



第十四届全国大学生软件创新大赛

文档编号: SWC2021-T20210533-花生队



落笔云烟

Magpie's Pen

技术研究报告

Version: V2.2.0



花生队

2021.04.10

All Rights Reserved

# 目录

<b>1</b>	<b>问题聚焦</b>	<b>1</b>
1.1	问题描述	1
1.2	问题抽象	1
1.3	问题定位	2
1.4	问题评估	4
1.5	问题分解	8
<b>2</b>	<b>相关工作</b>	<b>12</b>
<b>3</b>	<b>技术方案</b>	<b>15</b>
3.1	技术方向	15
3.2	技术选择	16
3.3	结果期望	33
<b>4</b>	<b>技术实践</b>	<b>34</b>
4.1	使用的开发框架及依赖的库	34
4.2	技术实践过程	34
<b>5</b>	<b>结果验证</b>	<b>39</b>

## 文档修订历史

序号	修订原因	版本号	作者	修订日期	备注
1	创建技术研究报告	V1.0.0	A	2020.11.15	
2	创建问题聚焦模块	V1.1.0	C	2020.11.15	
3	更新问题聚焦模块	V1.1.1	C	2020.11.18	
4	创建相关工作模块	V1.2.0	B	2020.11.19	
5	创建技术方案模块	V1.3.0	C	2020.12.10	
6	更新技术方案模块	V1.3.1	C	2020.12.20	
7	创建技术实践模块	V2.0.0	C	2021.2.3	
8	更新技术方案模块	V2.0.1	C	2021.2.15	更新技术方向
9	更新技术方案模块	V2.0.2	C	2021.2.18	更新技术选型
10	更新技术实践模块	V2.1.0	C	2021.3.10	
11	更新技术实践模块	V2.1.1	C	2021.3.17	更新技术实现过程
12	更新技术实践模块	V2.1.2	C	2021.3.30	更新技术实现过程
13	文档核对	V2.2.0	B, C	2021.4.4	

# 1 问题聚焦

## 1.1 问题描述

“落笔云烟”是一个使用深度学习算法辅助用户练习硬笔书法的软件。项目按功能可以分成三个模块：书写图像识别、单字分析与评价、用户个性化服务。以下将从这三个模块对所涉及到的流程与问题简要描述：

1. 书写图像检测：用户上传书写练习的图像（包括但不限于字帖、作业等），系统对上传的图像进行处理与分析，并将处理后的结果保存于系统内，方便系统其他功能直接使用。同时，系统会对用户上传的练习图像进行整体分析，并将评价结果反馈给用户。
2. 汉字识别与纠正：该模块是本应用的核心功能，应用可以识别用户书写的汉字的笔画、结构，并与目标字体对比，为用户生成多方面的书写评价指导，帮助用户更快地发现自己书写的缺陷所在，向正确的方向练习。
3. 用户个性化服务：该功能模块主要为用户提供个性化的服务，包括以下几个方面：书写推荐——根据用户薄弱字体分析的结果为用户推荐同样结构的字体进行练习；学习追踪——依据历史数据，项目可为用户提供具体字体风格变化情况以及评分变化曲线，记录用户的成长；娱乐模块——为解决现有练字所不得不面对的费事及枯燥，通过设定字体闯关等方式，为用户设定成长系统，变被动练习为主动训练。

## 1.2 问题抽象

根据 [1.1](#) 所列出的三大模块，本节将详细阐述各个模块包含的具体问题所对应的技术问题。

### ❖ 书写图像检测模块：

书写图像检测需要对用户上传的图像进行分析，识别出图像中的字，并向用户反馈总体的书写情况，因此涉及到以下技术问题：

- 图像的处理，去除图像中的噪声，并做一定程度的图像增强。
- 图像中单字的检测，检测出输入图像中的汉字，检测出单个汉字的边框并识别出对应的汉字。
- 整体书写情况评价，根据识别出的汉字的边框信息，汉字在图片中的位置信息等，给出输入的书写图片的整体书写评价。
- 书写报告生成，根据之前生成的整体书写评价，使用自然语言处理技术生成用户易于理解的书写报告。

### ❖ 书写识别与纠正模块：

本模块是本应用的核心功能，主要针对单个汉字进行处理，在获得“书写图像检测”模块检测出的单个汉字的图片与汉字信息后，本模块将识别出单个汉字的笔画与结构信息，并与用户选择的模版字对比，计算出用户书写的汉

字有缺陷的部分，同时对这个字的书写情况进行评分，并将结果反馈给用户提供指导。该功能需要对输入的单个字的图像进行分析，给出这个字的评分，并给用户反馈该字书写的缺陷，因此涉及到以下技术问题：

- 图像处理，针对上一模块给出的汉字的位置信息，对输入图片进行裁剪，并去除图像中的噪声，提高接下来的汉字识别模型的识别准确率。
- 汉字骨架的识别，对输入的汉字，识别出汉字中包含的所有笔画的位置信息以及对应的笔画信息。
- 汉字图像的对齐，根据检测出的笔画信息与输入手写字本身的图像信息对输入图片进行适当的旋转、缩放，将输入汉字图像标准化，方便后续的纠正。
- 汉字纠正，本功能由三个方面组成，详细内容请参考“技术方案”部分。  
一、根据手写汉字与模版汉字的骨架信息，得到两者的最适配位置，将该位置反馈给用户便于查看书写汉字与模版字的区别。二、根据检测出的骨架信息与模版字的骨架信息对比，通过不同的指标计算出用户书写的汉字存在缺陷的地方，将该信息反馈给用户。三、根据提前录入的专业汉字书写指导信息，匹配用户书写汉字对用的模块，根据此指导信息与识别出的书写骨架信息，提供专业的书写指导。
- 书写评价，根据检测出的骨架信息，结合“汉字纠正”模块的部分指标信息，对用户的书写进行评分，反馈给用户并记录。

#### ❖ 用户个性化服务：

该功能对用户的书写历史进行记录并建模分析，绘制出用户的总体评分曲线，并根据用户的书写历史推荐相似结构或易错的单字供用户练习；同时，根据用户的书写特点推荐最适合用户练习的字体、根据用户的字体特征生成用户的专属字体文件。因此涉及到以下技术问题：

- 根据书写历史推荐练习内容及相似字体，需要根据用户之间以及单字的相似度来动态计算推荐群及信任子群，同时需要考虑解决冷启动问题。
- 根据书写特征生成用户专属字体文件，以用户的历史书写的字体风格数据为基础，迁移生成个人字体库。

### 1.3 问题定位

对 1.2 节中所列出的技术问题总结如下：

技术问题	模块	业务领域	技术领域	备注
图像降噪、图像增强	书写图像检测	后台算法	数字图像处理	后台完成，不将结果反馈给用户
手写字检测	书写图像检测	后台算法	目标识别，深度学习	可以使用 AIUnit 在客户端完成，节约服务器计算资源
整体书写评价	书写图像检测	后台算法	数字图像处理	可以得到不同

			理, 机器学习	维度的原始评价信息, 需要经过处理后可以生成用户可以理解的信息
书写报告生成	书写图像检测	前端算法, 后台算法	可视化技术, 自然语言处理	根据整体书写评价得到的信息生成用户可以理解的书写报告
汉字骨架识别	书写识别与纠正	后台算法	计算机视觉, 深度学习, 图算法, 匹配算法	根据输入的汉字图片得到该汉字的骨架信息, 提供给后续算法进行分析
汉字图像对齐	书写识别与纠正	后台算法	深度学习, 数字图像处理	将输入的汉字图像进行适当的旋转与缩放, 使输入的汉字标准化, 方便后续的分析
汉字纠正	书写识别与纠正	后台算法	算法, 专家系统	首先找出模版字与书写汉字最为匹配的位置和角度。之后根据检测出的汉字骨架, 找出书写汉字的问题。最后根据录入的专家指导信息, 匹配最合适的指导信息。
汉字评分	书写识别与纠正	后台算法	算法	首先根据检测出的汉字骨架, 判断是否缺少笔画。之后根据模版汉字与用户书写汉字的各个笔画之间的角

				度、长度的差异信息进行评分。最后根据数据库中建模完成的汉字的结构信息, 结合识别出的骨架信息, 评价当前书写的汉字的结构分。
--	--	--	--	--

## 1.4 问题评估

技术问题	技术性	普适性	研究热度	问题热度
图像降噪	技术选取范围较广, 技术难度不大。但考虑到书写图像环境因素的不确定因素较大, 因此需要选择鲁棒性高的算法。	普适性高, 针对书写图像问题优化后的具有高鲁棒性的图像降噪算法可以适应大部分的书写环境 (指田字格、练习纸等含有图案或打印字体的书写环境)	OCR 问题的图像降噪技术在 70 年代便已被提出, 在深度学习学习方法出现后效果又得到了显著提升。目前相关算法已经可以实现商业化大规模应用, 近几年来研究热度有所下降。	广泛应用于各类视觉, 图像处理任务中, 热度较高。
手写字检测	手写字检测可以看作目标检测问题。目前主流的目标检测算法皆已成熟, 技术难度较低。	主流的目标检测算法即可完成手写字识别任务, 对大部分手写图像均可准确识别, 普适性高。	Faster-RCNN 与 YOLO 模型的提出标志着目标检测问题的成熟化解决方案。现阶段大部分的目标检测任务均可使用这两类模型很好地完成, 因此近几年的研究热度有所下降。	手写字的目标检测任务被广泛应用于 OCR 等任务当中, 热度较高。
整体书写评价	整体书写评价涉及到文字提取与整齐度计算两个部分, 文	普适性高, 目标检测模型与整齐度计算算	手写图像检测问题见“手写字检测”部	整体书写评价在当下主要还是靠人

