面向统一网络的快速文本识别

- 译者：李英杰
- FOTS: Fast Oriented Text Spotting with a Unified Network
  - text detect
  - deeplearning
  - object detect
- 时间：2018-05-05
- paper

摘要：

- 在文档分析领域，随机场景文本识别是最难以及最有价值的挑战之一。大多数方法将文本检测和识别划分为两个任务。作者提出一个统一的可端到端训练的FOTS网络，该网络能够同时进行检测和识别，在两个任务之间共享计算结果和视觉信息。特别之处在于，引入RoIRotate在检测和识别之间共享卷积特征。得益于共享卷积策略，FOTS具有较少的计算量相比于文本检测网络基线。并且，联合训练方法能够学习更通用的特征能够使我们的方法表现优于两阶段的训练方法。在ICDAR 2015, ICDAR 2017 MLT, and ICDAR 2013数据集的实验证明，该方法明显优于最先进的方法，朝着开发第一个面向实时文本识别系统更进一步。在ICDAR 2015文本识别数据集上比目前最好的结果高出5%,同时保持了22.6fps。

1.介绍

- 在计算机视觉领域中[49, 43, 53, 44,14, 15, 34]，自然图像中的文本阅读在文档分析，场景理解，机器人导航和图像检索方面的众多实际应用，致使他的关注量不断增加。尽管之前在文本检测和识别方面的成果巨大，但是文本模式的巨大差异和背景的高度复杂对文本识别仍然是个挑战。
- 在场景文本阅读中，最常见的方法是将其划分为文本检测和文本识别来分别处理两个单独的任务 [20, 34]。基于深度学习的方法在这两方面都占据主导的地位。
  - 在文本检测中，通常使用卷积神经网络提取场景图像的特征映射，然后用不同的解码器对区域进行解码[49, 43, 53].
  - 文本识别中，在一个接一个的文本区域之上，网络进行连续的预测。这导致大量的时间成本尤其当图像中有很多文本区域。另外一个问题是，忽略了在检测和识别中共享视觉线索的相关性。单个检测网络不能由文本识别的标签来监督，反之亦然。
- 在本文中，建议同时考虑文本检测和识别。促使了FOTS系统可以被端到端训练。与之前的两阶段文本检测相比，作者通过卷积神经网络学习更通用的特性，在文本检测和识别之间共享该特性，两项任务之间的监督是互补的。因为特征提取通常占用较大的时间，它将计算缩小到单个检测网络。如图1所示。

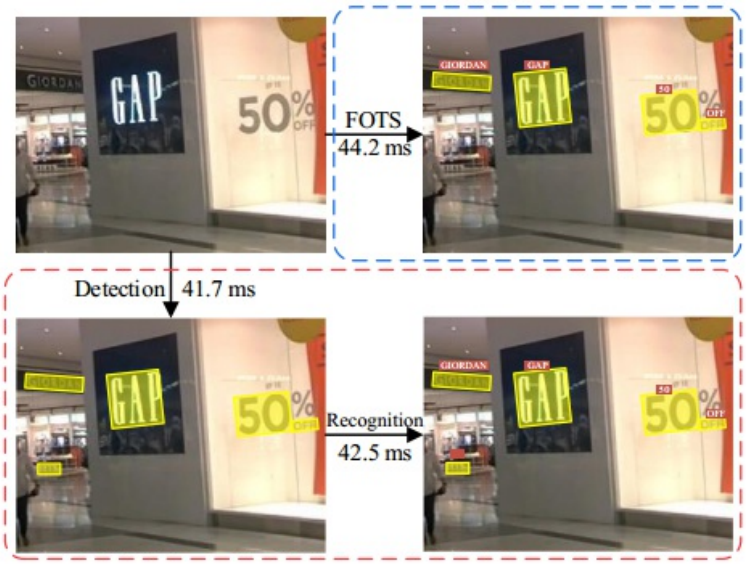

Figure 1: Different to previous two-stage methods, FOTS solves oriented text spotting problem straightforward and efficiently. FOTS can detect and recognize text simultaneously with little computation cost compared to a single text detection network (44.2ms vs. 41.7ms) and almost twice as fast as the two-stage method (44.2ms vs. 84.2ms). This is detailed in Sec. 4.4.

