpha_{pos}$。我们的方法只有一个超参数 $\gamma \leqslant 1$，它控制着VLR的范围。当 $\gamma = 0$ 时，所有预设锚框与GT边界框之间的DIoU满足 $0 \leqslant x_{ij} \leqslant \alpha_{pos}$ 的位置将被确定为VLR。当 $\gamma \to 1$ 时，有价值的定位区域（VLR）将逐渐收缩为空。我们选择使用DIoU [53]，因为它对于靠近物体中心的位置赋予更高的优先级。

类似于标签分配，我们的方法在多层FPN上为每个位置分配属性。通过这种方式，一些位于真实边界框之外的位置也会被考虑进来。因此，我们实际上可以将VLR视为主要蒸馏区域向外的扩展。请注意，对于无

锚检测器（如FCOS），我们可以在特征图上使用预设的锚点，同时不改变其回归形式，以便保持定位学习的无锚类型。而对于通常在每个位置设置多个锚框的锚点检测器，我们会展开这些锚框以计算DIoU矩阵，并为它们分配属性。

3.6 选择性区域蒸馏

根据以上描述，用于训练学生网络 $\boldsymbol{S}$ 的logit模仿总损失可以表示为：

$$\begin{aligned}\mathcal{L} =&\lambda_0\mathcal{L}_{\text{cls}}(\mathcal{C}_S,\mathcal{C}^{gt})+\lambda_1\mathcal{L}_{\text{reg}}(\mathcal{B}_S,\mathcal{B}^{gt})+\lambda_2\mathcal{L}_{\text{DFL}}(\mathcal{B}_S,\mathcal{B}^{gt})\\&+\lambda_3\mathbb{I}_{\text{Main}}\mathcal{L}_{\text{LD}}(\mathcal{B}_S,\mathcal{B}_T)+\lambda_4\mathbb{I}_{\text{VL}}\mathcal{L}_{\text{LD}}(\mathcal{B}_S,\mathcal{B}_T)\\&+\lambda_5\mathbb{I}_{\text{Main}}\mathcal{L}_{\text{KD}}(\mathcal{C}_S,\mathcal{C}_T)+\lambda_6\mathbb{I}_{\text{VL}}\mathcal{L}_{\text{KD}}(\mathcal{C}_S,\mathcal{C}_T),\end{aligned} \tag{8}$$

其中前三项与基于回归的检测器的分类和边界框回归分支完全相同，即 $\mathcal{L}_{\text{cls}}$ 是分类损失，$\mathcal{L}_{\text{reg}}$ 是边界框回归损失，$\mathcal{L}_{\text{DFL}}$ 是分布聚焦损失 [12]。 $\mathbb{I}$Main 和 $\mathbb{I}$VL 分别是主要蒸馏区域和有价值的定位区域的蒸馏掩码。 $\mathcal{L}_{\text{KD}}$ 是KD损失 [1]， $\mathcal{C}_S$ 和 $\mathcal{C}_T$ 分别表示学生网络和教师网络的分类头输出逻辑值， $\mathcal{C}^{gt}$ 是真实类别标签。

所有蒸馏损失将根据它们的类型被赋予相同的权重因子。更明确地说，LD损失的权重因子与边界框回归项的权重因子相同，知识KD损失的权重因子与分类项的权重因子相同。值得一提的是，由于LD损失具有足够的指导能力，DFL损失项可以禁用。另外，我们可以选择启用或禁用四种类型的蒸馏损失，以便有选择地在不同区域对学生进行蒸馏。

4 实验

在本节中，我们进行了全面的消融实验和分析，以展示所提出的LD和蒸馏方案在具有挑战性的大规模MS COCO [70]基准、PASCAL VOC [71]和航拍影像DOTA数据集 [72]上的优越性。

4.1 实验设置

MS COCO. 我们使用 train2017（118K张图像）进行训练，val2017（5K张图像）用于验证。我们还通过提交到COCO服务器，在MS COCO test-dev 2019数据集（20K张图像）上进行了评估。实验是在mmDetection [73]框架下进行的。除非另有说明，我们使用ResNet [74] 作为骨干网络，并结合FPN [75] 作为颈部网络，使用FCOS风格的无锚头用于分类和定位。消融实验的训练计划设置为单尺度 1× 模式（12个epochs）。对于其他训练和测试超参数，我们完全遵循GFocal [12]的协议，包括分类任务使用QFL损失，边界框回归任务使用GIoU损失等。我们使用标准的COCO式评估方式，即平均精度（AP）。所有的基准模型都采用相同的设置重新训练，以便与我们的LD进行公平比较。

PASCAL VOC. 我们还提供了在另一个流行的目标检测基准测试上的实验结果，即PASCAL VOC [71]。我们使用VOC 07+12训练协议，即将VOC 2007的trainval集和VOC 2012的trainval集（16551张图像）联合起来进行训练，然后使用VOC 2007的测试集（4952张图像）进行评估。初始学习率为0.01，总训练epochs设置为4。在第3个epoch之后，学习率会减小10倍。为了全面评估定位性能，我们报告了平均精度（AP）以及5个不同IoU阈值下的mAP，即AP50、AP60、AP70、AP80和AP_{90}。

DOTA. 对于旋转LD的评估，我们在经典的航拍图像数据集DOTA [72]上报告了检测结果。我们遵循标准的mmRotate [61]的训练和测试协议。训练集和验证集分别包含1403张和468张图像，在文献中这些图像是随机选择的。这些巨大的图像被裁剪成形状为600 × 600的小子图像，这与官方实现中的裁剪协议保持一致。在实践中，我们获得了大约15,700个训练patches和5,300个验证patches。除非另有说明，所有超参数都遵循mmRotate的默认设置，以进行公平比较。我们以AP和5个不同IoU阈值下的mAP为指标进行结果报告，这与PASCAL VOC保持一致。由于内存限制，教师网络使用ResNet-34 FPN，并进行2×的训练计划（24个epochs），而学生网络使用ResNet-18 FPN，并进行1×的训练计划（12个epochs）。

4.2 消融实验

LD的温度参数 τ. 我们的LD引入了一个超参数，即温度 τ。表1(a) 报告了使用不同温度的LD的结果，其中教师模型是具有 AP 44.7 的 ResNet-101，学生模型是ResNet-50。在这里，只采用了主要蒸馏区域。与Tab. 1(a)中的第一行相比，不同的温度一致地导致更好的结果。在本文中，我们简单地将LD中的温度设置为 $\tau = 10$，并在所有其他实验中固定使用该值。