	字提取可以使用目标检测方式完成, 技术性一般, 整齐度计算主要使用数字图像处理的方法完成, 技术原理较为简单, 具体实现时可以针对问题作出相应优化。	法在单独问题下均可取得理想效果, 两者结合可以处理大部分书写图像的评价问题。	分。整齐度计算算法现阶段相关研究较少, 但可以利用自研究的整齐度模型进行计算。	工的方式来评价, 对于该问题的数字化解决方案的需求很大, 问题热度目前一般。
书写报告生成	根据整体书写评价功能给出的各个维度的评价, 使用 NLP 与可视化技术给出用户方便理解的报告。该问题只需要使用到简单的 NLP 方法, 技术性较低。可视化展示方面可以从多个维度进行探究, 技术性中等。	普适性高, 由于“整体书写评价”部分已经给出具体的评价数据, 只需要将数据合理地展示给用户即可, 结合 NLP 与可视化技术, 大部分的情况下均可取得良好效果。	对于给定数据生成自然语言的问题属于 NLG 领域, NLG 领域的技术仍在不断发展, 但是本项目中仅需要使用 Slot Filling 相关的技术即可, 此方面的研究已经相对成熟, 近些年来研究热度不高。	报告生成是当下众多 APP 都广泛应用的一项技术, 包括但不限于学习报告、诊断报告等。即将数字化的信息以人方便接受的方式传递给用户, 当下的应用热度很高。
汉字骨架识别	汉字骨架识别任务需要从输入的图片中识别出 26 种汉字的基本笔画, 并给出各个笔画中关键点的位置。不同的汉字含有的笔画并不相同, 一个汉字中常常含有多个相同的笔画, 笔画的关键点常常会有重叠的情况出现, 以上问题都大大加大了预测出汉字笔画的难度。我们通过使用深度学习模型预测关键点与书写笔画的方向场, 之后通过二分匹配算法与最优化算法	普适性高, 本方法直接从笔画预测入手, 深度学习模型预测的基本单元是书写的笔画, 因此输入的汉字可以是任意的汉字, 甚至是没有见过的汉字。对于数据库中有记录的汉字, 在预测出基础的骨架后, 我们通过图算法与最大匹配算法找出该汉字最优的笔画的	汉字骨架识别是我们提出的一个全新的概念, 目前学术界还未有相关的研究。不过人体的关键点检测是很热门的研究领域, 相比较于人体的关键点检测, 汉字的骨架检测面临如下问题: 一、人体的关键点数量是一定的, 汉字中笔画是不一定的, 因此笔画	目前并没有任何 APP 提供了汉字骨架识别的功能。与之相似的是部分 APP 提供了对汉字笔画的语义分割功能, 不过这类 APP 通常需要额外的硬件辅助追踪书写的轨迹与像素信息, 或是只能在毛笔书写的个别汉字上进行, 局限



	<p>寻找对应汉字笔画的最优集合,从而实现高效、统一的汉字骨架预测方法。本方法适用于任何汉字,并且有较高的准确率与时效性,技术难度高。</p>	<p>集合,从而实现更高精度的笔画预测,在我们标注的数据集上的实验表明,在各种手写和打印的字体上,本方法均可取得很好的效果。</p>	<p>的关键点也是不一定的。二、人体的关键点数量较少,常用的是 19 个关键点,而想要预测所有汉字的骨架,至少需要 87 个关键点。三、人体骨架的训练图片中包含所有待预测的关键点,而汉字的训练图片中只含有部分的关键点,因此训练的难度较大,对于数据集的组成要求较高。四、人体骨架检测前通常需要提供人身的检测框来区分不同的人,并且用提供的语义分割 mask 来过滤噪声,我们所标注的汉字关键点数据集中并不含这两个信息,因此会进一步加大预测的难度。我们的工作将骨架预测的概念引入到了手写字识别领域,不仅可以作为纠正书写的基础,使用识别出的骨架信息也可以很好地</p>	<p>性较大。我们提出的汉字骨架识别方法适用于任何书写的汉字,并且不需要额外的硬件支撑。</p>
--	---	--	--	--

			去除背景噪声，提升 OCR 任务的准确性。	
汉字图像对齐	由于输入的汉字的大小、角度受制于检测框的大小和图片拍摄角度的影响，输入的汉字图片并不能与模版字直接比较。需要首先将输入的汉字缩放到一定的比例，再旋转至正确的方向后，才能与标准字进行对比，计算相应的指标。我们通过识别出的骨架关键点信息对输入汉字进行缩放，再通过深度学习模型预测旋转的角度，从而获得大小、角度统一的汉字与骨架信息。技术难度中等。	普适性高，本方法不依赖于汉字的类别，通过检测出的汉字骨架进行缩放，因此消除了原始输入图片的噪声的影响，通过深度学习模型预测旋转的角度，适用于任何输入的汉字。因此本方法普适性高。	汉字图像对齐任务并没有相关的研究。	目前市场上的 APP 大多忽视了汉字对齐问题的重要性，输入图片中书写的汉字若是不对齐，在后续的纠正和评分功能中会出现较大的误差。仅仅对对齐后的汉字进行处理也可以降低纠正与评分部分算法的复杂度，提升后续任务的效果。
汉字纠正	汉字纠正功能包含三个方面，首先根据输入汉字图像和模版汉字图像检测出的骨架信息得到这两者最适合的对应位置和角度，并返回给前端显示。之后根据输入和目标汉字的骨架信息计算这两者各个指标的差异，将结果反馈给用户。最后根据专家系统中录入的指导信息，匹配目前书写的问题，将指导建议反馈给用户。该功能涉及的面较广，且对先验知识匹配算法的要求较高，所以技术	本功能针对用户手写的汉字从三个方面给出指导方案。因此可以适应大部分的情况，普适性较高。	本功能涉及三个技术方面分别是最优位置的寻找，书写标准的计算和专家系统。前两者是我们项目中特有的功能，专家系统适用于完成那些没有公认的理论和方法、数据不精确或信息不完整、人类专家短缺或专门知识十分昂贵的任务上。构建适用于汉字纠正的专家系统	目前市场上的 APP 所拥有的纠正功能大多比较死板，我们的功能通过三个方面为用户提供综合全面的指导建议，普适性和实用性更高。本问题目前的热度较高。

	难度较高。		涉及到对汉字结构的分解和复用, 这方面也是我们的项目所特有的, 具体实现请参考技术方案部分。	
单字评分	本项目从三个维度为用户书写的汉字评分。首先根据检测出的汉字骨架, 判断是否缺少笔画。之后根据模版汉字与用户书写汉字的各个笔画之间的角度、长度的差异信息进行评分。最后根据数据库中建模完成的汉字的结构信息, 结合识别出的骨架信息, 评价当前书写的汉字的结构分。最后的评分是这三者的综合。因为不同汉字涉及的标准不同, 并且计算与模版汉字的差异时需要注意的问题较多, 所以该功能技术难度较高。	本功能从三个维度综合对用户手写的汉字进行评分, 因此普适性高。	本功能涉及的技术均为本项目特有, 暂未有相关技术研究。	在经过市场调研后我们发现有个别针对书法练习的 APP 也有评分功能, 不过评分方式过于机械, 且受限于部分汉字, 实用性不高。本功能从三个维度综合考量用户的书写情况, 可以很好地克服这一问题。本问题目前热度较高。

## 1.5 问题分解

技术问题	子问题	描述	难度	依赖关系
图像降噪	①图像降噪	输入一张图片, 去除图片中的噪声	中等	
手写字检测	②手写字检测	输入降噪后的手写字的图片, 检测出图片中所有手写的字, 返	简单	依赖①

		回 Bounding Box 和对应标签		
整体书写评价	③单字评价	在多字图片中提取单字重心及 Bounding Box	中等	依赖②
	④整齐度计算	计算整体重心偏移量方差以及距离方差, 并进行合理评估打分	简单	依赖③
书写报告生成	⑤参数预处理	根据整体书写评价得到的各项参数做相关预处理	简单	依赖③④
	⑥自然语言生成	根据得到的各项评价参数生成用户可以理解的书写报告	中等	依赖⑤
汉字骨架识别	⑦关键点热力图/笔画 PAF 预测	根据输入的图片, 使用深度学习模型预测关键点的热力图, 笔画方向 PAF 图	高	依赖②
	⑧OKS 算法	通过 NMS 算法的变形 OKS 算法, 过滤掉距离较近的同类型的关键点	中等	依赖⑦
	⑨PAF 路径计算	通过计算不同连接路径的向量与 PAF 的积分, 得到这一连接的分数	高	依赖⑦⑧
	⑩最优子集搜索	根据汉字包含的笔画信	高	依赖⑦⑧

		息, 构建多个可能的集合, 计算不同集合的 PAF 分数, 得到最优的笔画集合		
汉字图像对齐	⑪图像缩放	通过检测出的关键点对图像进行缩放	低	依赖⑦⑧⑨⑩
	⑫图像旋转	通过深度学习模型预测端正一个汉字所需的旋转角度	中等	依赖②
汉字纠正	⑬模版图像位置匹配	根据检测出的汉字骨架, 找出模版汉字与输入汉字的最佳对应位置	低	依赖⑦⑧⑨⑩
	⑭差异计算	根据检测出的汉字骨架, 判断是否缺少笔画、笔画间的长度、倾斜度的差异, 并将结果反馈给用户	低	依赖⑦⑧⑨⑩
	⑮专家系统匹配	根据检测出的汉字骨架与识别出的汉字信息, 匹配专家系统中已经录入好的指导信息, 判断当前书写的缺陷所在, 返回对应的指导建议	高	依赖②⑦⑧⑨⑩
单字评分	⑦单字评价模型	根据识别出的汉字骨架与汉字纠正	中等	依赖②⑦⑧⑨⑩⑭

		模块的差异 计算结果，给 出当前汉字的 评分		
--	--	---------------------------------	--	--

## 2 相关工作

### 图像去噪

图像去噪声是对图像做预处理,使得处理后的图像更适合用于文字识别与目标检测等任务。常用的图像去噪方法包括二值化并设置阈值 (binarization)、模糊逻辑 (fuzzy logic)、图像直方图 (histogram)与使用仿真算法和遗传算法为基础的方法。Farahmand 与 Ganchimeg 等人 [1][2] 总结了使用传统数字图像处理与启发式算法进行图像降噪的方法。Liu 等人 [3] 使用噪声水平函数 (Noise Level Function, NLF) 与高斯条件随机场 (Gaussian Conditional Random Field, GCRF) 提出了一种可以自动估计图像中的色彩噪声并产生去噪后清晰图片的方法。Sobia 等人 [4] 提出了去除乌尔语文档图片中的孔洞与噪声的方法。这些方法大多基于传统的数字图像处理技术,并没有使用深度学习领域的方法。

CycleGAN [5] 在图像-图像翻译领域取得了很好的效果,Sharma 等人 [6] 将 CycleGAN 方法应用到图像去噪领域,在多个数据集上均取得了良好的效果。本项目中将主要采用基于 CycleGAN 的图像降噪方法。

### 手写字检测

手写字检测本质上是目标检测问题,目标检测任务的模型可以分为基于 R-CNN 的模型和 YOLO 系列模型。Ren 等人 [7] 提出了 Faster R-CNN 模型,该模型解决了传统 R-CNN 模型无法端到端训练问题,并且使用 RPN 网络大大减少了提出区域的时间,在效率和准确度上均取得了很好的成果,这也意味着目标检测领域算法的成熟。Redmon 等人 [8] 提出了 YOLO 模型,该模型是目标检测算法的另一大家族,YOLO 算法可以在保持高准确率的同时完成实时的目标检测,很适用于实际应用场景。本项目中主要采用 Faster R-CNN 作为手写字检测的模型。

