Developing a System for Automatic Detection of Books

Chapter · January 2022

DOI: 10.1007/978-3-030-84760-9_27

Kaniz Fatema
Chittagong University of Engineering & Technology

Md Razu Ahmed
International Islamic University Chittagong

Developing a System for Automatic Detection of Books

[1]Kaniz Fatema, [2]Razu Ahmed, and [1]Mohammad Shamsul Arefin*

[1]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram-4349, Bangladesh
[2]Department of Electronic and Telecommunication Engineering, International Islamic University Chittagong, Chattogram-4318, Bangladesh
u1504045@student.cuet.ac.bd, razu17@yahoo.com, sarefin@cuet.ac.bd
*Corresponding Author

Abstract. Managing a large number of books is a challenging task for humans. Our proposed system would assist librarians in efficiently sorting books and keeping an up-to-date inventory of huge books. Many methods for detecting only the book spine in a picture have been created. However, not every strategy is perfect. So, this paper presents an automated book detection system that can quickly identify books by reading their titles. Firstly, images are captured from random position. Secondly books in the image are segmented by using Canny Edge Detection and line segmentation. After book spine segmentation stage, an OCR engine is applied to the segmented images to extract book titles and ready for recognition of next stage. Then maximum string matching score is calculated with extracted text for all data of a column of database where all book's Title are enlisted. For which index of database system will get the maximum score, book category information will be extracted of that index.

Keywords: Canny edge detection, line segment, Hough Transform, image segmentation, Tesseract OCR, title extraction, Fuzzy String Matching.

1 Introduction

Despite the rapid advancement of many forms of media, books remain an important source of knowledge in today's world. As the enormous number of books continues to grow on a regular basis, a few book-related tasks that were once performed by humans, such as sorting, locating, and storing books, have become increasingly difficult. Keeping up the up-to-date stock of a expansive number of books may be a troublesome task for people. It is difficult for a person to naturally produce an list of all the books in his or her home or workplace without undergoing the procedure of writing within the title, author names, publishers, and version of individual book into a machine. In a similar way, on a much larger scale, a librarian create an stock of all the books right now on the bookshelves in an productive way. Manual stock production of books is extremely costly in terms of human labour and vulnerable to errors. An automated book detection system can be the solution of the problem. One way to identify the books is to label each book with a unique identifier like a RFID or bar-code and studied the label employing a specialized reader. There's also a third choice that is to utilize pictures from a advanced camera for distinguishing books. Conveying camera-based book discovery arrangements is more cost-effective since no physical labels must be attached to individual books.

Our proposed method has introduced a new solution for book management system. In this system firstly, images are taken from a variety of angles. Secondly book spines in the image are segmented. Segmenting and recognizing book spines in a photo of a bookshelf is exceptionally challenging since each spine contains a little surface region containing few picture highlights and the other spines' picture highlights make disarray in feature-based picture coordinating. Here we utilize an open source Tesseract OCR which is commonly utilized to perform extraction of texts from picture. Extracted texts are utilized as keywords to identify the category of the book comparing it's title with data set. Here we implement Fuzzy String Matching to calculate maximum string matching score with extracted text for all data of a column of database where all book's Title are enlisted. For which index of database system will get the maximum score, book category information will be extracted of that index. The automated book detection system has a wide range of applications. Book recognition can be used to automate libraries, such as keeping track of book inventories. The segmented book spine images could be used in computer graphics and virtual reality applications to populate virtual bookshelves with actual book spines. It can also be implemented in book shops.

2 Related Work

In most of earlier related works, majority authors centered on a particular portion of the framework. Many papers have looked at how to recognize books on bookshelves.

In [1], D. Lee, Y. Chang, J. Archibald, and C. Pitzak develop a framework for library shelf-reading process automation by matching book-spine images.