LD *vs.* 伪 BBox 回归. 教师边界框回归（TBR）损失 [7]是增强学生网络的定位头的初步尝试，即Fig. 1中的

表 1
消融. 我们在MS COCO val2017数据集上展示LD和有价值的定位区域VLR的消融实验结果。

τ	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
–	40.1	58.2	43.1	23.3	44.4	52.5
1	40.3	58.2	43.4	22.4	44.0	52.4
5	40.9	58.2	44.3	23.2	**45.0**	53.2
10	**41.1**	**58.7**	**44.9**	**23.8**	44.9	**53.6**
15	40.7	58.5	44.2	23.5	44.3	53.3
20	40.5	58.3	43.7	**23.8**	44.1	53.5

(a) **LD中的温度参数 τ**：使用较大的 τ 值的广义Softmax函数带来了显著的收益。我们默认将 τ 设置为10。教师网络为ResNet-101，学生网络为ResNet-50。

ε	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
–	40.1	58.2	43.1	23.3	44.4	52.5
0.1	40.5	58.3	43.8	23.0	44.2	52.7
0.2	40.2	58.2	43.6	23.1	44.0	53.0
0.3	40.1	58.4	43.1	23.6	43.9	52.5
0.4	40.3	58.4	43.4	22.8	44.0	52.6
LD	**41.1**	**58.7**	**44.9**	**23.8**	**44.9**	**53.6**

(b) **LD *vs.* 伪边界框回归 [7]**: 相比于伪边界框回归，我们的LD能够更有效地传递定位知识。教师网络为ResNet-101，学生网络为ResNet-50。

γ	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
–	40.1	58.2	43.1	23.3	44.4	52.5
1	41.1	58.7	44.9	23.8	44.9	53.6
0.75	41.2	58.8	44.9	23.6	45.4	53.5
0.5	41.7	59.4	45.3	24.2	45.6	54.2
0.25	**41.8**	**59.5**	**45.4**	24.2	45.8	**54.9**
0	41.7	**59.5**	**45.4**	**24.5**	**45.9**	54.0

(c) **γ在VLR中的作用**。：在有价值的定位区域上进行LD对性能有积极的影响。我们默认将 γ 设置为0.25。教师网络为ResNet-101，学生网络为ResNet-50。

伪边界框回归。 TBR损失可以表示为：

$$\mathcal{L}_{\text{TBR}} = \lambda \mathcal{L}_{\text{reg}}(\mathcal{B}^s, \mathcal{B}^{gt}), \text{if } \ell_2(\mathcal{B}^s, \mathcal{B}^{gt}) + \varepsilon > \ell_2(\mathcal{B}^t, \mathcal{B}^{gt}), \tag{9}$$

其中，$\mathcal{B}^s$ 和 $\mathcal{B}^t$ 分别表示学生和教师的预测边界框，$\mathcal{B}^{gt}$ 表示真实边界框， ε 是预定义的边界， $\mathcal{L}_{\text{reg}}$ 表示GIoU损失 [52]。在这里，只采用了主要蒸馏区域。从Tab. 1(b)中，我们可以看到当在Eq. (9)中使用适当的阈值$\varepsilon = 0.1$时，TBR损失确实产生了性能增益（+0.4 AP和+0.7 AP_{75}）然而，TBR损失使用了粗糙的边界框表示，其中不包含检测器的任何定位不确定性信息，从而导致次优的结果。相反，我们的LD直接获得了41.1的AP和44.9的AP_{75}，因为它利用了包含丰富定位知识的边界框概率分布。

VLR中的不同γ. 新引入的VLR具有参数γ，它控制VLR的范围。如Tab. 1(c)所示，当γ的取值范围从0到0.5时，AP保持稳定。在这个范围内，AP的变化大约在0.1左右。随着γ的增加，VLR逐渐收缩为空。性能也逐渐下降到41.1，即仅在主要蒸馏区域上进行LD。对参数γ的敏感性分析实验证明，在VLR上进行LD对性能有积极的影响。在其余的实验中，为了简单起见，我们将γ设置为0.25。

选择性区域蒸馏. 关于KD和LD的作用以及它们的优选区域，有几个有趣的观察结果。我们在Tab. 2中报告了相关的消融实验结果，其中“Main”表示在主要蒸馏区域上进行logit模仿，即标签分配的正样本位置，“VLR”表示有价值的定位区域。对于MS COCO数据集，我们可以看到进行“Main LD”、“VLR LD”和“Main KD”都有助于学生网络的性能提升。这表明主要蒸馏区域包含了有价值的分类和定位知识，而分类KD相比LD的效果较差。然后，我们将分类KD扩展到更大的范围，即VLR。然而，我们观察到进一步将“VLR KD”引入并没有带来任何改进（Tab. 2的最后两行）。这就是为什么我们采用了所提出的选择性区域蒸馏方法来处理COCO数据集的主要原因。

表 2
对于KD和我们的LD的**选择性区域蒸馏**评估。COCO数据集的师生对使用ResNet-101→ResNet-50，VOC 07+12数据集的师生对是ResNet-101→ResNet-18。

LD		KD		MS COCO val2017			VOC 07+12		
Main	VLR	Main	VLR	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
				40.1	58.2	43.1	51.8	75.8	56.3
✓				41.1	58.7	44.9	53.0	75.9	57.6
✓	✓			41.8	59.5	45.4	53.4	76.3	**58.3**
✓	✓	✓		**42.1**	**60.3**	**45.6**	53.1	76.8	57.6
✓	✓	✓	✓	42.0	60.0	45.4	**53.7**	**77.3**	58.2

接下来，我们检查了在PASCAL VOC数据集上的KD和LD的作用。从Tab. 2中可以看出，将定位知识转移给主要蒸馏区域和VLR都是有益的。

然而，由于不同的知识分布模式，分类知识蒸馏也显示出类似的性能下降。通过比较Tab. 2中的第3行和第4行，“Main KD”导致了性能下降，而“VLR KD”对学生网络产生了积极影响。这表明选择性区域蒸馏可以在它们各自有利的区域充分发挥KD和LD的优势。

