EAST：AN EFFICIENT AND ACCURATE SCENE TEXT DETECTOR

Megvii Technology Inc.,Beijing,China

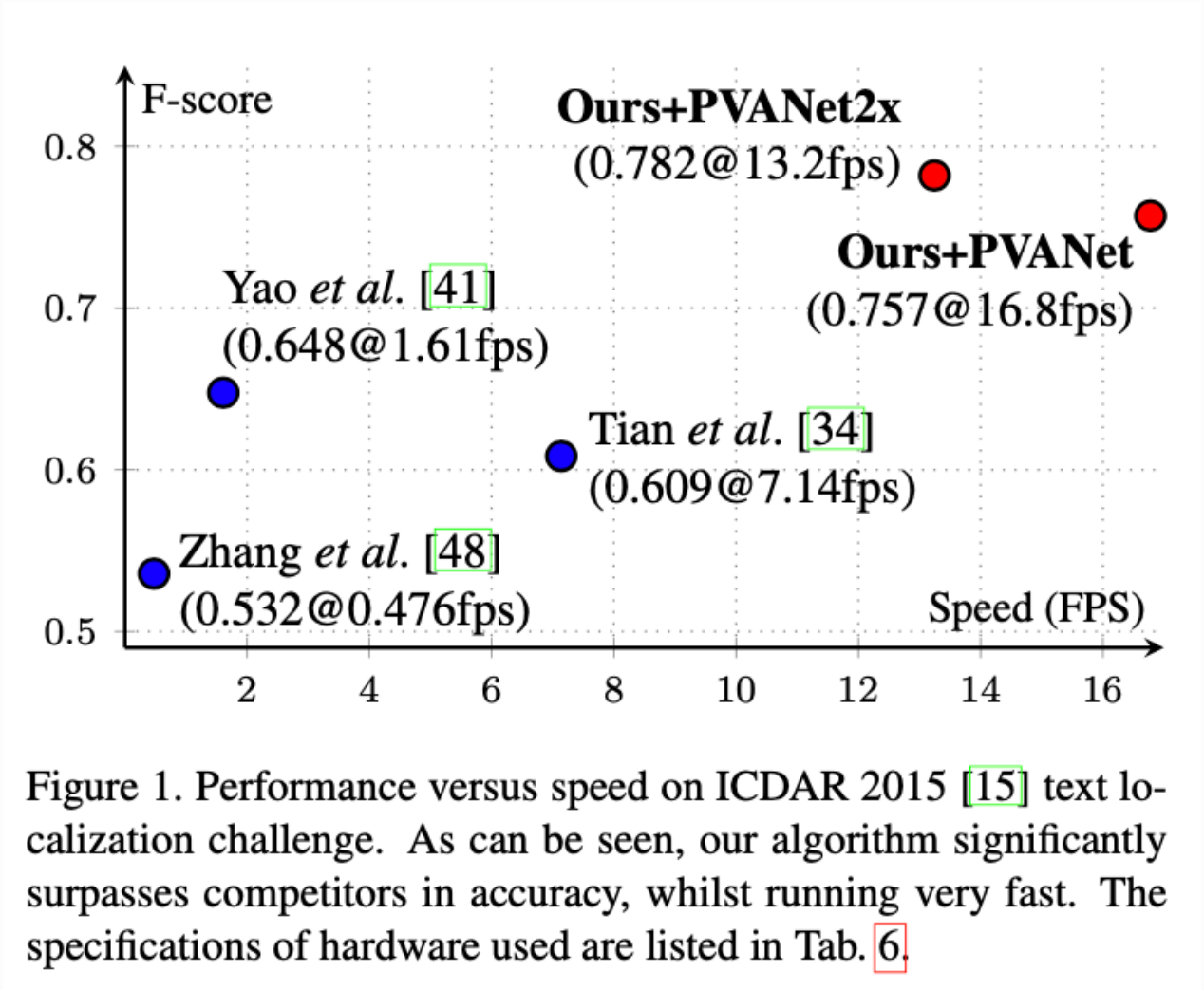

Figure 1. Performance versus speed on ICDAR 2015 [15] text localization challenge. As can be seen, our algorithm significantly surpasses competitors in accuracy, whilst running very fast. The specifications of hardware used are listed in Tab. 6.

