PaddleOCR技术调研与测试报告

PaddleOCR旨在创建多语言、出色领先且实用的 OCR 工具，帮助用户训练更好的模型并与实践结合。其支持多种与OCR相关的前沿算法，并开发了PP-OCR和PP-Structure，打通了数据生产、模型训练、压缩、推理和部署的全流程。PP-OCR主要由 DB 文本检测、检测框矫正和CRNN 文本识别三部分组成。该系统从骨干网络选择和调整、预测头部的设计、数据增强、学习率变换策略、正则化参数选择、预训练模型使用以及模型自动裁剪量化 8 个方面，采用 19 个有效策略，对各个模块的模型进行效果调优和瘦身，最终得到整体大小为 3.5M 的超轻量中英文 OCR 和 2.8M 的英文数字 OCR。

PaddleOCR chooses **DB** and **CRNN** as the basic detection and recognition models and proposes a series of models, named **PP-OCR**. Based on the capabilities of PP-OCR, PaddleOCR releases the **PP-Structure toolkit** for document scene tasks, including two major tasks: layout analysis and table recognition.

PaddleOCR integrates many OCR algorithms, **text detection algorithms** include **DB, EAST, SAST**, etc., **text recognition algorithms** include **CRNN, RARE, StarNet, Rosetta, SRN** and other algorithms.

基础知识

Basic concept of OCR

Optical character recognition是全称。在正式开始识别图片中的文字之前，还有一系列的预处理操作（**Pre-processing**，为了提高成功识别的机会）手段包括：

1. 去歪斜（扫描时没有正确对齐）。
2. 去除正负斑点，平滑边缘。
3. 二值化：把所需要识别的文本和背景分离。
4. 线删除。
5. 局部分析/分区：将列、段落、标题等标识为不同的块。这个对表格识别很重要。
6. 线条和单词检测：为单词和字符形状建立基线，必要时将单词分开。
7. Identification of the script is necessary, before the right OCR can be invoked to handle the specific script.
8. 字符分割。每个字符的OCR需要分开并最后重新连接起来
9. 标准化纵横比和比例。

在整体的识别过程中，主要分为两大块：**text detection and text recognition**. The text detection module first uses detection algorithms to **detect text lines in the image**. And then the recognition algorithm to **identify the specific text in the text line**.

Detection Model

基于深度学习的文本检测算法大致可以分为以下几类：

1. 基于目标检测的方法。一般文本框预测出来后，通过NMS过滤最终的文本框，多为四点文本框，对于弯曲的文本场景并不理想。典型的算法有 **EAST** 和 **Text Box** 等方法。
2. 基于文本分割的方法。将文本行作为分割目标，然后通过分割结果构建外部文本框，可以处理弯曲文本，对于文本跨场景问题效果并不理想。典型的算法有**DB**、**PSENet**等方法。
3. 混合目标检测和分割方法。

Recognition Model

识别模型的输入一般是背景信息较少的文本行图像，文本信息占主要部分。 识别算法可以主要分为两类：

1. The text prediction module of the recognition algorithm is based on CTC, and the commonly used algorithm combination is **CNN+RNN+CTC**.
2. Attention-based method. The text prediction module of the recognition algorithm is based on Attention, and the commonly used algorithm combination is **CNN+RNN+Attention**.

CNN

学习网址；代码讲解网址；

卷积层（提取图像的局部信息特征）+池化层（大幅降低参数量级））+全连接层（传统神经网络）

卷基层：使用一个过滤器（卷积核）来过滤图像的各个小区域，从而得到这些小区域的特征值；**卷积层的通过卷积核的过滤提取出图片中局部的特征，跟的人类视觉的特征提取类似**。

池化层（数据降维，避免过拟合）：简单说就是下采样，他可以大大降低数据的维度。

全连接层：经过卷积层和池化层处理过的数据输入到全连接层，得到最终想要的结果。经过卷积层和池化层降维过的数据，全连接层才能”跑得动”，不然数据量太大，计算成本高，效率低下。

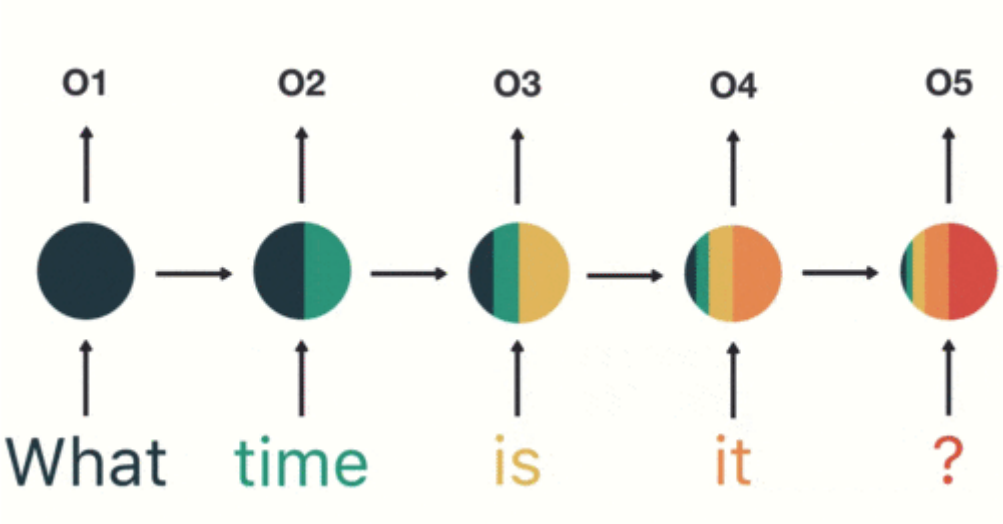
典型的 CNN 并非只是上面提到的3层结构，而是多层结构，例如 LeNet-5： **卷积层 – 池化层- 卷积层 – 池化层 – 卷积层 – 全连接层**。

RNN

学习网址，代码网址；

CNN是一个输入对应一个输出，可是有的时候是需要处理序列数据的。比如要理解“你的名字是什么”这个问题，这里每个字词都需要知道而且还要知道顺序。**而RNN就是用来解决序列数据的**。典型的，就是我们项目中的文字内容。

RNN 跟传统神经网络最大的区别在于每次都会将前一次的输出结果，带到下一次的隐藏层中，一起训练。



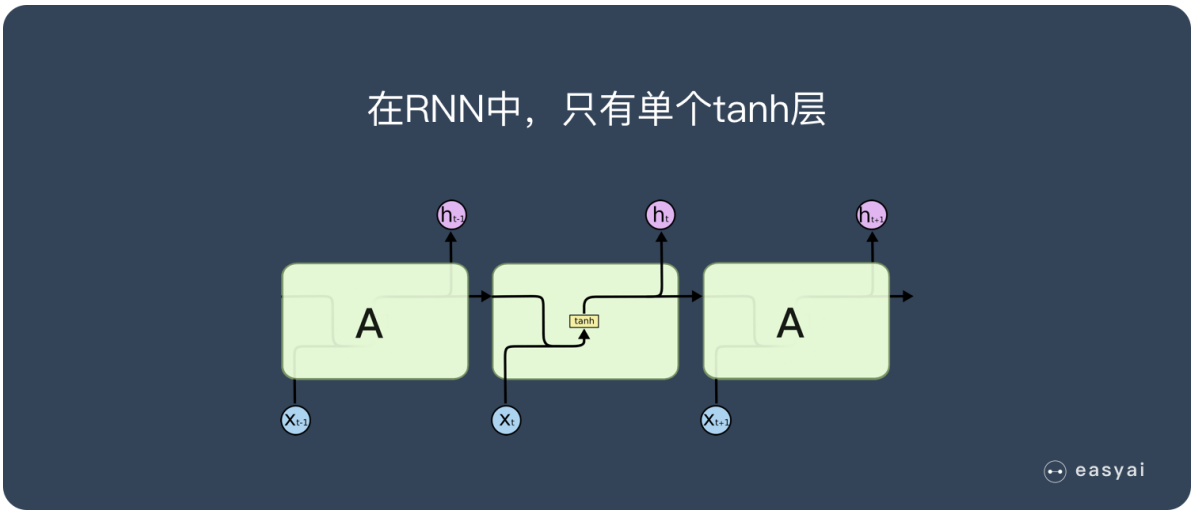
缺点：短期的记忆影响较大（如橙色区域），但是长期的记忆影响就很小（如黑色和绿色区域）（梯度消失），因此无法处理很长的输入队列；训练RNN需要投入很大的成本。因此提出了改进，LSTM就是一种。GRU是一种LSTM的变体（Gated Recurrent Units）

LSTM

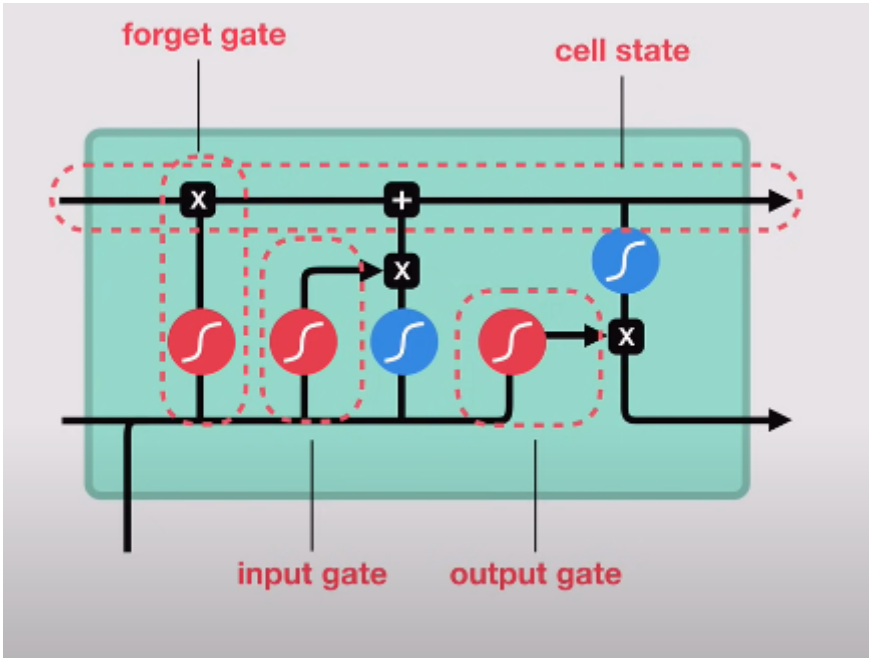
学习网址；

相比于RNN，中间的内部结构更能使LSTM对“文本划重点”。

RNN：标准的RNN，只有tanh，如图：



而LSTM：



cell state:重要的信息可以直接通过。这样就不会导致重要信息在网络传播中的影响力逐渐降低了；而其他的gate是为了向cell state中加入或者移除信息的。The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training.

红色是sigmoid activations: it squishes values between 0 and 1。作用：any number getting multiplied by 0 is 0, causing values to disappears or be “forgotten.” Any number multiplied by 1 is the same value therefore that value stay’s the same or is “kept.”

首先看forget gate: decides what information should be thrown away or kept；0忘1记

其次是input gate: 帮助更新cell state。同样先将两个输入传给sigmoid函数，转换成0-1.0不重要1重要，再把同样的值传给tanh函数压缩到-1~1用来帮助调节网络。之后把这两个结果相乘。sigmoid 输出将决定从 tanh 输出中保留哪些重要信息。

cell state的更新：先和forget gate的输出相乘，可能会“忘记”一些信息；之后和input gate的输出相加更新。

最后是output gate：将之前的隐藏状态和当前输入传递给一个 sigmoid 函数，再将刚更新的cell state传递给 tanh 函数。将两个输出相乘来决定更新的隐藏状态应该携带什么信息。

伪代码：

```
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
    candidate = candidate_layer(combine)
    it = input_layer(combine)
    Ct = prev_ct * ft + candidate * it
    ot = output_layer(combine)
    ht = ot * tanh(Ct)
    return ht, Ct



ct = [0, 0, 0]
ht = [0, 0, 0]
for input in inputs:
    ct, ht = LSTMCELL(ct, ht, input)
```

PSENET

渐进式扩展网络。Progressive Scale Expansion Network

原始的论文，依然采用基于分割的方式，但是对文本行不同核大小做预测，然后采用渐进式扩展算法扩展小尺度kernel到最终的文本行大小。在小尺度kernel之间存在比较大的margin，因此能够很好的区分相邻的文本行。所以弯取文本的处理会变得很好。比之前的准确率高出6%多。

这里有模型的详细介绍和复现。百度的“论文复现赛”，是一个很不错的学习资源。

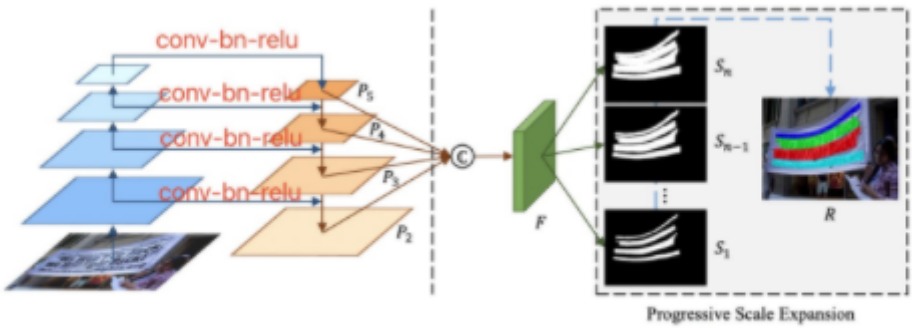


Figure 3. Illustration of our overall pipeline. The left part of pipeline is implemented from FPN [21]. The right part denotes the feature fusion and the progressive scale expansion algorithm.

FCN

全卷积层。论文地址。通常CNN网络会在若干个卷积神经网络后面接全连接神经网络，将二维的feature map转化为一组固定长度的特征向量。FCN语义分割算法对图像做像素级的分类，去掉了CNN网络中的全连接层，代替以反卷积层。反卷积层会对feature map进行上采样，使它恢复原输入图像的尺寸，从而使每个像素都可以产生一个预测。

感受野(Receptive Field)

在卷积神经网络中，感受野（Receptive Field）的定义是卷积神经网络每一层输出的特征图（feature map）上的像素点在输入图片上映射的区域大小。更简单来说，就是特征图上的一个点对应输入图上的区域。

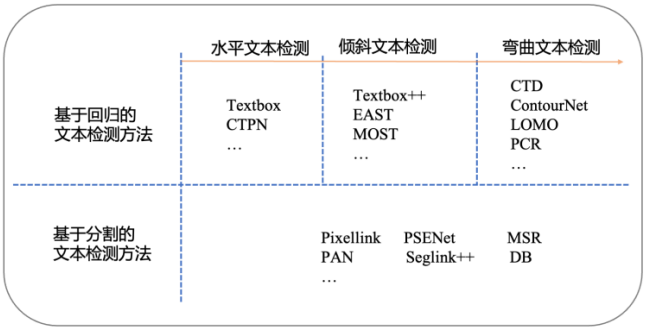
学习网址

PaddleOCR技术流程

大致处理流程

- 流程概括：
  - while (进行版面分析):
    - if (版面分析结果是文字、标题、图片、列表) then
      - do (文本检测+文本识别)
    - else do (表格结构化处理)
- 因此针对英文的手写体识别，主要流程是进行：文本检测+文本识别。
- 如果其中包括表格，也需要进行表格的结构化处理；

- 文本检测：对给定输入图像找出文本的区域，可以是单字符位置或者整个文本行位置；
- 针对英文印刷体/手写体识别，主要存在以下难点：
  1. 受到文字颜色、大小、字体、形状、方向以及文本长度的影响；
  2. 可能会有模糊，阴影，亮度等因素的影响；
  3. 文本密集甚至重叠会影响文字的检测；



•4

Regression-based Detection Model

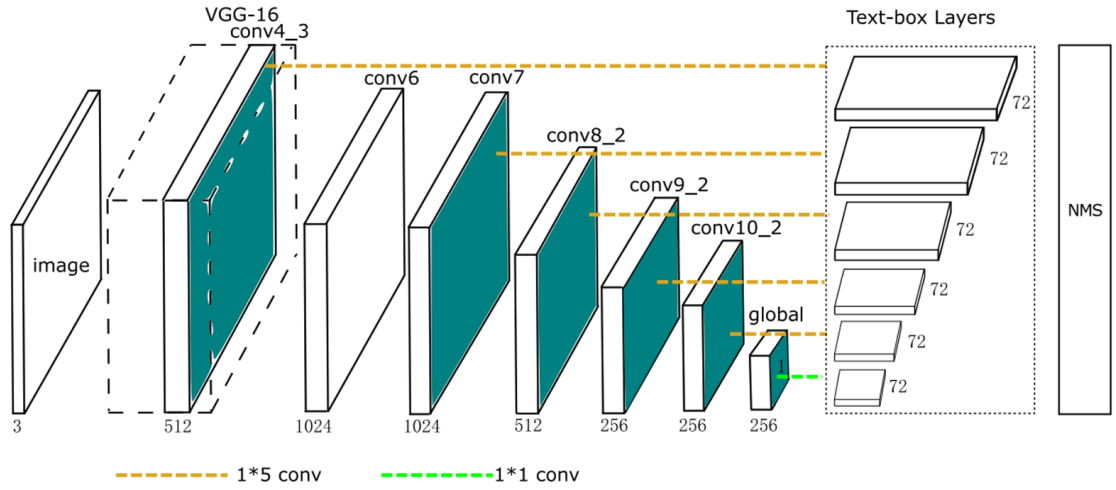
水平文本检测

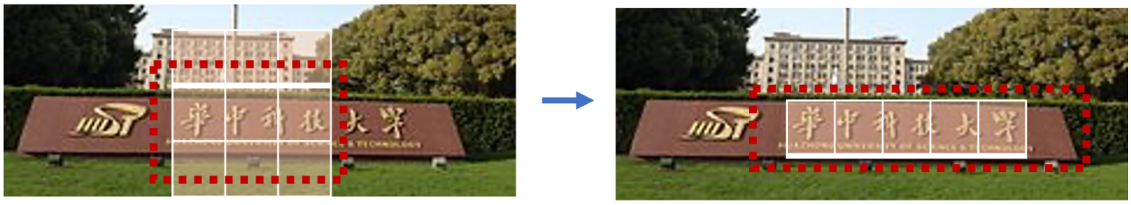
TextBox算法基于SSD算法改进而来，CTPN根据二阶段目标检测Fast-RCNN算法改进而来。

TextBox将默认文本框更改为适应文本方向和宽高比的规格的四边形，提供了一种端对端训练的文字检测方法，并且无需复杂的后处理。

- 采用更大长宽比的预选框
- 卷积核从3x3变成了1x5，更适合长文本检测
- 采用多尺度输入

下图为TextBox框架图：

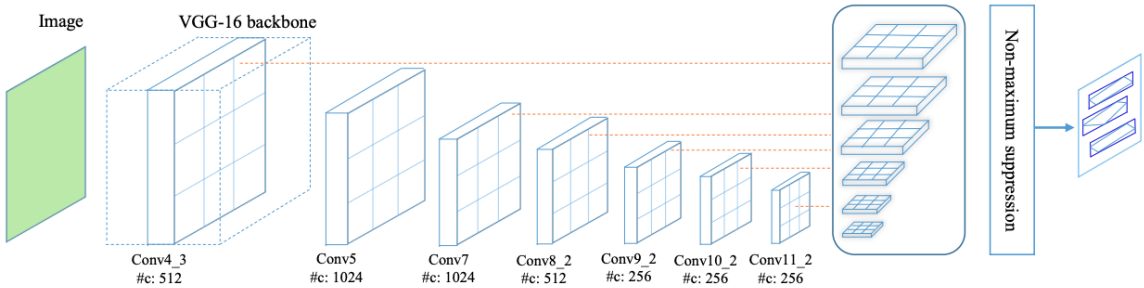




CTPN基于Fast-RCNN算法，设计了基于RCNN的模块让整个网络从卷积特征中检测到文本序列，二阶段的方法通过ROI Pooling获得了更准确的特征定位。但是TextBoxes和CTPN只支持检测横向文本。

倾斜文本检测

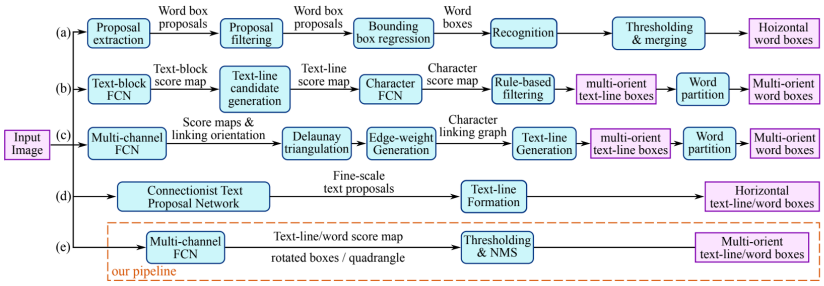
TextBox++在TextBox基础上进行改进，支持检测任意角度的文本。主要做了三个方面的改进：首先修改预选框的宽高比；其次是修改卷积核的大小；最后是会输出旋转框的表示信息。



MOST提出TFAM模块动态的调整粗粒度的检测结果的感受野。在训练过程中给出了一个改进的损失函数以帮助平衡训练，处理不同尺度的文本实例。该方法可以和EAST方法结合，在检测极端长宽比和不同尺度的文本有更好的检测效果和性能。

EAST针对倾斜文本的定位问题，提出了two-stage的文本检测方法，包含 FCN特征提取和NMS部分。EAST提出了一种新的文本检测pipline结构，可以端对端训练并且支持检测任意朝向的文本，并且具有结构简单，性能高的特点。FCN支持输出倾斜的矩形框和水平框，可以自由选择输出格式。

- 如果输出检测形状为RBox，则输出Box旋转角度以及AABB文本形状信息，AABB表示到文本框上下左右边的偏移。RBox可以旋转矩形的文本。
- 如果输出检测框为四点框，则输出的最后一个维度为8个数字，表示从四边形的四个角顶点的位置偏移。该输出方式可以预测不规则四边形的文本。



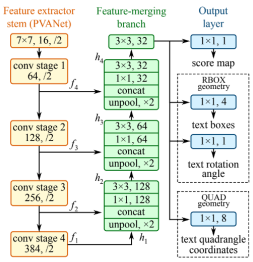


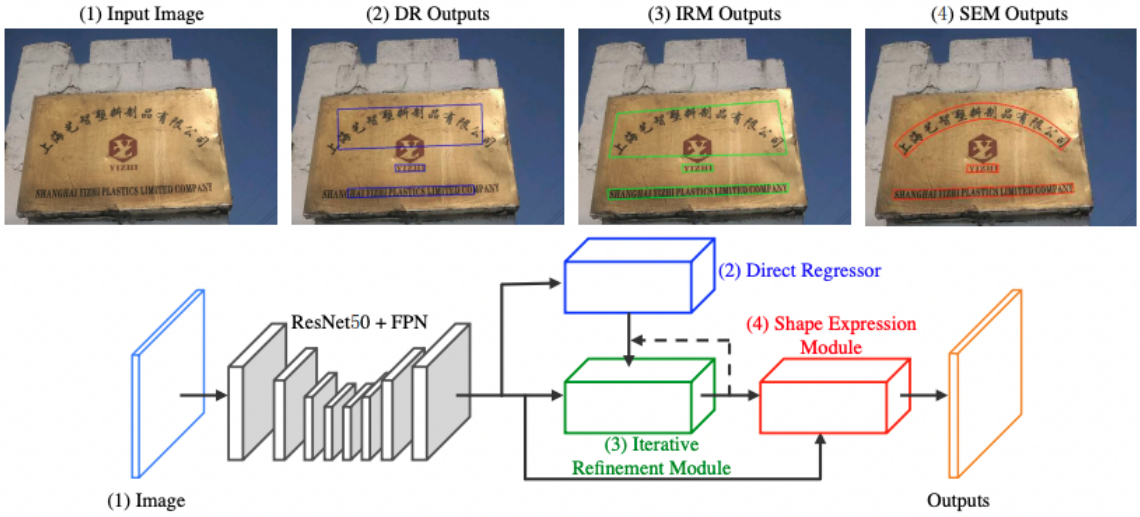
Figure 3. Structure of our text detection FCN.

