计算机科学
COMPUTER SCIENCE

面向汉字点选验证码的轻量级高效识别方法

金鑫豪，池凯凯

引用本文

金鑫豪，池凯凯. 面向汉字点选验证码的轻量级高效识别方法[J]. 计算机科学，2024，51(11A): 240100031-9.

JIN Xinhao, CHI Kaikai. Lightweight and Efficient Recognition Method for Chinese Character Click-based CAPTCHA [J]. Computer Science, 2024, 51(11A): 240100031-9.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240100031

面向汉字点选验证码的轻量级高效识别方法

金鑫豪　池凯凯
浙江工业大学计算机科学与技术学院　杭州 310013
(xinhaojin@qq.com)

摘　要　数字化浪潮下,企业日益依赖机器人流程自动化(Robot Process Automation,RPA)技术来降低成本、提高效率,以保持竞争力。但流程中部分环节面临汉字点选验证码识别的难题,限制了自动化水平的进一步提高。现有研究方案存在数据集制作难度大、模型泛化性能差、模型复杂度与性能之间不平衡等问题。为此,提出一种数据集制作成本低、模型泛化性能好且轻量化的汉字点选验证码识别方法。具体而言:首先采用经过针对性改进的 YOLOv8-n 显著轻量化汉字检测模型,然后对汉字图片进行分割、矫正等预处理操作,接着采用泛化性强的 PaddleOCR 模型进行汉字识别,降低了场景迁移的成本,并通过识别概率矩阵得到最佳匹配结果,进一步提高了准确率。此外,设计了一种半自动的汉字检测数据集构建流程并公开了数据集。该研究旨在推动汉字点选验证码的自动识别技术的发展,促进企业流程自动化水平的提升。

关键词:流程自动化;验证码识别;YOLOv8;PaddleOCR;轻量化
中图分类号　TP391

Lightweight and Efficient Recognition Method for Chinese Character Click-based CAPTCHA

JIN Xinhao and CHI Kaikai
School of Computer Science and Technology,Zhejiang University of Technology,Hangzhou 310013,China

Abstract　With the advent of digitalization,enterprises increasingly rely on robotic process automation technologies to reduce costs and improve efficiency,thus maintaining competitiveness. However,the automation level is hindered by the challenge of Chinese character click-based CAPTCHA recognition in certain process steps. Existing research on this problem faces difficulties in dataset creation,poor model generalization performance,and an imbalance between model complexity and performance. To address these issues,this paper proposes a low-cost dataset creation approach and a lightweight Chinese character click-based CAPTCHA recognition method with excellent generalization performance. Specifically,a significantly lightweight version of the YOLOv8-n model,tailored for Chinese cha-racter detection,is employed in this study. Subsequently,preprocessing operations such as segmentation and rectification are applied to the CAPTCHA images. The highly versatile PaddleOCR model is utilized for Chinese character recognition,reducing the cost of scene adaptation. Furthermore,the best matching result is obtained through the recognition probability matrix,further enhancing accuracy. Additionally,a semi-automatic Chinese character detection dataset construction process is designed and made publicly available. This research aims to promote the development of automated Chinese character click-based CAPTCHA recognition techniques,enhance the level of enterprise process automation.

Keywords　Process automation,Verification code recognition,YOLOv8,PaddleOCR,Lightweight

1　引言

随着数字化转型浪潮的兴起,企业面临的压力日益增大,需要提高效率、降低成本,以保持竞争力。传统的人工操作效率低下,成本高昂,已经无法满足企业的需要。与此同时,人工智能技术快速发展,为流程自动化提供了技术基础。机器人流程自动化(RPA)[1]技术应运而生,成为企业数字化转型的重要工具,它可以模拟人类的操作,自动完成重复性、规则性的任务。这可以极大地提高效率,降低成本,解放人力,使企业能够专注于更具战略性和创造性的工作,在各个领域都有广泛应用前景,例如金融[2]、审计[3]、财务[4]、银行[5]、电网[6]、医疗[7]等领域。

在常见的用户自动登录、财务领域发票自动查验等操作的 RPA 实现中,经常需要验证码自动点选。然而,验证码的自动点选仍然难以实现,正如 Sganderla 等[8]指出,在某些含有高级验证码的网站上,自动化操作面临很大的挑战。汉字点选验证码是一种对人类用户友好但对机器而言很难识别的高级验证码,在中文网站广泛流行。也正因为如此,汉字点选验证码往往是许多自动化流程中的一个效率瓶颈,阻碍了流程自动化水平的进一步提高。

传统的 RPA 技术只适合完成简单重复的规则性任务,并没有被赋予智能,而未来的发展方向是走向智能流程自动化(Intelligent Process Automation,IPA)[9-10]。解决汉字点选验证码自动识别难题,就是迈向智能流程自动化的其中一步,可

基金项目:国家自然科学基金面上项目(62272414)
This work was supported by the General Program of the National Natural Science Foundation of China(62272414).
通信作者:池凯凯(kkchi@zjut.edu.cn)

以实现更复杂流程的自动化，最大限度减少人工干预，提高工作效率和生产力，同时也能为验证码的设计和安全性提供参考。这并不是一种恶意的黑客技术，相反，如果能够合理利用，对正规流程的自动化水平提高将起着重大作用。而且，大多数网站有足够的安全冗余设计来防止恶意攻击，如 IP 访问控制、用户行为分析、后台安全监控等，因此即使使用者有恶意动机，单纯的验证码自动识别技术造成的负面影响也非常有限。

汉字点选验证码主要提供商是网易易盾和极验，其风格相似，且识别难度较高，具有代表性。对于这类验证码，识别方法是相同的，总体上可以分为两步：汉字检测和汉字识别。已有的解决方案采用传统的机器视觉算法或较过时的目标检测算法来检测汉字，采用人工设计的 CNN 来识别汉字存在不少问题。

1）数据集制作代价大。传统方法识别汉字需要制作汉字数据集经过 CNN 训练，制作数据集需要花费大量人力来进行手动分类。

2）模型泛化能力差。自制汉字识别数据集的汉字风格相对单一，且数量有限，只能应对单一风格的汉字验证码。而如今汉字点选验证码的风格与以往不同且还在不断迭代。模型泛化能力有限，场景迁移需要重新制作数据集，代价高。

3）模型复杂度与性能之间不平衡。已有的方案采用 Faster-RCNN 来做汉字检测，这是一个较复杂和重量级的模型，存在推理成本高、检测代价和性能不平衡的问题。但是，在一些 RPA 应用场景中有必要设计和使用轻量化模型。例如财务自动化场景中实际使用的办公计算机可能是老旧的性能较差的机器，需要更高效的模型以保障流畅运行。此外，如果把模型运行在云端服务器对外提供接口，那么在多线程和并发环境下，轻量化模型在节省计算资源、减少内存占用、缩短推理时间等方面的优势就会被显著扩大。

