

Decoupled Attention Network for Text Recognition

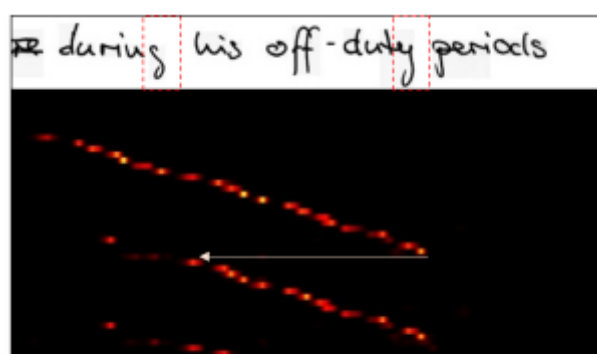
来源：20-AAA1、华南理工

资料：[论文](#)、[代码][<https://github.com/Wang-Tianwei/Decoupled-attention-network>]

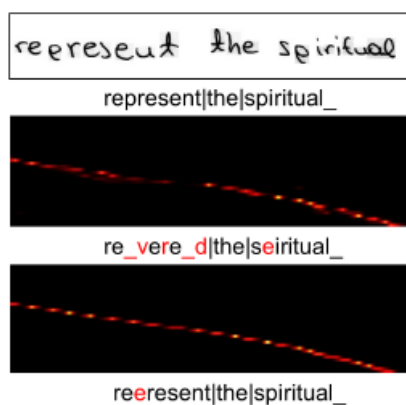
背景和思路

目前端到端的文本识别，主要基于LSTM+CTC和seq2seq+Attention两种框架。CTC的缺点是依赖于独立输出的假设，而Attn方法能更好的利用序列的相关信息，主流研究的较多。

Attn方法主要通过**学习特征权重**来解决字符对齐的问题，但传统的Attn方法需要两个输入：1) encoder的图片特征；2) 历史的解码信息。实践中发现这种Attn方法会有**字符特征对齐不够准确**的问题。



Decoding result:
{...off-dating his...}



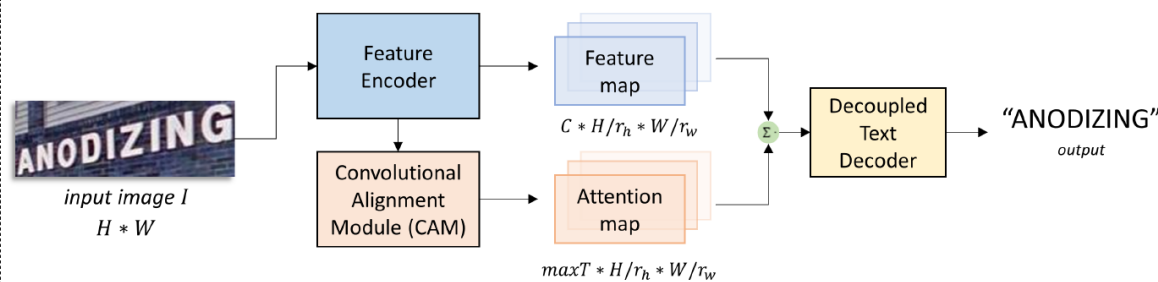
论文认为：传统Attn方法利用历史的预测结果来做对齐操作，这样一旦历史预测结果有错误，误差就很容易在之后的对齐中有**积累效应**，导致越来越对不准。

论文提出的新思路是：将字符对齐和解码预测两个过程解耦，直接拿掉解码阶段的反馈。字符对齐本质上是在做特征匹配，论文从图片特征融合的角度来构建新的注意力方法。

算法关键点

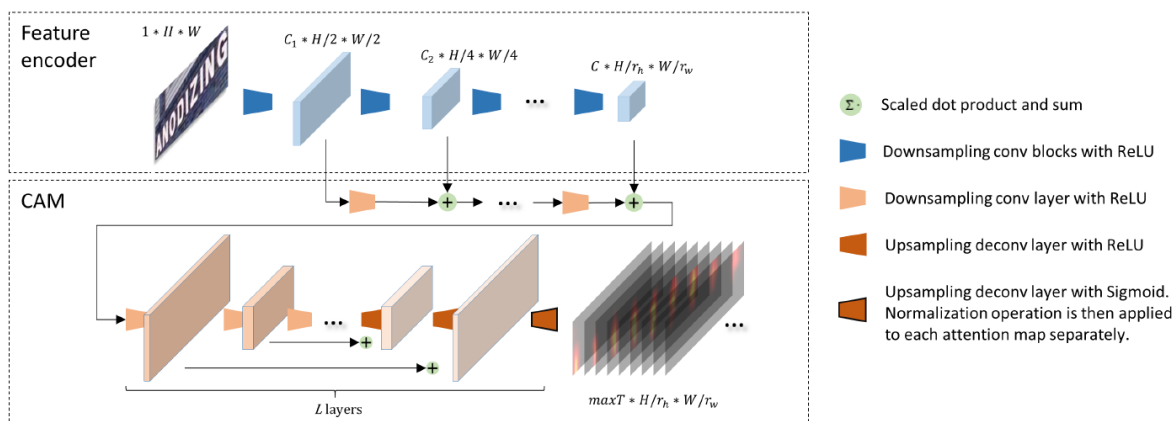
DAN算法主要包括三个模块：**CNN特征提取模块**，**CAM对齐模块**和**RNN解码模块**。CNN计算图片的特征图，CAM也计算一张特征图，两者融合后，送入解码模块。