弯曲文本检测

利用回归的方法解决弯曲文本的检测问题，一个简单的思路是用多点坐标描述弯曲文本的边界多边形，然后直接预测多边形的顶点坐标。

CTD提出了直接预测弯曲文本14个顶点的边界多边形，网络中利用LSTM层以细化顶点的预测坐标，实现了基于回归方法的弯曲文本检测。

LOMO针对长文本和弯曲文本问题，提出迭代的优化文本定位特征获取更精细的文本定位。该方法包括三个部分：坐标回归模块DR，迭代优化模块IRM以及任意形状表达模块SEM。它们分别用于生成文本大致区域，迭代优化文本定位特征，预测文本区域、文本中心线以及文本边界。迭代的优化文本特征可以更好的解决长文本定位问题以及获得更精确的文本区域定位。



Segmentation-based Detection Model

DB

DB全称为（Differentiable Binarization）

参考网址1、网址2;

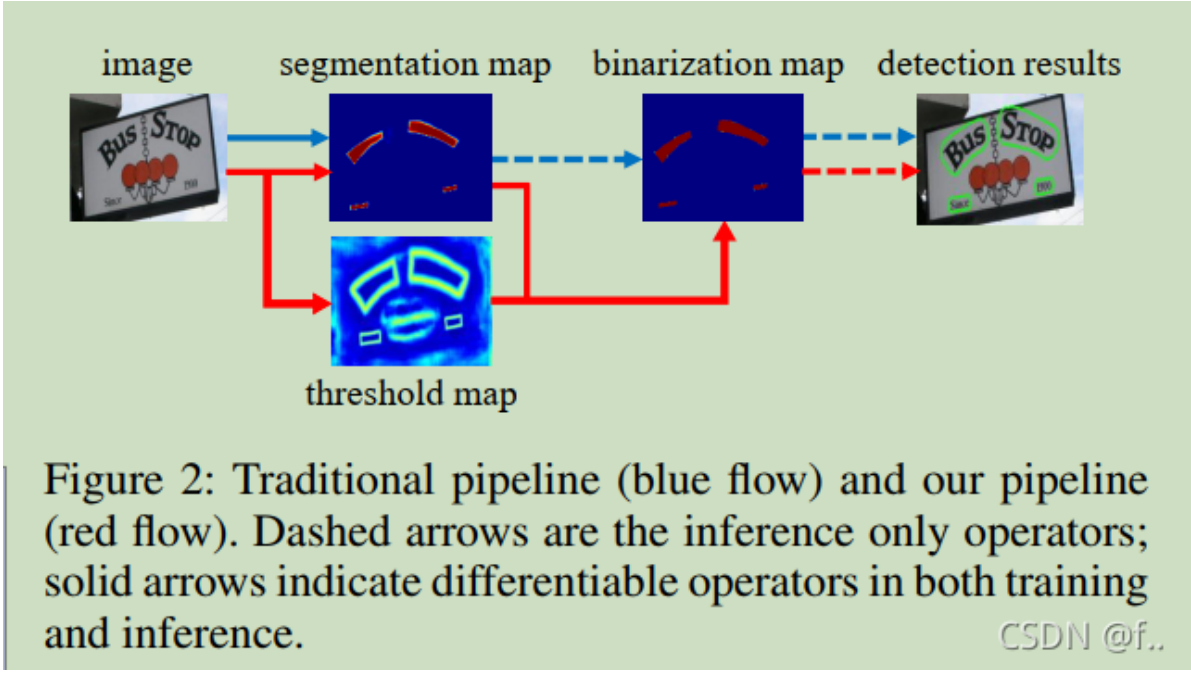


Figure 2: Traditional pipeline (blue flow) and our pipeline (red flow). Dashed arrows are the inference only operators; solid arrows indicate differentiable operators in both training and inference.

蓝线——传统意义上的文本检测， 需对segmentation map 人为的设定一个阈值，转换为二值化图，即binarization map，然后，通过binarization map中的红色区域，慢慢扩张，找到文字区域，即detection results。 属于自底向上的过程，先找到像素点，再确定文字区域。

缺点在于 segmentation map中区域选择的阈值是人为固定的。

红线——与传统方法区别在于阈值选取方面，通过网络预测每一个位置处的阈值，而不是采用一个固定的值，可以很多的将背景和前景分离出来，但是这样的操作会遇到一个问题：给训练带来了梯度不可微的情况，因此，对于此二值化提出了Differentiable Binarization （DB）来解决不可谓的问题。因此，BD可以在分割网络中执行二值化的过程，可以自适应的设置二值化阈值，不仅可以简化后处理，并且提高了文本检测的性能。

网络结构详述：

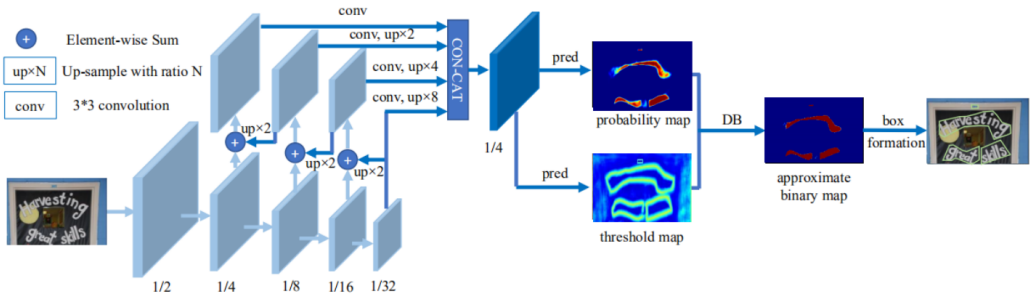


Figure 3: Architecture of our proposed method, where "pred" consists of a 3×3 convolutional operator and two de-convolutional operators with stride 2. The "1/2", "1/4", ... and "1/32" indicate the scale ratio compared to the input image.

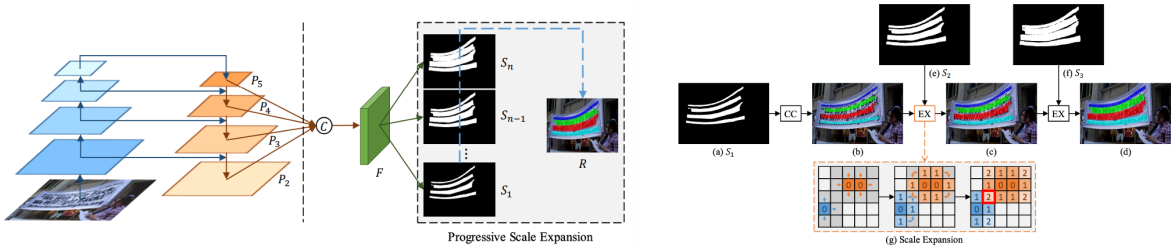
输入图片经过特征金字塔 backbone，之后进行上采样，到同一尺寸F。之后特征图FFF用来同时预测概率图PPP和阈值图 T，由 F 和 T 计算后近似得到二值图 B 。

整个网络是一个基于backbone 的 FPN 结构；FPN 结构接两个head 分别输出probability map 和 threshold map，同时图片 size 从 1/4(H, W) 上采样回原图大小；最后两个 map 通过 DB 运算得到最后的 binary map。

Pixel link

其采用分割的方法解决文本检测问题，分割对象为文本区域，将同属于一个文本行（单词）中的像素链接在一起来分割文本，直接从分割结果中提取文本边界框，无需位置回归就能达到基于回归的文本检测的效果。

针对基于分割的文本算法难以区分相邻文本的问题，**PSENet**提出渐进式的尺度扩张网络学习文本分割区域，预测不同收缩比例的文本区域，并逐个扩大检测到的文本区域，该方法本质上是边界学习方法的变体，可以有效解决任意形状相邻文本的检测问题。



假设PSENet后处理用了3个不同尺度的kernel，如上图s1,s2,s3所示。首先，从最小kernel s1开始，计算文本分割区域的连通域，得到(b)，然后，对连通域沿着上下左右做尺度扩张，对于扩张区域属于s2但不属于s1的像素，进行归类，遇到冲突点时，采用“先到先得”原则，重复尺度扩张的操作，最终可以得到不同文本行的独立的分割区域。

PP-OCR Models Compression

Knowledge Distillation

There are several main methods to **compress the model and reduce the amount of its parameters**. **Such as pruning, quantification, knowledge distillation, etc**., where knowledge distillation refers to the use of teacher models to guide student models to learn specific tasks, to ensure that the small model obtains a relatively large performance improvement under the condition of unchanged parameters.

knowledge distillation算法集成在PaddleOCR中，具体来说，具有以下特点：它支持任何网络的相互学习、同时对子网数量没有限制，只需在配置文件中添加即可、配置文件任意配置损失函数、支持知识蒸馏训练、预测、评估、导出等所有模型相关环境

最终效果：在不添加任何耗时预测的情况下，模型的准确率可以提升 3% 以上。结合学习率调整策略和模型结构微调策略，最终提升5%以上。

知识就是指对于网络模型中参数权重的一些抽取/迁移的操作

一个复杂而强大的Teacher Model（简称Net-T），以及另一个简单而弱小的Student Model（简称Net-S），由Net-T完整地学习Ground Truth，然后再由Net-S同时学习Net-T的Logit和Ground Truth，最终Net-S作为应用模型，而Net-T并不进行部署上线。

logit是什么：模型输出的对于各个类别的概率预测值；Net-S既要学习正例，也要学习反例。

在2015年的那篇论文中，作者Hinton将问题限定在**分类问题**下，分类问题的共同点是模型最后会有一个softmax层，其输出值对应了相应类别的概率值。在知识蒸馏时，由于我们已经有了一个泛化能力较强的Teacher模型，我们在利用Teacher模型来蒸馏训练Student模型时，可以直接让Student模型去学习Teacher模型的泛化能力。一个很直白且高效的迁移泛化能力的方法就是：使用softmax层输出的类别的概率来作为“Soft-target”。**hard-target就是真实值**。传统的神经网络训练方法是定义一个损失函数，目标是使预测值尽可能接近于真实值。

- **Hard-target**：原始数据集标注的 one-shot 标签，除了正标签为 1，其他负标签都是 0。
- **Soft-target**：Teacher模型softmax层输出的类别概率，每个类别都分配了概率，正标签的概率最高。

这样简单的模型就也可以学习到负标签中的知识。

Quantization

a more complex model would achieve better performance in the task, but it also leads to some redundancy in the model. Quantization is a technique that reduces this redundancy by reducing the full precision data to a fixed number, so as to reduce model calculation complexity and improve model inference performance.

Pruning

Model Pruning is a technique that reduces this redundancy by removing the sub-models in the neural network model, so as to reduce model calculation complexity and improve model inference performance.

权重剪枝：也就是将模型中的一些不重要的权重删除

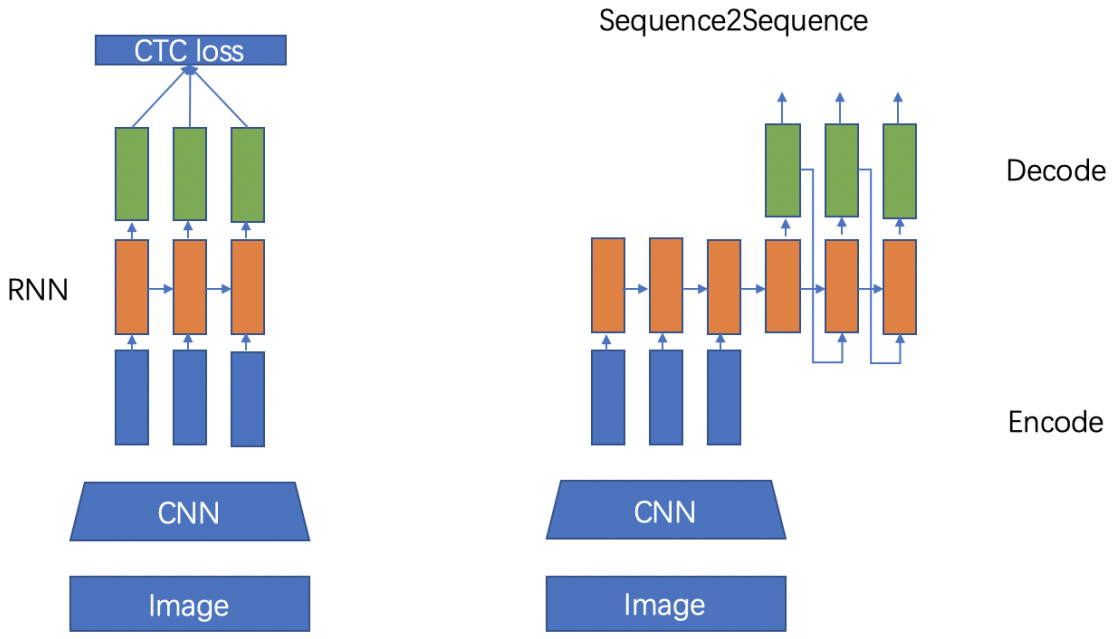
神经元剪枝：实践中往往是删除参数矩阵的一行元素

Regular Text Recognition Model

文本识别的主流算法有两种，分别是基于 CTC (Conectionist Temporal Classification) 的算法和 Sequence2Sequence 算法，区别主要在解码阶段。

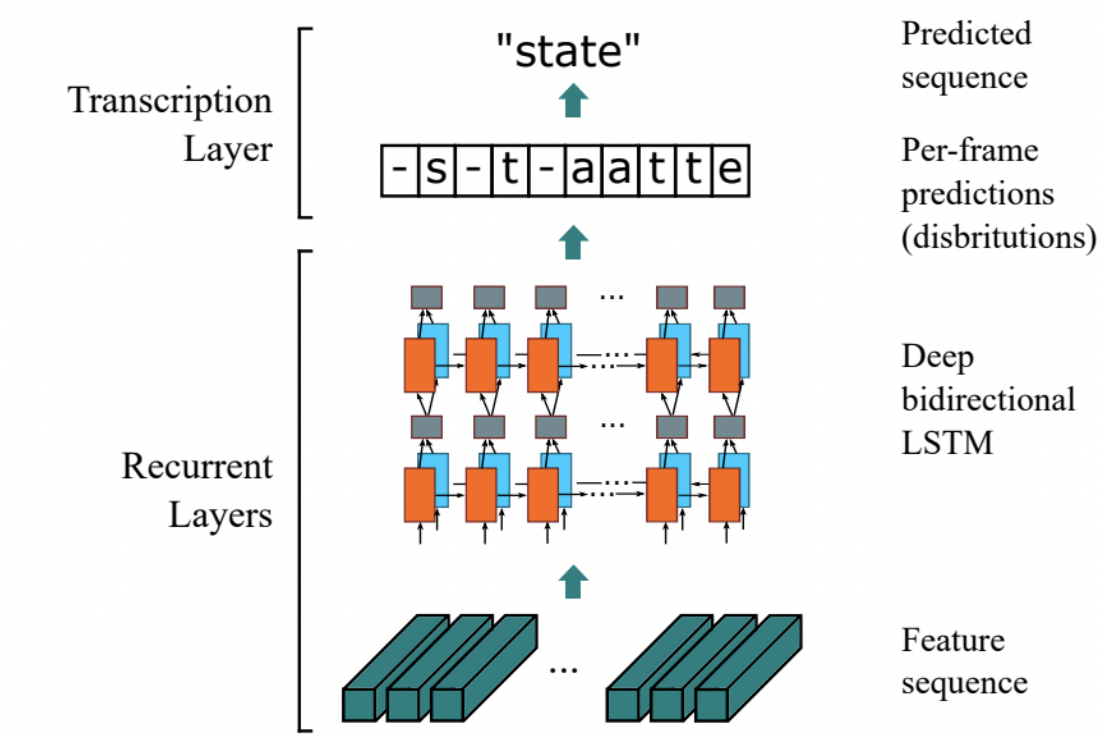
基于 CTC 的算法

基于 CTC 的算法是将编码产生的序列接入 CTC 进行解码；基于 Sequence2Sequence 的方法则是把序列接入循环神经网络(Recurrent Neural Network, RNN)模块进行循环解码，两种方式都验证有效也是主流的两大做法。（下图，左：基于CTC的方法，右：基于Sequece2Sequence的方法）



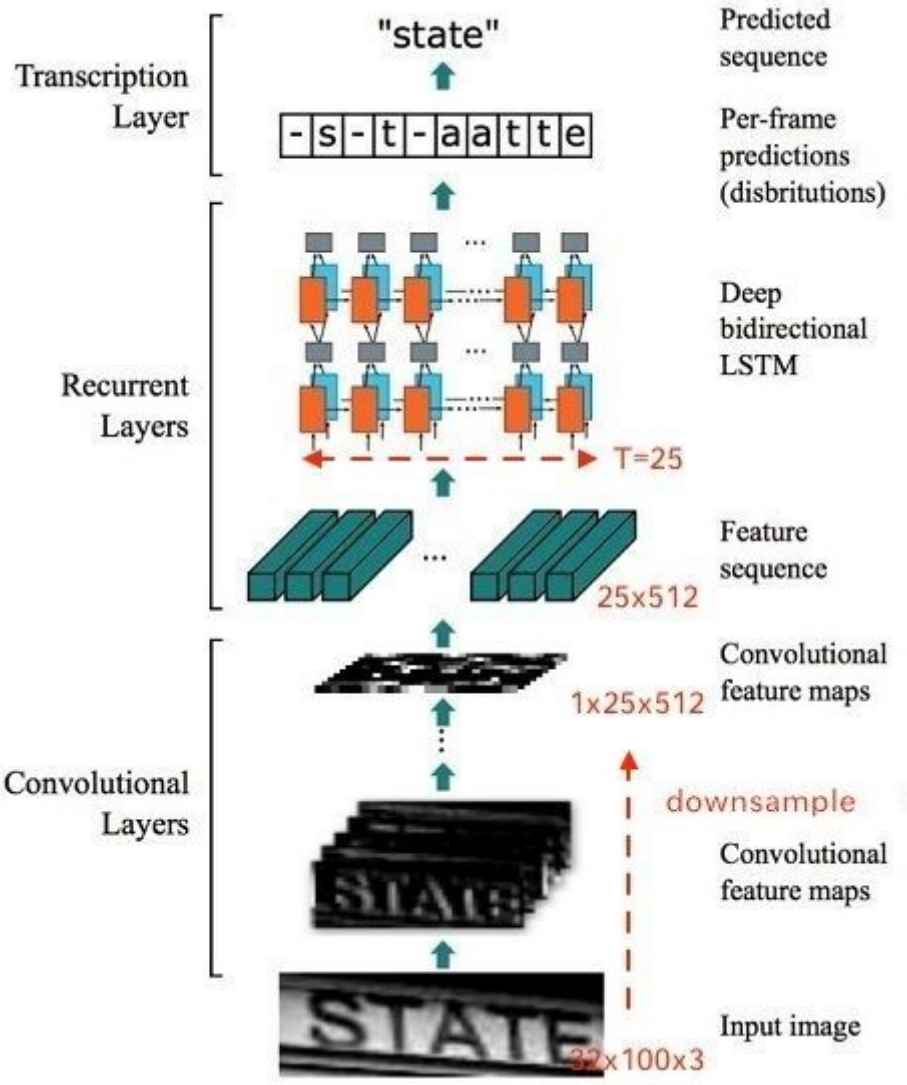
基于CTC的算法

基于 CTC 最典型的算法是CRNN (Convolutional Recurrent Neural Network)[1]，它的特征提取部分使用主流的卷积结构，常用的有ResNet、MobileNet、VGG等。由于文本识别任务的特殊性，输入数据中存在大量的上下文信息，卷积神经网络的卷积核特性使其更关注于局部信息，缺乏长依赖的建模能力，因此仅使用卷积网络很难挖掘到文本之间的上下文联系。为了解决这一问题，CRNN文本识别算法引入了双向 LSTM(Long Short-Term Memory) 用来增强上下文建模，通过实验证明双向LSTM模块可以有效的提取出图片中的上下文信息。最终将输出的特征序列输入到CTC模块，直接解码序列结果。该结构被验证有效，并广泛应用在文本识别任务中。Rosetta[2]是FaceBook提出的识别网络，由全卷积模型和CTC组成。Gao Y[3]等人使用CNN卷积替代LSTM，参数更少，性能提升精度持平。



学习网址; 代码学习; (该代码支持不定长英文识别)

CRNN



假设输入图像大小为 $(32, 100, 3)$，注意提及图像都是 $(Height, Width, Channel)$形式。

- Convlutional Layers

这里的卷积层就是一个普通的CNN网络，用于提取输入图像的Convolutional feature maps，即将大小为 $(32, 100, 3)$ 的图像转换为 大$(1, 25, 512)$小的卷积特征矩阵.

