北京大学学报(自然科学版) 第 60 卷 第 6 期 2024 年 11 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 60, No. 6 (Nov. 2024)
doi: 10.13209/j.0479-8023.2024.088

基于图像分割技术的智能造字研究

蒋建斌[1,2] 黄松[1] 吴建国[1,†]

1. 北京大学工学院, 北京 100871; 2. 北京北大方正电子有限公司, 北京 100086; † 通信作者, E-mail: j.wu@pku.edu.cn

摘要 目前, 基于风格学习的智能造字技术生成的字体与用户手写风格相似度低, 基于GPU (graphics processing unit)风格迁移的方法成本高昂。为解决上述问题, 利用深度学习和图像分割技术, 提出一种新型智能造字方法, 在保持高度相似风格的同时, 满足用户个性化需求, 并降低成本。采用DeepLab v3+技术, 用户输入的775个字体图像经过数据质量评估模型筛选后, 通过图像分割模型进行部件拆分, 然后精细地调整部件并去除噪点, 最终矢量化后生成TrueType字体。与现有技术相比, 该方法能够显著地提升相似度并降低成本, 可以有效地满足用户个性化定制需求。

关键词 智能造字; DeepLab v3+; 数据质量评估模型; 图像分割模型; TrueType

An Image Segmentation Based Technology for Intelligent Character Creation

JIANG Jianbin[1,2], HUANG Song[1], WU Jiangguo[1,†]

1. College of Engineering, Peking University, Beijing 100871; 2. Beijing Founder Electronics Co., Ltd, Beijing 100086; † Corresponding author, E-mail: j.wu@pku.edu.cn

Abstract Currently, the intelligent character creation technology based on style learning generates fonts with low similarity to the user's handwritten style, and the method based on Graphics Processing Unit (GPU) style transfer is expensive. To solve the problems above, a new intelligent character creation method is proposed using deep learning and image segmentation technology, which can maintain a highly similar style while meeting the personalized needs and reducing the generation cost. Using DeepLab v3+ technology, 775 font images input by users are filtered by a data quality evaluation model and are split to components by image segmentation models. Then, components are finely adjusted and noise is removed, and finally TrueType fonts are generated after vectorization. Compared with existing technologies, this method significantly improves the similarity and reduces the cost, and can effectively meet the personalized customization needs of users.

Key words intelligent character creation; DeepLab v3+; data quality evaluation model; image segmentation model; Truetype

随着数字化和信息化技术的不断发展, 人们开始追求数字产品的个性表达, 字体逐渐融入人们的生活。自1960年上海印刷技术研究院着手"黑宋仿楷"字体开发到2012年的50多年间, 国内市场上中文字体的总量为421款[1]。2013—2017年, 仅3~5家字体厂商在各手机主题商店售卖字体, 2017—2019年, 字体厂商的数量迅速发展到近100家, 字体数量达到10000款以上。如今, 随着厂商平台的不断涌现和个性字体的发展, 市场总量字体超过20000款, 其中个性化字体超过50%。企业为提升用户体验, 增加个性化字体的制作, 如手机系统字定制, 车企品牌宣传字体等。个人用户亦热衷购买和应用个性化字体, 市场需求逐年增长。面对企业高昂的制作成本和用户个性化需求, 亟需发展新的字体生成技术, 用低成本来满足大量用户个性化定制需求。

国家自然科学基金(72171003, 71932006)资助
收稿日期: 2024–01–08; 修回日期: 2024–06–28

智能造字最早由 Xu 等[2]在 2009 年提出。智能造字技术将汉字分解为部件, 并通过部件复用拼接成汉字, 用户仅输入少量手写样本, 系统即可生成相似风格的汉字。该技术将字视为图像进行处理, 随着机器学习在图像生成方面的发展, 智能造字领域出现 zi2zi[3]、北京大学王选计算机研究所的智能造字系列[4–8]、DeepLab v3+[9]、Cycle-GAN[10]以及 DG-FONT[11]等方法。在高质量矢量字体生成方法中, 有 SCFont[12]、DeepVecFont[13]、DeepVecFont-v2[14]和基于扩散模型的 UDiffText[15]等方法。

随着字库的需求增长, 智能造字技术不断进步。国内智能造字分为专业与娱乐两大类, 专业造字采用 GPU (graphics processing unit)风格迁移, 基于专业设计师的设计生成字体, 广泛应用于企业用字和手机主题商店, 风格相似度高, 成本也高。娱乐型造字通过风格学习, 基于用户书写生成字体, 用于手机系统或输入法等场景中, 成本低, 错字率也低, 风格相似度一般。

本文采用分类模型和相似度模型等机器学习方法, 并结合 DeepLab v3+等深度学习工具, 提出一种基于图像分割的智能造字技术。该技术通过构建数据质量评估模型、图像分割模型和字体生成模型, 在提升字体风格相似度的同时降低生成成本。最后, 分析并评估图像分割技术在智能造字领域的应用效果。

1 DeepLab v3+图像分割

DeepLab 是谷歌的语义分割算法系列, 可提取图像特征并进行精准分割。自 2014 年起, 共推出 4 个版本, 其中 v3+包含 Fully Convolutional Networks (FCN)[16]、Xception[17]、深度可分卷积[17]、空洞卷积[16]、Atrous Spatial Pyramid Pooling (ASPP) [18]和编码器–解码器等部分。

FCN 是图像语义分割全卷积网络, 包括全卷积和反卷积。全卷积(例如 Visual Geometry Group (VGG)[19]和 ResNet[20]等)用于提取特征, 反卷积进行上采样, 获得与原图同样大小的输出结果。Xception 基于 Inception 结构[21–22], 采用深度可分卷积(包括逐通道卷积和逐点卷积)进行改进, 其结构如图 1 所示。

逐通道卷积将原图像分成 N 个通道, 每一个通道被 $n\times n$ 卷积核卷积, 生成 N 个特征图谱。例如, 5×5×3 的彩色图片经 3×3 卷积核卷积运算, 因有 3 个通道, 故产生 3 个 3×3 的特征图谱。若宽度为 W, 高度为 H, 逐通道卷积部分的参数个数 $N_{\mathrm{Depthwise}}$ 和计算量 $C_{\mathrm{Depthwise}}$ 可表示为

