Editing Practice

ORIGINAL ARTICLE

A one-stop solution for generating, inserting, and verifying quick response codes in scientific journals based on Python

Xinyue Niu[1,*], Lijuan Qi[1,2], Mei Fang[1,2]

[1]Editorial Office of Computer Systems and Applications, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

[2]Editorial Office of Journal of Software, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

ABSTRACT

Background: This study aims to make quick response (QR) code processing operable, reliable, and efficient. It discusses the exploration and practice of QR code processing in *Computer Systems & Applications*, providing a useful reference for editors. **Methods:** Common problems encountered when manually handling QR codes are summarized and analyzed. A QR code-integrated tool is designed using Python language to propose an integrated solution for QR codes, and the tool is compared with manual solutions. **Results:** The self-developed QR code-integrated tool can realize the automatic processing of tasks, such as generating, inserting, and verifying QR codes, thereby achieving high efficiency and reliability. For example, it takes about 2 hours to handle 45 papers by hand, whereas automated operations require only about 30 seconds, without any errors. The use of tools to automatically verify QR codes no longer relies on uniform resource locator (URL) activation, achieving perfect integration between QR code processing flow and the traditional editorial publishing process. **Conclusion:** Using Python, this study demonstrates that high-quality QR code settings can be achieved efficiently in single articles in scientific journals. Computer automation operations have high efficiency and strong reliability in dealing with repetitive work. Scientific journal editors should think creatively and continuously improve their work efficiency by adopting computer technology to refine workflow.

Key words: science and technology journals, quick response code, Python, editorial work

A two-dimensional barcode is a geometric pattern organized in some specific way and distributed on a plane that records data symbols.[1] It can store large amounts of textual information, images, or even videos without relying on databases or networks. Therefore, it has been widely used in the fields of digital payment, product traceability, and push advertising. In May 2011, *Cell* used a quick response (QR) code as its cover image. By scanning the QR code, one can watch videos about malaria parasites entering blood circulation through infected cells. This marked the beginning of the use of QR codes in scientific journals.[2]

By using QR codes, scientific journals are exploring a new reading and information dissemination scenario.[3] These codes can be placed on the cover, or back cover of a journal, or displayed on the front page or on the content of papers. It can quickly guide readers to new media, compensate for the limitations of printed journals in dissemination, expand the publicity channels of scientific journals, enhance reader experience, increase interaction between authors and readers, and achieve enhanced publication and extended reading of articles.[4,5] Tan *et al.*[6] believe that editors should master the skill of making QR codes and help optimize the numbers and

***Corresponding Author:**
Xinyue Niu, Editorial Office of Computer Systems and Applications, Institute of Software, Chinese Academy of Sciences, No. 4 Zhongguancun South Street, Haidian District, Beijing 100190, China. Email: niuxinyue@iscas.ac.cn; https://orcid.org/0000-0002-1493-5750

Received: 16 December 2024; Revised: 21 December 2024; Accepted: 27 December 2024
https://doi.org/10.54844/ep.2024.0824

positions of journal QR codes. Although QR codes play an intermediary role in the multimedia era, only a few scientific journals have used them. Yang conducted a study of the member journals of the Society of China University Journals,[7] analyzing the number of journals that included QR codes in individual articles, as well as the content and format of these QR codes. The results showed that currently, only 24.9% of member journals have integrated QR codes on a single paper, with no specific rules governing their positions, sizes, or styles. Some links to the content of QR codes are of low value, making it difficult to provide high-quality value-added services. Zhang *et al.* found that some papers' QR code links are invalid after scanning and thus cannot provide the services needed.[8] The authors emphasized that editors should pay attention to the validity of QR codes. Wang stated that while providing rich content, QR codes also involve many problems, such as the uneven quality of link contents for QR codes, thereby increasing editing costs by requiring more checks for them.[9]

To reduce editing and proofreading costs, some researchers have attempted to use batch processing methods to create QR codes. Guo *et al.* introduced the QR code generator ZZY.QR (http://zzy.my/blog/post/ZZYQR)[10] and described the steps for using it in a batch encoding operation. Zhang *et al.* basing their study on that of Guo *et al.* introduced the usage method of the CaoLiao QR code and studied the general norms for placing QR codes in domestic and foreign journals.[10,11] Batch coding indeed reduces manual labor costs for academic journals with dozens of publications per issue. In practical work, besides the task of creating QR codes, inserting and verifying them also require manual operations. Zeng used *Computer Engineering* as an example to share experiences in generating QR codes using the RAYS system (Wuhan Key Laboratory of Integrated Development, China).[12] The RAYS system considers integrating the creation of QR codes into its intelligent editing system from the beginning, but this QR code content is limited to the RAYS question-and-answer system, does not support the customization of QR code links, and fails to provide an automated solution for inserting and verifying QR codes. Gong *et al.* introduced the use experience and implementation details of QR codes used by journals under the publishing media group of China Coal Science Research Institute Publishing Co., Ltd.[13] In the process of creating and verifying QR codes, a web page using source data from papers is first created, and uniform resource locators (URLs) and corresponding QR codes are obtained; then QR codes are inserted and verified while proofreading, and, finally, the contents of the webpage after final revision are replaced. The processing flow of QR codes described in the literature has been relatively mature,[13] but there are still two shortcomings: (1) QR codes need manual handling by editors, which requires much work; and (2) when verifying the URL inside the QR code, the editorial department needs to build the homepage of the paper first and fill in or replace the contents of the webpage again after the official publication of the paper. That is, during the publishing process, website contents need to be handled repeatedly, resulting in complex publishing processes.