In [2], Quoc et al. create a system for identifying books keeping on bookshelves utilizing robots with attached cameras. They discover individual book spines by identifying the book boundaries inside the picture. From the sectioned book spines, the content locations have been distinguished utilizing edges and utilized character distinguishing proof to perused the content. Their experiment on a few book spines can be described that picture highlights were created fundamentally for characteristic scenes and hence don't work well with pictures that have plain writings.

A technique for identifying low-quality characters in scene images using several dictionaries ic presented in [3]. They proposed an approach to recognize character by selecting template.

Chen et al. [4] use SURF highlights for recognizing books utilizing their spine images. They utilize smart-phones to capture pictures of bookshelves and segment the book spines from the pictures. Each book spine is at that point questioned against a book spine picture database. Area data from the portable gadget is utilized to construct a location-aware book stock.

In [5] authors show how to use a mobile book tracking device to create a location-aware book inventory by taking images of books stacked on shelves. Here, they use book spine segmentation process to build their module.

Loechtefeld et al. [6] recommend utilizing optical following strategies to identify books from bookshelves utilizing camera phones whereas employing a projector to show book from the bookshelf direction data.

Yeh T. et al. [7] present a technique for searching a document using images, text and OCR. They considered three features like plain text, visual features of figure and OCR text included in the figures.

Herbert et al. [8] introduce SURF, a new scale- and rotation-invariant detector and descriptor. In terms of repeatability, distinctiveness, and robustness, it approaches or even outperforms their previously proposed schemes.

Crasto et al. [9] show an intelligently bookshelf that is on the basis of a camera-projector framework where they utilize the color histogram of the book spines to distinguish books.

Matsushita et al. [10] too present an intelligently bookshelf whereas employing a distinctive approach for book Book Spine Extraction Cross breed Spine Acknowledgment Framework Text-based Acknowledgment Picture Feature-based Acknowledgment Remote Organize Book Spine Image/Text Database Inquiry Characters Figure1: Versatile book spine recognition framework. acknowledgment. Rather than recognizing the book spines specifically, they recognize book covers utilizing neighborhood picture highlights when books are being evacuated from the bookshelf.

In [12], Nuzhat et al. present an algorithm for book spine segmentation and title extraction from the multi cell images.

Karunakaran P [13] focuses on analyzing botnet traffic to determine the domain name of the server's IP address. The domain name is generated in a unique way by this botnet. Attackers create a large number of domains, resulting in a large amount of DNS traffic. The text features are detected using both public and real-time environments datasets, as well as knowledge-based feature extraction.

Vijayakumar. T. et al. [14] introduce a visual saliency guided cognitive classification model for retrieving information from multimedia data storage.

In summary we can say, Most of the previous works were done only the extraction of book name from book cover, or book spine recognition. But work on detecting the book's category has not been done yet. So, in our proposed method , we combine both spine segmentation and text extraction process and also add a book categorization module with it.

3 System Architecture And Design

The topic of automatic book detection and recognition is well-known in the field of computer vision and image processing. To address this issue, a variety of approaches have been established. Our proposed framework is depicted in Figure. 1. The framework consists of the following major steps: (1) Book Spine Segmentation (2) Information Extraction (3) Book's Type Categorization.

