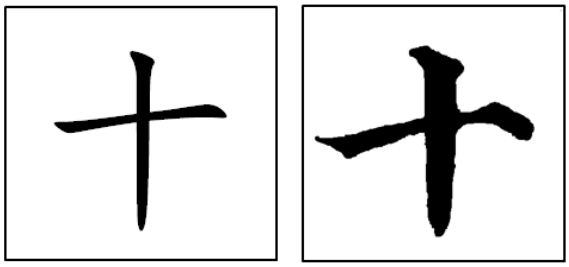
abstract

汉字笔画拆分的难点是分割同时属于两个笔画以上的交叉区域。与有着较强规律性标准的硬笔书法相比，软笔书法有更多的不规则笔画形式。因此，我们提出了一个多标签语义分割插件 Stroke-Seg 来提取软笔手写体汉字的笔画。我们不仅将汉字的类型和笔画数作为先验知识结合起来，还提出了一次输出所有笔画的多标签分割概念，并定义了其损失函数。此外，为了验证其可行性，构建了毛笔书法笔画分割数据集(BCSS)，并从笔画层面提出了新的评价标准。通过在 BCSS 上的实验，我们的网络在笔画层面的准确率达到 99.6%，比现有的语义分割方法提高了 3.8%。此外，在其他不同字体上进行测试时，Stroko-Net 也显示出良好的通用性。

1. Introduction

为了更加有效得帮助人们学习中国书法汉字，我们将人工智能与书法结合，构建了书法评测系统。要想进一步提升书法美学测评的精准性，能够给出有针对性的指导意见，需要进一步从笔画角度进行测评，这就需要首先将汉字中的每个笔画进行拆分。现有拆分方法多研究笔画更为标准的印刷体和硬笔（用钢笔）汉字。然而正如图所示，本研究需要拆分的软笔汉字（用毛笔）相比之下笔画更加多变，起承转合使其形态更为丰富。因此，我们基于先前的工作，在已有基础上进一步开展基于语义分割网络进行软笔手写体笔画拆分的研究。



pen-and-ink calligraphy　　soft brush calligraphy

中国汉字的笔画分割相比其他文字更加复杂，不同笔画之间的交错重叠和笔画的多样性提升了处理难度。主要的难点在于不同笔画之间的交叉点区域同时属于两个甚至多个笔画，但常规的图像分割方法每个像素点只能分为一类。因此，至今有关汉字笔画拆分的研究主要便是针对交叉点不同的处理方法展开的。无论是细化汉字使用骨架等信息提取笔画，还是在深度学习网络中将交叉区域作为独立笔画新的一类进行分割，都让笔画的原有信息的缺失。这对于软笔书法的汉字分割将更加突出。若每次只分割一种笔画，的确可以解决交叉点的问题，但是计算成本过高。所以我们希望提出一种方法能够在一个网络中，让每个像素点（特别是交叉点区域）都能够被赋予多个标签，进而同时完整地输出所有笔画。

汉字笔画交叉区域的重叠问题有属于这一问题独一无二的特性——既不是如今较为热门的如 CoCo 数据集中物体的重叠问题【C】，这种问题中研究的两个物体之间有着明确的重叠与被重叠关系，每个像素点依旧只归属于一个物体。也与医学中同时分割器官和其中的病灶问题有差异【医学中出现的多标签分割】，他处理的是从属关系的多标签分割:病灶区域在器官内，且这种问题可以在分割出器官后再提取其中的病灶区域【】，对多标签的需求并不大。相比之下，汉字的交叉点所属的两个笔画是相互独立的，与图像多标签分类问题在概念上更加接近，是一种新的多标签的图像分割问题。

因此，我们提出了基于 transunet 架构的多标签分类的分割方法‘网络名称’，which can 将物体按照类别数 n 一次性输出 n 张 mask 图。当然，这种方法并不局限于某一种网络架构，而是通过新的输出模式，构建了一种多标签分割的体系化方法。未来验证方法的有效性，我们在 e3c 数据集的基础上进一步构建了目前最大的软笔汉字笔画分割数据集 ccss。测试结果显示网络能够很好地分割出手写体字的所有笔画，并有着一定的泛用性。此外，我们还基于其他分割网络，包括 fcn、unet、deeplabv3 和 transdeeplab 进行了验证，实验结果都证明本方法是成功的。

我们的贡献点如下：

（1）构建了最大的软笔手写书法笔画拆分数据集，并在数据集中提出“笔画类”的概念解决了部分类别笔画数目过少的问题；

（2）我们提出了一种能够适用于多标签分割任务的网络架构。经测试，这种架构使用于绝大多数分割网络；

（3）结合汉字分割任务本身的特殊性，一方面我们在网络中加入先验知识，帮助网络更好地理解拆分问题。另一方面以笔画为基础提出了新的评价标准，更好评价网络理解汉字的性能。