An analysis of the implementation methods of existing editorial departments shows that most studies commonly introduced batch generation of QR codes, but they rarely introduced insertion and verification. At present, there are many limitations in the current QR code processing process, such as large workloads, a complex operation flow, and high coupling between processes. The introduction of computer technology is considered to improve the process and reduce workloads. The entire process of QR code handling can be divided into multiple stages, namely, generation, insertion, and verification, which are repetitive operations. The verification process is highly dependent on machine recognition. The workflow fits with the computer processing flow. Python language is known for its ease of learning, powerful functions, and rich open-source frameworks, making it an ideal language for rapid application development.[14] Some editors have introduced Python language into editorial work, using Python to develop a set of generation and printing programs that simplify editorial work and promote journal integration development.[15] Capitalizing on the advantages and features of Python, this study designed and built a QR code-integrated tool based on Python, realizing the full automation of QR code generation, insertion, and verification.

This study takes *Computer Systems & Applications* as an example to discuss the exploration and practice of the entire process of QR code processing. It is hoped that this will provide a reference for editors and their peers. First, this study introduces the original manual process of handling QR codes and analyzes the shortcomings of the existing process. Second, it presents the development tools, design ideas, and operation processes of integrated solutions for QR codes. Third, it compares the flow chart designed with the original manual process of handling QR codes and analyzes the application value of the integrated solution for QR codes. Finally, it summarizes previous work and looks forward to the future use of Python in editing work.

MANUAL PROCESSING FLOW AND ANALYSIS OF QR CODES

Since its issue no. 1, volume 38, in 2019, *Computer Systems & Applications* has issued QR codes for each

paper (Figure 1). Readers can scan the QR code to read the full electronic version of a paper.

Manual processing of QR codes

Figure 2 shows the entire process of manually processing QR codes in *Computer Systems & Applications*. It mainly includes generating and inserting a two-dimensional code. These two tasks are handled before typesetting the paper for publication. Editors use the CaiLiao QR Code Generator (https://cli.im/) to create unique QR codes for each article, and then insert them into the final version of the electronic manuscript submitted by the author in Word format. Code verification is arranged during the blue sample proofreading stage. As a black-and-white filled matrix is used to store data for the QR code, human eyes cannot identify whether the QR code is correct during editing, processing, and proofreading; its corresponding link is the URL of the paper on the journal's official website—which is invalid before online publication and unable to perform QR code verification work. Therefore, the editorial department of *Computer Systems & Applications* schedules the online publication work before the printing of the current issue's papers. During the proofreading stage of the blue proofs, QR codes for each paper are scanned and verified. If any errors are found in the QR codes, they are replaced with the correct ones, and the online full-text versions are updated accordingly.

Analysis of process characteristics

The manual processing flow of QR codes can complete the work of generating, inserting, and verifying QR codes. However, it involves limitations, such as a large workload, repeated operations, difficult verification, and complex processes, as described in the following.

The operation is repeated, and the workload is large

In the QR code generation stage, editors need to use the CaoLiao QR Code Generator to generate and save individual QR codes. In the QR code insertion stage, editors need to open each paper Word document one by one, insert pictures, and adjust their positions and sizes. In the QR code verification stage, editors also need to scan every paper's QR code with a mobile phone one by one, open URLs, and check information, such as the titles and authors of papers. Obviously, this processing flow entails much work, which consumes much time and energy for editors. *Computer Systems & Applications* publishes about 45 papers per issue. A skilled editor needs approximately 1.5 hours to complete all the QR code generation and insertion tasks in an issue, while it takes them roughly 30 min to finish the QR code verification task.

It is easy to make mistakes, as the checking process is difficult

The whole process of handling QR codes by hand involves many manual operations that are cumbersome and repetitive, making it very likely that errors will occur. As the QR code consists of black-and-white matrices that cannot be verified visually, verification at this stage is indispensable. *Computer Systems & Applications* uses WeChat's scan code function to verify a QR code. This feature requires the link stored in the QR code to work; during editing and proofreading, the website is not yet completed, so comparison or correction is not possible. Errors will only be found after publication on the network.

The dependency between processes is too high

A contradiction exists in the manual processing of the QR code process. That is, before journal publication, QR code verification is needed, and this depends on network publishing. To address this contradiction, *Computer Systems & Applications* will advance network publishing ahead of paper publishing. That is; after proofreading for delivery printing, the journal will publish online papers of the current issue at the blue sample stage to check QR codes. If there are errors found during checking, they are corrected, and the printed version of the current page is remade while replacing the network edition simultaneously. It can be seen from this description that the three stages of network publishing, QR code verification, and paper publishing are strongly dependent on one another's results. In the actual publishing process, editors are under great pressure, and they reserve enough time to deal with unexpected situations.