- 连接检测和识别的关键是ROIRotate,它根据面向检测边框从特征映射中得到适当特征。体系结构如图2所示。首先，用共享卷积提取特征图，在特征网络的基础上建立了全卷机网络的面向文本检测的分支，预测了检测边界框。 RoIRotate操作符从feature map中提取与检测结果相对应的文本提议特征。然后，将文本提议特征送入递归神经网络（RNN）编码器和Connectionist Temporal Classification (CTC)中，用于文本识别。由于网络中所有的模块都是可微的，所以整个系统可以进行端到端的训练。这个目前最好的面向文本检测和识别的框架。在没有复杂的后处理和超参数优化的情况下，网络可以很容易的得到训练。

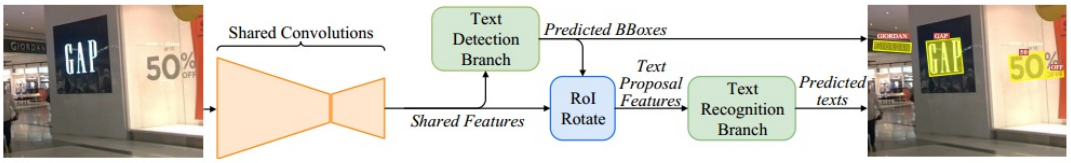

Figure 2: Overall architecture. The network predicts both text regions and text labels in a single forward pass.

- 本文贡献如下：
  - 为快速定向的文本识别提供了一个端到端的可训练框架。通过共享卷积调整，网络可以检测和识别文本，计算开销小，达到实时检测。
  - 引进RoIRotate，该操作能够从卷积调整图中提取相应的区域。将文本检测和识别统一到端到端的通道中。
  - FOTS优于目前最好的方法在大量的文本检测和识别基准数据集包括 ICDAR 2015 [26], ICDAR 2017 MLT [1] and ICDAR 2013 [27]。

相关的工作

2.1 文本检测

- 大多数传统的文本检测方法认为文本是字符组成部分。这些基于字符的方法首先在图像中定位字符，然后将将他们分组到单词或者文本行中。给予滑动窗口的方法 [22, 28, 3, 54] 和基于 connected-components 的方法 [18, 40, 2]是传统方法的两个代表性的类别。
- 近年来，许多基于深度学习的方法直接检测图像中的单词。Tian等[49]采用垂直锚定机制来预测固定宽度的预测方案，然后将其连接起来.MA等[39]通过旋转RPN和旋转ROI池化，为任意定向的文本引入一种新的基于旋转的框架。shi等[43]首先预测文本片段，然后使用链接预测将他们连接到完整的实例中.ZHOU等[53]和HE等[15]提出对多导向场景文本的深度直接回归方法。

2.2 Text Recognition

- 一般的，场景文本识别的目的是解码一个由常规裁剪但长度可变的文本图像组成的标签序列。大多数之前的方法 [8, 30] ，都能捕获单个字符，然后再细化错误分类的字符。除了字符级方法之外，最近的文本区域标识方法可分为三类：

- 基于单词分类
- 基于序列对标签的解码
- 基于序列到序列的模型方法

2.3 Text Spotting

- 大多数之前的文本识别方法首先使用文本检测模型生成文本提议，然后识别。 他们有一个单独的文本识别模型。
- 本文方法的优点：
  - 引入RoIRotate，使用完全不同的文本检测算法来解决更加复杂和困难的情况，之前的方法只适用于水平文本。
  - 本文方法在速度和性能方面都优于之前的方法，特别的，几乎零代价的文本识别分支确保我们的文本识别系统能够有实时的速度，尽管之前的方法对于 600× 800 pixels像素的输如图片具有900ms的处理时间。

方法

- FOTS是一个端到端的可训练框架，能够同时检测和识别自然场景中的所有词。包括4个部分：共享卷积，文本检测分支，RoIRotate操作，文本识别分支。

3.1 整体架构

- 共享网络的体系结构如图所示：

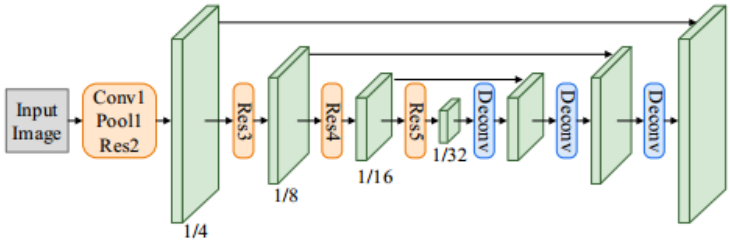

Figure 3: Architecture of shared convolutions. Conv1-Res5 are operations from ResNet-50, and Deconv consists of one convolution to reduce feature channels and one bilinear upsampling operation.

