Information Processing and Management 59 (2022) 103101

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Library on-shelf book segmentation and recognition based on deep visual features

Shuo Zhou [a,b,1], Tan Sun [a,b,1], Xue Xia [a,b], Ning Zhang [a,b], Bo Huang [c], Guojian Xian [a,b], Xiujuan Chai [a,b,*]

[a] Agricultural Information Institute, Chinese Academy of Agricultural Sciences, 100081, Beijing, China
[b] Key Laboratory of Agricultural Big Data, Ministry of Agriculture and Rural Affairs, 100081, Beijing, China
[c] School of Ocean Engineering, Harbin Institute of Technology, Weihai, 264200, Weihai, China

ARTICLE INFO

Keywords:
Library information management
On-shelf book recognition
Book spine segmentation
Deep learning
Library robot

ABSTRACT

On-shelf book segmentation and recognition are crucial steps in library inventory management and daily operation. In this paper, a detailed investigation of related work is conducted. RFID and barcode-based solutions suffer from expensive hardware facilities and long-term maintenance. Digital Image processing and OCR techniques are flawed due to a lack of accuracy and robustness. On this basis, we propose a visual and non-character system utilizing deep learning methods to accomplish on-shelf book segmentation and recognition tasks. Firstly, book spine masks are extracted from the image of on-shelf books by instance segmentation model, followed by affine transformation to rectangle images. Secondly, a spine feature encoder is trained to learn the deep visual features of spine images. Finally, the book inventory search space is constructed and the similarity metric between spine visual representations is calculated to recognize the target book identity. To train the models we collect high-resolution datasets of 10k-level and develop a data annotation software accordingly. For validation, we design simulated scenarios of recognizing 3.6k IDs from 5.6k book spines and achieve a best top1 accuracy of 99.18% and top5 accuracy of 99.91%. Furthermore, we develop a prototype of a mobile library management robot with embedded edge intelligence. It can automatically perform on-shelf book image capturing, spine segmentation and recognition, and target book grasping workflow.

1. Introduction

The library has always been the storage and retrieval place of knowledge and information. The management of its bibliography retrieval and book location is its most basic and important function. After the manual labor phase of relying entirely on librarians to retrieve and organize books, modern libraries widely use advanced electronic technologies. With the development of integrated-circuit, computer and database science, the popularization of barcode (Jampour, KarimiSardar, & Estakhroyeh, 2021; McCarthy & Wilson, 2011), RFID (Cheng, Huang, Xu, Hu, & Wang, 2016; Chu, 2015; Coyle, 2005; Mohammed et al., 2019; Ramkumar, Karthikeyan, Rajkumar, Venkatesh, & Praveen, 2020; Zhang, Zhang, & Wu, 2018), Bluetooth (Lee, Kim, & Jeong, 2014) and even WiFi (Bogdándy, 2018) techniques have gradually saved librarians from the primitive workflow of hand-writing and eye-seeking. Library management has entered the era of typing and searching. These technologies realize book recognition normally by firstly

* Corresponding author at: Agricultural Information Institute, Chinese Academy of Agricultural Sciences, 100081, Beijing, China.
E-mail address: chaixiujuan@caas.cn (X. Chai).
[1] Equal contribution.

https://doi.org/10.1016/j.ipm.2022.103101
Received 26 May 2022; Received in revised form 30 August 2022; Accepted 21 September 2022
Available online 12 October 2022

attaching and installing some customized micro hardware to papers and books and then specialized equipment is used to sense and read target book information. This progress has partially reduced the manual labor intensity, but there are still many drawbacks: tedious human work is still unavoidable, the entire system is highly complicated, and its construction and maintenance costs a lot, not to mention the potential cyber security risks. Besides, the above-mentioned non-contact book recognition solutions focus on the identity of a single book. The precise positioning and recognition of densely stacked books on bookshelves remain a major challenge.

To avoid the disadvantage caused by the natural limitation of hardware, research on software-based methods has been extensively carried out. With the help of digital image processing algorithms, low-level visual features of the target image can be obtained using cheap cameras. For the scenario of books on the shelf, many algorithms based on the principles of hue color statistics and line detection have been implemented to find the region of the book's spine on the image, followed by detection and recognition of characters in this region of interest (ROI). Image processing methods demand strict requirements for photometric condition and imaging quality, hence inevitably involve complex pre-and-post processing steps and hand-crafted features design steps, which will lead to inaccuracy, instability and limited application scope of the algorithm.

Naturally, after obtaining information about the ROI where a single book spine is located in an image of dense books, the long-standing spine recognition solution relies on optical character recognition (OCR). The most intuitive way is to directly recognize the text in the book spine region to obtain the title of the target book. While the instability of the OCR algorithm can cause character errors, omissions and redundancy issues, or generate meaningless symbols that make it infeasible to run title retrieval. Another idea is to recognize the book call number from the sticker attached at the bottom of the spine. This method takes advantage of the relatively fixed position of the label and the structured number layout. However, it requires a complete view and accurate recognition of the small call number, which is not always the case. Call number-based ways are obstructed due to problems of occlusion and the variable orientation of the sticker pasting.

With the rise of deep learning, artificial intelligence studies are fueled and the science of computer vision has been developed by leaps and bounds. The deep learning method based on convolutional neural network (CNN) is very superior in solving several key vision problems, such as object classification, detection and segmentation. On certain datasets, CNNs have outperformed humans. It is now the consensus that the trained CNN can extract high-level visual features from the target image end-to-end. For certain types of tasks such as object recognition and image retrieval, it achieves better performance than all traditional image processing methods and most other model-based methods. Deep learning provides new ideas and possibilities for solving various visual problems in many fields.

In recent years, with the rapidly evolving of artificial intelligence and robotics field, a vital component of a smart library is automation and intelligence. A mature system for recognizing books on shelves is indispensable. In this paper, We reform old ideas of digital image processing and deprecate the character recognition problem that traditional book recognition task has to solve. We believe that not only characters or call numbers on the spine region can be used as key information for recognition, but also the color, texture and artistic design of the entire spine area within the visible range can be utilized as clues to recognize a book. Based on this notion we exploit the powerful visual representation capabilities of deep learning models and innovatively propose solution that is superior in the aspect of cost, accuracy and applicability. To the best of our knowledge, this paper is the first to use deep learning methods to solve the problem of on-shelf book segmentation and recognition. And there are few robot designs for the library-like scenarios.

The main contributions of this paper can be summarized as follows:

- We propose an innovative deep learning paradigm where the CNN model of instance segmentation is used to segment book spines in on-shelf book images.
- A visual feature encoder is obtained to extract the deep visual features of book spine images, through which similarity metric between spines can be derived.
- To train the deep learning models, 10k-instance datasets of segmentation and recognition are collected and annotated. A book spine class annotation software is developed.
- We develop a prototype of the mobile library robot that can automatically perform the workflow of capturing image, book segmentation, spine recognition and book grasping.

The rest of the article is structured as follows: Section 2 starts with related work about different solutions to book segmentation and recognition. Section 3 summarizes the research objectives of this paper. Section 4 elaborates the novel methods we propose and the datasets collected for on-shelf book segmentation and recognition. Details of gallery&probe construction, performance comparison, robot design and experimental results are discussed in Section 5. In Section 6 discussion and outlook are proceed and we conclude the article with Section 7.

2. Related work

Along the development path of library automating and intelligentizing, the whole life cycle tracking, monitoring and analyzing of books are the most important components. Books are stored and displayed on the shelves for the longest time, so the management of on-shelf books determines the efficiency of a library's function. Critical steps include single spine location of densely-arranged books, recognition of a single book, books loading on/off, inventory daily checking and sorting. Among them, book segmentation and recognition are the pre-task for all following operations. There have been many studies on book-related segmentation and recognition as well as the smart library.

2.1. ROI segmentation

A large part of related research in this field involves the localization and segmentation of book regions, call numbers and characters. Hu, Zhou, and Ye (2005) compute pixels color gradient in local windows and their difference and position relation are used to locate the call number region. To solve the problem that RFID cannot accurately locate the position of dense books on bookshelves, literature (Ng et al., 2011) adopts a shallow neural network which trained with localized generalization loss to improve the image matching accuracy. Zhu and Yang (2011) apply pixel-wise domain connection to segment Chinese characters in ancient books. For call number extraction on the book spine label, Duan, Zhao, and Anwar (2012) analyze canny edge detector and hough transform's defect and adopt contour clustering according to characters' geometrical features. To overcome segment difficulties caused by book orientations under various viewpoints and proximity, literature (Talker, Moses, & Ieee, 2014) formulates the candidates' segmentation as a 5-parameter minimization problem and a graph-based method is used to filter the candidates. Yu, Zhang, and Cheng (2015) propose a two-phase framework including hough transform and SVM recovery scheme to segment book spines. Like many other studies (Fang, Zhao, & Du, 2014; Hu, Tang, & Lei, 2016; Nevetha & Baskar, 2015; Tsai, Shou, Hsieh, & Chang, 2018), the traditional ROI location and segmentation methods heavily depend on digital image preprocessing and require multiple pre and post processes. Besides, almost all the mentioned works ultimately rely on OCR engines to recognize characters in ROI to finally complete the task of book recognition. There are also many studies aimed to optimize OCR methods. A statistical learning model for post-processing OCR errors is presented in Mei, Islam, Moh'd, Wu, and Milios (2018). OCR errors with no training data are studied to improve retrieval performance in Ghosh, Chakraborty, Parui, and Majumder (2016) and machine learning models are incorporated in some modern OCR engines (Cao, Liu, Dong, & Yang, 2019; Jampour et al., 2021; Nevetha & Baskar, 2015; Rigaud, Burie, & Ogier, 2017).