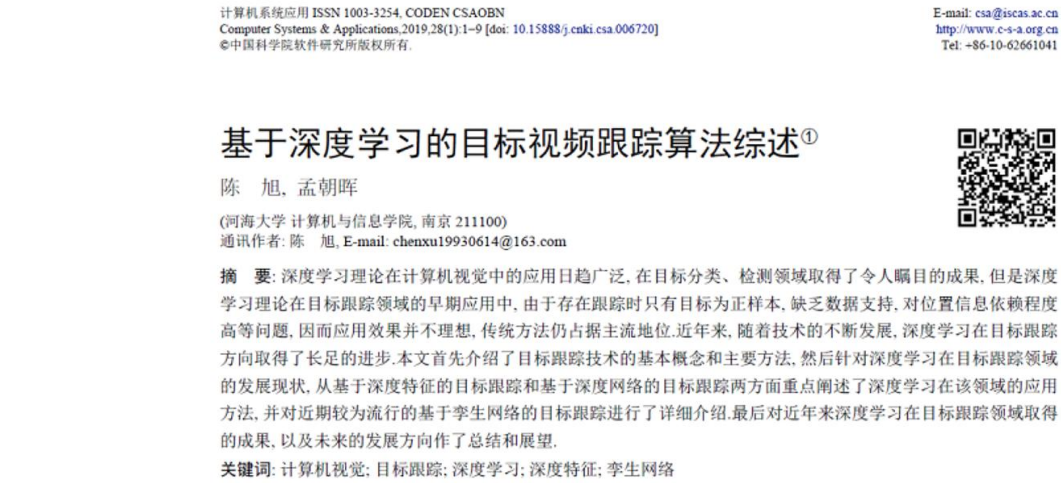
INTEGRATED QR CODE SOLUTION

To address problems in the manual processing of QR codes, this study developed a unified processing tool for QR codes based on Python to achieve full automation processing of QR codes.

Development tools

An integrated development environment

PyCharm (Community Edition), developed by JetBrains for Python developers, is a free, integrated development environment and is plug and play.[16] It comes with an embedded Python interpreter and a graphical run-debugging interface so that developers do not need to install Python themselves, configure environments, or input commands; they only need to open PyCharm to write code and run Python programs. In addition, PyCharm includes standard libraries in Python and supports common operations, such as file access, directory access, and operating system services. At the same time, users can install third-party extension


计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(1):1–9 [doi: 10.15888/j.cnki.csa.006720]
©中国科学院软件研究所版权所有.

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于深度学习的目标视频跟踪算法综述①

陈 旭, 孟朝晖

(河海大学 计算机与信息学院, 南京 211100)
通讯作者: 陈 旭, E-mail: chenxu19930614@163.com

摘 要: 深度学习理论在计算机视觉中的应用日趋广泛, 在目标分类、检测领域取得了令人瞩目的成果, 但是深度学习理论在目标跟踪领域的早期应用中, 由于存在跟踪时只有目标为正样本, 缺乏数据支持, 对位置信息依赖程度高等问题, 因而应用效果并不理想, 传统方法仍占据主流地位.近年来, 随着技术的不断发展, 深度学习在目标跟踪方向取得了长足的进步.本文首先介绍了目标跟踪技术的基本概念和主要方法, 然后针对深度学习在目标跟踪领域的发展现状, 从基于深度特征的目标跟踪和基于深度网络的目标跟踪两方面重点阐述了深度学习在该领域的应用方法, 并对近期较为流行的基于孪生网络的目标跟踪进行了详细介绍.最后对近年来深度学习在目标跟踪领域取得的成果, 以及未来的发展方向作了总结和展望.

关键词: 计算机视觉; 目标跟踪; 深度学习; 深度特征; 孪生网络

Figure 1. Example of QR code for *Computer Systems & Applications*. QR, quick response.

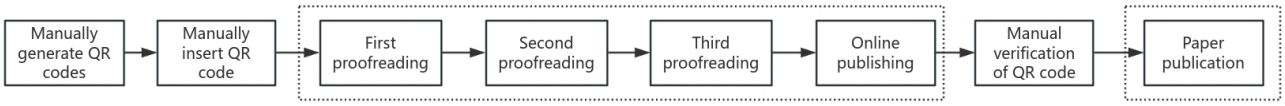


Figure 2. Process of manually processing QR codes. QR, quick response.

libraries through graphic interfaces to achieve more functions.

Third-party extension libraries

The Python Package Index is an official third-party extension library index warehouse of Python that stores the extended libraries published and shared by other members in the Python community.[17] In this study's developed QR code-integrated tool, the following are mainly used: (1) pywin, which calls the Windows application programming interface (API) to insert or delete operations on Word files; (2) PyPDF, which cuts out or parses PDF files; (3) qrcode, which generates or parses QR code images according to configuration settings.

