原文地址：EAST: An Efficient and Accurate Scene Text Detector

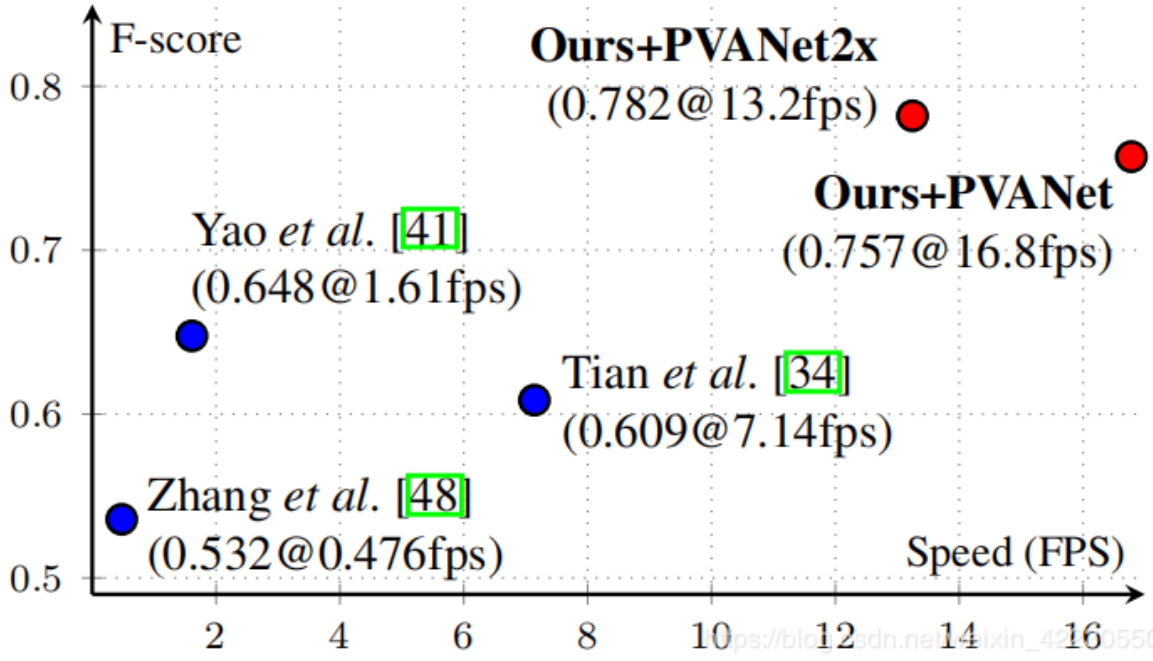
@[TOC] **高效准确的场景文本检测器**

摘要

先前的场景文本检测方法已经在各种环境中通过了性能基准。但是，在应对挑战性场景时（即使配备深度神经网络模型），因为整体性能取决于管道中多个阶段和组件的相互作用。在这项工作中，我们提出了一个简单但功能强大的管道，该管道可以在自然场景中产生快速而准确的文本检测。 流水线可以通过单个神经网络直接预测完整图像中任意方向和四边形形状的单词或文本行，从而消除了不必要的中间步骤（例如，候选聚合和单词划分）。 流水线的简单性使得我们可以集中精力设计损失函数和神经网络架构。 在包括ICDAR 2015，COCO-Text和MSRA-TD500在内的标准数据集上进行的实验表明，该算法在准确性和效率上均明显优于最新方法。 在ICDAR 2015数据集上，提出的算法在720p分辨率下以13.2fps的速度获得0.7820的F得分。

1.引言

近年来，提取和理解自然场景中包含的文本信息变得越来越重要和流行，这可以通过ICDAR系列竞赛的空前参与者[30、16、15]和NIST开展的TRAIT 2016评估来证明。 文本检测作为后续过程的前提条件，在文本信息的提取和理解的整个过程中起着至关重要的作用。 先前文本检测方法[2、33、12、7、48]已经在该领域的各种基准上获得了有希望的性能。 文本检测的核心是特征设计，以区分文本和背景。 传统上，特征是手动设计的[5、25、40、10、26、45]以捕获场景文本的属性，而在基于深度学习的方法中[3、13、11、12、7、48]可以从训练中直接学习有效的特征数据。



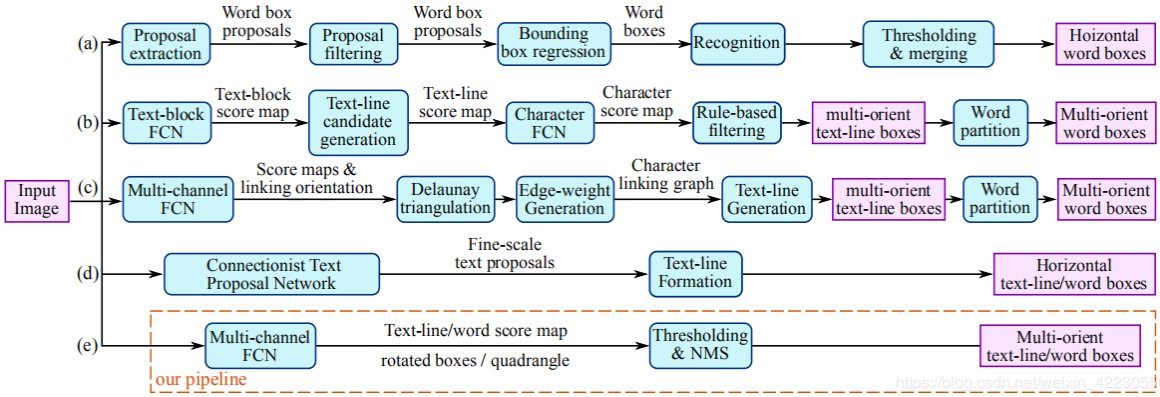
> 图1. ICDAR 2015 [15]文本本地化挑战的性能与速度。 可以看出，我们的算法在运行速度非常快的同时，其准确度大大超过了竞争对手。 选项卡6中列出了使用的硬件规格。*