Furthermore, the segmentation of book is also applied in many other publishing, library and document fields. Zhou and Liu (2009) uses a page frame segmentation algorithm to insert the selected contextual ads in print-on-demand. The work (Panichkriangkrai, Li, & Hachimura, 2013) develops an interactive system with GUI which supports text line and character extraction, to analyze woodblock-printed book images. Dutta, Biswas, and Das (2021) present a shape-aware CNN to segment different kinds of text boxes and speech balloons on comic document images. In Hu, Wang, Li, and Wang (2021), ancient Tibetan books text line segmentation is realized through a complicated process, where a CNN is used to classify two different degrees of adhesion. Zhang, Cai, Jiang, and Wang (2017) use Harris corner detection to solve the problem of leaf extraction of ancient Chinese books.

2.2. Book recognition

Framework and system for book recognition have been designed in several closest related works. Quoc and Choi (2009) propose a 3-stage workflow combining various tricks. High frequency filtering is used to extract book regions, canny edge detector and m-estimator sample consensus algorithm are used to separate a single book, plus another 4 steps to locate the title and characters. Chen, Tsai, Girod, Hsu, Kim, and Singh (2010) begin with a canny edge map and conduct multiple operations to extract the book spine. Then several low-level features are formed into a bag of features, followed by geometry and pose calculation to recognize the target. Together with mobile phone GPS and WiFi function, they propose a system to build book inventories using a smartphone. After this work, Tsai et al. (2011a) propose a hybrid approach to recognize book spines using a mobile phone. A text-based recognition pipeline outputs keywords to search a text database, while a feature-based pipeline matches the query book to an image database. Two sets of candidates are linearly combined to form the final recognition result. A phone-server system is proposed in Fowers and Lee (2012) to present an enhanced SIFT descriptor rather than general grayscale feature detectors to process book spine images. Xiu and Baird (2012) propose an adaptation policy where two models correct each other. Cross entropy between iconic and linguistic models is detected to yield higher whole-book recognition accuracy. Ul Ekram, Chaudhary, Yadav, Khanal, and Aslan (2017) explore the integration of barcode and the widely-used OCR engine Tesseract to support book sorting in the library. In general, these systems (Cao et al., 2019; Ul Ekram et al., 2017; Zurek et al., 2013) have identical workflow: ROI region is firstly located and character-level recognition is followed.

2.3. Model based methods

In addition to digital image processing ways, training and model-based approaches have also been investigated. Deep learning methods and deep neural network models (Dong, Wang, & Abbas, 2021), which have developed rapidly in recent years, are increasingly being applied to book and text recognition problems. Zhu, Yang, Wu, and Guo (2015) compare the performance of shallow machine learning SVM and deep learning CNN for the book classification task. An 8-layer simple CNN is adopted in the experiment and a 10-class book cover dataset is constructed to train and test the models. Better recognition results show the superiority of CNN. Several works (Bing, Tomiyama, & Meng, 2020; Lyu, Akama, Tomiyama, & Meng, 2019; Sichao & Miwa, 2020) study deep learning methods to segment and recognize the text line and characters of early Japanese books and cursive Japanese. Wang, Tang, and Lei (2018) extract label characters and then design a 4-layer CNN to classify the text. Shi, Tang, and Lu (2021) use deep learning-based OCR to recognize the barcode characters in the book cover on a conveyor belt. Soheili, Yousefi, Kabir, and Stricker (2017) combine sub-word clustering and LSTM neural network results to reduce character recognition error rate. A capsule network (Li et al., 2018) is proposed to complete linguistic word segmentation in ancient Chinese medical books. Yang et al. (2017) adopt CNN and recurrent neural nets (RNN) with CTC loss function for scene text reading. Although CNN's superior ability to extract image features has been discovered, these methods are still limited in the issue of character recognition. Actually, OCR-like solutions take the problem of book recognition as book title recognition, which discards other visual information in the spine region.

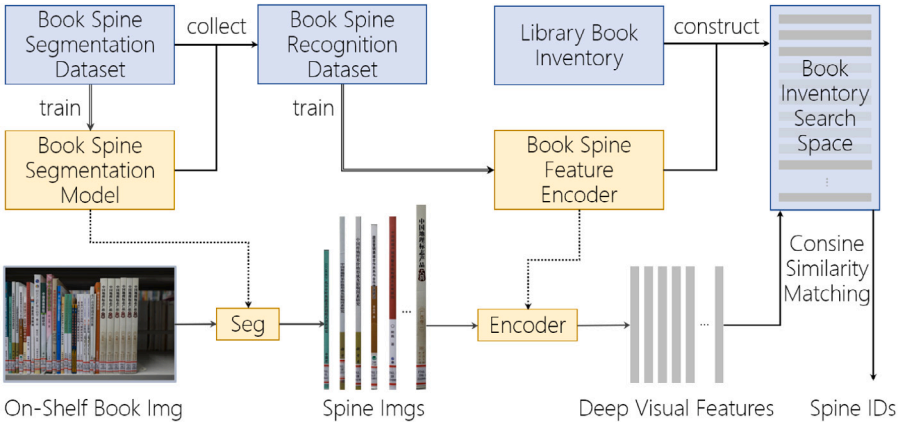

Fig. 1. System workflow.

2.4. Smart library and robot

Thanks to the above studies and modern technologies, the library is evolving to get smarter. Enjarini and Graser (2014) invent a wheelchair-mounted robot FRIEND with a stereo & ToF camera and a 7 DoF arm, who can grasp a target book from a book cart. Algorithms utilize both RGB and disparity information in the book segmentation process. In 100 runs, the FRIEND system completes the grasping tasks autonomously 84 times and with user assistance, it can succeed up to 99/100. Literature (Rodriguez-Osoria, Nuno-Maganda, Hernandez-Mier, & Torres-Huitzil, 2014) implements a book scanning system on a laptop and Raspberry Pi platform to segment and store images of the individual book pages. Animireddy, Singh, Neha, and Natarajan (2018) demonstrate a prototype of a robotic library assistant, who can move to predefined locations and performs barcode scanning to find the target book. Zhang and Cui (2017) apply nano-conductive ink printing and the ultra-high frequency RFID tag in the smart library. Ramkumar et al. (2020) build an android application to control the lights and fans using voice. The system updates information about borrowing and lending books. Jampour et al. (2021) incorporate Faster R-CNN in a shelf-reader robot to detect barcode region on bookshelves and multiple steps of post-process of the barcode are conducted to decode the ID of books.

3. Research objective

The research objective of this work is to address the three problems of on-shelf book recognition: (1) Existing book spine segmentation solutions require complicated pre-and-post steps of digital image processing, along with unreliable accuracy. (2) Almost all methods of book identification rely on character recognition to get the title or call number text and then retrieve them in the library search engine. Barcode-based methods are not robust because the sticker can be worn and blurred. Character-based ways ignore other useful information in the spine region and are limited by restricted dictionary. (3) Smart library demands a robot with edge intelligence that can automatically complete the whole workflow of book searching and management.

Existing software solutions to the issue of recognizing books on shelves in libraries are defective in various ways. They either rely on character segmentation and recognition or cannot handle on-shelf scenarios effectively, resulting in high cost, low accuracy and poor adaptability. In this paper, different from the previous strategy of character or call number segmentation and recognition, a novel book spine recognition method is proposed to realize the book identification. Concretely speaking, a deep learning paradigm is proposed to segment book spines end-to-end and extract deep visual features from the whole spine area, which greatly reduces the preconditions for system application and avoids the alteration of original library facilities. This system is cost-effective, easy to maintain, highly accurate and expandable.

4. Methodology

The overall workflow of our system is illustrated in Fig. 1. On-shelf book images are captured to construct the spine segmentation dataset and it is used to train the spine segmentation model. Then we collect and label the segmented spine images as the spine recognition dataset. The spine feature encoder is trained with the recognition dataset. In the deployment phase, the deep visual features of the library inventory spine images are extracted to build the inventory search space. In application, we calculate a spine's deep visual feature and measure its cosine similarity with inventory items to find the target ID. After registering the library collections as the search space, our system needs to be deployed only once and supports ultimately adding new books.