2. related work

2.1 semantic segementation

语义分割是像素级别的图像识别和理解[1]。随着第一个语义分割的主流网络 FCN 网络的提出[6]，相关研究相继展开。虽然 FCN 网络有着结构简洁高效的优点，但过于简单的结构对于复杂问题有些乏力，这促使许多经典的网络在其之后提出，包括 U-Net 和 Deeplabv3 等。卷积神经网络（CNN）计算参数少、提取局部信息能力高的特点使其大放异彩，一时间占领了深度学习的主导地位。纵观笔画分割领域内深度学习网络的发展，有许多研究都基于 CNN 网络进行笔画拆分【】。我们也在实验中基于这三种经典网络进行了对比实验。

transformer 提出后，x 将其应用到了图像领域，x 提出的 vit 则让 transformer 更加贴合图像特征，且进一步减少计算复杂度，再加之其优良的提取全局信息性能，让 vit 块大受欢迎。图像分割任务也随着等人将 unet 模型与 transformer 结合提出 transunet 迈上了一个新台阶。在此之后，大家开始着眼于更多的分割模型结构，基于 vit 块许多 CNN 中经典的网络架构都与 transformer 结合焕发出来新的生机。例如等人就在近期提出了 transdeeplab,将 deeplabv3 的卷积都使用 transformer 结构进行替换。对于笔画分割来说，transformer 全局信息的能力正是其所需要的。因此本文基于 transunet 完成 “”的构建，并使用 transdeeplab 进行了对比实验。据我们所知，“”是第一个将 transformer 应用于笔画拆分的网络。

2.2 书法笔画拆分

汉字本身作为一种文字，其有许多特点可以挖掘，加之较强的规律性与每个人不同的书写习惯，早期研究笔画拆分更倾向于使用更加方便设计的传统机器学习方法。但是大多数研究在有规律可寻的标准印刷体上使用角点检测、骨架提取等手工特征完成，手写体较为复杂，手写字体的不确定性导致骨架中的笔画不再连续，相关研究较少，直到近些年来，随着深度学习等理论的完善与进步，相关研究才有所增加，【pr】其结合手工特征和神经网络，使用细化的方法提取汉字的骨架后再进行分割，取得了喜人的成果。不过这种方法更适用于硬笔，对于我们的软笔应用场景会丢失笔画的信息，而且从应用场景出发，我们的项目致力于为书法笔画评价打下基础，因此在原图中直接分割是必要的。

已有研究也使用了一些方法来解决交叉点的问题，如【deepstroke】选择将交叉点作为独立笔画新的一类，这种方法将原本的一个笔画分成了三个部分，会导致网络学习不到笔画的完整信息，在分割时一个笔画分成多段对网络会产生一定的误导。当然也有论文【41】为了避免笔画交叉区域的问题，将每类笔画使用不同的网络分割，这种方法在分割所有笔画时需要多个网络进行多次处理，代价太大。Bi 等[18] 构建了全笔画网络分割模型，网络可以一次性输出所有笔画。但其笔画没有固定分组，每一张图片的输出只是输出一个固定的笔画，网络学习到的信息只是每一个训练集中字的笔画拆分方法，而非笔画拆分系统。【C】的方法思路和数据集形式与我们不同，他使用双阶段的实例分割的方法，在原图中检测到每一个笔画后再在感兴趣框内分割笔画，其数据集由硬笔字组成。我们希望提出一种方法能够在一个网络中同时完整地输出所有笔画，并减小计算开支，提升网络的鲁棒性。因此我们提出了基于多标签分类的笔画分割模型‘网络名称’。

3. Experimental and discussions

dataset collection and annotation

我们在数据集 E3C 的基础上制作了软笔汉字笔画分割数据集 CCSS（Chinese Calligraphy strokes segementation）。E3C 数据集包含了 40 张模板图片与 11008 幅

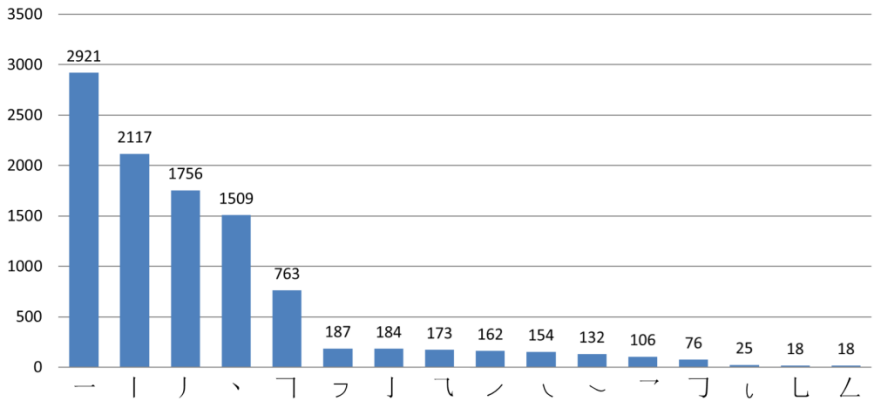
书法副本图像，图像采集于湖南美术出版社书法教材的三至六年级的小学生在书法课程中的临摹练习作业。所有原始的书法临摹图像均在标准光照下，使用智能手机的摄像机按照 1：1 的长宽比进行捕获。