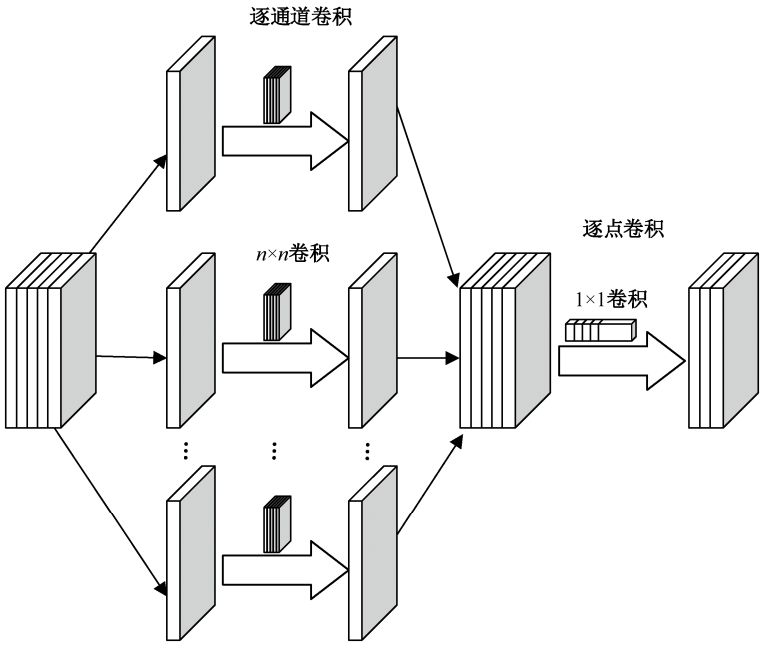

图 1 深度可分卷积结构示意图[9]

Fig. 1 Detailed structural diagram of depthwise separable convolution[9]

$$N_{\text{Depthwise}}=W\times H\times N,$$

$$C_{\text{Depthwise}}=W\times n\times(W-n+1)\times(H-n+1)\times N\text{。}$$

在示例中, $N_{\text{Depthwise}}$ 和 $C_{\text{Depthwise}}$ 分别为

$$N_{\text{Depthwise}}=3\times3\times3=27, \tag{1}$$

$$C_{\text{Depthwise}}=3\times3\times(5-2)\times(5-2)\times3=243\text{。} \tag{2}$$

逐通道卷积保持特征图谱数量 N 不变。如需生成新特征图谱, 需要逐点卷积。逐点卷积 N 个特征图谱与 M 个 1×1 的卷积核进行卷积运算, 产生 M 个新特征图谱。若宽度为 W, 高度为 H, 输入通道数为 N, 输出通道数为 M, 则逐点卷积部分的参数个数 $N_{\text{Pointwise}}$ 和计算量 $C_{\text{Pointwise}}$ 可表示为

$$N_{\text{Pointwise}}=1\times1\times N\times M,$$

$$C_{\text{Pointwise}}=1\times1\times W\times H\times N\times M\text{。}$$

示例中, $N_{\text{Pointwise}}$ 和 $C_{\text{Pointwise}}$ 分别为

$$N_{\text{Pointwise}}=1\times1\times3\times4=12, \tag{3}$$

$$C_{\text{Pointwise}}=1\times1\times3\times3\times3\times4=108\text{。} \tag{4}$$

经过逐点卷积后, 得到 4 个特征图谱, 维度与常规卷积输出一致。

ASPP 通过不同采样率的空洞卷积网络, 解决检测目标大小差异问题。空洞卷积通过插入行扩张卷积核, 基于扩张率确定插值行数, 旨在扩大感受野和捕获多尺度上下文信息。深度神经网络中为了增加感受野且降低计算量, 一般进行降采样。增加感受野会降低空间分辨率。相比降采样, 空洞卷积在不降低分辨率的同时, 扩大感受野且降低计算量, 适用于检测分割大目标和精确定位小目标。空洞卷积可以通过设置不同的扩张率参数给网络带来不同的感受野, 即获取多尺度信息。空洞卷积的实际卷积核大小 K 为

$$K=k+(k-1)(r-1), \tag{5}$$

其中, k 为原始卷积核大小, r 为空间卷积的扩张率。例如, 一个 3×3 的卷积核, 通过扩张率 2 插值, 实际卷积核大小为

$$K=3+(3-1)(2-1)=5, \tag{6}$$

即变成一个 5×5 的新卷积核。ASPP 网络设计 4 种不同的采样率(6, 12, 18 和 24), 对输入的特征图谱使用空洞卷积, 对应的 4 个空洞卷积的卷积核为 3×3。

编码器–解码器网络的结构如图 2 所示。编码器通过空洞卷积和 ASPP 模块检测猫图片目标。ASPP 模块通过 1×1 卷积和采样率为 6, 12 和 18 的 3×3 卷积进行池化, 检测空洞卷积提取的特征。解码器通过空洞卷积连接低级特征, 经 1×1 卷积、上采样 4 倍与编码器检测后的特征连接, 再通过 3×3 的卷积特征细化, 最终上采样 4 倍输出预测结果。

2 模型原理

本文提出的智能造字算法运用图像分割技术, 包含数据质量评估模型、图像分割模型和字体生成模型三部分。数据质量评估模型是对输入手写字体图像的质量进行判定, 图像分割模型是将手写字体图像按照字体结构分割成多个部件, 字体生成模型是将图像分割得到的字体部件去噪后组合, 通过矢量化生成个性化字体。

