后输入到回归层, 生成空间变换系数 θ, 使用定位网络 f_{loc} 来预测 2D 仿射变换矩阵 $A_\theta{}^n$.

$$f_{\text{loc}}(I)=A_\theta^n=\begin{bmatrix}\theta_1^n & \theta_2^n & \theta_3^n\\ \theta_4^n & \theta_5^n & \theta_6^n\end{bmatrix} \tag{2}$$

采样网络根据定位网络生成的变换系数 θ 构建一个采样网格 $G_i{}^n$, $G_i{}^n=(x_i{}^n,y_i{}^n)$, T_θ 为二维空间的变换函数, 输出特征图 V 上的坐标 $(x_i{}^n,y_i{}^n)$ 通过 T_θ 映射到输入特征图 I 上的坐标 $(u_i{}^n,v_i{}^n)$, 对应关系为:

$$\begin{pmatrix}u_i^n\\ v_i^n\end{pmatrix}=T_\theta\left(G_i^n\right)=A_\theta^n\begin{pmatrix}x_i^n\\ y_i^n\\ 1\end{pmatrix} \tag{3}$$

$$=\begin{bmatrix}\theta_1^n & \theta_2^n & \theta_3^n\\ \theta_4^n & \theta_5^n & \theta_6^n\end{bmatrix}\begin{pmatrix}x_i^n\\ y_i^n\\ 1\end{pmatrix} \tag{4}$$

可微图像采样根据上面的处理结果, 完成对输出特征图上每个坐标点的采样转换工作, 并且采用双线性插值的方式来表示:

$$V_i^n=\sum_h^H\sum_w^W I_{hw}^n\max\left(0,1-\left|u_i^n-h\right|\right)\max(0,1-|v_i^n-w|) \tag{5}$$

其中, $I_{hw}{}^n$ 是输入特征图 I 在通道 n 处 (h, w) 位置的坐标, 是输出的特征图 V_i^n 在 n 通道 (x_i^n, y_i^n) 位置处的坐标, n 代表特征图的通道数, H、W 分别代表输入特征图的高度和宽度. 通过上述的 3 部分组成的空间变换网络可以独立地插入到神经网络中, 并可以在网络中不断训练来修正参数完成对特征信息的仿射变换.

2.3 BGRU 网络

Hochreiter 等[15]出了一种循环神经网络最常见的变形——长短时记忆模型 LSTM. 循环神经网络会以不受控制的方式在每个单位步长内重写自己的记忆, 而 LSTM 有专门的学习机制能够在保持先前状态的情况下, 记忆当前时刻数据所输入的特征. LSTM 神经网络包含有 3 个门函数: 输入门、输出门和遗忘门.

LSTM 的改进版本 GRU[16]只包含两个门函数: 更新门和重置门. 更新门 z_t 表示过去时刻的状态记忆信息保存到当前时刻的程度, 更新门 z_t 的值越大, 过去时刻的状态记忆信息保存到当前时刻的信息就越多. GRU 的重置门 r_t 表示当前时刻忽略过去时刻的状态信息的程度, 重置门越小说明当前时刻保存信息越少, 对过去时刻忽略的信息就越多. 对 GRU 而言, 由于 GRU 参数更少, 不容易发生过拟合, 收敛速度更快, 因此其实际消耗时间要少很多, 这就大大加速了算法的迭代过程[16]. GRU 神经网络的传播公式如下所示:

$$Z_t=\sigma(w_2[h_{t-1},x_t]+b_z) \tag{6}$$

$$r_t=\sigma(W_r[h_{t-1},x_t]+b_r) \tag{7}$$

$$\varphi_t=\tanh(W_h[r_t*h_{t-1},x_t]+b_h \tag{8}$$

$$h_t=(1-z_t)*h_{t-1}+z_t*\varphi_t \tag{9}$$

其中, x_t 表示当前时刻的输入; h_t 表示上一时刻的输出; W_z、W_r、W_h 表示对应的权重矩阵; z_t、r_t 分别表示更新门和重置门; *表示矩阵元素相乘. 本文使用深度双向 GRU 神经网络模型如图 2 所示, 网络包含左右两个序列方向上下两层的 GRU 网络. 其中每层的 GRU 包含的隐含节点数目为 256 个, 文本特征序列 $x=\{x_1, x_2, \cdots, x_t\}$ 在该层进行正向和反向处理后输出中间序列 $m=\{m_1, m_2, \cdots, m_t\}$. 将中间序列作为第二层的输入进行正向和反向后输出向量 $y=\{y_1, y_2, \cdots, y_t\}$, y 包含了序列每一帧的预测概率值.