- 将低层特征映射和高级语义特征映射连接起来。共享卷积产生的特征映射的分辨率是输入图像的1/4。
- 文本检测分支使用共享卷积产生的特征，输出文本中每个像素预测。
- 检测分支所产生定向文本区域，RoIRotate将该区域相应的共享卷积特征转化为固定的高度，同时保持原始区域的比例。
- 最后，文本识别分支识别提议区域中的词。采用CNN和LSTM编码文本序列信息，最后是CTC解码器。文本识别分支的结构如表1所示。

Type	Kernel [size, stride]	Out Channels
conv_bn_relu	[3, 1]	64
conv_bn_relu	[3, 1]	64
height-max-pool	[(2, 1), (2, 1)]	64
conv_bn_relu	[3, 1]	128
conv_bn_relu	[3, 1]	128
height-max-pool	[(2, 1), (2, 1)]	128
conv_bn_relu	[3, 1]	256
conv_bn_relu	[3, 1]	256
height-max-pool	[(2, 1), (2, 1)]	256
bi-directional_lstm		256
fully-connected		$\|S\|$

Table 1: The detailed structure of the text recognition branch. All convolutions are followed by batch normalization and ReLU activation. Note that height-max-pool aims to reduce feature dimension along height axis only.

3.2 文本检测分支

- 采用全卷积网络作为文本检测器,由于在自然场景图像中有很多小文本框，所以作者在共享卷积中对原始输入图像的特征映射从1/32调整1/4的大小。
- 提取共享特征后，每个卷积输出单词中密集的单个像素预测。
    - 第一个通道计算每个像素为正样本的概率。原始文本区域缩小版的像素被认为是积极（正）的。
    - 对于每一个正样本，以下4个通道预测该像素与包含这个像素的边框（上 下 左 右）的距离。
    - 最后一个通道预测相关边界框的方向。
- 应用阈值和NMS对这些正样本进行最终的检测。

  在实验中，观察到很多类似于文本文本线条的模式很难被分类，例如栅栏和格子等，采用**online hard example mining (OHEM)**来更好的区分这些模式，这也解决了类别不平衡的问题。为ICDAR 2015 数据集提升了2%的F-measure改进。

  检测分支的损失函数由两项构成，文本分类项和边框回归项。

- 文本分类项可看作为向下采样的得分图的像素分类损失。只有缩小版的原始文本区域被认为是积极的区域，当该区域在边界框之间与这个区域被当作是“NOT CARE”，没有造成分类损失。在得分图中通过OHEM选择的积极元素的集合表示为 Ω ,分类损失函数为：

$$
\begin{aligned}
L_{\text{cls}} &= \frac{1}{|\Omega|}\sum_{x\in\Omega}\mathbf{H}(p_x, p_x^*) \\
&= \frac{1}{|\Omega|}\sum_{x\in\Omega}(-p_x^*\log p_x - (1-p_x^*)\log(1-p_x))
\end{aligned} \tag{1}
$$

    - "| . |"表示集合中元素的数量
    - H(px, p∗x)表示交叉熵损失。px是预测的的得分图，p∗x是表示是否为文本的二元标签。
- 回归损失，采用 IoU 损失和旋转角度损失。他们对目标的形状，尺寸和方向的变化都很稳健：

$$
L_{\text{reg}} = \frac{1}{|\Omega|}\sum_{x\in\Omega}\text{IoU}(\mathbf{R}_x, \mathbf{R}_x^*) + \lambda_\theta(1-\cos(\theta_x, \theta_x^*)) \tag{2}
$$

    - IoU(Rx, R∗x) 是预测框 Rx,和真实框R∗x之间的IoU损失.
    - 第二项是旋转角度损失， θx 和 θ∗x 分别代表预测方向和真实方向。
    - 在实验中，将超参数 λθ 设置为10 。
- 最后的检测损失函数为：

$$
L_{\text{detect}} = L_{\text{cls}} + \lambda_{\text{reg}} L_{\text{reg}} \tag{3}
$$

  超参数 λreg 用来平衡两者的损失，实验中设置为1.