2.1 数据质量评估模型

数据质量评估模型通过 TfClasifier 分类模型训练得到, 用于评估用户手写字体的质量, 筛选高质量字体用于图像分割。TfClasifier 使用 Inception v3 模型, 用于解决分类问题。首先, 对输入图像进行预处理, 调整大小并归一化像素。然后, 通过模型进行前向传播。Inception v3 的总体架构如表 1 所示。模型由 10 层深度学习网络构成, 包括卷积、池化和线性插值等, 每层具有特定的匹配大小和输入大小。模型通过分解大卷积为小卷积以及非对称卷积来减少参数, 从而加速运算, 减轻过拟合, 并提升模型表达能力和特征多样性。前向传播过程中, 通过标准卷积和模块序列的卷积核以及最大池化操作来提取图像特征, 随后通过重复模块来加深模型, 细化特征。在模型尾部, 特征图尺寸减小, 通道数增加, 经卷积层、可分离卷积和全局平均池化转化为向量后进行分类。在分类过程中, 特征向量被送入一个或多个全连接层, 映射到目标类别。最后, 模型输出“好”与“坏”的概率分布。通过调整损失函数和优化参数反复迭代, 提升分类的准确性。

2.2 图像分割模型

图像分割模型通过 DeepLab v3+模型进行训练, 用于字体图像分割。它将输入图片的每个像素划分类别, 如将大小为[h, w, c]的图像输出为[h, w, 1], 每个像素值代表一个类别。该模型结合编码器–解码器和 ASPP, 前者捕捉更多的边界信息, 后者提取更多的特征信息。编码器基于 Xception_65 网络, 输

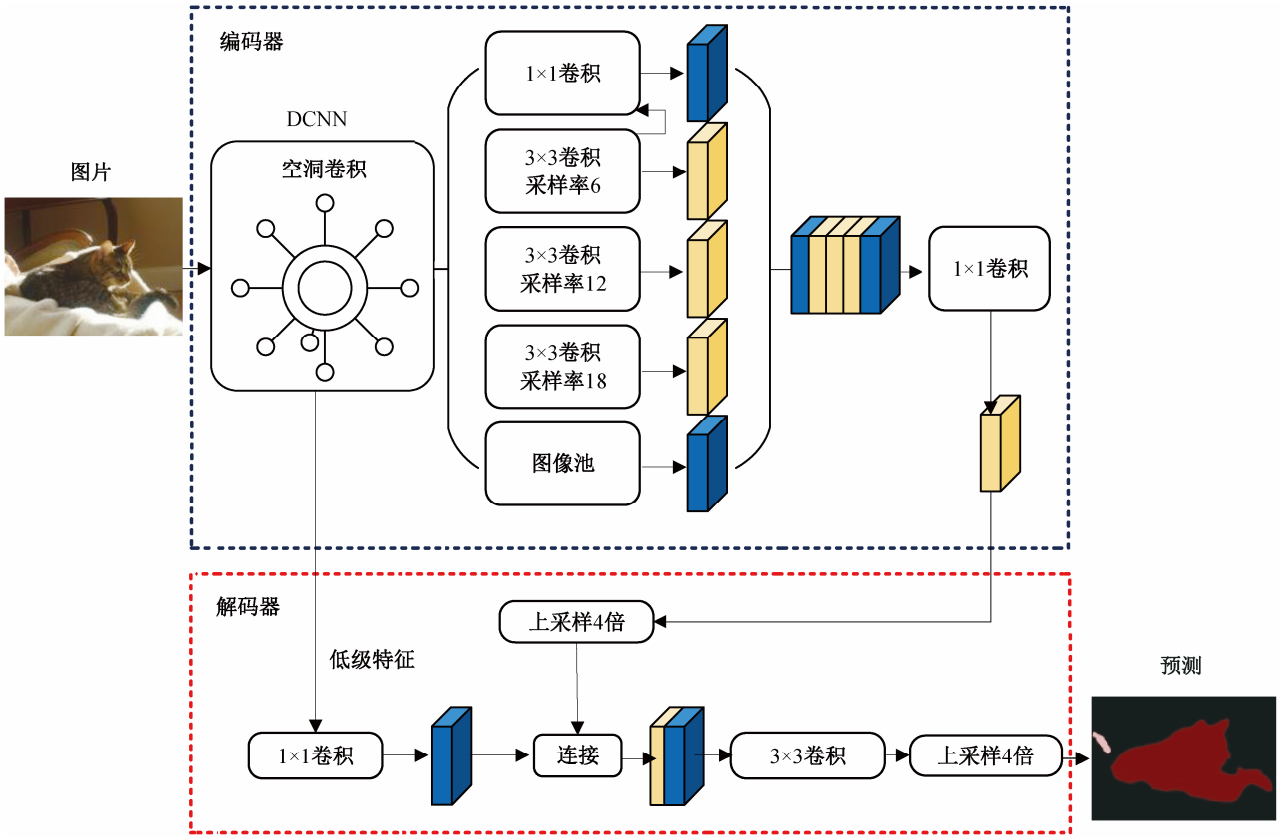

图 2 编码器–解码器网络结构[9]

Fig. 2 Encoder-decoder network structure[9]

表 1 Inception v3 的总体架构[22]

Table 1 Architecture of Inception v3 [22]

类型	匹配大小/步长或备注	输入
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
pool	8×8	8×8×2048
linear	logits	1×1×2048
softmax	classifier	1×1×1000

入数据依次通过进入流、中间流和退出流 3 个流程，最终得到结果。图像分割模型网络结构如图 3 所示。进入流包含两层卷积和 3 个块结构，每块包含 3 个 stride=2 的深度可分离卷积层，并且添加残差连接通道。先用 3×3 卷积进行通道上的分离计算，然后使用 1×1 卷积对特征进行合并，每一个块结构都添加 BN 层。中间流迭代 16 次，每一单元块结构卷积核是 728。退出流由带残差连接的卷积核(728 与 1024)和两个可分离卷积(1536 与 2048)组成。采用 Xception_65 的某一节点 A 来获取图像信息，再加入 ASPP 中。ASPP 将不同膨胀率的空洞卷积核全局池化和上采样后的输出连接在一起，作为编码器输出部分。解码器部分选取 Xception_65 中 A 节点之前的 B 节点，再将编码器的输出上采样成 B 的大小后连接，经过卷积和上采样，得到最终的分割结果。

2.3 字体生成模型

通过数据质量评估模型和图像分割模型，将用户输入的个性化字体图分割成 1777 个部件。基于楷体的部件映射关系，生成 6763 字所需部件。随后，字体生成模型通过对部件拼成后的字体进行笔画粗细调整、噪点去除、图片矢量化以及字体生成，实现用户个性化字体的定制。