### 汉字骨架识别

汉字骨架识别主要参考了人体姿态检测的相关技术,人体姿态检测的任务是从图片或者视频中检测出人体的各个部分,这一任务可以分为单人的姿态检测和多人的姿态检测,其中单人的姿态只需要在给定的检测框内检测出人体相应的关键点,该任务较为简单,多人的姿态检测需要检测出图片中多个人的姿态。深度学习在视觉领域取得突破后,CNN 为主的方法也开始在姿态检测领域广泛采用 [9-13],这类方法在处理多人的姿态识别问题时,通常是首先检测出人体的边框,再在边框内进行单人的姿态检测。但这一类方法的受到检测框的制约,在检测框出现问题时,效果往往会较差,并且检测的时间与图像中人的数量成正比,不能做到实时检测的效果。Cao 等人 [14] 为了解决这一问题,提出了 OpenPose 框架,这也是我们的方法的基础。该方法可以做到多人实时检测,效率不受图中所出现人物数量的影响。

## 参考文献

- [1] Farahmand, Atena, Hossein Sarrafzadeh, and Jamshid Shanbehzadeh. "Document image noises and removal methods." (2013).
- [2] Ganchimeg, Ganbold. "History document image background noise and removal methods." *International Journal of Knowledge Content Development & Technology* 5.2 (2015): 11-24.
- [3] Liu, Ce, et al. "Automatic estimation and removal of noise from a single image." *IEEE transactions on pattern analysis and machine intelligence* 30.2 (2007): 299-314.
- [4] Javed, Sobia Tariq, et al. "Background and punch-hole noise removal from handwritten urdu text." *2017 International Multi-topic Conference (INMIC)*. IEEE, 2017.
- [5] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *Proceedings of the IEEE international conference on computer vision*. 2017.
- [6] Sharma, Monika, Abhishek Verma, and Lovekesh Vig. "Learning to clean: A GAN perspective." *Asian Conference on Computer Vision*. Springer, Cham, 2018.
- [7] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016): 1137-1149.
- [8] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [9] L. Sigal and M. J. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in CVPR, 2006.
- [10] X. Lan and D. P. Huttenlocher, "Beyond trees: Common-factor models for 2d human pose recovery," in ICCV, 2005.
- [11] L. Karlinsky and S. Ullman, "Using linking features in learning non-parametric part models," in ECCV, 2012.
- [12] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in CVPR, 2013.
- [13] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in ECCV, 2016.
- [14] Cao, Zhe, et al. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." *IEEE transactions on pattern analysis and machine intelligence* 43.1 (2019): 172-186.





### 3 技术方案

#### 3.1 技术方向

功能	技术问题	技术方向
书写图像检测	图像降噪	数字图像处理, 计算机视觉, 深度学习
	手写字检测	计算机视觉, 深度学习
	整体书写评价	数字图像处理
	书写报告生成	自然语言处理
书写识别与纠正	汉字骨架识别	深度学习, 计算机视觉, 二分匹配, 微积分, 线性代数, 优化算法
	汉字图像对齐	计算机视觉, 深度学习
	书写汉字纠正	传统算法, 专家系统
	书写汉字评分	传统算法
用户个性化服务	练习推荐	推荐系统, 信息检索
	用户字库生成	计算机视觉, 深度学习

## 3.2 技术选择

### 3.2.1 汉字骨架识别

汉字骨架识别的目标是识别出汉字的各个笔画的关键点与连接信息。本项目后续的汉字纠正、汉字评分等功能均依赖于识别出的汉字骨架信息，因此本项目对于汉字骨架识别的准确性和效率均有较高的要求。

汉字骨架识别的概念是我们受人体骨架识别的启发而提出的，图 1 分别是汉字骨架与人体骨架的效果图。汉字骨架检测与人体骨架检测一样，都是要检测出关键连接（汉字：笔顺，人体：肢体）上的关键点，然后连接对应连接上的关键点。但与人体骨架识别不同的是，汉字骨架识别存在以下困难：

- (1) 人体骨架的关键点数量是一定的，而对于汉字骨架来说，当输入的汉字不一样时，包含的关键点的数量和类型都是不一样的。
- (2) 人体骨架识别只需要 19 个不同类型的 keypoints，并且输出中一定包含所有的 keypoints。对于汉字骨架识别而言，想要识别出所有的笔画，至少需要 87 个不同类型的 keypoints，并且在输出中大部分的 keypoints 并不存在。
- (3) 汉字不同的笔画的相似度很高，并且由于不同人的书写习惯的问题，同一笔画的写法也会有很大的差异。例如在图 1 的“云”字中，上方的“横”与“点”这两个笔画的相似度很高，而该字中的两个“横”的差异却较大。

基于这些问题，我们提出了如下的汉字骨架识别框架，其流程图如图 2 所示。我们从 CMU 提出的 OpenPose 模型中得到启发，使用 PAFs (Part Affinity Fields) 来编码汉字书写图像中的笔画方向信息，通过预测关键点位置的置信图来得到不同笔画的关键点的位置。得到 keypoints 的置信图和笔画的 PAFs 后，便可以通过贪心算法来计算不同连接的分数，之后通过识别出的汉字信息，构造可能的汉字笔画的集合，通过计算不同集合的分数得到可能性最大的连接集合，将集合中的 keypoints 与相连即可得到汉字的骨架结构。下面将具体阐述该流程涉及到的各个方法的细节。



Figure 1 汉字骨架与人体骨架

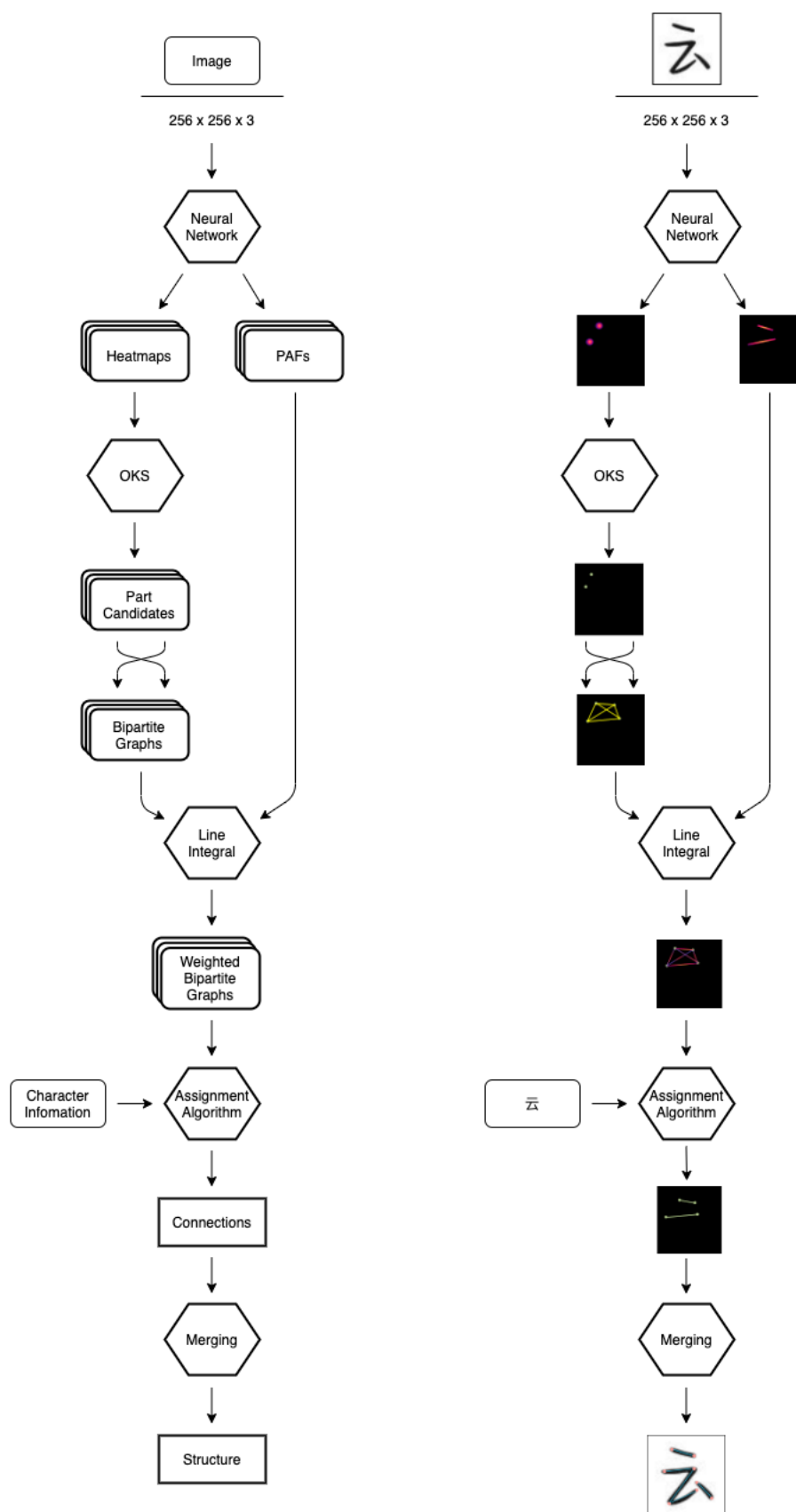


Figure 2 汉字骨架识别流程图

### (1) 模型架构

模型架构如图 3 所示, 输入一张图片, 模型首先使用骨干网络提取图片的特征, 这里我们选择的是 ResNet50 作为我们模型的骨干网络, 之后通过 6 个 PAF 阶段迭代地预测 PAFs, 在每一个模块结束时, 计算该阶段的输出与目标的 PAFs 图的 Loss, 最后的 PAF 模块的 Loss 是这 6 个 PAF 预测阶段的 Loss 的和。将最后一个阶段的 PAFs 图与模型的特征图拼接, 作为关键点置信图预测模块的输入, 关键点置信图预测同样经过 6 个阶段, 每个阶段的输出均与目标的置信图计算 Loss, 并将这六个阶段的 Loss 累加, 最后模型的 Loss 是 PAF 预测阶段的 Loss 与关键点置信图预测阶段的和。