INTRODUCTION

提取并理解自然场景下的文字变得越来越重要。

因为像ICDAR系列的比赛和NIST的TRAIT2016这阵子空前火爆。

(traiT2016)(https://www.nist.gov/programs-projects/text-recognition-algorithm-independent-evaluation-trait-2016)，美国政府机构组织的场景文字识别活动，用于儿童黄色影相的犯罪资料调查。

NIST：国家标准与技术研究所(usa)

在文字信息提取和理解中，文字检测是必备步骤，是整个环节中的关键角色。现有的文字检测工作【2,33,12,7,48】已经取得了不错的性能，在各种各样的数据集上。这项算法的核心是设计特征来区分前景和背景。传统的方法就是人工设计了一些特征，如【5，25，40，10，26，45】，深度学习就是用学习的方法来学到一些有效的特征。

但是现存的方法，无论是传统还是基于深度学习的，整个算法大多数是经过了几个阶段和组件的。这样的方法大概率是次优的而且更加耗时。因此这写方法的准确率和速度理落地还很远。

本文提出一个能落地的方法：又快又准。只包含两个步骤. 本文实现了一个fully convolutinal network(FCN) 模型，这个模型可以直接检测一个词。也可以直接检测一个句子。没有任何冗余和慢的中间步骤。这个全卷积的模型输出的检测结果作为候选框（可以以带角度矩形给出，也可以一个四边形对应的8个点来给出），然后将这些候选框输入到Non-Maximum Suppression，得到最终的结果。

本方法与现有方法比性能更好，速度更快。这个可以参考后面的标准数据库上的定性和定量分析。

本方法在ICDAR 2015上的F-scale 为0.7820.MSRA-TD500上为0.7608，COCO-Text上为0.3942。性能上sota,速度最快可以达到16.8fps,性能最高时的模型也可以达到13.2fps,输入图片是720p，显卡Titan-X GPU。

本文贡献：

1. 提出一个两阶段文字检测模型：FCN和local-aware NMS.FCN直接预测位置，不包含中间冗余的那些耗时步骤。
2. 本方法经过不同的训练，可以直接检测一个词，也可以检测一行文字。检测结果可以用一个旋转矩形【RBOX】给出，也可以直接给出四边形【QUAD】的四个点。
3. 性能，速度都显著超过sota.

RELATED WORK

场景文字检测是计算机视觉中的一项比较长期的研究。有很多方法，也有一些综述和细节分析文章【50，35，43】。我们这里会说一些聚焦在与文本思路联系比较紧密的方法上。

传荣方法主要是人工设计的特征。基于边缘检测和极致区域的提取的方法有Stroke Width Transform 和 Maximally Stable Extremal Regions.也有基于局部对称特性的特征，有基于FAST key point检测方法的FASText .但是这些方法后来都落后于深度学习的方法，无论是在准确率还是鲁棒性上，特别是低分辨率和文字有扭曲的时。

最近深度学习是新的主要方法。有人用MSER来获取候选，然后用深度卷积网络做分类，精简误识别。有人用结合滑窗给出每个尺度上的热图。有人开发一个垂直anchor,结合CNN-RNN这样的联合模型来检测一行水平文字。【48】提出了一个FCN，输出热图，然后用组件映射的方法估计角度。

这些方法在标准数据上取得了非常好的性能。

但是他们都有个缺点：阶段太多，组件太多。比如post filtering，candidate aggregation,line formation,word partition.这些都会大大加大训练难度，降低性能，增加耗时。