3.3 RoIRotate

- 应用RoIRotate转换定向的特征区域为与坐标轴平行的特征图，如图4所示：

Figure 4: Illustration of RoIRotate. Here we use the input image to illustrate text locations, but it is actually operated on feature maps in the network. Best view in color.

固定输出高度，保持长宽比不变，以处理文本长度的变化。 与ROI池化和ROI Align相比，RoIRotate提供了一个更加通用的操作，提取ROI区域的

特征。 对比RRPN中提出的RROI池化，RROI池化通过max-pooling将一个旋转的区域转化为一个固定尺寸的区域，**本文使用（bilinear interpolation ）双线性插值计算输出值。**这个操作避免了ROI和提取的特征之间的不一致，使得输出特征的变量的长度更适合于文本识别。

- 该过程可以被划分为两步。
  - 1.通过对文本区域的预测或者真实框的坐标来计算仿射变换参数
  - 2.将仿射变换应用于各个区域的共享特征图，得到文本区域的规范水平特征图。 第一步表示为：

$$t_x = l * \cos\theta - t * \sin\theta - x \tag{4}$$

$$t_y = t * \cos\theta + l * \sin\theta - y \tag{5}$$

$$s = \frac{h_t}{t+b} \tag{6}$$

$$w_t = s * (l + r) \tag{7}$$

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= s \begin{bmatrix} \cos\theta & -\sin\theta & t_x\cos\theta - t_y\sin\theta \\ \sin\theta & \cos\theta & t_x\sin\theta + t_y\cos\theta \\ 0 & 0 & \frac{1}{s} \end{bmatrix} \tag{8}$$

  - M为仿射变换矩阵
  - ht（本文设置为8）, wt 经过仿射变换后的高和宽。
  - (x, y)为共享特征图中点的坐标
  - (t, b, l, r)代表该点至 top, bottom, left, right边的距离， θ 为区域的方向。通过真实框或者检测分支能够给出(t, b, l, r)和 θ 的值。 得出放射参数，通过仿射变换能够很容易得出最终的ROI特征。

$$\begin{pmatrix} x_i^s \\ y_i^s \\ 1 \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{9}$$

and for $\forall i \in [1 \dots h_t], \forall j \in [1 \dots w_t], \forall c \in [1 \dots C]$,

$$V_{ij}^c = \sum_n^{h_s} \sum_m^{w_s} U_{nm}^c k(x_{ij}^s - m; \Phi_x) k(y_{ij}^s - n; \Phi_y) \tag{10}$$

where V_{ij}^c is the output value at location (i, j) in channel c and U_{nm}^c is the input value at location (n, m) in channel c. h_s, w_s represent the height and width of the input, and Φ_x, Φ_y are the parameters of a generic sampling kernel $k()$,

公式定义了插值法，特别是双线性插值法，由于文本提议的宽度可能会有所不同，在实践中，将特征图填充为最长的宽度，忽略了识别损失函数中的识别部分。

Spatial transformer network [21] 以相似的方式使用仿射变换，通过不同的方法得到变换参数，主要用于图像领域。RoIRotate将共享卷积产生的特征图作为输入，生成所有文本提议的特征图，具有固定的高度和不变的长宽比。

对于不同的分类目标，文本识别对检测噪声非常敏感。在预测文本区域中出现的一个小的错误可以切断几个字符，对网络训练不利，在训练中使用真实文本区域代替预测文本区域。After RoIRotate，经过转换的特征图进入文本识别分支。

3.4 文本识别分支

- 文本识别分支的目标是，利用共享的卷积提取并通过RoIRotate进行转换的区域特征来预测文本标签。考虑到文本区域中标签序列的长度，LSTM的输入特征沿着高度轴减少2次，即该输入特征高度是原始区域的共享卷积高度的1/4（**作者在这里是宽度减少，笔者认为有误！**）。在密集文本区域中的一部分可区别特征，特别是那些形状较窄的字符，将要被裁剪。该文本识别的分支包括VGG-like[47]中的连续卷积结构，只沿着高度轴的池化，一个双向的LSTM [42, 16],一个全连接和最终的CTC解码器[9]。
  - 首先，空间特征输入若干的卷积操作和沿着高度轴降维的池化操作，该操作提取更高层次的特征。为了简单表述，所有的报告结果基于VGG-

like连续层。如上图表1所示。

- 其次，所提取的高层次的特征图 L∈R^{C× H× W} 按照以时间为主要的排序方式形成序列 l1, ..., lW ∈ R^{C× H}，之后输入RNN进行编码。在这里，作者使用双向LSTM（每个方向有256个输出通道）来获取输入序列特征的依赖范围。