- Recurrent Layers

这里的循环网络层是一个深层双向LSTM网络，在卷积特征的基础上继续提取文字序列特征。由于CNN输出的Feature map是$(1, 25, 512)$大小，所以对于RNN最大时间长度 $T = 25$即有25个时间输入，每个输入 X_t 列向量有 $D = 512$)。

- Transcription Layers

将RNN输出做softmax后，为字符输出。

CTC(Connectionist Temporal Classification)

这一层为转录层，将 RNN 对每个特征向量所做的预测转换成标签序列的过程。其中的数学原理较为复杂，详情请查看这里；详细解释其作用：

端到端 OCR 识别的难点，在于如何处理不定长序列的对齐。这个问题可转化为时序依赖的文本图像问题，此时使用 CTC（Connectionist Temporal Classification, CTC）的损失函数对 CNN 和 RNN 进行端到端的联合训练以帮助问题的解决。

目前来说，在Tensorflow中官方实现了CTC接口：

```
tf.nn.ctc_loss(
    labels,
    inputs,
    sequence_length,
    preprocess_collapse_repeated=False,
    ctc_merge_repeated=True,
    ignore_longer_outputs_than_inputs=False,
    time_major=True
)
```

Pytorch的CTC接口：`torch.nn.CTCLoss(blank=0,reduction='mean',zero_infinity=False)`；

所以，CTC总结来说，CTC是一种Loss计算方法，用CTC代替Softmax Loss，训练样本无需对齐。特点如下：

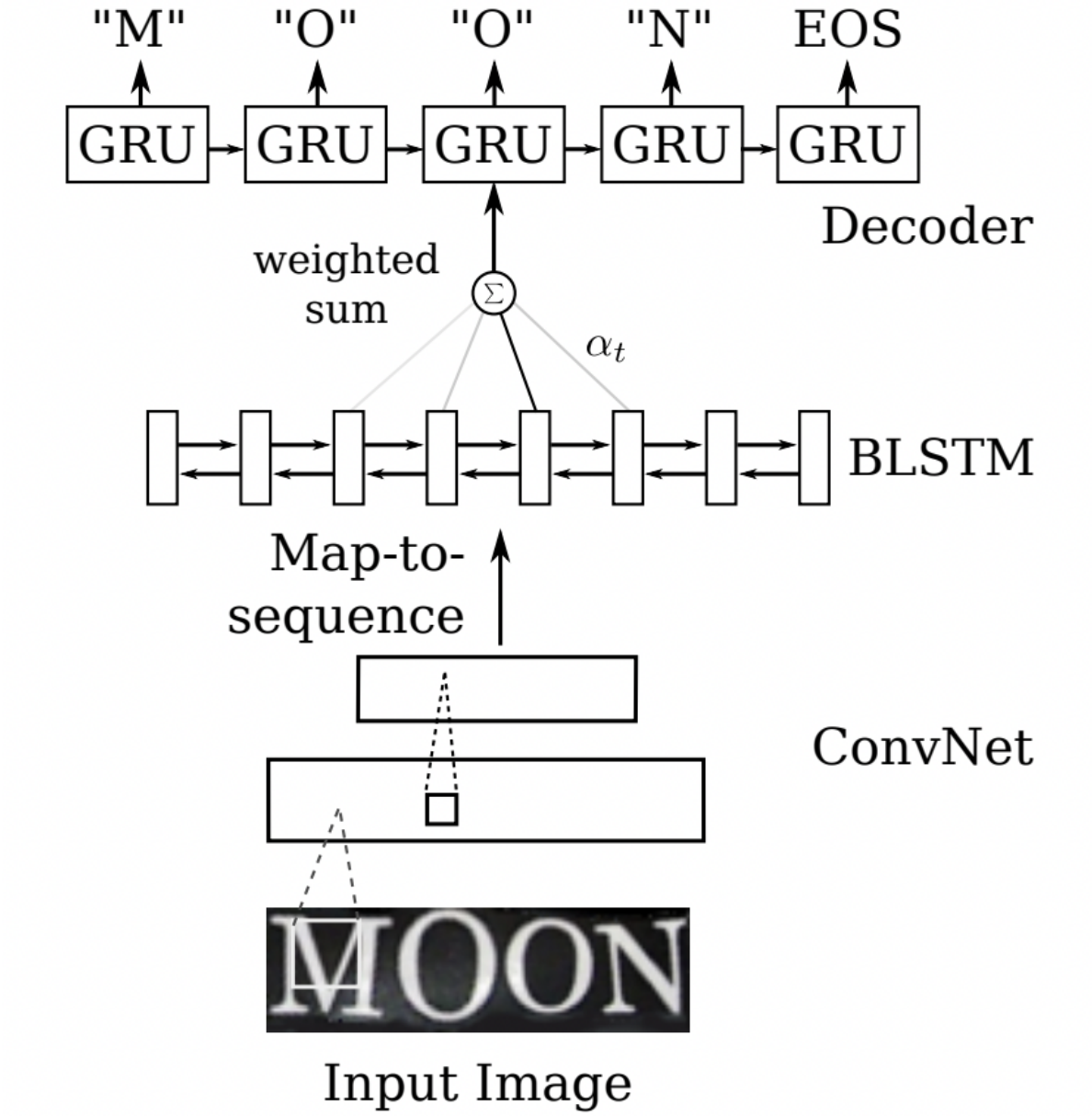
- 引入blank字符，解决有些位置没有字符的问题
- 通过递推，快速计算梯度

而综合CRNN+CTC，就是将CNN/LSTM/CTC三种方法结合：

- 首先CNN提取图像卷积特征
- 然后LSTM进一步提取图像卷积特征中的序列特征
- 最后引入CTC解决训练时字符无法对齐的问题

Sequence2Sequence 算法

Sequence2Sequence 算法是由编码器 Encoder 把所有的输入序列都编码成一个统一的语义向量，然后再由解码器Decoder解码。在解码器Decoder解码的过程中，不断地将前一个时刻的输出作为后一个时刻的输入，循环解码，直到输出停止符为止。一般编码器是一个RNN，对于每个输入的词，编码器输出向量和隐藏状态，并将隐藏状态用于下一个输入的单词，循环得到语义向量；解码器是另一个RNN，它接收编码器输出向量并输出一系列字以创建转换。受到 Sequence2Sequence 在翻译领域的启发，Shi[4]提出了一种基于注意的编解码框架来识别文本,通过这种方式，rnn能够从训练数据中学习隐藏在字符串中的字符级语言模型。



Sequence2Sequence 结构图

以上两个算法在规则文本上都有很不错的效果，但由于网络设计的局限性，这类方法很难解决弯曲和旋转的不规则文本识别任务。为了解决这类问题，部分算法研究人员在以上两类算法的基础上提出了一系列改进算法。

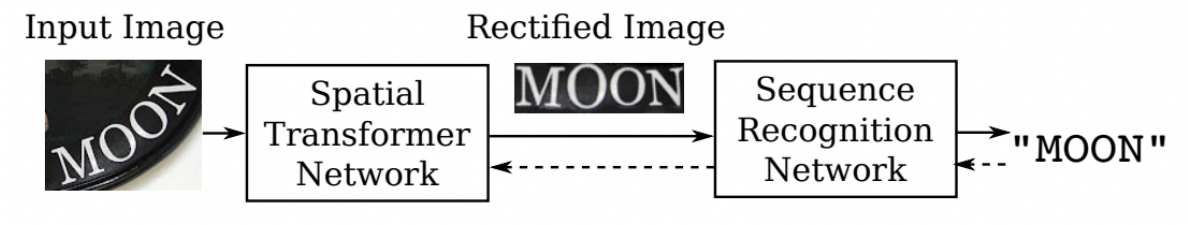
Irregular Text Recognition Model

主要可以分为四大类：基于校正的方法；基于 Attention 的方法；基于分割的方法；基于 Transformer 的方法。

基于校正的方法

基于校正的方法利用一些视觉变换模块，将非规则的文本尽量转换为规则文本，然后使用常规方法进行识别。

RARE模型首先提出了对不规则文本的校正方案，整个网络分为两个主要部分：一个空间变换网络 STN(Spatial Transformer Network) 和一个基于Sequence2Squence的识别网络。其中STN就是校正模块，不规则文本图像进入STN，通过TPS(Thin-Plate-Spline)变换成一个水平方向的图像，该变换可以一定程度上校正弯曲、透射变换的文本，校正后送入序列识别网络进行解码。



RARE 结构图

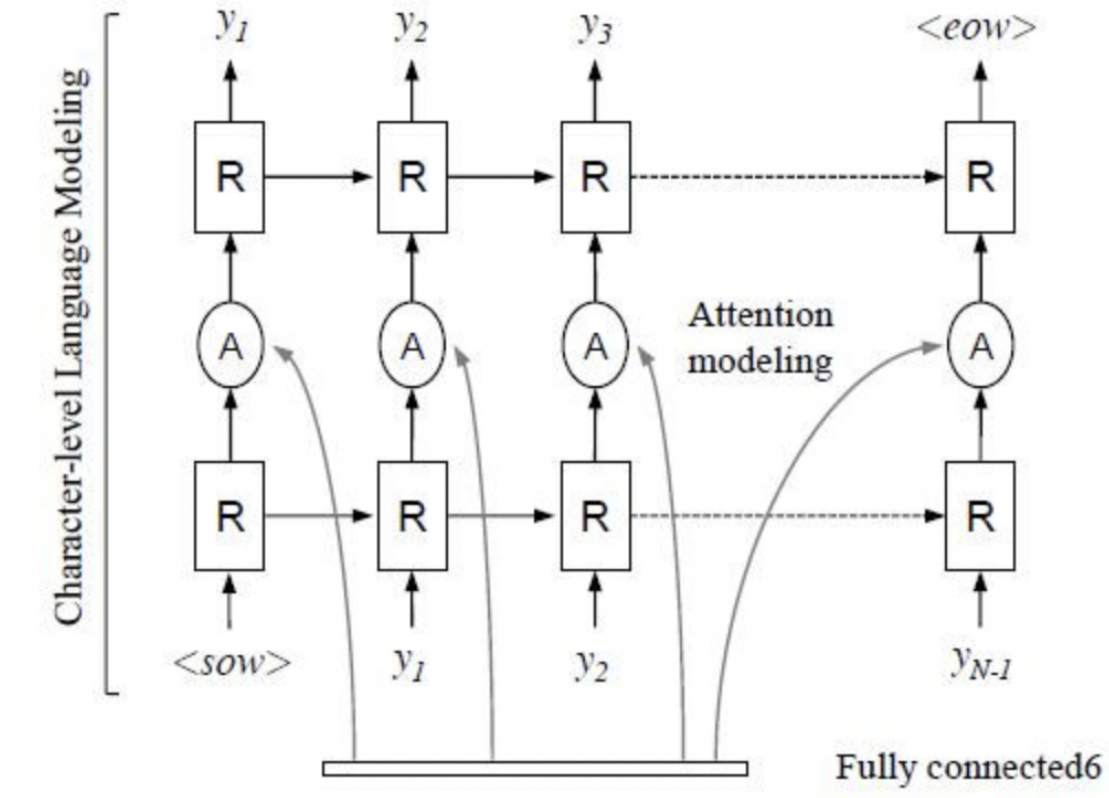
RARE论文指出，该方法在不规则文本数据集上有较大的优势，特别比较了CUTE80和SVTP这两个数据集，相较CRNN高出5个百分点以上，证明了校正模块的有效性。基于此，同样结合了空间变换网络(STN)和基于注意的序列识别网络的文本识别系统。

基于校正的方法有较好的迁移性，除了RARE这类基于Attention的方法外，STAR-Net[5]将校正模块应用到基于CTC的算法上，相比传统CRNN也有很好的提升。

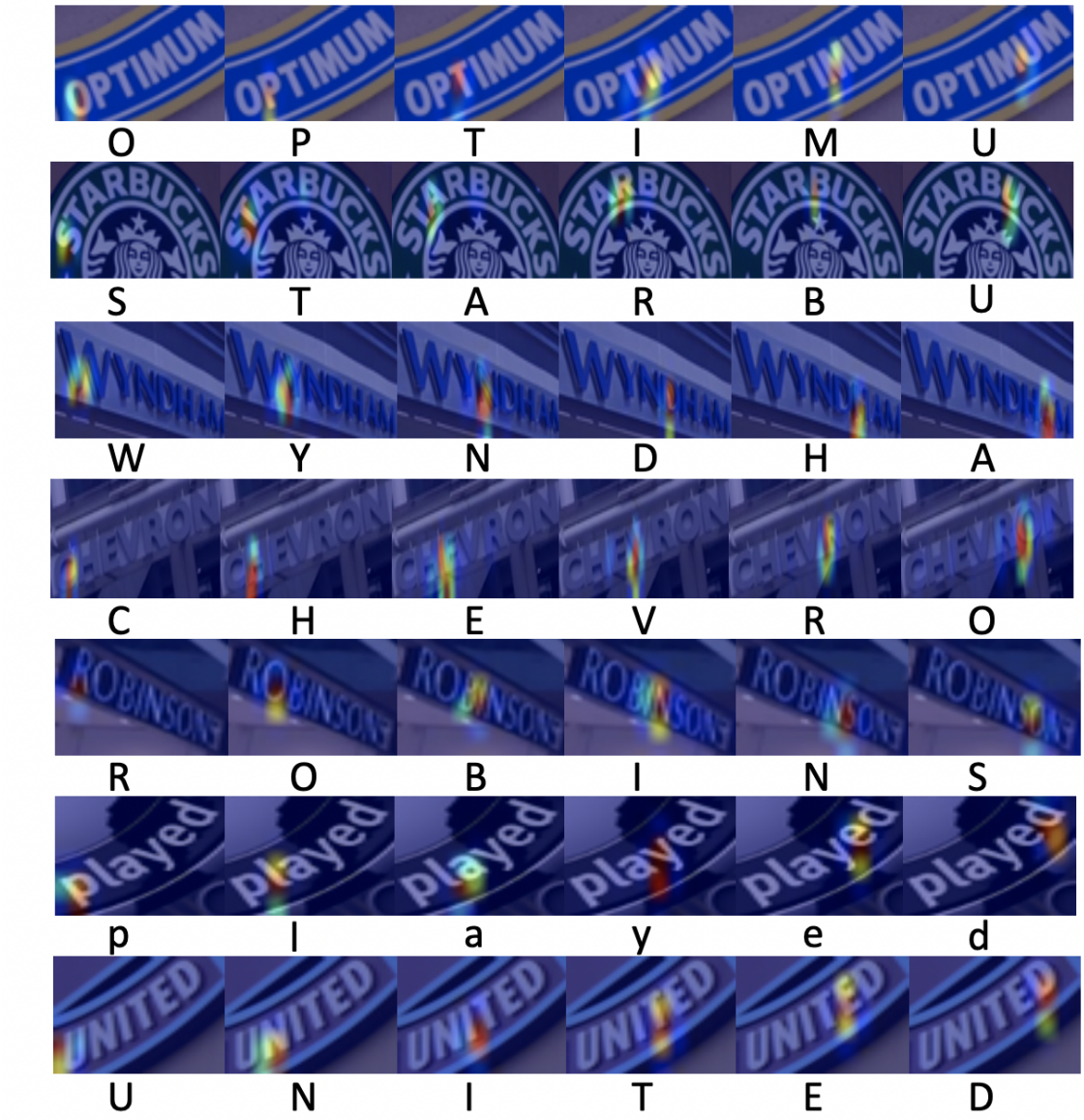
基于Attention的方法

基于 Attention 的方法主要关注的是序列之间各部分的相关性，该方法最早在机器翻译领域提出，认为在文本翻译的过程中当前词的结果主要由某几个单词影响的，因此需要给有决定性的单词更大的权重。在文本识别领域也是如此，将编码后的序列解码时，每一步都选择恰当的context来生成下一个状态，这样有利于得到更准确的结果。

R^2AM首次将 Attention 引入文本识别领域，该模型首先将输入图像通过递归卷积层提取编码后的图像特征，然后利用隐式学习到的字符级语言统计信息通过递归神经网络解码输出字符。在解码过程中引入了Attention 机制实现了软特征选择，以更好地利用图像特征，这一有选择性的处理方式更符合人类的直觉。

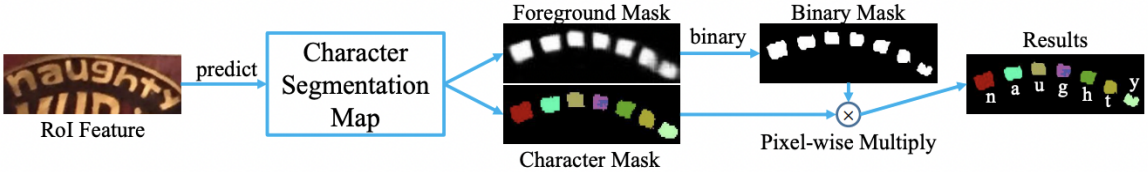


后续有大量算法在Attention领域进行探索和更新，例如SAR将1D attention拓展到2D attention上，校正模块提到的RARE也是基于Attention的方法。实验证明基于Attention的方法相比CTC的方法有很好的精度提升。



基于分割的方法

基于分割的方法是将文本行的各字符作为独立个体，相比与对整个文本行做矫正后识别，识别分割出的单个字符更加容易。它试图从输入的文本图像中定位每个字符的位置，并应用字符分类器来获得这些识别结果，将复杂的全局问题简化成了局部问题解决，在不规则文本场景下有比较不错的效果。然而这种方法需要字符级别的标注，数据获取上存在一定的难度。Lyu[9]等人提出了一种用于单词识别的实例分词模型，该模型在其识别部分使用了基于 FCN(Fully Convolutional Network) 的方法。从二维角度考虑文本识别问题，设计了一个字符注意FCN来解决文本识别问题，当文本弯曲或严重扭曲时，该方法对规则文本和非规则文本都具有较优的定位结果。



Mask TextSpotter 结构图

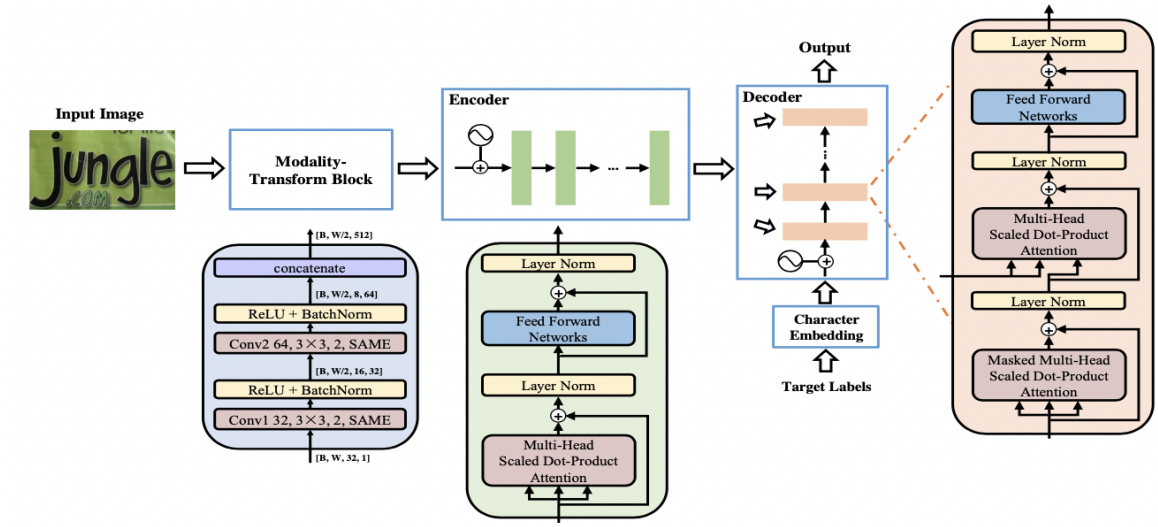
基于Transformer的方法

随着 Transformer 的快速发展，分类和检测领域都验证了 Transformer 在视觉任务中的有效性。如规则文本识别部分所说，CNN在长依赖建模上存在局限性，Transformer 结构恰好解决了这一问题，它可以在特征提取器中关注全局信息，并且可以替换额外的上下文建模模块（LSTM）。

一部分文本识别算法使用 Transformer 的 Encoder 结构和卷积共同提取序列特征，Encoder 由多个 MultiHeadAttentionLayer 和 Positionwise Feedforward Layer 堆叠而成的block组成。MulitHeadAttention 中的 self-attention 利用矩阵乘法模拟了RNN的时序计算，打破了RNN中时序长时依赖的障碍。也有一部分算法使用 Transformer 的 Decoder 模块解码，相比传统RNN可获得更强的语义信息，同时并行计算具有更高的效率。

SRN算法将Transformer的Encoder模块接在ResNet50后，增强了2D视觉特征。并提出了一个并行注意力模块，将读取顺序用作查询，使得计算与时间无关，最终并行输出所有时间步长的对齐视觉特征。此外SRN还利用Transformer的Eecoder作为语义模块，将图片的视觉信息和语义信息做融合，在遮挡、模糊等不规则文本上有较大的收益。

NRTR使用了完整的Transformer结构对输入图片进行编码和解码，只使用了简单的几个卷积层做高层特征提取，在文本识别上验证了Transformer结构的有效性。