本文是a deep FCN-based pipeline 只指最终目标： text detection，“词”或者“行”的检测。不需要任何中间步骤，直接end-to-end的训练和优化。

结果模型是一个轻量级的神经网络，与以前的所有算法比，性能大大提高，速度大大提升。

METHODOLOGY

关键组件：神经网络模型，一个训练了用来直接从全图信息，来预测文字存在性和位置的模型。此模式是全卷积的，适用于文字检测的，输出了对文字位置的像素级的密集预测。没有中间商赚差价。后续处理就是一个阈值选点和NMS来预测文字区域的几何位置。我们将这个检测器命名为：EAST.因为它是一个Efficient and Accuracy Scene Text 检测流程。

3.1 算法流程

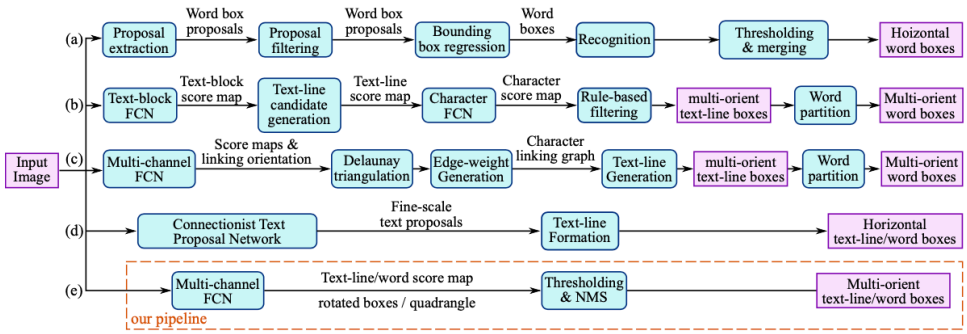

Figure 2. Comparison of pipelines of several recent works on scene text detection: (a) Horizontal word detection and recognition pipeline proposed by Jaderberg *et al.* [12]; (b) Multi-orient text detection pipeline proposed by Zhang *et al.* [48]; (c) Multi-orient text detection pipeline proposed by Yao *et al.* [41]; (d) Horizontal text detection using CTPN, proposed by Tian *et al.* [34]; (e) Our pipeline, which eliminates most intermediate steps, consists of only two stages and is much simpler than previous solutions.

图2(e)是我们的算法流程，遵从DenseBox【9】设计规则，将图片输入到一个FCN中，输出得到多通道的像素级的score map和geometry map。

一个输出的通道是score map,它每一个像素的值属于(0,1),其它几个通道综合在一起，代表了一个几何区域，这个区域把文字包围起来。这几个通道的值是基于像素的位置来表示的。同一个像素位置的score代表了这个几何区域的置信度。

我们试验了两种几何形状来表示文字区域，ratated box(RBOX) 和 quadrangle(QUAD),他们会对应不同的loss functions.Thresholding会应用在每个预测区域上。符合我们人工设定阈值的文字区域才会被输入到后面的Non-maximum-suppression.NMS的结果就是我们的最终输出。

3.2 Network Design

设计文字检测的模型，必须考虑几个因素。文字区域大小有别，大文字需要后层特征。小文字需要前层特征。中等大小文字需要中间层特征。

这就要求做文字检测的特征要利用来自不同级别的层。

HyperNet 遇到了同样的问题，但是它融合了巨多的channels的巨大的特征图，这就显著增加了计算量。

解决这个悲剧，我们使用U-shape思路(29),对于特征，我们采取逐步融合的策略，同时保证特征足够小.两者兼顾，我们最能既能得到多层特征，又能保证一个小的计算开销。