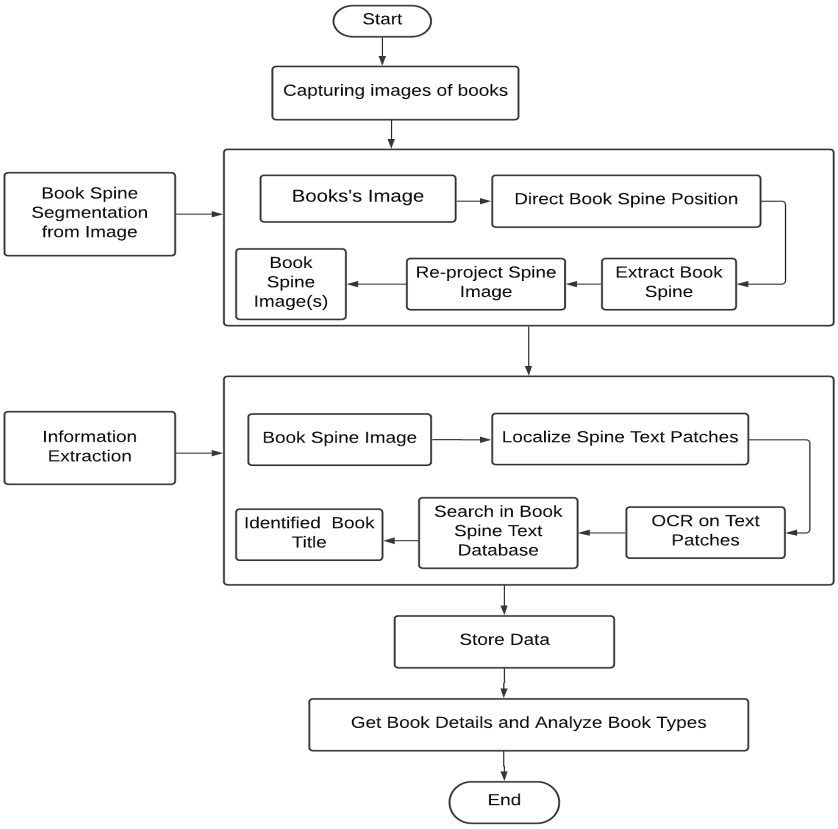


Fig. 1. System architecture of the proposed framework

In the following subsections, a step-by-step overview of the proposed method is described precisely.

3.1 Dataset Description

The whole image dataset consists of 50 digital images of different library bookshelves that are used as the inputs for the proposed method. Figure. 2 shows a sample input dataset where images contains books organized horizontally or vertically.

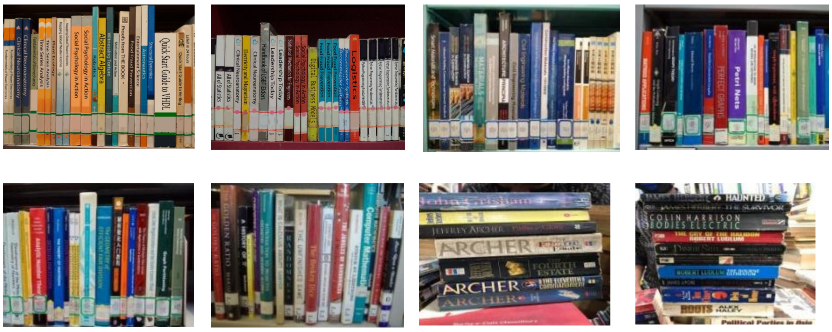


Fig. 2. Sample input data set

For categorization of books, a database containing book details is used which is collected from different websites [16, 17]. In total, 145351 samples were collected for analysis. There are many features including book title, author, edition, copyright year, language, subject classification . As book spine generally contains book title and author names, sometimes contains only book title, so, we mainly focus only the book title feature of the dataset and subject classification feature for book categorization. After extracting book title from book spine, title is compared with the book title of each row of this dataset. When the title is matched, system extract the category of that book from the database. If any book's information is missing in the database, the book's data are added to the database manually to enrich our dataset.

3.2 Data Preprocessing

In this proposed method, a photo of a bookshelf with vertically or horizontally arranged books is taken as input and pre-processed before the extraction of the feature. The image is re-sized so its dimensions can be 450×600 pixel where height is 450 pixels and width is 600 pixels. The height is calculated based on the sample's actual size, then the width is selected so the sample contains the average aspect ratio. Then Gaussian blur is applied to smooth the image and remove noise from it. The next step is to segment the image into sub images containing each book's spine.