然而，现有的方法，无论是传统方法还是基于深度神经网络的方法，都主要由几个阶段和组件组成，这可能是次优且耗时的。 因此，这种方法的准确性和效率仍然不能令人满意。 在本文中，我们提出了只有两个阶段的快速，准确的场景文本检测管道。 管道使用完全卷积网络（FCN）模型，该模型可直接生成单词或文本行级别的预测，不包括多余和缓慢的中间步骤。 生成的文本预测（可以是旋转的矩形或四边形）被发送到“非最大抑制”以产生最终结果。 与现有方法相比，根据标准基准的定性和定量实验，提出的算法可显着提高性能，同时运行速度更快。 具体来说，所提出的算法在ICDAR

2015 [15]上达到0.7820的F值（在多尺度下进行测试时为0.8072），在MSRA-TD500 [40]上达到0.7608的F值，在COCO-Text [36]上达到0.3945的F值， 先进的性能算法，平均所需时间更少（对于性能最佳的Titan-X GPU，在720p分辨率下为13.2fps，对于速度最快的模型为16.8fps）。 这项工作的贡献包括三个方面： •我们提出了一种场景文本检测方法，该方法包括两个阶段：完全卷积网络和NMS合并阶段。 FCN直接生成文本区域，不包括多余且耗时的中间步骤。 •管道可以灵活地生成单词级别或行级别的预测，根据特定的应用，其几何形状可以旋转为方框或四边形。
•所提出的算法在准确性和速度上均明显优于最新方法。

2.相关工作

长期以来，场景文本检测和识别一直是计算机视觉中的活跃研究主题。 研究了许多启发性的想法和有效的方法。 全面的评论和详细的分析可以在调查论文中找到。 本节将重点介绍与算法最相关的工作。 常规方法依赖于手动设计的特征。 基于笔划宽度变换（SWT）[5]和最大稳定的末梢区域（MSER）[25、26]的方法通常通过边缘检测或末梢区域提取来寻找候选字符。 张等 [47]利用文本的局部对称性，设计了用于文本区域检测的各种功能。 FASText [2]是一种快速的文本检测系统，它对著名的FAST关键点检测器进行了修改和改进，以进行笔画提取。 但是，这些方法在准确性和适应性方面都落后于基于深度神经网络的方法，尤其是在处理挑战性场景（例如低分辨率和几何失真）时。

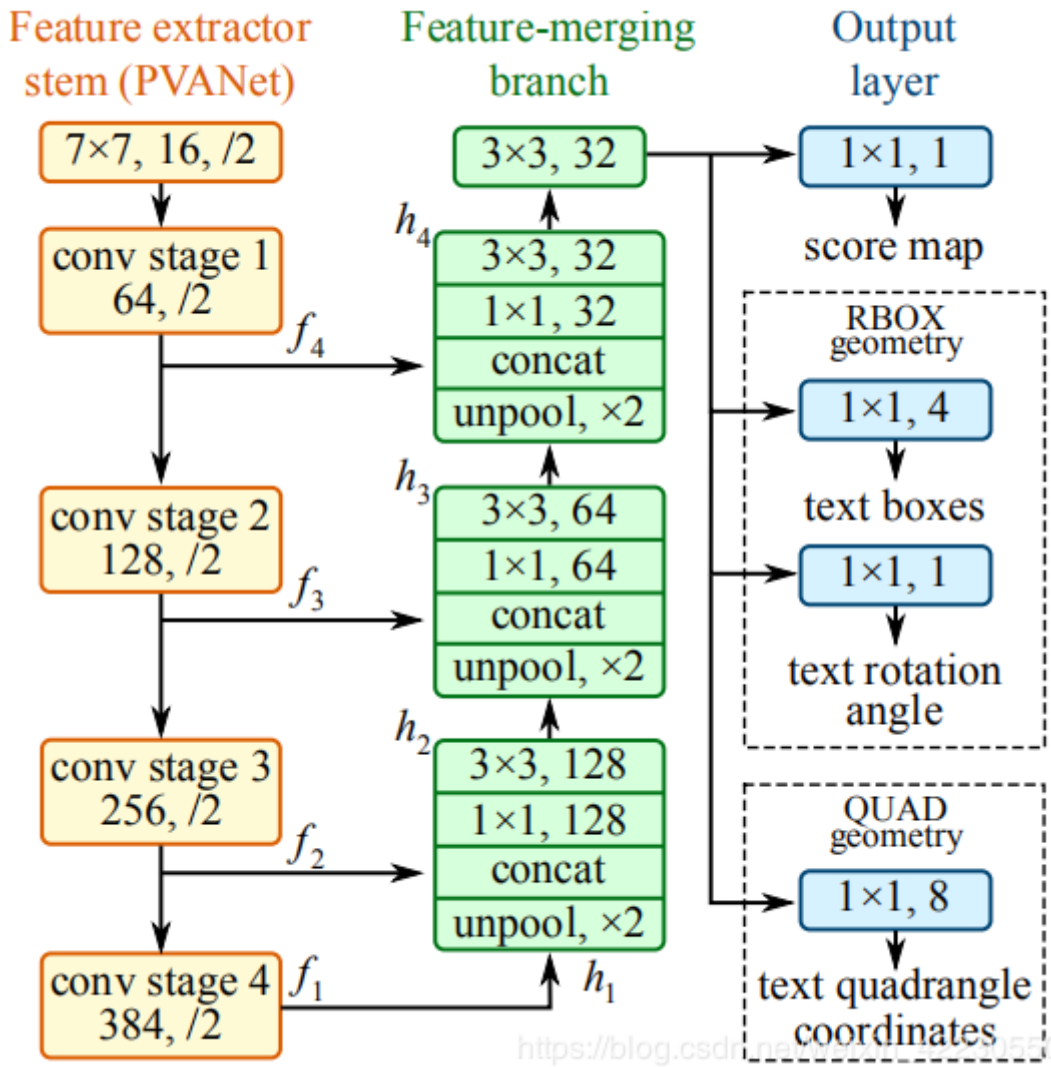


> 图2.关于场景文本检测的一些最新著作的流水线比较：（a）Jaderberg等人提出的水平单词检测和识别流水线。 [12]； （b）Zhang等人提出的多方向文本检测管道。 [48]； （c）Yao等人提出的多方向文本检测管道。 [41]； （d）Tian等人提出的使用CTPN进行水平文本检测。 [34]； （e）我们的管道消除了大多数中间步骤，仅包括两个阶段，比以前的解决方案简单得多。