图3是我们模型的简略图。

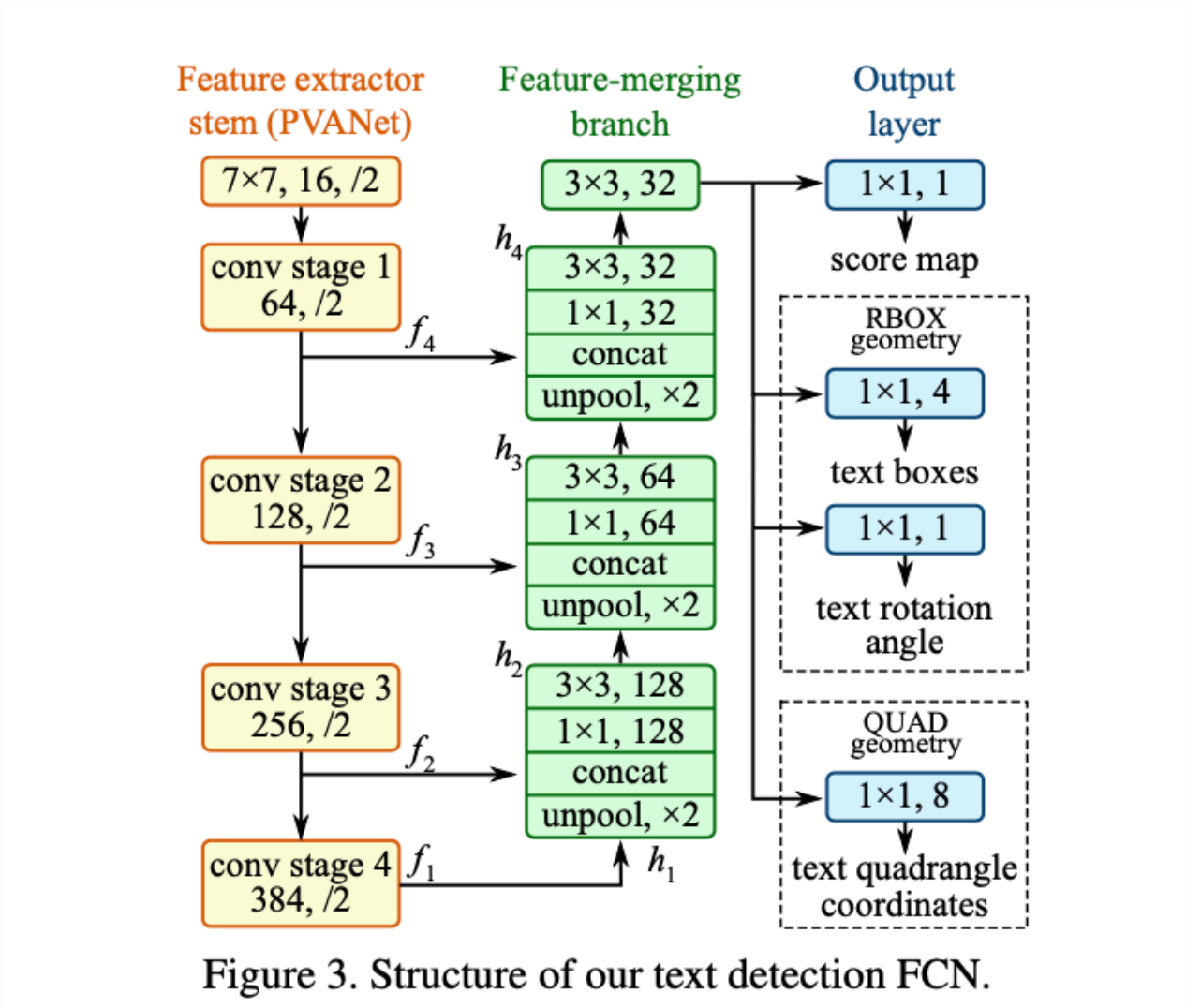

Figure 3. Structure of our text detection FCN.

模型由3部分组成：特征提取stem,特征融合分支和输出层。

stem:可以是基于ImageNet dataset训练的拥有交叉卷积和池化层的卷积网络。从stem抽取四个层的特征图，我们定义为： f_i ，它们的大小分别为输入图片的$\frac{1}{32}$,$\frac{1}{16}$,$\frac{1}{8}$,$\frac{1}{4}$.