3.3 Book Spine Segmentation Module

Since books in bookshelves are usually oriented vertically, the function of this module is to detect the vertical edges and then extract the book spines. If the input image contains horizontally organized books, this system rotate it clock-wise and apply the post-processes. To detect edges, at first Canny edge detection algorithm is applied to the image. Then, using an erosion operation, small and undesirable areas are removed. As a result, an image is created with all vertical edges, where the book spine boundaries are defined by all vertical lines. After that, the Hough line transform is used on image to detect lines that are resemble to book edge. It simply returns the array of (ρ,θ). Using these values vertical lines are drawn on image that are considered as edges of books.

Individual book spines are separated in the next stage. This necessitates a left-to-right scan of the coordinate points of each detected line. The width of each book spine is determined by measuring the distance between two consecutive lines.

$$width = x_{i+1} - x_i \quad (1)$$

Here x_i is the top x-coordinate of i^{th} line and x_{i+1} is the top x-coordinate of i^{th+1} line and image height is considered as length.

Each book spine is isolated using the width and length from the original image. This segmentation procedure results in a series of segmented book spine images which will be fed into the next module.

Algorithm1: Book Spine Segmentation

Input: An image of books
Require: Segment individual book spine

1. **begin**
2. pre-processed image
3. computation of edges with Canny edge detection
4. apply Hough line transform where parameter is canny image, threshold value and theta.
5. draw vertical lines according to houghlines
6. according to vertical lines crop images
7. **for** each segmented images **do**
   Insert them into an array
8. **end for**
9. print array
10. **end**

3.4 Information Extraction

The content on the book spines regularly contains the title and the author names. For the extraction of content, each segmented picture are passed to an OCR. In this module we have used Tesseract OCR. It is an open-source text recognition engine that is used to extract text from a large document, or it can also be used to extract text from an image of a single text line. First of all, the input image is a gray scale image which has to be converted to binary image by thresholding. The thresholded image is used to get a clear difference between white pixels and black pixels. If this image is delivered to Tesseract, it will easily detect the text area and will give more accurate results. To get better performance Tesseract, it is configured with the parameters **Engine Mode (--oem)** to 3 and **Page Segmentation Mode (--psm)** to 6 to assume a single block of text. Tesseract gives the output as a dictionary. This dictionary has content of the input image such as its detected text location, position information, height, width, confidence score, text etc. As images have a mixture of digits,symbols, other characters, and text and it is not specified to Tesseract that a field has either only text or only digits, so Confidence score (greater than 30) is considered to detect whether it belongs to text or digits. In the "text" information of the dictionary extracted single words are found as array. To get the full text, array elements are joined. After extracting the book data from the spine content, the information will be put away in a array. Recognizing book category will be done by analyzing this data.

Algorithm 3: Text extraction from image

Input: A book spine image
Require: Extract text from image

```
begin
for each segmented image do
  Transpose image anti-clockwise
  Convert gray scale image to binary image
  Configure --oem=>3 and --psm=>6
  Feed image to Tesseract OCR to get image details in a dictionary
  for each word in 'text' array of dictionary do
     if word is not null
        append word to a list
     else
        do nothing
     end if
  end for
  join all elements of the list
  append the resultant text to another list
  print the list
  end
```

3.5 Books Categorization

We primarily focus on the book title feature of the dataset because book spines usually include both the book title and the author names, but sometimes only the book title and subject classification feature is considered for book categorization. The book title is extracted from the book spine and compared to the book title of each row in this dataset. When a title is found that matches, the machine searches the database for the book's type. Generally book's category prediction can be done with various Machine learning approaches. But it gives very low accuracy because Tesseract OCR sometimes can't read different textual style and some letters can be read because of lighting effect or bad quality of images. So, here we implement Fuzzy String Matching to calculate maximum string matching score with extracted text for all data of a column of database where all book's Title are enlisted. For which index of database system will get the maximum score, book category information will be extracted of that index. The algorithm for book categorization is given here.