Design ideas

The development idea of a QR code-integrated tool is mainly based on two aspects: (1) repetitive operations, such as opening/closing files, adjusting image sizes, and inserting images at specified positions, are delegated to tools for automation; and (2) the generation and verification of QR codes are achieved by calling Python library functions. The editorial office of *Computer Systems & Applications* began exploring the integration of QR codes using Python in August 2021, gradually improving its functionality and optimizing editing processes (Figure 3). Editors first use an integrated tool to generate/insert QR codes. After completing processing, they then format the paper's Word document and engage in traditional editing workflows. During this process, editors can verify QR codes any time with the integrated tool. The cost of manual verification and the time spent verifying QR codes are very low, allowing for multiple verifications. Currently, *Computer Systems & Applications* arranges two rounds of QR code verification during the second and third proofreading stages. The advantages include the following: (1) moving QR code verification forward to the second proofreading stage so that errors can be detected and corrected early, preventing subsequent procedures from being affected; and (2) multiple verifications, which ensure accurate and error-free QR codes. Once all verifications have been completed, editors can complete follow-up processes according to their schedules, including online and print publication.

The core work of the integrated processing tool for QR codes is the completion of the generation and insertion of QR codes, as well as their verification. This study introduces the development ideas of this tool by taking the example of generating and inserting QR codes (Figure 4). First, user configuration information is read, and all files that need to be processed under the specified directory are scanned according to the file directories designated in the configuration information. Second, based on the configuration information, the

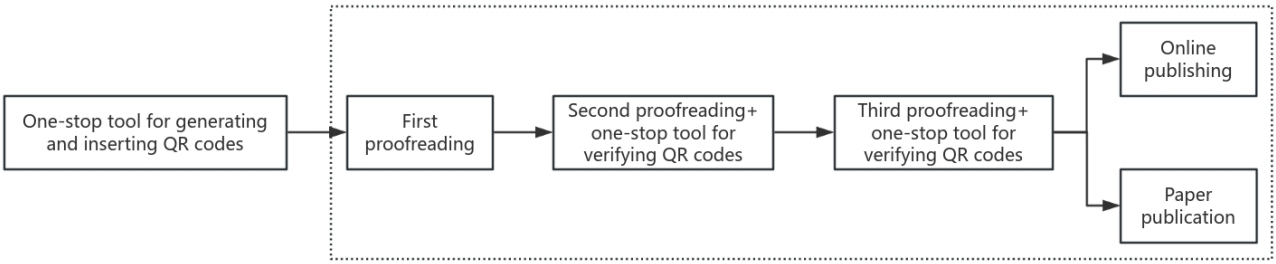


Figure 3. Process of automatic processing QR codes. QR, quick response.

program determines whether to splice the generated content or read the QR code content for each file, generates a QR code image, adjusts the picture position and size according to the configuration information, and then inserts it to a specified location in the corresponding Word document. Finally, after all file processing has been completed, the tool generates the processed results, indicating failure or success of the process. All QR code verification processes are similar, with the differences being that each file's processing stage captures an image from a designated location, identifies information contained within the image QR code, and compares this against expected data.

Operation manual of tools

The QR code-integrated tool developed in this study can complete the generation and insertion of a QR code, as well as its verification.

Generation/insertion of QR codes

The QR code-integrated tool combines the generation and insertion of a QR code into one step to make it more convenient and efficient. The interface is shown in Figure 5. The user sets the directory to the folder where the Word document that needs processing is located. According to different options for "journal", the QR code-integrated tool supports two modes for generating/inserting QR codes: the general version mode and the personalized mode.

General version mode

The user needs to select "General" in the "Journal" tab to enter this mode (Figure 5a). The general version supports three identification (ID) extraction methods: front, back, and Excel. In the Excel method, users establish a relationship between file names and QR code meanings through an Excel table (Figure 6). Users create their own Excel files containing two columns, A and B, representing the file name of the word document and the expected content in the QR code, respectively. Each row in the Excel file represents the corresponding relationship between column A's referred file and column B's referred content. The contents of the QR