针对上述问题，本文提出了一种轻量化的汉字点选验证码方法，在实现高识别精度的同时，模型复杂度大幅降低，泛化性提高，数据集制作成本降低。本文的具体贡献如下：

1）提出了一种基于改进 YOLOv8 的轻量化汉字检测方法，用于检测验证码图片中的汉字的位置。针对数据集目标较小和尺寸相对单一的特点，在 YOLOv8 基础上进行针对性改进，模型参数量、计算量、权重文件和推理时间大幅减少，检测性能几乎不降低。

2）提出了一种基于倾斜矫正和概率矩阵的汉字识别方法。首先使用 GrabCut[11] 算法实现汉字前后景分割。通过二值化、轮廓汇总和最小外接矩形框计算汉字倾斜角度，进而矫正倾斜。接着利用 PaddleOCR[12] 模型生成候选汉字（提示文本）概率矩阵。最后通过迭代删除最大概率值和对应的行列，实现汉字矩形框和候选汉字最佳匹配。

3）提出了一种半自动制作和扩充数据集的方法并公开了数据集。利用少量标签训练出基础模型，使用模型预测结果作为新数据标签，人工检查微调，不断迭代，可以快速扩充数据集。相比人工逐个标注，可以极大地节省工作量，提高效率。

2 相关工作

验证码是为了防止机器人攻击而设计的一种人机识别技术。绝大多数验证码都是基于文字嵌入的图像验证码，如英文、数字、汉字验证码。已经有很多研究者探索出了多种针对不同类型验证码的自动识别方法。

英文和数字验证码是应用最广泛的一种验证码。在英文和数字验证码识别方面，许多研究者做出了贡献。他们使用了多种机器学习算法[13]，如神经网络、k-最近邻、支持向量机和决策树，并比较它们的识别准确率和速度。一些研究者采用了多任务学习 CNN[14]、生成式对抗性网络[15] 等模型来处理复杂验证码图像，以提高识别效果。其中，多任务学习 CNN 方法在字符识别上的表现优于支持向量机，基于生成对抗网络的方法被用于解决遗传文本验证码问题，相较于传统方法更高效。另外，一些研究者提出了简单且高效的 CNN 模型[16]，无需分割验证码即可进行识别。

与英文和数字不同，汉字验证码数量庞大、结构复杂，识别难度大。常规的汉字验证码如图 1 所示，汉字位置相对固定，背景相对单一，字体工整且干扰较少，不少研究者在这方面做出了贡献。Wu 等[17] 提出了 MGLCR 和 CCR 两种方法，分别基于多尺度 Gabor 滤波[18] 和逻辑回归，以及卷积神经网络。这些方法在攻击带有噪声的汉字验证码方面表现出了良好的效果。另外，Wang 等[19] 提出了改进的 DenseNet 模型，该模型在中英文数字验证码识别任务中表现出色。而 Zhang 等[20] 提出了名为 Captchanet 的卷积神经网络，通过 3 种改进的训练策略和自定义的汉字训练集，提高了验证码识别的性能。此外，Wang 等[19] 也参考 LeNet-5 提出了一种卷积神经网络模型，通过增加卷积核的数量和 dropout 层来有效提取特征并达到高识别率。

图 1 汉字验证码

Fig. 1 Chinese character captcha

另一种更具有挑战性的汉字验证码是基于鼠标点击的汉字点选验证码，通常伴随着文字旋转、扭曲、复杂背景和随机位置，识别难度进一步加大。如图 2 所示，要求按顺序点击汉字位置，样例中需要依次点击“蔬菜汤”3 个字，这对验证码识别技术提出了更高的要求。目前也有一些研究者做过相关的工作。Bi 等[21] 提出了一种基于卷积神经网络的汉字验证码顺序选择系统，将任务分为几个子过程：基于 Faster R-CNN[22] 的汉字检测、汉字识别和基于 N-Gram[23] 的词序恢复。在汉字识别过程中，构建了一个定制的数据集，单字识别准确率达到了 98.43%。类似地，Hu[24] 使用传统图像处理方法和深度学习定位方法两种不同的方式对汉字点选验证码进行预处理和分割操作，然后将获得的单个字符或字符块放入训练好的识别网络中进行识别，也取得了较好的效果；You[25] 也提出了一种结合 YOLO 系列的目标检测算法加上深度卷积神经网络来识别点选汉字验证码的方法，并取得了不错的效果。但这些方法仍存在较大的改进空间，例如汉字检测模型没有针对小尺寸小目标做轻量化设计，模型存在较大冗余；汉字识别需要针对特定场景定制汉字数据集，代价较高且泛化性较差。

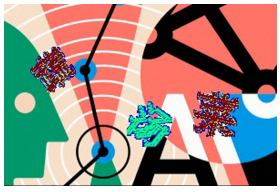

图 2　点选汉字验证码

Fig. 2　Chinese character selection captcha

3　数据集

3.1　自动爬取验证码图片

对于点选验证码图片的获取，由于验证码图片的风格多种多样且不存在相关的公开数据集，因此需要使用自动化程序来爬取真实的验证码场景中的图片。为此，本文可以使用两个工具(Selenium 和 Urllib)来自动化爬取验证码图片。

Selenium 是一种开源的 Web 自动化工具，用于模拟用户的行为，自动执行一系列任务，例如模拟鼠标、键盘操作、自动填写表单、检查网页元素、抓取网页内容等。它可以用于主流的浏览器，如 Chrome，Edge，Firefox，IE，Safari 等，非常灵活实用。

Urllib 是 Python 标准库中的一个模块，它提供了一系列的函数来操作 URL。它不仅可以用于发送 HTTP 请求、接收和解析响应，还可以进行 URL 的操作，例如解析 URL、拆分 URL、构建 URL 等等。此外，Urllib 还可以用于文件的下载、上传等操作。

通过这两个工具，可以有效地自动化获取验证码图片，从而大幅降低数据集源图片的获取成本。爬取点选验证码的具体步骤如下：

1)打开包含点选验证码的网页，并使用浏览器的开发者工具或查看源代码，找到验证码图片元素和刷新按钮在 HTML 树中的位置。

2)使用 Selenium 或其他自动化工具，根据元素的位置定位到验证码图片元素和刷新按钮的对象。

3)通过获取验证码图片元素的 Src 属性，得到图片的 URL 地址。

4)使用 Urllib 或其他 HTTP 库，根据得到的 URL 链接下载验证码图片到本地，并以一定的规则进行命名，例如使用 5 位数的自增序号，方便统计整理。

5)使用 Selenium，操作刷新按钮对象，模拟点击，实现验证码图片的刷新。

6)重复步骤 3)－步骤 5)，直到获得足够数量的验证码图片。

制作目标检测数据集时，通常使用标注工具，如 LabelImg，LabelMe 等。标注工具可以标注图像中的目标，一般使用矩形框来标注目标的位置，同时添加类别等信息。标注完成后，可以将标注结果保存为指定格式，例如 VOC 格式，或保存为 XML 文件，以便在训练模型时使用。