轻量级检测器的LD结果. Tab. 3 报告了我们的蒸馏方案(COCO上的“Main LD + VLR LD + Main KD”)，其中对一系列轻量级学生网络进行了蒸馏，包括ResNet-18、ResNet-34和ResNet-50。对于所有给定的学生网络，我们的LD都可以稳定地提高检测性能，而无需任何复杂的操作。从这些结果中，我们可以看到我们的LD分别将ResNet-18、ResNet-34和ResNet-50的AP提高了+1.7、+2.1和+2.0，将AP_{75}提高了+2.2、

表 3
轻量级检测器的LD的定量结果。教师模型是ResNet-101。结果报告在MS COCO val2017数据集上。

Student	LD	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-18		35.8	53.1	38.2	18.9	38.9	47.9
	✓	37.5	54.7	40.4	20.2	41.2	49.4
ResNet-34		38.9	56.6	42.2	21.5	42.8	51.4
	✓	41.0	58.6	44.6	23.2	45.0	54.2
ResNet-50		40.1	58.2	43.1	23.3	44.4	52.5
	✓	42.1	60.3	45.6	24.5	46.2	54.8

表 4
LD在各种流行的密集目标检测器上的定量结果。教师模型是ResNet-101，学生模型是ResNet-50。结果报告在MS COCO val2017数据集上。

Student	LD	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet [63]		36.9	54.3	39.8	21.2	40.8	48.4
	✓	39.0	56.4	42.4	23.1	43.2	51.1
FCOS [17]		38.6	57.2	41.5	22.4	42.2	49.8
	✓	40.6	58.4	44.1	24.3	44.1	52.3
ATSS [76]		39.2	57.3	42.4	22.7	43.1	51.5
	✓	41.6	59.3	45.3	25.2	45.2	53.3

+2.4和+2.5。

在其他密集目标检测器上的应用. 我们的LD可以灵活地应用于其他密集目标检测器，包括基于锚框和不基于锚框的类型。我们采用分而治之的蒸馏方案LD应用于几种最近流行的检测器，如RetinaNet [63]（基于锚框）、FCOS [17]（不基于锚框）和ATSS [76]（基于锚框）。根据Tab. 4中的结果，我们可以看到我们的LD可以稳定地将基线模型的AP提高约2分。

任意方向目标检测器. 作为我们LD的直接扩展，旋转边界框需要额外的概率分布，即旋转角度分布。我们对两个任意方向目标检测器进行了必要的最小修改：1）基于密集回归的旋转检测器Rotated-RetinaNet [63]的基础；2）最近流行的2D高斯分布建模检测器GWD [38]。我们遵循mmRotate [61]的训练和测试协议。我们使用ResNet-34作为教师模型，使用ResNet-18作为学生模型以节省GPU内存。结果报告在DOTA-v1.0 [72]的验证集上。

结果已在Tab. 5中展示，表明我们的LD也可以成功应用于旋转目标检测器，并在航空图像检测中取得了显著的提升。特别是，在更严格的IoU阈值下，如AP70、AP80、AP_{90}，我们获得了令人印象深刻的提升。这显示了我们的LD的卓越兼容性，它不仅可以应用于水平边界框，还可以应用于旋转边界框。此外，值得一提

表 5
旋转LD在流行的任意方向目标检测器上的定量结果。教师模型是ResNet-34，学生模型是ResNet-18。结果报告在DOTA-v1.0的验证集上。

Student	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
R-RetinaNet [63]	33.7	58.0	54.5	42.3	22.9	4.7
LD (ours)	39.1	63.8	61.1	48.8	28.7	8.8
GWD [38]	37.1	63.1	60.1	46.7	24.7	6.2
LD (ours)	40.2	66.4	63.6	50.3	28.2	8.5

的是，我们的LD不依赖于边界框的表示方式以及建模的优化方式（水平边界框预测使用IoU-based loss [52], [53]，旋转边界框预测使用2D高斯建模 [38]）。

4.3 Logit模仿 *v.s.* 特征模仿.

到目前为止，我们已经验证了我们的LD和选择性区域蒸馏在蒸馏不同类型的目标检测器中的有效性。我们提出的LD以及分类KD提供了一个统一的对数似然模仿框架。这自然引出了一些有趣的问题：

- 关于检测性能方面，与特征模仿相比，logit模仿表现如何？特征模仿是否始终优于logit模仿？
- 这两种不同的蒸馏技术有哪些特点？深度特征表示和logits学习到的是否不同？

在本小节中，我们将回答上述问题。

数值结果的定量比较. 首先，我们将我们提出的LD与几种最先进的特征模仿方法进行比较。我们采用选择性区域蒸馏，即对主要蒸馏区域进行KD和LD，并对VLR进行LD。由于现代检测器通常配备有FPN [75]，我们遵循先前的工作 [9], [27], [28]，重新实现他们的方法，并将所有特征模仿施加在多层FPN上进行公平比较。在这里，“FitNets” [2]对整个特征图进行蒸馏。“DeFeat” [28]意味着在GT框外的特征模仿损失权重大于GT框内的权重。“Fine-Grained” [9]方法在接近的锚点位置上对深度特征进行蒸馏。“GI Imitation” [27]方法根据学生和教师的鉴别性预测选择蒸馏区域。“Inside GT Box”表示我们选择与特征模仿区域在FPN层上具有相同步幅的与GT框重叠的区域。“Main Region”表示我们在主要蒸馏区域内进行特征模仿。

从Tab. 6中我们可以看出，对整个特征图进行蒸馏获得了+0.6 AP的增益。通过在GT框外部位置设置更大损失权重（DeFeat [28]），性能略优于在所有位置使用相同损失权重的情况。 Fine-Grained [9]关注GT框附近的位置，产生了41.1 AP的结果，与使用Main Region进

