场景：做一个通用型的多种证件解析服务

需求：调研一种又新又快的定位模型。要求：

1）支持倾斜的文字，可以是4点定位或分割法后获取box，但不能是2点的定位；

2）快速，过往的psenet需要至少0.6s，pan和db在一些场景中效果差一点但快，是否有更好平衡速度和效果的方法；

3）方便改输出通道数量，这种一般是分割？；

4）边缘准确；

5）适用于中文大字典

方法：从3个方法研究，pp、mmocr、papers with code

## PP

https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.6/doc/doc\_ch/PP-OCRv3\_introduction.md

PP-OCRv3检测模块仍基于DB算法优化，而识别模块不再采用CRNN，换成了IJCAI 2022最新收录的文本识别算法SVTR

## mmocr

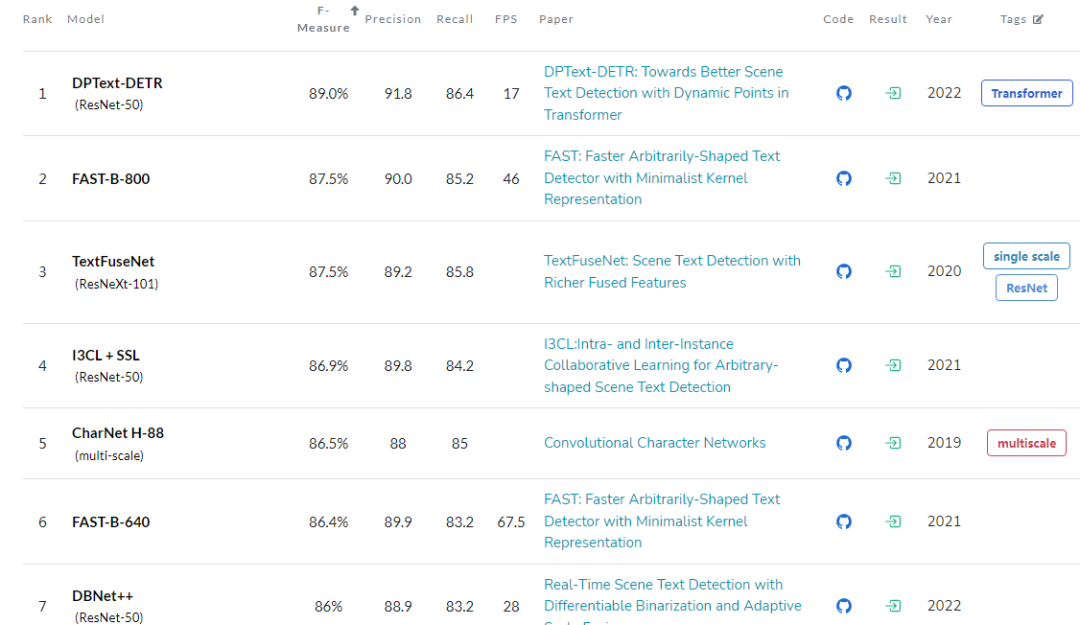
https://gitee.com/open-mmlab/mmocr#%E6%A8%A1%E5%9E%8B%E5%BA%93

最新的检测识别是dbnet++、SVTR

## papers with code

从ppocr和mmocr看到，2者选择的方案都是dbnet++及SVTR，dbnet++是2022年的算法。在paperswithcode中，不是特别高的排名，有个疑问，为啥最后会选用这个呢？出于速度考虑？？？

https://paperswithcode.com/paper/real-time-scene-text-detection-with-1



目前考虑的方法

DPText-DETR，排名最高，后续试了效果还不错，感动，哪怕是有600多切片的图片，速度也在0.1-0.2s

FAST-B-800，排名2

DBNet++ 商业化落地最多的

排除的方法

TextFuseNet不适合中文

I3CL + SSL 没相关资料文章，怕有坑，本身也不是最好的2个

CharNet H-88 不适合中文

## N种排名靠前的算法

DPText-DETR 京东探索研究院

时间：2022.07

开源：https://github.com/ViTAE-Transformer/DeepSolo

paper：DPText-DETR: Towards Better Scene Text Detection with Dynamic Points in Transformer