2.3.1 笔画粗细调整

针对粗细不均的字体部件图像，本文提出一种

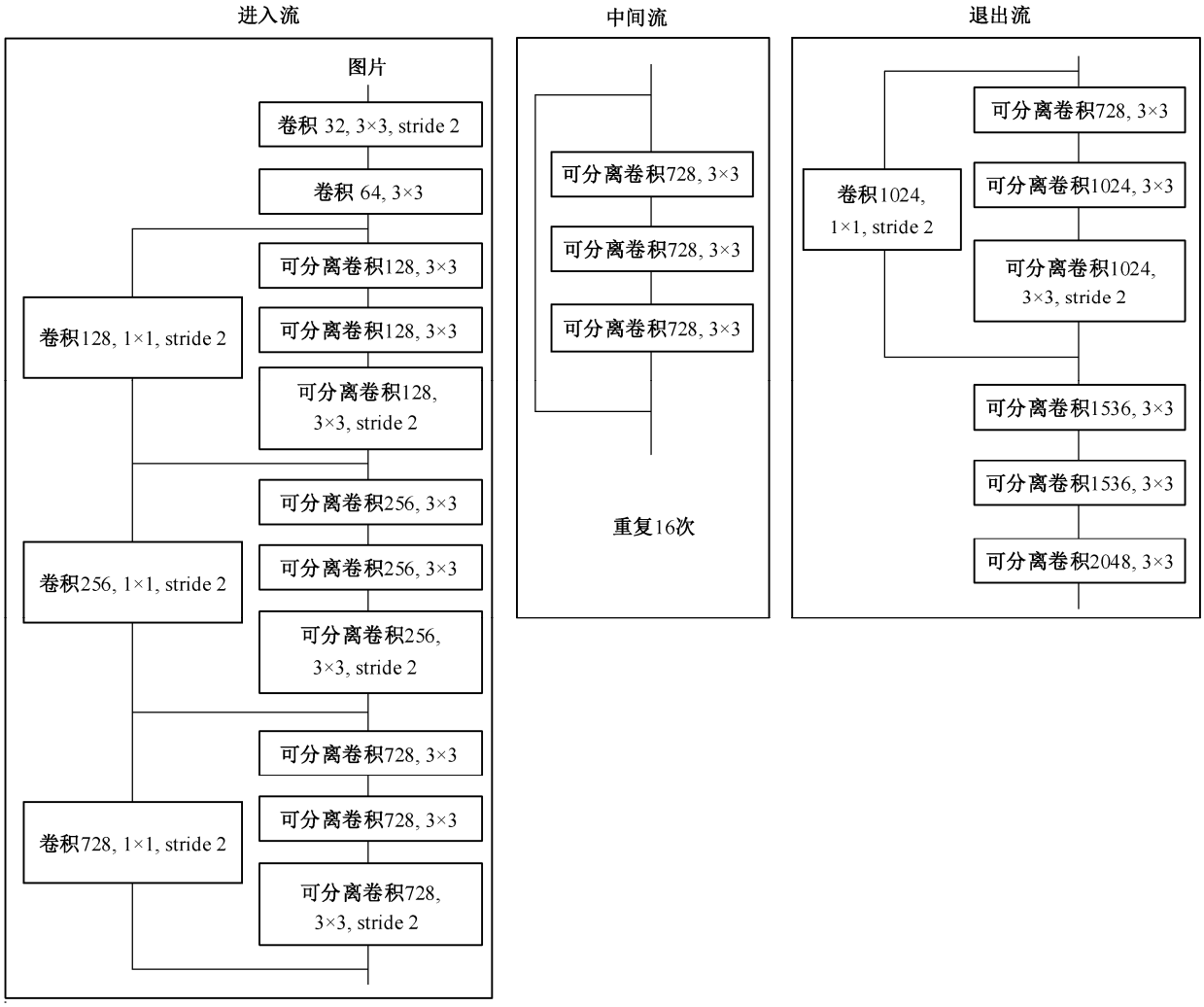

图 3　图像分割模型的组织结构[9]

Fig. 3　Organizational structure of image segmentation model[9]

基于模板比对的笔画粗细调整算法。首先加载模板数据，缩放其骨架像素，以便适应输入图像的大小，确定最小阈值。通过对比模板与输入图像，调整字体笔画的粗细。对于非符号字体，若其尺寸小于模板阈值，则放大至阈值，并中心对齐，根据笔画宽度变化，调整图像区域。笔画过宽则减少宽度，过细则增加宽度。对于符号字体，按最大尺寸等比例缩放，并类似地调整笔画粗细。图 4 展示字符“j”的调整效果。

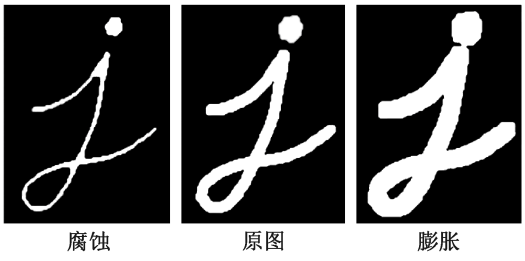

图 4　字符“j”的膨胀腐蚀效果

Fig. 4　Dilation and erosion effects of the character “j”

2.3.2 噪点去除

针对字体图像分割后的噪点问题，本文提出一种自动去噪方法。首先通过图像特征划分闭合区域，并定义区域的像素点数目为 P。设定阈值 V(本文取为 100，可调)，对区域进行分类。$P>V$ 时，为大区域，其余为小区域。对于大区域，噪点将通过大区域是否为边缘区域进行判定，当大区域内所有的像素点都不处于中心区域时，则当前大区域被判定为边缘区域，其中所有的像素点都被视为噪点；对于小区域，若像素点距离最近有效大轮廓超过阈

图 5 “临”字的“好”和“坏”输入集

Fig. 5 Input sets of “good” and “bad” examples for the character “临”