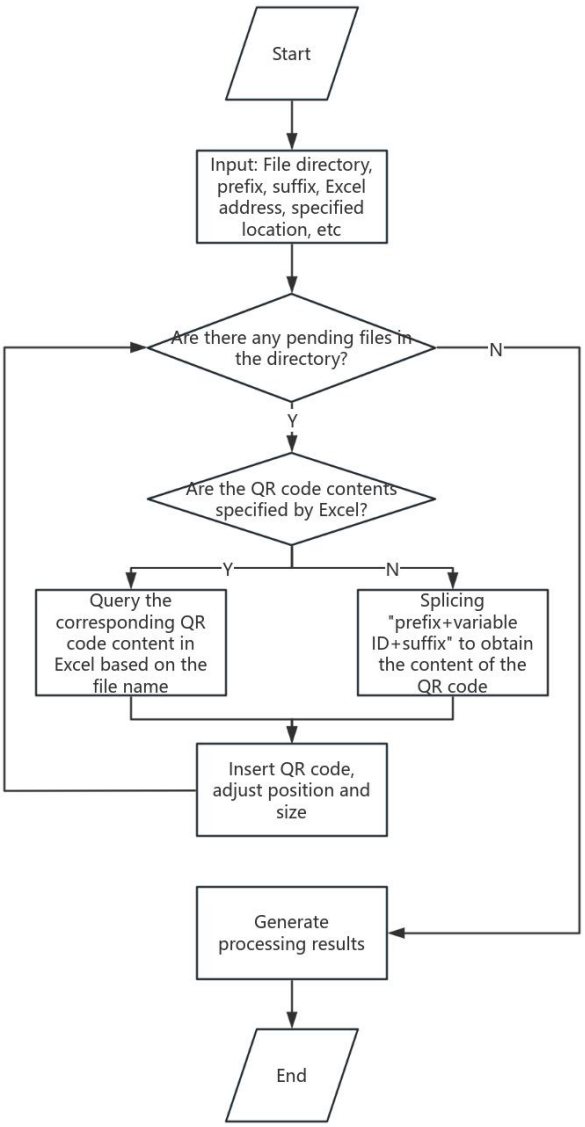


Figure 4. Design of generation/insertion process. QR, quick response.

code can be specified by the users themselves, including full-text links for papers, digital object unique identifier (DOI) links, video links, image links, WeChat public

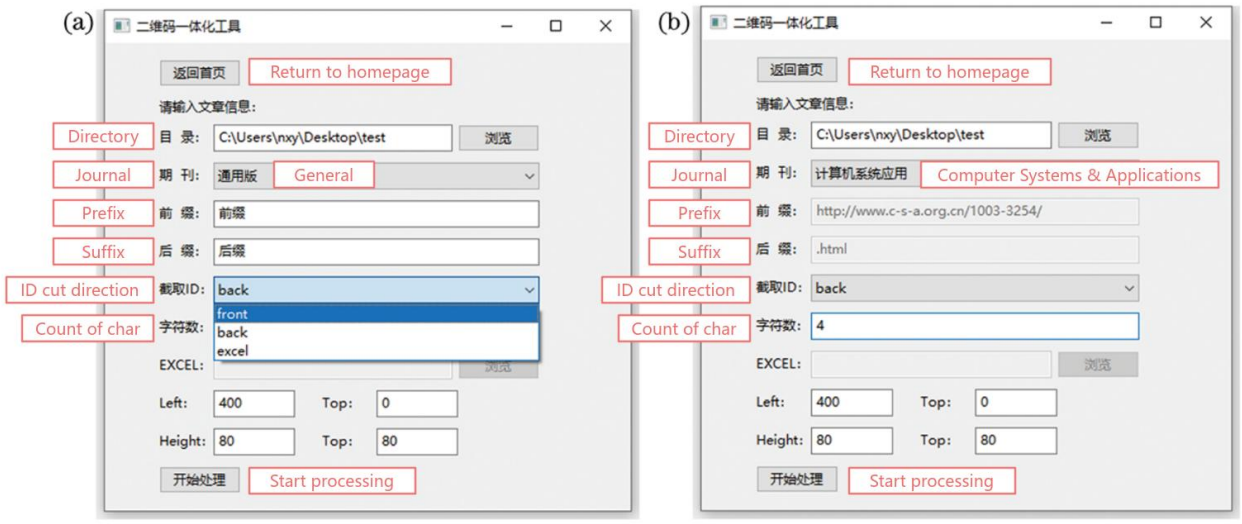


Figure 5. Interface for generating/inserting QR codes in different modes. (a) General version mode. (b) Personalized mode. QR, quick response; ID, identification.

	A	B
1	**File name**	**Content**
2	0001.doc	https://dx.doi.org/10.15888/j.cnki.csa.000001
3	0002.docx	https://dx.doi.org/10.15888/j.cnki.csa.000002
4	0003.doc	https://dx.doi.org/10.15888/j.cnki.csa.000003
5	0004.docx	https://dx.doi.org/10.15888/j.cnki.csa.000004

Figure 6. Excel file structure indicating the correspondence between file name and ID. ID, identification.

accounts, and texts. The tool reads the Excel file according to the file name to query the corresponding content and generate a QR code. This method is suitable for situations in which there is no correlation between the content contained in the QR code and the file name.

In practical editing, the naming of paper documents usually has some meaning and is often related to their QR codes. Therefore, two settings were added to the integrated tool for QR codes: front and back. The front setting refers to cutting off N characters from the beginning of a file name, while the back setting refers to cutting off N characters from the end of it.

If a QR code based on the file name can be directly generated, an Excel table will not be used. For example, in *Computer Systems & Applications*, the file name is the acceptance number of each article and is also part of the address on the first page of a paper (Figure 7).

The QR code link can be assembled according to the file name. When the content corresponding to the QR code meets the format of prefix + variable ID + suffix, and the variable ID can be extracted from the file name, then the prefix and suffix can be fixed, and a suitable method for extracting the variable ID is selected based on actual conditions. The integrated tool automatically assembles into one link, generates a QR code, and inserts it at the specified position in the corresponding document. This operation reduces the workload of making an Excel table while avoiding errors caused by manual input.