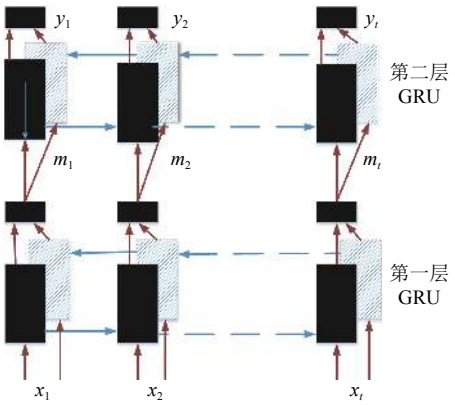

图 2 双向 GRU 网络

3 本文模型结构

自然场景图片背景复杂, 汉字种类繁多结构复杂, 一方面使得简单的卷积网络 (AlexNet[17]网络和 VGG[18]网络) 难以完全提取图像的底层特征细节, 另一方面随着网络复杂性的提高又导致网络出现过拟合、参数繁多、难收敛等现象. DenseNet 网络结构先提取图像底层特征, 并通过空间变换网络对中文字符在大小、宽高比、角度、倾斜等方面进行 2D 仿射变换, 从而提高文字的识别率. 循环神经网络层 BGRU 对输入的特征序列 x 进行标记预测, 输出序列 y 包含上下文信息可以得到距离较宽的文字进行更加精确的预测信息. 对

于包括含有模糊不清的并含有其它特殊文字, 特征序列包含的上下文信息也有更好的优化处理效果. CTC 层计算输入序列 $y=\{y_1,y_2,\cdots,y_t\}$ 对应的 (汉字、英文字母、数字和标点共 5990 个字符) 种标签元素序列的概率分布, 映射函数通过去除空格和去重操作后输出可能的序列 ι, 统计并计算每个标签序列的条件概率 $p(\iota/y)$, 求出标签序列 ι. CTC 输出的序列标签 ι 参照概率字典上的文字, 就可得到图片上的文字信息. 将得到的文本送入到敏感语义分类器当中进行分类.

在模型训练的过程中, 首先对图片进行归一化到相同尺寸且标签的长度保持一致, 本模型统一尺寸设置为 280×32, 标签长度设置为 20. 通过一个卷积核为 7×7, 卷积步长为 2 的卷积层和卷积核为 3×3, 卷积步长为 2 的最大池化层. DenseNet 包含了 3 个 DenseNet Block 模块, 每个模块包含了 16 个 dense layer 层, 生成率 (growth rate, 即加进每层的卷积核数设置) 为 12. 实验中使用的过渡层包括批归一化层、Relu 层和 1×1 的卷积层, 然后是 2×2 的池化层, 同时去掉 DenseNet 的全连接层直接连接到 STN 网络上. 实验中将 Transition Layer3 作为子卷积网络, 后面接入一个线性回归层和 Relu, 作为定位网络. 根据定位网络回归得到的变换系数 θ 进行仿射变换, 采样网格的生成和输出图片的采样, 完成对图片的矫正工作, 在此过程中特征图的尺度保持不变.

4 实验

4.1 实验数据集

实验数据集 CTW (Chinese Text in the Wild) 包含 32 285 张图像, 总共有 1018 402 个中文字符, 并包含平面文本、凸起文本、城市文本、农村文本、亮度文本、远处文本、遮挡文本, 数据集大小为 31 GB. 以 (8:1:1) 的比例将数据集分为训练集 (25 887 张图像, 812 872 个汉字), 测试集 (3269 张图像, 103 519 个汉字), 验证集 (3129 张图像, 103 519 个汉字). Caffe-OCR 中文合成数据集是人工生成的自然场景文本数据集, 利用中文语料库, 通过字体、大小、灰度、模糊、透视、拉伸等变化随机生成, 共 360 万张图片, 图 3 是部分人工合成图片的示例. 将该数据作为敏感文字图片训练集, 同时图像分辨率为 280×32.

