Available online at: http://www.ijmtst.com/vol5issue11.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 05, Issue No: 11, November 2019

SIS: Smart Identifying System for Library Books

Bhupendra Singh

Gandhi Smarak Degree College, Surjannagar, Moradabad

To Cite this Article

Bhupendra Singh, "SIS: Smart Identifying System for Library Books", *International Journal for Modern Trends in Science and Technology*, Vol. 05, Issue 11, November 2019, pp.-166-173.

Article Info

Received on 01-November-2019, Revised on 19-November-2019, Accepted on 21-November-2019, Published on 26-November-2019.

ABSTRACT

Finding Books in the Library with a variety of collections is really a tough job for the readers. Lately, positioning technologies such as RFID or Wi-Fi is implemented in some libraries to help the readers to save their time. Certainly, it is difficult for most libraries to afford human resources and budgets for perfect maintenance of the system. The proposed system uses Template Matching algorithm. It is based on Image processing that scans image of the book and identify the location of that particular book. Initially a snippet will be attached for every book in the library and also stored in library database. when user requires a book, he/she will first enter the name or author of that particular book. The library software will find the corresponding snippet of that book and search the required book among the books through library cameras. The process involved in searching of snippet in the library cameras is detailed below. The experimental result for four books is shown and therefore the accuracy obtained is 97.8%.

Keywords—*Machine Learning, Image Processing, Template Matching Algorithm, OpenCV, Python, Smart Library, Cameras*

1.INTRODUCTION

There are many valuable and long-standing resources for knowledge and learning, in physical library collections. Finding books, from a large collection is really a tedious manual work where some books may misplace in such large collections. By using the concept of Machine learning the Books that are arranged in unordered manner can be found easily. Template matching Algorithm used in this project makes the searching of books easier compared to the traditional approach. This concept can be used in schools and colleges library to help the students easier and faster to choose their books from the racks.

Books are found using Deep learning method by recognizing the title that are in the book rack. This technique may fail due to damaged books. RFID or NFC method is used with circuits so it may be complicated and difficult for the readers to find the book. The objective of this project is to find the book in the library efficiently and quickly. The idea of this digital library visualization system is to improve user's experience to locate the book automatically. Nowadays, in this existing problem of library book management, it is important to use advanced technology to improve the efficiency of library searching and management statistics.

Digital library systems have many repositories of serves as gateway that serves digitized data like images, text and audio. This method has sufficient search tools to find the image that has been searched by the user effectively from the stored

database. Text based method is the most common way to find the book in this modern digital library.

2. RELATED WORK

Mohammad Imrul Jubair and Prianka Banik (2013) proposed a technique that has many processes. The first step is to capture the image and is filtered using a technique called noise removing technique. Second step is the result of the filtered image is processed into two process row extraction of the bookshelf and book region detection. The captured image can be filtered by existing noise removal techniques. Noise occurred due to salt and pepper noise or impulse noise will create a huge impact at the level of detection in the sector of binary conversion and edge detection. This will be an issue in these results. So, this noise could be removed using one of the techniques that is proposed and hence salt and pepper noise is eliminated. The property of a bookshelf is the horizontal line detection which will detect the processed image. Then canny edge detection is used at this process on that processed image so that undesirable pixels are completely dislodged. At last, the number of pixels of each horizontal line are taken into account and threshold value is determined [1].

Gregory Short and Beomjin Kim (2014) have proposed a system that visualizes a three tired system to give comfort to the users while searching. The first tier would be depiction of possibly related data from the stored database. The next tier is getting a screenshot that captures the previous tier by the user. Then implementing distinct levels of visual abstraction is to display some different levels to be detailed at each of the view, then it will distil out extraneous information from the client's display passing them to focus on that level at a particular time, hence reduces the whole work. The main purpose is to display the user towards many data that are possibly relevant to that interest. Searching of this process is applied to the whole database, based on the user's search that is related with basic information i.e., title of the book, author, year of publication, then ratings of that book [2].