The size and position of an image are specified by four parameters (Figure 5). Left indicates the location of the image relative to the left edge of the canvas, Top indicates the location of the image relative to the top edge of the canvas, Height indicates the height of the image, and Width indicates the width of the image. The units for these four parameter tables are all pounds.

Personalized mode

The QR code-integrated tool can also design personalized patterns according to the characteristics of journals. For example, in *Computer Systems & Applications*, the prefix and suffix of the URL linked by the QR code

Table 1: Comparison of manual operations and one-stop solutions using 45 papers as examples

Method	Efficiency	Accuracy	Verification time
Manual	Generation/insertion takes 1.5 hours, and verification takes 0.5 hours, indicating large workload, and low efficiency	On average, there are 1-2 errors, indicating poor reliability	The website needs to be valid for verification, scheduled for the blue sample stage. If errors are found, the electronic version of the paper needs to be updated
One-stop solution	Generation/insertion takes 25 seconds, verification takes 5 seconds, indicating the high efficiency of automated operations	Zero errors, strong reliability	No need for a website to take effect, can be verified at any time, and can be verified multiple times, making the editing process more reasonable

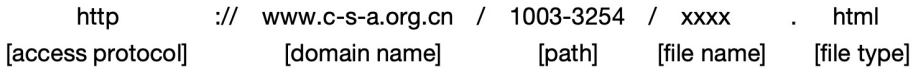


Figure 7. Schematic of the homepage website for the paper *Computer Systems & Applications.*

have fixed content, which has been configured in the program (Figure 5b). After entering the interface, users select a corresponding journal. They can use the built-in configuration, further reducing manual operation workflow, reducing the workload of editors, and avoiding errors introduced by manual input. At present, QR code integration tools support personalized modes for *Computer Systems & Applications* and *Journal of Software.*

After configuration is completed, users can click "Start Processing" to wait for the tool to automatically generate and recognize QR codes. As shown in Figure 8a, when all processing has been successful, a message prompt box pops up on the page, indicating that it is successfully processed. If there are failures during processing, as shown in Figure 8b, a message prompt box appears on the page indicating partial file failure, allowing users to find corresponding files and view the reasons for failure.

QR code verification

The QR code verification process is similar to that of generation/insertion. First, users place all files to be processed in a specific folder and specify this file directory in the tool. The files can be Word or PDF. Second, users select the verification mode, which supports both the general version and the personalized modes. This mode uses methods consistent with those used when generating QR codes. After completing the configuration, users click on the "Start Processing" button to wait for the tool to automatically handle the process.

After processing is completed, a message prompt box pops up to show the results of the process. If all QR codes are correct, the message indicates that QR code verification is successful; if some files contain incorrect QR codes, the file names of these files with errors are listed, and users can find the corresponding files for checking error causes and addressing them accordingly.

APPLICATION VALUE OF THE QR CODE-INTEGRATED SOLUTION

Comparison with traditional manual operation methods

The integrated solution based on Python QR code will assign a large amount of repetitive and cumbersome work to machine automation, significantly improving efficiency. Meanwhile, the use of software to recognize QR codes solves the problem of QR code verification and optimizes the editing and processing workflow. Compared with manual operation processes, the designed integrated solution for QR codes has advantages in terms of time cost and reliability. Taking 45 papers per issue of *Computer Systems & Applications* as an example, the comparison between integrated solutions and manual operation (Table 1).

Fully automatic operation with high efficiency

In the traditional manual operation process, editors need to manually generate QR codes individually and save them in a specified position of the designated file. Checking the QR code requires scanning each QR code, opening corresponding web pages, and confirming the codes, which involve substantial detailed work. In the integrated solution for QR codes, editors only need to open an integrated tool for QR code generation and insertion or check the QR code, select a specific folder, and complete these tasks at once. For example, *Computer Systems & Applications* has 45 articles per issue. The total time needed to automatically generate and insert QR codes when using this tool is about 25 seconds; checking QR codes takes about 5 seconds. If done manually, these processes would take 2 hours. After the integrated solution for the QR code was adopted by *Computer Systems & Applications*, the editor's working efficiency improved significantly.

Intelligent machine processing with strong reliability

Compared to the instability and unreliability of manual operation, software follows a certain logic for the automatic generation, insertion, and verification of QR codes, resulting in extremely low error probability. When initially conceptualizing this tool, the authors traced the effect of manually handling QR codes from recent issues of *Computer Systems & Applications*. The results showed that during the blue sample review stage, one to two errors were generally detected per issue; one issue had six errors because of improper Excel table processing at the QR code generation stage. After the use of the integrated solution for QR codes, the accuracy rate of generating/inserting QR codes reached 100%, meaning that all QR codes were correctly generated and accurately placed in the designated positions within the papers. The program also includes some options to avoid errors caused by manual input. The program saves processing logs during the process, and if an error is found in subsequent verification stages, it can be determined by backtracking on the processing log where the error comes from.