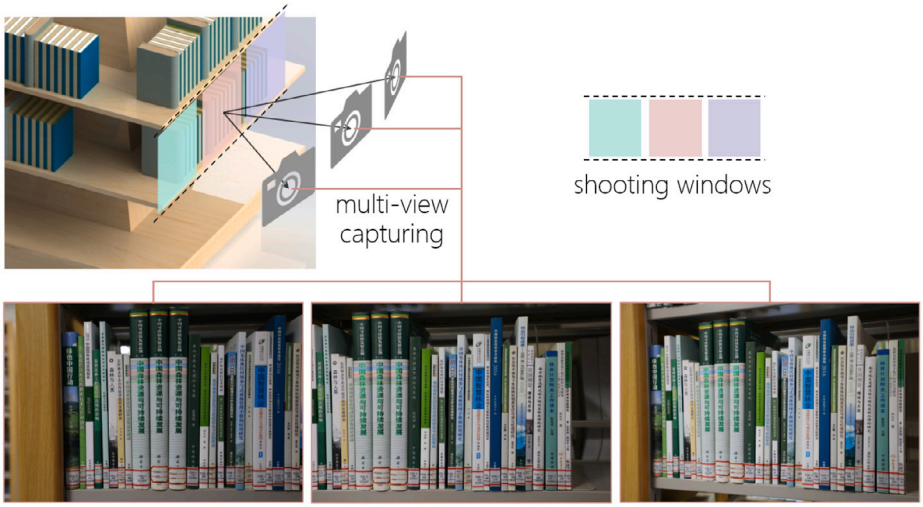

Fig. 2. On-shelf book images collection.

Table 1
Dataset construction overview.

Image number		Spine segmentation	Label format	Dataset
924	315	/ 6145 spines	Instance mask Identity	Book spine segmentation Book Spine recognition
	609	9232 spines	Identity	Gallery and probe

4.1. Dataset construction

We collect required data in real library scenarios. The National Agricultural Library of CAAS currently has a collection of more than 2.1 million volumes of Chinese and English literature and is a typical modern library with a grand scale of books, which makes it a suitable place to collect data. We use a household digital camera to capture RGB images of on-shelf books that are densely arranged in rows on the bookshelves. The upper and lower bookshelf clapboards of the target floor are roughly included in the camera's field of view to ensure all the spines of this row of books are captured. To increase the instance diversity of the dataset and get multiple images of the same book, we slide the shooting window with no overlap along the bookshelf row. For each shooting window, we move the camera to take multiple images with different view angles (Fig. 2). At this stage, we have collected a total of 924 usable on-shelf book images with 1920 × 1080 resolution. In the subsequent sections, book spine segmentation dataset, the spine recognition dataset and simulation experiments dataset are built based on these images like Table 1. Refer to the following sections for detailed explanations of the dataset using.

4.2. On-shelf book segmentation

Location of ROI is always the foremost step to recognize the target book. In this paper, segmentation of book spine is implemented by instance segmentation CNN model and affine transformation processing.

4.2.1. Book spine segmentation dataset

The instance segmentation task demands pixel-level labels. Open-source online annotation tool (Skalski, 2019) is used to complete the labeling mission of the dataset. From the collected 924 images, 315 images consisting of multi-view images sets at the same shooting window are selected and manually annotated. Four points are assigned to define a quadrilateral mask for each spine area in the image, as Fig. 3 shows. On average, each image contains more than 15 books i.e. spine region instances. We divide the 315 images into training set (297 images) and test set (18 images) for the segmentation task.

4.2.2. Book spine segmentation model

Extraction of spine regions from an on-shelf book image is a typical instance segmentation task in the machine vision field. It requires not only determining the location of different classes of targets in the image but also segmenting them pixel-wise from the background. Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017) is a milestone in the deep learning methods of solving instance segmentation problems. Mask R-CNN's backbone integrates ResNet (He, Zhang, Ren, & Sun, 2016) and FPN in which the design of skip connection and feature pyramid aggregation greatly improve the model's ability to represent deep visual

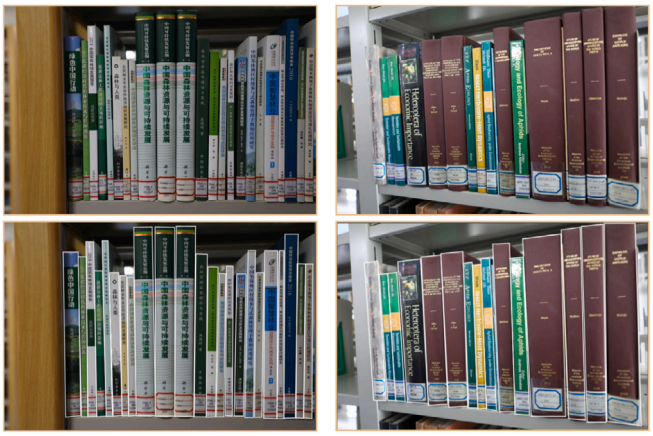

Fig. 3. Book spine mask label. Upper: original image; Lower: labeled image.

Table 2
Evaluation metrics of instance segmentation model.

mask_rcnn_R50_FPN_3x	bbox			segm		
	AP	AP50	AP75	AP	AP50	AP75
Benchmark	41.00	/	/	37.20	/	/
w/o pretrain	64.69	94.51	77.52	64.62	93.66	75.09
Pretrained	84.05	**98.87**	98.72	81.92	**98.87**	97.86

features. Its architecture is very mature and effective so is widely used. The book spine segmentation dataset is used to train a mask_rcnn_R50_FPN_3x model to end-to-end segment out book spines in on-shelf book images. In training, the learning rate is set to 2.5e−4 and batch size 128, max iteration 10,000 respectively. With or without pre-trained weights on COCO dataset (Lin et al., 2014), instance segmentation evaluation results on the 18 test images with 273 spine instances are shown in Table 2 shows. The AP refers to average precision and AP50, AP75 represent the AP value at IoU = 0.5, 0.75, where IoU (intersection over union) is the intersection degree between the predicted bounding-box/mask and the ground truth. Among these metrics, the AP50 is mostly concerned. With input image size being 1000 × 562 we can find that the AP50 of the model using pre-trained weights reaches 98.87, which performs far better than the one trained from scratch. The pre-training with the large amount of COCO data makes the segmentation model has more powerful feature extraction capability. This mode of using pre-trained models and fine-tuning them on custom datasets is a type of transfer learning and is widely applied in tasks with small datasets. The trained model takes an image as input and outputs region masks of all book spines, as Fig. 4 shows. It is robust to the position, thickness, height and orientation of books in the field of vision, and is also invariant to changes of imaging factors in the natural environment.

The predicated mask of a book spine is not a real quadrilateral, but actually an irregular long strip area. Considering that the books in vision have a variety of gestures, an affine transformation is applied to every mask to produce a regular rectangle book spine image. As Fig. 5 shows, the minimum bounding box of a mask is calculated, which is usually inclined in the image. Then the box's center point, rotation angle and four corner positions are used to rotate the original image until the box becomes a regular rectangle in the image. Finally, the rotated four coordinates can define the cropping area. Fig. 6 gives two examples of book spine extraction result after processing of the book spine segmentation and affine transformation.

4.3. Book spine recognition

Since spine images have been segmented from the on-shelf book image, an optional solution for spine recognition is the classification of spines. Object classification task in computer vision requires an image containing the target as input and predicting its category end-to-end. More precisely, spine classification will be a fine-grained classification task, because all objects to be classified belong to one category of the book spine and we need to divide these spines into finer categories. However, we do not solve the spine recognition problem totally following the object classification way. Because on the one hand, that will request a definite and known category range, which greatly limits the application of the system. On the other hand, it will be a major challenge to obtain enough data to well solve the problem of fine-grained classification. In this paper, the book recognition pipeline is basically querying by example way. Deep visual features of spine image content are extracted and their similarity distances are measured to judge whether two spine images belong to the same book or not. And to obtain the spine deep visual feature encoder, we set up a classification task to train a spine classification CNN model. The model's backbone is taken as the encoder which is expected to produce the deep visual feature of the input spine image.

Fig. 4. Spine instance segmentation results. Different colored masks represent different spine instances.

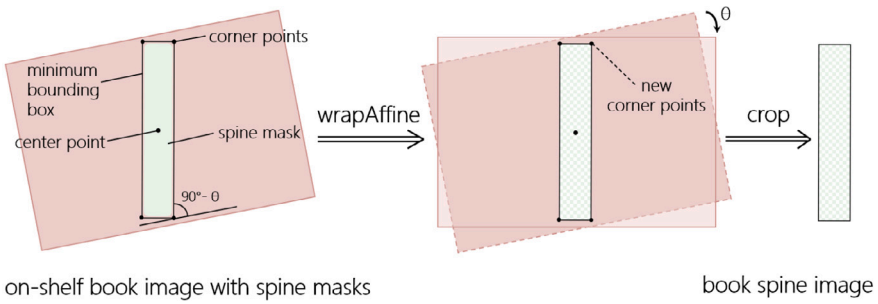

Fig. 5. Affine transformation.

4.3.1. Book spine recognition dataset

From the aforementioned 315 labeled images in the segmentation dataset, we totally collect 6145 spine images. To train a classification model, images with class labels are essential. In this paper, the class of a spine image is specific to the title of the book. For good classification performance, every ID category requires multiple images containing the target and here is the meaning of multi-angle shooting when collecting images. In the implementation, the segmented spine images are stored in a folder named after the original on-shelf image from where they come, and the multiple images at the same shooting window are adjacent in the datasets folder. Based on this, we developed a book spine class labeling software, as Fig. 7.