表 6
Logit模仿 *vs.* 特征模仿. “Ours” 表示我们使用选择性区域蒸馏，即, “Main LD + VLR LD + Main KD”. “*”表示我们移除了“Main KD”. 教师模型是ResNet-101，学生模型是 ResNet-50 [74]。结果在 MS COCO val2017 上报告.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Baseline (GFocal [12])	40.1	58.2	43.1	23.3	44.4	52.5
FitNets [2]	40.7	58.6	44.0	23.7	44.4	53.2
Inside GT Box	40.7	58.6	44.2	23.1	44.5	53.5
Main Region	41.1	58.7	44.4	24.1	44.6	53.6
Fine-Grained [9]	41.1	58.8	44.8	23.3	45.4	53.1
DeFeat [28]	40.8	58.6	44.2	24.3	44.6	53.7
GI Imitation [27]	41.5	59.6	45.2	24.3	45.7	53.6
Ours	42.1	60.3	45.6	24.5	46.2	54.8
Ours + FitNets	42.1	59.9	45.7	25.0	46.3	54.4
Ours + Inside GT Box	42.2	60.0	45.9	24.3	46.3	55.0
Ours + Main Region	42.1	60.0	45.7	24.6	46.3	54.7
Ours + Fine-Grained	42.4	60.3	45.9	24.7	46.5	55.4
Ours* + Fine-Grained	42.1	59.7	45.6	24.8	46.1	54.8
Ours + DeFeat	42.2	60.0	45.8	24.7	46.1	54.4
Ours + GI Imitation	42.4	60.3	46.2	25.0	46.6	54.5

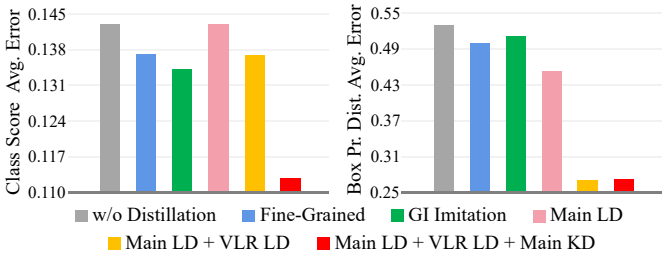


图 4. SOTA特征模仿方法与我们的LD的视觉比较。我们展示了P4、P5、P6和P7 FPN层级上教师和学生之间分类分数和框概率分布的平均L1误差。教师模型是ResNet-101，学生模型是ResNet-50 [74]。结果在MS COCO val2017上进行评估.

行特征模仿的结果相当。 GI模仿 [27]在寻找用于特征模仿的区域时获得了41.5 AP。

由于学生和教师之间预测存在很大差距，模仿区域可能出现在任何地方。

尽管这些特征模仿方法有显著的改进，但它们并没有明确考虑知识分布模式。相反，我们的方法可以通过选择性区域蒸馏传递知识，直接获得42.1 AP的结果。值得注意的是，我们的方法是在logits上操作而不是深度特征，这表明我们的LD是使logit模仿超越特征模仿的关键组成部分。此外，我们的方法与前面提到的特征模仿方法是正交的。 Tab. 6显示，使用这些特征模仿方法，我们的性能可以进一步提高。特别是，在GI模仿的情况下，我们将强大的GFocal基线提高了+2.3 AP和+3.1 AP_{75}。

表 7
师生对之间的平均皮尔逊相关系数。'GI'：GI模仿。'Ours'：我们的具有选择性区域蒸馏的逻辑模仿方案。结果在MS COCO val2017上评估.

	w/o distillation	GI	Ours	Ours + GI
deep features	-0.0042	0.8175	-0.0031	0.8373
bbox logits	0.9222	0.9326	0.9733	0.9745

师生误差比较. 我们首先检查分类得分和框概率分布的平均师生误差，如图4所示。可以看出，Fine-Grained特征模仿 [9]和GI模仿 [27]按预期减少了这两个误差，因为分类知识和定位知识混合在特征图上。我们的“Main LD”和“Main LD + VLR LD”与Fine-Grained [9]和GI模仿 [27]相比，具有可比或更大的分类分数平均误差，但具有较低的框概率分布平均误差。这表明这两种设置仅使用LD可以显著减少教师和学生之间的框概率分布距离，但不能减少分类头部的错误。如果我们在主要蒸馏区域施加分类KD，即得到“Main LD + VLR LD + Main KD”，则分类分数平均误差和框概率分布平均误差都可以减少。

我们还可视化了P5和P6 FPN层级上学生和教师之间的定位头logit的L1误差总和。如Fig. 5所示，与“无蒸馏”相比，我们可以看到GI模仿 [27]确实减少了教师和学生之间的定位差异。需要注意的是，我们特意选择了一个性能略优于GI模仿的模型（“Main LD + VLR LD”）进行可视化。我们的方法明显减少了这个误差并缓解了定位的不确定性。

在图 Fig. 6 中，我们分别绘制了学生和老师之间的平均误差，分别以深度特征、类别logit和bbox logit为基础。可以看出，这三种类型的错误在测试分辨率变化时表现出几乎一致的趋势。有趣的是，我们发现即使逻辑模仿可以缩小bbox logit和分类逻辑的错误，它仍然学习到与老师完全不同的特征表示。从 Fig. 6 的左侧可以看出，我们的方法增加了学生的特征表示与老师之间的距离。此外，Tab. 7 显示，逻辑模仿在老师和学生之间的特征表示之间产生了几乎为零的皮尔逊相关系数。这表明，如果仅通过逻辑模仿对学生进行训练，它会产生与老师的特征表示相距较远且非线性相关的特征表示。虽然如此，我们仍然可以获得良好的逻辑分数以实现良好的泛化效果。 Tab. 7 的最后一列和 Fig. 6 表明，逻辑模仿不仅能在距离上接近老师的逻辑分数，还能在线性相关性上接近。