Po-Kai Liao (2015) suggests that most of the book searching technology in library management are using wireless technology. Example, most use RFID to resolve library positioning from 2008. At present smart mobile phones are increasingly used in day-to-day life, so wireless technology usage has also increased. In National Chung Cheng University library, the library visitors can connect to the library Wi-Fi by their own smart phone and locate their position. Then they can scan the Wi-Fi sensors to locate the books on the arranged bookshelves by their mobile phones precisely. Another method for Wi-Fi location is RFID technology that is used to position the books and the users can access the books by Wi-Fi location. From that the users can easily locate the path of the book by their smart phone [4].

He at el (2017) proposed a method to locate the books by RFID technology. The method proposed was a hardware module that is fully functioned computer. Another end module consists of antenna and an electronic tag called as the reader module. This module involves a controller to perform a specific task, a RF band and a baseband module. Antenna is used to transmit and receive the RF signals between modules. Each book labelled has an electronic tag which encodes the information of that particular book. The electronic module uses RFID named PR9000 and the radio frequency (RF) chip named JR20X0. Computer software includes the information of books, user-system interaction interface, RFID data transmission and reception process program [5].

Xiao Yang at el (2017) have proposed that spine image of book has been identified based upon the text recognition, then it is used for searching and indexing from the database stored. The text recognition is a conventional approach that has to segment first and recognize each of the character in the word and predict based on the language of the model or combination of rules. However, this approach is highly sensitive to various contortion in images that cause character level segmentation inadequacy. To deviate from the character segmentation process, a cast text recognition is used at sequential labelling process and a sequence of characters is recognized simultaneously. Each character is standardized through a Soft max function and interpreted as an exudation of character at a precise time. Stochastic gradient descent (SGD) method that is used in this optimization and the CTC gradient loss has been efficiently controlled using a method called forward backward dynamic programming [6].

In all the proposed methods, books are found using Deep learning method by recognizing the title that is in the book rack. This technique may fail due to damaged books. RFID or NFC method is used with circuits so it is complicated and difficult for the readers to find the book.

3. SYSTEM ARCHITECTURE

The method deployed in this work is Template Matching algorithm. It is a technique in digital image processing to locate little elements of picture that is matched to a template image. The proposed approach in this work uses Neural Networks and Deep Learning classifiers like Visual Geometry Group (VGG). Initial process of the project is to label each book with a specified snippet and it must be stored as a template image for that specified book in the library database.

Fig. 1. System Architecture

Figure 1 shows the system architecture of the whole project. The user had to enter name or author in the library software. The software will be embedded in the fully functioning library system. The software locates the book by accessing the surveillance camera which is installed in most libraries. The output shows the exact location of the book wherever it is placed.

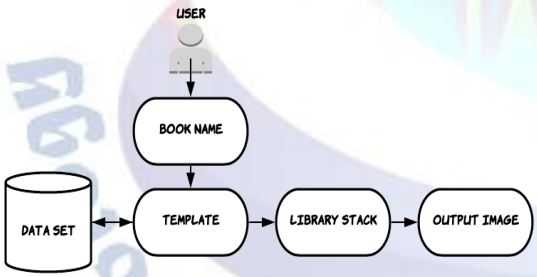

Fig. 2. Block Diagram Process

Figure 2 shows the diagram of the overall system. Once the user enters the book name, it searches the corresponding template from the stored database. Then the template is searched using the library camera to locate where the book is placed and once the template is matched with the snippet labelled in that book then the book will be marked and shown in the system screen with that exact location. This algorithm works well even if the book is misplaced or not arranged properly.

3.1. Template matching

Template matching is one of the functions in OpenCV module that is used for Image processing and Video processing. It helps to find the template image in the source image and gives the location where it is placed.