鉴于含有敏感文字图片的特殊性, 在互联网网络平台只收集到 360 张含有敏感信息文字的图片, 同时制作 2000 含有少量字符的敏感文字图片作为敏感图片测试数据集. 数据利用中文语料库 (新闻和常见用语), 通过字体、大小、灰度、模糊、透视、拉伸等变化随机生成. 包含汉字、英文字母、数字和标点共 5990 个字符, 每个样本固定 20 个以下字符, 字符随机截取包含敏感词汇和非敏感词汇的句子. 图 4(a) 为未标注的图片, 图 4(b)、图 4(c) 标注图片.

图 3 人工合成的图片

(a) (b) (c)

图 4 含有敏感字符的图片

4.2 实验设置

实验基于 Pytorch 和 Keras 框架, 所有实验的训练和测试是在计算机配置为内存为 16 GB, 显卡 GPU 为 GTX TITAN X 的服务器上进行的.

本模型的输入尺寸为设置为 280×32, 采用随机梯度下降法 (SGD) 进行训练. 动量和权重衰减分别被设置为 0.9 和 2.5×10^{-4}, 首先对数据集 CTW 进行处理归一化处理, 学习率初始值为 10^{-4}, 学习率每隔 10 K 次迭代变为原来的 0.5 倍. 然后在 Caffe-ORC 中文合成数据集进行训练, 采用的是 ADADELTA 梯度下降优化算法, 该算法是对 Adagrad 的扩展, 方案对学习率进行自适应约束, 但是进行了计算上的简化. Adagrad 会累加之前所有的梯度平方, 而 Adadelta 只累加固定大小的项, 并且也不直接存储这些项, 仅仅是近似计算对应的平均值. 初始学习率设置为 0.01. 在模型微调的阶段, 没有设置特定的终止迭代次数, 保证对每一个模型结构进行充分训练, 直到各个模型最终收敛为止. 随后, 我们使用一个 10^{-4} 的权重衰减, 并使用高斯分布来初始化权重. 在数据集里, 我们在每一个卷积层 (除了第

一个) 后加上一个 Dropout 层, 并设随机丢弃率 (dropout rate) 为 0.2. 测试误差仅对一项任务进行一次评估. 为了验证本文模型的有效性, 在敏感图片测试数据集上设置了两种类型的对比: (1) VGG、ResNet、DenseNet 之间常用典型的卷积神经网络的对比. (2) 循环神经网络两个变种 GRU 与 LSTM 的对比. 具体包括以下的端到端文本识别模型实验: “DenseNet+CTC”、“DenseNet+STN+BGRU+CTC”、“DenseNet+BGRU+CTC”、“ResNet+CTC”、“+ResNet+BGRU+ CTC”、“VGG+CTC”、“VGG+BGRU+CTC”、“DenseNet+STN+LSTM+CTC”.

实验设置采用了控制变量法的准则, 包括对数据集的训练和测试, 尽可能地控制其他因素对实验的影响, 这些影响因素可能包括: 优化方法、机器配置、学习率和迭代次数等. 本文采用的识别算法评价标准为编辑距离 (Edit Distance)、单词识别准确率 (Words Recognition Accuracy). 编辑距离[19]指的是任意两组字符串 st1 和 st2, 由其中一组字符串转化为别一组字符串所需最少的编辑次数. 通常编辑距离越大, 也就说明两组字符串相似度越低, 编辑距离越小, 则说明相似度越高. 编辑距离相似度表示为:

$$\delta = 1 - e/\max\{\text{length}(\text{st1}), \text{length}(\text{st2})\} \quad (10)$$

单词识别准确率指的是正确识别序列的总数与标签序列总数的比值. 识别准确率表示为:

$$\partial = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \quad (11)$$

4.3 实验结果与分析

4 个卷积神经网络结构的模型在数据试集 CTW 上的文字识别准确率随迭代次数变化的曲线图如图 5 所示, 实验设置每训练迭代 5 k 周期测试一次. 通过实验曲线图中实验数据对比, 本文设置的模型网络 (DenseNet+STN+BGRU+CTC) 测试准确率要高于没有 STN 结构的网络, 随着测试次数的提升, 含有 STN 结构的网络收敛最快, 最终的准确率为 0.87, 比没有 STN 结构的网络更加稳定, 识别准确率更高.