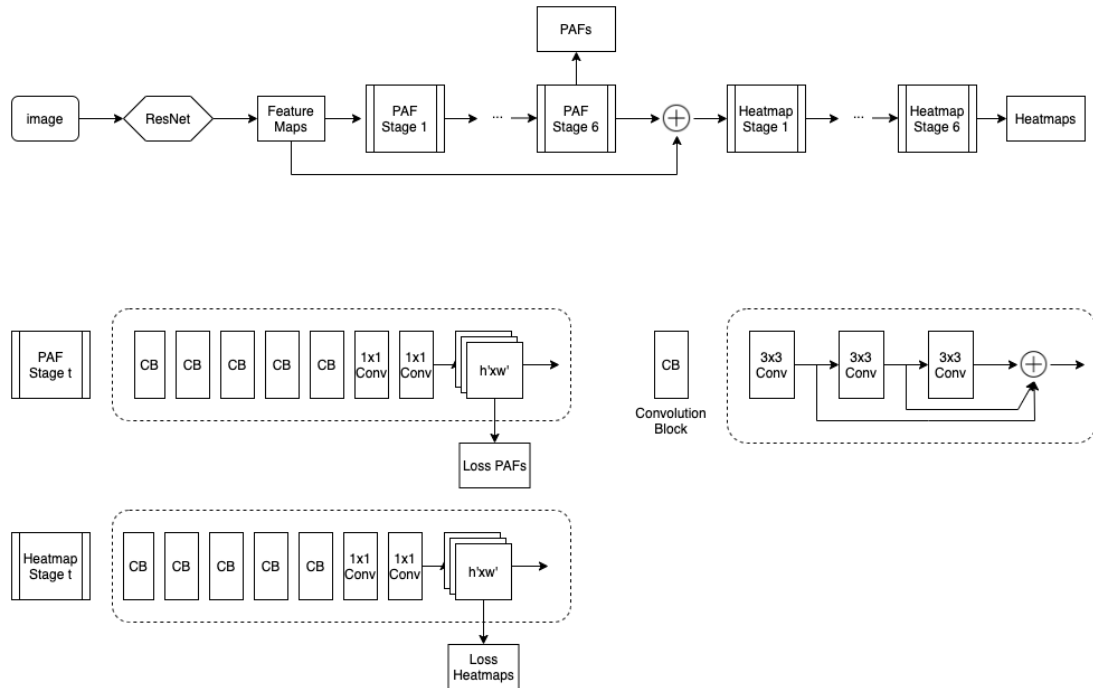


Figure 3 汉字骨架识别模型

PAF 预测模块与置信图预测模块的网络结构一样, 均由 6 个阶段的子模块  $\phi^t$  组成。每个子模块由 5 个卷积模块组成, 这 5 个卷积模块不含有激活函数, 每个卷积模块由三个 3x3 卷积层组成, 每个卷积层均设置  $\text{stride}=1$ ,  $\text{padding}=1$ , 因此不会改变输入的尺寸, 这三个卷积层之间使用了跳过连接的方式来缓解梯度消失的问题。每个子模块经过 5 个卷积模版 CB 后, 使用了两个 1x1 卷积来改变输入特征的通道数, 使得输出的特征的通道数与目标的 PAFs/Heatmaps 一致, 在我们的应用中, PAFs 的通道数为 122, Heatmaps 的通道数为 87。经过第  $t$  个 PAF 阶段的输出如下:

$$L^t = \phi^t(F, L^{t-1}), \forall 1 \leq t \leq N \quad 1$$

$L^t$  为第  $t$  个阶段的 PAF 预测, 通过 6 个阶段的迭代预测, 可以得到更加精确的 PAF 预测图。将最后一个阶段的 PAF 预测图  $L^N$  与图像的特征图  $F$  拼接得到关键点置信图预测阶段的输入  $L^{N'}$ , 之后同样经过  $N$  个同样的置信图预测阶段  $\rho^t$  得到最终的置信图, 与 PAF 预测阶段不同的是, 置信图预测的每个阶段都需要将 PAF 最后一个阶段的预测结果传入, 其输出如下:

$$S^t = \rho^t(F, L^N, S^{t-1}), 1 \leq t \leq N \quad 2$$

模型最终输出的 PAFs 取 PAF 模块最后一个阶段的输出, 最终输出的关键点置信图取置信图模块最后一个阶段的输出。汉字数据集中的目标 PAFs 为  $L^*$ ,  $L^* \in \mathbb{R}^{h \times w \times 122}$ , 目标的关键点置信图为  $S^*$ ,  $S^* \in \mathbb{R}^{h \times w \times 88}$ , 包含 87 个关键点和 61 个连接。对于每一个阶段的输出, 都计算其与目标图的 L2 Loss, 将所有阶段的 Loss 累加得到最终的 Loss, 其公式化描述如下:

$$f_L^{t_i} = \sum_{c=1}^C \sum_p ||L_c^{t_i}(p) - L_c^*(p)||_2^2 \quad 3$$

$$f_S^{t_k} = \sum_{j=1}^J \sum_p ||S_j^{t_k}(p) - S_j^*(p)||_2^2 \quad 4$$

最终模型的 Loss 是 PAFs 部分的 Loss 与置信图部分 Loss 的和, 其公式化描述如下:

$$f = \sum_{t=1}^N f_L^t + \sum_{t=1}^N f_S^t \quad 5$$

## (2) 关键点置信图

汉字的关键点置信图用于检测各个笔画的关键点, 依据笔画的复杂度, 关键点的数量也会变化。我们项目中所定义的笔画与关键点如图 4 所示, 当需求发生改变时, 可以通过增加表示笔画的关键点的方法来获得更加细粒度的笔画表示。由于汉字中通常只包含部分笔画, 而我们的模型需要对所有的汉字进行预测, 所以我们生成的目标关键点置信图中依然包含所有的笔画关键点信息。

我们需要为数据集中的每一张汉字图片根据这张图片所标注的关键点生成

对应的关键点置信图  $S^*$ ，由于一个汉字包含不同的关键点，一种类型的关键点可能存在多个，假设一共存在  $J$  种不同的关键点（本项目中  $J=87$ ），则每种关键点均用一张置信图表示，在表示第  $j$  个关键的置信图中，若是存在  $K$  个  $j$  类关键点，对于第  $k$  个该关键点，其周围的像素  $p$  的置信度可以用如下公式计算：

$$S_{j,k}^*(p) = \exp\left(-\frac{|p - x_{j,k}|_2^2}{\sigma^2}\right) \quad 6$$

第  $j$  种关键点的置信图中像素  $p$  的置信度由这  $K$  个关键点生成像素  $p$  的置信度相加得到：

$$S_j^*(p) = \max_k S_{j,k}^*(p) \quad 7$$































	横		横钩		弯钩
	竖		竖弯钩		横折弯钩
	撇		撇折		竖弯
	点		竖提		横折弯
	横折		竖折		横折折折钩
	捺		撇点		横斜钩
	提		竖折折钩		横折折撇
	横折钩		斜钩		竖折撇
	竖钩		横撇弯钩		竖折折
	横撇		横折提		横折折

Figure 4 汉字笔画与关键点

通过预测输入汉字图片对应的每种笔画关键点的置信图，可以得到每种关键点的

热力图,但是从热力图中提取关键点常常面临的一个困难是热力图中存在相邻的同种类型的关键点,这些点实际上表示的都是同一个点。在目标检测任务中同样存在着类似的问题,即检测出的重合度很大的同一目标的多个检测框,目标检测任务中常用 NMS 算法来解决这一问题,保留下置信度最高的检测框。不过 NMS 算法在关键点检测问题上却并不适用,因为 NMS 计算的是不同检测框之间的 IoU,检测点并不存在 IoU 这一概念。在我们的项目中,为了解决这一问题,我们使用 OKS (Object Keypoint Similarity) 来代替 IoU 作为衡量两个关键点相似程度的指标,使用 NMS 算法的变形 NMS-OKS 算法来去除重复的关键点,OKS 的计算公式如下所示:

$$OKS = \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right) \quad 8$$

其中  $d_i$  表示两个关键点之间的欧式距离,  $s$  在 COCO 数据集中表示检测目标的面积,这一数值在我们的项目中无法得到,我们将其与剩下的常数  $k_i$  作为一个常数,在项目测试过程中进行调整。

我们的模型在我们所标注的数据集上取得了显著的效果,图 5 是在“云”字上的热度图输出,左图是横的起始点的预测图,右图是所有关键点的预测图。

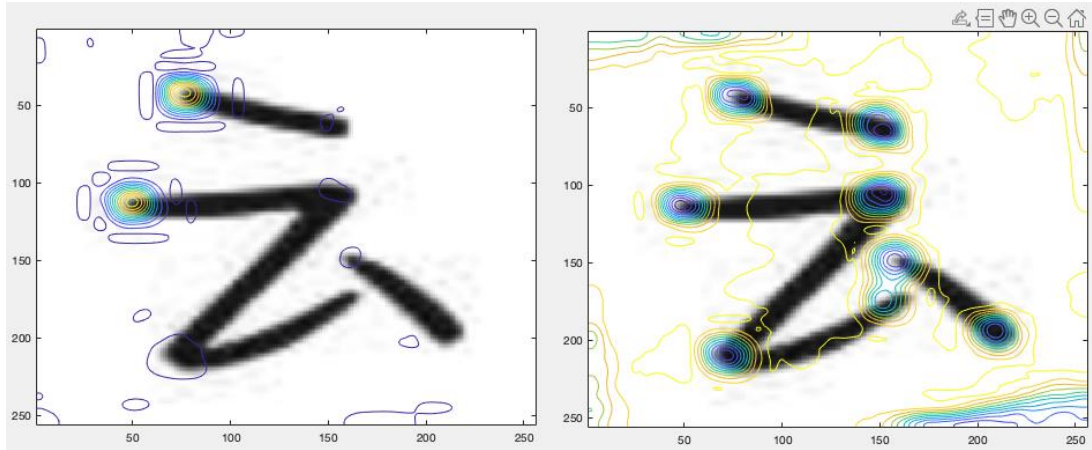


Figure 5 左图: 横的起始点的热度图, 右图: 所有关键点的热度图

### (3) 笔画方向关系场 PAFs

当检测出不同类型的关键点后,关键的问题就是这些关键点应该如何连接。如果是完全图,若是检测出  $N$  个关键点,则共有  $N!$  个连接,这些连接中大部分都是不正确的。实际上汉字的每一个笔画均可以看作由不同的线段连接而成,我们将这些线段称为连接,每个连接均包含起始点和结束点,则该问题就可以简化



对于所有的起始点如何连接它们对应的结束点，这样一个二分图匹配问题。例如对于图 5 中的“云”字，由于“云”中有两个横，检测出的关键点共有四个，横的起始点与结束点的连接也有四条，其中最上面和最下面是正确的连接。

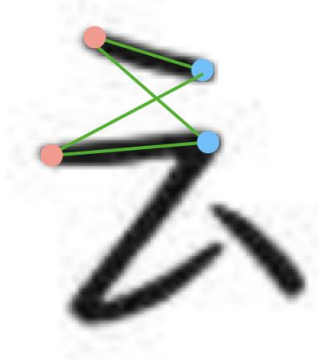


Figure 5 云字中横的二分图连接问题

为了区分正确的笔画连接与错误的笔画连接，我们从 OpenPose 模型中借鉴了 PAFs 这一想法。PAFs 对于每一个连接图中的每一个像素，均用一个二维的向量表示该像素所在位置的方向信息，因此 PAFs 可以用来编码每个笔画的书写方向。每张图片的目标 PAFs 信息（ground truth）是通过标注的关键点计算得到的。