- 然后，在这两个方向中的每一个时间步长计算隐藏状态 h1, ..., hW ∈R^{D}，将每个时间步长的隐藏状态求和送入全连接层，在字符类别集合空间S中给每个状态分配 xt ∈ R^{|S|}。为了避免在例如ICDAR 2015之类的小的训练集上过拟合，在全连接之前加入Dropout机制。
- 最后，CTC将以上架构（经过全连接处理）中的分类得分转化为标签序列。给出在字符集S中每个ht的可能分配xt。真实的标签序列y* = {y1, ..., yT}, T<=W,关于标签y*的条件概率是所有可能状态的和：

$\boldsymbol{y}^* = \{y_1, ..., y_T\}, T \leqslant W$, the conditional probability of the label $\boldsymbol{y}^*$ is the sum of probabilities of all paths π agreeing with [9]:

$$p(\boldsymbol{y}^*|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\boldsymbol{y}^*)} p(\pi|\boldsymbol{x}) \tag{11}$$

where $\mathcal{B}$ defines a many-to-one map from the set of possible labellings with blanks and repeated labels to $\boldsymbol{y}^*$. The training process attempts to maximize the log likelihood of summation of Eq. (11) over the whole training set. Following [9], the recognition loss can be formulated as:

$$L_{\text{recog}} = -\frac{1}{N} \sum_{n=1}^{N} \log p(\boldsymbol{y}_n^*|\boldsymbol{x}) \tag{12}$$

where N is the number of text regions in an input image, and $\boldsymbol{y}_n^*$ is the recognition label.

Combined with detection loss L_{detect} in Eq. (3), the full multi-task loss function is:

$$L = L_{\text{detect}} + \lambda_{\text{recog}} L_{\text{recog}} \tag{13}$$

where a hyper-parameter λ_{recog} controls the trade-off between two losses. λ_{recog} is set to 1 in our experiments.

3.5 实验细节

- 在ImageNet dataset[29]上对模型进行预训练。训练过程包含两步：

  - 1.在Synth800k dataset [10] 对网络训练10 epochs
  - 2.用real data fine-tune 模型知道收敛为止。 在不同的训练任务中采用不同的训练数据集，在in Sec.4 将重点讨论。 在ICDAR 2015 and ICDAR2017 MLT 数据集中中的一些模糊文本区域被标记为“DO NOT CARE”, 在训练中忽视他们。
- 对深度神经网络来说，数据增加很重要。尤其当真实数据的数量有限。

- 在本文中:

  - 1. 图像中较长边的尺寸从640 pixels 调整至 2560 pixels
  - 2.图像在 [−10◦, 10◦] 的范围内随意的旋转。
  - 3.保持图像的宽度不变，调整高度的相对比率从0.8 到 1.2.
  - 4.从变换的图片中随机抽样并裁剪为640×640大小。
- 如 Sec. 3.2所述，采用OHEM来提高性能。对于每张图片，挑选512个 hard negative samples，512个 random negative samples 和 all positive samples 用来进行分类。因此，正负比例从1：60上升到1：3。对于边框回归，从每张图片中挑选128 hard positive samples 和128 random positive samples进行训练。

- 在测试阶段，在从文本预测分支中获取文本区域之后，RoIRotate应用阈值筛选和NMS对文本区域进行处理，将处理后的文本区域输入检测分支得到最终的识别结果。对于多尺度测试，合并所有尺度的结果然后进行NMS得到最终的结果。

实验

- 作者在最近三个具有挑战性的公共基准测试集上对方法进行评估，ICDAR 2015 [26], ICDAR 2017 MLT [1] and ICDAR 2013 [27], 在文本定位和识别中刷新了记录。所有使用的训练数据都是公开的。

4.1 基准数据集

- ICDAR 2015
- ICDAR 2017
- ICDAR 2013

4.2 与两阶段检测方法的对比

- 不同于以往将文本检测和识别分为两个无关的任务的工作，本文方法共同训练了这个任务，文本检测和识别能够相互受益。为了验证这一点，作者建立了一个两个阶段的系统，文本检测和识别模型分开训练。检测网络通过移除本文网络中的识别分支而建立的。类似的，检测分支从网络中移除得到识别网络。对于识别网络，从原图像裁剪的文本行区域被用作训练数据，类似于之前的文本识别方法 [44, 14, 37]。
- 如表2，3，4所示，联合训练方式更加优异。