NRTR 结构图

SRACN使用Transformer的解码器替换LSTM，再一次验证了并行训练的高效性和精度优势。

PaddleOCR代码框架

来源：点击这里；

```
PaddleOCR
├── configs                                 // Configuration file, you can config
the model structure and modify the hyperparameters through the yml file
│   ├── cls                                 // Angle classifier config files
│   │   ├── cls_mv3.yml                     // Training config, including
backbone network, head, loss, optimizer and data
│   ├── det                                 // Text detection config files
│   │   ├── det_mv3_db.yml                  // Training config
│   │   ...
│   └── rec                                 // Text recognition config files
│       ├── rec_mv3_none_bilstm_ctc.yml     // CRNN config
│       ...
├── deploy                                  // Depoly
│   ├── android_demo                        // Android demo
│   │   ...
```

```
|   ├── cpp_infer                          // C++ infer
|   |   ├── CMakeLists.txt                 // Cmake file
|   |   ├── docs                           // Docs
|   |   |   └── windows_vs2019_build.md
|   |   ├── include                        // Head Files
|   |   |   ├── clipper.h                  // clipper
|   |   |   ├── config.h                   // Inference config
|   |   |   ├── ocr_cls.h                  // Angle class
|   |   |   ├── ocr_det.h                  // Text detection
|   |   |   ├── ocr_rec.h                  // Text recognition
|   |   |   ├── postprocess_op.h           // Post-processing
|   |   |   ├── preprocess_op.h            // Pre-processing
|   |   |   └── utility.h                  // tools
|   |   ├── readme.md                      // Documentation
|   |   ├── ...
|   |   ├── src                            // Source code files
|   |   |   ├── clipper.cpp
|   |   |   ├── config.cpp
|   |   |   ├── main.cpp
|   |   |   ├── ocr_cls.cpp
|   |   |   ├── ocr_det.cpp
|   |   |   ├── ocr_rec.cpp
|   |   |   ├── postprocess_op.cpp
|   |   |   ├── preprocess_op.cpp
|   |   |   └── utility.cpp
|   |   └── tools                          // Compile and execute script
|   |       ├── build.sh                   // Compile script
|   |       ├── config.txt                 // Config file
|   |       └── run.sh                     // Execute script
|   ├── docker
|   |   └── hubserving
|   |       ├── cpu
|   |       |   └── Dockerfile
|   |       ├── gpu
|   |       |   └── Dockerfile
|   |       ├── README_cn.md
|   |       ├── README.md
|   |       └── sample_request.txt
|   ├── hubserving                         // hubserving
|   |   ├── ocr_cls                        // Angle class
|   |   |   ├── config.json                // Serving config
|   |   |   ├── __init__.py
|   |   |   ├── module.py                  // Model
|   |   |   └── params.py                  // Parameters
|   |   ├── ocr_det                        // Text detection
|   |   |   ├── config.json                // serving config
|   |   |   ├── __init__.py
|   |   |   ├── module.py                  // Model
|   |   |   └── params.py                  // Parameters
|   |   ├── ocr_rec                        // Text recognition
|   |   |   ├── config.json
|   |   |   ├── __init__.py
|   |   |   ├── module.py
|   |   |   └── params.py
|   |   └── ocr_system                     // Inference System
|   |       ├── config.json
|   |       ├── __init__.py
|   |       ├── module.py
```

```
|   |       └── params.py
|   ├── imgs                              // Inference images
|   |   ├── cpp_infer_pred_12.png
|   |   └── demo.png
|   ├── ios_demo                          // IOS demo
|   |   ...
|   ├── lite                              // Lite depoly
|   |   ├── cls_process.cc                // Pre-process for angle class
|   |   ├── cls_process.h
|   |   ├── config.txt                    // Config file
|   |   ├── crnn_process.cc               // Pre-process for CRNN
|   |   ├── crnn_process.h
|   |   ├── db_post_process.cc            // Pre-process for DB
|   |   ├── db_post_process.h
|   |   ├── Makefile                      // Compile file
|   |   ├── ocr_db_crnn.cc                // Inference system
|   |   ├── prepare.sh                    // Prepare bash script
|   |   ├── readme.md                     // Documentation
|   |   ...
|   ├── pdserving                         // Pdserving depoly
|   |   ├── det_local_server.py           // Text detection fast version, easy
to deploy and fast to predict
|   |   ├── det_web_server.py             // Text detection full version, high
stability distributed deployment
|   |   ├── ocr_local_server.py           // Text detection + recognition fast
version
|   |   ├── ocr_web_client.py             // client
|   |   ├── ocr_web_server.py             // Text detection + recognition full
version
|   |   ├── readme.md                     // Documentation
|   |   ├── rec_local_server.py           // Text recognition fast version
|   |   └── rec_web_server.py             // Text recognition full version
|   └── slim
|       └── quantization                  // Quantization
|           ├── export_model.py           // Export model
|           ├── quant.py                  // Quantization script
|           └── README.md                 // Documentation
├── doc                                   // Documentation and Tutorials
|   ...
├── ppocr                                 // Core code
|   ├── data                              // Data processing
|   |   ├── imaug                         // Image and label processing code
|   |   |   ├── text_image_aug            // Tia data augment for text
recognition
|   |   |   |   ├── __init__.py
|   |   |   |   ├── augment.py            // Tia_distort,tia_stretch and
tia_perspective
|   |   |   |   ├── warp_mls.py
|   |   |   ├── __init__.py
|   |   |   ├── east_process.py           // Data processing steps of EAST
algorithm
|   |   |   ├── iaa_augment.py            // Data augmentation operations
|   |   |   ├── label_ops.py              // label encode operations
|   |   |   ├── make_border_map.py        // Generate boundary map
|   |   |   ├── make_shrink_map.py        // Generate shrink graph
|   |   |   ├── operators.py              // Basic image operations, such as
reading and normalization
|   |   |   ├── randaugment.py            // Random data augmentation operation
```

```
|   |   |   ├── random_crop_data.py         // Random crop
|   |   |   ├── rec_img_aug.py              // Data augmentation for text
recognition
|   |   |   └── sast_process.py             // Data processing steps of SAST
algorithm
|   |   ├── __init__.py                     // Construct dataloader code
|   |   ├── lmdb_dataset.py                 // Read lmdb dataset
|   |   ├── simple_dataset.py               // Read the dataset stored in text
format
|   ├── losses                              // Loss function
|   |   ├── __init__.py                     // Construct loss code
|   |   ├── cls_loss.py                     // Angle class loss
|   |   ├── det_basic_loss.py               // Text detection basic loss
|   |   ├── det_db_loss.py                  // DB loss
|   |   ├── det_east_loss.py                // EAST loss
|   |   ├── det_sast_loss.py                // SAST loss
|   |   ├── rec_ctc_loss.py                 // CTC loss
|   |   ├── rec_att_loss.py                 // Attention loss
|   ├── metrics                             // Metrics
|   |   ├── __init__.py                     // Construct metric code
|   |   ├── cls_metric.py                   // Angle class metric
|   |   ├── det_metric.py                   // Text detection metric
    |   ├── eval_det_iou.py                 // Text detection iou code
|   |   ├── rec_metric.py                   // Text recognition metric
|   ├── modeling                            // Network
|   |   ├── architectures                   // Architecture
|   |   |   ├── __init__.py                 // Construct model code
|   |   |   ├── base_model.py               // Base model
|   |   ├── backbones                       // backbones
|   |   |   ├── __init__.py                 // Construct backbone code
|   |   |   ├── det_mobilenet_v3.py         // Text detection mobilenet_v3
|   |   |   ├── det_resnet_vd.py            // Text detection resnet
|   |   |   ├── det_resnet_vd_sast.py       // Text detection resnet backbone of
the SAST algorithm
|   |   |   ├── rec_mobilenet_v3.py         // Text recognition mobilenet_v3
|   |   |   └── rec_resnet_vd.py            // Text recognition resnet
|   |   ├── necks                           // Necks
|   |   |   ├── __init__.py                 // Construct neck code
|   |   |   ├── db_fpn.py                   // Standard fpn
|   |   |   ├── east_fpn.py                 // EAST algorithm fpn network
|   |   |   ├── sast_fpn.py                 // SAST algorithm fpn network
|   |   |   ├── rnn.py                      // Character recognition sequence
encoding
|   |   ├── heads                           // Heads
|   |   |   ├── __init__.py                 // Construct head code
|   |   |   ├── cls_head.py                 // Angle class head
|   |   |   ├── det_db_head.py              // DB head
|   |   |   ├── det_east_head.py            // EAST head
|   |   |   ├── det_sast_head.py            // SAST head
|   |   |   ├── rec_ctc_head.py             // CTC head
|   |   |   ├── rec_att_head.py             // Attention head
|   |   ├── transforms                      // Transforms
|   |   |   ├── __init__.py                 // Construct transform code
|   |   |   └── tps.py                      // TPS transform
|   ├── optimizer                           // Optimizer
|   |   ├── __init__.py                     // Construct optimizer code
|   |   └── learning_rate.py                // Learning rate decay
|   |   └── optimizer.py                    // Optimizer
```

```
|   |   └── regularizer.py                  // Network regularization
|   ├── postprocess                         // Post-processing
|   |   ├── cls_postprocess.py              // Angle class post-processing
|   |   ├── db_postprocess.py               // DB post-processing
|   |   ├── east_postprocess.py             // EAST post-processing
|   |   ├── locality_aware_nms.py           // NMS
|   |   ├── rec_postprocess.py              // Text recognition post-processing
|   |   └── sast_postprocess.py             // SAST post-processing
|   └── utils                               // utils
|       ├── dict                            // Minor language dictionary
|            ....
|       ├── ic15_dict.txt                   // English number dictionary, case
sensitive
|       ├── ppocr_keys_v1.txt               // Chinese dictionary for training
Chinese models
|       ├── logging.py                      // logger
|       ├── save_load.py                    // Model saving and loading functions
|       ├── stats.py                        // Training status statistics
|       └── utility.py                      // Utility function
├── tools
|   ├── eval.py                             // Evaluation function
|   ├── export_model.py                     // Export inference model
|   ├── infer                               // Inference based on Inference
engine
|   |   ├── predict_cls.py
|   |   ├── predict_det.py
|   |   ├── predict_rec.py
|   |   ├── predict_system.py
|   |   └── utility.py
|   ├── infer_cls.py                        // Angle classification inference
based on training engine
|   ├── infer_det.py                        // Text detection inference based on
training engine
|   ├── infer_rec.py                        // Text recognition inference based
on training engine
|   ├── program.py                          // Inference system
|   ├── test_hubserving.py
|   └── train.py                            // Start training script
├── paddleocr.py
├── README_ch.md                            // Chinese documentation
├── README_en.md                            // English documentation
├── README.md                               // Home page documentation
├── requirements.txt                         // Requirements
├── setup.py                                // Whl package packaging script
├── train.sh                                // Start training bash script
```

使用学习

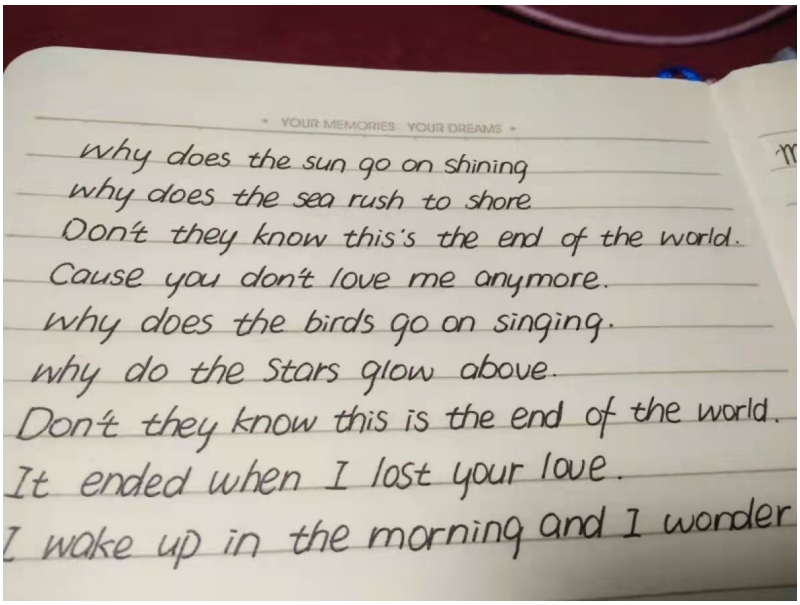
PaddleOCR终端命令框架总结

```
paddleocr --image_dir [图片相对路径] [其他参数]
#[其他参数]包括:
--use_gpu false #不使用GPU
--det false #不检测字符，一般适用于一张图几乎就是一个单词的情况
--rec false #不识别字符
--lang en #指定识别的语言
--use_angle_cls true #用于控制是否启用方向分类模型
--type=structure #use the layout analysis function of PaddleOCR
--ocr_version PP-OCRv3#选择OCR Model version
```

其中的layout analysis是指：对文档的5类区域进行划分，包括文本、标题、列表、图片和表格。前三类区域，直接使用OCR模型完成对应区域的文本检测识别，并将结果保存为txt。对于表格区域，经过表格结构化处理，将表格图片转换为相同表格样式的Excel文件。 图片区域将被单独裁剪成图像。

Baseline简易测试

手写体



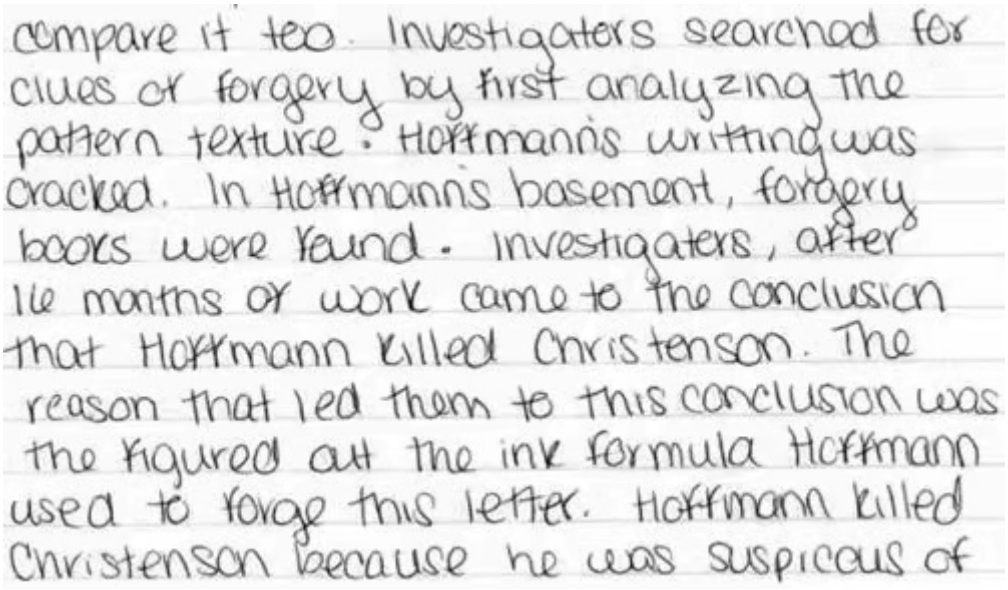
指令：`paddleocr --image_dir ./4.jpg --lang en`

结果如下：

```
ppocr INFO: [[[433.0, 170.0], [594.0, 174.0], [593.0, 191.0], [432.0, 187.0]], ('OURMIEMORIES', 0.9486512541770935)]
ppocr INFO: [[[611.0, 175.0], [748.0, 178.0], [747.0, 194.0], [611.0, 192.0]], ('YOUR DREAMS', 0.9177137017250061)]
ppocr INFO: [[[116.0, 190.0], [233.0, 201.0], [228.0, 257.0], [111.0, 247.0]], ('why', 0.9858251214027405)]
ppocr INFO: [[[245.0, 207.0], [347.0, 210.0], [345.0, 251.0], [243.0, 248.0]], ('does', 0.9926618337631226)]
ppocr INFO: [[[363.0, 212.0], [793.0, 227.0], [792.0, 270.0], [362.0, 255.0]], ('the_sun qo.on shining', 0.7657021284103394)]
ppocr INFO: [[[104.0, 252.0], [215.0, 265.0], [210.0, 312.0], [99.0, 299.0]], ('why', 0.9461738467216492)]
ppocr INFO: [[[210.0, 268.0], [325.0, 268.0], [325.0, 306.0], [210.0, 306.0]], ('does', 0.9739726185798645)]
ppocr INFO: [[[339.0, 269.0], [800.0, 279.0], [799.0, 317.0], [339.0, 308.0]], ('the Sea_rush to shore', 0.8920543193817139)]
ppocr INFO: [[[93.0, 316.0], [225.0, 325.0], [222.0, 366.0], [90.0, 357.0]], ('Dont', 0.7044990062713623)]
ppocr INFO: [[[244.0, 325.0], [1120.0, 327.0], [1120.0, 375.0], [244.0, 372.0]], ("they knowthis's the end ofthe worldl.", 0.9035105109214783)]
ppocr INFO: [[[73.0, 382.0], [221.0, 391.0], [218.0, 435.0], [70.0, 426.0]], ('Cause', 0.8776683807373047)]
ppocr INFO: [[[238.0, 395.0], [906.0, 395.0], [906.0, 438.0], [238.0, 438.0]], ('you_dont loveme anymore', 0.902344822883606)]
ppocr INFO: [[[61.0, 448.0], [180.0, 452.0], [179.0, 509.0], [60.0, 506.0]], ('why', 0.9300214648246765)]
ppocr INFO: [[[205.0, 455.0], [916.0, 462.0], [916.0, 509.0], [205.0, 502.0]], ('does the birds qoon singing', 0.8640798926353455)]
ppocr INFO: [[[40.0, 523.0], [152.0, 523.0], [152.0, 593.0], [40.0, 593.0]], ('why', 0.9326891899108887)]
ppocr INFO: [[[185.0, 531.0], [255.0, 531.0], [255.0, 577.0], [185.0, 577.0]], ('do', 0.9398139119148254)]
ppocr INFO: [[[274.0, 527.0], [830.0, 524.0], [830.0, 582.0], [274.0, 584.0]], ('the Stars qlow above', 0.8904147148132324)]
ppocr INFO: [[[19.0, 603.0], [181.0, 603.0], [181.0, 662.0], [19.0, 662.0]], ('Don华', 0.8282080888748169)]
ppocr INFO: [[[112.0, 692.0], [294.0, 692.0], [294.0, 745.0], [112.0, 745.0]], ('ended', 0.9277950525283813)]
ppocr INFO: [[[299.0, 684.0], [920.0, 667.0], [922.0, 732.0], [301.0, 750.0]], ('when I lost your love', 0.8210762739181519)]
ppocr INFO: [[[14.0, 695.0], [79.0, 695.0], [79.0, 743.0], [14.0, 743.0]], ('It', 0.8159412145614624)]
```

识别结果小结：
单词之间没有空格；
大小写识别失误；
单独的字母识别失误，例如go识别成qo（这可能涉及到自然语言处理的部分）；
莫名其妙多识别字符（例如world—>worldl）

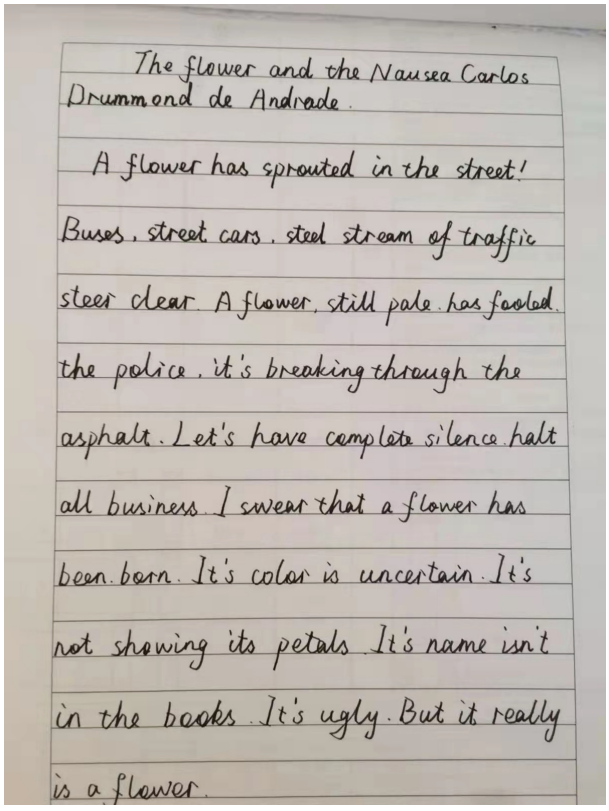
准确率35/72 (48.61%)；



结果如下:

```
ppocr INFO: **********./1.jpg**********
ppocr DEBUG: dt_boxes num : 11, elapse : 1.4532041549682617
ppocr DEBUG: rec_res num  : 11, elapse : 0.17545533180236816
ppocr INFO: [[[5.0, 5.0], [484.0, 1.0], [484.0, 22.0], [5.0, 26.0]], ('Cmpareffeo.Invostiaaters searcnodfex', 0.7774724364280701)]
ppocr INFO: [[[1.0, 27.0], [452.0, 25.0], [452.0, 52.0], [1.0, 54.0]], ('clues crforgerubuhrstanaluzingthe', 0.8452155590057373)]
ppocr INFO: [[[5.0, 56.0], [456.0, 55.0], [456.0, 79.0], [5.0, 80.0]], ('paherntextureHorrmannswrnduas', 0.8753896951675415)]
ppocr INFO: [[[2.0, 80.0], [454.0, 78.0], [454.0, 102.0], [2.0, 104.0]], ('Crackod.InHommonnsbasemoot.foreru', 0.8154370188713074)]
ppocr INFO: [[[5.0, 110.0], [451.0, 102.0], [451.0, 129.0], [5.0, 136.0]], ('hoseednveshaateafer', 0.6951985359191895)]
ppocr INFO: [[[2.0, 135.0], [471.0, 131.0], [471.0, 154.0], [2.0, 158.0]], ('1temontns or uory camo+o*no conclusrcn', 0.7050172090530396)]
ppocr INFO: [[[1.0, 163.0], [454.0, 156.0], [454.0, 179.0], [1.0, 187.0]], ('mhatHorrmannylleocnrrrstensoo.Tho', 0.8111082911491394)]
ppocr INFO: [[[10.0, 190.0], [496.0, 185.0], [496.0, 207.0], [10.0, 211.0]], ('reasonthafledthontothisconclusiancoas', 0.8082813024520874)]
ppocr INFO: [[[11.0, 215.0], [487.0, 211.0], [487.0, 231.0], [11.0, 235.0]], ('thohouredautheinkformulaHcrrmang', 0.8340094089508057)]
ppocr INFO: [[[2.0, 240.0], [480.0, 236.0], [481.0, 259.0], [2.0, 264.0]], ('usedofovoe+hieletfe.Hcrrmannkiled', 0.7437182664871216)]
ppocr INFO: [[[5.0, 268.0], [483.0, 265.0], [483.0, 285.0], [5.0, 288.0]], ('Cnvistenscnpecausehecssuserceusof', 0.8163309693336487)]
```

准确率 2/70（2.857%）；面目全非;