是否支持曲线文本：是

速度：

解读：https://blog.csdn.net/moxibingdao/article/details/128910689

https://zhuanlan.zhihu.com/p/569496186

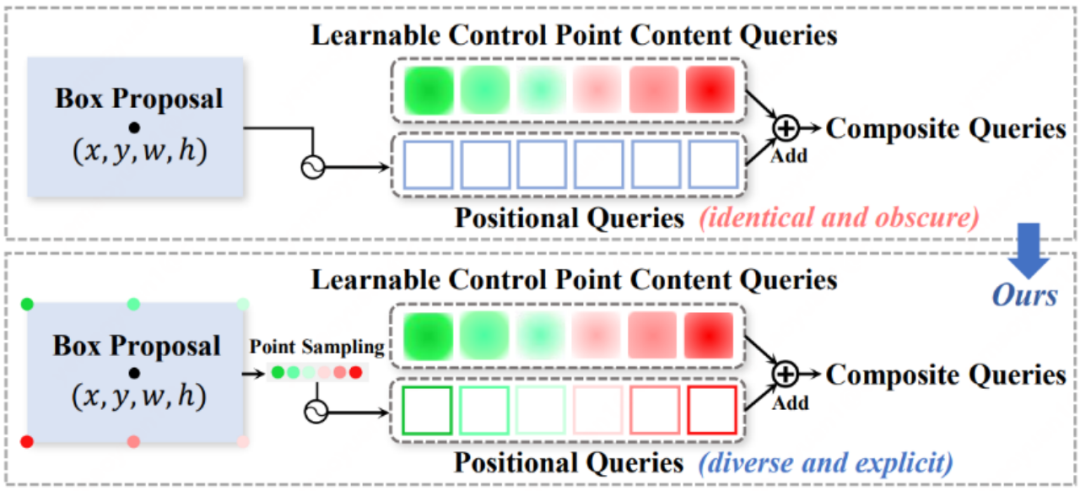
原先算法的缺陷：

1）用xywh表示位置先验信息导致训练慢

2）基于阅读顺序的标注方法（即文本开始循环一圈）降低了模型性能。

本文创新点：

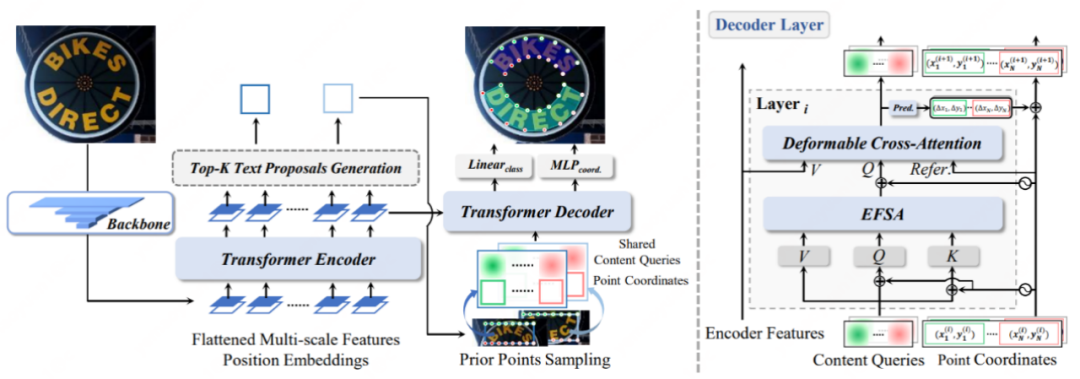
显式点query构建(Explicit Point Query Modeling, EPQM) 方法，用上下边界多点均匀采样得到的N点代替xywh的box，显示细化的位置先验信息有助于加速收敛。即用固定数量控制点代替检测框