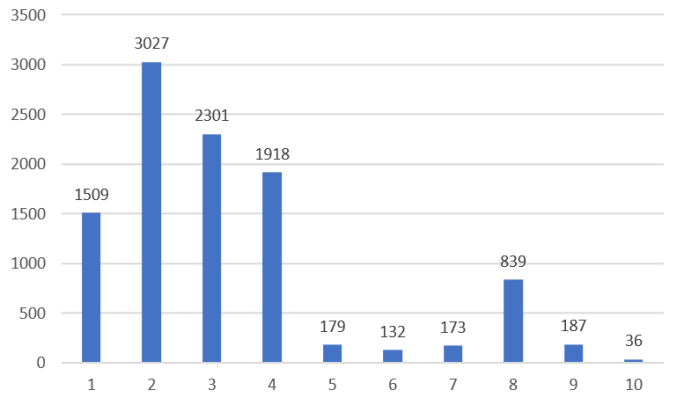
CCSS 数据集来自于 E3C，我们从间架结构和笔画类型等方面进行考量，在其中选取了 26 个具有代表性结构的汉字，包含 24 个标准字体和手写的 998 张书法副本（calligraphy copy image）共 1022 张图片，总计 19 种 10446 个笔画。有我们在对图片进行了二值化预处理后，将汉字的每一个笔画进行人工标注，最终构建完成数据集。图为部分例子【图 1】。

Label processing

虽然中国汉字的笔画共 32 种，但是笔画的出现频率有很大差距，部分笔画结构复杂并不常见，多出现在生僻字中，因此我们的数据集涵盖的更为常用的 16 中笔画能够包含大多数常用汉字，对于未出现的笔画，我们也将归为一类进行处理。如图【图】为 CCSS 中笔画的统计情况，可以看到常用的笔画在汉字中出现概率很高，我们选取的 16 种笔画之间已经出现了频率差距很大的问题，这种严重的不平衡问题会影响到算法的性能。因此，在汉字中相同笔画不会有交叉区域的前提下，我们提出将笔画按类别定义标签，一是考虑笔画的形状特点，二则考虑汉字中笔画间的位置关系，避免在一类中出现交叉点问题，最终将 16 种笔画归为 10 个类别，对应关系如表所示【表】。最终 10 类的笔画分布如图，其中最后一类是不在前 9 类的笔画范围内的其他笔画类型，这一类中

包含了数据集较少的两类笔画以及数据集中不包含的其他笔画类型，进而给网络留出可扩展的空间，给网络能将所有笔画都预测成功的可能性。





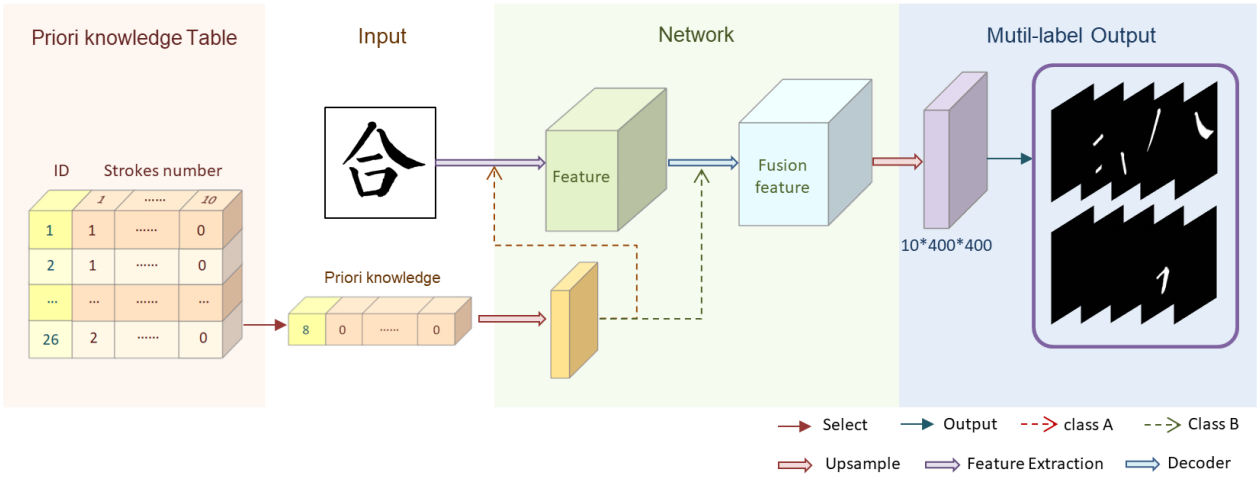
Class	笔画种类	笔画个数	类别笔画总数
1	点（丶）	1509	1509
2	横（一）	2921	3027
	横沟（𠂇）	106	
3	竖（丨）	2117	2301
	竖钩（亅）	184	
4	撇（丿）	1756	1918

	提（㇒）	162	
5	捺（㇏）	154	179
	捺钩（㇌）	26	
6	斜钩（㇀）	132	132
7	横斜钩（㇇）	173	173
8	横折钩（㇆）	76	839
	横折（㇕）	763	
9	横撇（㇇）	187	187
10	竖弯钩（㇄）	18	36
	撇折（㇚）	18	
	……（㇋、㇈、㇉）	/	

4. the proposed stroke segementation method

如图【图】所示，我们的笔画分割网络由以下几个三个组成：先验知识特征层，分割网络框架和多标签输出层。我们结合笔画分割任务多标签输出的需求和汉字本身的特点，加入了先验知识特征层和多标签输出层构建成网络。至于 network 部分，所有的语