最近，场景文本检测领域进入了一个新时代，基于深度神经网络的算法逐渐成为主流。黄等 [11]首先使用MSER找到候选者，然后使用深度卷积网络作为强分类器以消除误报。 Jaderberg等人的方法[13]以滑动窗口方式扫描图像，并使用卷积神经网络模型为每个比例生成密集的热图。后来，Jaderberg等 [12]使用CNN和ACF来搜寻候选单词，并使用回归进一步完善它们。田等 [34]开发了垂直锚，并构建了一个CNN-RNN联合模型来检测水平文本行。与这些方法不同，Zhang等 [48]提出利用FCN [23]生成热图，并使用分量投影进行方向估计。这些方法在标准基准上获得了出色的性能。但是，如图2（a-d）所示，它们主要由多个阶段和组件组成，例如通过后置过滤，候选聚集，行形成和字分配进行的误报消除。多个阶段和组件可能需要详尽的调整，导致性能欠佳，并增加了整个管道的处理时间。 在本文中，我们设计了一个基于FCN的深层管道，直接针对文本检测的最终目标：单词或文本行级别检测。 如图2（e）所示，该模型放弃了不必要的中间组件和步骤，并允许端到端训练和优化。最终的系统配备了单个轻量级的神经网络，在性能和速度上都明显超过了以前的所有方法。

3.方法

该算法的关键组成部分是神经网络模型，该模型经过训练可以直接从完整图像中预测文本实例的存在及其几何形状。 该模型是适用于文本检测的全卷积神经网络，可输出单词或文本行的每像素密集预测。 这消除了中间步骤，例如候选提议，文本区域形成和单词划分。 后处理步骤仅包括对预测的几何形状进行阈值处理和NMS。 该检测器被称为EAST，因为它是一种高效，准确的场景文本检测管道。



> 图3.我们的文本检测FCN的结构。

3.1.管道

图2（e）说明了我们的管道的高级概述。 该算法遵循DenseBox [9]的一般设计，其中将图像送入FCN，并生成像素级文本得分图和几何图形的多个通道。 预测通道之一是分数图，其像素值在[0，1]的范围内。 其余通道表示从每个像素的角度将单词括起来的几何形状。 分数代表在相同位置预测的几何形状的置信度。 我们针对文本区域尝试了两种几何形状，即旋转框（RBOX）和四边形（QUAD），并为每种几何设计了不同的损失函数。 然后将阈值应用于每个预测区域，在该区域中，其分数超过预定义阈值的几何形状将被视为有效并保存以供以后进行非最大抑制。 NMS之后的结果被认为是管道的最终输出。

3.2.网络设计

在设计用于文本检测的神经网络时，必须考虑几个因素。 如图5所示，在早期阶段，由于单词区域的大小变化很大，因此确定大单词的存在将需要神经网络后期的特征，而预测包围小单词区域的精确几何结构则需要低水平的信息 。 因此，网络必须使用不同级别的特征来满足这些要求。 HyperNet [19]在特征图上满足这些条件，但是在大型特征图上合并大量通道将显着增加以后阶段的计算开销。 为了解决这个问题，我们采用了U型[29]的思想，可以逐渐合并特征图，同时保持较小的上采样分支。 最终我们得到了可以同时利用不同级别的特征并保持少量计算成本的网络。 我们的模型的示意图如图3所示。该模型可以分解为三个部分：特征提取器，特征合并分支和输出层。 特征提取器可以是在ImageNet [4]数据集上经过预训练的卷积网络，具有交织的卷积和池化层。从词干中提取四个级别的特征图，表示为f_i，其大小分别为输入图像的$\frac{1}{32}$，$\frac{1}{16}$，$\frac{1}{8}$和$\frac{1}{4}$。在图3中，描绘了PVANet [17]。 在我们的实验中，我们还采用了众所周知的VGG16 模型，该模型提取池化-2到池化-5之后的特征图。 在特

征合并分支中，我们逐渐将它们合并：

$$g_i = \begin{cases} \text{unpool}(h_i) & \text{if} \quad i \leq 3 \\ \text{conv}_{3\times3}(h_i) & \text{if} \quad i = 4 \end{cases} \tag{1}$$

$$h_i = \begin{cases} f_i & \text{if} \quad i = 1 \\ \text{conv}_{3\times3}(\text{conv}_{1\times1}([g_{i-1}; f_i])) & \text{otherwise} \end{cases} \tag{2}$$

其中 g_i 是合并基础，h_i是合并特征图，而运算符$[\cdot;\cdot]$表示沿着通道轴的串联。 在每个合并阶段，将从最后阶段开始的特征图首先送到解池层，以使其尺寸加倍，然后与当前特征图连接。 接下来，$conv_{1\times1}$瓶颈[8]减少了通道数量并减少了计算量，紧接着是$conv_{3\times3}$，它融合了信息以最终产生此合并阶段的输出。 在最后的合并阶段之后，$conv_{3\times3}$层将生成合并分支的最终特征图，并将其送到输出层。