Algorithm3: Books Categorization

Input: Extracted book's title and book database
Require: Select the book's category

```
begin
for each title do
  max=0
  for each index of book_database do
    Calculate similarity score between extracted title and book_title in by Fuzzy
    String Matching
    if score>max
      max=score
      z=index
    end if
    if max>0
      extract book's category name from database and store it in a list
    else
      do nothing
    end if
  end for
end for
print the list
end
```

Here at first, we declare a variable *max* initialized with zero(0). Then we calculate string matching score between an single extracted title and book_title value of each row using fuzz function. If resultant score is greater than *max,* then the score is assigned to *max* and the index of that row is saved in another temporary variable. We apply this process for each row of the dataset. Thus, we can find out the index of that row for which the string matching score is maximum. Using the index, we extract the subject_classification column's value of that index which indicate the book's category. Same procedure is applied for all extracted book title and thus we get all books' category.

4 Implementation and Experimental Result

4.1 Experimental Setup

The proposed system has been implemented on a machine having windows 10, Core i5 2.4 GHz with 8GB RAM. Python 3.6.7(version) is used for developing it. Tesseract OCR is used for text extraction**.**

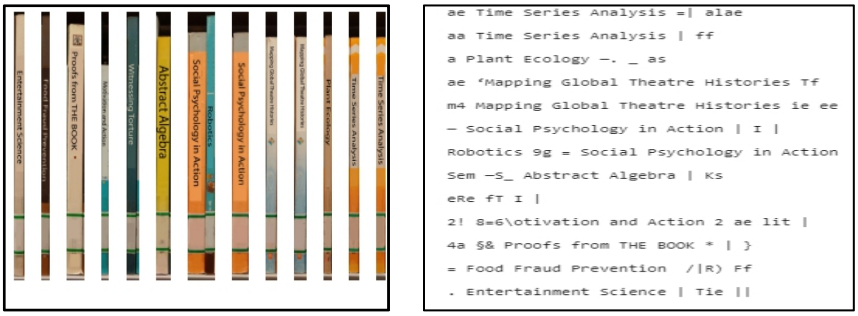
4.2 Implementation

Images containing books organized horizontally or vertically are the inputs of our system. The following figures shows the simulation result after applying the proposed method on different images. Figure 3(a) shows input image whereas Figure 3(b) shows vertical boundary lines of the edge detected images.In Figure 3, we can see that an image with many books is taken and the book spine boundaries are detected. After that book spines are segmented according to boundaries shown in Figure 4(a).

(a) (b)

Fig. 3. (a) input image, (b)vertical boundary line as detected edge

In Figure 4(a) shows some of segmented book spine images and Figure 4(b) shows extracted texts by OCR from each book spine and text extraction process is described briefly later.



(a) (b)

Fig. 4. (a)segmented individual book's spine, (b) extracted book's title

Text extraction process is described step by step in the following which is shown in Figure 5. Here we consider one of the segmented book spine which is input to this module.We pre-process image by converting into gray scale and do binarization. Since the characters of the title derived from the book images have low resolutions, it can be difficult to identify them. So, considering the circumstances, this module proposes a technique where words of the title are segmented by creating bounding box shown in Figure 5(d) and then OCR read them. After getting all the extracted words of the title, we join them and get the whole title of that book.

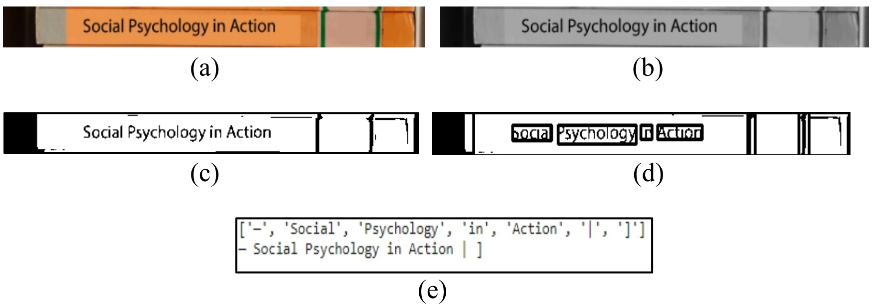


Fig. 5. (a) Segmented book spine, (b)Gray scale image, (c)Binary image, (d) Localize text patches, (e) Extracted book title

After extracting all the books' title containing in the input image, system search each book's title in the database containing details of books. When title matched, system extract the category of the that book from database and show it as output.