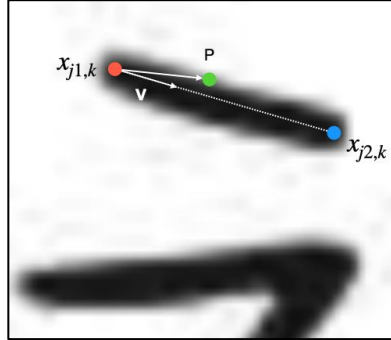


Figure 6 横的 PAFs 计算示意图

如图 6 所示，横的两个关键点  $x_{j1,k}$  与  $x_{j2,k}$  是由标注文件得到的，对于其他位置上的像素  $p$ ，若像素  $p$  在这个横上，则  $p$  上的方向向量等于从  $j1$  指向  $j2$  的单位向量。公式化描述为：

$$v = (x_{j2,k} - x_{j1,k}) / |x_{j2,k} - x_{j1,k}|_2 \quad 9$$

判断  $p$  是否在某一连接上有两条规则, (1)  $p$  的方向不能与  $v$  相反, (2)  $p$  距离  $v$  的距离不能超过一个预先设定的阈值 (我们的项目中设置为 5 个像素)。其公式化描述如下:

$$0 \leq v \cdot (p - x_{j_1,k}) \leq l_{c,k} \text{ and } |v_{\perp} \cdot (p - x_{j_1,k})| \leq \sigma_l \quad 10$$

在测试阶段, 通过计算候选的连接的方向向量与这一连接路径上的 PAFs 向量的点积的积分, 即可得到这一连接的分值。若是某一连接是正确连接, 这一连接的路径上的 PAFs 基本都与连接的单位向量处于相同方向, 若不是正确的连接, 连接路径上的大部分像素的 PAFs 均为 0, 他们与连接的单位向量的积分会小于正确的连接的积分。实际在计算时, 由于图像的像素是离散的值, 计算时通过对路径上的像素点采样得到一系列离散的点, 通过计算这些离散点与连接的单位向量的点积和即可得到不同连接的分值。其公式化描述如下:

$$E = \int_{u=0}^{u=1} L_c(p(u)) \cdot \frac{d_{j_2} - d_{j_1}}{|d_{j_2} - d_{j_1}|_2} du \quad 11$$

其中  $p(u)$  是对两个端点上的位置的插值函数。

$$p(u) = (1 - u)d_{j_1} + ud_{j_2} \quad 12$$

图 7 是我们的模型预测出的笔画横的 PAFs 方向场, 可以看到在横的笔画附近的像素处的向量均为正确的方向。

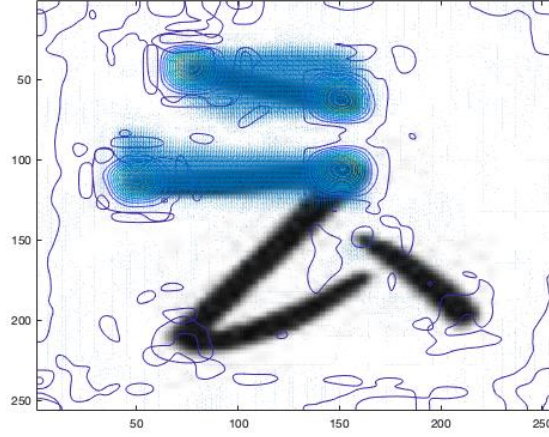


Figure 7 横的 PAFs 方向场

#### (4) 笔画集合分配算法

在获得了各个连接的分数的后, 需要根据检测出的字的种类判断需要保留多少连接。由于任何一个关键点只能是一个连接的起始点或结束点, 所以其入度和出度均不能大于 1, 也就是任何一个关键点只能和一个关键点相连。以上约束基本可以确定识别出的汉字的大致骨架, 不过由于我们并不清楚应该保留多少的连接, 所以最后检测的结果可能会出现多识别或少识别的问题。

为了解决这一问题, 我们引入了汉字的笔画信息, 根据目标检测模块检测出的汉字的信息, 从数据库中查询到该汉字含有的所有笔画, 构建一个包含所有笔画的集合, 从所有可能的连接中寻找满足该集合的候选, 该候选需要满足任意一个连接中的关键点的入度和出度最大为 1, 计算候选笔画集合的 PAFs 分数, 取分数最大的笔画集合作为识别出的最终笔画集合。由于任何一个关键点只能与另一个关键点相连或不存在连接, 这一问题可以看作二分匹配问题, 故可以使用匈牙利算法求解。假设待识别汉字的候选笔画集合为  $Z_c$ , 起始关键点的集合为  $D_{j1}$ , 结束关键点的集合为  $D_{j2}$ ,  $z_{j1,j2}^{m,n}$  表示属于  $j1$  的关键点  $m$  与属于  $j2$  的关键点  $n$  是否可以相连, 这一信息可以由预先定义的笔画关键点元数据中得到, 只有属于同一连接的两个关键点的  $z_{j1,j2}^{m,n}$  为 1, 其余情况均为 0,  $E_{mn}$  表示两个关键点  $m$  与  $n$  之间连接的分数的, 可以由他们路径上的 PAF 积分计算得到, 详细计算过程见 PAFs 部分。因此本算法的目标就是找到最大的候选笔画集合  $Z_c$ , 满足以下条件:

$$\begin{aligned}
 \max_{Z_c} E_c &= \max_{Z_c} \sum_{m \in D_{j1}} \sum_{n \in D_{j2}} E_{mn} \cdot z_{j1,j2}^{mn} \\
 \text{s.t.} \quad &\forall m \in D_{j1}, \sum_{n \in D_{j2}} z_{j1,j2}^{mn} \leq 1 \\
 &\forall n \in D_{j2}, \sum_{m \in D_{j1}} z_{j1,j2}^{mn} \leq 1
 \end{aligned}$$

图 8 是我们的方法识别出的汉字的笔画图，在不同的手写风格下，依然可以有较高的识别准确率。

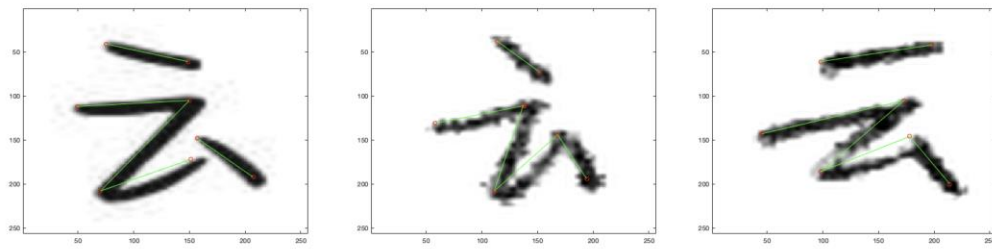


Figure 8 汉字骨架识别结果

### 3.2.2 汉字图像对齐

由于检测出来的汉字框的大小、拍摄的图片视角的不确定性，导致最后检测出来的汉字的角度和大小会有较大的差异。这样检测出的汉字不适合和标准字进行对比从而计算各个尺度。为了更好地计算标准字和待纠正汉字之间的差异，需要将待检测的汉字调整至正确的角度，并缩放至适合的大小。

对于待检测汉字的缩放方面，我们根据之前模型检测出的汉字骨架，计算骨架中所有关键点中上下左右四个方向最大的值，也就是可以完全覆盖汉字骨架的最小边框，用这个边框的左上端点和右下端点对汉字图像进行缩放，得到缩放后的汉字图像。

对于待检测汉字的旋转方面，需要使用深度学习模型预测旋转的角度，我们通过收集字体库中的汉字，将其旋转一定角度，之后使用深度学习模型预测该汉字旋转的角度，从而实现了汉字的旋转角度预测。结合汉字的缩放大小与旋转角度，便可得到适合进行对比计算的标准手写汉字图片。

### 3.2.3 汉字纠正

汉字纠正共分为三个模块，首先是计算输入汉字图片与模版汉字图片之间的最适合匹配位置，并生成这两者的蒙版，将其反馈给前端供用户查看自己书写的汉字与标准字之间的差异所在。其次是通过对比检测出的输入汉字的骨架与标准字之间骨架的差异，依据不同的尺度计算这两者的不同，并将问题反馈给用户。最后是根据识别出的汉字信息，在我们事先构建的专家系统中匹配该汉字对应的一系列需要注意的书写建议，根据检测出的汉字骨架判断是否存在相应的问题，

最后将问题与建议反馈给用户。

输入汉字与模版汉字最佳位置的匹配算法流程如下: 根据这两个汉字图片检测出的骨架, 在  $[min\_scale, max\_scale]$  的区间内对输入汉字图片进行缩放, 同时对输入汉字图片在  $[min\_degree, max\_degree]$  区间内进行旋转, 对于输入汉字转换后的每一个状态, 计算它与模版汉字所有关键点的 OKS 值并累加作为这两个图片都相似分数, OKS 的计算见公式 8, 最后取相似分最大的输入汉字的状态作为目标状态并进行转换, 对输入汉字与目标汉字分别做二值化处理并将结果反馈给前端。

输入汉字与模版汉字骨架的差异计算主要考虑了每个连接的长度和倾斜度两个方面, 对于这两个汉字骨架中的每一个连接, 通过计算每个连接的长度差异与倾斜度差异, 如果差异超过了允许范围, 则判断当前笔画的书写存在问题, 记录下问题最后将结果反馈给前端。

汉字纠正的最后一个模块是专家系统, 我们在数据库中提前录入书法书中的指导信息, 例如对各个部首的书写建议, 对汉字结构的书写建议, 并建立对应的匹配方式。通过汉字的目标检测环节可以得到汉字的分类信息, 根据数据库中预先建立好的汉字结构模型, 可以得到汉字的结构, 并判断不同笔画所属的结构。对于不同的结构, 将其递归分解为最小不可分结构, 也就是表一中的独体结构, 对独体结构中的所有笔画进行指导信息匹配, 若是该结构存在指导信息且问题且存在问题, 就将该指导信息反馈给用户。最后根据所有存在于汉字中的结构进行指导信息匹配, 若是存在相应的指导建议, 判断是否存在问题, 若是存在问题将结果反馈给用户。

Table 1 汉字结构分解表

结构方式	间架比例	样例	编号	元件编号	对应方式
独体结构	独立	米、日、云	1	1	直接对应
品字形结构	各部分相等	品、森、鑫	2	2、3、4	上、左下、右下
上下结构	上下相等	思、华	3	5、6	上、下
	上小下大	霜、花	4	7、8	上、下
	上大下小	基、想	5	9、10	上、下
上中下结构	上中下相等	意	6	11、12、13	上、中、下
	上中下不等	褒、裹	7	14、15、16	上、中、下
左右结构	左右相等	村、联	8	17、18	左、右
	左窄右宽	伟、搞	9	19、20	左、右