值, 则被视为噪点。通过此方法, 可以有效地去除字体图像中的噪点。

2.3.3 图片矢量化

标准的 TrueType 格式字体存储的是字形矢量数据, 去噪后的二值化图像需矢量化。首先对图像进行归一化处理, 将图片缩放到统一基数。缩放方式是智能缩放, 通过字体的大小设定固定的参数, 确保图片的缩放不会太大或太小。然后, 按照图像顺序进行矢量化。先提取图像轮廓, 通过删除代价算法找出关键点, 再通过骨架线曲线拟合, 得到 TrueType 轮廓, 最后将矢量化结果存在指定结构体中, 生成 TTF 所需数据。

2.3.4 字体生成

本文提出一种字体自动生成的方法。图片矢量化后生成文本数据, 通过读取矢量化文本数据, 创建 Glyf 等表, 按照对应表的结构赋值, 并记录表偏移量。按照偏移量赋值后, 依次写入所有表数据, 生成标准的 TrueType 字体。

3 数值实验

3.1 模型数据处理

3.1.1 数据质量评估模型数据处理

1) 数据处理。本文利用 TfClasifier 分类模型构建数据质量评估模型, 用于判别用户手写字体图像的质量。在训练前, 人工为用户手写字体图像标注“好”与“坏”标签。模型基于 1354 套字体中的 607 个字(每字 1354 张图片)进行训练, 最终选取 130 个字作为测试集。图 5 展示“临”的标签输入集。

2) 模型训练。数据标注后, 将利用数据质量评估模型对 130 字进行逐字训练, 实现对输入字体质量的好坏分类。经过加载标注数据、预训练模型、长时间训练和生成单字模型数据等流程, 最终为每个字生成单独质量评估模型。

3) 模型预测。加载训练好的数据质量评估模型, 对用户输入字体图像进行预处理, 并预测其质量的好坏。最终将预测结果结合人工决策来确定参数值。待预测图片总数为 A, 逐张读取图片列表进行预测, 并按预测值进行降序排列。预测结果记为 T, 若 T 为“好”则增加计数 G。R 为预测“好”字体的图片占比, 即 $R=G/A$。基于 254 套用户字的评分(包括字形结构和部件质量)如表 2 所示。如 $R\geqslant 0.4$, 则判定为“好”字体。由此筛选出的高质量字体可提高部件分割准确率。

3.1.2 图像分割模型数据处理

1) 数据处理。使用语义分割模型 DeepLab v3+ 进行图像分割。训练前需准备标签数据, 通过人工方式标注 775 字每个固定顺序的部件。标注完成后, 导出三类数据: 整字原图、部件的 8 位灰度图 1 (部件位置为 1 值, 背景为 0 值, 全黑效果)以及部件的 8 位灰度图 2 (部件位置值为 255, 背景为 0 值)。例如, “临”字导出 3 个部件的三类数据中, 部件“丨丨”的数据如图 6 所示。人工核验数据(主要核验部件 8 位灰度图 2)后, 将图片生成为 record 文件,

表 2 字体数据质量评分表

Table 2 Font data quality scoring table

字体名	R	字形结构	部件质量	结论
字体 A	0.9125	9	9	好
字体 B	0.8250	8	9	好
字体 C	0.7750	8	7	好
字体 D	0.5000	8	6	好
字体 E	0.4125	5	5	好
字体 F	0.4000	4	4	坏
字体 G	0.4000	7	7	好
字体 H	0.3500	5	4	坏
字体 I	0.2500	3	3	坏

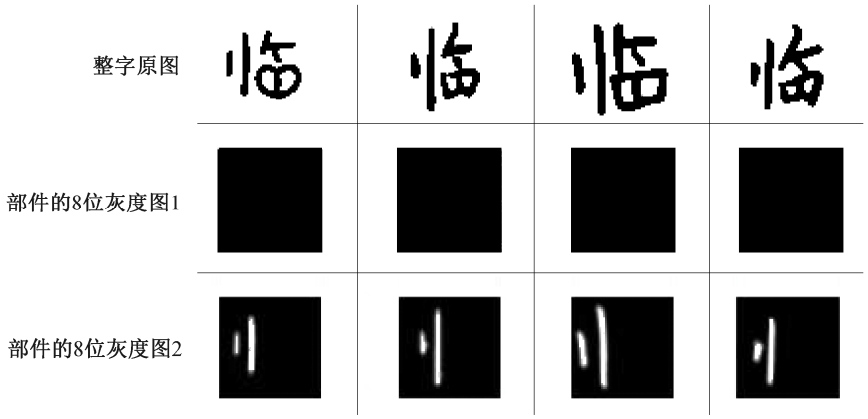

图 6　整字原图、部件的 8 位灰度图 1 和部件的 8 位灰度图 2

Fig. 6　Original full-character image, component 8-bit grayscale image 1 and component 8-bit grayscale image 2

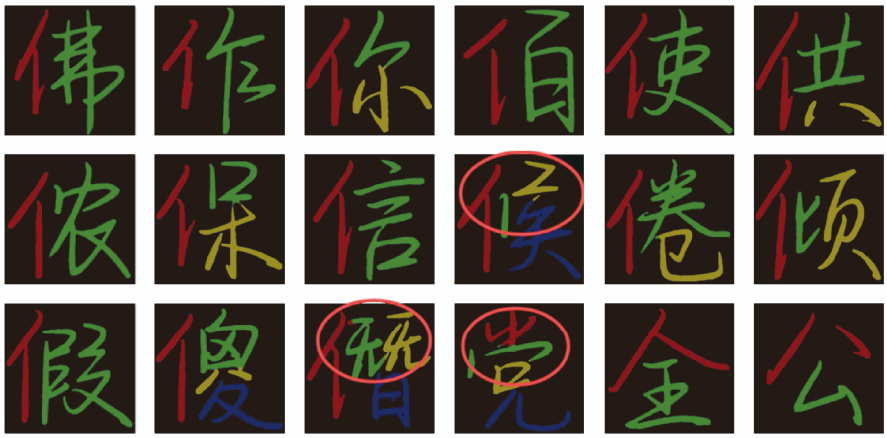

图 7　图像分割模型分割整体结果

Fig. 7　Segmentation result of the image segmentation model

方便模型读取。

2) 图像分割模型。对数据质量模型评估为“好”的字体进行图像分割。在分割造字中，将输入的775字拆分为部件，并为每个部件训练图像分割模型，部分字的分割结果如图7所示。经统计，775个字的分割结果中只有4个错误，准确度为99.5%。然而，当部件有重叠或者部件距离较近时，当前的图像分割算法仍然存在分割不准确的问题，如图8所示。