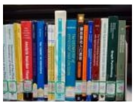
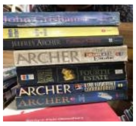
4.3 Performance Evaluation

To evaluate the efficiency of the proposed system, we calculate the accuracy with the following equation and performance accuracy of each module for various images is shown in Table 1.

$$accuracy = \frac{No.\, of\ corrected\ correspondence}{No.\, of\ correspondence} * 100\% \quad (2)$$

Table 1. Performance analysis for proposed system

Sample	No. Of books	Correctly detected books	Book spine segmenta-tion accuracy	Text extraction accuracy	Book type categoriza-tion accuracy
	23	21	91.3%	95.45%	94.44%
	22	16	72.72%	93.75%	93.75%
	14	13	92.85%	84.61%	84.61%
	12	9	75%	77.78%	71.42%
	17	12	70.6%	64.7%	64.7%

Sample	No. Of books	Correctly detected books	Book spine segmenta -tion accuracy	Text extraction accuracy	Book type categoriza -tion accuracy
	18	9	50%	50%	50%
	17	10	58.82%	52.94%	50%
	7	4	57.14%	50%	50%
	14	9	71.4%	66.67%	63.67%
	15	11	73%	72.72%	72.72%

The issues we have confronted in this extend is in utilizing Tesseract open source OCR because it is troublesome for it to recognize texts of distinctive textual styles. Sometimes few characters are also missing in the extracted text as OCR can't scan perfectly from low quality image. Low quality image can be produced due to light effect.In this case, Machine learning algorithms give low accuracy for book type prediction. For this reason, we calculate the maximum string matching score with extracted text for all data of a column of database where all book's Title are enlisted. For which index of database system will get the maximum score, book category information will be extracted of that index.

Segmentation is also a challenging part for this system. Sometimes detected edges shows jagged appearance, so lines drawing on them also have same jagged appearance. So cropping images according to line has been become troublesome for this proposed system. As image quality is also a cause for getting less accuracy, We are trying to improve our pre-processing steps for image and text recognition to increase the accuracy.

Another issue is that, if any book's data are missing in database, this system failed to recognize the category of that book. So, we also keep enriching our dataset with new data.

4.4 Comparison with other existing framework

The performance of the proposed method is to detect category of all books from a image containing set of books organized horizontally or vertically. The proposed method in [11] only spine segmentation is accomplished. In [12] spine segmentation and text extraction is done. In both proposed method there were no detection of book type . From the comparison in conclusion we can say that, our proposed method is better than other existing frame work.

Table 2. Comparison with other existing framework

Method	Book Spine segmentation accuracy	Text extraction accuracy	Book type categorization accuracy
Method 1[11]	92%	Not done	Not done
Method 2[12]	95%	86%	Not done
Proposed Method	95%	87%	90%

4.5 Future work

During the implementation of this system, we have faced some challenges that is the cause of less accuracy. To overcome the challenges, we have intention to make few changes in our system. As the segmentation module of our system is facing difficulties, we will try to improve its efficiency. Another cause of less accuracy is missing of letter during text extraction. Reading different textual style is also difficult for Tesseract OCR. So, we will work in future to overcome these problems.

The proposed technique is limited to detect book spines kept in only one row and one column of a bookshelf or table. And we work only with English books. We have tried to work with Bangla books, but a good Bangla OCR is not developed yet. This work can be expanded in the future to work with multiple rows and columns of multilingual books.