AP景观. 从特征级别或逻辑级别提取目标检测器是一个高维非凸优化问题，实际上容易但理论上困难。为了

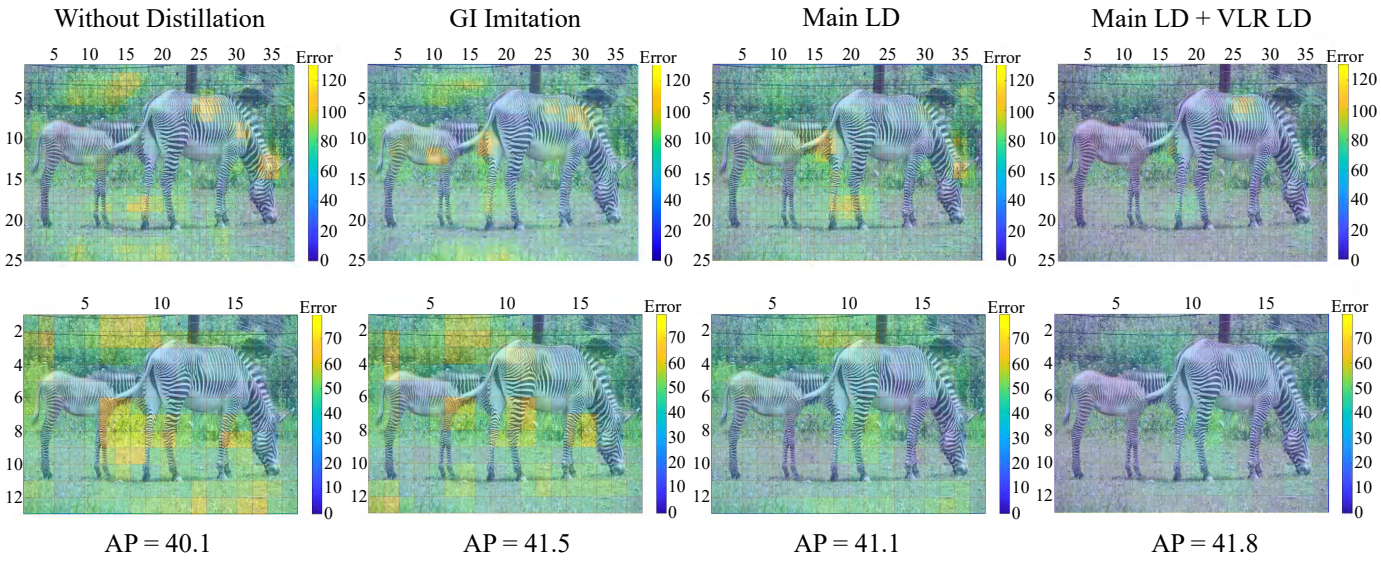


图 5. 我们对最先进的特征模仿方法与我们的LD进行了视觉比较。我们展示了P5（第一行）和P6（第二行）FPN层级上教师和学生之间定位头部logit的每个位置的L1误差总和。教师模型是ResNet-101，学生模型是ResNet-50 [74]。我们可以看到，与GI模仿方法 [27]相比，我们的方法（“Main LD + VLR LD”）可以显著减少几乎所有位置的误差。颜色越深表示效果越好。最好在彩色视图中查看。

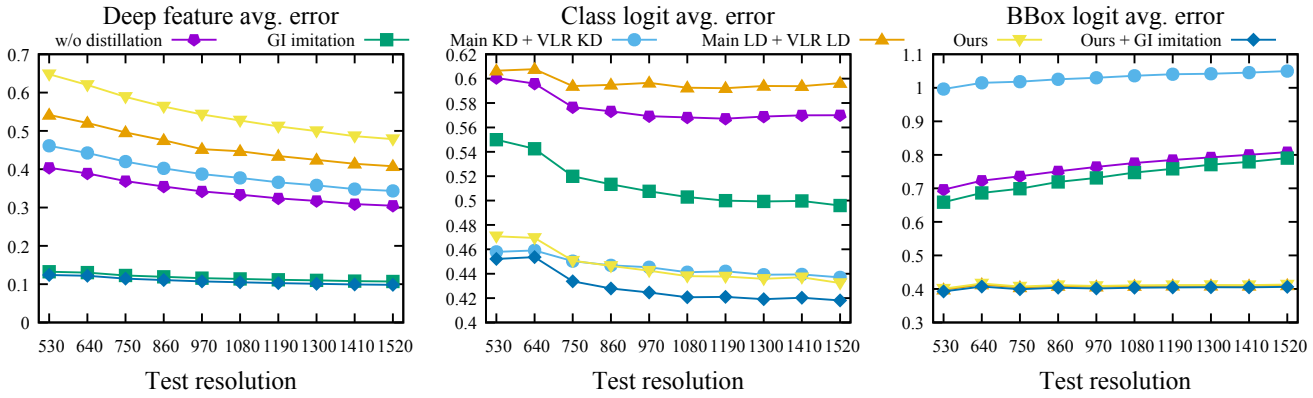


图 6. 在（左侧）深度特征表示，（中间）类别logits和（右侧）边界框logits上的平均师生误差。“Ours”表示“Main LD + VLR LD + Main KD”。曲线是在MS COCO val2017上进行评估.

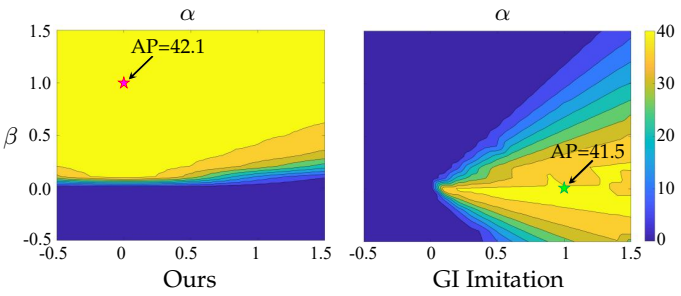


图 7. 二维轮廓图显示了特征子空间中的AP景观。这些AP景观在MS COCO val2017上进行了评估。

更好地理解逻辑模仿和特征模仿的行为，我们提出了一种新的可视化方法，称为AP景观，专门用于目标检测，以观察学习特征表示中微小扰动引起的AP变化。

在 [77]中采用了一种经典的方法，该方法通过线性插值两个网络的参数来研究损失曲面的可视化。

在我们的可视化中，我们特别关注特征表示的经验性特征化以及它们如何影响最终的性能。考虑到两个特征表示 M_f 和 M_l，它们是分别通过使用特征模仿和逻辑模仿训练的检测器学到的，我们在 2D 投影空间 $M_f \oplus M_l$ 内可视化了AP景观。我们使用两个标量参数 α 和 β，通过加权和 $M(\alpha, \beta) = \alpha M_f + \beta M_l$ 来获得一个新的特征表示。请注意，当 $\alpha = 0$ 且 $\beta = 1$ 时，这表示特征表示是由逻辑模仿方法预测的，相反，当 $\alpha = 1$ 且 $\beta = 0$ 时，表示特征模仿。然后，我们将 $M(\alpha, \beta)$ 输入到下游头部，并绘制最终的AP分数。由于计算负担较重，我们将 $\alpha, \beta \in [-0.5, 1.5]$ 以可视化2D AP景观。