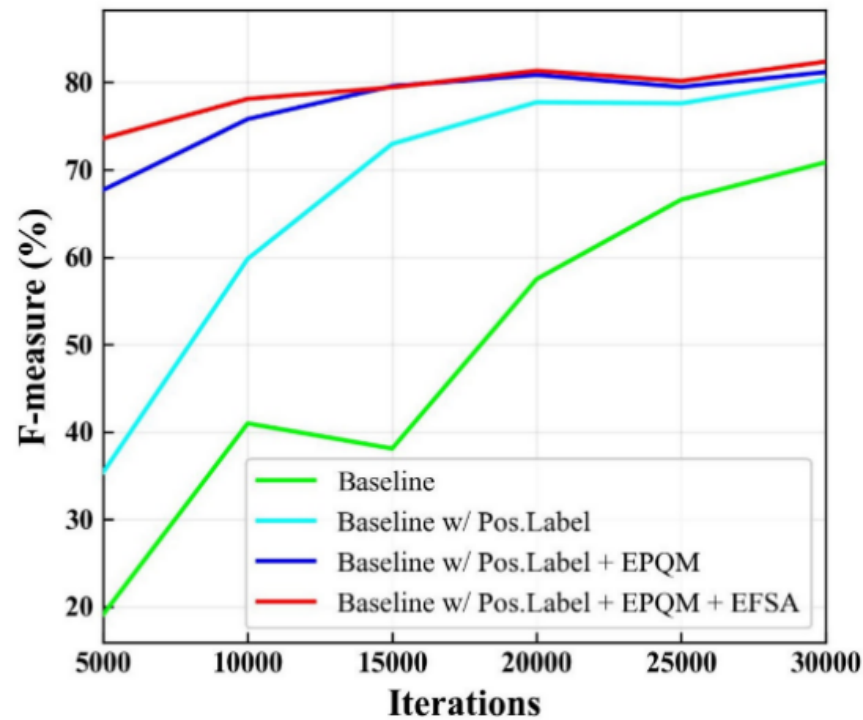
2）增强的因子化自注意(Enhanced Factorized Self-Attention, EFSA) 模块，挖掘同一文本实例内

控制点query之间的关系。引入了环形卷积与实例内自注意力并行以提供显式的环形引导，明确地模拟多边形点序列的圆形，引入更多的先验以充分挖掘实例内不同控制点query的关系。增强的实例内关系建模与实例间关系建模共同构成了EFSA模块。

3）不依赖文本阅读顺序的标签形式

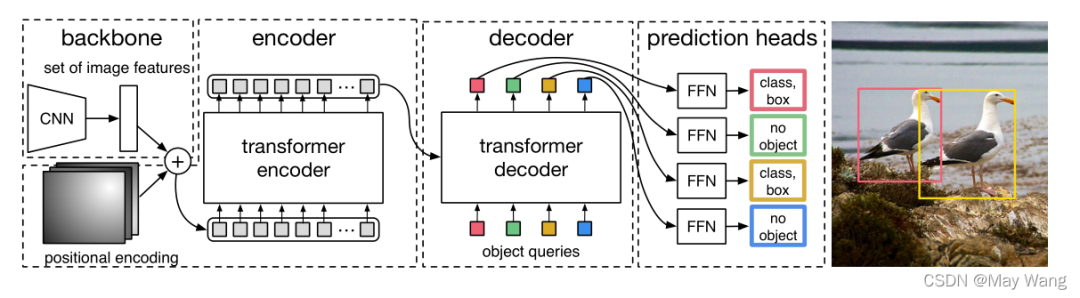


消融实验如下，可以看到不同新模块对速度及精度的影响



基于DEtection TRansformer (DETR) 框架，detr是2020年facebook基于vision transformer做的目标检测工作，没有了anchor和box操作。架构如下图：

https://blog.csdn.net/qq\_35831906/article/details/124118569



FAST-B-800

时间：2021

开源：是

paper：FAST: Faster Arbitrarily-Shaped Text Detector with Minimalist [Kernel](https://cloud.tencent.com/developer/techpedia/1992?from_column=20065&from=20065) Representation