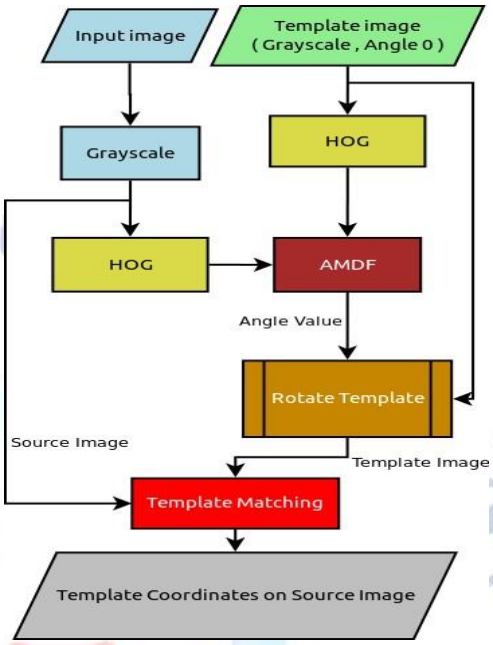

Fig. 3. Flow Diagram

Figure 3 shows the internal flow of the template matching algorithm. It requires two parameters that is source image and template image. The source image is the input image to find the template whether it is present or not. The template image is the snippet where it is fed to the database once initially. The template image should be found in the source image, if it is found the template image will be marked in the source image with clean borders as output. The template matching algorithm is one of the algorithms present in Deep Convolutional Neural Networks method that is used to pass the image through totally different hidden layers and the information of that image will be present in each layer. The feature of the image will use those vectors that are obtained from the network. This feature extraction in Deep Neural networks is effective in any searching process that involves template matching.

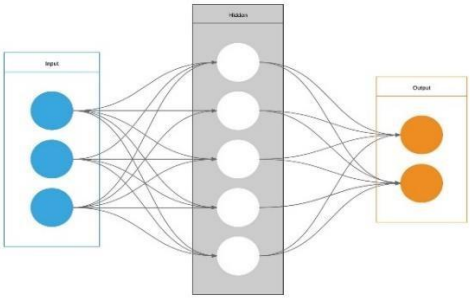

Fig. 4. Layer Output

Figure 4 shows the hidden layer present among the template matching which holds each information of that template and it will be processed at the end giving the results.

The template matching algorithm will be effective when the template image occupies majority of the source image and it will be quick to locate the template in that source image. The template matching algorithm depends upon sampling points of the source image. Resolution is one of the factors to reduce the sampling points. High resolution of source image will be effective in gathering more number of sampling points and hence the template image will be found easily and quickly.

In some cases, the template image cannot be matched directly. So, it must be implemented by using eigen spaces matching, that will match with different conditions, different perspectives, contrasts of color and different matching objects. Example, if the books are not properly placed like in slanted positions (angles tilted), in that case eigenspaces of different snippets or templates of that particular book must be stored in the database. Then the book will not be missed during searching and gives more accurate location.

In additional it is also possible to block the snippet by another book or any other objects. In such case, the template will not be found in that source image. For example, a snippet of book is torn or folded then only half of the snippet is visible or the camera angle could not find the whole snippet, the problem becomes complicated. The possible solution for this problem is to store possible number of templates that covers the snippet. Dividing the template image into a greater number of templates, which are stored in the database. So, the algorithm finds one of the templates matches with a source image and possibly gives the output. In this method every possible search makes the template matching algorithm work effectively.

3.2. Template-Based Matching Using Cross Relation or Sum of Absolute Differences

The methodology of template matching using cross relation uses a template image and locate to a particular feature of that search image. This method is often performed on grey pictures or edge pictures. From the process the place in the source image where the highest cross correlation value is obtained matches with the template image more effectively so that large image values are multiplied by mask values. This process is implemented by taking search image as an input image, the input image will be S(x, y), where (x, y) are the coordinates of every element within that input image. Then the template image will be T(xt,yt), where (xt,yt) are the coordinates of every element of the template image. Then merely move the centre/origin of the template T(x t, y t) above every (x, y) purpose within that input image and calculate the sum of products between the S(x, y) and T(x t, y t) in the template. Then the possible position of the template in the input image is considered and the highest cross correlation value obtained is the exact location of the template image in source image. This methodology is usually remarked as 'Linear spatial Filtering'.

$$SAD(x,y) = \sum_{i=0}^{Trows} \sum_{j=0}^{Tcols} Diff(x+i, y+j, i, j) \quad (1)$$