从 Fig. 7 中，我们可以看出，logit mimicking 学习了稳健的特征表示，即红色五角星位于 $(0, 1)$ 处，周围是一个平坦且表现良好的 AP 分数区域。其次，我

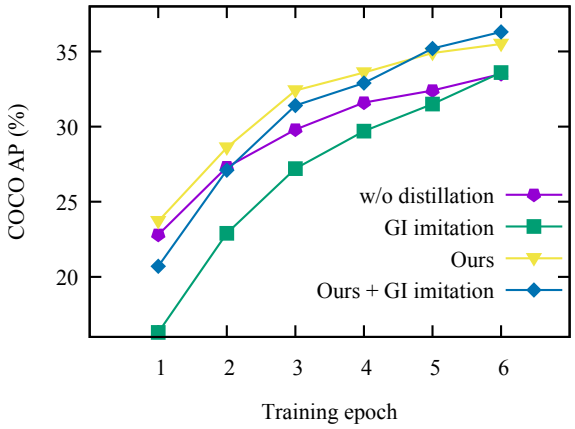


图 8. 早期训练阶段的平均精度（AP）。特征模仿显著减缓了收敛速度并导致了次优的泛化性能。 Logit mimicking（我们的方法）可以减少早期训练阶段的训练难度。

们观察到 GI imitation 产生了比 logit mimicking 更为陡峭的 AP 景观。我们将 GI imitation 的景观陡峭性归因于硬 l_2 损失监督。在这种情况下，对于学生来说，从教师那里模仿高级和先进的特征表示是困难的，这对应于一个训练周期更长、精度更高的重型检测器。相反，logit mimicking 给予了特征表示更多的自由学习空间，从而实现更好的泛化能力。正如 Fig. 8 所示，logit mimicking 还可以减少早期训练阶段的优化难度，而特征模仿在早期训练阶段的收敛速度较慢，泛化性能较差。

总结. 基于以上的结果和观察，我们可以得出以下结论：

- 当明确进行定位知识蒸馏时，Logit mimicking在目标检测中可以优于特征模仿。
- 特征模仿可以增加教师和学生之间特征表示的一致性，但会带来一些缺点，如特征的稳健性较差和训练收敛较慢。通过选择性区域蒸馏的Logit mimicking可以显著提高教师和学生之间的对数一致性，保持特征的学习自由度，从而加速训练过程并更有利于知识蒸馏性能。这表明，改进知识蒸馏性能的关键因素并不是教师和学生之间特征表示的一致性。

4.4 与SOTA方法的比较

我们将我们的LD与最先进的密集目标检测器进行比较，通过在GFocalV2 [57]上应用我们的LD来进一步提升性能。对于COCO val2017数据集，由于大多数先前的方法都使用ResNet-50-FPN骨干网络，采用单尺度1×的训练计划（12个epochs）进行验证，我们也在这个设置

表 8
在 COCO *val2017* 和 *test-dev2019* 上与SOTA方法的比较. **TS**: 训练方案. '1×': 单尺度训练12epochs. '2×': 多尺度训练24epochs.

Method	TS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 backbone on val2017							
RetinaNet [63]	1×	36.9	54.3	39.8	21.2	40.8	48.4
FCOS [17]	1×	38.6	57.2	41.5	22.4	42.2	49.8
SAPD [81]	1×	38.8	58.7	41.3	22.5	42.6	50.8
ATSS [76]	1×	39.2	57.3	42.4	22.7	43.1	51.5
BorderDet [82]	1×	41.4	59.4	44.5	23.6	45.1	54.6
AutoAssign [83]	1×	40.5	59.8	43.9	23.1	44.7	52.9
PAA [84]	1×	40.4	58.4	43.9	22.9	44.3	54.0
OTA [85]	1×	40.7	58.4	44.3	23.2	45.0	53.6
GFocal [12]	1×	40.1	58.2	43.1	23.3	44.4	52.5
GFocalV2 [57]	1×	41.1	58.8	44.9	23.5	44.9	53.3
LD (ours)	1×	**42.7**	**60.2**	**46.7**	**25.0**	**46.4**	**55.1**
ResNet-101 backbone on test-dev 2019							
RetinaNet [63]	2×	39.1	59.1	42.3	21.8	42.7	50.2
FCOS [17]	2×	41.5	60.7	45.0	24.4	44.8	51.6
SAPD [81]	2×	43.5	63.6	46.5	24.9	46.8	54.6
ATSS [76]	2×	43.6	62.1	47.4	26.1	47.0	53.6
BorderDet [82]	2×	45.4	64.1	48.8	26.7	48.3	56.5
AutoAssign [83]	2×	44.5	64.3	48.4	25.9	47.4	55.0
PAA [84]	2×	44.8	63.3	48.7	26.5	48.8	56.3
OTA [85]	2×	45.3	63.5	49.3	26.9	48.8	56.1
GFocal [12]	2×	45.0	63.7	48.9	27.2	48.8	54.5
GFocalV2 [57]	2×	46.0	64.1	50.2	27.6	49.6	56.5
LD (ours)	2×	**47.1**	**65.0**	**51.4**	**28.3**	**50.9**	**58.5**
ResNeXt-101-32x4d-DCN backbone on test-dev 2019							
SAPD [81]	2×	46.6	66.6	50.0	27.3	49.7	60.7
GFocal [12]	2×	48.2	67.4	52.6	29.2	51.7	60.2
GFocalV2 [57]	2×	49.0	67.6	53.4	29.8	52.3	61.8
LD (ours)	2×	**50.5**	**69.0**	**55.3**	**30.9**	**54.4**	**63.4**

下报告结果，以便进行公平比较。对于COCO test-dev 2019数据集，我们按照先前的工作 [57]，包含了使用多尺度1333 × [480 : 960]、2×的训练计划（24个epochs）训练的LD模型。训练是在一台具有8个GPU的机器节点上进行的，每个GPU的批量大小为2，初始学习率为0.01，以进行公平比较。在推理阶段，采用单尺度测试（[1333 × 800]分辨率）。对于不同的学生网络，如ResNet-50、ResNet-101和ResNeXt-101-32x4d-DCN [78], [79]，我们还选择了不同的教师网络，分别是ResNet-101、ResNet-101-DCN 和 Res2Net-101-DCN [80]。