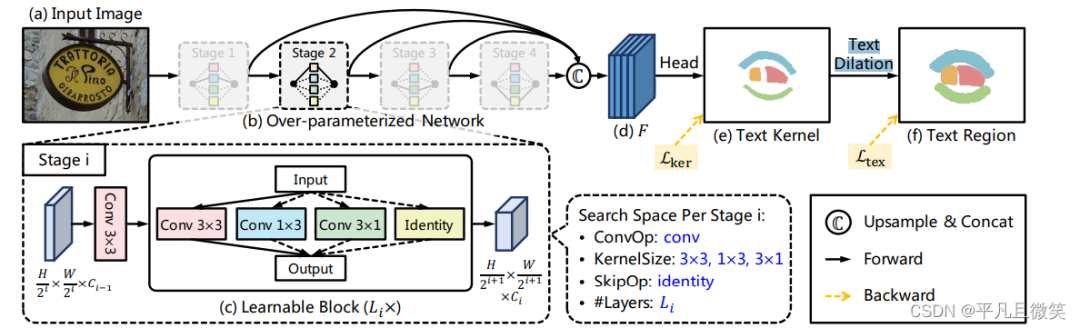
是否支持曲线文本：是

速度：快

NAS网络搜

backbone+类似残差结构+预测层(head)+Text dilation

https://blog.csdn.net/qq\_44498420/article/details/125593141



TextFuseNet

时间：2020

开源：

paper：TextFuseNet: Scene Text Detection with Richer Fused Features

是否支持曲线文本：是

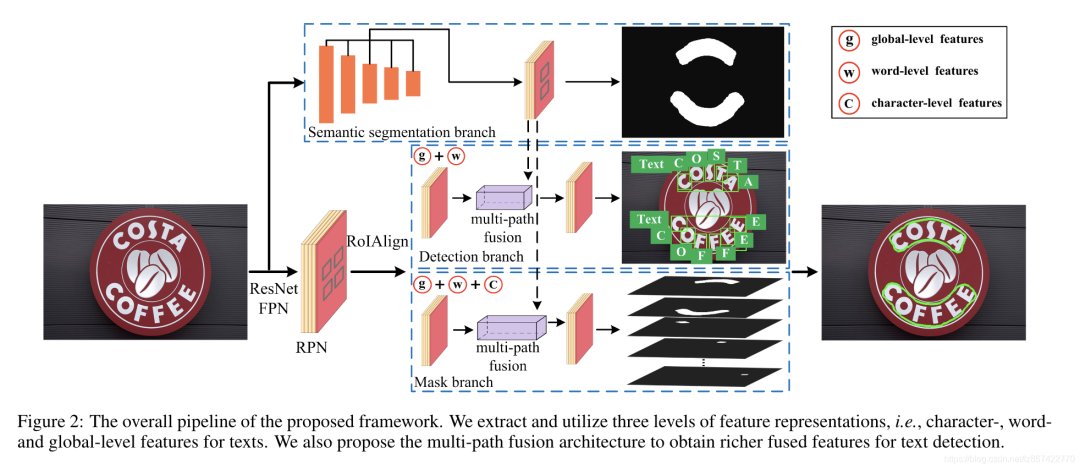
是否适合中文：否

速度：

解读：https://blog.csdn.net/lz867422770/article/details/10917027

参考maskTextspotter和maskRCNN的思想，做实例分割。框架如下图，提取字符，单词和全局级别的特征，并引入多路径融合体系结构以融合它们以进行准确的文本检测。