Method	Detection			Method	End-to-End			Word Spotting		
	P	R	F		S	W	G	S	W	G
SegLink [43]	74.74	76.50	75.61	Baseline OpenCV3.0+Tesseract [26]	13.84	12.01	8.01	14.65	12.63	8.43
SSTD [13]	80.23	73.86	76.91	Deep2Text-MO [51, 50, 20]	16.77	16.77	16.77	17.58	17.58	17.58
WordSup [17]	79.33	77.03	78.16	Beam search CUNI+S [26]	22.14	19.80	17.46	23.37	21.07	18.38
RRPN [39]	83.52	77.13	80.20	NJU Text (Version3) [26]	32.63	-	-	34.10	-	-
EAST [53]	83.27	78.33	80.72	StradVision_v1 [26]	33.21	-	-	34.65	-	-
NLPR-CASIA [15]	82	80	81	Stradvision-2 [26]	43.70	-	-	45.87	-	-
R^2CNN [25]	85.62	79.68	82.54	TextProposals+DictNet [7, 19]	53.30	49.61	47.18	56.00	52.26	49.73
CCFLAB_FTSN [4]	88.65	80.07	84.14	HUST_MCLAB [43, 44]	67.86	-	-	70.57	-	-
Our Detection	88.84	82.04	85.31	Our Two-Stage	77.11	74.54	58.36	80.38	77.66	58.19
FOTS	91.0	85.17	87.99	FOTS	81.09	75.90	60.80	84.68	79.32	63.29
FOTS RT	85.95	79.83	82.78	FOTS RT	73.45	66.31	51.40	76.74	69.23	53.50
FOTS MS	**91.85**	**87.92**	**89.84**	FOTS MS	**83.55**	**79.11**	**65.33**	**87.01**	**82.39**	**67.97**

Table 2: Comparison with other results on ICDAR 2015 with percentage scores. "FOTS MS" represents multi-scale testing and "FOTS RT" represents our real-time version, which will be discussed in Sec. 4.4. "End-to-End" and "Word Spotting" are two types of evaluation protocols for text spotting. "P", "R", "F" represent "Precision", "Recall", "F-measure" respectively and "S", "W", "G" represent F-measure using "Strong", "Weak", "Generic" lexicon respectively.

Method	Precision	Recall	F-measure
linkage-ER-Flow [1]	44.48	25.59	32.49
TH-DL [1]	67.75	34.78	45.97
TDN_SJTU2017 [1]	64.27	47.13	54.38
SARI_FDU_RRPN_v1 [39]	71.17	55.50	62.37
SCUT_DLVClab1 [1]	80.28	54.54	64.96
Our Detection	79.48	57.45	66.69
FOTS	80.95	57.51	67.25
FOTS MS	**81.86**	**62.30**	**70.75**

Table 3: Comparison with other results on ICDAR 2017 MLT scene text detection task.

Method	Detection		Method	End-to-End			Word Spotting		
	IC13	DetEval		S	W	G	S	W	G
TextBoxes [34]	85	86	NJU Text (Version3) [27]	74.42	-	-	77.89	-	-
CTPN [49]	82.15	87.69	StradVision-1 [27]	81.28	78.51	67.15	85.82	82.84	70.19
R^2CNN [25]	79.68	87.73	Deep2Text II+ [51, 20]	81.81	79.47	76.99	84.84	83.43	78.90
NLPR-CASIA [15]	86	-	VGGMaxBBNet(055) [20, 19]	86.35	-	-	90.49	-	76
SSTD [13]	87	88	FCRNall+multi-filt [10]	-	-	-	-	-	84.7
WordSup [17]	-	90.34	Adelaide_ConvLSTMs [32]	87.19	86.39	80.12	91.39	90.16	82.91
RRPN [39]	-	91	TextBoxes [34]	91.57	89.65	83.89	93.90	91.95	85.92
Jiang *et al.* [24]	89.54	91.85	Li *et al.* [33]	91.08	89.81	84.59	94.16	92.42	**88.20**
Our Detection	86.96	87.32	Our Two-Stage	87.84	86.96	80.79	91.70	90.68	82.97
FOTS	88.23	88.30	FOTS	88.81	87.11	80.81	92.73	90.72	83.51
FOTS MS	**92.50**	**92.82**	FOTS MS	**91.99**	**90.11**	**84.77**	**95.94**	**93.90**	87.76

Table 4: Comparison with other results on ICDAR 2013. "IC03" and "DetEval" represent F-measure under ICDAR 2013 evaluation and DetEval evaluation respectively.