如Tab. 8所示，我们的LD将SOTA GFocalV2的AP分数提高了+1.6，AP_{75}分数提高了+1.8，

当使用ResNet-50-FPN骨干网络时。当使用ResNet-101-FPN和ResNeXt-101-32x4d-DCN以多尺度2×训练时，我们实现了最高的AP分数，分别为47.1和50.5，在相同的骨干网络、neck和测试设置下，优于所有现有的密集目标检测器。更重要的是，我们的LD不会引入任何额外的网络参数或计算开销，因此可以保证与GFocalV2完全相同的推理速度。

5 总结

在本文中，我们提出了一种灵活的密集目标检测定位蒸馏方法，以及基于新的有价值定位区域的选择性区域蒸馏方法。我们展示了以下两点：1）对于目标检测，logit模仿可能比特征模仿更有效；2）在进行目标检测蒸馏时，通过选择性区域蒸馏来传递分类和定位知识是重要的。我们希望我们的方法能够为目标检测领域提供新的研究启示，以便开发更好的蒸馏策略。在未来，将LD方法应用于稀疏目标检测器（如DETR系列 [86]），异构目标检测器组合，以及其他相关领域，例如实例分割、目标跟踪和三维目标检测，都值得进一步研究。此外，由于我们的LD方法在优化效果上与分类蒸馏方法相当，一些改进的蒸馏方法可能也能够为LD带来增益，例如关系蒸馏 [23]、自蒸馏 [87], [88]、教师助理蒸馏 [24]和解耦蒸馏 [89]等。跨架构蒸馏利用最近的先进分类模型作为教师模型（如 [90], [91], [92], [93], [94]）也是一个有趣的探索方向。

致谢

本研究得到了国家自然科学基金委员会（NSFC）的支持（项目编号：62176130、62272311）、中国科协青年科技之星计划（编号：YESS20210377）、中央高校基本科研业务费专项资金（南开大学，项目编号：63223049）以及黑龙江省自然科学基金的资助（项目编号：YQ2022F004）。

参考文献

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[2] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Int. Conf. Learn. Represent.*, 2015.

[3] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Int. Conf. Learn. Represent.*, 2017.

[4] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: network compression via factor transfer," in *Adv. Neural Inform. Process. Syst.*, 2018, pp. 2765–2774.

[5] X. Jin, B. Peng, Y. Wu, Y. Liu, J. Liu, D. Liang, J. Yan, and X. Hu, "Knowledge distillation via route constrained optimization," in *Int. Conf. Comput. Vis.*, 2019.

[6] G.-H. Wang, Y. Ge, and J. Wu, "Distilling knowledge by mimicking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[7] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Adv. Neural Inform. Process. Syst.*, 2017.

[8] R. Sun, F. Tang, X. Zhang, H. Xiong, and Q. Tian, "Distilling object detectors with task adaptive regularization," *arXiv preprint arXiv:2006.13108*, 2020.

[9] T. Wang, L. Yuan, X. Zhang, and J. Feng, "Distilling object detectors with fine-grained feature imitation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[10] L. Zhang and K. Ma, "Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors," in *Int. Conf. Learn. Represent.*, 2020.

[11] Z. Kang, P. Zhang, X. Zhang, J. Sun, and N. Zheng, "Instance-conditional knowledge distillation for object detection," in *Adv. Neural Inform. Process. Syst.*, 2021.

[12] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: learning qualified and distributed bounding boxes for dense object detection," in *Adv. Neural Inform. Process. Syst.*, 2020.

[13] H. Qiu, H. Li, Q. Wu, and H. Shi, "Offset bin classification network for accurate object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Eur. Conf. Comput. Vis.*, 2016.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inform. Process. Syst.*, 2015.

[17] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Int. Conf. Comput. Vis.*, 2019.

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Adv. Neural Inform. Process. Syst.*, 2017.

[19] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: A novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 222103, pp. 1–21, 2020.

[20] Z. Zheng, R. Ye, P. Wang, D. Ren, W. Zuo, Q. Hou, and M. Cheng, "Localization distillation for dense object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[21] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Int. Conf. Learn. Represent.*, 2017.

[22] J.-H. Bae, D. Yeo, J. Yim, N.-S. Kim, C.-S. Pyo, and J. Kim, "Densely distilled flow-based knowledge transfer in teacher-student framework for image classification," *IEEE Transactions on Image Processing*, vol. 29, pp. 5698–5710, 2020.

[23] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[24] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Association for the Advancement of Artificial Intelligence*, 2020.

[25] W. Son, J. Na, J. Choi, and W. Hwang, "Densely guided knowledge distillation using multiple teacher assistants," in *Int. Conf. Comput. Vis.*, 2021.

[26] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[27] X. Dai, Z. Jiang, Z. Wu, Y. Bao, Z. Wang, S. Liu, and E. Zhou, "General instance distillation for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[28] J. Guo, K. Han, Y. Wang, H. Wu, X. Chen, C. Xu, and C. Xu, "Distilling object detectors via decoupled features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[29] D. Zhixing, R. Zhang, M. Chang, S. Liu, T. Chen, Y. Chen *et al.*, "Distilling object detectors with feature richness," in *Adv. Neural Inform. Process. Syst.*, 2021.

[30] G. Li, X. Li, Y. Wang, S. Zhang, Y. Wu, and D. Liang, "Knowledge distillation for object detection via rank mimicking and prediction-guided feature imitation," in *Association for the Advancement of Artificial Intelligence*, 2022.

[31] S. Gidaris and N. Komodakis, "Locnet: Improving localization accuracy for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[32] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[33] J. Wang, W. Zhang, Y. Cao, K. Chen, J. Pang, T. Gong, J. Shi, C. C. Loy, and D. Lin, "Side-aware boundary localization for more precise object detection," in *Eur. Conf. Comput. Vis.*, 2020.

[34] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[35] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[36] T. Kong, F. Sun, C. Tan, H. Liu, and W. Huang, "Deep feature pyramid reconfiguration for object detection," in *Eur. Conf. Comput. Vis.*, 2018.

[37] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "Scrdet: Towards more robust detection for small, cluttered and rotated objects," in *Int. Conf. Comput. Vis.*, 2019.

[38] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with gaussian wasserstein distance loss," in *International Conference on Machine Learning (ICML)*, 2021.

[39] X. Yang, X. Yang, J. Yang, Q. Ming, W. Wang, Q. Tian, and J. Yan, "Learning high-precision bounding box for rotated object detection via kullback-leibler divergence," in *Adv. Neural Inform. Process. Syst.*, 2021.