每个卷积的输出通道数如图3所示。我们将分支中卷积的通道数保持较小，这仅增加了特征提取器上计算开销的一小部分，从而使网络计算效率更高。 最终输出层包含几个$conv_{1\times1}$操作，以将32个特征图通道投影到得分图F_s和多通道几何图F_g的1个通道中。 几何输出可以是RBOX或QUAD之一，汇总在Table 1中。

Geometry	channels	description
AABB	4	$\mathbf{G} = \mathbf{R} = \{d_i \vert i \in \{1,2,3,4\}\}$
RBOX	5	$\mathbf{G} = \{\mathbf{R}, \theta\}$
QUAD	8	$\mathbf{G} = \mathbf{Q} = \{(\Delta x_i, \Delta y_i) \vert i \in \{1,2,3,4\}\}$

> 表1.输出几何设计

对于RBOX，其几何形状由4个轴对齐的边界框（AABB）R和1个通道旋转角度θ表示。 R的公式与[9]中的公式相同，其中4个通道分别代表从像素位置到矩形的顶部，右侧，底部，左侧边界的4个距离。 对于QUAD Q，我们使用8个数字来表示从四边形的四个角顶点$\{p_i \vert i \in \{1、2、3、4\}\}$到像素位置的坐标偏移。 由于每个距离偏移都包含两个数字$(\Delta x_i, \Delta y_i)$，所以几何输出包含8个通道。

3.3.标签生成

3.3.1.四边形的得分图生成

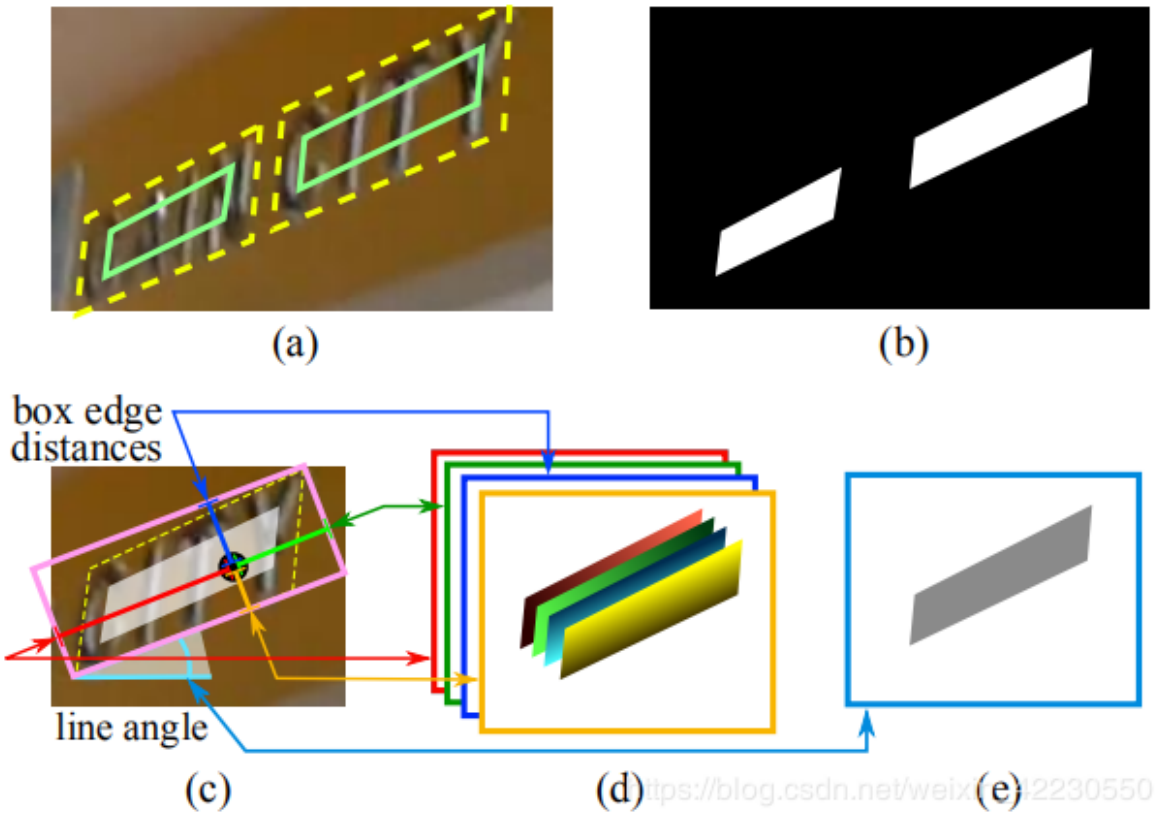


图4.标签生成过程：（a）文本四边形（黄色虚线）和缩小的四边形（绿色实线）；（b）文本得分图；（c）RBOX几何图生成；（d）每个像素到矩形边界的距离的4个通道；（e）旋转角度。

不失一般性，我们仅考虑几何形状为四边形的情况。分数图上四边形的正区域设计为原始图的缩小版本，如图4（a）所示。对于四边形$Q = \{p_i | i \in \{1,2,3,4\}\}$，其中$p_i = \{x_i, y_i\}$是四边形上按顺时针顺序排列的顶点。为了缩小Q，我们首先为每个顶点p_i计算参考长度r_i：

$$r_i = \min\left(\mathrm{D}(p_i, p_{(i \bmod 4)+1}), \mathrm{D}(p_i, p_{((i+2) \bmod 4)+1})\right) \quad (3)$$

其中$D(p_i, p_j)$是p_i和p_j之间的距离L2。我们首先缩小四边形的两个较长边缘，然后缩小两个较短的边缘。对于每对两个相对的边缘，我们通过比较其长度的平均值来确定"更长"的一对。对于每个边缘$< p_i, p_{(i \bmod 4)+1} >$，我们分别通过沿边缘向内移动两个端点$0.3r_i$和$0.3r_{(i \bmod 4)+1}$来缩小它。