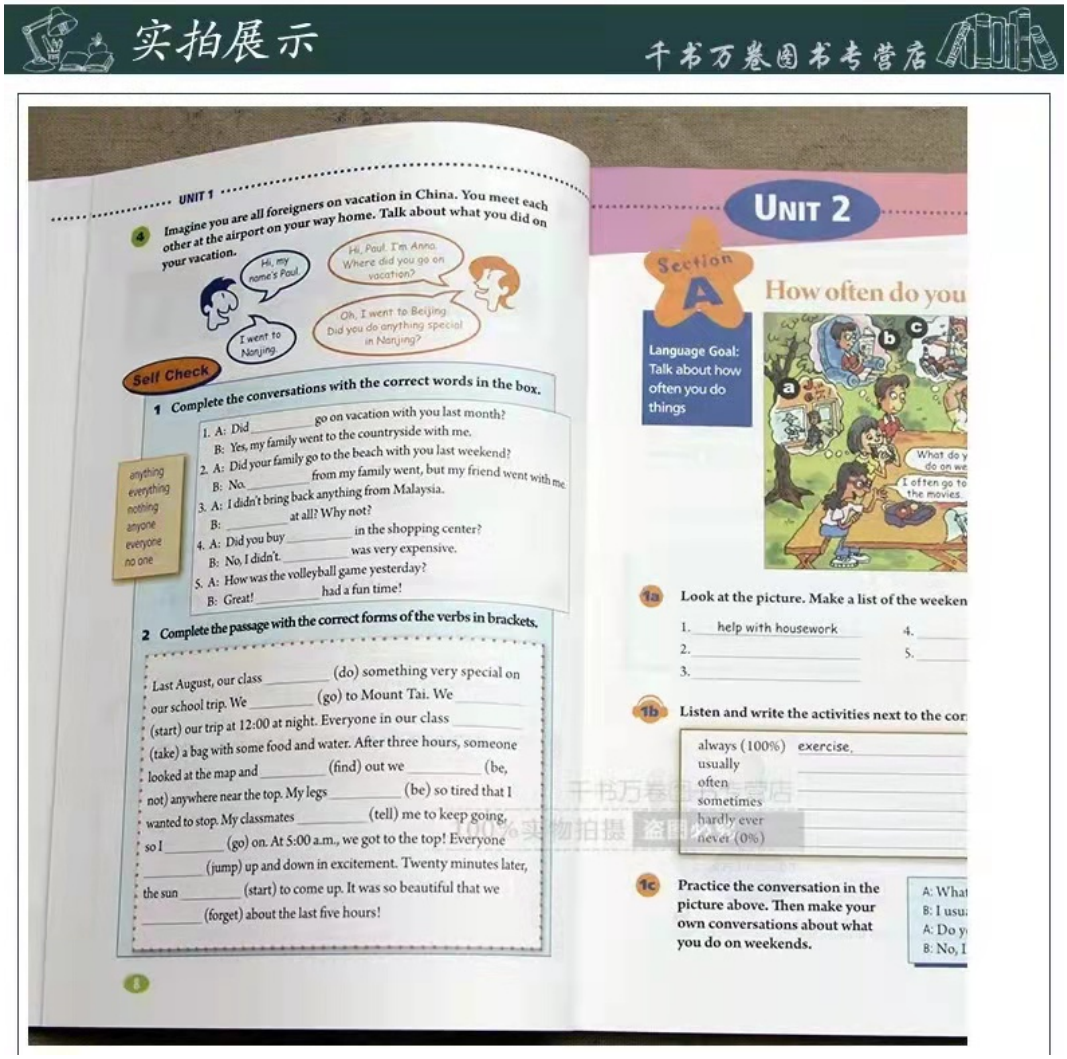


结果如下:

```
[2022/06/09 00:14:20] ppocr INFO: **********5.jpg**********
[2022/06/09 00:14:22] ppocr DEBUG: dt_boxes num : 15, elapse : 2.1470601558685303
[2022/06/09 00:14:22] ppocr DEBUG: rec_res num  : 15, elapse : 0.23544979095458984
[2022/06/09 00:14:22] ppocr INFO: [[[259.0, 88.0], [1049.0, 116.0], [1047.0, 171.0], [257.0, 143.0]], ('The flawer and the Nausea Carlos', 0.8802585005760193)]
[2022/06/09 00:14:22] ppocr INFO: [[[123.0, 155.0], [675.0, 162.0], [674.0, 219.0], [122.0, 211.0]], ('Drummond de Andrade', 0.888118326663971)]
[2022/06/09 00:14:22] ppocr INFO: [[[176.0, 292.0], [1047.0, 298.0], [1046.0, 360.0], [175.0, 353.0]], ('A flower has sprouted intha sreot!', 0.8110594749450684)]
[2022/06/09 00:14:22] ppocr INFO: [[[117.0, 425.0], [834.0, 437.0], [833.0, 489.0], [116.0, 476.0]], ('Buse.sreet car.tel ttream', 0.7188194990158081)]
[2022/06/09 00:14:22] ppocr INFO: [[[837.0, 433.0], [1060.0, 433.0], [1060.0, 517.0], [837.0, 517.0]], ('yftraffie', 0.7438724040985107)]
[2022/06/09 00:14:22] ppocr INFO: [[[105.0, 548.0], [1103.0, 557.0], [1102.0, 642.0], [104.0, 633.0]], ('Stour denr Aflawersti polahofanld', 0.7177930474281311)]
[2022/06/09 00:14:22] ppocr INFO: [[[106.0, 688.0], [1031.0, 693.0], [1031.0, 780.0], [106.0, 775.0]], ('tha policits branking throgh tha', 0.7678844928741455)]
[2022/06/09 00:14:22] ppocr INFO: [[[109.0, 828.0], [1106.0, 822.0], [1106.0, 900.0], [109.0, 907.0]], ("aphalt.Let's hova caploh silena halt", 0.7841176390647888)]
[2022/06/09 00:14:22] ppocr INFO: [[[109.0, 961.0], [394.0, 971.0], [392.0, 1031.0], [107.0, 1021.0]], ('all buinan', 0.769573450088501)]
[2022/06/09 00:14:22] ppocr INFO: [[[404.0, 968.0], [1038.0, 968.0], [1038.0, 1030.0], [404.0, 1030.0]], ('Iswen that aflower has', 0.794857382774353)]
[2022/06/09 00:14:22] ppocr INFO: [[[102.0, 1104.0], [1053.0, 1095.0], [1054.0, 1163.0], [103.0, 1172.0]], ('bemn.bomn lts cnlor w uncertain.lts', 0.7090038061141968)]
[2022/06/09 00:14:22] ppocr INFO: [[[99.0, 1237.0], [422.0, 1251.0], [419.0, 1326.0], [96.0, 1312.0]], ('Not shawing', 0.7968688607215881)]
[2022/06/09 00:14:22] ppocr INFO: [[[436.0, 1251.0], [1090.0, 1234.0], [1091.0, 1303.0], [438.0, 1319.0]], ('to petals lts name int', 0.7455381155014038)]
[2022/06/09 00:14:22] ppocr INFO: [[[94.0, 1382.0], [1110.0, 1385.0], [1109.0, 1463.0], [94.0, 1460.0]], ('intha books lts uyy But it really', 0.7779029607772827)]
[2022/06/09 00:14:22] ppocr INFO: [[[94.0, 1535.0], [394.0, 1533.0], [394.0, 1592.0], [94.0, 1593.0]], ('iafloner', 0.5361188650131226)]
```

准确率26/76 (34.21%)

印刷体



```
[2022/06/09 00:24:18] ppocr INFO: **********2.jpg**********
[2022/06/09 00:24:20] ppocr DEBUG: dt_boxes num : 60, elapse : 2.1825852394104004
[2022/06/09 00:24:21] ppocr DEBUG: rec_res num  : 60, elapse : 0.349597692489624
[2022/06/09 00:24:21] ppocr INFO: [[[93.0, 18.0], [233.0, 15.0], [234.0, 49.0], [93.0, 53.0]], ('实拍展示', 0.9987072944641113)]
[2022/06/09 00:24:21] ppocr INFO: [[[477.0, 34.0], [691.0, 32.0], [691.0, 55.0], [477.0, 57.0]], ('千书万卷图书专营店', 0.9362644553184509)]
[2022/06/09 00:24:21] ppocr INFO: [[[558.0, 145.0], [633.0, 148.0], [632.0, 173.0], [557.0, 169.0]], ('UNIT 2', 0.9805595278739929)]
[2022/06/09 00:24:21] ppocr INFO: [[[118.0, 195.0], [167.0, 187.0], [168.0, 198.0], [120.0, 206.0]], ('wourwaction', 0.5354294776916504)]
[2022/06/09 00:24:21] ppocr INFO: [[[566.0, 209.0], [718.0, 213.0], [718.0, 233.0], [566.0, 229.0]], ('How often do you', 0.9405663013458252)]
[2022/06/09 00:24:21] ppocr INFO: [[[483.0, 260.0], [547.0, 260.0], [547.0, 271.0], [483.0, 271.0]], ('languageGoal', 0.8489136099815369)]
[2022/06/09 00:24:21] ppocr INFO: [[[93.0, 279.0], [145.0, 273.0], [147.0, 288.0], [95.0, 294.0]], ('Selt Chec', 0.8324475884437561)]
[2022/06/09 00:24:21] ppocr INFO: [[[480.0, 271.0], [548.0, 275.0], [548.0, 289.0], [480.0, 285.0]], ('Talkabouthow', 0.9321569800376892)]
[2022/06/09 00:24:21] ppocr INFO: [[[477.0, 286.0], [540.0, 288.0], [540.0, 302.0], [477.0, 300.0]], ('often youdo', 0.9452173113822937)]
[2022/06/09 00:24:21] ppocr INFO: [[[480.0, 302.0], [510.0, 302.0], [510.0, 316.0], [480.0, 316.0]], ('things', 0.9946689605712891)]
[2022/06/09 00:24:21] ppocr INFO: [[[147.0, 319.0], [183.0, 315.0], [184.0, 326.0], [148.0, 331.0]], ('LAD', 0.8094165325164795)]
[2022/06/09 00:24:21] ppocr INFO: [[[94.0, 390.0], [111.0, 390.0], [111.0, 401.0], [94.0, 401.0]], ('Gyan', 0.6629325151443481)]
[2022/06/09 00:24:21] ppocr INFO: [[[208.0, 381.0], [275.0, 377.0], [276.0, 391.0], [209.0, 395.0]], ('ATALWHNOT', 0.5576425194740295)]
[2022/06/09 00:24:21] ppocr INFO: [[[256.0, 392.0], [355.0, 390.0], [356.0, 404.0], [256.0, 406.0]], ('intheshoppingcenter!', 0.8119673728942871)]
[2022/06/09 00:24:21] ppocr INFO: [[[94.0, 404.0], [113.0, 404.0], [113.0, 415.0], [94.0, 415.0]], ('wenyo', 0.510026752948761)]
[2022/06/09 00:24:21] ppocr INFO: [[[143.0, 402.0], [211.0, 396.0], [213.0, 409.0], [144.0, 416.0]], ('4.A: DIDYOU BUY', 0.6944988965988159)]
[2022/06/09 00:24:21] ppocr INFO: [[[146.0, 417.0], [209.0, 411.0], [210.0, 424.0], [147.0, 430.0]], ('B: NALDDNT', 0.7983251214027405)]
[2022/06/09 00:24:21] ppocr INFO: [[[257.0, 408.0], [338.0, 406.0], [338.0, 419.0], [257.0, 421.0]], ('WAsWerYexpensive.', 0.6711394190788269)]
[2022/06/09 00:24:21] ppocr INFO: [[[235.0, 439.0], [298.0, 435.0], [298.0, 446.0], [235.0, 450.0]], ('hadafun tme:', 0.6346502900123596)]
[2022/06/09 00:24:21] ppocr INFO: [[[145.0, 446.0], [189.0, 441.0], [190.0, 455.0], [146.0, 460.0]], ('B: GREAT!', 0.727346658706665)]
[2022/06/09 00:24:21] ppocr INFO: [[[504.0, 441.0], [712.0, 444.0], [712.0, 457.0], [504.0, 454.0]], ('Lookat thepicture.Makealistoftheweeke', 0.9448941349983215)]
[2022/06/09 00:24:21] ppocr INFO: [[[529.0, 465.0], [620.0, 465.0], [620.0, 478.0], [529.0, 478.0]], ('helpwithhousework', 0.9572677612304688)]
[2022/06/09 00:24:21] ppocr INFO: [[[504.0, 484.0], [512.0, 484.0], [512.0, 493.0], [504.0, 493.0]], ('2.', 0.8789125680923462)]
[2022/06/09 00:24:21] ppocr INFO: [[[242.0, 497.0], [387.0, 497.0], [387.0, 514.0], [242.0, 514.0]], ('(do)somethingveryspecial on', 0.803813219070434 6)]
[2022/06/09 00:24:21] ppocr INFO: [[[100.0, 507.0], [194.0, 501.0], [195.0, 518.0], [101.0, 524.0]], ('LaStAngst', 0.5664306879043579)]
[2022/06/09 00:24:21] ppocr INFO: [[[109.0, 525.0], [185.0, 520.0], [186.0, 534.0], [110.0, 538.0]], ('rschoolt', 0.5626640915870667)]
[2022/06/09 00:24:21] ppocr INFO: [[[231.0, 516.0], [336.0, 512.0], [337.0, 527.0], [231.0, 530.0]], ('(GO) TO MOUNT TAI. WE', 0.7541301250457764)]
[2022/06/09 00:24:21] ppocr INFO: [[[501.0, 526.0], [716.0, 530.0], [716.0, 543.0], [501.0, 539.0]], ('Listenandwritetheactivitiesnexttothecor', 0.960379958152771)]
[2022/06/09 00:24:21] ppocr INFO: [[[589.0, 556.0], [631.0, 556.0], [631.0, 566.0], [589.0, 566.0]], ('exercIse', 0.6330747008323669)]
[2022/06/09 00:24:21] ppocr INFO: [[[239.0, 569.0], [300.0, 569.0], [300.0, 583.0], [239.0, 583.0]], ('(ND) OUT WE:', 0.7355493903160095)]
[2022/06/09 00:24:21] ppocr INFO: [[[353.0, 571.0], [376.0, 571.0], [376.0, 582.0], [353.0, 582.0]], ('(BE:', 0.5392864346504211)]
[2022/06/09 00:24:21] ppocr INFO: [[[518.0, 568.0], [548.0, 568.0], [548.0, 579.0], [518.0, 579.0]], ('usually', 0.7812061309814453)]
[2022/06/09 00:24:21] ppocr INFO: [[[295.0, 586.0], [379.0, 586.0], [379.0, 599.0], [295.0, 599.0]], ('(be)SOTIREDTHAT 1', 0.6550401449203491)]
[2022/06/09 00:24:21] ppocr INFO: [[[518.0, 582.0], [539.0, 582.0], [539.0, 592.0], [518.0, 592.0]], ('OftEN', 0.6590120792388916)]
[2022/06/09 00:24:21] ppocr INFO: [[[99.0, 592.0], [245.0, 585.0], [246.0, 602.0], [100.0, 609.0]], ('weethetopMyegs', 0.6378820538520813)]
[2022/06/09 00:24:21] ppocr INFO: [[[518.0, 595.0], [565.0, 595.0], [565.0, 606.0], [518.0, 606.0]], ('Sometimes', 0.7537936568260193)]
[2022/06/09 00:24:21] ppocr INFO: [[[105.0, 610.0], [219.0, 605.0], [219.0, 619.0], [106.0, 624.0]], ('WantedtostopMydassmates', 0.837291777339417)]
[2022/06/09 00:24:21] ppocr INFO: [[[423.0, 614.0], [462.0, 614.0], [462.0, 629.0], [423.0, 629.0]], ('拍摄', 0.9852978587150574)]
[2022/06/09 00:24:21] ppocr INFO: [[[515.0, 609.0], [566.0, 609.0], [566.0, 620.0], [515.0, 620.0]], ('hardly cver', 0.852721631526947)]
[2022/06/09 00:24:21] ppocr INFO: [[[160.0, 621.0], [376.0, 619.0], [376.0, 639.0], [160.0, 641.0]], ('WEGottotHETOPEvEYO', 0.5547904372215271)]
[2022/06/09 00:24:21] ppocr INFO: [[[513.0, 620.0], [567.0, 623.0], [566.0, 637.0], [512.0, 634.0]], ('H(0%)', 0.7623171806335449)]
[2022/06/09 00:24:21] ppocr INFO: [[[176.0, 658.0], [371.0, 656.0], [371.0, 672.0], [176.0, 675.0]], ('Sstart)tocomeupItwassobeautifulthatwe', 0.8312612175941467)]
[2022/06/09 00:24:21] ppocr INFO: [[[501.0, 657.0], [653.0, 658.0], [653.0, 672.0], [501.0, 670.0]], ('Practicetheconversationinthe', 0.9744216203689575)]
[2022/06/09 00:24:21] ppocr INFO: [[[100.0, 665.0], [128.0, 665.0], [128.0, 674.0], [100.0, 674.0]], ('the sun', 0.7771157622337341)]
[2022/06/09 00:24:21] ppocr INFO: [[[682.0, 661.0], [718.0, 658.0], [719.0, 673.0], [683.0, 676.0]], ('A WHAL', 0.8526968955993652)]
[2022/06/09 00:24:21] ppocr INFO: [[[501.0, 671.0], [651.0, 673.0], [651.0, 686.0], [501.0, 685.0]], ('PictureaboveTenmakeyour', 0.7808661460876465)]
[2022/06/09 00:24:21] ppocr INFO: [[[683.0, 675.0], [720.0, 675.0], [720.0, 689.0], [683.0, 689.0]], ('B:LUSU:', 0.9039231538772583)]
[2022/06/09 00:24:21] ppocr INFO: [[[502.0, 687.0], [648.0, 687.0], [648.0, 700.0], [502.0, 700.0]], ('OWN ConveRSAtionsAbOuT whAT', 0.6443169116973877)]
[2022/06/09 00:24:21] ppocr INFO: [[[683.0, 688.0], [719.0, 691.0], [718.0, 704.0], [682.0, 702.0]], ('ADO Y', 0.8226985931396484)]
[2022/06/09 00:24:21] ppocr INFO: [[[501.0, 701.0], [602.0, 700.0], [602.0, 713.0], [501.0, 714.0]], ('youdoonweekends.', 0.9058097004890442)]
[2022/06/09 00:24:21] ppocr INFO: [[[681.0, 704.0], [718.0, 704.0], [718.0, 718.0], [681.0, 718.0]], ('BNo,l', 0.8917449712753296)]
```

Contents

Units	Functions	Grammar
Unit 1 (P1~16) Stay Healthy	Seeing a Doctor Talking about Health	Modal Verbs: should, need
Unit 2 (P17~32) Great People	Talking about Great People and Their Lives	Adverbial Clause with "before", "after", "as" Attributive Clause with "who", "that"
Unit 3 (P33~48) Safety	Reminding and Warning	Imperatives
Unit 4 (P49~64) Stories and Poems	Enjoying Literature	Past Continuous Tense Modal Verb: must
Unit 5 (P65~80) Look into Science	Talking about Science	Attributive Clause
Unit 6 (P81~96) Movies and Theatre	Talking about Entertainment	Object Complements Passive Voice
Unit 7 (P97~112) Work for Peace	Talking about Disputes and Settlements	Conjunctions: and, but, or, so
Unit 8 (P113~128) Culture Shapes Us	Talking about Cultural Differences	Adverbial Clause with "though" or "although"
Unit 9 (P129~144) Communication	Talking about Interpersonal Communication	
Unit 10 (P145~160) Get Ready for the Future	Talking about the Future and Giving Wishes	

Vocabulary (P161~176)
Structures and Expressions (P177~181)
Irregular Verbs (P182~183)
Grammar (P184~188)

结果如下：

```
ppocr INFO: **********./3.jpg**********
ppocr DEBUG: dt_boxes num : 59, elapse : 1.714942455291748
ppocr DEBUG: rec_res num  : 59, elapse : 0.27222490310668945
ppocr INFO: [[[87.0, 89.0], [472.0, 87.0], [472.0, 150.0], [87.0, 153.0]], ('Contents', 0.9721197485923767)]
ppocr INFO: [[[151.0, 179.0], [209.0, 183.0], [208.0, 206.0], [150.0, 203.0]], ('Units', 0.9863810539245605)]
ppocr INFO: [[[342.0, 180.0], [449.0, 182.0], [449.0, 203.0], [341.0, 201.0]], ('Functions', 0.9862281084060669)]
ppocr INFO: [[[584.0, 178.0], [693.0, 182.0], [693.0, 205.0], [583.0, 201.0]], ('Grammar', 0.975610077381134)]
ppocr INFO: [[[88.0, 226.0], [206.0, 226.0], [206.0, 243.0], [88.0, 243.0]], ('UNit 1(P1-16)', 0.8566757440567017)]
ppocr INFO: [[[288.0, 226.0], [410.0, 226.0], [410.0, 243.0], [288.0, 243.0]], ('Seeing a Doctor', 0.9179188013076782)]
ppocr INFO: [[[520.0, 234.0], [720.0, 234.0], [720.0, 252.0], [520.0, 252.0]], ('Modal Verbs: should', 0.9193161129951477)]
ppocr INFO: [[[87.0, 247.0], [195.0, 247.0], [195.0, 268.0], [87.0, 268.0]], ('Stay Healthy', 0.9721660614013672)]
ppocr INFO: [[[287.0, 245.0], [446.0, 244.0], [447.0, 265.0], [287.0, 267.0]], ('TalkingaboutHealth', 0.8974689841270447)]
ppocr INFO: [[[517.0, 278.0], [758.0, 275.0], [758.0, 296.0], [518.0, 299.0]], ('Adverbial Clause withbeforer', 0.9556236863136292)]
ppocr INFO: [[[87.0, 306.0], [218.0, 306.0], [218.0, 327.0], [87.0, 327.0]], ('Unit 2(P17-32', 0.9372753500938416)]
ppocr INFO: [[[289.0, 307.0], [495.0, 307.0], [495.0, 325.0], [289.0, 325.0]], ('Talking about Great People', 0.9582605957984924)]
ppocr INFO: [[[517.0, 302.0], [613.0, 302.0], [613.0, 320.0], [517.0, 320.0]], ('afteras"', 0.8925629258155823)]
ppocr INFO: [[[85.0, 328.0], [198.0, 328.0], [198.0, 350.0], [85.0, 350.0]], ('Great People', 0.9730417132377625)]
ppocr INFO: [[[289.0, 328.0], [406.0, 328.0], [406.0, 346.0], [289.0, 346.0]], ('and Their Lives', 0.946256697177887)]
ppocr INFO: [[[519.0, 330.0], [752.0, 327.0], [752.0, 348.0], [519.0, 351.0]], ('Attributive Clause withwho"', 0.9534560441970825)]
ppocr INFO: [[[517.0, 355.0], [566.0, 355.0], [566.0, 372.0], [517.0, 372.0]], ('sthat', 0.8308337330818176)]
ppocr INFO: [[[85.0, 389.0], [218.0, 389.0], [218.0, 410.0], [85.0, 410.0]], ('Unit3(P33-48', 0.9275140166282654)]
ppocr INFO: [[[289.0, 402.0], [471.0, 402.0], [471.0, 419.0], [289.0, 419.0]], ('RemindingandWarning', 0.9737339019775391)]
ppocr INFO: [[[516.0, 399.0], [610.0, 396.0], [610.0, 418.0], [517.0, 420.0]], ('Imperatives', 0.9395374655723572)]
ppocr INFO: [[[85.0, 409.0], [142.0, 413.0], [140.0, 436.0], [84.0, 431.0]], ('Safety', 0.9385850429534912)]
ppocr INFO: [[[87.0, 451.0], [220.0, 451.0], [220.0, 472.0], [87.0, 472.0]], ('UNit 4 (P49-64)', 0.8481643199920654)]
ppocr INFO: [[[518.0, 445.0], [690.0, 448.0], [689.0, 465.0], [517.0, 462.0]], ('Past Continuous Tense', 0.9687612652778625)]
ppocr INFO: [[[288.0, 462.0], [437.0, 461.0], [437.0, 482.0], [288.0, 484.0]], ('Enjoying Literature', 0.9888198971748352)]
ppocr INFO: [[[87.0, 476.0], [243.0, 476.0], [243.0, 493.0], [87.0, 493.0]], ('Stories and Poems', 0.9256062507629395)]
ppocr INFO: [[[520.0, 477.0], [656.0, 477.0], [656.0, 495.0], [520.0, 495.0]], ('Modal Verb:must', 0.942721426486969)]
ppocr INFO: [[[89.0, 516.0], [218.0, 516.0], [218.0, 533.0], [89.0, 533.0]], ('UNIT 5 (P65-80)', 0.8806130886077881)]
ppocr INFO: [[[288.0, 524.0], [457.0, 523.0], [458.0, 544.0], [288.0, 545.0]], ('Talking about Science', 0.9228999614715576)]
ppocr INFO: [[[517.0, 522.0], [659.0, 523.0], [659.0, 544.0], [517.0, 543.0]], ('Attributive Clause', 0.983651876449585)]
ppocr INFO: [[[86.0, 535.0], [234.0, 537.0], [234.0, 558.0], [85.0, 557.0]], ('Look into Science', 0.9786447882652283)]
ppocr INFO: [[[88.0, 578.0], [218.0, 578.0], [218.0, 595.0], [88.0, 595.0]], ('Unit 6(P81-96)', 0.9174280166625977)]
ppocr INFO: [[[520.0, 571.0], [678.0, 571.0], [678.0, 589.0], [520.0, 589.0]], ('Object Complements', 0.9539687633514404)]
ppocr INFO: [[[290.0, 586.0], [505.0, 585.0], [505.0, 606.0], [290.0, 607.0]], ('Talking about Entertainment', 0.9468781352043152)]
ppocr INFO: [[[86.0, 597.0], [253.0, 599.0], [252.0, 620.0], [85.0, 619.0]], ('Movies and Theatre', 0.9861759543418884)]
ppocr INFO: [[[517.0, 602.0], [624.0, 602.0], [624.0, 620.0], [517.0, 620.0]], ('Passive Voice', 0.972594141960144)]
ppocr INFO: [[[86.0, 637.0], [228.0, 638.0], [228.0, 660.0], [85.0, 658.0]], ('Unit 7(P97-112)', 0.9131489396095276)]
ppocr INFO: [[[292.0, 640.0], [497.0, 640.0], [497.0, 657.0], [292.0, 657.0]], ('Talking aboutDisputesand', 0.9776421189308167)]
ppocr INFO: [[[520.0, 650.0], [741.0, 650.0], [741.0, 667.0], [520.0, 667.0]], ('Conjunctions:andbut.orso', 0.962110936641693 1)]
ppocr INFO: [[[88.0, 663.0], [218.0, 663.0], [218.0, 681.0], [88.0, 681.0]], ('WorkforPeace', 0.9865747094154358)]
ppocr INFO: [[[292.0, 664.0], [377.0, 664.0], [377.0, 678.0], [292.0, 678.0]], ('Settlements', 0.9926381707191467)]
ppocr INFO: [[[85.0, 700.0], [239.0, 699.0], [239.0, 720.0], [86.0, 722.0]], ('Unit8(P113-128)', 0.9274618029594421)]
ppocr INFO: [[[292.0, 702.0], [460.0, 702.0], [460.0, 719.0], [292.0, 719.0]], ('Talking about Cultural', 0.9409067630767822)]
```