Alternate way to handle the translation issues on pictures using template matching is to check the pixel intensities, using the SAD (Sum of absolute differences) in equation (1). A pixel of that input image with co-ordinates (xs,ys) is the intensity Is(xs,ys) and a pixel of the template with the coordinates (xt,yt) is the intensity It(xt,yt). So, the absolute difference of the pixel intensity is shown in equation (2)

$$Diff(xs, ys, xt, yt) = |\ Is(xs, ys) - It(xt, yt)\ | \quad (2)$$

4. IMPLEMENTATION OF ALGORITHM

Input the source image. Convert the source image from RGB image to Grayscale image. Next input the template image fetched from the stored database. Once storing the breadth and height of template in w and r, it has a tendency to initialise a variable to track that region and the image scale with the most effective match. Then the iteration of the multiple scales of the image starts using the linspace() function in NumPy module. This function requires 3 parameters, the beginning value, the ending value, and the number of equal chunk slices. It starts from cent percent of the initial size of that image and work in dropping to 20% of the initial size in twenty equally sized percentage chunks.

It will resize the image to the current scale and manipulates the ratio to the new width, and hence it's necessary to track the ratio. It ensures that the template matching image should be smaller than input image. If the template image is larger, then matchTemplate() function in OpenCV module can make a decision and throw an error, so simply break the loop in this case.

At this time, template matching is applied to the resized image: The minMaxLoc()function in OpenCV module takes the correlation output and returns the 4 tuple which has the minimum correlation value, the maximum correlation value, the minimum (x, y)-coordinate, and the maximum (x, y)-coordinate.

After that the template image matches the source image, the regions that are matched at each iteration are updated using a variable. The maximum value of correlation, maximum (x,y) coordinate, ratio of original width to current image and width of the resized image area are also the variables that should be updated. At last bounding box is drawn on the input image using the updated variables and the output image with bounding box is displayed on the screen.

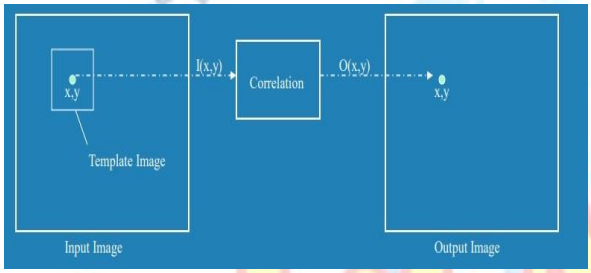

Fig. 5. Correlation of Input Image

Figure 5 shows the correlation of the template image found on the input or source image. The correlation matching is done pixel by pixel. The template image is also called as bi-level image.

4.1. Implementation Results

A Small circle image template used in various databases is taken as an example.

Fig. 6. Template Image

Figure 6 shows a small circle which is taken as a template image. In further dataset each image in the database will be processed with template image and the result will obtained as shown below.

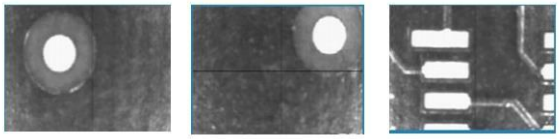

Fig. 7. Dataset

Figure 7 shows a collection of images in the dataset. Each image of the book will be act as a source image. The template image will be searched in the source image to get the desired position in the source image as an output of the template matching algorithm.

Normally, all the template images of each book will be stored in the database. Also, deletion and insertion of a template image will be taken care by the library management system.

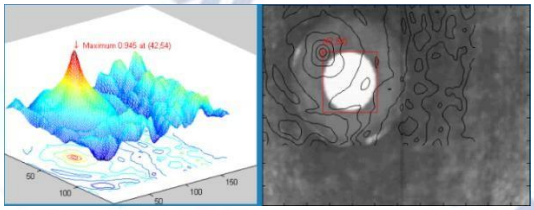

Fig. 8. Result of First Image

Figure 8 shows the output of the first image in the dataset. That the template image is searched and get bounded in the source image. The rectangular box in the left side of source image is exact image of the template image. This is how the template matching algorithm works.