3) 图像分割模型优化。为了解决上述问题，本文提出两种方法，对图像分割模型进行优化。

方法1：数据增广，解决分割不精确、粘连拆分不准确等问题。在320套数据基础上，进行整字透视变换、整字缩放、整字旋转、部件位移以及笔画粗细调整，生成1000多套新数据。经过上述优化

图 8　图像分割模型拆分部件的出错示例

Fig. 8　Error examples of component segmentation by the image segmentation model

流程，部分组合如表3所示。通过实验比较，本文采用测试效果最好的数据增广组合，即“整字透视变换＋整字缩放＋整字旋转＋部件位移＋笔画粗细

表 3 数据增广过程

Table 3 Data augmentation process

已尝试组合	效果
变形: 300 图, 只训练经过变形的数据(在垂直和水平方向做三角变形); 原始图+变形: 600 图	预测结果整体有改善, 分割更精确, 部分图粘连拆分准确
原数据扩大5倍, 在透视、缩放、旋转、位移变换的基础上, 加上笔画宽度变换, 笔画宽度变换执行的概率设为 0.3	笔画宽度只做了一个像素的腐蚀以及两个像素的膨胀, 对结果影响不大
透视变换+原始图缩放2倍+随机位置+整字的旋转+部件位移、整字不位移(300 图)+变形(300 图): 2400 图, 16 个周期	模型精度比较高, 预测的效果有改善, 但不明显
透视变换+原始图缩放 3 倍+随机位置+部件位移: 2700 图	没有明显的改善和提升
整字透视变换+整字缩放+整字旋转+部件位移+笔画粗细(90 图), 循环 48 个周期, 去掉有问题图: 1800 图	模型精度非常高, 损失率在 0.6 左右, 整体效果比之前好

(90图), 循环48个周期, 去掉有问题图: 1800图"。

方法 2: 使用预训练模型。在模型数据增广的基础上, 使用预训练模型, 对所有部件进行重新训练, 结果如表4所示。可以看出, 优化后的模型表现更优, 能够显著地提升图像分割准确率, 因此本文最终采用该模型。

3.2 分割后数据处理

3.2.1 特殊字数据处理

在3.1节中, 图像分割模型将775字图片分割为1777个部件图。智能造字算法基于这些部件图生成6763个字的所有部件图, 通过调整笔画粗细, 使部件宽度接近用户书写字的平均宽度, 保持字体笔画宽度一致。随后, 部件移动拼字, 优化单个字的结构。拼字的过程中, 有的拼字效果不佳, 如"洪"字的第3个部件"八"、"虎"下面的"几"、"华"下面的"十"、"荒"下面的"川"以及"先"下面的部件等, 这些部件在各个字体中出现频率高, 共计280次, 将包含这些部件的字标记为特殊字, 经下述步骤后, 效果如图9所示。

表 4 优化后图像分割错误率对比

Table 4 Comparison of image segmentation error rates before and after model optimization

字体名	旧模型 错误总数	旧模型 错误率%	新模型 错误总数	新模型 错误率%
字体 A	2	0.13	1	0.07
字体 B	4	0.25	0	0
字体 C	15	0.94	0	0
字体 D	6	0.38	2	0.13
字体 E	35	2.18	14	0.88
字体 F	5	0.32	1	0.07
字体 G	22	1.37	6	0.38
字体 H	14	0.88	3	0.19
字体 I	2	0.13	3	0.19
字体 J	17	1.06	6	0.38
字体 K	14	0.88	2	0.13

针对特殊字的特定部件, 基于笔画端点或交叉点位置, 通过移动策略计算该部件和参考字其他部分的关键位置参数, 再按照目标字体各部件的相对位置确定特定部件的移动方案。针对其他部件, 移动的策略是根据参考字中各部件的相对位置, 计算目标字中各个部件的位置。具体方法如下: 已知参考字部件的重心(X_1, Y_1)、几何中心(X_0, Y_0)、宽高(W_1, H_1)以及目标字初始几何中心(M_0, N_0)和初始宽高(W_2, H_2), 计算目标字部件重心坐标(X_2, Y_2)。根据$(X_1-X_0)/W_1=(X_2-M_0)/W_2$, 得到$X_2$, 同理计算出$Y_2$, 从而确定部件在目标字体中的位置。

3.2.2 生成效果对比

TrueType字体是通过字体图像的噪点去除和矢量化后, 基于部件组合而成。采用整体效果评估、相似度量化评分和问卷调查来判断生成字体的正确率。

基于用户手写的775字, 通过不同的造字方式, 生成6763字的对比效果, 整体效果如图10所示。每款字包含用户输入字、GPU造字、风格学习造字和分割造字。结果显示, GPU造字整体效果最好, 风格学习造字的风格接近楷体且整齐, 而分割造字的结构欠佳。

相似度量化可以分为粗、细两类指标。粗指标将相似度分为很相似(1)、中等(0.5)和不相似(0)。细指标从粗指标中筛选出中等和不相似字体, 从字

图 9　特殊字优化前后效果对比

Fig. 9　Comparison of effects before and after optimization for special characters

形结构(40%)、笔画粗细(30%)、笔画形状(20%)和连笔(10%) 4 个维度进行评分。综合粗指标占比与细指标评分, 得出造字方法的相似度。本文对 201 套字体进行粗评, 并对其中 10 套进行细评, 结果如表 5 所示。

从表 5 可知相似度得分为(0.38×45+0.58 ×70+1.0×86)/(45+70+86)=71.5%, 表明图像分割造字与原字的相似度高。进一步对比 GPU 造字与风格学习造字的相似度评分, 发现 GPU 造字的相似度最高(75.8%), 风格学习造字的相似度较低(63.6%)。