- 因为文本识别监督有助于网络学习字符级的特征，所以FOTS在检测中表现好。为了更加详细的分析，列举文本检测中4个常见的问题：

    - Miss: 遗漏一些文本区域
    - False:将一些非文本区域错误的作为文本区域
    - Split:将一些文本区域错误的拆分为几个单独的部分
    - Merge:将一些独立的文本区域错误的合并为一个整体
- 如图5所示，与“our detection”相比，FOTS明显减少了这4类错误，具体来说，“our detection”检测方法关注的是整个文本特征而不是字符级别的特征， 当文本区域有非常大的方差或者文本区域与背景相似等，这将导致该方法不能表现最优。

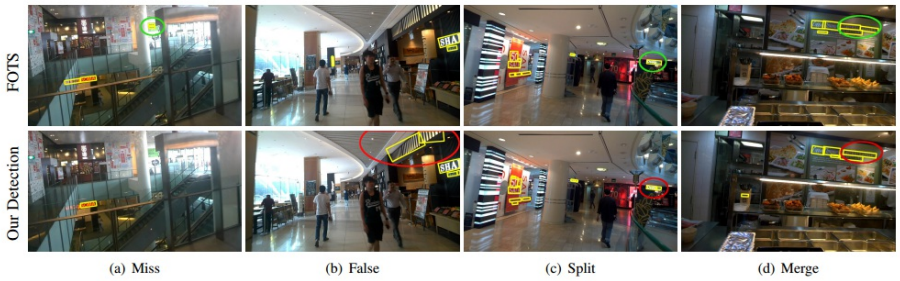

Figure 5: FOTS reduces Miss, False, Split and Merge errors in detection. Bounding boxes in green ellipses represent correct text regions detected by FOTS, and those in red ellipses represent wrong text regions detected by “Our Detection” method. Best view in color.

- 由于有监督的文本识别能够促使模型学习字符更通用的细节，FOTS能够学习具有不同模式的单词中的不同字符间的语义级信息。同时也增强了具有相似模式的字符和背景之间的差异。

4.3 比较最先进的结果

- 如表2,3,4所示，与最新的方法进行比较结果都更优。
    - ICDAR 2017 MLT中没有文本识别任务，只有文本检测的结果
    - ICDAR 2013 所有的文本区域都用水平边框标记，而其中的许多区域略有倾斜。由于模型使用ICDAR 2017 MLT数据进预训练，它可以预测文本区域的方向。
    - 最终的文本识别结果保持了预测的方向，以获得更好的性能，由于评估协议的局限性，检测结果是网络预测的最小水平限定矩形。
    - 此外，在ICDAR 2015 的文本识别的任务中，比目前最好的方法[43, 44] 在F-measure指标方面提升了15%。
        - 在单尺度测试中，FOTS将来自于ICDAR2015, ICDAR 2017 MLT and ICDAR 2013数据集的输入图像的较长边分别调整为 2240, 1280, 920 以得到最好的结果
        - 对多尺度测试应用3-5个尺寸

4.4 速度和模型大小

- 如表5所示：
    - 速度方面：检测和识别的统一训练的网络结构是两阶段训练结构的2倍
    - 在ICDAR 2015 and ICDAR 2013 测试集上测试所有的方法，这些测试集有68个文本识别标签，评估所有的测试图像并且计算平均速度。
        - 在ICDAR 2015上，FOTS使用2240× 1260作为输入，“Our Two-Stage”使用2240× 1260作为检测和32像素高度的裁剪文本区域进行识别。
        - 在ICDAR 2013，将输入图片大小调整为920，使用32像素高度图像块进行识别。
    - 为了达到实时速度，“FOTS RT”（ResNet-34）代替ResNet-50并使用1280× 720 的图像作为输入。
    - 表5中的所有结果在一个改进的Caffe [23]版本中测试得出。
    - 使用了一个 TITAN-Xp GPU

5 结论

- FOTS是面向场景文本识别的端到端可训练框架。
- 提出一个新的RoIRotate操作，将检测和识别统一到一个端到端通道中。
- 通过共享卷积，文本识别的步骤几乎是零代价，使得系统能够实时运行。