图3中，我们使用了PVANet(17)，在我们的实验中，我们使用了众所周知的VGG16(32)模型，我们把它的从pooling-2至pooling-5的层抽取出来了。

后面的特征融合分支，我们按照下面公式逐步融合：

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

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

g_i是融合base,h_i是融合得特征图，操作符号$[;]$代表在channel轴上的连接。在每一个融合阶段，先进行一个unpooling层来倍增其大小，然后再从chahnel轴上与本层特征连接。其次用一个$conv_{1x1}$ bottleneck【8】来减少channels和计算量，最后用一个$conv_{3x3}$来综合处理输出本阶段最终的输出特征图。

这里一共4个阶段。

最后一个阶段的输出经过一个$conv_{3x3}$layer，我们就得到融合分支的最终输出，这个输出就到了我们的output layer.

图3中每个卷积运算的输出channels。我们保证了卷积层的channels少。从而实现了计算量与stem网络相比只是毛毛雨，使我们的网络高效。

最终的输出层包含了一些$conv_{1x1}$操作，把前面的特征从32c映射到1c的score map F_s,还有多通道的geometry map F_g.geometry 输出可以是RBOX或者QUAD的其中之一。如下tab.1

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	$\mathbf{G} = \mathbf{Q} = \{(\Delta x_i, \Delta y_i) \vert i \in \{1,2,3,4\}\}$

Table 1. Output geometry design

对于RBOX,用4个通道表示axis-aligned bounding box(AABB)R,1个channel 表示rotation angle theta.4个通道的值分别是像素位置到R的top, right,botton,left边界的距离。

对于QUAD Q,我们用8个值来表示像素位置到四边形四个顶点（每个顶点两个值）的偏移量。

LABEL GENERATION

3.3.1 生成Quadrangle对应的Score Map

在score map上，正例的区域用原四边形的缩减区域来表示。比如Fig.4(a)

定义一个$\mathbf{Q} = \{p_i | i \in \{1, 2, 3, 4\}\}$,这里$p_i = \{x_i, y_i\}$,是四边形的按照顺时针旋转的四个顶点，

缩减$\mathbf{Q}$,我们首先计算每个顶点p_i 之间距离的参考长度r_i

$$r_i = min(D(p_i, p_{i\ mod\ 4}), D(p_i, p_{((i+2)\ mod\ 4)+1)})$$

$D(p_i, p_j)$ 是 p_i 和 p_j 的L_2距离.

我们先缩短两个长边，然后再缩短两个短边。怎么确定长短边？通过比较它们的平均长度。

对于每一个边$[pi, p(imod4) + 1]$,我们将它的两个点沿着相应的直线分别朝内移动0.3r_i 和 0.3 $r_{(i\ mod\ 4)+1}$

解读：这里每个边上的有两个点p_i,p(i mod 4)+1，对每个点都向内缩小0.3。若r_i是长边，r_(i mod 4)+1就是短边

3.3.2 Geometry Map Generation

网络的输出可以是RBOX或者QUAD其中之一。RBOX生成的方法在图4(c-e)