3.3.2.几何图生成

如第二节（3.2）所述。几何图谱是RBOX或QUAD之一。RBOX的生成过程如图4（c-e）所示。

对于文本区域以QUAD样式标注的那些数据集（例如ICDAR 2015），我们首先生成一个旋转的矩形，该矩形覆盖面积最小的区域。然后，对于每个具有正分数的像素，我们计算其到文本框4个边界的距离，并将它们放置到RBOX地面实况的4个通道中。对于QUAD地面实况，在8通道几何图中的每个具有正分数的像素的值是其从四边形的4个顶点处的坐标偏移。

3.4.损失函数

损失函数可以表述为

$$L = L_s + \lambda_g L_g \quad (4)$$

其中L_s和L_g分别代表得分图和几何体的损失，而λ_g权衡两次损失之间的重要性。在我们的实验中，我们将λ_g设置为1。

3.4.1.得分图损失

在大多数最先进的检测管道中，训练图像都是通过平衡采样和硬负片挖掘精心处理的，以应对目标对象的不平衡分布[9，28]。 这样做可能会改善网络性能。 但是，使用这样的技术不可避免地会引入不可微的阶段和更多的参数来进行调整以及更复杂的管线，这与我们的设计原理相矛盾。 为了简化训练过程，我们使用[38]中引入的类平衡交叉熵：

$$\begin{aligned} L_{\mathrm{s}} &= \text{balanced-xent}(\hat{\mathbf{Y}}, \mathbf{Y}^*) \\ &= -\beta \mathbf{Y}^* \log \hat{\mathbf{Y}} - (1-\beta)(1-\mathbf{Y}^*) \log(1-\hat{\mathbf{Y}}) \end{aligned} \tag{5}$$

其中$\hat{Y} = F_s$是得分图的预测，而Y^*是真实值。 参数β是正负样本之间的平衡因子，由下式给出

$$\beta = 1 - \frac{\sum_{y^* \in \mathbf{Y}^*} y^*}{|\mathbf{Y}^*|}. \tag{6}$$

这种平衡的交叉熵损失，由Yao等人首先在文本检测中作为得分图预测的目标函数使用。 我们发现它在实践中效果很好。

3.4.2.几何体损失

文本检测的一个挑战是自然场景图像中文本的大小差异很大。 直接使用L1或L2损失进行回归将导致损失偏向更大和更长的文本区域。 由于我们需要为大型和小型文本区域生成准确的文本几何形状预测，因此回归损失应该是比例不变的。 因此，我们在RBOX回归的AABB部分采用IoU损失，并在QUAD回归中采用尺度标准化的平滑L1损失。

RBOX 对于AABB部分，我们采用了IoU损耗，因为它对于不同尺度的物体是不变的。

$$L_{\mathrm{AABB}} = -\log \mathrm{IoU}(\hat{\mathbf{R}}, \mathbf{R}^*) = -\log \frac{|\hat{\mathbf{R}} \cap \mathbf{R}^*|}{|\hat{\mathbf{R}} \cup \mathbf{R}^*|} \tag{7}$$

其中$\hat{\mathbf{R}}$代表预测的AABB几何形状，$\mathbf{R}^*$为其对应的真实值。 很容易看出相交矩形的宽度和高度$|\hat{\mathbf{R}} \cap \mathbf{R}^*|$是

$$\begin{aligned} w_{\mathbf{i}} &= \min(\hat{d}_2, d_2^*) + \min(\hat{d}_4, d_4^*) \\ h_{\mathbf{i}} &= \min(\hat{d}_1, d_1^*) + \min(\hat{d}_3, d_3^*) \end{aligned} \tag{8}$$

其中d1，d2，d3和d4分别表示从像素到其相应矩形的顶部，右侧，底部和左侧边界的距离。 联合面积由以下公式给出：

$$|\hat{\mathbf{R}} \cup \mathbf{R}^*| = |\hat{\mathbf{R}}| + |\mathbf{R}^*| - |\hat{\mathbf{R}} \cap \mathbf{R}^*|. \tag{9}$$

因此，可以容易地计算交点/联合区域。 接下来，旋转角的损失计算为

$$L_\theta(\hat{\theta}, \theta^*) = 1 - \cos(\hat{\theta} - \theta^*). \tag{10}$$

其中$\hat{\theta}$是对旋转角度的预测，而θ^*表示真实值。 最后，整体几何损失是AABB损失和角度损失的加权总和，由

$$L_{\mathrm{g}} = L_{\mathrm{AABB}} + \lambda_\theta L_\theta. \tag{11}$$

在我们的实验中，λ_θ设置为10。 请注意，无论旋转角度如何，我们都会计算L_{AABB}。这可以被看作是四边形 IoU 的近似值时的角度是完全预测。 尽管在训练期间不是这种情况，但它仍可以为网络施加正确的梯度以学习预测$\hat{\mathbf{R}}$。

QUAD 我们通过添加为单词四边形设计的额外归一化项来扩展在[6]中提出的平滑L1损失，通常在一个方向上更长。 令Q的所有坐标值为有序集

$$\mathrm{C}_{\mathbf{Q}} = \{x_1, y_1, x_2, y_2, \ldots, x_4, y_4\} \tag{12}$$

那么损失函数可以写成：

$$\begin{aligned} L_{\mathrm{g}} &= L_{\mathrm{QUAD}}(\hat{\mathbf{Q}}, \mathbf{Q}^*) \\ &= \min_{\tilde{\mathbf{Q}} \in P_{\mathbf{Q}^*}} \sum_{\substack{c_i \in \mathrm{C}_{\mathbf{Q}}, \\ \tilde{c}_i \in \mathrm{C}_{\tilde{\mathbf{Q}}}}} \frac{\mathrm{smoothed}_{L1}(c_i - \tilde{c}_i)}{8 \times N_{\mathbf{Q}^*}} \end{aligned} \tag{13}$$