义分割网络经过调整，均能够作为网络框架使用，这里我们结合先验知识层主要介绍性能最优的 transunet 版本模型。



4.1 先验知识特征层

构建

汉字笔画拆分与医学影像或其他分割任务不同，其有着明确是背景和前景关系，网络只需要考虑笔画的部分如何拆分即可。但笔画的拆分有以下几个难点：（1）笔画之间错综复杂，每个交叉点都可以组成不同的笔画组合；（2）不同种类的笔画有部分形态相似，例如图【图】中，横和点的形态相似，在分类时容易混淆；（3）复杂的笔画是可以拆解成为简单笔画的，例如图中的‘横折’就可以拆解成为‘横’和‘竖’；‘撇折’可拆解为‘撇’和‘横’，因此具体的分割方法和分割类别网络本身很难界定。

汉字笔画的种类是有限的，在知晓汉字是什么之后，笔画的种类及其数量则也可以得知。目前已经有了较为成熟的印刷体和手写汉字识别技术，汉字信息获取对于书法智能评价应用场景或其他应用是较为简单的工作。而当得知汉字种类及其笔画类型和数量这一重要的先验知识后，网络则已知了汉字中有什么笔画以及每一类笔画的数量，而这些内容是与网络的每一维输出图像中所包含的笔画数(mask 区域的数量)是一一对应的，因此理解并分割汉字笔画将更加容易。

我们围绕着数据集 CCSS 构建了汉字的类型及其笔画和笔画数量信息表。信息表中包含了汉字种类和笔画信息两部分。汉字种类部分编码了 ccss 中的 26 种汉字的编号，对于训练集中不包含的汉字，我们统一使用编号‘27’提示网络输入的图片是一个全新的汉字。笔画信息部分则涵盖了 ccss 和测试集中所有汉字的笔画信息。最终，我们构建了一个大小为 x*10 的信息表，用于生成输入网络的特征层。

构建与引入

如图【图】，信息表中的信息进行编码后即可生成先验知识特征层并加入到网络当中。对于汉字和笔画这样重要的信息我们希望网络能够充分的学习，因此希望将其加入得浅一些【知乎连接】,以减少信息的缺失。在网络实际构建时，我们将 unet 和 transunet 归为 a 类网络结构，由于这类网络在 upsample 时其浅层特征是逐层加和的，在浅层网络中能够获得先验知识信息，所以还是将先验特征层与输入图像一起输入网络，使得先验知识充分利用。此外，由于 fcn 本身结构较为简单，实验也证明由于特征层是直接与网络的特征层拼合的，在较浅层再加入先验知识层反而会误导网络将先验知识认为是图片的一部分，进而得到错误的掩码输出，因此 fcn 网络也被归为了 a 类网络。对于 deeplab 和 transdeeplab 这一类网络我们则归为 b 类，其特点是有着提取特征的 encoder 和汇总特征信息并输出最终结构的 decoder 两个模块。则在 decoder 部分与浅层特征和深层特征拼接，一同完成 decoder 部分。

具体加入时，需要对信息进行归一化的操作，使其与图像的值范围相同，防止信息的真值数字过大影响到网络的计算，具体计算公式如下：

$$V_{facture} = \frac{V_{\text{info}}}{n_{\text{character}}} \times \max\left(V_{\text{layer}}\right)$$

$$V_{facture} = \frac{V_{\text{info}}}{n_{\text{stroke}}} \times \max\left(V_{\text{layer}}\right)$$

其中，$V_{facture}$是特征层的输入值，V_{info} 是信息的真值，$n_{\text{character}}$ 是汉字种类编号的最大值，这里设为 27，n_{stroke} 是笔画的最大阈值，考虑到汉字中除过很少的偏僻字，基本不会有一类笔画的数量过多，因此本文设为 15。V_{layer}则是拼合层的阈值上限，例如 a

类网络中，先验知识层将与归一化后的图片一起进入网络，那么V_{layer}的值则为图片归一化后的最大值 1。

4.2 多标签输出

普通的分割任务中每个像素点只能分为一类，而在笔画分割任务中，交叉点部分的像素点属于多个笔画，及像素点有多个类别归属。由于希望输出一次性输出汉字所有的笔画以提升系统效率，我们改进输出部分的网络结构，使得网络能够一次性完成单张汉字图片与多类笔画之间的映射。对于给定的书法字符图像，我们将该字符分割到的笔画表示为$S = s_1, s_2 \cdots, s_{10}$。则图像 X 及其映射关系如下:

$$S = N(P_{stroke})$$

其中，N 代表网络函数，P_{stroke}为汉字图像，S 代表网络的输出,其维数为(10, w, h)，是 10 张与原始输入图像大小相同的每个类别的笔画结果。在判定像素点的分类时，每个类别的判定独立计算，计算公式如下：

$$r_p = (o_p \geq \lambda) \times 1 + (o_p < \lambda) \times 0$$

其中o_p是像素点某一类的预测输出，λ为设定的判定阈值，r_p指的是pix级别的$result$，文中实验所取的λ值为 0.6 。

正如前文所述，汉字笔画分割问题变成了一个多标签分割问题，那么原有的损失函数也不再适用，我们需要重新定义一个损失函数。从数学定义出发，多标签分割将多分类问题化为了多维度的二项分布问题，其实每一类的标签只有两种可能：“本像素点是此类笔画”或者“本像素点不是此类笔画”。在本文的笔画分割中，我们结合参考了 BCE 和 Dice 两种损失函数的特点——BCE 是从微观上逐像素进行拉近，而 Dice Loss 相当于从全局上进行考察，提出了适用于笔画分割任务的损失函数 BDLoss（L_S），其定义如下：

$$L_S = \sum_i^{10} \mu_i L_{s_i}, \ \mu_i = \frac{n_i}{\sum_1^{10} n_i}$$

$$L_{s_i} = \sigma L_{S_i - B} + (1 - \sigma) L_{S_i - D}$$

其中，L_{s_i}为某一类笔画所对应的输出第 i 维所对应的损失函数，其有被分为L_{S_i-B}和L_{S_i-D}两部分。σ是两种损失对应的权重，本文设为 0.5。μ_i为每一类笔画的损失权重，权重的设置主要是考虑到网络包含了 10 维代表着每一类型笔画的输出，且从 3.3 可以得知这些类别包含笔画的数量差距较大，因此在计算损失函数时，我们将每一类得到的L_{s_i}的公式按数据的概率分布值进行加权。