This executable software can display all the spines which segmented from a batch of images at the same shooting window simultaneously. Every spine instance is clickable. After creating a new category ID, just clicking the spine instance will remove it

Fig. 6. Segmented spine image results. Left: On-shelf image; Right: segmented spine images.

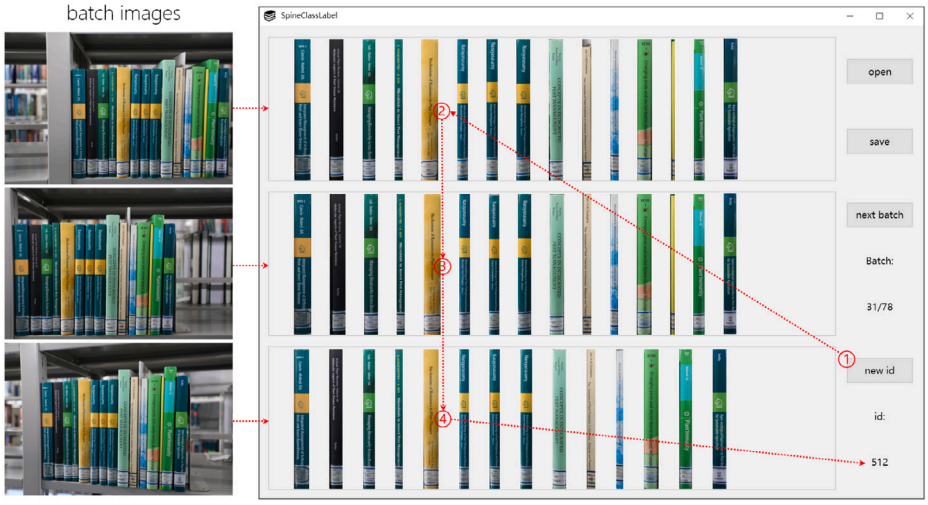

Fig. 7. Book spine class annotation software. For each new identity, user should click following the order as shown.

Fig. 8. Book spine recognition dataset augmentation. For every set of images, the first is original spine, and the remaining four images from left to right, are blur, bottom crop, horizontal flip, left crop, around crop.

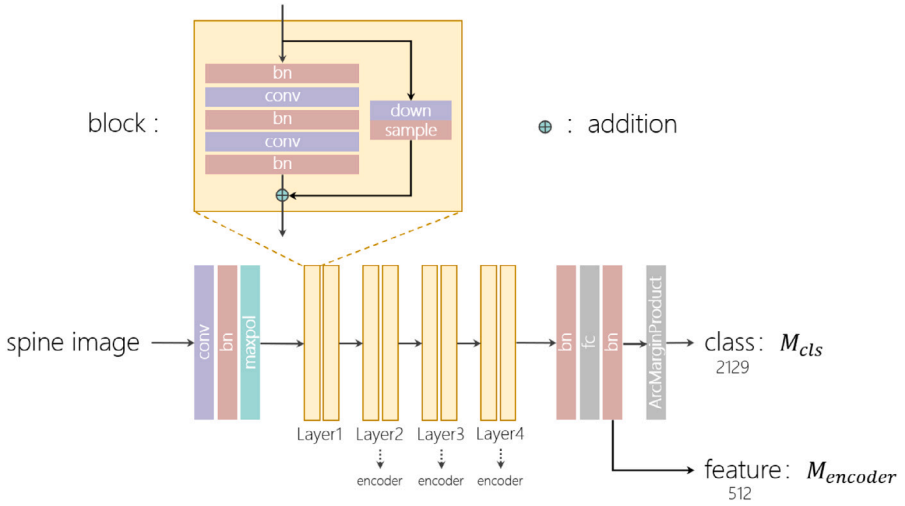

Fig. 9. Book spine classification model and deep visual feature encoder architecture.

Table 3
Single factor gain ablation study.

	With	Best_epoch	Val	Test	TOP1	TOP3	TOP5
Loss function	FLoss	20	51.250	52.656	0.9595	0.9803	0.9860
Online aug	HisEqu	58	51.562	49.844	0.9652	0.9833	0.9870
Optimizer	Ada	43	51.719	50.938	0.9638	0.9812	0.9876
Offline aug	DataAug	26	92.108	92.484	0.9853	0.9947	0.9970
CELoss+Sgd		18	42.656	43.438	0.9582	0.9801	0.9851

from the display area and bind the spine image to the current ID automatically. So the general process of labeling work is to create the new category ID and then click all the spines of the same title from the display areas. The shooting windows do not overlap with each other, and there are often multiple copies of the same book in the library. After labeling there may be multiple spine images under one ID folder. 2129 IDs are finally assigned and the classification task can be specified as 2129-class object classification task. Online data augmentation like histogram equalization (HisEqu) and offline data augmentation strategies (DataAug) (Fig. 8) including directional edge cropping, random horizontal flipping and blurring are carried out, and the final images number of recognition dataset reaches 40,000. Train, validation and test sets are split according to a ratio of 8:1:1.

4.3.2. Deep visual feature encoder

The architect of our classification model is as Fig. 9. The main pipeline design is inspired by the ideas from ResNet (He et al., 2016). ResNet is a milestone of deep learning methods in computer vision. The residual block and skip connection structure offer a good solution to the problem of gradient explosion, initiating the application of increased layer depth in deep learning models. As one of the classical model backbones, its design philosophy provides a benchmark for the vision mission. In our baseline model, stochastic gradient descent optimizer (Sgd) and an alternative activation function Prelu (He, Zhang, Ren, & Sun, 2015) are adopted, which can improve the model fitting ability and reduce the risk of over-fitting with almost no additional parameters. Except the basic loss function CrossEntropy loss (CELoss), We use additive angular margin (Deng, Guo, Xue, & Zafeiriou, 2019) loss (as Eq. (1)) to enhance the model's classification performance. The θ in the equation is the angle of the model weight and feature vector and the m and s represent the hyperparameter of angular margin. In this paper, m is 0.5 and s is 64. An arc-margin-product layer follows the last fully connected (FC) and batch normalization (BN) layers to predict the ID from the 2129 class.

$$Loss = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{i=1,i\neq y_i}^{n}e^{s\cos\theta_j}} \tag{1}$$

The book spine recognition dataset is used to train the CNN model end-to-end. In the process of training the model M_{cls}, images are all resized to 80 × 800 pixel, the batch size is assigned as 16 and the max training epoch number is 100. The accuracy of the classification refers to the classifying correct rate performed by the model on the test dataset, which evaluates the performance for spine category classification. The classification accuracy of the model is non-deterministic but positively correlated with the subsequent search performance, as detailed in the next subsection. we designed numerous experiments to determine the most effective optimization methods. The ablation studies explore the effects of multiple factors on model performance, including using online data augmentation of HisEqu, using offline DataAug integrating multiple processing as described in Subsection 4.3.1, using

Table 4
Single factor impairment ablation study.

	Without	Best_epoch	Val	Test	TOP1	TOP3	TOP5
Loss function	FLoss	39	95.570	95.550	0.9890	0.9964	0.9979
Online aug	HisEqu	59	94.284	94.403	0.9869	0.9959	0.9977
Optimizer	Ada	95	93.987	93.552	0.9876	0.9952	0.9979
Offline aug	DataAug	33	46.562	52.031	0.9622	0.9835	0.9886
FLoss+HisEqu+Ada+DataAug		88	96.786	**96.106**	**0.9918**	0.9991	0.9991

Fig. 10. Grad-cam visualization results of some book spines.

the adagrade Tables 3 and 4 shows. The baseline model uses CELoss and Sgd while the final architecture (FinalArc) uses FLoss, HisEqu, Ada and DataAug. Histogram equalization is a digital image processing method that increases the feature clarity of an image and can be used to enhance the local contrast without affecting the global contrast. The focal loss allows the model to focus more on the hard-to-classify samples during training by reducing the weights of the easy samples.

The val and test column record the trained model's classification accuracy on the validation and test dataset. Table 3 shows results of the single-factor gain study. In the first four rows, each row indicates that a single method is correspondingly replaced or added on the basis of the baseline model. Table 4 shows results of the single-factor impairment study, where optimization methods are separately excluded from the FinalArc. We can conclude that every aforementioned individual method offer gain in model M_{cls} classification performance, where offline data augmentation contributes the most. Overall, the final model uses histogram equalization, focal loss, adagrade and offline data augmentation. To further validate and understand the internal mechanisms of the trained CNN model, the grad-cam (Selvaraju et al., 2017) method is adopted. Grad-cam is one of the visualization solutions for interpreting CNN in classification tasks, and can roughly represent what regions of the input images have the greatest impact on the inference results of the model. We process several typical spines from Layer4 of the model. In Fig. 10, heatmaps are aligned to the resized image where red regions represent highly sensitive areas and blue ones represent the opposite. From (a) we can see that the model naturally pays most attention to the characters and their surrounding area on the spine of the book, (b) and (c) show that for the series of books, the model finds their discriminative region at the volume and series number position.