3.2　半自动化标注数据集

然而，标注目标是一项费时的任务。为了加快这一过程，可以采用半自动化标注方法，即手动标注一小部分数据，其余数据使用自动化程序辅助标注。具体而言，可以按以下步骤进行：

1)首先，手动标注一小部分数据，例如 100 张图片；

2)将这 100 张图片和标注文件放入模型训练中，训练模型，得到初步成果；

3)使用该训练模型对未标注的数据进行推理，例如 400 张图片，将推理结果按照 VOC 格式整理成 XML 文件，这样就得到了一个相对粗略的标注结果；

4)使用标注工具，如 LabelMe，打开标注文件，检查标注结果，删除错误标注结果，微调不精确的标注结果，新增漏标的目标，这样比手动标注 400 张图片容易得多；

5)将已标注的 500 张图片放入模型，继续训练模型；

6)使用新训练的模型对剩余的数据进行推理，例如 2 500 张图片，再次检查标注结果，只需微调即可。

通过以上步骤，本来需要手动标注 3 000 张图片，现在只需要精确标注 100 张，修改 400 张，检查 2 500 张，进行部分微调即可，大大减少了工作量，使扩大数据集变得更加简单。

本研究公开了汉字检测数据集，旨在为相关研究的进一步发展做出贡献[1)]。

4　方法

4.1　汉字检测

4.1.1　基础模型选取

YOLO[26]系列模型是一种基于深度学习的目标检测算法，它的优势主要有以下几点：1)速度快。采用了统一的回归网络，不需要复杂的框架和多次推理，可以实现实时的目标检测。2)精度高。YOLO 系列模型基于整张图片进行预测，而不是基于局部区域，可以利用全局信息提高检测精度，YOLOv5 在绝大多数场景下的精度已经超过了 Faster R-CNN 和 SSD[27]等算法。3)通用性好。YOLO 系列模型学习到的图片特征更为通用，可以适应不同的场景和任务，还支持任意大小的输入，能很好处理不同尺度目标，且一次训练即完成，收敛速度更快。4)模型结构相对简单。YOLO 系列模型没有使用复杂的骨干网络和注意力机制，而是使用了简单的卷积层和池化层，降低了模型的参数量和计算量。

YOLO 系列目标检测模型在公开数据集 COCO2017 上的表现如表 1[28]所列。由于部分模型在不断迭代更新，因此本文比较的是各个模型的最新公开版本，如 YOLOv5(7.0)，与最初的 YOLOv5 相比已经有了极大的提升。

表 1　YOLO 系列模型在 COCO2017 数据集上的表现[28]

Table 1　Performance of YOLO series models on COCO 2017 dataset[28]

Model	Input Size	mAP50:95	Params/M	FLOPs/G
YOLOv5(7.0)-n	640	28.0	1.90	4.50
YOLOv5(7.0)-s	640	37.4	7.20	16.50
YOLOv5(7.0)-m	640	45.4	21.20	49.00
YOLOv6(2.0)-n	640	37.5	4.70	11.40
YOLOv6(2.0)-s	640	45.0	18.50	45.30
YOLOv6(2.0)-m	640	50.0	34.90	85.80
YOLOv7-t	416	35.2	6.20	5.80
YOLOv7	640	51.2	36.90	104.70
YOLOv8-n	640	37.3	3.20	8.70
YOLOv8-s	640	44.9	11.20	28.60
YOLOv8-m	640	**50.2**	25.90	78.90
YOLOX(0.3.0)-n	416	25.8	0.91	1.08
YOLOX(0.3.0)-t	416	32.8	5.06	6.45
YOLOX(0.3.0)-s	640	40.5	9.00	26.80
YOLOX(0.3.0)-m	640	46.9	25.30	73.80
PPYOLOE+-s	640	43.7	7.93	17.36
PPYOLOE+-m	640	49.8	23.43	49.91

1) https://github.com/xinhaojin/click-based-captcha-dataset

本研究期望找到 YOLO 系列模型复杂度和性能的平衡点，因此筛除了较复杂的模型，选出计算量小于 30 GFLOPs 的 10 个相对轻量化的模型，测试其在自定义汉字检测数据集上表现，如表 2 所列。

表 2 在自定义数据集上的表现

Table 2 Performance on custom dataset

Model	Model Size/MB	Params/M	FLOPs/G	mAP50:95
YOLOv5(7.0)-n	3.80	1.76	4.20	83.60
YOLOv5(7.0)-s	13.70	7.02	15.80	83.30
YOLOv6(2.0)-n	9.94	4.63	11.37	81.10
YOLOv7-t	12.30	6.00	13.10	80.10
YOLOv8-n	**6.20**	**3.00**	**8.10**	**84.70**
YOLOv8-s	85.40	11.13	28.50	85.30
YOLOX(0.3.0)-n	7.24	0.90	1.08	75.12
YOLOX(0.3.0)-t	38.60	5.03	6.44	75.31
YOLOX(0.3.0)-s	68.50	8.94	26.76	75.84
PPYOLOE+-s	29.10	7.61	8.08	77.15

由表 2 可知，对于汉字检测数据集，最新的 YOLOv8 的 mAP 表现最佳，明显优于 YOLOX[29]（0.3.0）和 PPYOLOE+[30]，优于 YOLOv6[31]（2.0），略优于 YOLOv5（7.0）。而 YOLOv8-s 相比 YOLOv8-n，mAP 提升不明显，参数量和计算量却增加几倍，因此综合考虑采用 YOLOv8-n 作为基础模型做进一步改进。

YOLOv8 是 Ultralytics 公司在 2023 年 1 月 10 日开源的 YOLOv5 的另一个重大更新版本，目前支持图像分类、物体检测和实例分割任务。它建立在以前 YOLO 版本的成功基础上，并引入了新的功能和改进，以进一步提升性能和灵活性。YOLOv8 支持全面的视觉 AI 任务，包括检测、分割、姿态估计、跟踪和分类。YOLOv8 采用了一个新的骨干网络、一个新的 Ancher-Free 检测头和一个新的损失函数，还提供了 N/S/M/L/X 尺度的不同大小模型，用于满足不同场景需求。YOLOv8-n 是 YOLOv8 中最轻量的版本。然而针对本研究的场景，仍然存在进一步轻量化的空间。