识别结果粗略小结:
结果整体上确实比手写体更加准确;
标点符号存在缺漏（“after”, “as”—>’afteras”）
漏掉部分的单词识别（例如need）
单词之间有时无法识别空格
不是一次完整识别完一个“块”（例如表格的一整个单元格）识别的顺序和人的阅读习惯存在差异。

印刷体测试问题分析

编写如下程序`self.py`:

```
from paddleocr import PaddleOCR,draw_ocr
# Paddleocr supports Chinese, English, French, German, Korean and Japanese.
# You can set the parameter `lang` as `ch`, `en`, `fr`, `german`, `korean`,
`japan`
# to switch the language model in order.
ocr = PaddleOCR(use_angle_cls=True, lang='en') # need to run only once to
download and load model into memory
img_path = './train_data/en_hand_writing/4.jpg'
result = ocr.ocr(img_path, cls=True)
for line in result:
    print(line)

# draw result
from PIL import Image
image = Image.open(img_path).convert('RGB')
boxes = [line[0] for line in result]
txts = [line[1][0] for line in result]
scores = [line[1][1] for line in result]
im_show = draw_ocr(image, boxes, txts, scores, font_path='/home/zima-
blue/learning/ZKWG/paddleocr_backup/PaddleOCR/doc/fonts/simfang.ttf')
im_show = Image.fromarray(im_show)
im_show.save('result4.jpg')
```

运行`self.py`文件，统计在原始paddleocr下四张示例图片的准确率：

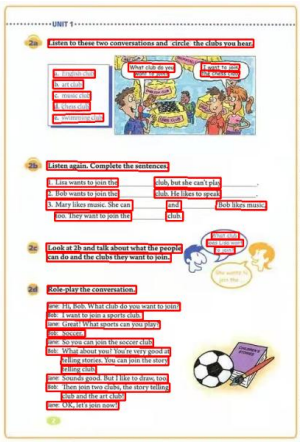
图1：199/396，50.25%

```
1: ••• 0.628
2: UNIT1 0.955
3: UNIT 2 0.912
4: Hi Poullm Anno 0.804
5: Where did you go on 0.940
6: Seetion 0.914
7: vocation? 0.935
8: How often do you 0.933
9: Oh,Iwent to Beling 0.807
10: pads un.op n pig 0.576
11: in Nanjing 0.853
12: Norying 0.848
13: Language Goal: 0.965
14: Talk about how 0.958
15: Self Check 0.927
16: often you do 0.962
17: things 0.997
18: 1.ADid 0.887
19: What do y 0.925
20: anything 0.908
21: loftengo to 0.930
22: oonwe 0.940
23: eveyching 0.884
24: BNo 0.867
25: nothing 0.921
26: the movies. 0.919
27: B: 0.894
28: at all? Why not? 0.867
29: atyone 0.849
30: in the shopping center? 0.963
31: evegone 0.936
32: 4.A:Did you buy 0.890
33: no one 0.940
34: was very expensive. 0.954
35: 5A:Howwas the volleyall game yesterday 0.793
36: had a fun time! 0.969
37: B:Great 0.877
38: Look at the picture.Make a list of the weeken 0.941
39: 1. 0.778
40: help with housework 0.943
41: 2. 0.977
42: do something very special on 0.913
43: 3. 0.952
44: Last Augustour class 0.896
45: go to Mount Tai. We 0.909
46: our school trip We 0.818
47: Listen and write the activities next to the cor 0.955
48: startour trip at 12:00 at
night.Everyone in our class 0.862
49: always(100% 0.919
50: exercise 0.997
51: (be, 0.971
52: find)out we 0.934
53: looked at the map and 0.913
54: usually 0.995
55: THE 0.534
56: beso tired that 1 0.886
57: often 0.991
58: noanywhere near the topMy leg 0.918
59: tell) me to keep going. 0.852
60: sometimes 0.962
61: wanted to stop. My classmates 0.848
62: hardly ever 0.875
63: AE 0.505
64: goon.At 5:00 a.m.we got to the top!Everyone 0.841
65: sol 0.640
66: ner0% 0.846
67: (jump) up and down in exci
tement.Twenty minutes later, 0.882
68: start) to come up.It was so beautiful that we 0.896
69: Practice the conversation in the 0.972
70: the sun 0.931
71: picture above.Then make your 0.962
72: A:What 0.909
73: forgetabout the last five hours 0.929
74: g:luou 0.693
75: own conversations about what 0.976
76: ADoy 0.938
77: you do on weekends. 0.943
78: No,I 0.723
```

图2：150/168，89.29%

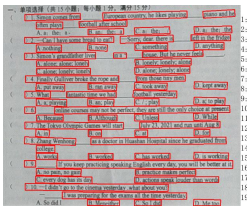
```
1: Contents 0.931
2: Functions 0.992
3: Units 0.995
4: Grammar 0.997
5: Unit 1P1~16 0.919
6: Seeing a Doctor 0.959
7: Modal Verbs:should.need 0.947
8: Stay Healthy 0.960
9: Talking about Health 0.958
10: Adverbial Clause withbefore" 0.953
11: Unit 2(P17~32 0.925
12: Talking about Great People 0.955
13: "after""as" 0.801
14: Great People 0.973
15: and Their Lives 0.953
16: Attributive Clause with "who 0.931
17: "that" 0.981
18: Unit 3(P33~48 0.939
19: Reminding and Warning 0.960
20: Imperatives 0.997
21: Safety 0.995
22: Unit 4P49~64 0.949
23: Past Continuous Tense 0.961
24: Enjoying Literature 0.968
25: Stories and Poems 0.958
26: Modal Verbmust 0.955
27: Unit 5(P65~80 0.939
28: Talking about Science 0.963
29: Attributive Clause 0.977
30: Look into Science 0.964
31: Unit 6(P81~96 0.927
32: Object Complements 0.970
33: Movies and Theatre 0.948
34: Talking about Entertainment 0.963
35: Passive Voice 0.967
36: Unit 7P97~112 0.927
37: Talking about Disputes and 0.960
38: Conjunctions: and,but. or,so 0.855
39: Work for Peace 0.966
40: Settlements 0.989
41: Talking about Cultural 0.942
42: Unit 8(P113~128 0.908
43: Adverbial Clause withthough" 0.949
44: Differences 0.994
45: Culture Shapes Us 0.959
46: or"although' 0.903
47: Unit 9P129~144 0.966
48: Talking about Interpersonal 0.967
49: Communication 0.998
50: Communication 0.998
51: Unit 10(P145~160 0.913
52: Talking about the Future and 0.957
53: Get Ready for the 0.917
54: Giving Wishes 0.962
55: Future 0.938
56: VocabularyP161~176 0.979
57: Structures and Expressions(P177~181) 0.947
58: Irregular Verbs(P182~183 0.937
59: Grammar(P184~188 0.972
```

图3：166/212，78.3%



```
1: Listen to these two conver
sations and circle the clubs you hear. 0.960
2: What club do you 0.957
3: I want to join 0.959
4: a.English club 0.911
5: want to join? 0.923
6: the chess club 0.945
7: b.art club 0.925
8: c. music club 0.832
9: d.chess cluh 0.887
10: e. swimming club 0.901
11: Listen again. Complete the sentences. 0.940
12: 1.Lisa wants to join the 0.941
13: club,but she cant play 0.885
14: 2.Bob wants to join the 0.936
15: club. He likes to speak 0.909
16: 3.Mary likes music. She can 0.948
17: and 0.999
18: Bob likes music 0.950
19: too. They want to join the 0.932
20: club. 0.988
21: What.club 0.912
22: Look at 2b and talk about what the people 0.955
23: does Lisa want 0.915
24: to join? 0.904
25: can do and the clubs they want to join. 0.954
26: Role-play the conversation. 0.984
27: Jane:Hi,Bob.What club do you want to join 0.928
28: Bob:I want to join a sports club 0.945
29: Jane:Great!What sports can you play 0.925
30: BobSoccer. 0.957
31: Jane:So you can join the soccer club. 0.932
32: Bob: What about you? You're very good at 0.926
33: telling stories. You can join the story 0.925
34: telling club. 0.938
35: Jane:Sounds good, But I like to draw,too. 0.895
36: Bob Then join two clubs,the story telling 0.927
37: club and the art club! 0.944
38: JaneOK,let's join now! 0.926
```

图4：195/255，76.47%



```
1: European country, he likes pla ying. 0.946
2: piano and he 0.999
3: YiShen comes from 0.859
4: often plays 0.980
5: football after school 0.955
6: C.a then. 0.687
7: B.anie hehsa 0.670
8: Da a ther the 0.780
9: Sorry, dear there is 0.859
10: SZh Can I have someibread lo cat? 0.815
11: lefl in the fridge 0.876
12: Cosomething 0.950
13: Binone 0.676
14: A nothing 0.909
15: D anything 0.986
16: insc. 0.784
17: eSimon s grandfather lives 0.890
18: house But he never feels 0.934
19: Aalone alone lonely 0.823
20: B lonely: Jonelyalone 0.854
21: Ghalone toncly loncly 0.792
22: Balone lonely.alone 0.883
23: J4Finally Gulliver broke the rope and 0.944
24: from those tiny men. 0.956
25: Chook away 0.886
26: Avput away 0.899
27: Bran away 0.979
28: D. kept away 0.926
29: SWhat 0.882
30: fantastic time we had 0.927
31: football yesterday. 0.967
32: B an play 0.808
33: C./*play 0.787
34: A playing 0.923
35: D. a to play 0.885
36: 16 0.654
37: online courses may not be
perfect, they are still th
e only choice at present. 0.967
38: C Unless 0.851
39: A.Because 0.949
40: B Although 0.946
41: D While 0.902
42: DaThe Tokyo Olympic Games will start 0.880
43: July 23 2021 and run unti Aug 8 0.952
44: Ain 0.952
45: B.on 0.939
46: G.at 0.655
47: D for 0.871
48: Y B.Zhang Wenhong 0.908
49: as a doctor in Huashan Hos
pital since he graduated from 0.953
50: college. 0.937
51: A works 0.956
52: C. has worked 0.902
53: Baworked 0.920
54: D. is working 0.936
55: 9. 0.737
56: If you keep practicing spe
aking English every day, y
ou will be better at it. 0.929
57: A no pain, no gain 0.922
58: B.practicemakes.iperiect 0.885
59: Cevery dog has its day 0.930
60: Diactions speaklouder than words 0.914
61: Yi 10 yo- I didn't go to t
he cinoma yesterday what aoout you? 0.851
62: I was preparing for the ex
ams all the time yesterda y. 0.946
63: B. Wanciher 0.776
64: Ga SoDid 0.770
65: D Metoo 0.826
```

出于实训安排的原因，目光聚焦于印刷体识别中存在的典型问题：

· 背景干扰问题；

· 文字密集，导致检测框定文本框时无法很好的进行分类；

· 照片清晰度的干扰；

· 图片弯曲导致文本框框定的问题：

■ 识别结果小结：

单词之间没有空格；

大小写识别失误；

单独的字母识别失误，例如go识别成qo；

莫名其妙多识别字符（例如world—>worldl）

1. 受到文字颜色、大小、字体、形状、方向以及文本长度的影响；
2. 可能会有模糊，阴影，亮度等因素的影响；
3. 文本密集甚至重叠会影响文字的检测；

Model Training

模型选择

Detection Model

着重针对文字检测中“任意形状文本检测，弯取文本检测”、“密集文本没有办法很好进行”分离等问题进行分析，选取针对性强的模型进行训练和测试。

最终选取核心算法为SAST的模型。

baseline原始模型的detection model使用的核心算法是DB算法，是基于分割的文本检测算法目前都有如下共同的问题（如图1）：1）文本实例相互靠近，难以通过语义分割来分离文本实例；2）长文本实例容易被分割

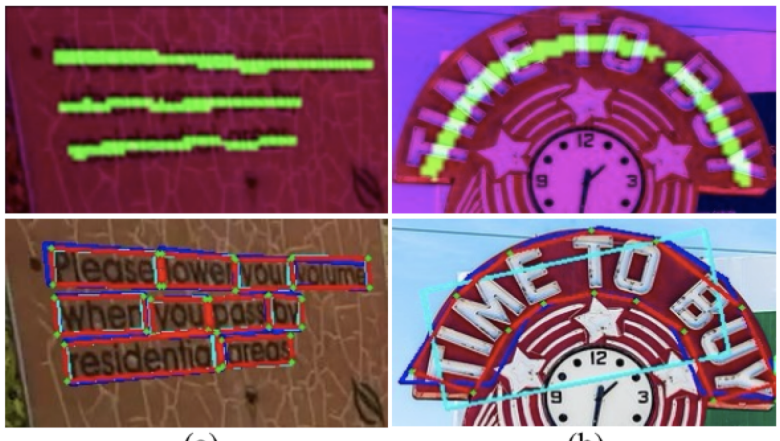


Figure 1: Two common challenges for segmentation-based text detectors: (a) Adjacent text instances are difficult to be separated; (b) The long text may break into fragments. The first row is the response for text regions from segmentation branch. In the second row, cyan contours are the detection results from EAST [47], while red contours are from SAST.

而SAST，即是一种基于语义分割的文本检测器，使用基于FCN的框架并进行预测文本特征，使用高效的点对齐方法，能够准确高效的检测任意长度的文本。

网络架构：

backbone：Resnet50的FPN（特征金字塔网络），将最后的全连接层以FCN全卷积层替代。其中CAB模块产出了特征增强表示。每个文本实例都会得到TCL（文本中心线）、TCO（文本中心偏移）、TVO（文本顶点偏移）和TBO（文本边界偏移）特征图。在后处理部分，使用点对边对齐方法来分割文本实例(text instance segmentation)。TVO特征图返回4个文本区域的顶点，这四个顶点特征，被视作文本区域的高层目标中心。通过计算检测文本框中的低层中心和高层目标中心的距离，TCL特征图中的像素能够被分成几组文本实例。之后在文本实例的中心线上选取合适的数量点，借助TBO映射，计算上下边界的对应点，最终实现任意形状文本的detection。

基本架构如下图所示（图3）：

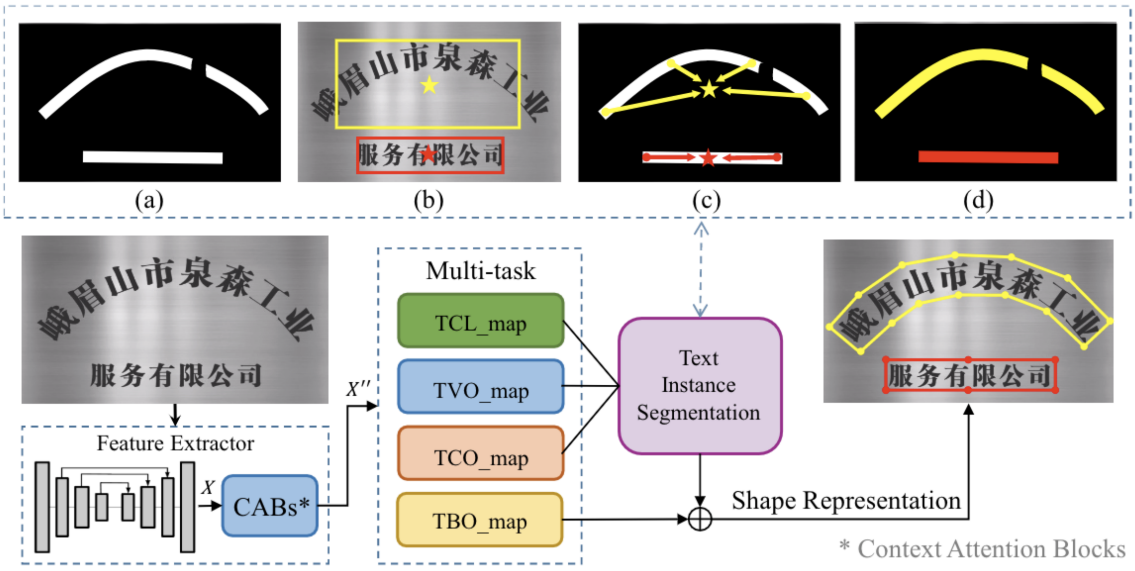


Figure 3: The pipeline of proposed method: 1) Extract feature from input image, and learn TCL, TBO, TCO, TVO maps as a multi-task problem; 2) Achieve instance segmentation by Text Instance Segmentation Module, and the mechanism of point-to-quad assignment is illustrated in c; 3) Restore polygonal representation of text instances of arbitrary shapes.

CAB是什么：Context Attention Block

因为基于FCN的文本检测器受限于局部感受野和短程上下文信息。因此我们设计了一个Context Attention Block来计算长程相关性来获得有效特征。（如图4）

对于特征图X，是backbone的输出，尺寸是NHWC。三个紫色的卷基层（1*1）是为了收集水平上下文信息；三个蓝色的卷基层是为了获取垂直上下文特征。中间的段路径是为了保持局部特征。

最终，原始的X经过两个CAB之后，每个像素最终可以捕获所有像素的依赖关系，得到一个更有效的内容增强特征图，从而避免在阅读长文本/弯取文本时感受野有限造成的问题

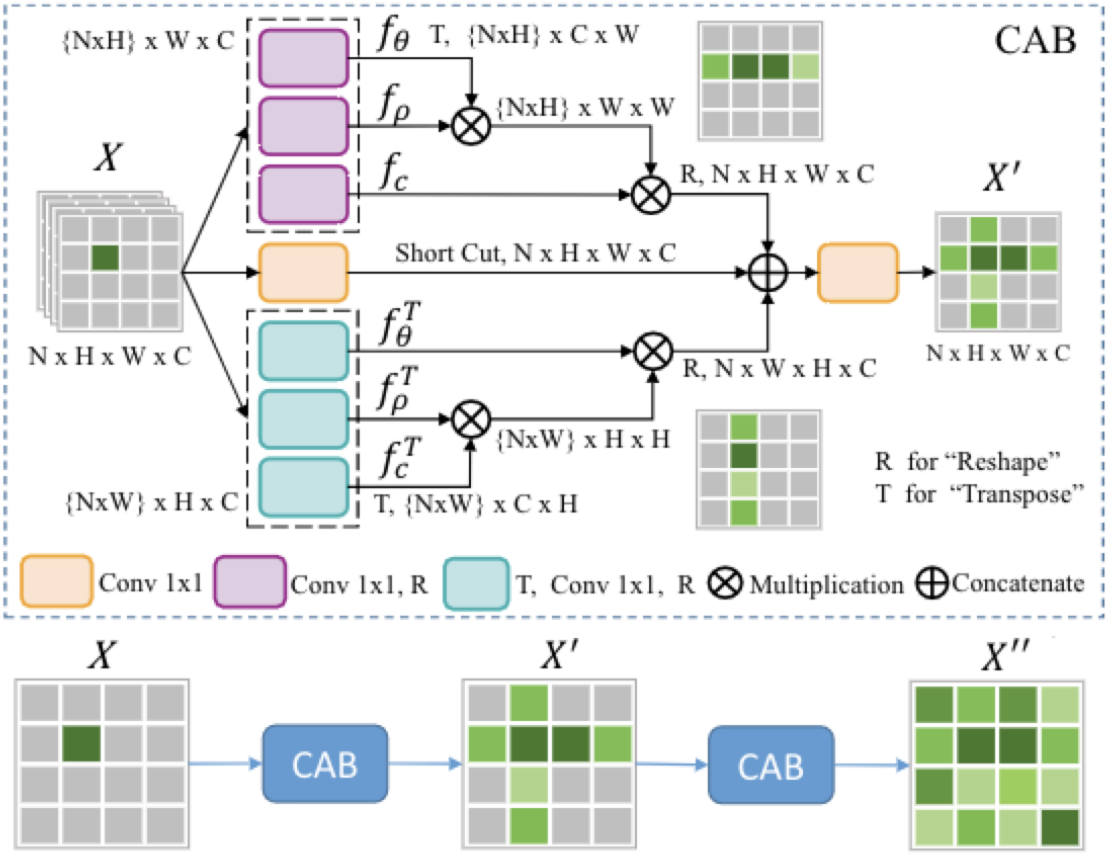
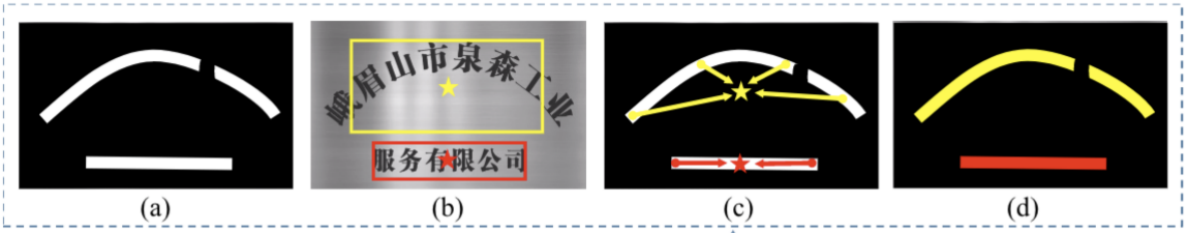


Figure 4: Context Attention Blocks: a single CAB module aggregates pixel-wise contextual information both horizontally and vertically, and long-range dependencies from all pixels can be captured by serially connecting two CABs.