maskTextspotter是单词级别的检测分割，不适合中文场景。



**CharNet H-88**

时间：2019

开源：是

paper：Convolutional Character Network

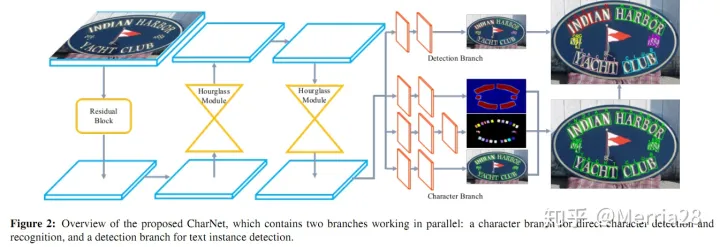
是否支持曲线文本：是

是否适合中文：否

速度：

https://zhuanlan.zhihu.com/p/90683589

在字符级annotation的基础上完成了文本检测和识别的one-stage网络。又是字符级，做中文不考虑。



**FAST-B-640**

比800弱一点

**DBNet++**

时间：2022

开源：是

paper：

Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion

是否支持曲线文本：

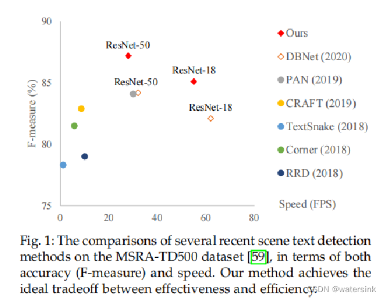
速度：

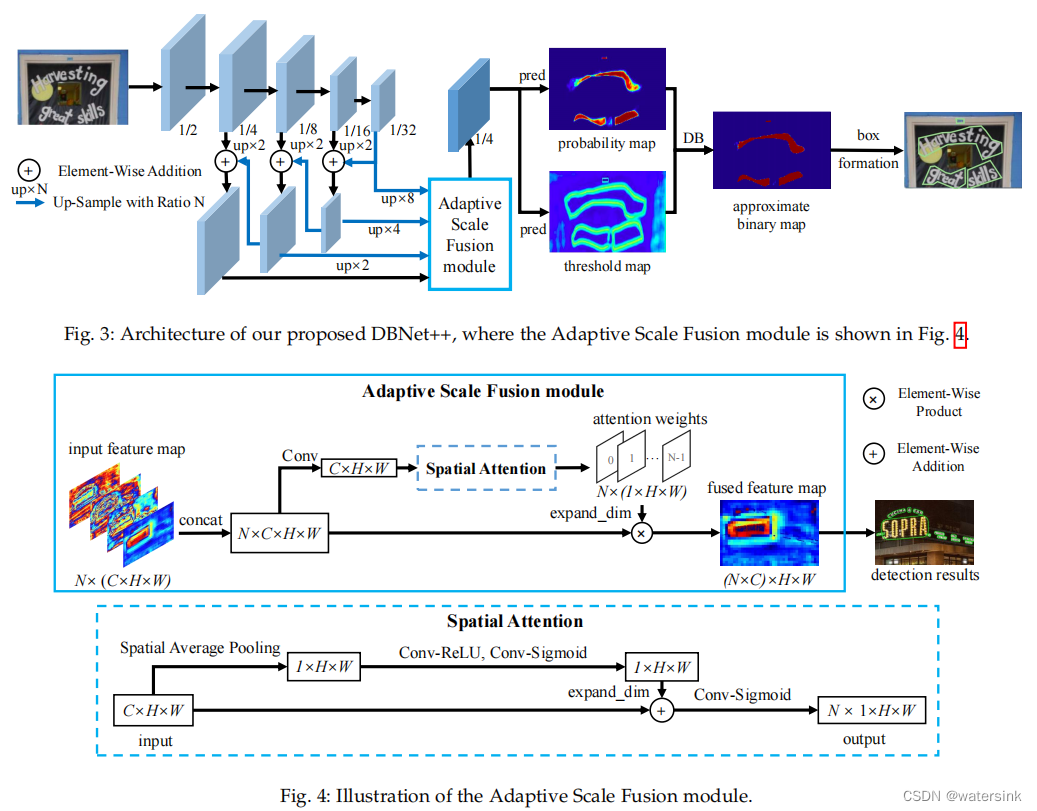
https://blog.csdn.net/qq\_14845119/article/details/127103550

DBNet：Real-time Scene Text Detection with Differentiable BinarizationReal-time Scene Text Detection with Differentiable Binarization