对于具体的损失函数L_{S_i-B}和L_{S_i-D}，其具体定义如下：

$$L_{S_i-B} = -\sum_j y_{t_j} \log y_{o_j} + \left(1 - y_{t_j}\right) \log \left(1 - y_{o_j}\right)$$

$$L_{S_i-D} = 1 - \frac{\sum_{\mathrm{j}} 2y_{t_j} y_{o_j}}{\sum_{\mathrm{j}} \left(y_{t_j}^2 + y_{o_j}^2\right)}$$

其中y_{o_j} is the j-th pixel in the output map of 网络。y_{t_j}是 j 目标图像中对应位置的值。

5. Experimental and discussions

5.1 Evaluation metrics

考虑到笔画分割任务的目的是分割出完整的笔画，便于下游任务使用。那么在评价网络的性能优劣时，原有的评价标准已经不能完全适用，除过逐像素点计算的常用评价标准，我们还将从笔画层面进行结果评价。

像素级的评价标准

像素级的评价标准我们首先选取了 ACC 和 miou 两个评价标准进行评价。此外，考虑到本任务主要关注的对象还是笔画本身的分割效果，而背景部分却占比很大，导致了结果不能够客观反映笔画分割的情况。因此，我们又提出了两种评价标准 stroke-recall 和 stroke-iou 以关注笔画的分割优劣。计算公式如下：

$$ACC = \frac{P_{TP} + P_{TH}}{P_{TP} + P_{FP} + P_{FN} + P_{TN}}$$

$$M - IoU = \frac{1}{2}\left(\frac{Y_{t-B} \cap Y_{o-B}}{Y_{t-B} \cup Y_{o-B}} + \frac{Y_{t-S} \cap Y_{o-S}}{Y_{t-S} \cup Y_{o-S}}\right)$$

$$\text{Stroke} - \text{Recall} = \frac{P_{TP}}{P_{TP} + P_{FN}}$$

$$\text{Stroke} - IoU = \frac{Y_{t-S} \cap Y_{o-S}}{Y_{t-S} \cup Y_{o-S}}$$

其中，P_{TP}和P_{TN}代表实际和预测均为笔画和背景的像素点，P_{FP}和P_{FN}则代表实际与预测不相符的像素点。Y_{t-B}和Y_{o-B}分别代表背景的实际值和预测值，同理，Y_{t-S}和Y_{o-S}分别代表笔画的实际值和预测值。

笔画级评价标准

通用的像素点级别评价标准用于评价某一个像素点的分割结果是否正确进而验证网络的通用性能，而笔画分割更多的是在关注分割得到的笔画是否能为下游任务所用，因此除过像素级别的评价之外，我们还提出从笔画层面评价笔画的分割优劣。笔画分割的错误分为错分和漏分（多分），为了更加全面的评价我们提出了两种评价标准综合考量网络性能。第一种是笔画准确率 RSAcc，评价的是以笔画为单位的笔画分割准确率，当预测结果和目标值的 iou 大于 0.9 时则认为笔画分割正确。在【C】中也提出了相似的方法，但是由于硬笔和软笔的笔画粗细形态等有所差异，因此在硬笔中被视为标准苛刻的标准在软笔中是较为容易达成的，计算公式如下：

$$\text{RSAcc} = \frac{1}{n}\sum^{n}\frac{n_r}{n_c}$$

其中，n是测试集的样本数量，N_r是一个字中笔画拆分正确的笔画数量，N_c是汉字的笔画总数，最后将测试集中的所有的汉字进行分别计算后取平均即可得到最终的笔画准确率。

第二个是笔画错分率 WSAcc，笔画拆分中可能会出现错分、漏分以及一个笔画被多次分割到不同种类的笔画中等等情况，其中，因为笔画错分的数量可能会超过笔画原有数量，但笔画的总数是衡量笔画分割水平的重要标准，笔画越多则笔画分割难度应当越高，因此评价的总体部分依旧类似于 RSAcc 的架构，使用错误笔画除以笔画总数，此外由于错误笔画数可能会大于笔画总数，为防止评价标准数字膨胀，评价使用反正弦函数将结果归至 0 到 1 之间。计算公式如下：

$$\mathrm{WSAcc} = \frac{1}{n}\sum^{n}\frac{\arctan\frac{n_w}{n_s}}{\frac{\pi}{2}}$$

其中，N_w是笔画错分为多段、漏分以及多分等情况的笔画数量，只要出现了不应该有的 mask 结果，则均认为是错误分割的笔画。

5.2 Implementation details

In the implementation, we trained the network in 200 epochs using a batch size of 4, with NVIDA RTX 3090 GPU and PyTorch 1.12. The experiment is based on transunet, fcn, unet, deeplabv3 and transdeeplab 5 network frameworks for training. In addition to transdeeplab, we evaluation our model at a resolution of 400*400, according to the parameter, transdeeplab using the size of 384*384. We use Adam optimizer with (0.9, 0.990) betas, and base learning rate varies from 0.001 to 0.005 depending on the kind of network.