针对图像分割生成的字体进行问卷调查, 要求识别 100 字中的 50 个图像分割字(图 11), 其中 10 位被调查者的挑选结果如表 6 所示。可以看出, 除专业字体设计师(B)外, 平均正确率仅为 45.8% (低于 60%), 表明图像分割造字与原字的相似度高, 难以区分。进一步对比 GPU 造字与风格学习造字的用户调查结果, 发现 GPU 造字相似度最高(平均正确率为 43.6%), 风格学习造字的相似度较低(平均正确率为 51.8%)。

基于整体效果评估、相似度量化评分和问卷调查正确率 3 个指标, 3 种造字方法的对比情况如表 7 所示。可以看出, 图像分割造字生成的字体与原字风格的相似度高于风格学习造字。两者均使用 CPU, 耗时 30~40 分钟, 成本相近。相比 GPU 造字和风格学习造字, 图像分割造字既可以提升风格相似度, 又能够降低成本。

4 结论与讨论

本文提出一种基于图像分割的智能造字技术。用户手写 775 字, 通过数据质量评估模型筛选, 利用图像分割模型对字体图像进行分割和后续处理(如笔画粗细调整、移动、去噪和矢量化等), 最终

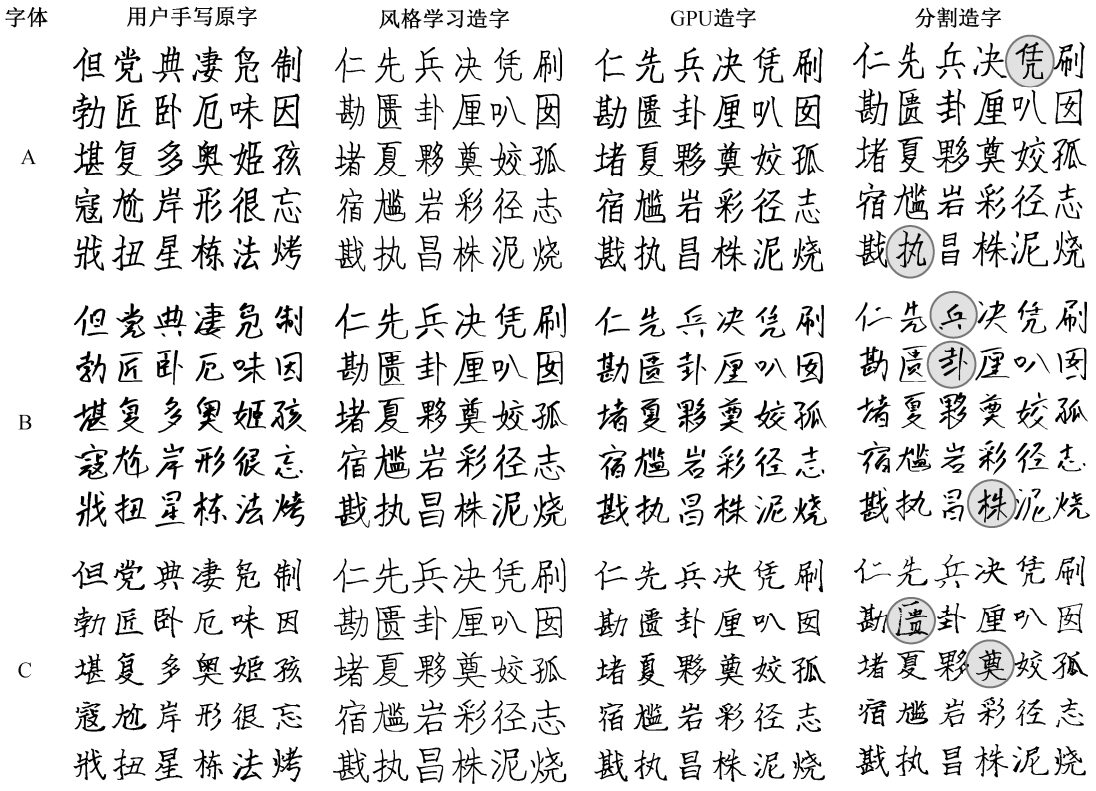

图 10 3 种字体的整体效果对比

Fig. 10 Comparison of effects among three font types

表 5 相似度评分表

Table 5 Similarity scoring table

粗指标评分	字体数	字体列表	细指标评分						相似度
			字形结构	笔画粗细	笔画形状	连笔	总计值	平均值	
0	45	字体 1	0	0.3	0.1	0	0.4	0.38	0.715
		字体 2	0	0.2	0.1	0	0.3		
		字体 3	0	0.3	0.1	0	0.4		
		字体 4	0	0.3	0.1	0	0.4		
		字体 5	0.2	0.1	0.1	0	0.4		
0.5	70	字体 a	0.2	0.3	0.1	0	0.6	0.58	
		字体 b	0.1	0.3	0.1	0	0.5		
		字体 c	0.3	0.3	0.1	0	0.7		
		字体 d	0.2	0.3	0.1	0	0.6		
		字体 e	0.1	0.3	0.1	0	0.5		
1	86							1.0	

生成 6763 字，形成标准的 Truetype 字体。利用数据质量评估模型和图像分割模型等进行建模，通过评估最后生成的字体效果，得到以下结论。

1) 数据质量评估模型预先分级筛选数据，由于用户手写字体质量参差不齐，既节省生成字体时间，提升用户体验，又可以增强系统的鲁棒性。

2) 预训练模型基于实际数据，可以提高预测准确性，实现图像分割全自动化且精准。通过人工挑选真实用户手写的 6763 字，进行数据增广，实现图像分割准确率的提升。

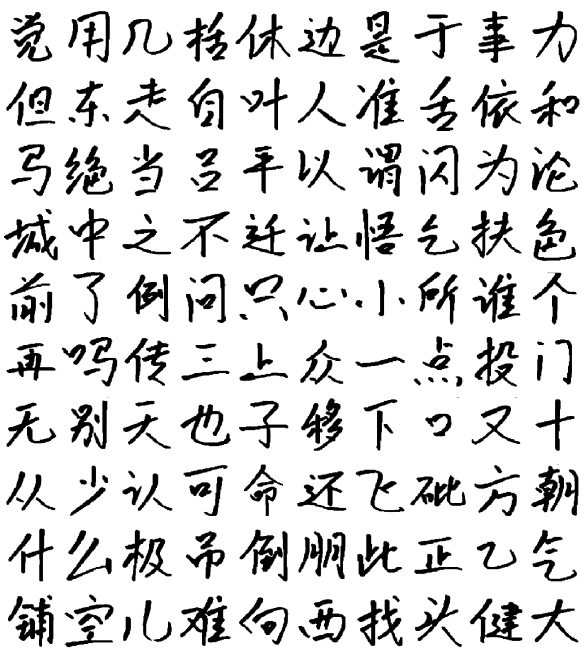

图 11　原字与图像分割生成字的随机分布

Fig. 11　Random distribution of original characters and characters generated by image segmentation