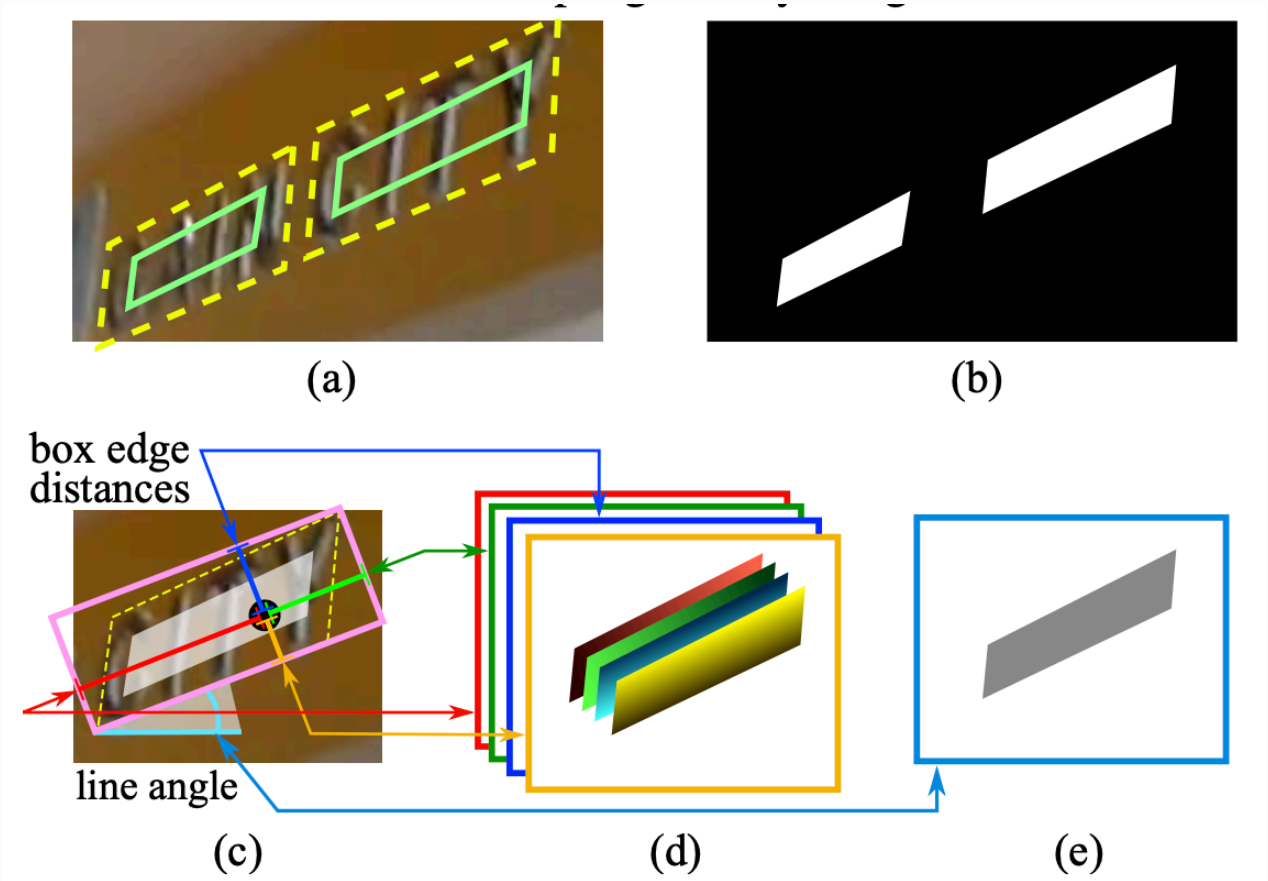

Figure 4. Label generation process: (a) Text quadrangle (yellow dashed) and the shrunk quadrangle (green solid); (b) Text score map; (c) RBOX geometry map generation; (d) 4 channels of distances of each pixel to rectangle boundaries; (e) Rotation angle.