4.1.2 针对汉字检测改进的 YOLOv8-n 模型

1）去除多尺度冗余

YOLOv8 原生模型采用的多层检测策略设计包含小、中、大 3 个目标尺寸对应的检测层。这有利于覆盖不同尺寸目标的检测。而在本研究的数据集中，经统计，20 000 多个目标的平均宽和高分别为 36.8 和 35.6 像素，标准差为 8.4 和 8.3 像素，这意味着目标尺寸接近，尺度相对单一。因此检测层的多尺度的设计就显得有些冗余了，同时，检测层的参数量较大。本研究简化了多尺度检测层，如图 3 所示，只有一个 Detect 层输出，显著简化了网络复杂程度和参数量。

2）降低下采样倍率

在 YOLOv8 原始模型的 Backbone 模块中，通过 5 个层级的下采样运算，最终实现了 32 倍的特征图下采样。这有利于提取深层次特征并利用更广阔的感知范围。

然而，对于本研究的数据集而言，目标尺度较小，过于浓缩的特征坐标会失去原目标结构信息，同时过大的感知区域在此情况下也难以发挥作用。另一方面，小、中、大检测层分别连接在 8 倍、16 倍、32 倍下采样的特征图后，由于去除了大检测层分支，最后一个 32 倍下采样后的特征层已经没有利用价值。基于以上两个方面的考量，本研究对原有的下采样模型进行了调整，将下采样倍率从 32 倍降低到了 16 倍。这一改动旨在使特征图上保留更多的原目标结构细节，同时使感受野范围适应数据集目标规模特征。如图 3 所示，直到 Backbone 末尾 SPPF 模块，一共经过了 4 次 stride 为 2 的下采样。

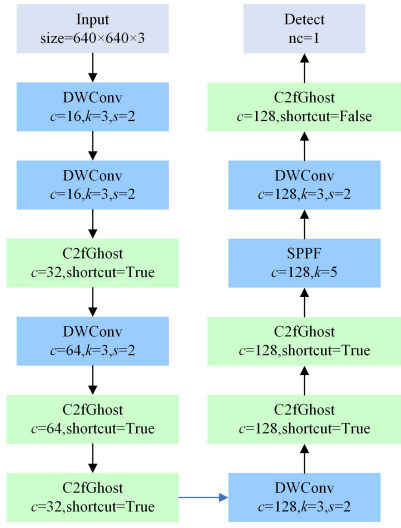

图 3 改进的 YOLOv8-n 结构图

Fig. 3 Structure diagram of the improved YOLOv8-n

3）引入 DWConv

DWConv[32]，全称 Depthwise Separable Convolution（深度可分离卷积），是一种卷积神经网络的基本组件，由深度卷积（Depthwise Convolution）和逐点卷积（Pointwise Convolution）两部分组成。深度卷积指对于每个输入通道，使用一个单独的卷积核进行卷积操作，即在空间维度上对每个通道进行卷积，而不是在通道维度上进行卷积。这样可以减少卷积运算的计算量和参数量，并且在一定程度上保持了特征图的空间信息。逐点卷积指使用 1×1 的卷积核对深度卷积的输出进行卷积操作。逐点卷积可以增加特征图的通道数，以便后续的网络层能够更好地提取特征。DWConv 最早在 MobileNet[33] 系列网络中被引入，并取得了良好的效果。目前，在许多轻量化网络中，如 ShuffleNet[34]，EfficientNet[35] 等，都使用了 DWConv 作为基本组件。本研究使用 DWConv 替换了普通 Conv，如图 3 所示。

4）引入 C2fGhost

C2fGhost[36] 模块是在 C2f 模块基础上进行的优化。它主要采用了 GhostBottleneck 模块，该模块包括 4 个卷积层：依次为 1×1 卷积层、3×3 深度可分离卷积层、Ghost 模块和 1×1 卷积层。Ghost 模块是通过参数共享的方式减少计算量和参数量的一种优化方法。在 GhostBottleneck 模块中，Ghost 模块用于对 3×3 深度可分离卷积层的输出进行特征扩张。此外，GhostBottleneck 模块还采用了深度可分离卷积层和残差连接来提高模型的效率和性能。总之，GhostBottleneck 模块适用于构建轻量级深度神经网络，如 MobileNetV3 和 EfficientNet-Lite 等模型。本研究考虑到了 C2fGhost 的这一优势，使用 C2fGhost 替换了普通的 C2f 模块，如图 3 所示。

4.2 汉字识别

4.2.1 汉字分割

在汉字点选验证码的识别中，通常需要将验证码中的汉字与背景分离，以便进行后续的文字识别或者图像处理操作。常用的汉字分割方法有基于颜色分割、基于形态学操作分割、基于图像分割算法分割等等。而本文采用的是基于目标检测模型预测的汉字位置矩形框，并通过 GrabCut 算法对矩形框中的汉字进行前后景分割。

GrabCut 是一种基于图像分割的算法，同时也是一种迭代式的前后景分割方法。其基本思想是根据用户指定的前景和背景的样本，利用高斯混合模型（Gaussian Mixture Model，GMM[37]）进行分割。本文使用 OpenCV 中的 GrabCut 算法实现汉字点选验证码的分割。下面将详细介绍基于 GrabCut 算法的汉字点选验证码分割方法。

1）汉字位置矩形框预测

首先，需要通过改进的 YOLOv8-n 模型对验证码图片中的汉字位置进行预测，得到每个汉字的位置矩形框，如图 4 所示，以极验汉字点选验证码为例。

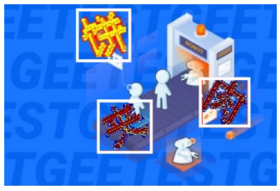

图 4　汉字检测结果

Fig. 4　Chinese character detection results

2）汉字位置矩形框扩展

在进行汉字点选验证码的前后景分割时，需要保证汉字及其周围的背景都被正确地标注为前景或背景。如果仅仅使用汉字的位置矩形框进行分割，有可能会将部分背景像素标注为汉字的前景，或者将部分汉字像素标注为背景，从而影响分割的准确性。因此，需要对汉字位置矩形框进行一定的扩展，以便将汉字周围的背景也包含进来。这样，在分割时可以更加准确地将汉字像素和背景像素分别标注为前景和背景。这里使用 0.1 作为扩展比例进行扩展，即在矩形框的基础上，每个边向外扩展 0.1 倍矩形框的宽度和高度。

3）汉字前后景分割

在获取了汉字位置矩形框并扩展后，就可以将其作为 GrabCut 算法的输入进行前后景分割。GrabCut 算法需要用户指定前景和背景的样本。本文采用了简单的方式：将扩展后的汉字位置矩形框内部所有像素点设为前景样本，将外部所有像素点设为背景样本。然后将这些样本作为 GrabCut 算法的输入，得到前后景分割的结果。具体而言，每一次 GrabCut 只能分割出一个汉字，如图 5 所示，因此多个汉字需要多次分割，然后把每一次的分割结果合并在一起，组成一张只包含汉字前景的图像，如图 6 所示。