Overall architecture of DAN



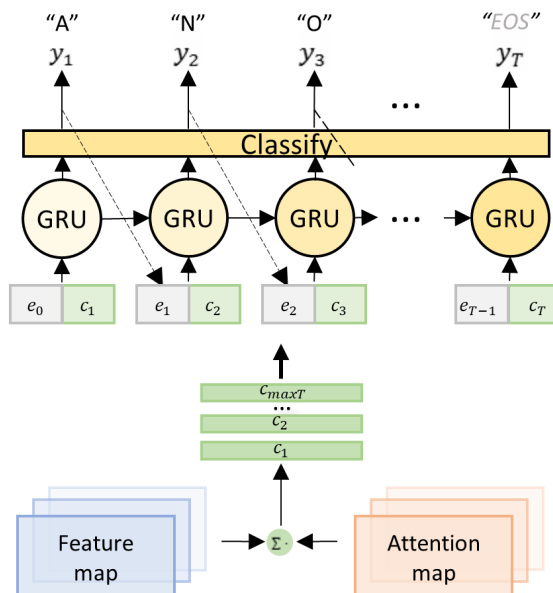
• 对齐模块CAM

借鉴FPN的思路，先对ResNet提取的各层特征图做下采样，再融合在一起得到特征图M；再借鉴FCN的思路，在特征图M上做编解码得到一张注意力特征图 $[T, H/r, W/r]$ 。



• RNN解码模块

接受融合后的特征向量 C ，将 C 送入GRU组成的RNN网络作预测，最后接一层线性层做预测输出。



文本向量 C 的计算（即怎么融合两张特征图，构成一个RNN序列输入）：

$$c_t = \sum_{x=1}^{W/r_w} \sum_{y=1}^{H/r_h} \alpha_{t,x,y} F_{x,y}$$

```
C = feature.view(nB, 1, nC, nH, nW) * A.view(nB, nT, 1, nH, nW) # 扩展操作
C = C.view(nB, nT, nC, -1).sum(3).transpose(1,0) # 求和操作
```

求解GRU的隐含状态；t时刻的预测器输出以及 loss函数：

$$h_t = GRU((e_{t-1}, c_t), h_{t-1}),$$

$$y_t = wh_t + b,$$

$$Loss = - \sum_{t=1}^T \log P(g_t | I, \theta),$$

算法效果

| Methods | Rect | 2D | Regular | | | | Irregular | | |
|----------------------------------|------|----|-------------|-------------|-------------|--------------------------|-------------|-------------|-------------|
| | | | IIIT5k | SVT | IC03 | IC13 | SVT-P | CUTE80 | IC15 |
| (Cheng et al. 2017) ¹ | | | 87.4 | 85.9 | 94.2 | 93.3 | - | - | 70.6 |
| (Cheng et al. 2018) | | | 87.0 | 82.8 | 91.5 | - | 73.0 | 76.8 | 68.2 |
| (Bai et al. 2018) ¹ | | | 88.3 | 87.5 | 94.6 | 94.4 | - | - | 73.9 |
| (Liu et al. 2018) | | | 89.4 | 87.1 | 94.7 | 94.0 | 73.9 | 62.5 | - |
| (Shi et al. 2018) | ✓ | | 93.4 | 89.5 | 94.5 | 91.8 | 78.5 | 79.5 | 76.1 |
| (Fang et al. 2018) | | | 86.7 | 86.7 | 94.8 | 93.5 | - | - | 71.2 |
| (Luo, Jin, and Sun 2019) | ✓ | | 91.2 | 88.3 | 95.0 | 92.4 | 76.1 | 77.4 | 68.8 |
| (Liao et al. 2019) ¹ | | ✓ | 92.0 | 86.4 | - | 91.5 ¹ | - | 79.9 | - |
| (Li et al. 2019) | | ✓ | 91.5 | 84.5 | - | 91.0 | 76.4 | 83.3 | 69.2 |
| (Xie et al. 2019) | | ✓ | - | - | - | - | 70.1 | 82.6 | 68.9 |
| (Zhan and Lu 2019) | ✓ | | 93.3 | 90.2 | - | 91.3 | 79.6 | 83.3 | 76.9 |
| DAN-1D | | | 93.3 | 88.4 | 95.2 | 94.2 | 76.8 | 80.6 | 71.8 |
| DAN-2D | | ✓ | 94.3 | 89.2 | 95.0 | 93.9 | 80.0 | 84.4 | 74.5 |

¹ character-level annotation required.

H/r=1就是1D识别器，适合长的规则文本的识别；H/r > 1就是2D识别器，适合不规则文本的识别。在IC13和IC03等数据集上论文的实验效果还是比较先进的。

值得学习的地方：

- 特征融合的思路，提取更准确更丰富的特征
- 在特征输入和输出层加dropout层。