5.3 Implementation Results

本节首先讨论了网络损失函数、先验知识部分对网络的提升。之后我们使用多标签网络分割框架进行了不同网络的改造，讨论不同网络的性能与区别。然后我们与现有的几个方法进行比较，证明我们的网络性能优异。最后，我们选取不在 CCSS 数据集中的图像进行笔画分割，并展示了部分分割结果。

5.3.1 消融实验

为了验证损失函数和先验知识对网络的作用，我们基通过对网络结构中的损失函数和先验特征模块进行消融实验来证明方法的有效性。其中，损失函数我们分别使用 bce、dice 和 bdloss 三种进行对比，结果如表所示。

损失函数	先验特征	ACC	$M-IoU$	Stroke − Recall	Stroke $-IoU$	RSAcc	WSAcc
Bce	No	**98.96**	94.35	90.47	88.93	90.18%	0.072
Dice	No	/	/	/	/	/	/
BDLoss	No	96.29	**94.53**	**96.06**	**89.27**	**94.53%**	**0.06**
Bce	Yes	98.79	93.39	88.77	87.04	94.95%	0.05
Dice	Yes	98.74	93.51	89.12	87.29	95.52%	0.056
BDLoss	Yes	**99.18**	**97.24**	**96.02**	**94.58**	**99.58%**	**0.005**

可以看到在没有先验特征模块时，注重全局信息的 dice 损失函数的网络不能收敛，而关注微观上像素状态的 bce 函数虽然正确率高于本文提出的损失函数，但从笔画任务角度分析，更加重要的笔画级评价 rsacc 和 wsacc 都还有一定的差距。则从侧面证明笔

画分割任务是需要网络对汉子图像在宏观与微观角度兼并的一项图像分割任务。当加入先验知识模块后，网络性能总体上有很大的提升，最终的结果RSAcc达到了 99.58%，较没有加入时提升了 5.05%，而 wsacc 只有 0.005，较之前降低了 0.055，极大提升了网络的性能。

我们还注意到，实验结果有两处有趣的现象。第一，bce 函数的像素级评价标准都低于没有加入先验特征时的结果，而笔画级的评价标准则高于没有加入时的结果，说明在没有先验知识加入时，尽管笔画分割地更加完整，但有部分笔画网络没有分割正确。我们认为这种情况的发生是因为先验知识是汉字在全局上的一项重要特征，因此更加关注宏观信息的笔画级评价标准会在加入先验知识后有所提升，而更加关注微观细节时，则在像素级评价上有所优势。第二，dice 函数的网络在加入了先验知识后网络能够收敛并有着不错的分割结果，这里先验知识为网络指明了学习的方向，起到了很大的作用。

5.3.2 网络

除过使用 transunet 作为网络框架，为了验证网络的泛用性，我们还基于其他的分割网络进行了实验，包括 deeplabv3、fcn、unet 和 transdeeplab，结果如表所示。

Backbone	损失函数	先验特征	ACC	$M-IoU$	Stroke − Recall	Stroke $-IoU$	RSAcc	WSAcc
fcn	Bce	No	96.25	93.96	94.83	88.14	88.19%	0.1
	Dice	No	81.45	72.6	63.47	46.54	70.15%	0.281
	BDLoss	No	**97.96**	**96.34**	**96.55**	**92.81**	**96.26**	**0.036**
	Bce	Yes	95.73	93.85	95.71	87.94	96.00%	0.052

	Dice	Yes	82.32	81.18	96.18	63.06	70.23	0.186
	BDLoss	Yes	**98**	**96.39**	**96.59**	**92.91**	**98.40%**	**0.023**
unet	Bce	No	95.87	93.43	94.42	87.11	93.44%	0.082
	Dice	No	82.08	80.66	94.81	62.05	69.85%	0.2
	BDLoss	No	**97.64**	**95.76**	**95.96**	**91.68**	**95.93**	**0.042**
	Bce	Yes	95.97	93.61	94.63	87.47	95.52%	0.111
	Dice	Yes	90.76	84.15	81.41	68.98	89.97%	0.075
	BDLoss	Yes	**97.71**	**95.82**	**95.96**	**91.8**	**96.86%**	**0.045**
deeplab	Bce	No	97.73	96.28	**96.9**	92.69	94.53%	0.059
	Dice	No	/	/	/	/	/	/
	BDLoss	No	**98.51**	**97.01**	96.87	**94.14**	**94.62**	**0.049**
	Bce	Yes	98.02	96.49	96.77	93.11	96.69%	0.04
	Dice	Yes	89.75	88.41	96.37	77.25	38.32%	0.118
	BDLoss	Yes	**98.57**	**97.25**	**97.27**	**94.6**	**97.09%**	**0.029**
transdeeplab	Bce	No	92.81	88.21	88.05	76.9	73.42%	0.261
	Dice	No	75.95	72.34	78.97	45.8	62.77%	0.251