The process is more reasonable

A comparison of new and old processes shows that in the original manual operation process, there is a gap between editing and proofreading and QR code processing. By contrast, integrating QR codes solved this problem fully, integrating QR code processing with traditional editing and publishing processes. The verification of QR codes no longer depends on URL activation, which can be freely arranged during the editorial process, such as the second or third review. During the editorial process, once editors find problems with the QR codes, they can correct them immediately to avoid having to modify the QR codes again in the final stage of journal paper publication, further reducing correction costs. At the same time, the preverification of QR codes makes printing stages independent from online publication, making the editorial publishing process more reasonable.

Self-developed and extensible

The QR code-integrated tool described in this article was developed by the academic editors of *Computer Systems & Applications* according to their work experiences. It can be flexibly designed for personalized use. For example, it embeds two journals, *Computer Systems & Applications* and *Journal of Software*, which can automatically switch based on the editor's choice, simplifying the task of the editor as much as possible. At the same time, the tool does not limit the content of the QR code. Its general version has extensive functions that can be widely used in various forms of QR codes, such as official journal websites, paper homepages, video and picture links, and texts. Effective processing methods can flexibly address the problem of integrating a single paper with a QR code.

Comparison with existing solutions

Currently, some editors have been actively exploring the automated processing of QR codes.[10–12] This study examined and compared the advantages and disadvantages of this method with current methods (Table 2).

Existing solutions focus mostly on enabling the batch generation of QR codes. The integrated solution based on Python not only realizes the batch generation of QR codes but also achieves full automation processing for inserting and verifying QR codes, significantly reducing repetitive work in editing. At the same time, the automatic verification of QR codes frees them from being dependent on URL activation, resulting in the high accuracy and reliability of results and making the process of QR code editing reasonable.

CONCLUSION

In view of the low efficiency, error prone and difficult verification in QR code processing for science and technology journals, this study proposes a one-stop solution based on Python to automatically generate, insert, and verify QR codes. After experimental verification and trial use by two journals, *Computer Systems & Applications* and *Journal of Software*, it was found that the integrated tool significantly improves work efficiency, frees up manpower, enhances the reliability of QR code processing results, and perfectly integrates the QR code processing flow into traditional editing and proofreading processes.

In the future, the authors will explore two aspects. First, the QR codes generated by current tools support only black and white colors and do not allow free design, such as the addition of colored QR codes or the insertion of logo images into the codes. According to the needs of journal development, advanced QR code design functions will be added gradually. Second, this tool is currently being used in *Computer Systems & Applications* and *Journal of Software*, and it is expected that more features will be integrated into the tool and promoted to other journals in the future.

With the continuous progress of science and technology, editing and proofreading work has also moved toward digitalization. However, most tedious clerical tasks are still in the stage of manual processing, which has low automation levels. Computerized operations have high efficiency and reliability when dealing with repetitive work. With the development of various programming languages, rich integrated development tools and third-

Table 2: Comparison between the method proposed and in references 10-12

Methods	Solutions	Advantages	Limitations
[10]	ZZY.QR	Comprehensive personalized design function of QR code	(1) Batch coding content needs to be copied and pasted one by one (2) The insertion and verification of QR codes require manual operation
[11]	CaoLiao	(1) The personalized design function of QR code is comprehensive (2) The encoded content can be uploaded in batches	The insertion and verification of QR codes require manual operation
[12]	RAYS	Integration of QR code processing system and editing management system	(1) The coding content is limited to the RAYS Q&A system only (2) The insertion and verification of QR codes require manual operation
Proposed	One-stop solution	(1) The generation and insertion of QR codes are integrated, and verification is automated, making operation convenient and fast (2) The coding content can be defined independently and has a high degree of freedom (3) Reduce manual operations, achieve high accuracy and reliability in processing results (4) Independently developed, supporting personalized design	Only supports regular black and white QR codes

QR, quick response.