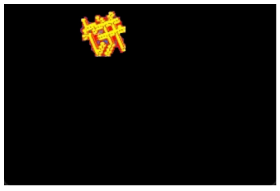

图 5　Grabcut 分割单字

Fig. 5　Grabcut split single word

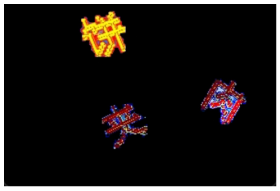

图 6　Grabcut 分割多字

Fig. 6　Grabcut split multiple words

最后，将得到的汉字分割结果保存下来，即可进行后续的文字识别或图像处理操作。

4）汉字倾斜矫正

汉字倾斜矫正是汉字识别的基础，如果文字倾斜，就会影响文字识别结果的准确性。以下是详细的矫正和识别步骤：

（1）首先，使用 OpenCV 将分割后的汉字前景图像进行二值化处理。遍历像素，只要不是黑色，就把灰度置为 255。

（2）对于每个汉字，使用 OpenCV 库函数 *findContours* 找到它的轮廓。

（3）然后，根据汉字轮廓计算最小外接矩形框是一个关键步骤，可以帮助确定汉字的倾斜方向和边界范围。使用 OpenCV 等计算机视觉库中的函数 *minAreaRect* 即可得到能够包含汉字的最小矩形框。该函数基于输入的二值化图像，计算出只包含目标轮廓点的最小旋转矩形和旋转角度，并用最小外接矩形框来记录该矫正后的汉字区域，如图 7 所示。

图 7　最小外接矩形

Fig. 7　Min area rect

（4）接下来，计算外接矩形框的倾斜角度，使用 *minAreaRect* 函数返回的旋转矩形的 angle 属性，如图 8 所示，旋转角度是水平轴逆时针旋转，与碰到的矩形的第一条边的夹角。

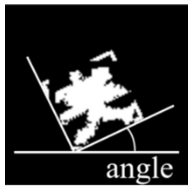

图 8　倾斜角度

Fig. 8　Skew angle

（5）最后，矫正汉字。以图 4 为例的极验汉字点选验证码，其旋转角度在±45°范围内，因此汉字有两种可能的旋转角度：一是如图 8 所示的左偏，此时矫正角度为 angle；二是右偏，矫正角度为－(90-angle)＝angle-90°。旋转矫正时先把汉字小图缩放成正方形，然后以中心点为旋转中心旋转，矫正后如图 9 所示。然而这是理想的情况，考虑到部分情况，如倾斜角度恰好为 45°左右，很可能被反向矫正导致旋转了 90°，或者，不像极验验证码一样有大致的旋转角度限制，一些旋转角度较大的汉字无法直接适用此方法，因此还需要额外处理。虽然以上倾斜矫正方法无法保证能够直接将汉字矫正，但却可以基本将汉字矫正到 0°，90°，180°，270° 4 个角度中的一个。

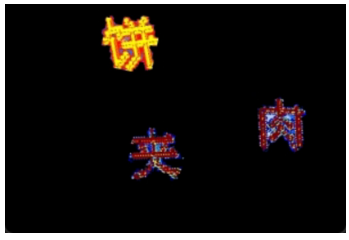

图 9　矫正后汉字图

Fig. 9　Corrected Chinese character image

4.2.2 汉字识别与匹配

汉字识别是汉字点选验证码识别方法中最后一个重要的步骤。由不同设计者设计的同一类型的验证码的风格大相径庭，针对某个风格自定义的文字识别模型，其泛化性能的表现难以胜任其他风格的新数据，而且针对新风格汉字重新训练数据集的成本较高。本文采用最先进的中文识别模型 PaddleOCR，并利用文字概率矩阵获得提示文字与文字小图的最佳匹配结果。

PaddleOCR 是基于 PaddlePaddle 深度学习框架搭建的端到端的开源 OCR(Optical Character Recognition)套件，旨在提供易用且功能强大的 OCR 解决方案。PaddleOCR 支持包括中英文在内的多种语言，具有超过 70 种通用字体和特殊场景字体的检测与识别能力(如手写字体、表格文字、竖排文字等)。其核心识别算法包括 EAST[38]，DB(Detecting Box)，CRNN[39]等多种结构，可以分别应用于文本检测和识别环节。PaddleOCR 中文识别是 PaddleOCR 套件中的一个重要功能。它能够完成对中文字符和汉字的准确检测和识别，被广泛应用于印刷体、手写体、数字等各种类型的文字识别场景，泛化性能远好于在单一风格自定义数据集上训练的文字识别模型。并且，PaddleOCR 是轻量的。

然而，对于倾斜的、变形的、包含复杂干扰的单个汉字，PaddleOCR 仍然存在着识别准确率不高的缺点。

4.2.1 小节中提到，汉字倾斜矫正方法能够将汉字矫正到 0°，90°，180°，270° 4 个角度，因此对于每个目标检测得到的汉字矩形框，能够生成 4 张图片。将每张图片都输入 PaddleOCR 模型中，获得其对应每个候选汉字的概率，形成概率矩阵。验证码题目中的候选汉字是“肉夹饼”，经过汉字检测和汉字倾斜矫正后，获得了 3×4=12 张汉字小图，其对应“肉夹饼”3 个字的概率如表 3 所列。

表 3 文字概率矩阵 1

Table 3 Text probability matrix 1

	肉	夹	饼
	6.62×10^{-8}	9.52×10^{-9}	0.86161
	2.45×10^{-10}	2.40×10^{-9}	0.71167
	0.00063	6.27×10^{-5}	0.00012
	1.58×10^{-5}	2.78×10^{-6}	1.82×10^{-5}
	8.57×10^{-7}	5.36×10^{-7}	2.73×10^{-5}
	0.81962	3.70×10^{-8}	1.67×10^{-7}
	1.10×10^{-6}	4.01×10^{-6}	9.30×10^{-7}
	9.74×10^{-5}	7.73×10^{-7}	1.21×10^{-5}
	1.38×10^{-6}	4.01×10^{-7}	4.98×10^{-6}
	1.27×10^{-6}	0.24233	1.52×10^{-8}
	2.28×10^{-7}	3.82×10^{-10}	7.94×10^{-9}
	0.00055	3.30×10^{-6}	0.00021