4.3.3. Spine features similarity matching

From the trained book spine classification model, the spine deep visual feature encoder $M_{encoder}$ can be obtained by excluding the last arc-margin-product layer from the FinalArc model. As Fig. 9 illustrates, the 512-dimension output vector of the last BN layer is the deep visual feature of the input spine image. We use the spine feature encoder $M_{encoder}$ to perform feature calculations on all the spine images in library collections. All feature vectors are saved into a single binary file as the book inventory search space i.e. the gallery database, as the next subsection states. Using the arcface loss, the learned embedding features are distributed on a hypersphere space. To ensure that the measurement of the prediction features is consistent with the nature of the model itself, the cosine similarity metric (Eq. (2)) is used to measure the distance between two features. In Eq. (2), F_a and F_b represent two spine features. During spine searching, the deep visual feature of the target spine are extracted in real-time, and the similarity between it and all the entries in the gallery database are calculated respectively. The one with the highest similarity is the final prediction

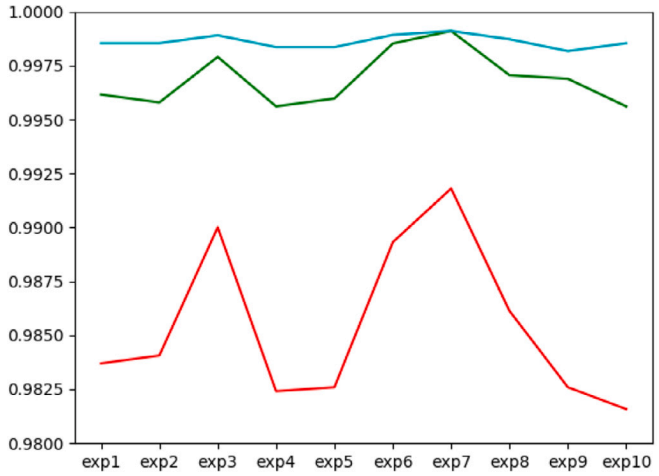

Fig. 11. The TOP 1, 3, 5 (in red, green and cyan color respectively) accuracy of 10 identification experiments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Recognition accuracy of encoder from different layers.

	TOP1	TOP3	TOP5
Layer2	0.9072	0.9470	0.9602
Layer3	0.9624	0.9842	0.9881
Layer4	0.9892	0.9968	0.9976
finalArc	0.9918	0.9991	0.9991

ID of the target.

$$similarity = \frac{F_a \cdot F_b}{\|F_a\| \|F_b\|} = \frac{\sum_{i=1}^{n} F_{ai} \times F_{bi}}{\sqrt{\sum_{i=1}^{n} (F_{ai})^2} \times \sqrt{\sum_{i=1}^{n} (F_{bi})^2}}, n = 512 \quad (2)$$

5. Experiments and results

In order to verify the effectiveness and superiority of the system, we simulate the on-shelf book segmentation and recognition scene of a real library. On-shelf book image is processed to output spine images and every spine image is queried in the book inventory search space.

5.1. Book recognition experiments

From the collected 924 images, we select 609 images that are not used for instance segmentation or classification model training to build the gallery and probe database. That is, in the recognition procedure the instance segmentation model and deep visual feature encoder have never processed the gallery or probe images before, which completely avoids the possibility of fake validation. The gallery simulates the library's collection of bibliography. It stores all the information about every book, including at least a standard spine image per item. While the probe simulates the segmented spines to be recognized. The accuracy is defined as how many spines in the probe can find the correct ID in the gallery. As described in previous sections, the book spine segmentation model extracts spine images from all the 609 on-shelf book images and 9232 book spines are totally collected. The spine class labeling software is used to assign them unique IDs manually and 3604 IDs are totally generated. The IDs are a series of numeric codes in this paper, which in real library will be associated with detailed information about books. There is at least one spine image under every ID folder, so we randomly take one image from each ID folder to build the gallery, and all the other images form the probe. For statistical reliability, we repeated the process of randomly sampling to build the gallery and probe dataset 10 times. For each sampling, we get 3604 IDs, i.e. 3604 spine images in the gallery and a total of 5628 images in the probe. During the system deployment phase, the deep visual feature encoder calculates all the 5628 probe features and they are saved into a gallery query file less than 7.5MB, which will be loaded into memory only once at system startup. In a practical application, for the new incoming books, a user needs simply to input their spine images to $M_{encoder}$ and update their deep visual features into the gallery file.

In our recognition experiments, the deep visual representation of a probe spine is extracted by the spine feature encoder $M_{encoder}$ and its cosine similarity with every item in the gallery is calculated to find the most similar spine ID. We output 5 gallery spine IDs with the highest similarity as candidates. The TOP 1, 3, 5 accuracy means the accuracy when the correct ID appears in the former 1, 3,

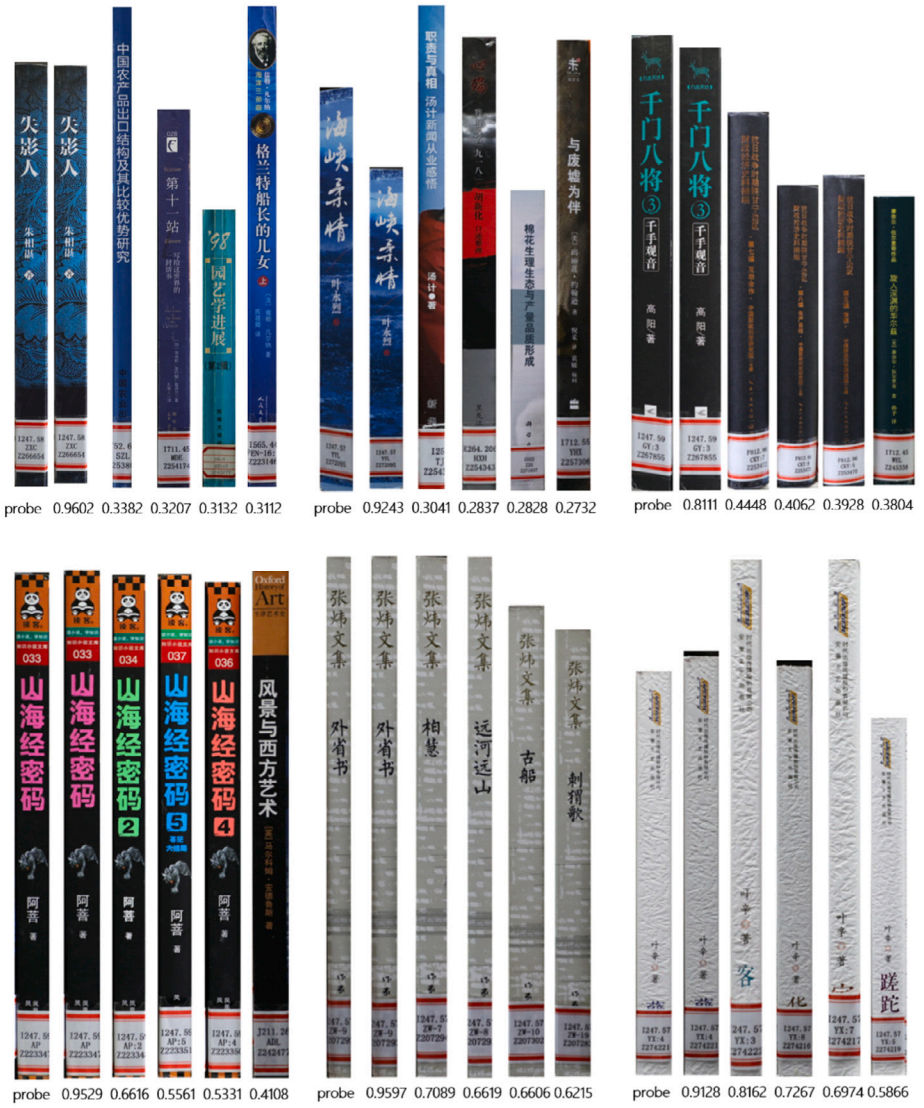

Fig. 12. Correct cases in recognition experiments. For every set of 6 images, *probe* is the spine to be recognized, the following 5 spines and bottom numbers represent the top 1–5 candidate spines in gallery database and their similarity with *probe*, respectively.

5 candidates. The accuracy results of 10 identification experiments are as Fig. 11. The main source of varying accuracy is the different quality of the spine image which is sampled to galley. A gallery image with distortion and poor illumination conditions will cause false identification of all probes of this id. In practical library scenarios, carefully constructed gallery datasets can greatly improve the accuracy of recognition. Statistically, our top1 accuracy is from 98.15% to the highest 99.18%, the mean and standard deviation are 98.54% and 0.35% respectively. The following experiments are carried out on the gallery and probe sampling results with the best identification performance. Given the positive correlation between M_{cls} and $M_{encoder}$ performance as Subsection Section 4.3.2 describes, extensive experiments to determine the best encoder are conducted as Tables 3 and 4. In the tables, the TOP 1, 3, 5 columns show the recognition accuracy of encoders from models using different optimization strategies and the best encoder comes from the FinalArc. The experiments' results show that the best top1 recognition accuracy of the probe reaches 99.18%, and the top5 accuracy reaches 99.91%. Several correct recognition examples are displayed in Fig. 12. We can see from the examples that the target spine is far more similar to the correct ID in the gallery than the other four candidates. Experiments are also carried out where visual features from different layers of the best encoder are taken as the spine encoding results, as Fig. 9 and Table 5 shows. We can conclude that as model depth increases, the visual representation of input image brings more accurate recognition. In further analysis, we show some wrong recognition cases as Fig. 13. Statistics draw conclusions that most of them are books from their publication series. Their visual features are very identical with only minor differences in the part number or series number region. From another perspective, the accuracy of the system is even higher if we take into account the fact that the books in the

Fig. 13. Error cases in recognition experiments. For every set of 7 images, *probe* is the spine to be recognized, *gallery* is its correct ID in gallery database, the middle 5 spines and bottom numbers represent the top 1–5 candidate spines from gallery database and their similarity with *probe*, respectively.