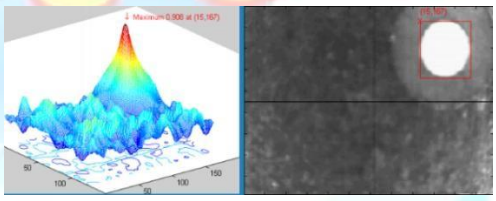

Fig. 9. Result of Second Image

Figure 9 shows the output image of the second image in the dataset. The template image is present in right top corner where the template image is shown bounded.

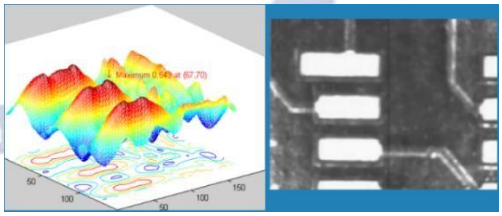

Fig. 10. Result of Third Image

Figure 10 shows no bounded boxes on the source image because the template image is not present in the source image. So, it can't box anything on the source image and shows the image as it is.

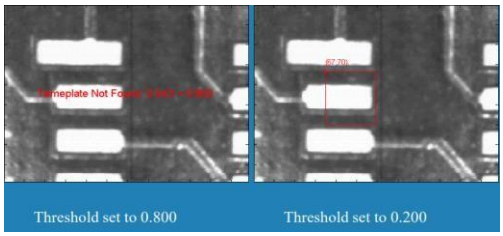

Fig. 11. Different Results for Different Threshold Values

Figure 11 shows different results for various threshold values. The threshold value set in left side image is same as in the figure 11. So, no output is shown in the left image. In the right-side image is shown a box in the middle as the threshold value is set to 0.2.

Fig. 12. Source Image and Template Image

Figure 12 shows a sample source image and template image to implement the algorithm.

Fig. 13. Output Image

Figure 13 shows the output of the algorithm. It uses TM_CCOEFF_NORMED method of comparison. There are six methods of comparison in literature.

- TM_CCOEFF
- TM_CCOEFF_NORMED
- TM_CCORR
- TM_CCOERR_NORMED
- TM_SQDIFF
- TM_SQDIFF_NORMED

5. RESULTS AND DISCUSSION

The proposed algorithm will be implemented for an image. In library stack, the camera will be recording the books in its coverage. Each book of the library must be covered by the cameras placed. So that all the books can be identified by this project. Systematically every book on the stack can be found whatever the positions may be and even if the book is misplaced. Live recording and pre-recorded video of the library can also implement the template matching algorithm. The video will split into frames and each frame will be matched with the template to get the desired output.

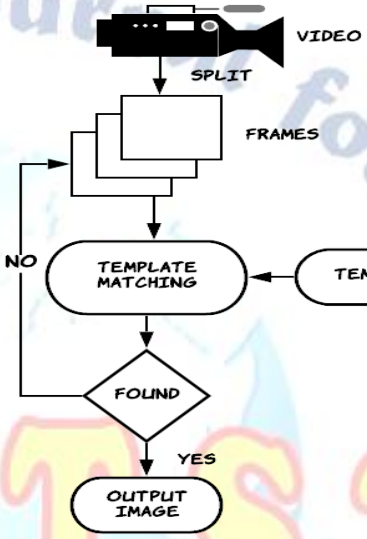

Fig. 14. Video Template Matching Flow Diagram

The user can enter the book name or author name to find in the library. Those books will have a template (Dataset). Each book has its dataset. The dataset will be created in the initial setup of this project. By entering the name, the specific dataset will be accessed and the template image for that book will be searched in all the cameras in the library. If the template matches, the book will be found and can be seen in the output on the screen as per the flow diagram as shown in the figure 14.

Fig. 15. Output-1

Figure 15 shows the books are arranged in a stack, which makes the algorithm easy to find the book given by the user. From the figure 15 user enters the book named **SOFTWARE ENGINEERING**. So, the system reads the input and the algorithm will make process to find the book(template) called SOFTWARE ENGINEERING and after finding it, the bounded box will be displayed on the output as shown in figure 15.