获得汉字矩形框和候选汉字最佳匹配结果的步骤如下：

1)对于每个汉字矩形框，从 4 个角度中筛选最佳角度，即先从 4 行概率中选择一行，选择的应当是 4 行中最大概率值所在行。在表 3 中，选择的应是第 1、第 6 和第 10 行。概率矩阵缩小到 3 行，如表 4 所列。

表 4 文字概率矩阵 2

Table 4 Text probability matrix 2

	肉	夹	饼
	6.62×10^{-8}	9.52×10^{-9}	0.86161
	0.81962	3.70×10^{-8}	1.67×10^{-7}
	1.27×10^{-6}	0.24233	1.52×10^{-8}

2)迭代查找最大值并删除所在行列，得到汉字矩形框和候选汉字的最佳匹配结果。具体而言，如表 4 所列，第一轮：最大概率为第一行第三列的 0.86161，意味着第一行和第三列是当前最佳匹配，即第一个矩形框和第三个候选汉字匹配，随即删除第一行和第三列，因为第一个矩形框不再与其他候选汉字匹配，第三个候选汉字不再与其他矩形框匹配。第二轮：如表 5 所列，在剩下的 2×2 概率矩阵中，最大概率值为第一行第一列的 0.81962，其对应原始概率矩阵中第二行第一列的位置，即第二个矩形框与第一个候选汉字匹配，随后删除其所在行列。第三轮：由于本例中汉字矩形框数量和候选汉字数量一致，因此此时概率矩阵只剩下一个概率(如果汉字矩形框数量大于候选汉字数量，此时概率矩阵还不止一行)，根据排除法，第三个矩形框与第二个候选汉字匹配。至此，已得到汉字矩形框与候选汉字的最佳匹配结果。

表 5 文字概率矩阵 3

Table 5 Text probability matrix 3

	肉	夹
	0.81962	3.70×10^{-8}
	1.27×10^{-6}	0.24233

根据匹配结果，以候选汉字为顺序，记录对应汉字矩形框中心点的坐标位置，即得到了汉字点选验证码的答案，即需要鼠标点击的位置及顺序，如图 10 所示。

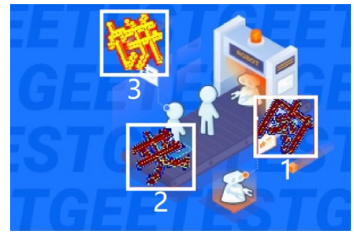

图 10 汉字点选验证码答案

Fig. 10 Answer to the Chinese character click-based captcha

5 实验

5.1 实验设置

本研究的实验环境及相关参数设置如表 6 所列。

表 6　实验环境设置

Table 6　Experimental environment settings

参数	内容
处理器	Intel(R) Core(TM) i9-10940X CPU @ 3.30GHz
显卡	GeForce RTX 3090
操作系统	Ubuntu 22.04.2 LTS
Python 版本	3.7.12
Pytorch 版本	1.12.1+cu116
训练批大小	16
训练轮次	300

5.2　汉字检测模型评估

5.2.1　检测层简化方案

如表 7 所列,检测层简化实验表明:删除一个或两个检测层,模型复杂度降低,而检测性能降低很小,甚至还有可能提高。删除小检测层的 Model6 的 mAP 值表现最好,达到 85.1,分别只保留中检测层和大检测层的 Model2 和 Model3 的 mAP 值达到了 85。对比模型的复杂度可以发现,Model2 在保持高性能的同时,参数量相比 Model3 下降 38.5%,计算量下降 15.7%,相比原 YOLOv8-n,参数量和计算量更是分别减少 32.3%和 27.2%。

表 7　检测层简化方案对比

Table 7　Comparison of simplified detection layer schemes

Model	小层	中层	大层	mAP50:95	Params/M	FLOPs/G
YOLOv8-n	√	√	√	84.7	3.00	8.1
Model1	√			84.5	1.60	6.1
Model2		√		85.0	2.03	5.9
Model3			√	85.0	3.62	6.2
Model4	√	√		84.8	1.99	7.3
Model5	√		√	84.4	2.78	7.4
Model6		√	√	**85.1**	3.30	7.0

5.2.2　下采样倍率对比

如表 8 所列,Model7,Model8,Model9 在 Model1,Model2,Model4 的基础上把下采样倍率从 32 倍降低到了 16 倍,其中 mAP 性能最好的是仅保留中检测层的 Model8(对应的 Model2 也是上一实验中综合表现最佳的模型)。Model8 相比 Model2,参数量和计算量分别大幅减少 46.3%和 45.8%,mAP 略下降 0.6%,属于可接受范围内的轻微损失。实验证明改进对于模型轻量化有很大益处,且代价不大。

表 8　下采样倍率对比

Table 8　Downsampling ratio comparison

Model	小层	中层	下采样倍率	mAP50:95	Params/M	FLOPs/G
Model1	√		32	84.5	1.60	6.1
Model2		√	32	85.0	2.03	5.9
Model4	√	√	32	84.8	1.99	7.3
Model7	√		16	84.0	0.62	5.2
Model8		√	16	84.4	1.09	3.2
Model9	√	√	16	83.9	1.02	6.5

5.2.3　DWConv 模块

如表 9 所列,DWConv 实验中 Model10 和 Model11 分别在原版 YOLOv8-n 和 Model8 的基础上进行修改,结果表明,DWConv 能够减少 20%左右的参数量和计算量,降低模型复杂度,且对 mAP 影响很小。

表 9　引入 DWConv 模块实验

Table 9　Experiment on DWConv module introduction

Model	DWConv	mAP50:95	Params/M	FLOPs/G
YOLOv8-n		84.7	3.00	8.1
Model8		84.4	1.09	3.2
Model10	√	84.5	2.44	7.0
Model11	√	84.1	0.82	2.3

5.2.4　C2fGhost 模块实验

如表 10 所列,C2fGhost 实验中 Model12 和 Model13 分别在原版 YOLOv8-n 和 Model8 的基础上进行修改,结果表明,C2fGhost 也能够减少 30%左右的参数量和计算量,且对 mAP 影响较小,Model13 相比 Model8 mAP 甚至有所提高。

表 10　引入 C2fGhost 模块实验

Table 10　Experiment on C2fGhost module introduction

Model	C2fGhost	mAP50:95	Params/M	FLOPs/G
YOLOv8-n		84.7	3.00	8.1
Model8		84.4	1.09	3.2
Model12	√	84.5	2.10	5.9
Model13	√	83.9	0.81	1.9

5.2.5　DWConv 和 C2fGhost 综合效果

综合 DWConv 和 C2fGhost 的有益效果,如表 11 所列,分别在原版 YOLOv8-n 和 Model8 的基础上同时应用了 DWConv 和 C2fGhost,Model14 在 YOLOv8-n 的基础上减少了 49%的参数量和 42%的计算量,mAP 降低 1%;Model15 在 Model8 的基础上减少了 48%的参数量和 66%的计算量,mAP 降低 1.3%。