但是一些数据的文字区域是标注的形式为QUAD的（e.g.,ICDAR 2015),QUAD是可以生成ROBX的，我们需要先建立一个最小外接旋转矩形 RBOX。然后对于score为正的像素，我们计算4个边界距离，从而形成ROBX GROUND truth的四个通道图。

QUAD 直接生成QUAD对应的GROUND TRUTH,每个正分的像素，对应8个通道上相应位置的像素，一共是8个像素，像素值是相对于四边形4个顶点的坐标偏移。

3.4. Loss Functions

Loss 公式：

$$L = L_s + \lambda_g L_g$$

L_s 和 L_g 分别表示score map的loss和geometry的loss。λ_g是对两个loss重要性的加权。在我们的实验里，设置$\lambda_g = 1$。

3.4.1 Loss for Score Map

很多soat模型里，训练的图片被精心处理，平衡采样法，硬负样本挖掘法等要被用来解决训练目标分布不均衡的问题。

做这些可以潜在的提高网络性能。

但是用这些技术有个缺点：

1. 引入了一个不可导的阶段
2. 更多的超参数需要去调整
3. 整个流程变得复杂冗余

这些缺点与本文的设计逻辑不符合。

我们用class-balanced cross-entropy 函数，完成了一个操作简单，性能更高的训练过程。

$$\begin{aligned} L_s &= balanced - xent(Y_{pred}, Y_{gt}) \\ &= -\beta Y_{gt} log Y_{pred} - (1-\beta)(1-Y_{gt}) log(1-Y_{pred}) \end{aligned}$$

上式中$Y_{pred} = F_s$是score map,Y_{gt}是ground _truth,β是正负样本的平衡因子，公式为：

$$\beta = 1 - \frac{|Y_{gt}|_+}{|Y_{gt}|}$$

这种平衡交叉熵首次应用在score map的预测中，是在(41)中。我们实验发现确实很好用。

3.4.2 Loss for Geometries

自然场景中的文字的尺寸分布，及其不友好。直接用$L1$或者$L2$来回归显然是不负责任的。因为这会导致我们的loss一直在优化size比较大的文字，就是那些大的或者长的文字。所以这个loss 要求要做到尺寸不变性。因此，对于RBOX，我们使用了IoU loss，对于QUAD,我们使用了一个尺寸归一化后的平滑$L1$ Loss

我们先来看ROBX的IoU loss

这个loss来自论文(46),这个loss显然可以做到尺寸不变性。

$$L_{AABB} = -logIoU(R_{pred}, R_{gt}) = -log\frac{|R_{pred} \cap R_{gt}|}{|R_{pred} \cup R_{gt}|}$$

上式中，$|R_{pred} \cap R_{gt}|$可以这样计算：

$$\begin{aligned} w_i &= min(d_2^{pred}, d_2^{gt}) + min(d_4^{pred}, d_4^{gt}) \\ h_i &= min(d_1^{pred}, d_1^{gt}) + min(d_3^{pred}, d_3^{gt}) \end{aligned}$$

d_1, d_2, d_3, d_4对应像素到上，右，下，左边界的距离。

那么并集的面积是：

$$|R_{pred} \cup R_{gt}| = |R_{pred}| + |R_{gt}| - |R_{pred} \cap R_{gt}|$$

交集和并集的面积都可以很容易训练出来。

旋转角度都loss可以这样计算：

$$L_{theta}(\theta^{pred}, \theta^{gt}) = 1 - cos(\theta^{pred} - \theta^{gt})$$

总的geometry loss就可以写为：

$$L_g = L_{AABB} + \lambda_\theta L_\theta$$

在我们的实验里，λ_θ 设置为了10.

注意到：我们计算L_{AABB}时忽略了角度。所以这个可以看成是近似的IoU.但是在训练中，它仍然可以发挥作用，因为angle loss 和 L_{AABB}可以一起发挥作用，引导网络训练处正确的RBOX

再来看看正确QUAD的loss

利用了smoothed-L1 loss,但是添加了一个归一化的term.