器和鉴别器在对抗中学习。每个生成器都试图“欺骗”相应的鉴别器，而鉴别器则学会了不被“欺骗”。为了使生成器保留原始输入文本的文字信息，CycleGAN 模型计算了循环一致性误差 (Cycle Consistency Loss)，该损失评估了往返于产生空间的图像在多少程度上与其原始版本相似。下面是 CycleGAN 模型的整体流程：

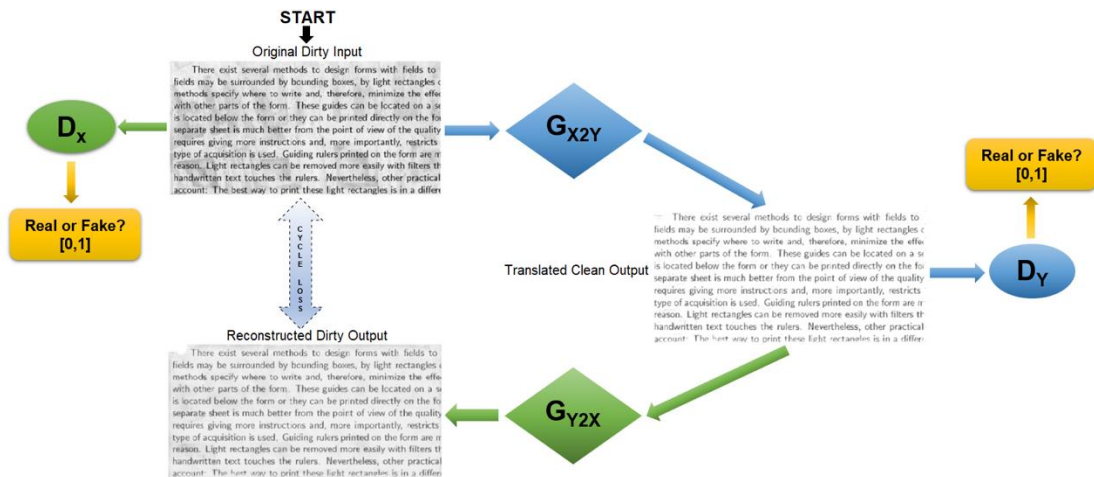


图 3.2.2 First Generator

第一个生成器  $G_{\{X2Y\}}$  将原始图片输入转换为清洗后的输出。鉴别器  $D_Y$  将尝试评估转换后的输出是真实图像还是生成器生成的图像。然后，鉴别器将提供所评估的图像是真实图像的可能性。

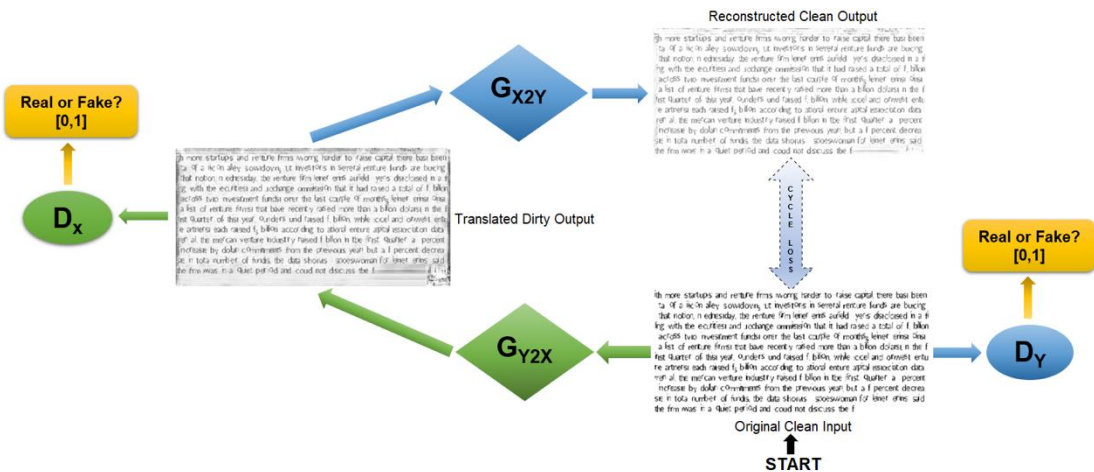


图 3.2.3 Second Generator

第二个生成器  $G_{\{Y2X\}}$  将原始的清洗后输入转换为转换后的有噪声的输出。鉴别器  $D_X$  将尝试从生成的图像中分辨出真实的图像。创建的模型将在两个方向上进行训练，分别带有一组有噪声的图像和一组无噪声图像。



### 3.2.5 手写字检测

手写字检测基于 Faster R-CNN 模型, Faster R-CNN 模型是 R-CNN 模型的第三个迭代版本, 其实现了端到端的检测, 因此可以有效利用 GPU 运算的并行性, 大大提升了目标检测的效率, 同时端到端的学习也提升了模型的特征学习效果, 使得模型的准确性也有所提升。Faster R-CNN 的整体架构如下图所示:

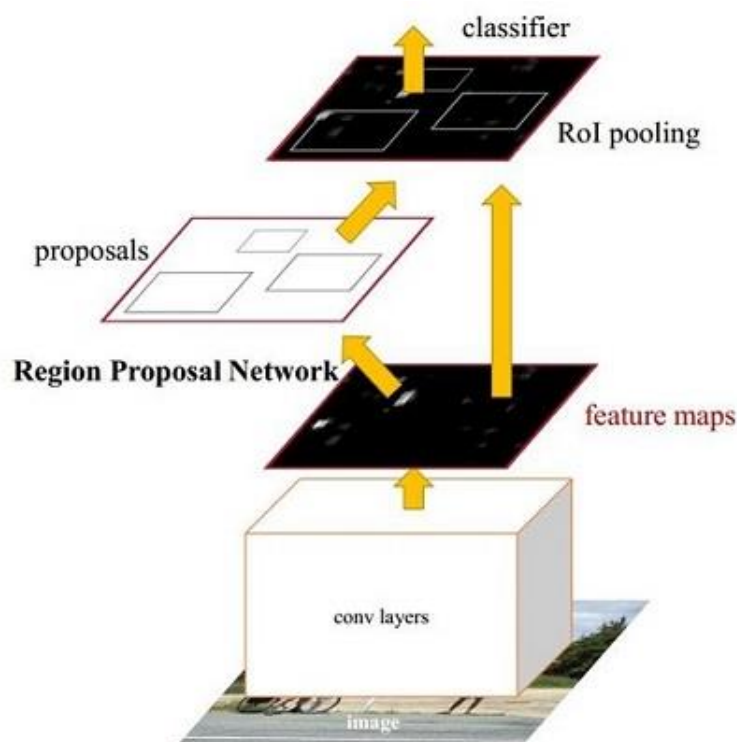


图 3.2.3 Region Proposal Network

Faster R-CNN 的核心部分是区域提出网络 (Region Proposal Network, RPN), RPN 可以单独从 Faster R-CNN 架构中抽出, 作为独立网络结构训练, RPN 的作用是针对给定的输入图片, 提取出可能包含目标的区域, RPN 的作用原理是将 Feature Maps 中的每一个点都作为 Anchor, 计算该点是否包含目标, 并且生成一个以这个点为基准的 Bounding Box。RPN 设置中预先设置了几个比例值, 对每个 Anchor 生成这几个比例值的 Bounding Box, 然后对每个 Bounding Box 计算其修正后的框的位置, 其包含四个值共包含四个值  $\{\Delta cx, \Delta cy, \Delta h, \Delta w\}$  代表 Bounding box 的中心的偏移量和 Bounding Box 的宽高的修正值。



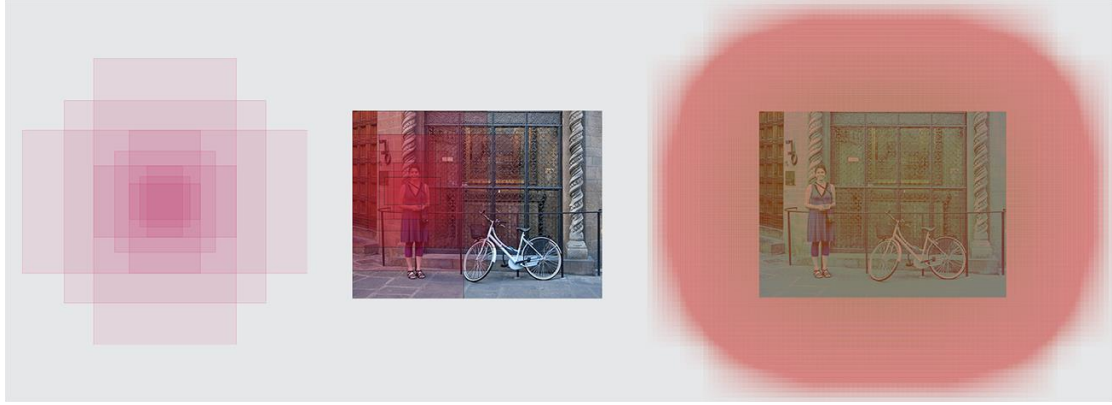


图 3.2.4 Different Ratio of Anchors

RPN 同时需要预测对应的 Bounding Box 中是否含有目标，这里并不是要检测出目标具体是什么，只需要区分 Anchor 所在的这个区域中是目标还是背景即可，所以其实可以看作一个二分类任务。RPN 识别出所有的 Bounding Box 后根据图片的 Bounding Box 与真实图片的 IoU 判断其是否包含目标。RPN 的 Loss 由两部分组成，一部分是分类的 Loss，另一部分是 Bounding Box 的四个值的回归 Loss，这两部分误差经反向传播后更新模型的参数。手写字检测任务只需要检测出当前字是否是手写字即可，所以其只需要两个标签，判断是否是手写字，因此可以直接使用 RPN 模型作为手写字识别模型。

### 3.2.6 整体书写评价

取图像分割模型中的单字分割结果序列作为输入，记为  $P =$

$\{p_1, p_2, \dots, p_n\}$ ，其中  $p_i = \{x_1, y_1, x_2, y_2\}$  代表单字的包围盒坐标。据此可以计算得到每一行字的重心方差  $S_g^2$ 、字距方差  $S_{space}^2$ 、单字面积方差  $S_{square}^2$ 、底部位置方差  $S_{bottom}^2$  以及