其中归一化项 N_{Q^*} 是四边形的短边长度，由下式给出

$$N_{\mathbf{Q}^*} = \min_{i=1}^{4} D(p_i, p_{(i \bmod 4)+1}), \tag{14}$$

P_Q是Q^*的所有等效四边形的集合，具有不同的顶点顺序。 由于公共训练数据集中四边形的注释不一致，因此需要此排序排列。

3.5.训练

使用ADAM 优化器对网络进行端到端训练。 为了加快学习速度，我们从图像中均匀采样512x512样本，以形成大小为24的小批量。ADAM的学习率从1e-3开始，每27300个小批量下降到十分之一，然后在1e-5停止。 训练网络，直到性能不再提高。

3.6.位置感知NMS

为了形成最终结果，应将阈值后幸存的几何形状由NMS合并。 NMS算法以$O(n^2)$的时间复杂度运行，其中n是候选几何图形的数量，这是不可接受的，因为我们正面临来自密集预测的数以万计的几何图形。 在假设来自附近像素的几何图形倾向于高度相关的假设下，我们建议逐行合并几何图形，并且在合并同一行中的几何图形时，我们将迭代地合并当前遇到的几何图形与最后合并的几何图形。 在最佳方案中，这种改进的技术可以在$O(n)$中运行。 即使最坏的情况与朴素一样，只要保持局部性假设，该算法在实践中就足够快地运行。 该过程在算法1中进行了总结。 值得一提的是，在WEIGHTEDMERGE（g，p）中，合并后的四边形的坐标由两个给定四边形的分数加权平均。 具体来说，如果$a = WEIGHTEDMERGE(g, p)$，则$a_i = V(g)g_i + V(p)p_i$和$V(a) = V(g) + V(p)$，其中a_i是 a 其中之一 下标 i 的坐标，而V(a)是几何 a 的分数。 实际上，我们有一个微妙的区别，我们是“平均”而不是“选择”几何形状，就像在标准NMS程序中那样，它起着投票机制的作用，反过来在输

入视频时引入了稳定效果。 尽管如此，在功能描述上我们仍然采用“ NMS”一词。

Algorithm 1 Locality-Aware NMS

```
function NMSLocality(geometries)
    S ← ∅, p ← ∅
    for g ∈ geometries in row first order do
        if p ≠ ∅ ∧ ShouldMerge(g, p) then
            p ← WeightedMerge(g, p)
        else
            if p ≠ ∅ then
                S ← S ∪ {p}
            end if
            p ← g
        end if
    end for
    if p ≠ ∅ then
        S ← S ∪ {p}
    end if
    return StandardNMS(S)
end function
```


4.实验

为了将所提出的算法与现有方法进行比较，我们在三个公共基准上进行了定性和定量实验：ICDAR2015, COCO-Text 和MSRA-TD500.

4.1.基准数据集

ICDAR 2015 用于ICDAR 2015健壮阅读竞赛[15]的挑战4。 它总共包括1500张图片，其中1000张用于训练，其余用于测试。 文本区域由四边形的四个顶点注释，对应于本文中的QUAD几何形状。 我们还通过拟合具有最小面积的旋转矩形来生成RBOX输出。 这些图像是由Google Glass附带拍摄的。 因此，场景中的文本可以处于任意方向，或遭受运动模糊和低分辨率的困扰。 我们还使用了ICDAR 2013的229张训练图像。

COCO-Text [36]是迄今为止最大的文本检测数据集。 它重用了来自MS-COCO数据集的图像[22]。总共标注了63,686张图像，其中选择了43,686张作为训练集，其余20,000张用于测试。 单词区域以轴对齐边界框（AABB）的形式注释，这是RBOX的特例。 对于此数据集，我们将角度θ设置为零。 我们使用与ICDAR 2015中相同的数据处理和测试方法。

MSRA-TD500 [40]是一个包含300个训练图像和200个测试图像的数据集。 文本区域具有任意的方向，并在句子级别进行注释。 与其他数据集不同，它包含英文和中文文本。 文本区域以RBOX格式注释。 由于训练图像的数量太少而无法学习深度模型，因此我们还利用了HUSTTR400数据集[39]中的400幅图像作为训练数据。

4.2.基本网络

除了COCO-Text以外，所有文本检测数据集与用于一般对象检测的数据集相比都相对较小，因此，如果所有基准均采用单个网络，则可能会出现过度拟合或不足的情况。 我们在所有数据集上实验了三种具有不同输出几何形状的基本网络，以评估所提出的框架。 这些网络汇总在表2中。

Network	Description
PVANET [17]	small and fast model
PVANET2x [17]	PVANET with 2x number of channels
VGG16 [32]	commonly used model