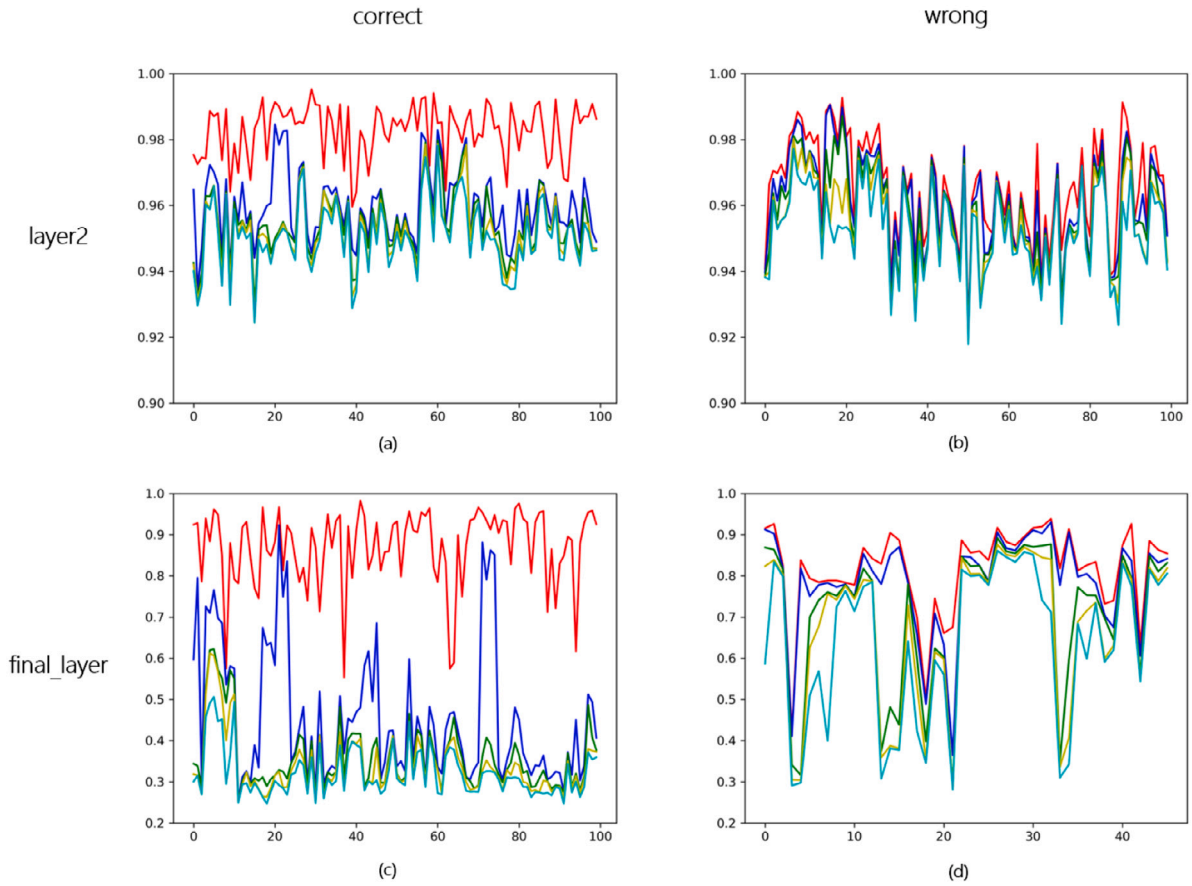

Fig. 14. Top 1–5 candidates' similarity values overview. (a) (b) are respectively correct and wrong recognition cases using encoder from layer2, (c) (d) from final_layer. Each value of the horizontal coordinate is a spine in probe. The vertical coordinate represents the similarity and the top 1–5 similarity values line are in red, blue, green, yellow and cyan color. Among them, (d) shows only 46 spine data, which is the total number of error cases. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

same publication series are always in the same location on a bookshelf. In addition, the top 1–5 similarity values overview of the correct vs. wrong cases and layer2 vs. final_layer is as Fig. 14 shows. In the horizontal comparison, we can find that for most correct cases, the top1 similarity value is far away from the other candidates. In the vertical comparison, we can summarize that the spine

Table 6
Performance comparison.

Method	Dataset scale	Pipeline	Metrics
Tsai et al. (2011b)	454 IDs2300 images	Text localization Text OCR + image matching	/ P 92%, R 60%, F 72%
Duan et al. (2012)	567 images	Call number localization Call number OCR	Recall 97.7% P 87.74%, R 89.59%, F 88.66%
Yang et al. (2017)	454 IDs9100 images	Text localization Text recognition	/ P 92%, R 90%, F 91%
Shi et al. (2021)	5666 images	Barcode localization Barcode decoding	Recall 99.17% Accuracy 64%–100%
Ours	3604 IDs5628 images	Spine segmentation Image matching	AP50 98.87% Recall 98.54%

Table 7
Spine segmentation speed at different platforms.

Resolution	bbox AP50	seg AP50	seg speed (s/img)		
			2070	AGX	TX2
1920 × 1080	96.76	96.76	0.225	1.296	8.721
1000 × 562	98.87	98.87	0.115	0.716	3.085
600 × 337	96.78	96.78	0.085	0.446	1.834

visual features is more discriminative as the layer depth increases. These patterns can also help us to evaluate the credibility of the recognition results based on the distribution of the top 5 values.

5.2. Performance comparison

As described in Section 2, there are a few similar works of visual book recognition focusing on the on-shelf scenario. To our knowledge, all previous approaches rely on character recognition. They evaluate the performance of the proposed methods mainly using the precision, recall and f-score of character recognition mission. And most of them either conduct experiments under specified conditions or have self-defined performance evaluation criteria in multiple processing stages. Despite this, we list some barely comparable related works in the Table 6 for the convenience of the readers. Observe that the literature (Yang et al., 2017) can barely serve as a historical state of the art and be compared in the aspect of book recognition performance. In this research, after the complicated processing of text localization and recognition, spine title retrieval performance is assessed, where recall is the percentage of correctly identified titles out of all query spines. 9100 spine images are recognized to 454 IDs and it has top1 recall of 0.90 and top5 of 0.964. In conclusion, our method outperforms older SOTA by a large margin in terms of simplicity, speed, and accuracy.

5.3. Library robot prototype

We also develop a prototype of an automatic library robot with our algorithms and system embedded. The key functions of this robot mainly include autonomous navigation and obstacle avoidance in library space, capturing the designated position of the bookshelves, segmentation and recognition of spines, and target book grasping. Its control logic is divided into two parts: vision processing and mechanical control, which cooperate with each other through real-time communication protocols. The robot is mainly composed of several parts: vision and mechanical central control, mobile chassis, visual perception camera, gripper system and transfer bookshelf, as Fig. 15 shows. To guide the action of the gripper, we use the Intel RealSense D435 as the vision information source. D435 is a depth camera supporting getting the 3D coordinate of the target pixel in the field of view. As for vision algorithms, the speed performance of the instance segmentation model on different platforms with different input image sizes is shown in the Table 7. The different resolutions of the input images bring different amounts of computation, which is ultimately reflected in time consumption. Meanwhile, the performance of the segmentation will be affected by input resolution to some extent. In experiments, PC graphics card Nvidia RTX 2070, AI computing platform Jetson AGX Xavier and Jetson TX2 are tested. AGX and TX2 are embedded AI computing devices with a wide range of standard hardware interfaces, which makes them suitable for both deep learning algorithm deployment and collaboration with smart hardware. From the experiment results, we can see that resolution 600 × 337 brings only a small degradation in segmentation performance while producing a great optimization in speed. For the prototype, image size 1000 × 562 is adopted for best results and AGX is selected as the preferred platform due to its outstanding computational performance and power efficiency. In the recognition step, the sequentially searching time consumption of a single spine is less than 560 ms at the current database scale, which depends on where the correct ID entry is located in the gallery file. The search speed bottleneck lies in the data architecture of the gallery and the search algorithm implementation, which are beyond the scope of this paper.