平均字距与平均单字面积比值  $E(P)$ 。最终文章的章法评价函数可以采用

$$E(Article) = S_g^2 + S_{space}^2 + S_{bottom}^2 + \frac{E(P)}{E_{std}(P)},$$

其中  $E_{std}(P)$  为预训练样本中得到的标准比值。

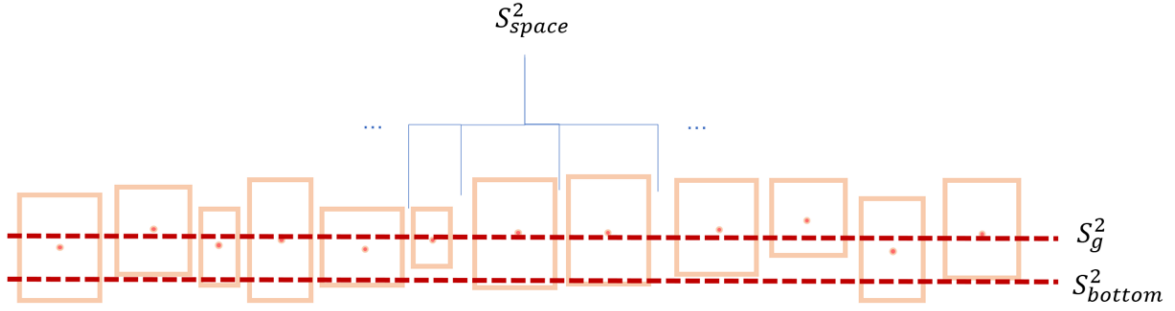


图 3.2.6 整体书写评价模型图

### 3.2.7 书法练习推荐

通过不确定近邻的协同过滤算法实现针对性推荐。由于易错字相似的用户群体在错字选择直接存在较强的相关度，因此可以根据基于用户之间易错字的相似性计算，来适当的选择预测目标的近邻作为推荐群，同时计算推荐群中对预测目标的信任子群，在充分结合推荐群和信任子群的基础上，通过不确定近邻因子分析度量来计算预测目标的推荐结果。该算法相比较传统的基于用户以及基于产品的协同过滤推荐算法，可以有效平衡用户群以及产品群推荐结果所带来不确定的影响，有效缓解用户评分数据极端稀疏情况使用传统性度量方法带来的问题，并显著提高推荐系统的推荐质量。算法包括以下五个步骤：

#### 1. 构建评分矩阵。

在推荐系统中，用户所有历史单字评分的数据库中包含  $s$  个用户的集合  $U = \{U_1, U_2, \dots, U_s\}$  和  $t$  个产品的集合  $I = \{I_1, I_2, \dots, I_t\}$ ，用户评分数据集可用一个  $s \times t$  阶矩阵表示， $s$  代表  $s$  个用户， $t$  列代表  $t$  个产品（见表 1）。假设某一用户  $U_a$  历史中  $I_j$  的评分为  $R_{a,j}$ ，则该评分体现了用户对单字的  $I_j$  的掌握程度。

	$I_1$	$\dots$	$I_j$	$\dots$	$I_t$
$U_1$	$R_{1,1}$	$\dots$	$R_{1,j}$	$\dots$	$R_{1,t}$
$\dots$	$R_{a,1}$	$\dots$	$R_{a,j}$	$\dots$	$R_{a,t}$
$\dots$	$R_{s,1}$	$\dots$	$R_{s,j}$	$\dots$	$R_{s,t}$

图 3.2.10 用户-单字评分矩阵  $R(s \times t)$ 

#### 2. 相似性计算

可以是用户之间的相似性计算，也可以是字形相似度间的计算。相关计算方法有夹角余弦、修正的夹角余弦、Pearson 相关系数、jaccard 相似度、巴氏系数等。

下列为采用修正的余弦相似度计算用户之间相似性，通过减去单字评分平均值将整体数据移动到空间原点。选取用户  $U_a$  和  $U_b$  打分的交集  $(I_{U_a} \cap I_{U_b})$ ，定义为

$$\text{sim}(U_a, U_b) = \frac{\sum_{i \in I'} (R_{a,k} - R_a) \times (R_{b,k} - R_b)}{\sqrt{\sum_{i_k \in I'} (R_{a,k} - R_a)^2} \times \sqrt{\sum_{i_k \in I'} (R_{b,k} - R_b)^2}}$$

其中  $R_a$  是用户对于  $a$  字得分的平均值，计算结果  $\text{sim}(U_a, U_b)$  的值落在  $[0, 1]$

区间中,  $\text{Sin}(U_a, U_b)$  值越大, 则表示用户  $U_a$  和  $U_b$  之间的相似性越高。基于产品的相似度计算过程与之类似。

### 3. 动态选择目标的推荐对象群

在进行邻近对象选择之前, 需要界定预测目标的推荐对象应该如何选取, 通过定义两个相似度计算的阈值, 只考虑选择与目标较为接近的作为推荐对象, 定义

$$S(U_a) = \{U_x | \text{sim}'(U_a, U_x) > \mu, a \neq x\}$$

$$S(I_j) = \{I_y | \text{Sim}'(I_j, I_y) > \nu, j \neq y\}$$

### 4. 在推荐对象中选择信任子群

针对目标进行推荐对象选择过程中, 相似度计算成了主要的衡量指标, 但是, 在实际的推荐系统中, 往往用户的相似度计算, 可能仅仅来源于对少数几个字的得分, 甚至可能只有一个共同评分的字, 这样的相似度计算, 存在较大的偶然因素。因此, 除了要考虑相似度, 也需要考虑两者之间共同评价单字的个数。计算共同打分数大于设定的阈值的用户推荐群, 定义为  $S'(U_a)$ , 计算目标项目推荐准确度较高的信任因子, 定义为  $S'(I_j)$ 。

$$S'(U_a) = \{U_x | \text{Sim}'(U_a, U_x) > \mu \& |I_{U_a} \cap I_{U_x}| > \varepsilon, a \neq x\}$$

$$S'(I_j) = \{I_y | \text{sim}'(I_j, I_y) > \nu \& |U_{I_j} \cap U_{I_y}| > \gamma, j \neq y\}$$

计算两个信任子群的对象个数, 分别计算  $|S'(U_a)| = m'$  和  $|S'(I_j)| = n'$

### 5. 不确定近邻的协同过滤算法

对于目标的在线用户  $U_a$  以及其他未浏览过的单字  $I_j$ , 同时结合用户的最近邻集和单字的最近邻集对用户单字上的得分进行预测, 推荐公式为:

$$R_{a,j} = \lambda \times \left( R_a + \frac{\sum_{x \in S(U_a)} \text{sim}'(U_a, U_x) \times (R_{x,j} - R_x)}{\sum_{U_x \in S(U_a)} \text{Sim}'(U_a, U_x)} \right) + (1 - \lambda) \times \left( R_i + \frac{\sum_{l_y \in S(I_j)} \text{sim}'(I_j, I_y) \times (R_{a,y} - \bar{R}_y)}{\sum_{l_y \in S(I_j)} \text{sim}'(I_j, I_y)} \right)$$

其中  $R_a, R_b$  分别表示用户  $U_a, U_b$ , 对其他字所有得分的均值,  $R_j, R_y$  表示字  $I_j, I_y$  已知所有用户得分的均值, 公式中根据用户  $U_a$  和字  $I_j$  的不确定近邻群进行推荐, 假如用户  $U_a$  的近邻群为空, 则完全按照字  $I_j$  的近邻群进行协同过滤, 若  $I_j$  的近邻群为空, 则完全按照用户  $U_a$  的近邻群进行协同过滤。

### 3.3 结果期望

功能	技术问题	结果期望
书写图像检测	图像降噪	可以明显地去除原始图片中的噪声, 因光线引起的像素值变化, 多余的线条、污点等。
	手写字检测	可以检测出所有手写的字, 并精确地生成包含单个字的边框。
	整体书写评价	可以生成多个维度的评价分数, 该分数应该是可解释的。
	书写报告生成	生成用户可以理解的一段话, 报告书写的好坏情况。
书写识别与纠正	汉字骨架识别	对于输入的汉字图片可以准确地预测出汉字的骨架, 可以判断出书写不清出的笔画并预测出大致位置。
	汉字图像对齐	可以将汉字图像调整至合适的大小和角度。
	汉字纠正	根据书写的汉字, 可以正确地反馈汉字的纠正信息。
	汉字评分	对与输入对汉字, 给出合理的评分。

## 4 技术实践

### 4.1 使用的开发框架及依赖的库

技术	框架	依赖库
深度学习模型	PyTorch	numpy, torchvision, pillow, opencv, scipy, matplotlib, tensorboard

### 4.2 技术实践过程

#### 4.2.1 汉字骨架识别

##### (1) 深度学习模型

模型结构如下图所示,我们使用的汉字骨架识别所使用的深度学习模型由骨干网络(ResNet50)、PAFs 预测分支与置信图预测分支组成。由于汉字识别任务预测的结果较为复杂,直接在数据集上进行学习很难收敛到全局最优点,经常面临欠拟合的问题。为了解决这一问题,我们提取出其骨干网络并构建了一个汉字分类网络,在 CAISA 数据集上先进行汉字分类任务,分类模型在 CAISA 数据集上可以达到 96% 的准确率。在训练完骨干网络后,我们将其学习到的模型权重加载到骨架识别模型中再进行骨架识别任务。