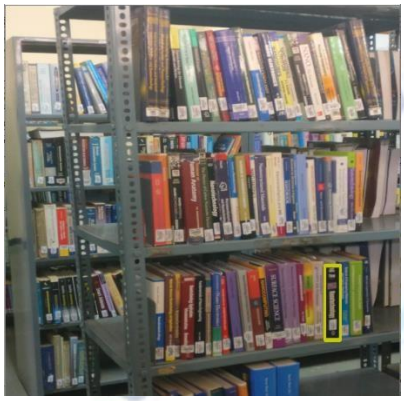

Fig. 16. Output-2

In this stack of books, they are arranged in a disorderly manner as shown in figure 16. The user enters the book name and the camera starts searching the book. But the camera is placed far away from the stack. So, the system takes an additional time to search the book than normal time and after finding it, the bounded box will be displayed on the output as shown in the figure 16.

Fig. 17. Output-3

Figure 17 shows some books that are arranged in a manner facing front side of the books. So, it is very easy to find the book. Whenever the user types the book name and the camera tends to make search of the book, Likewise, in this Figure 17 the user searches the book called "**RAM**" and the algorithm makes it an easy job as the books are faced in front side in the stack. After finding the book it will show bounded box on the output.

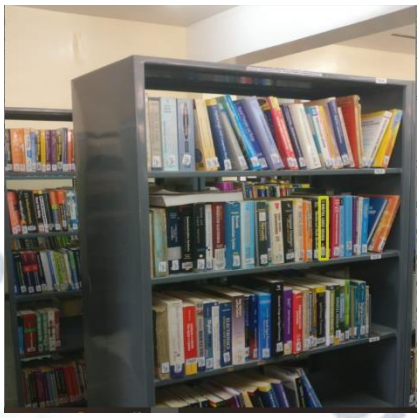

Fig. 18. Output-4

The books are found even when the books are misarranged as shown in figure 18. Now, whenever the user types the book name and the camera starts searching the book and the algorithm makes the job easier to find the job. When the book is found, it will be shown in a bounded box on the output.

6. CONCLUSION AND FUTURE WORK

In this paper a system for detecting and recognizing the books in the bookshelf accurately in a precise manner to build a digital library inventory is proposed. The algorithm speeds up the process thus the user can easily and quickly locate the books even if the books are arranged in a disorderly manner or misplaced. This idea visualization improves the user's experience and attracts more user to read the books in the library regularly.

In a future a three-tiered digital library visual image system to find books efficiently can be done. Many different criteria had to be thought-about e.g., color and dimension of a book. A more robust can be adopted which can be classifier trained to classify book and non-book elements which can detect books more accurately and in reduced time

REFERENCES

[1] Mohammad Imrul Jubair, Prianka Banik, "A Technique to Detect Books from Library Bookshelf Image", 9th International Conference on Computational Cybernetics July 8-10, Tihany, Hungary, 2013, PP 359-363, IEEE.

[2] Gregory Short, Beomjin Kim, "Multi-Tiered Visual Interfaces

[3] Book Search with Digital Library Systems", 6th International Conference on Multimedia, Computer Graphics and Broadcasting, 2014, PP 21-24 IEEE.

[4] Najad Al-Najdawi, Sara Tedmori, Omar Alzubi, Osama Dorgham, JafarAlzubi, "A Frequency Based Hierarchical Fast Search Block Matching Algorithm for Fast Video Communication", International Journal of Advanced Computer Science and Applications, May 2016, DOI: 10.14569/IJACSA.2016.070459.

[5] Po-Kai Liao,"The Development of Library Mobile BookFinding System Based on NFC", IIAI 4th International Congress on Advanced Applied Informatics, 2015, PP 148-153 IEEE.

[6] He He, Tao Liu, Ershen Wang, "Intelligent Book Positioning System for Library Based on RFID", 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2017, PP 723726 IEEE.

[7] XiaoYang, DafangHe, WenyiHuang, Alexander Ororbia ZihanZhou, DanielKifer, C.LeeGiles, "Smart Library: Identifying Books on Library Shelves using Supervised Deep Learning for Scene Text Reading", 2017, IEEE