Text Instance Segmentation:

文本实例分割的第一步是利用TCL和TVO特征图检测出候选文本四边形。将TCL特征图二值化，所有的像素值在[0, 1]之间，给定一个阈值，使用TVO特征图提供的四个顶点偏移来恢复相应的四边形边界。得到（b）。最后再将二进制TCL特征图中的映射文本区域聚类成为文本实例。如图（c）所示。



模型架构：

```
Architecture:
  model_type: det
  algorithm: SAST
  Transform:
  Backbone:
    name: ResNet_SAST
    layers: 50
  Neck:
    name: SASTFPN
```

```
    with_cab: True
  Head:
    name: SASTHead

Loss:
  name: SASTLoss

Optimizer:
  name: Adam
  beta1: 0.9
  beta2: 0.999
  lr:
  #  name: Cosine
    learning_rate: 0.001
  #  warmup_epoch: 0
  regularizer:
    name: 'L2'
    factor: 0
```

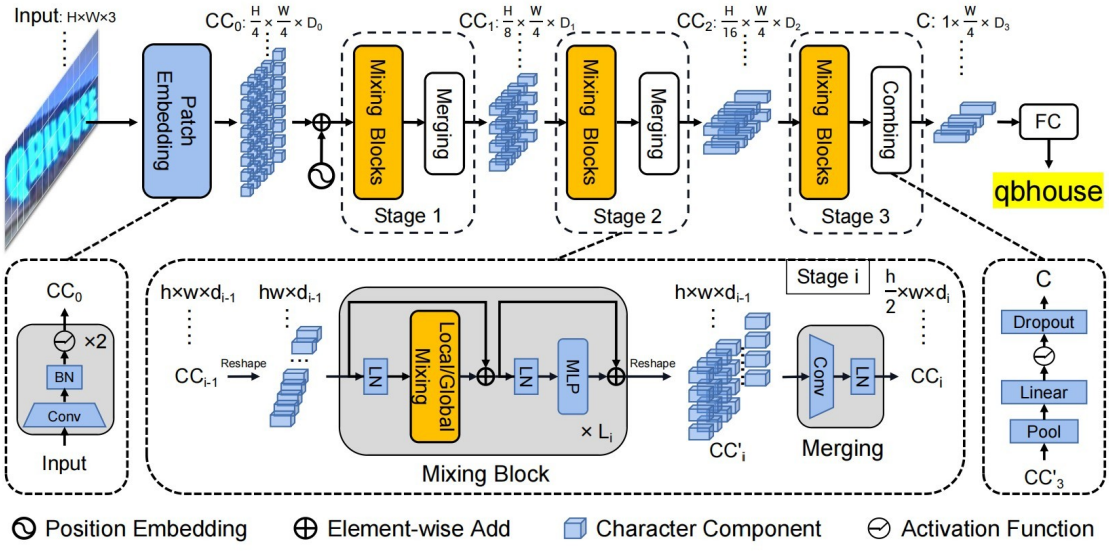
Recognition Model

onfig: [en_PP-OCRv3_rec.yml]();

核心算法：SVTR;

选取模型的原因：保留轻量级结构设计的同时去掉了LSTM，在这个实训期间，训练和部署更加便捷。

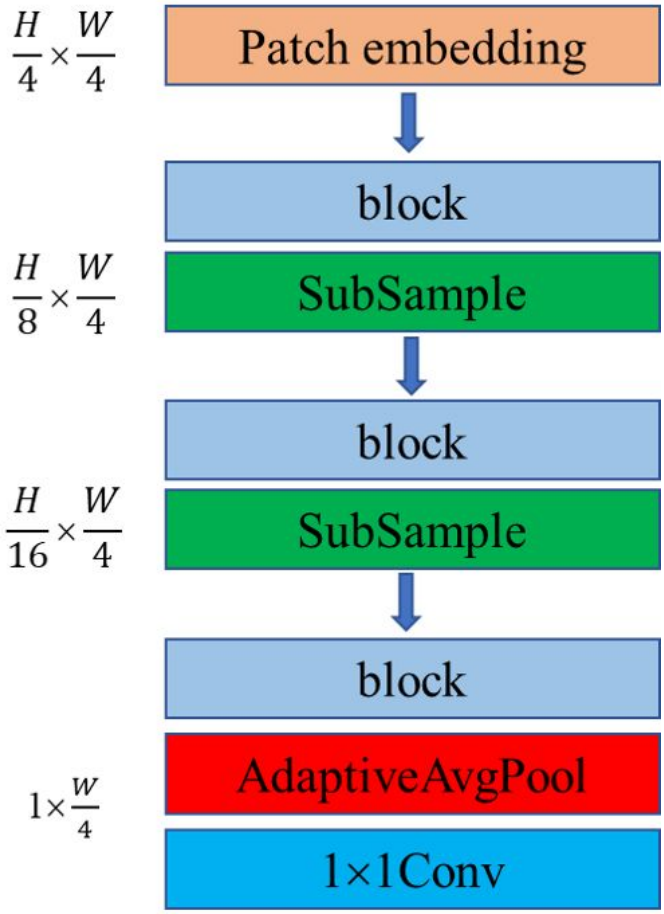
[参考网址1]()；[参考网址2]();



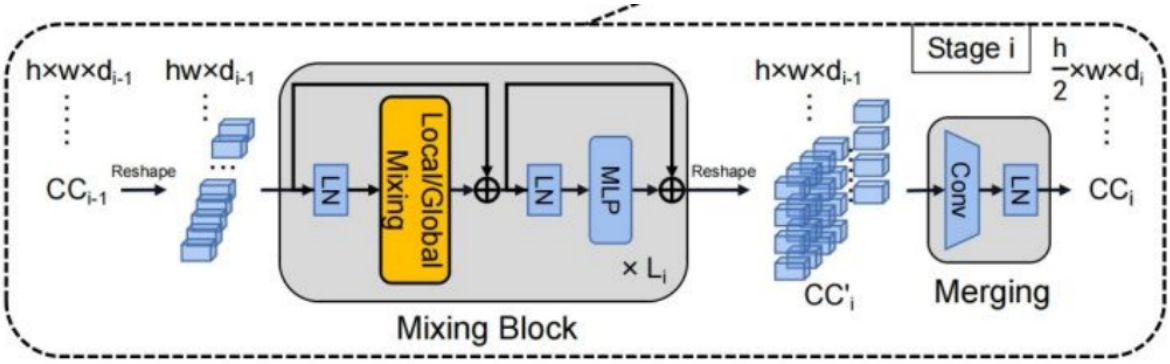
SVTR的具体结构如下：

Patch Embedding，SVTR使用两个步长为2的3×3卷积进行有重叠的patch embedding（延续的CNN的作风），从而感受野更大，提取局部信息的表达能力也会比swin的patch embedding要好（swin是直接使用一个步长为4的4×4卷积进行无重叠的patch embedding）。

下采样模块。延续了paddleocr里面的CRNN下采样结构。其中的SubSample模块是一个步长为（2，1）核大小为3×3的普通卷积，只对高度进行下采样，而非宽度。有两个原因：（1）宽维度所包含的文字信息笔记丰富，下采样会造成较多信息的丢失；（2）CTC解码前的Argmax序列输出长度越大，结果越稀疏，则包含的空字符越多，那么CTC解码时**连续相同的文字**就很大程度上可以被保留。



MixBlock：全局Mix与局部Mix（目的：使用不同接收场的self-attention来感知上下文的相关性）
Global Mixing评估字符之间的依赖性。由于文本和非文本是图像中的两个主要元素，这种通用的Mixing可以建立来自不同字符的组件之间的长期依赖关系。此外，它还能削弱非文本成分的影响，同时提高文本成分的重要性。



Local Mixing。如(b)所示，Local mixing评估了预定义窗口内组件之间的相关性。其目的是对形态特征进行编码，并建立特征内成分之间的关联，从而模拟对特征识别至关重要的笔画样例特征。
与Global Mixing不同，Local mixing考虑的是每个分量都有一个邻域。与卷积类似，混合是以滑动窗口的方式进行。窗口大小根据经验设置为7×11。

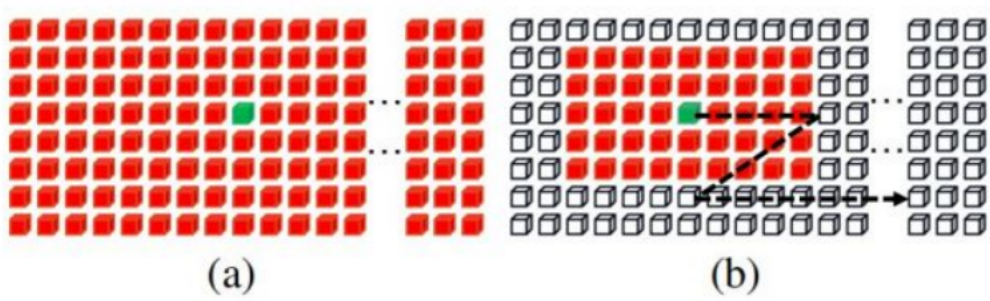


Figure 4: Illustration of (a) global mixing and (b) local mixing.

Merging。即下采样操作，和卷积的下采样一样。因为self-attention的计算量和特征图的宽高有关，宽高太大的话，计算复杂度暴涨，所以SVTR对其进行了下采样操作，在低分辨率的特征图上计算可以减少矩阵乘法的计算复杂度。

最后的部分，使用Combining进行维度压缩：首先将高度维度全局池化为1；然后是经过全连接层处理；字符被进一步压缩为一个特征序列。

数据集选取

使用数据集ICDAR2015。

此数据集样本较少，但是似乎对实训而言比较友好。包含1000张训练图像和500张测试图像；部分示例如下：

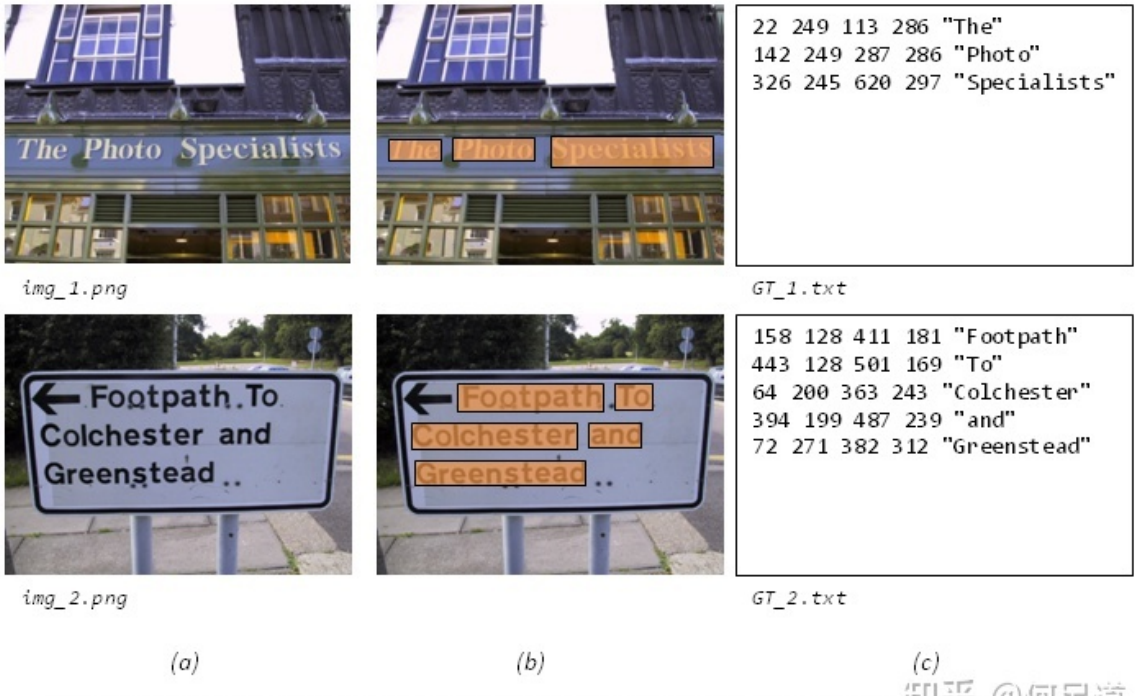


Figure 1. *(a) Original Images, (b) Visualisation of the ground truth, (c) Ground Truth text file.*

在阅读代码框架之后，新建文件夹下载数据，数据集组织如下：

```
/PaddleOCR/train_data/icdar2015/text_localization/
  └─ icdar_c4_train_imgs/         icdar数据集的训练数据
  └─ ch4_test_images/             icdar数据集的测试数据
  └─ train_icdar2015_label.txt    icdar数据集的训练标注
  └─ test_icdar2015_label.txt     icdar数据集的测试标注
```

训练过程(以Detection Model为例)

下载预训练模型：

```
wget -P ./pretrain_models/
https://paddleocr.bj.bcebos.com/dygraph_v2.0/en/det_r50_vd_sast_totaltext_v2.0_tr
ain.tar
```

开始训练：

```
python3 tools/train.py -c configs/det/det_r50_vd_sast_totaltext.yml \
     -o
Global.pretrained_model=./pretrain_models/./trained_models/det_r50_vd_sast_idar20
15
```

中途报错。Out of memory error on GPU 0。显示选择kill了进程。结果可能是权限赋予错误ubuntu重启连不上网了...备份重装系统；选择第二个办法：修改batch-size。

修改batch-size之后成功，训练过程截图：

```
samples: 80, ips: 9.93143
root INFO: epoch: [2/1200], iter: 160, lr: 0.001000, loss: 5.919452, loss_shrink_maps: 4.036129,
samples: 80, ips: 9.89948
root INFO: epoch: [2/1200], iter: 170, lr: 0.001000, loss: 5.775730, loss_shrink_maps: 3.924310,
samples: 80, ips: 9.98633
root INFO: epoch: [2/1200], iter: 180, lr: 0.001000, loss: 5.686600, loss_shrink_maps: 3.848886,
samples: 80, ips: 9.95261
root INFO: epoch: [2/1200], iter: 190, lr: 0.001000, loss: 5.562242, loss_shrink_maps: 3.743180,
samples: 80, ips: 9.75551
root INFO: epoch: [2/1200], iter: 200, lr: 0.001000, loss: 5.017828, loss_shrink_maps: 3.333572,
samples: 80, ips: 9.84531
root INFO: epoch: [2/1200], iter: 210, lr: 0.001000, loss: 5.017828, loss_shrink_maps: 3.333572,
samples: 80, ips: 9.87709
root INFO: epoch: [2/1200], iter: 220, lr: 0.001000, loss: 5.173910, loss_shrink_maps: 3.449896,
```

转移至九天毕昇平台上进行训练；

训练过程中的基本概念介绍见此。

learning rate:梯度在每次迭代中向损失函数的最优解移动的步长。 learning_rate.py代码实现了不同的strategies。

Regularization：正则化，为了有效避免算法过拟合。使用L1和L2正则化方法。L1的目的是为了减少参数绝对值的和；L2是为了减少参数平方和。

Evaluation Indicators：评价指标。

Detection stage：首先根据检测框和标注框的IOU进行评估。如果IOU大于某个阈值，则判断检测准确。这里的检测框和标签框用多边形表示。检测精度：所有检测帧中正确检测帧数的百分比，主要用于判断检测指标。检测召回率：正确检测帧在所有标记帧中的百分比。

Recognition stage：字符识别准确率，即正确识别文本行数与标记文本行数之比。

End-to-end statistics：端到端召回率：准确检测并正确识别文本行在所有标注文本行中的比例；端到端准确率：准确检测并正确识别检测到的文本行中的文本行数

模型测试

detection model

config：det_r50_vd_sast_icdar2015.yml；

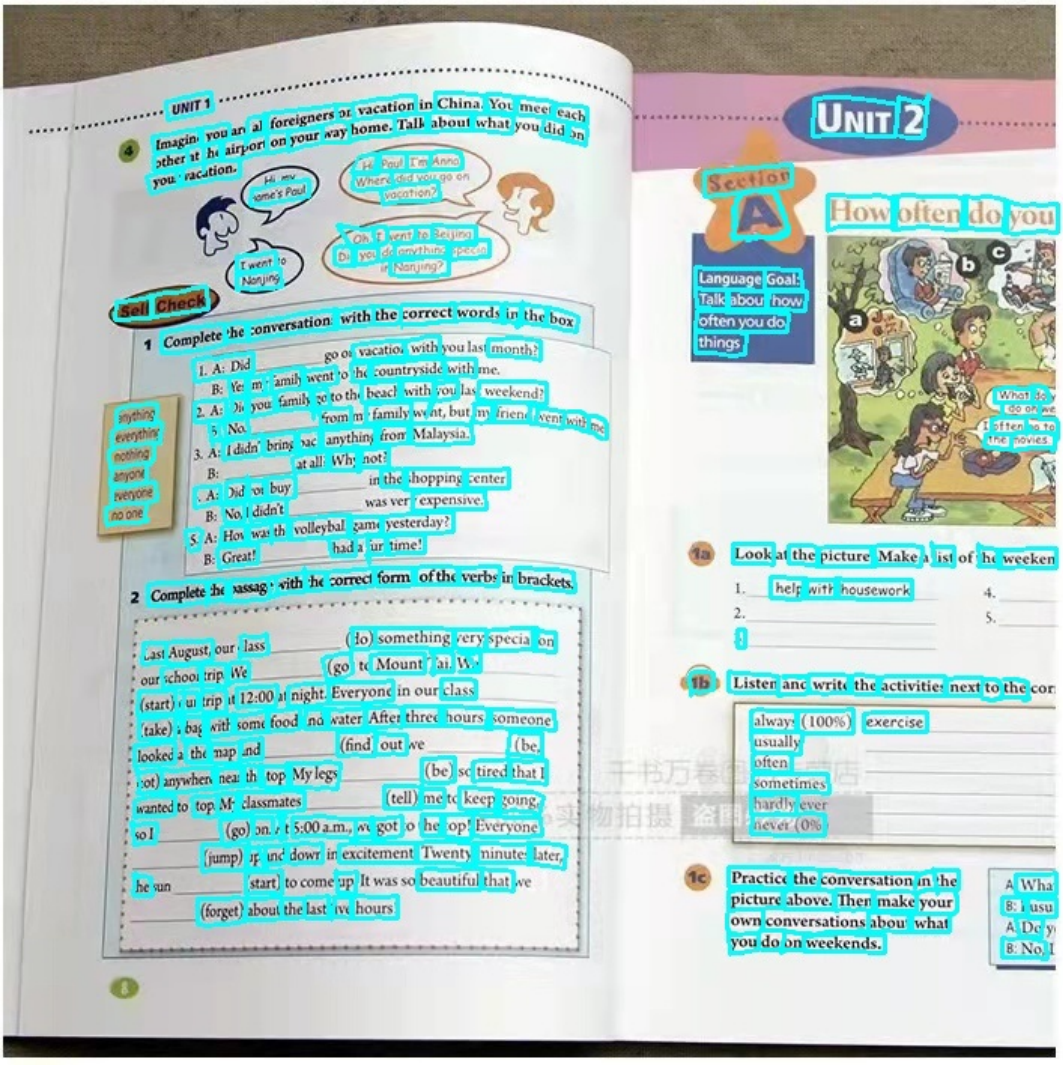
在数据测试集上进行测试：

```
python tools/eval.py -c configs/det/det_r50_vd_sast_icdar15.yml -o
Global.checkpoints=./trained_models/det_r50_vd_sast_idar2015_v2.0_train/best_accu
racy
```

```
[2022/06/20 18:23:34] ppocr INFO: resume from ./new_train/det_r50_vd_sast_icdar15_v2.0_train/best_acc
uracy
[2022/06/20 18:23:34] ppocr INFO: metric in ckpt ***************
[2022/06/20 18:23:34] ppocr INFO: hmean:0.8741522230595328
[2022/06/20 18:23:34] ppocr INFO: precision:0.9138655462184874
[2022/06/20 18:23:34] ppocr INFO: recall:0.8377467501203659
[2022/06/20 18:23:34] ppocr INFO: fps:2.5292118138203126
[2022/06/20 18:23:34] ppocr INFO: best_epoch:4188
[2022/06/20 18:23:34] ppocr INFO: start_epoch:4189
eval model:: 100%|                                                  | 500/500 [02:55<00:00,  2.84it/s]
[2022/06/20 18:26:30] ppocr INFO: metric eval ***************
[2022/06/20 18:26:30] ppocr INFO: precision:0.9138655462184874
[2022/06/20 18:26:30] ppocr INFO: recall:0.8377467501203659
[2022/06/20 18:26:30] ppocr INFO: hmean:0.8741522230595328
[2022/06/20 18:26:30] ppocr INFO: fps:4.61120329575511
```

```
#预测指令
python3 tools/infer/predict_det.py --det_algorithm="SAST" --
image_dir="./train_data/en_hand_writing/1.jpg" --
det_model_dir="./trained_models/det_r50_vd_sast_idar2015_infer" --
det_sast_polygon=True
```

效果如下：



UNIT 1

4 Imagine you are all foreigners on vacation in China. You meet each other at the airport on your way home. Talk about what you did on your vacation.

Hi, my name's Paul.
Hi Paul. I'm Anna. Where did you go on vacation?
I went to Nanjing.
Oh, I went to Beijing. Do you do anything special in Nanjing?

Self Check

1 Complete the conversations with the correct words in the box.

anything everything nothing anyone everyone no one

1. A: Did _____ go on vacation with you last month?
   B: Yes, my family went to the countryside with me.
2. A: Did you family go to the beach with you last weekend?
   B: No, _____ from my family went, but my friend went with me.
3. A: I didn't bring back anything from Malaysia.
   B: _____ at all. Why not?
4. A: Did you buy _____ in the shopping center?
   B: No, I didn't _____ was very expensive.
5. A: How was the volleyball game yesterday?
   B: Great! _____ had a fun time!

2 Complete the passage with the correct form of the verbs in brackets.

Last August, our class _____ (do) something very special on our school trip. We _____ (go) to Mount Tai. We _____ (start) our trip at 12:00 at night. Everyone in our class _____ (take) a bag with some food and water. After three hours, someone looked at the map and _____ (find) out we _____ (be) not anywhere near the top. My legs _____ (be) so tired that I wanted to stop. My classmates _____ (tell) me to keep going, so I _____ (go) on. At 5:00 a.m., we got to the top! Everyone _____ (jump) up and down in excitement. Twenty minutes later, the sun _____ (start) to come up. It was so beautiful that we _____ (forget) about the last five hours.

8

UNIT 2

Section A

Language Goal: Talk about how often you do things

How often do you

1a Look at the picture. Make a list of the weeken

1. help with housework
2. _____
3. _____
4. _____
5. _____

1b Listen and write the activities next to the cor

always (100%) exercise
usually
often
sometimes
hardly ever
never (0%)

1c Practice the conversation in the picture above. Then make your own conversations about what you do on weekends.

A: Wha
B: I usu
A: Do y
B: No, I

Contents

2a Listen to these two conversations and circle the clubs you hear.

a. English club
b. art club
c. music club
d. chess club
e. swimming club

2b Listen again. Complete the sentences.

1. Lisa wants to join the _____ club but she can't play _____.
2. Bob wants to join the _____ club. He likes to speak _____.
3. Mary likes music. She can _____ and _____. Bob likes music, too. They want to join the _____ club.

2c Look at 2b and talk about what the people can do and the clubs they want to join.

2d Role-play the conversation.

Jane: Hi, Bob. What club do you want to join?
Bob: I want to join a sports club.
Jane: Great! What sports can you play?
Bob: Soccer.
Jane: So you can join the soccer club.
Bob: What about you? You're very good at telling stories. You can join the story telling club.
Jane: Sounds good. But I like to draw too.
Bob: Then join two clubs, the story telling club and the art club!
Jane: OK. Let's join now!