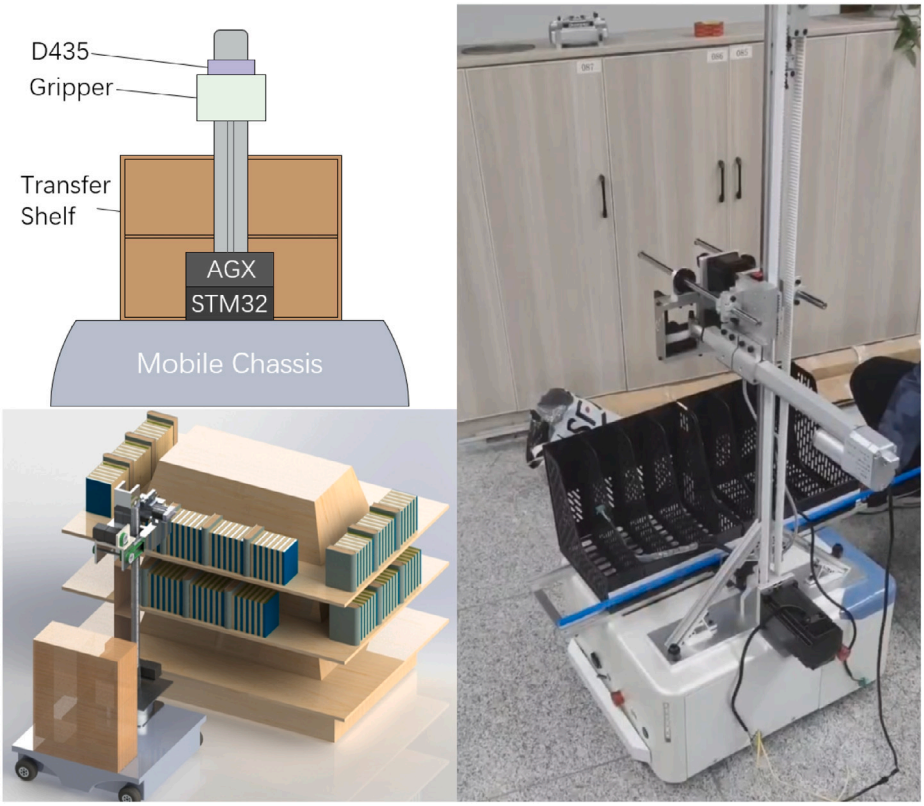

Fig. 15. Robot prototype.

Under the premise that the bookshelves and indoor facilities in the library remain unmoved, the mobile chassis with integrated navigation system will build a high-precision digital map of the library under human supervision. Combined with other environmental parameters such as the height of each floor of the bookshelf and the artificially designated functional areas in library, the robot can automatically plan a route to each work point and take images of on-shelf books in turn. The depth camera together with the gripper can move up and down along the vertical rail and captures the books straight ahead of view field after reaching an appropriate height. RGB and depth frames are captured and aligned, where RGB images is processed to get the spine region and ID. If needed, the gripping pixel position is determined based on the region information. Combined with the depth frame, the real 3D coordinates of the gripping position relative to the camera can be obtained. In AGX, spine segmentation and recognition can be completed within 1 s after image acquisition, with a premise that an on-shelf book image contains approximately 15 spines. The interested book can be gripped out of the bookshelf and placed in the transfer bookshelf, or the other way around.

6. Discussion

The previous article describes a library on-shelf book segmentation and recognition system and its deployment, which is excepted to automatically and high-precisely complete the work of on-shelf bibliography management. With improved accuracy and robustness, the proposed deep learning paradigm greatly simplifies the processing of on-shelf book image and completely decouples the dependence of recognition methods on library facilities such as barcode and call number sticker. For existing libraries being opened to the public, the application of the system will need to alter the existing collection database. Each entry needs to add an image of its spine and provide related data API. As for the spine registration, the spine segmentation function can be easily modified to get the spine image of a single book. A book registration device will be developed for the future application.

Despite the advantage of our methods, there are still parts that can be improved in future work. Spine segmentation algorithm is excepted to offer optional small models which are more suitable to run on edge computing devices and realize faster segmentation. The spine deep visual feature encoder can be optimized to better handle the fine-grained recognition of series books through collecting more data and model architecture innovation. Besides, the current system workflow can only complete the recognition task at book title level, not book instance level. The call number information of existing library books can be combined to recognize different book instance of the same title. When constructing the gallery file data structure, distance metrics can be used to evaluate and arrange the distribution of all spine visual features. Guiding the search path according to a n-ary tree way can optimize the speed when finding the target ID.

7. Conclusion

Smart libraries have higher and higher needs for automation and intelligence. With AI and robotics technology developing, traditional book inventory management and other human-intensive labor in library will and can be replaced by smart equipment. Facing the increasing demands of automation and intelligence for smart libraries, we have conducted a detailed investigation of related work in this field. In this paper, we propose a novel deep learning paradigm of on-shelf book segmentation and recognition based on convolutional neural networks. The proposed system is visual and non-character, which demands no specific appendage, language, art design, or corpus database, hence is economical and independent. The first 10k-level spine datasets are collected, including 315 on-shelf book images with spine instance segmentation label and 1.5w book spine images with ID label. Spine ID data annotation software is developed. The datasets and annotation software will be all open-sourced for the research community at https://github.com/surefyyq/DL-On-shelf-Books-Recognition. Extensive book recognition experiments verify the average accuracy up to 98.54%. The algorithm system is deployed on edge computing devices and a prototype of an automatic library robot that can accomplish the whole pipeline task is developed. This robot is excepted to significantly reduce the human labor and enhance the automation and intelligence level of libraries and other similar scenarios.

CRediT authorship contribution statement

Shuo Zhou: Conceptualization, Methodology, Software, Writing – original draft, Formal analysis. **Tan Sun:** Conceptualization, Supervision, Project administration. **Xue Xia:** Writing – review & editing, Visualization. **Ning Zhang:** Writing – review & editing, Validation. **Bo Huang:** Software, Hardware. **Guojian Xian:** Data curation, Resources. **Xiujuan Chai:** Project administration, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61976219; in part by the Science and Technology Innovation Program of the Chinese Academy of Agricultural Sciences under Grant CAAS-ASTIP-2016-AII.

References

Animireddy, S. P., Singh, K. P., Neha, & Natarajan, V. (2018). Robotic library assistant. In *Proceedings of the 2018 second international conference on inventive communication and computational technologies* (pp. 1443–1447).

Bing, L., Tomiyama, H., & Meng, L. (2020). Frame detection and text line segmentation for early Japanese books understanding. In *Icpram: Proceedings of the 9th international conference on pattern recognition applications and methods* (pp. 600–606). http://dx.doi.org/10.5220/0009179306000606.

Bogdándy, B. (2018). WiFi RSSI preprocessing library for android. In *2018 19th international carpathian control conference* (pp. 649–654). IEEE.

Cao, L. N., Liu, M. D., Dong, Z. Q., & Yang, H. (2019). Book spine recognition based on opencv and tesseract. In *2019 11th international conference on intelligent human-machine systems and cybernetics, vol. 1* (pp. 332–336). http://dx.doi.org/10.1109/Ihmsc.2019.0083.

Chen, D. M., Tsai, S. S., Girod, B., Hsu, C.-H., Kim, K.-H., & Singh, J. P. (2010). Building book inventories using smartphones. In *Proceedings of the 18th ACM international conference on Multimedia* (pp. 651–654).

Cheng, H., Huang, L., Xu, H., Hu, Y., & Wang, X. A. (2016). Design and implementation of library books search and management system using RFID technology. In *2016 international conference on intelligent networking and collaborative systems* (pp. 392–397). http://dx.doi.org/10.1109/INCoS.2016.35.

Chu, J. (2015). Applications of RFID technology [booksoftware reviews]. *IEEE Microwave Magazine*, *16*(6), 64–65. http://dx.doi.org/10.1109/MMM.2015.2419891.

Coyle, K. (2005). Management of RFID in libraries. *The Journal of Academic Librarianship*, *31*(5), 486–489.

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4690–4699).

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, Article 100379.

Duan, X. R., Zhao, Q. J., & Anwar, S. (2012). Identifying books in library using line segment detector and contour clustering. (pp. 998–1003). ISBN: 978-1-4673-0894-6 978-89-94364-26-1.

Dutta, A., Biswas, S., & Das, A. K. (2021). CNN-based segmentation of speech balloons and narrative text boxes from comic book page images. *International Journal on Document Analysis and Recognition*, *24*(1–2), 49–62. http://dx.doi.org/10.1007/s10032-021-00366-4.

Enjarini, B., & Graser, A. (2014). Color-depth-based book segmentation in library scenario for service robots. In *2014 Ieee international conference on autonomous robot systems and competitions* (pp. 229–234).

Fang, J. J., Zhao, Q. Q., & Du, M. F. (2014). Extraction and segmentation of books call number image for books on the shelves of library. *Applied Mechanics and Materials*, *614*, 374–377. http://dx.doi.org/10.4028/www.scientific.net/AMM.614.374.

Fowers, S. G., & Lee, D.-J. (2012). An effective color addition to feature detection and description for book spine image matching. *ISRN Machine Vision*, *2012*, Article 945973. http://dx.doi.org/10.5402/2012/945973.