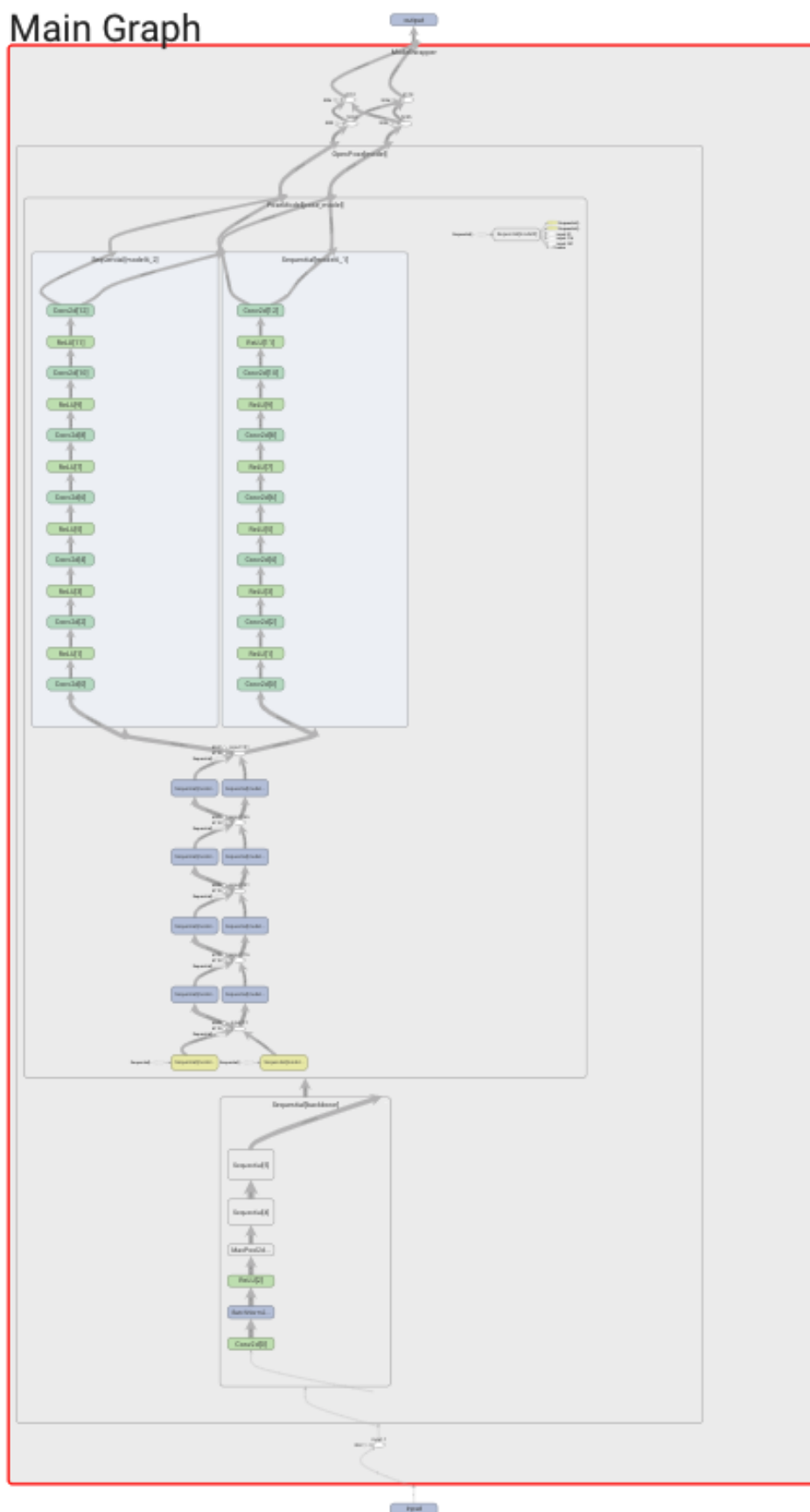


Figure 5 汉字骨架识别模型结构

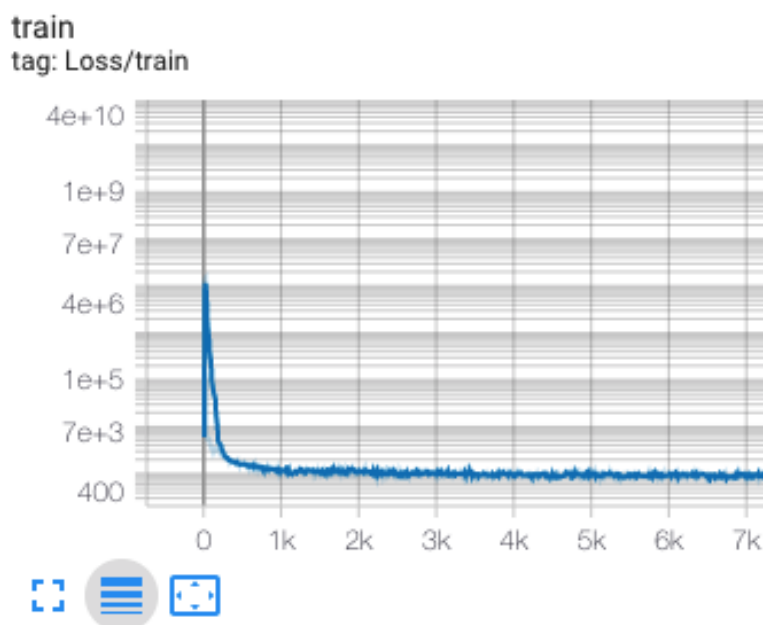


Figure 6 汉字骨架模型训练 loss

## (2) 数据集标注

由于现有的数据集并没有适合汉字骨架识别任务的标注信息，我们手动对收集到的汉字图片进行了标注作为训练汉字骨架识别模型的数据集。我们初步标注了 600 张“云”与 600 张“烟”字的图片作为模型的原始数据集，之后使用图像增强方法，对每张图片生成了 500 张增强后的图片用作训练，此时训练集由  $500 \times 1200 = 600000$  张图片构成，从预测结果显示效果十分理想。

为了加快我们的标注速度，提升标注的准确性，我们开发了一款适合于标注汉字关键点信息的数据集标注软件。该软件可以根据标注汉字的信息，自动定位到下一个待标注的笔画，并根据笔画的信息快速地得到下一个需要标注的关键点，从而实现快速的标注。标注软件界面与功能如下图所示：



Figure 7 我们开发的数据集标注工具

#### 4.2.2 汉字纠正

汉字纠正部分的技术难点在于专家系统的指导匹配,这部分需要首先录入专家的指导信息,之后根据汉字本身的结构信息与识别出的关键点信息匹配对应的指导建议。

在专家指导建议方面,我们收集了《田英章硬笔书法》中的部分指导建议,并根据其对应的笔画结构存入了数据库中,部分指导信息见下图。通过根据不同关键点之间的距离以及笔画的倾斜度信息可以匹配到相应的指导信息。

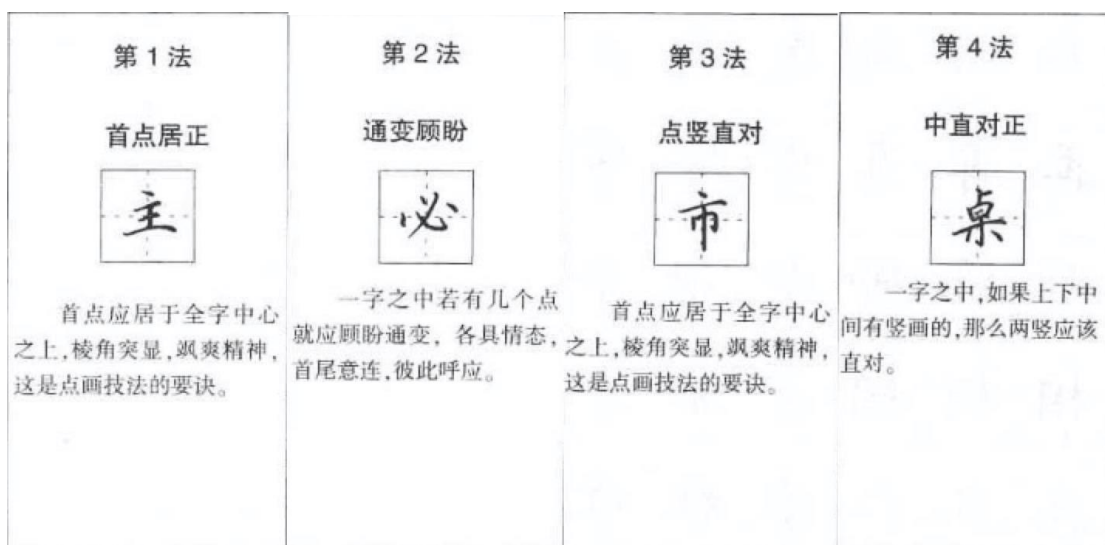


Figure 8 部分指导建议



汉字纠正面临的第二个问题是汉字信息的建模问题，汉字种类繁多，如果对每个汉字都分别建模，那么工作量将会非常大。我们将汉字按结构分解为可以复用的独立结构，并对独立结构就行纠正，最后对整体的结构进行纠正，可以有效利用汉字中相同的结构，降低工作量。汉字的结构建模方式如下所示：

```
{
  "characters": [
    {
      "id": int,
      "name": str,
      "structure": int,
      // 是否为最小构成单元
      "is_unit": int,
      "note": str,
    }
  ],

  "structures": [
    {
      "id": int,
      "name": str,
      "superstructure": str,
      "components": [
        id_1,
        id_2,
        ...
      ]
    }
  ],

  "components": [
    {
      "id": int,
      "name": str,
    }
  ]
}
```

## 5 结果验证

### 5.1.1 单字识别与分割

单字检测与分割算法可以检测出图片中存在的书写汉字,并识别出该汉字的边框,在我们的传入的测试图片中可以取得良好的效果,平均耗时在 2s 以内。以下是测试结果:

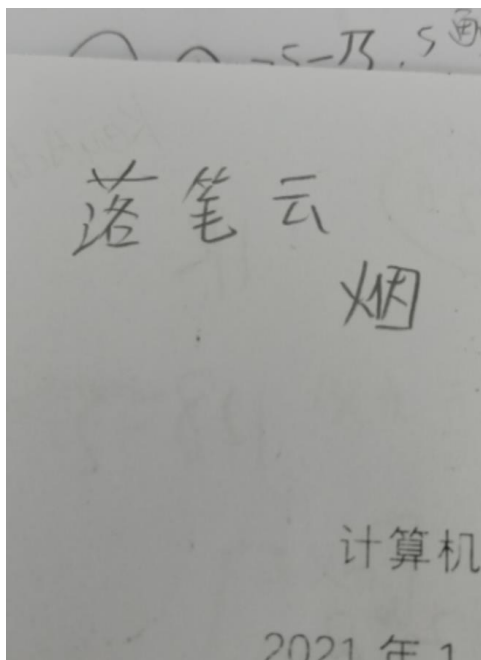


Figure 9 传入的图片

部分测试结果:

```
{
  "code": 10,
  "detail": [
    {
      "charId": 37,
      "childId": 1,
      "filename": "刀 1617959148.3122954.png",
      "filepath": "/static/crop_res/刀 1617959148.3122954.png",
      "judgId": "607018eb746624b0b095821b",
      "location": {
        "height": 37,
        "left": 233,
        "top": 8,
        "width": 23
      },
      "multi": 0,
```

```
    "name": "乃",
    "rating": 0.8,
    "times": 1,
    "username": "xuerunzhen"
  },
  {
    "charId": 2206,
    "childId": 1,
    "filename": "画 1617959148.3202343.png",
    "filepath": "/static/crop_res/画 1617959148.3202343.png",
    "judgeld": "607018eb746624b0b095821b",
    "location": {
      "height": 37,
      "left": 314,
      "top": 7,
      "width": 10
    },
    "multi": 0,
    "name": "画",
    "rating": 0.8,
    "times": 1,
    "username": "xuerunzhen"
  },
  ..
}
```

### 5.1.2 汉字骨架检测

汉字骨架检测可以识别出输入汉字图片的骨架,在我们已经标注的汉字上均可取得良好的效果,对于不同风格的书写汉字,我们的模型均可准确识别出该汉字中存在的对应笔画与关键点所在的位置。我们的模型同时支持 CPU 与 GPU 运行,在服务区的 CPU 上运行的平均时间在 3s 左右,以下是测试结果:



Figure 10 汉字骨架识别测试结果

### 5.1.3 汉字纠正

汉字纠正算法可以匹配书写汉字的书写缺陷，并提供对应的指导建议，本算法的平均运行时间在 1s 左右，以下是测试结果：