	BDLoss	No	**95.24**	**91.42**	**90.98**	**83.18**	**89.64**	**0.113**
	Bce	Yes	92.96	88.27	87.9	77	75.22%	0.245
	Dice	Yes	78.69	75.34	83.34	51.66	70.58%	0.234
	BDLoss	Yes	**95.17**	**91.45**	**91.21**	**83.24**	**90.58%**	**0.118**

单独分析每个网络都可以得到与 5.3.2 中所述都基本相同的结论：在加入了先验知识后，网络性能都有着显著的提升，而损失函数方面，同时兼顾微观和宏观的 bdloss 较相同情况下其他损失函数都有着更好的表现。纵观几个网络的表现， transunet 框架下的网络效果最好，其次是三个经典的 CNN 网络 deeplabv3、fcn 和 unet，最后是 transdeeplab。正如 2.1 所说，transformer 模块更加善于提取全局信息，而实验证明笔画分割任务需要兼顾宏观和微观信息，因此我们认为将 deeplab 所有模块都替换为 transformer 结构的 transdeeplab 并不能很好地发挥其优势。

5.3.3 对比实验

除过使用不同的 backbone 进行实验，我们还使用了其他两个研究提出的笔画分割方法在 CCSS 数据集上进行实验，这两种方法与本文的‘网络名称’都属于语义分割范畴的汉字笔画分割方法。【deepstroke】选择将交叉点作为独立笔画新的一类，与简笔画中的笔迹分割思路相似，是现有方法中一种典型的笔画分割方法。Bi 等[18]全笔画网络分割模型能够一次性拆分所有笔画，尽管笔画的输出较为冗杂，但是是软笔数据集中少有的研究。

实验结果如表所示。以【deepstroke】为代表的交叉点单独提取的方法虽然更加便捷，但是完整的笔画被交叉区域截断一定程度上影响了分割的性能，且在分割之后这一类方法还需要做笔画拼接，较为冗杂。 Bi 等[18]的网络将每个笔画作为一个输出，其

笔画没有固定分组，因此我们分析，网络学习到的信息只是每一个训练集中字的笔画拆分方法，限制了其性能。

	ACC	$M-IoU$	Stroke $-$ Recall	Stroke $-IoU$	RSAcc	WSAcc
Bi 等[18]	68.79	67.46	85.12	67.26	43.93	0.349
Ours	**99.18**	**97.24**	**96.02**	**94.58**	**99.58%**	**0.005**

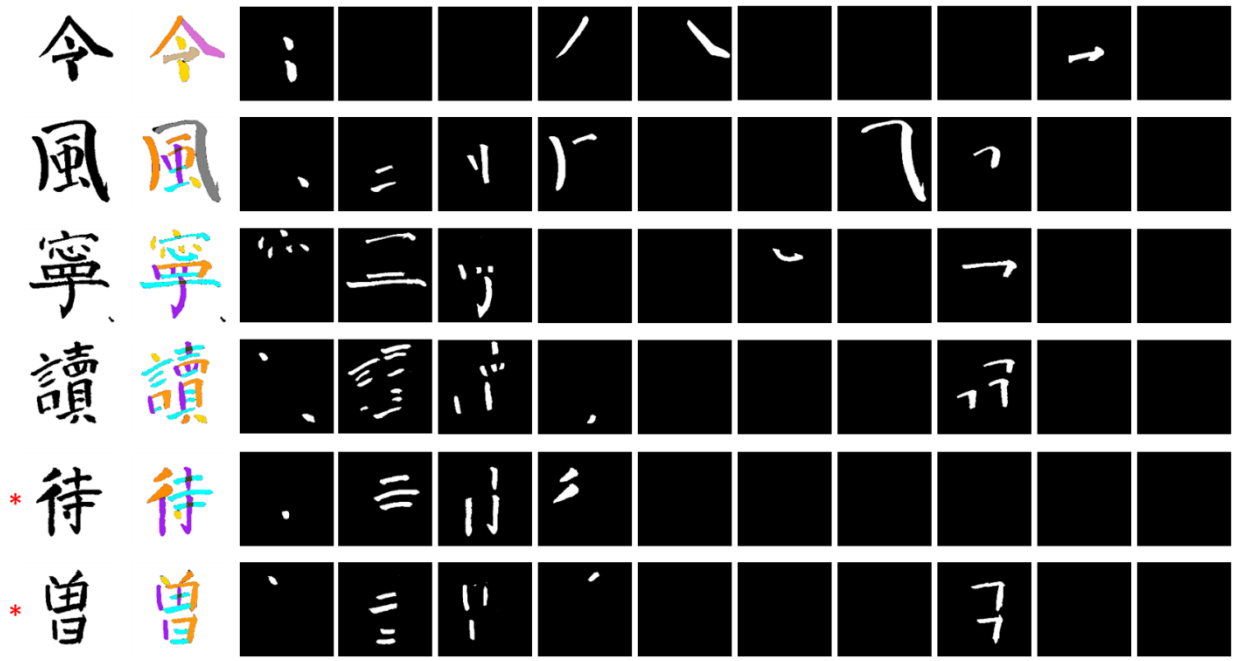
5.3.4 main results

本小节将展示我们网络在测试集上的结果，测试集包括 CCSS 中的 26 种汉字和除此之外的其他汉字来验证网络是否具有泛用性。测试集来源由两部分组成。一部分取自于 E3C 但不在 CCSS 中的书法图像。另一部分则是广泛采集的不同种风格的字，其中既有硬笔中楷体等印刷字体和 CCSE-W 数据集中的手写体图像，也有隶书、行书等软笔书法图像。最终，我们一共收集了 256 张测试图片，其中未出现过的字占比 51.56%。我们人工对结果进行判定：按照本文提出的RSAcc和 WSAcc 的计算方法，按照笔画是否能够为下游任务所使用的评价标准得到测试集的评价结果 RSAcc-H 和 WSAcc-H。具体信息如表所示。其中出现过的汉字正确率明显更高，也符合基本常识。