结果表明,同时应用 DWConv 和 C2fGhost 能够大幅减少参数量和计算量,相比之下,mAP 值的降低有限,意味着对于 YOLOv8-n 模型轻量化的收益大而代价较小,总体上取得了有益的效果。

表 11　DWConv 和 C2fGhost 综合效果

Table 11　Comprehensive effects of DWConv and C2fGhost

Model	DWConv	C2fGhost	mAP50:95	Params/M	FLOPs/G
YOLOv8-n			84.7	3.00	8.1
Model8			84.4	1.09	3.2
Model14	√	√	83.7	1.53	4.7
Model15	√	√	83.1	**0.57**	**1.1**

5.2.6　与其他模型的详细对比

经过多次改进后的 Model15 与原版 YOLOv8-n、YOLOv7-t 以及过去相关研究中用到的 Faster-RCNN 的详细对比如表 12 所列。

表 12　改进前后模型与其他模型的详细对比

Table 12　Detailed comparison of the improved model with other models

Model	Ours	YOLOv8-n	YOLOv7-t	Faster RCNN (ResNet50)
Precision	99.7	99.7	99.6	98.9
Recall	99.2	99.6	99.6	98.6
*mAP*50	99.5	99.5	99.8	98.9
*mAP*50:95	83.1	84.7	80.1	73.4
Params/M	**0.57**	3.00	6.00	28.29
FLOPs/G	**1.1**	8.1	13.1	946.5
Model size/MB	**1.3**	6.2	12.3	108.2
CPU time/ms	**16.4**	21.8	86.4	8882.3
GPU time/ms	**12.5**	14.8	77.6	104.2

综上所述,针对数据集目标较小且尺寸相似尺度相对

单一的特点，Model15(Ours)在 YOLOv8n 的基础上，去除了大小两个检测层，只保留一个中尺度检测层，Backbone 下采样率从 32 倍降低到了 16 倍，又借鉴了轻量级网络的设计思想，引入了 DWConv 和 C2fGhost。相比 YOLOv8n，模型参数量减少 81%，计算量减少 86.4%，模型权重文件减小 79%，CPU 推理时间减少 25%，代价是 mAP50:95 减小 1.6 个百分点。然而从准确率、召回率、F1-score 和 mAP50 来看，简化后的模型仍然收敛得足够好，对实际检测任务准确率的影响极小。

与其他目标检测模型相比，改进的模型在所有指标上都领先，其轻量化优势非常明显。值得一提的是，与先前研究广泛采用的 Faster RCNN(ResNet50)相比，模型计算量减少 99.9%，在 i9-10940X 上的推理时间减少 99.8%，在 RTX3090 上的推理时间则减少 88%。这种优势使得改进的模型在计算资源受限的设备上也能较快推理，在多线程并发场景下能较大程度地节省计算资源。

5.3 汉字识别方案评估

汉字识别方法依赖成熟的 PaddleOCR 模型，但汉字检测的图片后处理方法也必不可少，后处理方法在测试集上的表现如表 13 所列。实验评估了应用图片后处理方法后汉字点选验证码最终点选位置及顺序都成功的概率，结果表明了后处理方法的有效性：前后景分割可以提高约 2%的最终准确率，倾斜矫正可以提高 12%左右的最终准确率，使用概率矩阵获得最佳匹配的方法对最终准确率的贡献到达了 19%。

表 13 后处理方法效果对比

Table 13 Comparison of post-processing methods' effectiveness

Method	前后景分割	倾斜矫正	概率矩阵最佳匹配	Acc/%
Method1	√	√		77.0
Method2	√		√	84.5
Method3		√	√	94.0
Method4	√	√	√	96.0

本研究没有对比已有的相关研究的汉字识别方案，原因如下：

1)缺乏可比性。验证码风格迭代升级，风格不同、识别难度不同。原有的风格大多仍以标准印刷体为主，带有小角度旋转，没有明显扭曲干扰，其识别难度比现在主流的场景更简单。而本研究采用的新方案能够胜任较难场景的文字识别，若要测试其在更简单场景的表现，也缺乏必要性。

2)缺乏可行性。旧场景验证码图片已无法获取，且相关研究作者没有公开数据集，因此无论是将新方案应用到旧场景还是将旧方案应用到新场景都不具备切实条件。

3)PaddleOCR 采用了海量的场景文字训练模型，应用到本研究的场景中，在没有经过针对性训练的情况下，就能够胜任复杂文字场景，足以证明其泛化性能强。而旧方案多采用简单的 CNN 网络，针对特定场景的文字进行数据量有限、成本较高的训练，其局限性较大。

结束语 本研究面向流程自动化场景中汉字点选验证码识别的难题，并试图解决已有方法存在的数据集制作低效、泛化性弱、复杂度与性能不平衡等问题，提出了一种轻量级的高效识别方法。首先考虑到汉字检测数据集尺度单一且较小的特点，针对性地轻量化了 YOLOv8-n 模型，并且检测性能没有明显损失；然后对汉字图片进行分割、矫正等必要的预处理；再采用轻量的泛化性强的 PaddleOCR 模型进行汉字识别，并通过识别概率矩阵得到汉字图片和文本的最佳匹配结果。此外，设计了一种半自动的汉字检测数据集构建流程并公开了数据集。

实验证明了本研究方法的有效性。与已有方法相比，汉字检测数据集制作更加简单高效，检测模型大幅度轻量化，无需自制汉字识别数据集，场景迁移代价低，结合图片后处理方法，识别准确率大幅提高。

本研究旨在推动汉字点选验证码自动识别技术的发展，促进企业流程自动化水平的提升，同时为验证码设计和安全性提供参考。尽管本研究在点选验证码识别领域取得了一些成果，但仍有一些方面值得进一步探讨和完善，比如部分场景下汉字检测模型仍然存在检测错误的情况。下一步的工作中计划进一步改进汉字检测模型以减少检测错误，并考虑多次验证失败场景下的异常处理机制。

参 考 文 献

[1] ENRÍQUEZ J G, JIMÉNEZ-RAMÍREZ A, DOMÍNGUEZ-MAYO F J, et al. Robotic process automation: a scientific and industrial systematic mapping tudy[J]. IEEE Access, 2020, 8: 39113-39129.

[2] LIANG Y. Research on the Application of Financial Robot Process Automation Based on Machine Learning[C]// 2021 International Conference on Electronic Information Technology and Smart Agriculture(ICEITSA). IEEE, 2021: 474-477.

[3] HANDOKO B L, LINDAWATI A S L, MUSTAPHA M. Robotic process automation in audit 4.0[C]// The 2021 12th International Conference on E-business, Management and Economics. 2021: 128-132.