Ghosh, K., Chakraborty, A., Parui, S. K., & Majumder, P. (2016). Improving information retrieval performance on OCRed text in the absence of clean text ground truth. *Information Processing & Management*, *52*(5), 873–884. http://dx.doi.org/10.1016/j.ipm.2016.03.006.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
Hu, Z. L., Tang, J. S., & Lei, L. (2016). A hybrid algorithm for the segmentation of books in libraries. *9869*, In *Mobile multimedia/image processing, security, and applications 2016*. http://dx.doi.org/10.1117/12.2223338, Artn 98690k.
Hu, P. F., Wang, W. L., Li, Q. Q., & Wang, T. J. (2021). Touching text line segmentation combined local baseline and connected component for uchen tibetan historical documents. *Information Processing & Management*, *58*(6), http://dx.doi.org/10.1016/j.ipm.2021.102689, ARTN 102689.
Hu, X. F., Zhou, Y., & Ye, Q. T. (2005). Automatic call number localization in color book images. *Journal of Electronic Imaging*, *14*(4), http://dx.doi.org/10.1117/1.2135796, Artn 043017.
Jampour, M., KarimiSardar, A., & Estakhroyeh, H. R. (2021). An autonomous vision-based shelf-reader robot using faster R-CNN. *Industrial Robot-the International Journal of Robotics Research and Application*, http://dx.doi.org/10.1108/Ir-10-2020-0225.
Lee, T.-Y., Kim, K.-H., & Jeong, G.-M. (2014). Design of an easy-to-use bluetooth library for wireless sensor network on android. *Contemporary Engineering Sciences*, *7*(16), 801–805.
Li, S., Li, M. Z., Xu, Y. J., Bao, Z. Y., Fu, L., & Zhu, Y. (2018). Capsules based Chinese word segmentation for ancient Chinese medical books. *Ieee Access*, *6*, 70874–70883. http://dx.doi.org/10.1109/Access.2018.2881280.
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Springer.
Lyu, B., Akama, R., Tomiyama, H., & Meng, L. (2019). The early Japanese books text line segmentation base on image processing and deep learning. In *2019 international conference on advanced mechatronic systems* (pp. 299–304).
McCarthy, G., & Wilson, S. (2011). ISBN and QR barcode scanning mobile app for libraries. *Code4Lib Journal*, (13).
Mei, J., Islam, A., Moh'd, A., Wu, Y. J., & Milios, E. (2018). Statistical learning for OCR error correction. *Information Processing & Management*, *54*(6), 874–887. http://dx.doi.org/10.1016/j.ipm.2018.06.001.
Mohammed, M. N., Radzuan, W. M. A. W., Al-Zubaidi, S., Ali, M. A. M., Al-Sanjary, O. I., & Raya, L. (2019). Study on RFID based book tracking and library information system. In *2019 Ieee 15th international colloquium on signal processing & its applications* (pp. 235–238).
Nevetha, M. P., & Baskar, A. (2015). Automatic book spine extraction and recognition for library inventory management. In *Proceeding of the third international symposium on women in computing and informatics* (pp. 44–48). http://dx.doi.org/10.1145/2791405.2791506.
Ng, W. W. Y., Qiao, Y., Lin, L., Ding, H., Chan, P. P. K., & Yeung, D. S. (2011). Intelligent book positioning for library using RFID and book spine matching. In *2011 international conference on machine learning and cybernetics, vol. 2* (pp. 465–470). http://dx.doi.org/10.1109/ICMLC.2011.6016840.
Panichkriangkrai, C., Li, L., & Hachimura, K. (2013). Interactive system for character segmentation of woodblock-printed Japanese historical book images. In *2013 international conference on culture and computing* (pp. 200–+). http://dx.doi.org/10.1109/CultureComputing.2013.64.
Quoc, N. H., & Choi, W. H. (2009). A framework for recognition books on bookshelves. In *Emerging intelligent computing technology and applications, proceedings, vol. 5754* (pp. 386–395).
Ramkumar, R., Karthikeyan, B., Rajkumar, A., Venkatesh, V., & Praveen, A. A. A. (2020). Design and implementation of IOT based smart library using android application. *Bioscience Biotechnology Research Communications*, *13*(3), 56–62.
Rigaud, C., Burie, J. C., & Ogier, J. M. (2017). Segmentation-free speech text recognition for comic books. In *2017 14th Iapr international conference on document analysis and recognition, vol. 3* (pp. 29–34). http://dx.doi.org/10.1109/Icdar.2017.288.
Rodriguez-Osoria, V., Nuno-Maganda, M. A., Hernandez-Mier, Y., & Torres-Huitzil, C. (2014). Embedded image processing system for automatic page segmentation of open book images. In *Advances in visual computing, vol. 8888, Pt Ii* (pp. 531–540).
Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618–626).
Shi, X. H., Tang, K. C., & Lu, H. T. (2021). Smart library book sorting application with intelligence computer vision technology. *Library Hi Tech*, *39*(1), 220–232. http://dx.doi.org/10.1108/Lht-10-2019-0211.
Sichao, L., & Miwa, H. (2020). Algorithm using deep learning for recognition of Japanese historical characters in photo image of historical book. In *Advances in intelligent networking and collaborative systems, 1035* Incos - 2019, (pp. 181–189). http://dx.doi.org/10.1007/978-3-030-29035-1_18.
Skalski, P. (2019). Make sense. https://github.com/SkalskiP/make-sense/.
Soheili, M. R., Yousefi, M. R., Kabir, E., & Stricker, D. (2017). Merging clustering and classification results for whole book recognition. In *2017 10th Iranian conference on machine vision and image processing* (pp. 134–138).
Talker, L., Moses, Y., & Ieee (2014). *Viewpoint-independent book spine segmentation* (pp. 453–460).
Tsai, S. S., Chen, D., Chen, H., Hsu, C.-H., Kim, K.-H., Singh, J. P., et al. (2011a). pp. 1029–1032, http://dx.doi.org/10.1145/2072298.2071930.
Tsai, S. S., Chen, D., Chen, H., Hsu, C.-H., Kim, K.-H., Singh, J. P., et al. (2011b). Combining image and text features: A hybrid approach to mobile book spine recognition. In *Proceedings of the 19th ACM international conference on multimedia* (pp. 1029–1032).
Tsai, C. M., Shou, T. D., Hsieh, J. W., & Chang, M. T. (2018). Binarization of call number images for helping elderly retired volunteer to manage books in library. In *Proceedings of 2018 international conference on machine learning and cybernetics, vol. 2* (pp. 456–461).
Ul Ekram, M. A., Chaudhary, A., Yadav, A., Khanal, J., & Aslan, S. (2017). Book organization checking algorithm using image segmentation and OCR. In *2017 Ieee 60th international midwest symposium on circuits and systems* (pp. 196–199).
Wang, Z. M., Tang, J. S., & Lei, L. (2018). Book title recognition for smart library with deep learning. *10668*, In *Mobile multimedia/image processing, security, and applications 2018*. http://dx.doi.org/10.1117/12.2312245.
Xiu, P., & Baird, H. S. (2012). Whole-book recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(12), 2467–2480. http://dx.doi.org/10.1109/TPAMI.2012.50.
Yang, X., He, D., Huang, W., Ororbia, A., Zhou, Z., Kifer, D., et al. (2017). Smart library: Identifying books on library shelves using supervised deep learning for scene text reading. In *2017 ACM/IEEE joint conference on digital libraries* (pp. 1–4). IEEE.
Yu, C. C., Zhang, R. J., & Cheng, H. Y. (2015). Book spine segmentation for various book orientations. In *2015 Ieee 4th global conference on consumer electronics* (pp. 99–100).
Zhang, J. J., Cai, Y., Jiang, W., & Wang, C. Y. (2017). Harris corner detection based leaf image segmentation for ancient Chinese books. In *2017 10th international congress on image and signal processing, biomedical engineering and informatics*.
Zhang, F. J., & Cui, J. M. (2017). UHF RFID label nanometer printing technology and its application in smart libraries. *Tehnicki Vjesnik-Technical Gazette*, *24*(6), 1985–1989. http://dx.doi.org/10.17559/Tv-20170915032700.
Zhang, J., Zhang, Y. S., & Wu, X. L. (2018). Research of intelligent library based on RFID technology. In *2018 ninth international conference on information technology in medicine and education* (pp. 557–561). http://dx.doi.org/10.1109/Itme.2018.00129.
Zhou, H. N., & Liu, Z. Y. (2009). Page frame segmentation for contextual advertising in print on demand books. In *2009 Ieee computer society conference on computer vision and pattern recognition workshops, vol. 1 and 2* (pp. 403–408).
Zhu, L., & Yang, J. (2011). Ancient books Chinese characters segmentation based on connected domain and Chinese characters feature. *Smart Materials and Intelligent Systems, Pts 1 and 2*, *143–144*, 227–+. http://dx.doi.org/10.4028/www.scientific.net/AMR.143-144.227.
Zhu, B. B., Yang, L., Wu, X. Y., & Guo, T. C. (2015). Automatic recognition of books based on machine learning. In *2015 3rd international symposium on computational and business intelligence* (pp. 74–78). http://dx.doi.org/10.1109/Iscbi.2015.20.
Zurek, E. E., Guerrero, G., Reyes, C., Hernandez, R. J., Jabba, D., Wightman, P. M., et al. (2013). Fast identification process of library call numbers for on the shelf books using image processing and artificial intelligence techniques. In *2013 Ieee symposium on industrial electronics & applications* (pp. 222–226).