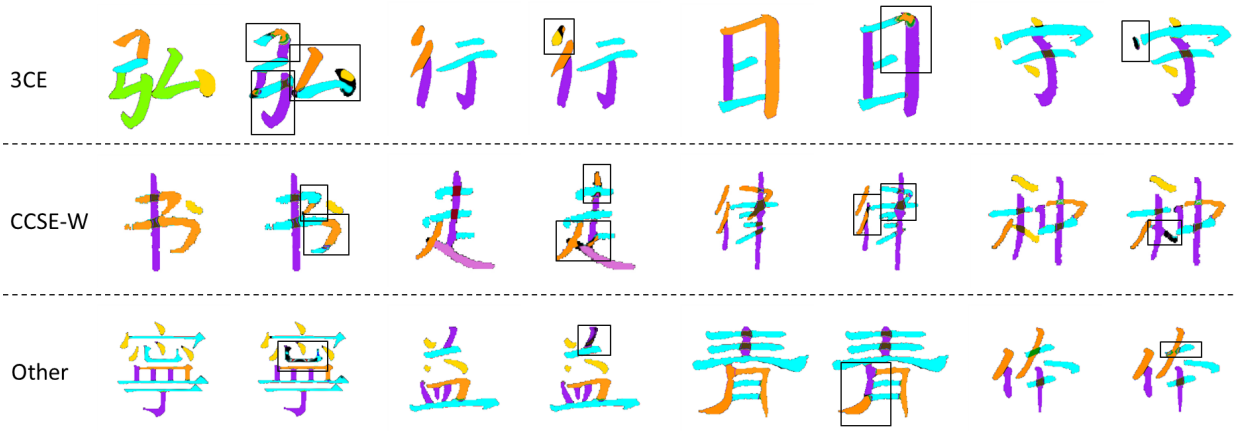
	CCSS 所含的 26 种字			CCSS 中没有的字			总计		
	数量	RSAcc	WSAcc	数量	RSAcc	WSAcc	数量	RSAcc	WSAcc
E3C	60	98.34	0.077	30	82.12	0.141	90	96.27	0.098

CCSE-W	4	98.33	0.011	65	81.33	0.152	69	82.32	0.144
Other	60	95.53	0.040	37	89.94	0.093	97		
总计	124	96.98	0.057	132	93.92	0.132	256	95.40	0.096

如图【图】所示是网络在测试集上的表现，图（a）详细展示了网络的分割结果，其中第一列输入的汉字图像，第二列从整体上展示了汉字分割结果，随后是网络的 10 张输出图片，分别代表着 3.3 中所列的 10 类笔画的分割结果。其中红色星号标出的汉字是 ccss 数据集中没有的汉字，从结果上看，拆分地效果尚佳。图（b）则展示了更多的分割结果，包括 3ce 数据集、CCSE-W 数据集和其他数据集中的汉字图像，同样红色星号标出的汉字未在 CCSS 中出现。令人欣喜的是本文提出的网络是基于软笔书法数据集进行设计并训练的，而在其他的硬笔印刷体和手写体数据集中依旧能够正确地完成分割，并且事实证明网络对未出现过，甚至较为复杂的汉字有一定的处理能力。



图【图】展示的是一些匹配错误的案例，左边给出了正确的分割图，右边则是错误分割掩码。错误主要分为几种。第一，对没有出现过的复杂笔画网络会拆分成基本笔画。例如第一个错误案例中存在两个数据集中未出现过的笔画，且都被拆分为组成这两个笔画基本笔画：“竖折勾”被拆分成了横和竖，“撇折”则拆分为撇和横。这说明是网络能够理解笔画的基本特征，但是在训练笔画有限的情况下网络并不能辨识所有笔画。第二，对于比较相似的笔画网络较难区分其种类。例如第二个错误将形似‘点’的笔画‘撇’错分为笔画‘点’，针对形态上较为相似的笔画如何正确分割这一问题还需继续探索。第三，对训练集中没有出现过的汉字网络的正确率明显降低，网络理解训练集中没有的汉字结构的能力较差。



5. conclusion

本文构建了第一个大型软笔汉字笔画分割数据集，为开展进一步工作奠定了坚实的基础。此外，我们还提出了“网络”算法，以高效地进行笔画拆分。为了充分利用汉字的先验信息，我们根据汉字特点构建了先验知识层帮助网络进一步理解。为了简化笔画分割流程，一次性输出所有笔画信息，我们基于 5 种网络框架改进了输出结构及损失函数，取得了不错的结果。实验结果证明，在 CCSS 数据集上训练的网络具有一定的泛用性，能够成功拆分硬笔印刷体和手写体的汉字，甚至是未在训练集中出现过的汉字。我们未来的工作将进一步完善算法并丰富数据集中汉字的种类，进一步提升笔画拆分的性能，便利下游任务如书法美学评价等进一步发展。