DBNet++：Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion

如下图，横轴是FPS，纵轴是F，db++牺牲了一点速度获得了性能的提升。



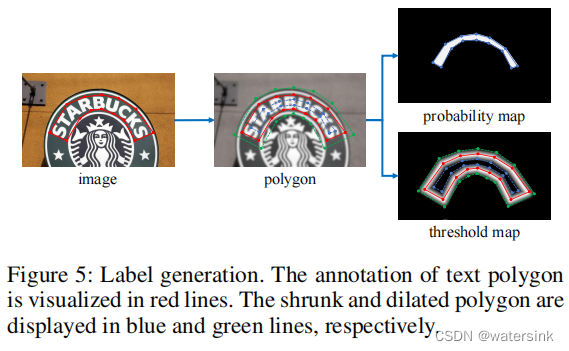


输出是概率图（probability map）和阈值图（threshold map）

DBNet++在DBNet的基础上增加了ASF（Adaptive Scale Fusion）模块。不同尺度的特征通过ASF模块处理，可以得到更佳的融合特征。ASF模块通过引入空间attention机制，使得融合后的特征更加鲁棒。

其中N表示要融合的特征数，这里N=4，表示从4个不同的分支引出的特征。

db差异二值化的作用：每个像素都使用不同的阈值进行二值化处理。而这个不同的阈值矩阵又是网络学习得到的。为了保证整个优化过程有梯度的传递，这里又将概率图和阈值图的差传入sigmoid函数，以此来保证梯度的传递。通过梯度优化，保证了不同的图片使用不同的阈值矩阵，达到最佳的二值化效果。如何处理相邻较近的文本：为了增大相邻文字之间的间距，缓解文字离得太近或者部分重叠的情况。概率图（probability map）的制作会在原始红色多边形的基础上，使用Vatti clipping算法，向内收缩D的距离。

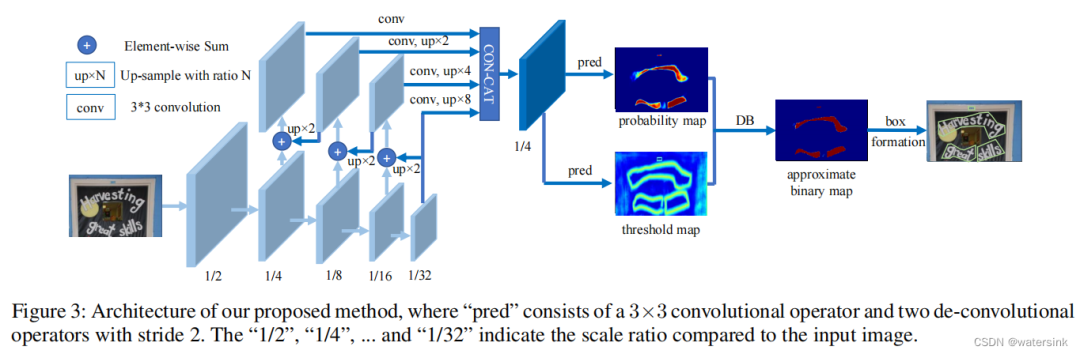


阈值图（threshold map）在红色多边形的基础上，分别向内收缩D距离形成蓝色多边形，向外扩张D距离形成绿色多边形。蓝色多边形和绿色多边形之间的像素形成阈值图。然后计算图内每个像素离最近的边（蓝色边，绿色边）的归一化距离，形成最终的阈值图。阈值图看起来中间像素亮，边缘像素暗。

后处理时，使用概率图（probability map）或者使用二值图（approximate binary map）都是可以的。两者在效果上是一样的。这样在推理过程中，就可以去掉网络中的二值化过程，直接使用概率图。这样网络中的二值化过程的loss就更像一个辅助loss，来使得网络训练的效果更好。

简单来说，后处理时通过概率图获取中心连通域，再用规则缩放边界。

**DBNet**



网络输入假设为w\*h\*3。网络整体结构采用FPN的设计思想，进行了5次下采样，3次上采样操作。最终的输出特征图大小为原图的1/4。网络头部部分，分别引出2个分支。一个负责预测概率图（probability map，(w/4)\*(h/4)\*1），代销为，另一个负责预测阈值图（threshold map，(w/4)\*(h/4)\*1）。概率图经过阈值图处理，进行二值化后得到二值图（approximate binary map，(w/4)\*(h/4)\*1）。最后经过后处理操作得到最终文字的边。