4 个典型卷积神经网络结构的模型在数据集 CTW 上的编辑距离相似度随迭代次数变化的曲线图如图 6 所示, 设置每迭代 5 k 次测试一次, 开始训练时模型编辑距离相似度就迅速攀升, 并且很快达到平稳状态, 曲线更加的平滑, 相比之下, 无 STN 的结构的曲线在测试次数为 2–13 的阶段波动更大, 但也慢慢趋于稳定.

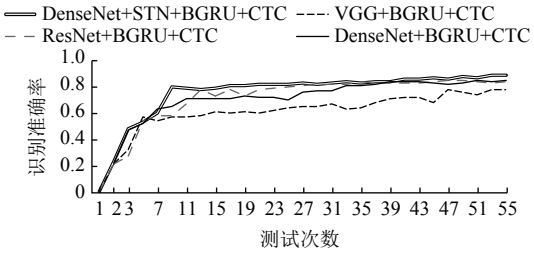

图 5 模型识别准确率变化曲线

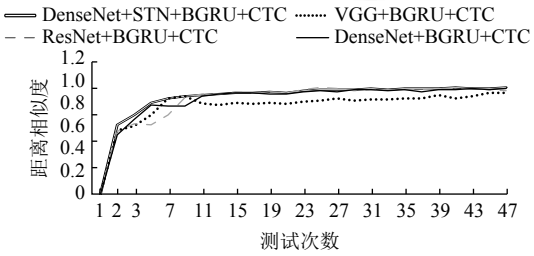

图 6 模型编辑距离相似度变化曲线

在 DenseNet 的输入特征图的数量并有效提高该网络的计算效率, 就引入一维卷积层到 DenseNet 网络中, 即 BN-ReLU-Conv1×1-BN-ReLU-Conv3×3 版本的合成函数 H_l, 也就是 DenseNet-B, 一维的卷积层能有效的减少实验中的输入特征. 在实验后续设置中, 为了进一步提高模型的紧凑性, 可以减小 Transition layers 层的特征映射数量. 当 Dense Block 模块包含了 m 个特征图时, 可以让 Transition layers 层生成不超过 θ_m 的最大整数个特征图, 其中 $0<\theta\leq 1$ 称为压缩因子. 当 $\theta=1$ 时, 转换过程中的特征图的数量保持不变. 我们称 DenseNet 当 $\theta<1$ 时为 DenseNet-C, 我们在实验中设定 $\theta=0.5$. 当同时使用了瓶颈层 (Bottleneck layers) 和压缩 (Compression) 的方法称为 DenseNet-BC[8]. 本文 DenseNet 网络设置为 DenseNet-BC.

通过分析表 1 使用的经典卷积网络与改进的 DenseNet-STN 网络以及 GRU 和 LSTM 的循环网络层对比. 相较于 VGG 和 ResNet, 具有一维卷积和压缩结构的 DenseNet-BC 结构具有更好的识别效果, 模型整体的文字识别准确率也就更高, 同时这表明在相同深度和宽度的神经网络框架下 DenseNet 可提高对特征信息的变现力.

在实验的过程中, 发现含有 DenseNet 特征网络的模型出现过拟合或者优化难等问题概率小于含有 VGG 网络和 ResNet 网络, 对参数的利用也更高效, 相同的数据集上, 可以得到更好的识别效果. 与此同时可以发

现, 基于 GRU 和 LSTM 网络的模型并没有在识别率上有直观的提升效果, 但是对比表 1 的对应实验模型的大小来看, 我们可以发现, 前者比后者有更小的内存容量, 由此说明在模型训练的过程占有更少的显存空间.

表 1 模型在敏感图片数据测试集的识别统计结果

试验模型	正确个数	准确率 (%)	总编辑距离
DenseNet+CTC	1827	77.4	3453
DenseNet+STN+BGRU+CTC	2131	90.3	2349
DenseNet+BGRU+CTC	2020	85.6	2765
ResNet+CTC	1761	74.6	6457
ResNet+BGRU+CTC	1987	84.2	4321
VGG+CTC	1331	56.4	9864
VGG+BGRU+CTC	1890	80.1	5218
DenseNet+STN+BLSTM+CTC	2119	89.8	2431