[4] BRANDSTATTER C, TSCHANDL M, MITTERBACK C. A Generic Process Model for the Introduction of Robotic Process Automation in Financial Accounting[C]// Proceedings of the 2023 9th International Conference on Computer Technology Applications. 2023: 12-18.

[5] AHMET UNAL M, BOLUKBAS O. The Acquirements of Digitalization with RPA(Robotic Process Automation) Technology in the Vakif Participation Bank[C]// Proceedings of the 4th International Conference on Information Science and Systems. 2021: 68-73.

[6] LIU S, QI X, LI H. Practice of Robot Process Automation in Power Grid Dispatching Report[C]// Proceedings of the 2022 4th International Conference on Robotics, Intelligent Control and Artificial Intelligence. 2022: 212-216.

[7] RATIA M, MYLLÄRNIEMI J, HELANDER N. Robotic process automation-creating value by digitalizing work in the private healthcare? [C]// Proceedings of the 22nd International Academic Mindtrek Conference. 2018: 222-227.

[8] SGANDERLA R B, FANTINATO M, THOM L H. Robotic Process Automation in Latin American Organizations: Survey and Evaluation of the Current State of Technology Adoption [C]// Proceedings of the XIX Brazilian Symposium on Information Systems. 2023: 459-467.

[9] KEDZIORA D, HYRYNSALMI S. Turning Robotic Process Automation onto Intelligent Automation with Machine Learning [C]// Proceedings of the 11th International Conference on Communities and Technologies. 2023: 1-5.

[10] KHOLIYA P S,KAPOOR A,RANA M,et al. Intelligent process automation:The future of digital transformation[C]// 10th International Conference on System Modeling & Advancement in Research Trends(SMART 2021). IEEE,2021:185-190.
[11] ROTHER C,KOLMOGOROV V,BLAKE A. "GrabCut" interactive foreground extraction using iterated graph cuts[J]. ACM Transactions on Graphics(TOG),2004,23(3):309-314.
[12] DU Y,LI C,GUO R,et al. Pp-ocr:A practical ultra lightweight ocr system[J]. arXiv:2009.09941,2020.
[13] BOSTIK O,KLECKA J. Recognition of CAPTCHA characters by supervised machine learning algorithms[J]. IFAC-Papers On-Line,2018,51(6):208-213.
[14] SACHDEV S. Breaking captcha characters using multi-task learning cnn and svm[C]// 4th International Conference on Computational Intelligence and Networks(CINE 2020). IEEE,2020:1-6.
[15] ZHANG N,EBRAHIMI M,LI W,et al. Counteracting dark Web text-based CAPTCHA with generative adversarial learning for proactive cyber threat intelligence[J]. ACM Transactions on Management Information Systems(TMIS),2022,13(2):1-21.
[16] THOBHANI A,GAO M,HAWBANI A,et al. CAPTCHA recognition using deep learning with attached binary images[J]. Electronics,2020,9(9):1522.
[17] WU X,DAI S,GUO Y,et al. A machine learning attack against variable-length Chinese character CAPTCHAs[J]. Applied Intelligence,2019,49:1548-1565.
[18] LUAN S,CHEN C,ZHANG B,et al. Gabor convolutional networks[J]. IEEE Transactions on Image Processing,2018,27(9):4357-4366.
[19] WANG J,QIN J,XIANG X,et al. CAPTCHA recognition based on deep convolutional neural network[J]. Math. Biosci. Eng,2019,16(5):5851-5861.
[20] ZHANG X,LIU X,SARKODIE-GYAN T,et al. Development of a character CAPTCHA recognition system for the visually impaired community using deep learning[J]. Machine Vision and Applications,2021,32:1-19.
[21] BI X,LIU X. Chinese Character Captcha Sequential Selection System Based on Convolutional Neural Network[C]// International Conference on Computer Vision,Image and Deep Learning(CVIDL 2020). IEEE,2020:554-559.
[22] GIRSHICK R. Fast r-cnn[C]// Proceedings of the IEEE International Conference on Computer Vision. 2015:1440-1448.
[23] CAVNAR W B,TRENKLEJ M. N-gram-based text categorization[C]// 3rd Annual Symposium on Document Analysis and Information Retrieval(SDAIR-94). 1994:161175.
[24] HU J. Research on Security of Chinese Point-and-Click CAPTCHA[D]. Xi'an:Xidian University,2018.
[25] YOU X. Research on Chinese Character Captcha Recognition Based on YOLO V2 [D]. Chengdu:Chengdu University of Technology,2019.
[26] REDMON J,DIVVALA S,GIRSHICK R,et al. You only look once:Unified,real-time object detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:779-788.
[27] LIU W,ANGUELOV D,ERHAND,et al. Ssd:Single shot multibox detector[C]// Computer Vision — ECCV 2016:14th European Conference,Amsterdam,The Netherlands,Part I 14. Springer International Publishing,2016:21-37.
[28] WANG C Y,BOCHKOVSKIY A,LIAOH Y M. YOLOv7:Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023:7464-7475.
[29] GE Z,LIU S,WANG F,et al. Yolox:Exceeding yolo series in 2021[J]. arXiv:2107.08430,2021.
[30] XU S,WANG X,LV W,et al. PP-YOLOE:An evolved version of YOLO[J]. arXiv:2203.16250,2022.
[31] LI C,LI L,JIANG H,et al. YOLOv6:A single-stage object detection framework for industrial applications[J]. arXiv:2209.02976,2022.
[32] CHOLLET F. Xception:Deep learning with depthwise separable convolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:1251-1258.
[33] HOWARD A G,ZHU M,CHEN B,et al. Mobilenets:Efficient convolutional neural networks for mobile vision applications[J]. arXiv:1704.04861,2017.
[34] ZHANG X,ZHOU X,LIN M,et al. Shufflenet:An extremely efficient convolutional neural network for mobile devices[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:6848-6856.
[35] KOONCE B. EfficientNet[M]// Convolutional Neural Networks with Swift for Tensorflow:Image Recognition and Dataset Categorization. 2021:109-123.
[36] HAN K,WANG Y,TIAN Q,et al. Ghostnet:More features from cheap operations[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020:1580-1589.
[37] REYNOLDSD A. Gaussian mixture models[J]. Encyclopedia of Biometrics,2009,741:659-663.
[38] ZHOU X,YAO C,WEN H,et al. East:an efficient and accurate scene text detector[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:5551-5560.
[39] SHI B,BAI X,YAO C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2016,39(11):2298-2304.

JIN Xinhao, born in 1998, postgraduate. His main research interest is robot process automation.

CHI Kaikai, born in 1980, Ph.D, professor, Ph.D supervisor, is a member of CCF(No. 72583S). His main research interests include wireless networks and machine learning.