一、单项选择（共 15 小题；每小题 1 分，满分 15 分）

( ) 1. Simon comes from _____ European country. he likes playing _____ piano and he often plays _____ football after school.
A. a; the; a - B. an; the; a C. a; the; / D. a; the; the

( ) 2. —Can I have some bread to eat? —Sorry, dear, there is _____ left in the fridge.
A. nothing B. none C. something D. anything

( ) 3. Simon's grandfather lives _____ in a _____ house. But he never feels _____.
A. alone; alone; lonely B. lonely; lonely; alone
C. alone; lonely; lonely D. alone; lonely; alone

( ) 4. Finally Gulliver broke the rope and _____ from those tiny men.
A. put away B. ran away C. took away D. kept away

( ) 5. What _____ fantastic time we had _____ football yesterday.
A. a; playing B. an; play C. /; play D. a; to play

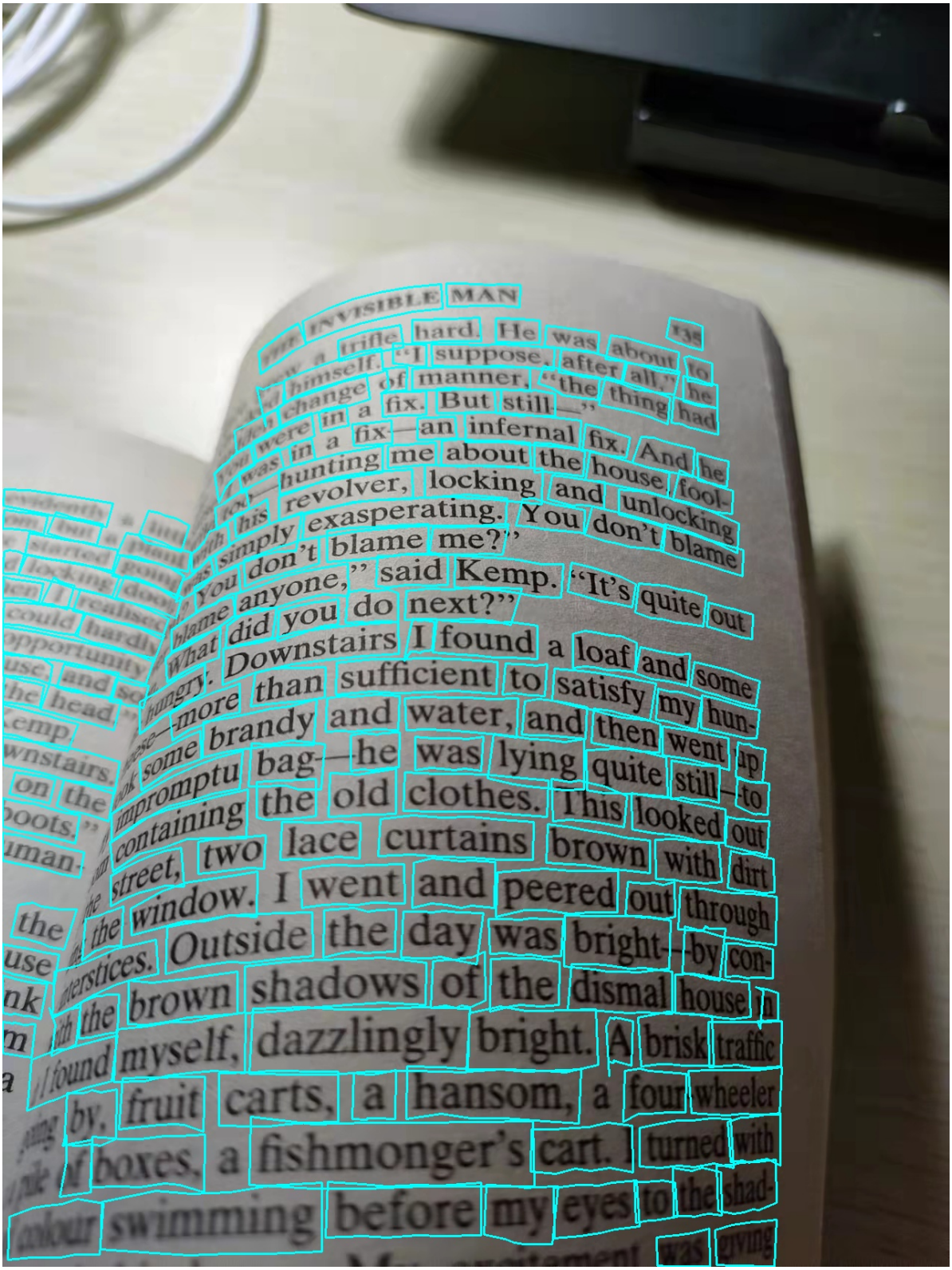
( ) 6. _____ online courses may not be perfect, they are still the only choice at present.
A. Because B. Although C. Unless D. While

( ) 7. The Tokyo Olympic Games will start _____ July 23, 2021 and run until Aug 8.
A. in B. on C. at D. for

( ) 8. Zhang Wenhong _____ as a doctor in Huashan Hospital since he graduated from college.
A. works B. worked C. has worked D. is working

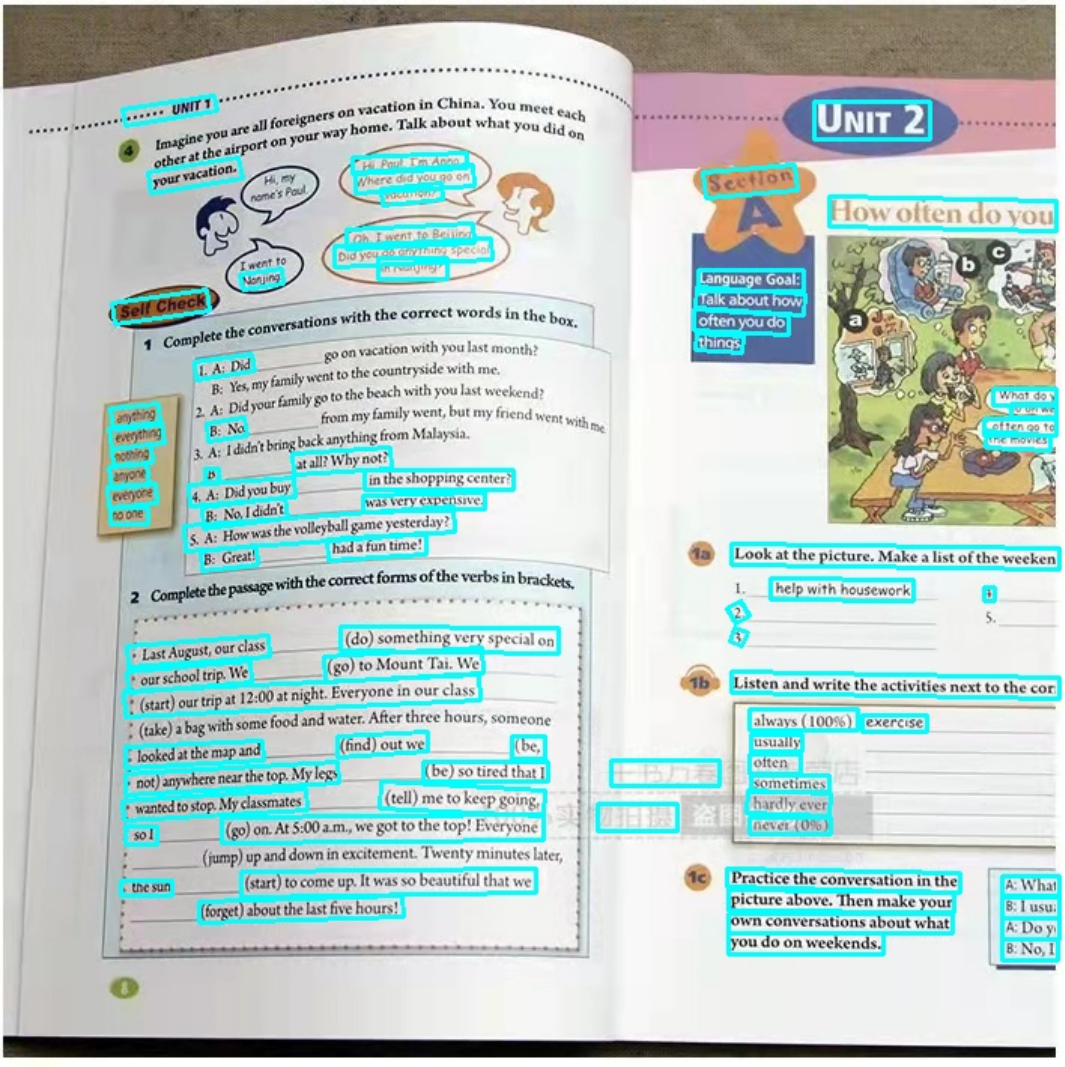
( ) 9. _____. If you keep practicing speaking English every day, you will be better at it.
A. no pain, no gain B. practice makes perfect
C. every dog has its day D. actions speak louder than words

( ) 10. —I didn't go to the cinema yesterday, what about you?
—_____. I was preparing for the exams all the time yesterday.
A. So did I B. Me neither C. So I did D. Me too



相比于默认模型：en_PP-OCRv3_det_slim_infer：

```
python3 tools/infer/predict_det.py --det_algorithm="DB" --
det_model_dir="./trained_models/en_PP-OCRv3_det_slim_infer/" --
image_dir="./train_data/en_hand_writing/" --use_gpu=True
```



UNIT 1

4 Imagine you are all foreigners on vacation in China. You meet each other at the airport on your way home. Talk about what you did on your vacation.

Hi, my name's Paul.
Hi, Paul. I'm Anna. Where did you go on vacation?
Oh, I went to Beijing. Did you do anything special in Beijing?
I went to Nanjing.

Self Check

1 Complete the conversations with the correct words in the box.

anything
everything
nothing
anyone
everyone
no one

1. A: Did _____ go on vacation with you last month?
B: Yes, my family went to the countryside with me.
2. A: Did your family go to the beach with you last weekend?
B: No, _____ from my family went, but my friend went with me.
3. A: I didn't bring back anything from Malaysia.
B: _____ at all? Why not?
4. A: Did you buy _____ in the shopping center?
B: No, I didn't. _____ was very expensive.
5. A: How was the volleyball game yesterday?
B: Great! _____ had a fun time!

2 Complete the passage with the correct forms of the verbs in brackets.

Last August, our class _____ (do) something very special on our school trip. We _____ (go) to Mount Tai. We _____ (start) our trip at 12:00 at night. Everyone in our class _____ (take) a bag with some food and water. After three hours, someone looked at the map and _____ (find) out we _____ (be, not) anywhere near the top. My legs _____ (be) so tired that I wanted to stop. My classmates _____ (tell) me to keep going, so I _____ (go) on. At 5:00 a.m., we got to the top! Everyone _____ (jump) up and down in excitement. Twenty minutes later, the sun _____ (start) to come up. It was so beautiful that we _____ (forget) about the last five hours!

8

UNIT 2

Section A

Language Goal: Talk about how often you do things

How often do you

a b c

What do you do on weekends?
I often go to the movies.

1a Look at the picture. Make a list of the weekend activities.

1. help with housework
2. _____
3. _____
4. _____
5. _____

1b Listen and write the activities next to the correct frequency.

always (100%) exercise
usually
often
sometimes
hardly ever
never (0%)

1c Practice the conversation in the picture above. Then make your own conversations about what you do on weekends.

A: What
B: I usu
A: Do y
B: No, I

Contents

2a **Listen to these two conversations and circle the clubs you hear.**

a. English club
b. art club
c. music club
d. chess club
e. swimming club

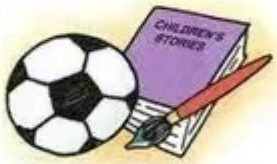
2b **Listen again. Complete the sentences.**

1. Lisa wants to join the _____ club, but she can't play _____.
2. Bob wants to join the _____ club. He likes to speak _____.
3. Mary likes music. She can _____ and _____ Bob likes music, too. They want to join the _____ club.

2c **Look at 2b and talk about what the people can do and the clubs they want to join.**

2d **Role-play the conversation.**

Jane: Hi, Bob. What club do you want to join?
Bob: I want to join a sports club.
Jane: Great! What sports can you play?
Bob: Soccer.
Jane: So you can join the soccer club
Bob: What about you? You're very good at telling stories. You can join the story telling club
Jane: Sounds good. But I like to draw, too.
Bob: Then join two clubs, the story telling club and the art club!
Jane: OK, let's join now!



一、单项选择（共15小题；每小题1分，满分15分）

( ) 1. Simon comes from _____ European country, he likes playing _____ piano and he often plays _____ football after school.
A. a; the; a B. an; the; a C. a; the; / D. a; the; the

( ) 2. —Can I have some bread to eat? —Sorry, dear. there is _____ left in the fridge.
A .nothing B. none C. something D. anything

( ) 3. Simon's grandfather lives _____ in a "_____ house. But he never feels _____.
A. alone; alone; lonely B. lonely; lonely; alone
C. alone; lonely; lonely D. alone; lonely; alone

( ) 4. Finally Gulliver broke the rope and _____ from those tiny men.
A. put away B. ran away C. took away D. kept away

( ) 5. What _____ fantastic time we had _____ football yesterday.
A. a; playing B. an; play C. / ; play D. a; to play

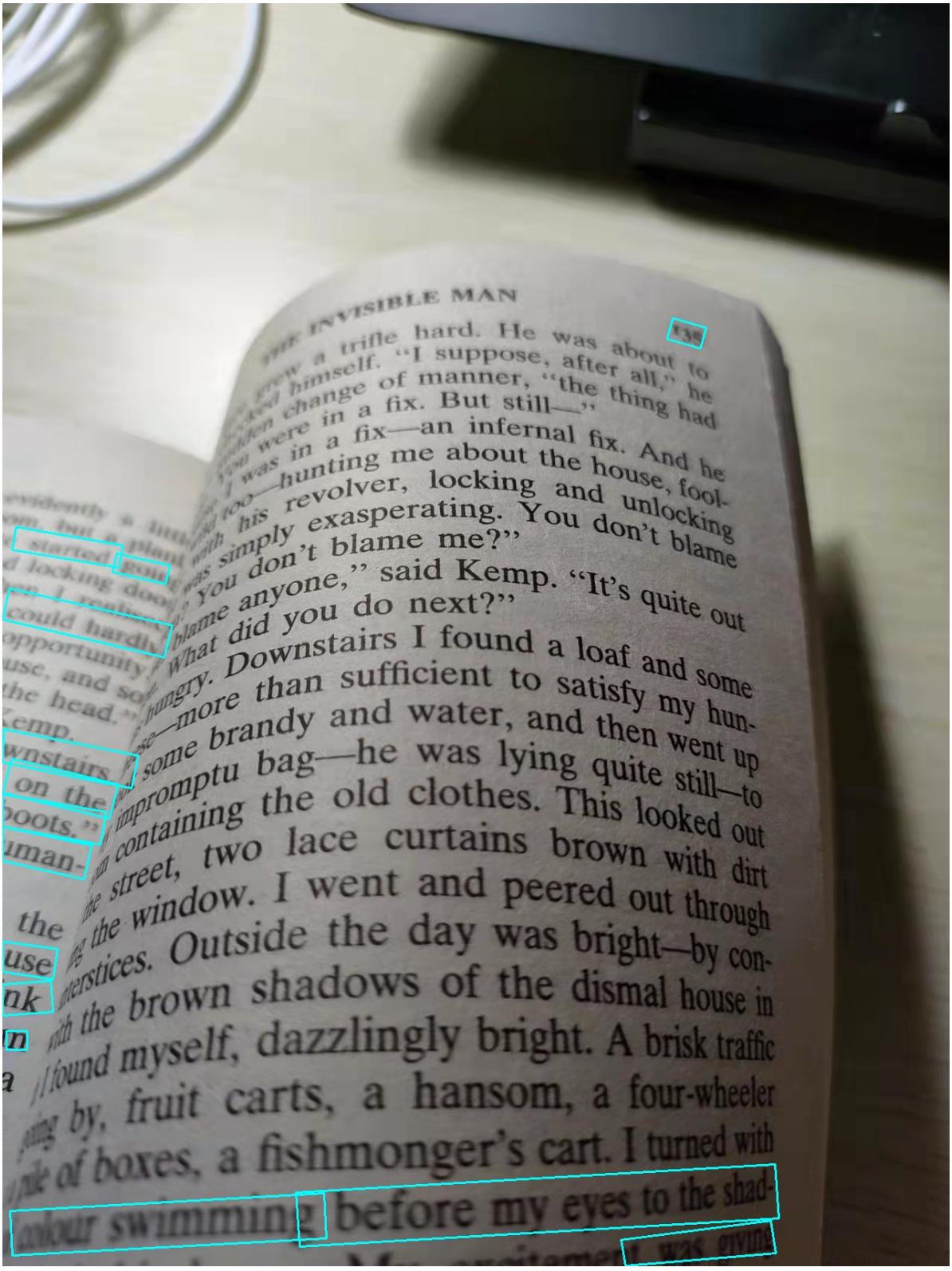
( ) 6. _____ online courses may not be perfect, they are still the only choice at present.
A. Because B. Although C. Unless D. While

( ) 7. The Tokyo Olympic Games will start _____ July 23, 2021 and run unti Aug 8.
A. in B. on C. at D. for

( ) 8. Zhang Wenhong _____ as a doctor in Huashan Hospital since he graduated from college.
A .works B. worked C. has worked D. is working

( ) 9. _____. If you keep practicing speaking English every day, you will be better at it.
A. no pain, no gain B. practice makes perfect
C. every dog has its day D. actions speak louder than words

( ) 10. —I didn't go to the cinema yesterday .what about you?
—_____ ,I was preparing for the exams all the time yesterday.
A. So did I B. Meneither C. So I did D. Me too



在第一和第三张，第五张图中显然都呈现了更好的效果，检测出了更多的文本框；并且在每个文本框中，对单词的框定更加准确（一词一框），虽然在图4中存在部分的"错框"现象，但总体来看更加全面和准确。

Recognition Model

config: en_PP-OCRv3_rec.yml;

测试指令:

```
python3 tools/infer_rec.py -c configs/rec/PP-OCRv3/en_PP-OCRv3_rec.yml -o
Global.pretrained_model=./trained_models/en_PP-OCRv3_rec_train/best_accuracy
Global.load_static_weights=false
Global.infer_img=./train_data/en_hand_writing/4.jpg
#Global.pretrained_model是模型的路径
#Global.infer_img是测试图片的相对路径
```

联合使用及不完善之处

方法一:

修改`paddleocr.py`文件:

首先将`MODEL_URLS`中`det`的`en`默认模型修改为自己选择的模型:

```
MODEL_URLS = {
    'OCR': {
        'PP-OCRv3': {
            'det': {
                'ch': {
                    'url':
                    'https://paddleocr.bj.bcebos.com/PP-OCRv3/chinese/ch_PP-OCRv3_det_infer.tar',
                },

                'en': {
                    'url':
                    'https://paddleocr.bj.bcebos.com/PP-OCRv3/english/en_PP-OCRv3_det_slim_infer.tar',
                },
```

再将`MODEL_URLS`中`rec`的`en`默认模型修改为自己选择的模型:

```
'en': {
    'url':

    'https://paddleocr.bj.bcebos.com/PP-OCRv3/english/en_PP-OCRv3_rec_infer.tar',
    'dict_path': './ppocr/utils/en_dict.txt'
},
```

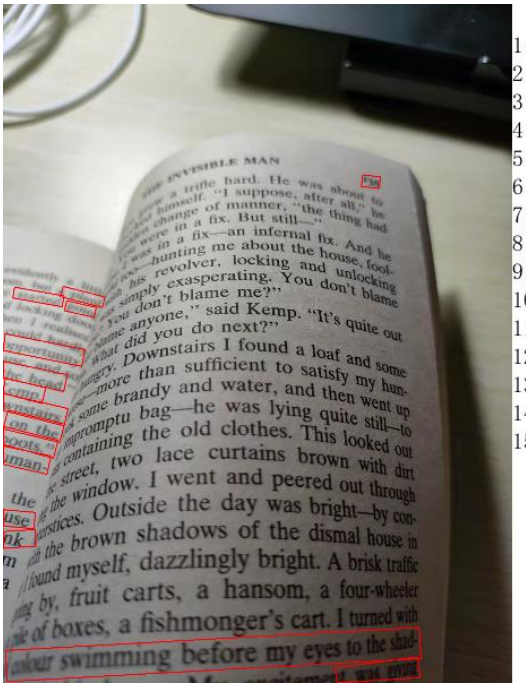
之后通过正常使用paddleocr语句就可以调用训练的模型进行使用了。

方法二:

联合模型处理:

```
#举例
python3 tools/infer/predict_system.py --
image_dir="./train_data/en_hand_writing/1.jpg" --
det_model_dir="./trained_models/en_PP-OCRv3_det_slim_infer/" --
rec_model_dir="./trained_models/en_PP-OCRv3_rec_infer" --use_angle_cls=false
```

但是由于目前的paddleocr框架的局限性，对于使用SAST算法进行检测的文本无法进行进一步的识别。换而言之，目前的paddleocr是无法对弯曲的文本进行很好的检测的。使用默认baseline的对下图进行识别，结果如下。可见结果非常糟糕。




```
1: 135    0.785
2: started    0.919
3: iptant    0.617
4: Roin    0.630
5: opportunity    0.974
6: the head    0.943
7: Kemp.    0.890
8: wnstairs.    0.978
9: on the    0.989
10: boots."    0.887
11: uman-    0.873
12: use    0.996
13: nk    0.977
14: ankour swimming before my eyes to the shad.    0.892
15: was giving    0.813
```

LDOUBLEV commented 4 days ago — Collaborator

目前识别不支持弯曲文本

ZimaBlue307 commented 20 hours ago

也就是说，目前的paddleocr，如果使用SAST进行了detection之后，没有任何办法进行进一步recognition是么？

LDOUBLEV commented 8 hours ago — Collaborator

是的

总结

任务小结

本次实训完成了如下工作：

1. 针对OCR和paddleocr中的常见技术，以及paddleocr的使用进行了详细的调研和记录；
2. 针对英文印刷体识别，使用paddleocr进行测试，并对其默认模型中存在的问题进行了细致分析；
3. 针对其中的问题与不足，找寻了相应的技术手段和解决方案；
4. 根据技术手段和解决方案，选取相应的模型，结合计算平台和本机进行训练与测试。
5. 比较自己选取与训练的模型和paddleocr默认模型的检测和识别效果，并进行适当分析；
6. 发现目前paddleOCR大框架下的不足之处：存在模型可以对弯取文本进行检测，但是无法对弯取文本在检测基础上进行进一步识别。

收获小结

1. 在两个月不到的不长不短的时间里，完成了对一个较为陌生的领域的了解，分析，深挖，实践，分析比较的过程，是一个难能可贵的完整的，以动手为核心的实战经历。
2. 中长期的阶段性任务，时间的安排和自我督促很重要。需要制定阶段性目标，有计划扎实的不紧不慢的完成，不可漫无目的的前进；目标方向和自律性十分重要。
3. 要充分利用好各处的资源。开源项目中的开源社区，大家的友好交流，可以帮助初学者解决许多问题；计算平台中的各类计算资源也可以对模型的部署和训练带来很大的帮助；组里的队友和老师的指导也是可以让自己少走很多的弯路，朝着正确的方向摸索前进。
4. 永远不要想着一口吃一个大胖子。例如，如果不对paddleocr的基础架构有十分清晰的了解，自己在遇到问题和分析问题的时候总会盲目和失焦。而当清楚其中的多种技术原理和短板时，才能更敏锐的把实际检测的问题套回到技术理论中。因此虽然我们要注重实践和代码动手能力的培养，但是基础理论的学习也十分重要。