3) 图像分割技术生成字体的风格高度相似于原字体。与传统的风格学习造字方法相比, 相似度更高。两者均使用 CPU, 且时间均为 30~40 分钟, 成本接近。GPU 造字的生成时间为 4~6 小时, 因此本文方法的成本远低于 GPU 方法。

基于图像分割技术的智能造字研究取得一定的效果, 但仍然有以下不足。

1) 图像分割准确性需提升。受限于图像分割模型, 分割效果目前不能达到百分百准确, 模型需持续优化。

2) 拼字结构存在问题。部件移动只是进行线性处理, 对复杂字体结构的拼接不够精准。后续工作中可研究 GPU 造字结构生成方法, 或以其结构为参考, 优化部件移动。

3) 部件复用存在缺陷。对于不同的字体复合, 生成部件所需尺寸大小不一, 形状存在细微的区别, 导致不是每个部件都适合所有字体复合。后续工作

表 6　10 位被调查者挑选的正确数

Table 6　Number of correct picks by 10 survey respondents

被调查者	第 1 行	第 2 行	第 3 行	第 4 行	第 5 行	第 6 行	第 7 行	第 8 行	第 9 行	第 10 行	平均正确率/%
A	1	3	3	3	1	3	2	3	4	3	52
B	3	3	3	4	2	3	3	5	5	2	66
C	2	1	1	1	1	2	1	2	2	2	30
D	2	3	4	4	3	3	2	2	2	2	54
E	2	3	5	3	1	2	2	4	1	4	54
F	1	3	4	3	2	3	3	4	2	3	56
G	1	3	4	2	2	1	2	2	4	3	48
H	0	1	2	3	1	2	1	2	2	1	30
I	2	2	2	2	0	2	0	1	1	1	26
J	2	2	4	3	2	1	1	2	4	0	42

表 7　3 种造字方法的对比

Table 7　Comparison of three character creation methods

性能	GPU 造字	风格学习造字	分割造字
与输入字的风格相似度	最高	一般	高
生成字整体效果	最好	一般	一般
生成处理器	GPU	CPU	CPU
生成成本	高	低	低
生成时间	4~6 小时	30~40 分钟	30~40 分钟
存在问题	有噪点	风格不像原字更像楷体	分割有错误, 结构有问题, 部件不合适

中需微调部件, 以便适应不同字体的复合。

4) 用户手写字体数据有限, 模型初具分割能力, 还需大量用户手写验证, 优化模型。

参考文献

[1] 虞金星. 我们的字体还不够多[N]. 人民日报, 2012–06–04 (012)

[2] Xu S, Jin T, Jiang H, et al. Automatic generation of personal chinese handwriting by capturing the characteristics of personal handwriting // Twenty-First IAAI Conference. Pasadena, 2009: 191–196

[3] Tian Y. zi2zi: Master chinese calligraphy with conditional adversarial networks [R/OL]. (2017)[2024–02–03]. https://github.com/kaonashi-tyc/zi2zi/

[4] Lian Z, Zhao B, Chen X, et al. EasyFont: a style learning-based system to easily build your large-scale handwriting fonts. ACM Transactions on Graphics (TOG), 2018, 38(1): 1–18

[5] 江月, 连宙辉, 唐英敏, 等. 一种基于深度神经网络的手写体中文字库自动生成方法: CN107644006B [P]. 2020–04–03

[6] 潘婉琼, 连宙辉, 唐英敏, 等. 一种基于骨架指导的文字图像矢量化方法及系统: CN103942552B [P]. 2017–02–08

[7] 易天旸, 连宙辉, 唐英敏, 等. 一种个性化汉字数字墨水的生成方法: CN103488339A [P]. 2014–01–01

[8] 赵波, 连宙辉, 唐英敏, 等. 一种基于风格学习的汉字合成方法: CN106611172A [P]. 2017–05–03

[9] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation // Proceedings of the European Conference on Computer Vision (ECCV). Munich, 2018: 801–818

[10] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks // Proceedings of the IEEE International Conference on Computer Vision. Venice, 2017: 2223–2232

[11] Xie Y, Chen X, Sun L, et al. DG-font: deformable generative networks for unsupervised font generation // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, 2021: 5130–5140

[12] Jiang Y, Lian Z, Tang Y, et al. Scfont: structure-guided Chinese font generation via deep stacked networks // Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, 2019: 4015–4022

[13] Wang Y, Lian Z. DeepVecFont: synthesizing high-quality vector fonts via dual-modality learning [EB/OL]. (2021)[2024–02–03]. https://doi.org/10.48550/arXiv.2110.06688

[14] Wang Y, Wang Y, Yu L, et al. Deepvecfont-v2: exploiting transformers to synthesize vector fonts with higher quality // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver, 2023: 18320–18328

[15] Zhao Y, Lian Z. UDiffText: a unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models [EB/OL]. (2023) [2024–02–03]. https://doi.org/10.48550/arXiv.2312.04884

[16] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, 2015: 3431–3440

[17] Chollet F. Xception: deep learning with depthwise separable convolutions // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 1251–1258

[18] Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834–848

[19] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2015)[2024–02–03]. https://doi.org/10.48550/arXiv.1409.1556

[20] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 770–778

[21] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, 2015: 1–9

[22] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 2818–2826