[40] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "Varifocalnet: An IoU-aware dense object detector," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[41] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2009.

[42] L. Han, P. Tao, and R. R. Martin, "Livestock detection in aerial images using a fully convolutional network," *Computational Visual Media*, vol. 5, no. 2, p. 221 – 228, 2019.

[43] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[44] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[45] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic R-CNN: Towards high quality object detection via dynamic training," in *Eur. Conf. Comput. Vis.*, 2020.

[46] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[47] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[48] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[49] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," *arXiv:1701.06659*, 2017.

[50] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[51] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: an advanced object detection network," in *ACM Int. Conf. Multimedia*, 2016.

[52] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A metric and a loss for bounding box regression," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[53] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and better learning for bounding box regression," in *Association for the Advancement of Artificial Intelligence*, 2020.

[54] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Transactions on Cybernetics*, 2021.

[55] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[56] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Int. Conf. Comput. Vis.*, 2019.

[57] X. Li, W. Wang, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[58] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Eur. Conf. Comput. Vis.*, 2018.

[59] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[60] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[61] Y. Zhou, X. Yang, G. Zhang, J. Wang, Y. Liu, L. Hou, X. Jiang, X. Liu, J. Yan, C. Lyu, W. Zhang, and K. Chen, "Mmrotate: A rotated object detection benchmark using pytorch," in *ACM Int. Conf. Multimedia*, 2022.

[62] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[63] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Int. Conf. Comput. Vis.*, 2017.

[64] W. Qian, X. Yang, S. Peng, Y. Guo, and J. Yan, "Learning modulated loss for rotated object detection," in *Association for the Advancement of Artificial Intelligence*, 2021.

[65] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *Eur. Conf. Comput. Vis.*, 2020.

[66] Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, and C. Yang, "PIoU loss: Towards accurate oriented object detection in complex environments," in *Eur. Conf. Comput. Vis.*, 2020.

[67] X. Yang, Y. Zhou, G. Zhang, J. Yang, W. Wang, J. Yan, X. Zhang, and Q. Tian, "The KFIoU loss for rotated object detection," in *Int. Conf. Learn. Represent.*, 2023.

[68] X. Yang, Q. Liu, J. Yan, A. Li, Z. Zhang, and G. Yu, "R3det: Refined single-stage detector with feature refinement for rotating object," in *Association for the Advancement of Artificial Intelligence*, 2021.

[69] J. Tang, R. Shivanna, Z. Zhao, D. Lin, A. Singh, E. H. Chi, and S. Jain, "Understanding and improving knowledge distillation," *arXiv preprint arXiv:2002.03532*, 2020.

[70] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.*, 2014.

[71] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[72] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3974–3983.

[73] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[75] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[76] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[77] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Adv. Neural Inform. Process. Syst.*, 2018.

[78] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[79] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[80] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2021.

[81] C. Zhu, F. Chen, Z. Shen, and M. Savvides, "Soft anchor-point object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[82] H. Qiu, Y. Ma, Z. Li, S. Liu, and J. Sun, "Borderdet: Border feature for dense object detection," in *Eur. Conf. Comput. Vis.*, 2020.

[83] B. Zhu, J. Wang, Z. Jiang, F. Zong, S. Liu, Z. Li, and J. Sun, "Autoassign: Differentiable label assignment for dense object detection," *arXiv preprint arXiv:2007.03496*, 2020.

[84] K. Kim and H. S. Lee, "Probabilistic anchor assignment with IoU prediction for object detection," in *Eur. Conf. Comput. Vis.*, 2020.

[85] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "OTA: Optimal transport assignment for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[86] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Eur. Conf. Comput. Vis.*, 2020.

[87] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *International Conference on Machine Learning (ICML)*, 2018, pp. 1607–1616.

[88] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Int. Conf. Comput. Vis.*, 2019, pp. 3713–3722.

[89] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[90] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[91] Q. Hou, C.-Z. Lu, M.-M. Cheng, and J. Feng, "Conv2former: A simple transformer-style convnet for visual recognition," *arXiv preprint arXiv:2211.11943*, 2022.

[92] Z. Dai, H. Liu, Q. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[93] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[94] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, p. 187 – 199, 2021.

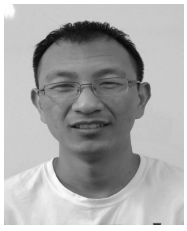
Zhaohui Zheng received the M.S. degree in computational mathematics from Tianjin University in 2021. He is currently a Ph.D. candidate with the School of Computer Science at Nankai University, Tianjin, China. His research interests include object detection, instance segmentation and knowledge distillation.

Rongguang Ye received the B.S. and M.S. degrees from the School of Mathematics, Tianjin University, Tianjin, China, in 2019 and 2022. He is now working at Intel Asia-Pacific Research And Development Ltd as an AI framework engineer. His research interests include object detection and computer vision.

Qibin Hou received his Ph.D. degree from the School of Computer Science, Nankai University. Then, he worked at the National University of Singapore as a research fellow. Now, he is an associate professor at School of Computer Science, Nankai University. He has published more than 30 papers on top conferences/journals, including T-PAMI, CVPR, ICCV, NeurIPS, etc. His research interests include deep learning and computer vision.

Dongwei Ren received two Ph.D. degrees in computer application technology from Harbin Institute of Technology and The Hong Kong Polytechnic University in 2017 and 2018, respectively. From 2018 to 2021, he was an Assistant Professor with the College of Intelligence and Computing, Tianjin University. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include computer vision and deep learning.

Ping Wang received the B.S., M.S., and Ph.D. degrees in computer science from Tianjin University, Tianjin, China, in 1988, 1991, and 1998, respectively. She is currently a Professor with the School of Mathematics, Tianjin University. Her research interests include image processing and machine learning.



Wangmeng Zuo received the Ph.D. degree from the Harbin Institute of Technology in 2007. He is currently a Professor in the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published over 100 papers in top tier journals and conferences. His publications have been cited more than 30,000 times in literature. He is on the editorial boards of IEEE TPAMI and IEEE TIP.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards including National Science Fund for Distinguished Young Scholars and ACM China Rising Star Award. He is on the editorial boards of IEEE TPAMI and IEEE TIP.