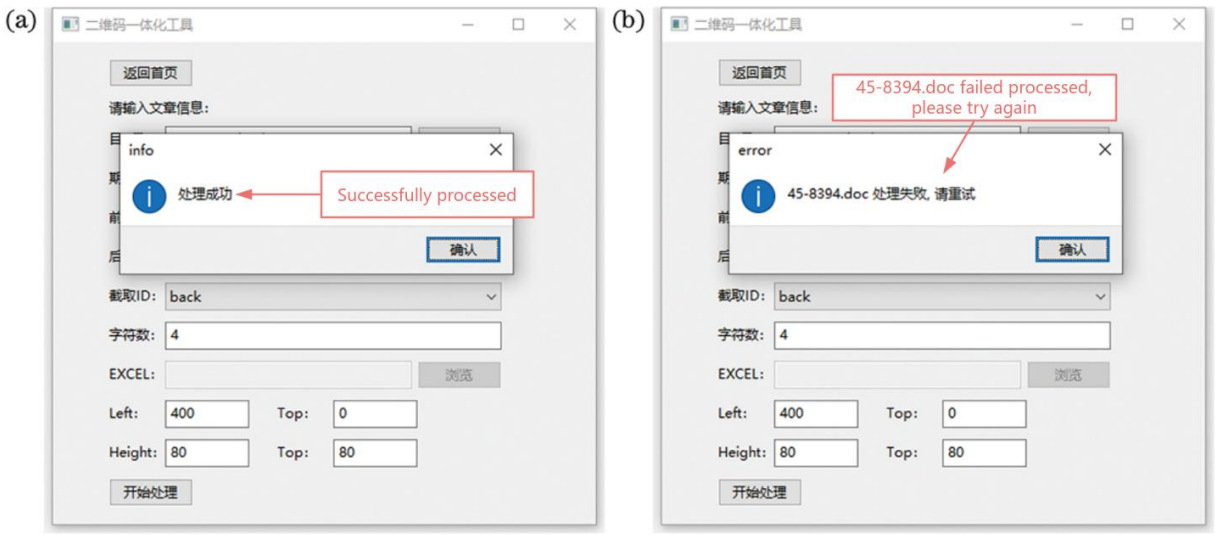


Figure 8. Generate/Insert processing example. (a) Successfully processed. (b) Processing failed.

party extension libraries reduce the threshold for tool development, making it possible for individuals to build lightweight editorial workflow modules. Scientific journal editors should give serious thought to technologies and continuously improve their work efficiency using computer technologies that perfect the work process.

DECLARATIONS

Secondary publication declaration

This article was translated with permission from the Chinese language version first published by the *Chinese Journal of Scientific and Technical Periodicals*. The original publication is detailed as: Niu XY, Qi LJ, Fang M. [An integrated solution for generating/inserting/verifying two-dimensional codes in scientific journals based on Python]. *Chin J Sci Tech Period.* 2022;33(8):1057-1064.

Acknowledgement

None.

Author contributions

Niu XY: Conceptualization, Writing—Original draft. Qi LJ: Methodology, Writing—Review and Editing. Fang M: Writing—Review and Editing. All authors have read and approved the final version of the manuscript.

Ethics approval
Not applicable.

Source of funding
This work was supported by National Natural Science Foundation of China (62042203).

Conflict of interest
The authors declare no competing interest.

Data availability statement
No additional data.

REFERENCES

1. Zhang HD, Sun JH, Bai J. [Application Study of Two-dimension Code in Paper Sci-tech Journals]. *J Libr Inf Sci Agric.* 2015;27(10):138-141.
2. Yan WN, Liu MY. [Visualized publication path of science journals in China under the perspective of media integration]. *China Publ J.* 2019;(11):37-41.
3. Jing Y, Zhang Y, Qin Y, Yin SY. [Connotation and strategy of using two-dimensional code in scientific journals under the background of media convergence]. *Acta Editol.* 2020;32(1):30-33.
4. Zheng XH, Tan J, He L. [Applications of new information and new technology in traditional sci-tech journals]. *Acta Editol.* 2018;30(3):304-306.
5. Zhang R, Yan QT, Wang XH, Xu LJ. [Editorial planning of the "QR code" linking content of academic journal]. *Acta Editol.* 2017;29(S2):S66-S67.
6. Tan X, Gao C, Zou CS, Duan CB, Yu PL. [Application examples of two-dimensional code in scientific journals]. *Chin J Sci Tech Period.* 2018;29(1):48-54.
7. Yang YX. [Analysis of current situation of two-dimensional codes integrated into a single paper and suggestions for its standardized application: A case study of scientific journals sponsored by Chinese universities]. *Chin J Sci Tech Period.* 2020;31(8):904-908.
8. Zhang MM, Tan BY, Lei Y. [Descriptive entries of single paper of Chinese scientific journals need to be perfected: Based on the investigation of medical and health core journals]. *Chin J Sci Tech Period.* 2018;29(12):1267-1273.
9. Wang HY. [A Brief Analysis of Problems and Countermeasures in Application of QR Code to Book Publishing]. *Media Sci Technol China.* 2021;(7):62-64.
10. Guo XL, Ji HT, Guo YM. [Batch encoding of QR codes for scientific journals and their applications]. *Chin J Sci Tech Period.* 2014;25(10):1267-1270.
11. Zhang HD, Sun JH, Lin HM. [Editing and application of two-dimensional code for scientific journals]. *Acta Editol.* 2015;27(6):552-554.
12. Zeng XR. [Application and analysis of QR code in integration innovation of science and technology journals: Taking *Computer Engineering* as an example]. Media. 2018;(5):37-39.
13. Gong ZQ, Zhu SC, Bi YH, Yang ZK, Dai YL, Wu YG. [Promotion of editing and proofreading quality and disseminating quality of scientific journals based on two-dimensional code]. *Chin J Sci Tech Period.* 2019;30(9):983-988.
14. The Python Tutorial. Python. Accessed November 17, 2021. https://docs.python.org/3/tutorial/index.html
15. Zhou L, Zhang YY, Zeng YL, *et al.* [Practice and thinking of realizing editorial work integration based on Python programming]. *Chin J Sci Tech Period.* 2021;32(3):403-410.
16. PyCharm: the Python IDE for professional developers by JetBrains. Accessed January 22, 2022. https://www.jetbrains.com/pycharm/
17. Python Package Index. PyPI. Accessed January 22, 2022. https://pypi.org/