5 Conclusion

In this paper, a new method is introduced for a modern system for recognizing and detect types of books from an image. This work is significant for managing lot of books. This system successfully applied on a set of images containing books organized horizontally or vertically. From an image, we have segmented the individual book spine image. Information is extracted from the segmented images and used as keywords to look through a book details database. The issue we have confronted in this extend is in utilizing Tesseract open source OCR because it is troublesome for it to recognize texts of distinctive textual styles. But at the end , this proposed system has achieved acceptable accuracy. This system can be used as the fundamentals for building virtual library or book shop.

References

1. Lee, D. J., Yuchou Chang, Archibald, J. K., & Pitzak, C. (2008). Matching book-spine images for library shelf-reading process automation. *2008 IEEE International Conference on Automation Science and Engineering*. https://doi.org/10.1109/coase.2008.4626503
2. Quoc, N., & Choi, W. (2009). A framework for recognition books on bookshelves. *Emerging Intelligent Computing Technology and Applications*, 386-395. https://doi.org/10.1007/978-3-642-04070-2_44
3. Sawaki, M., Murase, H., & Hagita, N. (n.d.). Character recognition in bookshelf images by automatic template selection. *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*. https://doi.org/10.1109/icpr.1998.711890
4. Chen, D. M., Tsai, S. S., Girod, B., Hsu, C., Kim, K., & Singh, J. P. (2010). Building book inventories using smartphones. *Proceedings of the international conference on Multimedia - MM '10*. https://doi.org/10.1145/1873951.1874043.
5. Chen, D., Tsai, S., Kim, K., Hsu, C., Singh, J. P., & Girod, B. (2010). Low-cost asset tracking using location-aware camera phones. *Applications of Digital Image Processing XXXIII*. https://doi.org/10.1117/12.862426
6. M. Loechtefeld, S. Gehring, J. Schoening, A. Krueger. Shelftorchlight.:Augmenting a shelf using a camera projector unit. UBIProjection 2010 - Workshop on Personal Projection (2010)

7. Yeh, T., & Katz, B. (2009). Searching documentation using text, OCR, and image. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*. https://doi.org/10.1145/1571941.1572123
8. H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool.:Speeded-up robust features (SURF). Computer Vision and Image Understanding, 110(3) (2008)
9. Crasto, D., Kale, A., & Jaynes, C. (2005). The smart bookshelf: A study of camera projector scene augmentation of an everyday environment. *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*. https://doi.org/10.1109/acvmot.2005.116
10. Matsushita, K., Iwai, D., & Sato, K. (2011). Interactive bookshelf surface for in situ book searching and storing support. *Proceedings of the 2nd Augmented Human International Conference on - AH '11*. https://doi.org/10.1145/1959826.1959828
11. Jubair, M. I., & Banik, P. (2013). A technique to detect books from library bookshelf image. *2013 IEEE 9th International Conference on Computational Cybernetics (ICCC)*. https://doi.org/10.1109/icccyb.2013.6617619
12. Tabassum, N., Chowdhury, S., Hossen, M. K., & Mondal, S. U. (2017). An approach to recognize book title from multi-cell bookshelf images. *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. https://doi.org/10.1109/icivpr.2017.7890886
13. P, K. (2021). Deep learning approach to DGA classification for effective cyber security. *December 2020*, *2*(4), 203-213. https://doi.org/10.36548/jucct.2020.4.003
14. T., V., & R., V. (2020). Retrieval of complex images using visual saliency guided cognitive classification. *Journal of Innovative Image Processing*, *2*(2), 102-109. https://doi.org/10.36548/jiip.2020.2.005
15. Do, Y., Kim, S. H., & Na, I. S. (2012). Title extraction from book cover images using histogram of oriented gradients and color information. *International Journal of Contents*, *8*(4), 95-102. https://doi.org/10.5392/ijoc.2012.8.4.095
16. *COVID19-engineering-Books-NLP-Dataset*. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/praveengovi/covid19engineeringbooksnlpdataset
17. *Goodbooks-10k*. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/zygmunt/goodbooks-10k?select=books.csv