一个四边形Q由4个点组成，可用如下集合表示：

$$C_Q = \{x_1, y_1, x_2, y_2, \ldots, x_4, y_4\}$$

这样 loss函数就可以写为：

$$\begin{aligned} L_g &= L_{QUED}(Q^{pred}, Q^{gt}) \\ &= \min_{Q^- \in P_{Q^{g_t}}} \sum_{c_i \in C_Q, c_i^- \in C_{Q^-}} \frac{smoothed_{L1}(c_i - c_i^-)}{8 \times N_{Q^{pred}}} \end{aligned}$$

里面的归一化项 $N_{Q^{gt}}$ 是四边形的短边，计算公式如下：

$$N_{Q^{gt}} = \min_{i=1}^{4} D(p_i, p_{(i \mod 4)+1})$$

P_Q是一个集合，Q^{gt}中的四个顶点以不通的顺序组成的集合。

这样组合一下是很有必要的，因为不同的数据集标注时，4个顶点的顺序不一致。

3.5 Training

网络端到端的用ADAM算法进行训练。

均匀的从图片里剪切512x512大小的图片组成大小为24的minibatch。加速训练。

学习率策略：从0.001 开始衰减，没27300minibatches，学习率缩小10倍。当学习率缩小到0.00001时,就不再降低。

停止训练的条件：模型性能不再改进。

3.5 Locality-Aware NMS

NMS里，因为有排序，所以复杂度为$O(n^2)$，n对应候选窗的个数。

我们这里密集预测，候选窗有数万，这是个麻烦。

假设：相邻像素的候选窗强相关，我们一行一行的融合这些框。

在一行内融合时，融合结果是可以迭代的。

这样的话，最好的结果是$O(n)$,但是最坏的结果就是$O(n^2)$,

但是因为假设的成功，融合算法在实际中特别高效。

Algorithm 1 Locality-Aware NMS

function NMSLOCALITY($geometries$)
  $S \leftarrow \varnothing,\ p \leftarrow \varnothing$
  for $g \in geometries$ in row first order **do**
    if $p \neq \varnothing \wedge$ SHOULDMERGE(g, p) **then**
      $p \leftarrow$ WEIGHTEDMERGE(g, p)
    else
      if $p \neq \varnothing$ **then**
        $S \leftarrow S \cup \{p\}$
      end if
      $p \leftarrow g$
    end if
  end for
  if $p \neq \varnothing$ **then**
    $S \leftarrow S \cup \{p\}$
  end if
  return STANDARDNMS(S)
end function

$WeightedMerge(g,p)$,用的对两个四边形的分数和几何数据加权平均的思路。

如果$a = WeightedMerge(g,p)$,则$a_i = V(g)g_i + V(p)p_i$,$V(a) = V(g) + V(p)$,

a_i是第i个坐标值,$V(a)$是a的score。

实际上，我们这里是取框的平均而不是像标准NMS那样去挑选一个框，像一个投票机.

这就带来了稳定效应。

尽管本质不同，我们还是用NMS来命名我们的算法：

4. EXPERIMENTS

为了做比较试验，我们在公开基准数据集ICDAR2015,COCO-Text和MSRA-TD500上做了定量和定性分析。

4.1 Benchmark Datasets

ICDAR 2015

COCO-Text

MARA-TD500

4.2. Base Networks

COCO-Text比较大，其它都特别小，因为COCO-Text是来自一般目标检测数据集的。我们的数据集是有大有小的。

如果只用一个模型，同时在三个数据集上做实验，显然会发生过拟合和欠拟合。所以我们用了3个不同的base networks,输出RBOX和输出为QUAD，在所有的数据集上实验了我们提出的算法。base networks如下表:

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

Table 2. Base Models

VGG16 ,常用base networks,也常用于text detection.

但它有两个缺点:

1. 感受野小，在conv5_3层，只有196的感受野。
2. 模型太大，太大。【记忆模型每层结构，总参数量】

PVANET 轻量化网络，Faster-RCNN网络的替代性feature extractor.

PVANET太小了，所以我们把它的 channel 双倍了:PVANET2x.

最后一层的感受野达到809，比VGG的196大太多了。

所有的模型都在ImageNet dataset上预进行了训练。

4.3 Qualitative Results

4.4 Quantitative Results

4.5. Speed Comparison

CONCLUSION AND FUTURE WORK