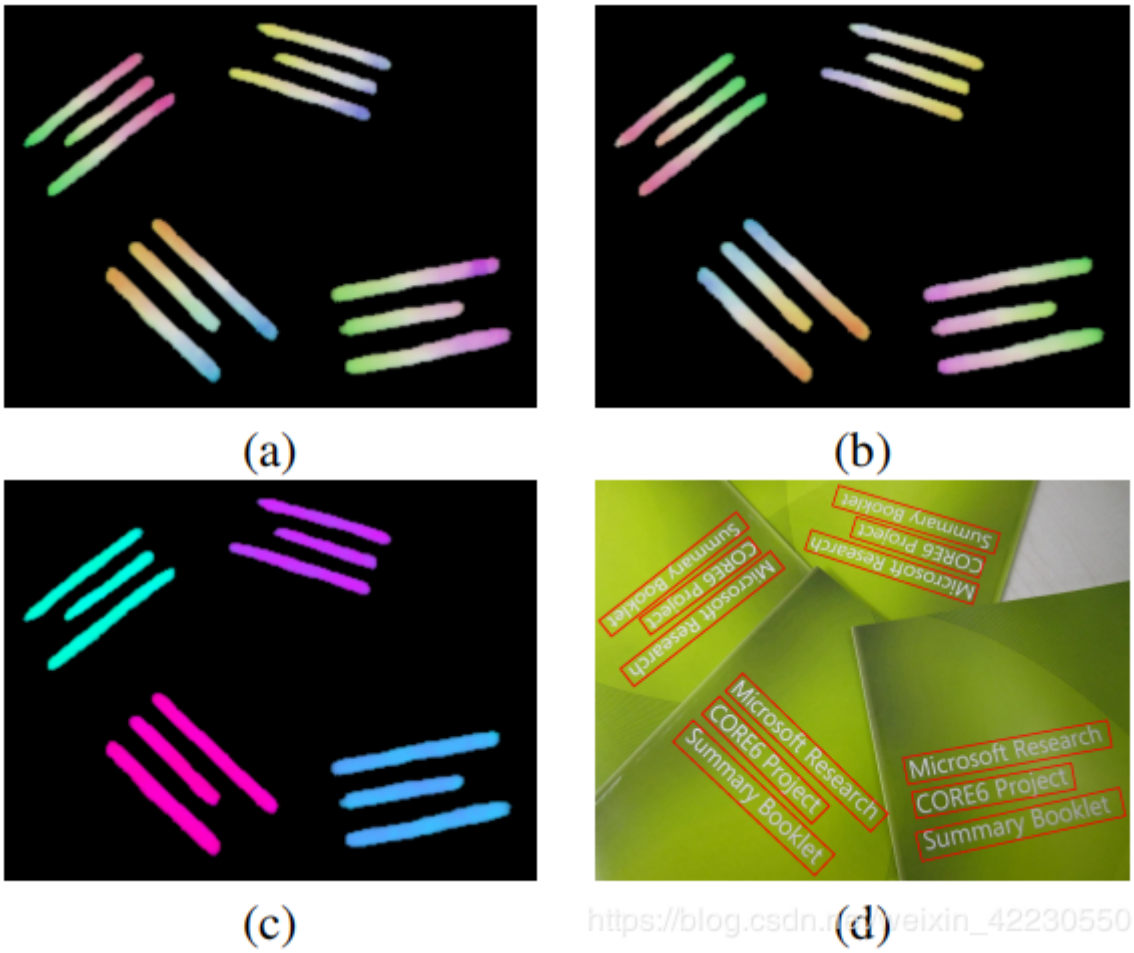
表2 基本模型

VGG16 [32]在许多任务中被广泛用作基础网络，以支持后续的特定于任务的微调，包括文本检测。 该网络有两个缺点：（1） 该网络的接受范围很小。 conv5_3的输出中的每个像素仅具有196的接收区域。（2） 这是一个相当大的网络。 **PVANET** 是[17]中引入的轻量级网络，旨在替代Faster-RCNN [28]框架中的特征提取器。 由于对于GPU来说太小了，无法充分利用计算并行性，因此我们还采用了 PVANET2x，它使原始PVANET的通道增加了一倍，从而利用了更多的计算并行性，同时运行速度比PVANET稍慢。 这将在第 4.5节中详细介绍。 最后一个卷积层的输出的接收区域为809，比VGG16大得多。 这些模型已在ImageNet数据集上进行了预训练。

4.3.定性结果

图5描述了所提出算法的几个检测示例。 它能够处理各种挑战性场景，例如照明不均匀，分辨率低，方向变化和透视变形。 此外，由于NMS程序中的投票机制，所提出的方法在具有各种形式的文本实例的视频上显示出很高的稳定性。 所提出的方法的中间结果如图6所示。可以看出，经过训练的模型可以生成高度精确的几何图谱和分数图谱，其中可以轻松形成不同方向的文本实例的检测。

图5.提出算法的定性结果。(a) ICDAR 2015. (b) MSRA-TD500. (c) COCO-Text.



> 图6.所提出算法的中间结果。 （a）估计的d1和d4几何图。 （b）d2和d3的估计几何图。 （c）文本实例的估计角度图。 （d）预测的文本实例的旋转矩形。 （a），（b）和（c）中的贴图用颜色编码，以像素方式表示方差（对于d1，d2，d3和d4）和不变性（对于角度）。 请注意，在几何图中，仅前景像素的值有效。

4.4.定量结果

如表3和表4所示，我们的方法在ICDAR 2015和COCO-Text上大大优于以前的最新方法。

在ICDAR 2015挑战4中，当以原始比例输入图像时，提出的方法的F得分为0.7820。 当使用同一网络在多个尺度（相对比例为0.5、0.7、1.0、1.4和2.0）上进行测试时，我们的方法在F得分中达到0.8072，就绝对值而言，它比最佳方法高出近0.16（0.8072与0.6477）。

使用VGG16网络比较结果，当使用QUAD输出时，提出的方法也比以前的最佳方法好0.0924，当使用RBOX输出时则优于0.116。 同时，这些网络非常有效，如第4.5节所示。

在COCO-Text中，所提出算法的所有三个设置都比以前的最佳执行者具有更高的准确性[41]。 具体地说，考虑到COCO-Text是迄今为止最大和最具挑战性的基准，F得分对[41]的改进为0.0614，而召回率的改进为0.053，这确认了所提出算法的优势。 请注意，我们也将[36]中的结果作为参考，但是这些结果实际上不是有效的基线，因为这些方法（A，B和C）是在数据注释中使用的。

相对于以前的方法，所提出算法的改进证明，直接针对最终目标并消除冗余过程的简单文本检测管道可以击败复杂的管道，甚至与大型神经网络模型集成的管道也能胜过。

如表5所示，在MSRA-TD500上，我们方法的所有三个设置均取得了优异的效果。 表现最佳的F分数（Ours + PVANET2x）略高于[41]。 与Zhang等人的方法相比。 [48]是先前发布的最先进的系统，性能最好的（Ours + PVANET2x）在F得分上提高了0.0208，在精度上提高了0.0428。