表 2 总结了各种模型组合在模型大小与运行时间方面的实验结果. 通过分析可以发现, 使用不同的特征提取层使网络模型在模型大小与平均识别时间上有着比较大的差距. 本文提出的 DenseNet-STN 特征提取网络结构在敏感图片测试集上的平均识别时间为 26.3 ms 每张图, 大致相当于 1 s 处理 38 张图片, 符合实际应用之中对敏感信息图片处理的要求. 通过比较发现, DenseNet-STN 网络结构处理图片用时最多, 其主要原因把时间用在特征提取和空间变换阶段. DenseNet 虽然具有较好的加强特征传递和提高特征利用率等优点, 但 DenseNet Block 内部联系紧密导致特征提取阶段相对 VGG 网络和 ResNet 网络的模型更加耗时, 占用更多的 GPU 显存, 而本实验中实验机器显卡内存只有 16 GB 基本够用. 在实际应用环境下对硬件平台有了更高的要求, 同时本模型对图片相对处理速度也符合要求.

表 2 模型在敏感图片数据测试集上的大小与时间效率的统计结果

试验模型	模型大小 (MB)	平均识别时间 (ms/图)
DenseNet+CTC	4.1	21.6
DenseNet+STN+BGRU+CTC	16.1	26.3
DenseNet+BGRU+CTC	12.4	24.2
ResNet+CTC	4.5	12.2
ResNet+BGRU+CTC	14.5	17.3
VGG+CTC	22.3	5.3
VGG+BGRU+CTC	32.1	7.8
DenseNet+STN+BLSTM+CTC	15.3	26.5

5 结束语

本文提出了一种网络敏感文字图片识别的新方法, DenseNet-STN 对复杂背景下的敏感文字图片进行特征信息提取和变换矫正, 相比于前人研究的 VGG 网络和 ResNet 网络, 实验表明本文模型能准确地识别被扭曲、3D 凹凸、艺术化、倾斜等复杂短文本, 模型的识别准确率、编辑距离相似度有着良好表现. 与此同时, 本文提出的方法在效率和算法还有很大的提高, 在缺乏足够样本下对文字图片进行分析理解仍旧是个难题.

参考文献

1 Yao C, Bai X, Shi BG, *et al*. Strokelets: A learned multi-scale representation for scene text recognition. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014. 4042–4049.

2 Bissacco A, Cummins M, Netzer Y, *et al*. PhotoOCR: Reading text in uncontrolled conditions. Proceedings of 2013 IEEE International Conference on Computer Vision. Sydney, Australia. 2013. 785–792.

3 Jaderberg M, Simonyan K, Vedaldi A, *et al*. Synthetic data and artificial neural networks for natural scene text recognition. arXiv: 1406.2227, 2014.

4 Goodfellow IJ, Bulatov Y, Ibarz J, *et al*. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv: 1312.6082, 2013.

5 吴财贵, 唐权华. 基于深度学习的图片敏感文字检测. 计算机工程与应用, 2015, 51(14): 203–206, 230. [doi: 10.3778/j.issn.1002-8331.1404-0236]

6 彭海. 基于异构计算的图片敏感文字检测系统[硕士学位论文]. 成都: 电子科技大学, 2018.

7 Graves A, Fernández S, Gomez F, *et al*. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. Proceedings of the 23rd International Conference on Machine Learning. Pittsburgh, PA, USA. 2006. 369–376.

8 Huang G, Liu Z, van der Maaten L, *et al*. Densely connected convolutional networks. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 2261–2269.

9 Srivastava RK, Greff K, Schmidhuber J. Training very deep networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal, Canada. 2015. 2377–2385.

10 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778.

11 Szegedy C, Liu W, Jia YQ, *et al*. Going deeper with convolutions. Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA. 2015. 1–9.
12 Szegedy C, Vanhoucke V, Ioffe S, *et al*. Rethinking the inception architecture for computer vision. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 2818–2826.
13 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning. Lille, France. 2015. 448–456.
14 Jaderberg M, Simonyan K, Zisserman A, *et al*. Spatial transformer networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal, Canada. 2015. 2017–2025.
15 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]
16 Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures. Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille, France. 2015. 2342–2350.
17 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, CA, USA. 2012. 1097–1105.
18 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.
19 刘宝龙. 基于图像分析和深度学习的船名标识字符检测与识别研究[博士学位论文]. 杭州: 浙江大学, 2018.