请注意，在我们的配备VGG16的算法的MSRA-TD500上，其性能比PVANET和PVANET2x的要差得多（0.7023与0.7445和0.7608），主要原因是VGG16的有效接收区域小于PVANET和PVANET2x，而MSRA-TD500的评估协议要求文本检测算法输出行级别而不是单词级别预测。

此外，我们还根据ICDAR 2013基准评估了Ours + PVANET2x。 它的召回率，精度和F得分达到0.8267、0.9264和0.8737，与以前的最新方法[34]相当，后者的召回率，精度和F得分分别为0.8298、0.9298和0.8769。

Algorithm	Recall	Precision	F-score
Ours + PVANET2x RBOX MS*	**0.7833**	0.8327	**0.8072**
Ours + PVANET2x RBOX	0.7347	**0.8357**	**0.7820**
Ours + PVANET2x QUAD	0.7419	0.8018	0.7707
Ours + VGG16 RBOX	0.7275	0.8046	0.7641
Ours + PVANET RBOX	0.7135	0.8063	0.7571
Ours + PVANET QUAD	0.6856	0.8119	0.7434
Ours + VGG16 QUAD	0.6895	0.7987	0.7401
Yao *et al.* [41]	0.5869	0.7226	0.6477
Tian *et al.* [34]	0.5156	0.7422	0.6085
Zhang *et al.* [48]	0.4309	0.7081	0.5358
StradVision2 [15]	0.3674	0.7746	0.4984
StradVision1 [15]	0.4627	0.5339	0.4957
NJU [15]	0.3625	0.7044	0.4787
AJOU [20]	0.4694	0.4726	0.4710
Deep2Text-MO [45, 44]	0.3211	0.4959	0.3898
CNN MSER [15]	0.3442	0.3471	0.3457

表3. ICDAR 2015挑战4偶然场景文本本地化任务的结果。 MS表示多尺度测试。

Algorithm	Recall	Precision	F-score
Ours + VGG16	**0.324**	**0.5039**	**0.3945**
Ours + PVANET2x	0.340	0.406	0.3701
Ours + PVANET	0.302	0.3981	0.3424
Yao *et al.* [41]	0.271	0.4323	0.3331
Baselines from [36]			
A	0.233	0.8378	0.3648
B	0.107	0.8973	0.1914
C	0.047	0.1856	0.0747

表4. COCO-Text上的结果。

Algorithm	Recall	Precision	F-score
Ours + PVANET2x	0.6743	**0.8728**	**0.7608**
Ours + PVANET	0.6713	0.8356	0.7445
Ours + VGG16	0.6160	0.8167	0.7023
Yao *et al.* [41]	**0.7531**	0.7651	0.7591
Zhang *et al.* [48]	0.67	0.83	0.74
Yin *et al.* [44]	0.63	0.81	0.71
Kang *et al.* [14]	0.62	0.71	0.66
Yin *et al.* [45]	0.61	0.71	0.66
TD-Mixture [40]	0.63	0.63	0.60
TD-ICDAR [40]	0.52	0.53	0.50
Epshtein *et al.* [5]	0.25	0.25	0.25

表5. MSRA-TD500上的结果。

4.5.速度比较

表6中演示了整体速度比较。我们报告的数字是使用性能最佳的网络以原始分辨率（1280x720）运行ICDAR 2015数据集中的500张测试图像的平均值。 这些实验是在服务器上使用具有Maxwell架构的单个NVIDIA Titan X显卡和Intel E5-2670 v3 @ 2.30GHz CPU进行的。 对于提出的方法，后处理包括阈值处理和NMS，而其他处理则应参考其原始论文。 所提出的方法不仅明显优于最新方法，而且由于简单高效的管道，计算成本也非常低。 从表6中可以看出，我们方法的最快设置以16.8 FPS的速度运行，而最慢的设置以6.52 FPS的速度运行。 甚至性能最好的型号Ours + PVANET2x的运行速度也为13.2 FPS。 这证实了我们的方法是最有效的文本检测器之一，该检测器可实现基准测试的最新性能。

Approach	Res.	Device	T_1/T_2 (ms)	FPS
Ours + PVANET	720p	Titan X	58.1 / 1.5	16.8
Ours + PVANET2x	720p	Titan X	73.8 / 1.7	13.2
Ours + VGG16	720p	Titan X	150.9 / 2.4	6.52
Yao *et al.* [41]	480p	K40m	420 / 200	1.61
Tian *et al.* [34]	ss-600*	GPU	130 / 10	7.14
Zhang *et al.* [48]*	MS*	Titan X	2100 / N/A	0.476

表6.不同方法下的总时间消耗 T1是网络预测时间，T2占后处理所用的时间。 对于田等。 [34]，ss-600表示短边是600，而130ms包括两个网络。 请注意，它们在ICDAR 2015上以2000的短边达到了最佳结果，这比我们的短得多。 对于张等。 [48]，MS表示他们使用了200、500、1000三个音阶，其结果是在MSRA-TD500上获得的。 对于三个模型，PVANET，PVANET2x和VGG16的理论每个像素的理论触发器分别为18KOps，44.4KOps和331.6KOps。

4.6.局限性

检测器可以处理的文本实例的最大大小与网络的接收区域成比例。 这限制了网络预测更长的文本区域（例如跨图像分布的文本行）的能力。

此外，由于垂直文本实例仅占用ICDAR 2015训练集中一小部分文本区域，因此该算法可能会遗漏或给出不准确的预测。

5.总结与未来工作

我们提供了一种场景文本检测器，该检测器可通过单个神经网络直接从完整图像中生成单词或行级别的预测。 通过结合适当的损失函数，检测器可以根据特定的应用预测文本区域的旋转矩形或四边形。在标准基准上进行的实验证实，该算法在准确性和效率上都大大优于以前的方法。 未来研究的可能方向包括：（1）修改几何公式以直接检测弯曲文本；（2）将检测器与文本识别器集成在一起；（3）将思想扩展到一般目标检测。