Bachelor of Science in Computer Science & Engineering

Developing a System for Automatic Detection of Books

by

Kaniz Fatema

ID: 1504045

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

April, 2021

Developing a System for Automatic Detection of Books

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Kaniz Fatema
ID: 1504045

Supervised by

Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled ‘**Developing a System for Automatic Detection of Books’** submitted by ID: 1504045, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Asaduzzaman
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

First of all,Thanks to almighty Allah for allowing me to successfully complete this thesis.

I would like to express my heartfelt gratitude to my supervisor Dr. Mohammad Shamsul Arefin ,Professor of Computer Science and Engineering Department at Chittagong University of Engineering and Technology. Without his continuous guidance, this thesis can't be materialized successfully. He encouraged me to take the challenges that i could never imagine to accomplish. His critical questions about my thesis always help me to think from different perspective and this helps me to improve efficiency of the thesis work.

Finally, I would also like to thank my parents and friends for their unconditional love, support, motivation, assistance in completing this project in such a short period of time.

Abstract

In this paper, we present a system to recognize books by reading its title on book spines. Firstly, images are captured from random position. Secondly books in the image are segmented by using line segment and Canny Edge Detection. 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: Book recognition, line segment, Hough Transform, Tesseract OCR, Fuzzy String Matching

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Nowadays, regardless to extremely develop of many kinds of media, books are essential information source to life. With the gigantic number of books keep developing regular, a few book related works where once in the past done by human creatures such as organizing, finding, storing books, etc., are getting increasingly troublesome. Keeping up an 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 inventory of all the books he/she has within the home or work environment, without going through the process of writing within the title, authors' title, publishers, and version of each book into a computer. Similarly, on a much bigger scale, a librarian create an stock of all the books right now on the bookshelves in an productive way. Manual creation of the stock is exceptionally high-priced in terms of human labor and error-prone. One way to recognize the books is to tag each book with an identifier such as an RFID or bar-code and studied the tag employing a specialized reader. Another way is to utilize pictures from a advanced camera for distinguishing books. Conveying camera-based book discovery arrangements is more cost-effective because there's no have to be join any physical labels to single books.

Our proposed method has introduced a new solution for book management system. In this system firstly, images are captured from random position. 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.

1.2 Framework/Design Overview

Proposed framework is depicted in Figure 1.1. The framework contains the following major steps: (1) Book Spine Segmentation (2) Information Extraction (3) Book's Type Categorization.

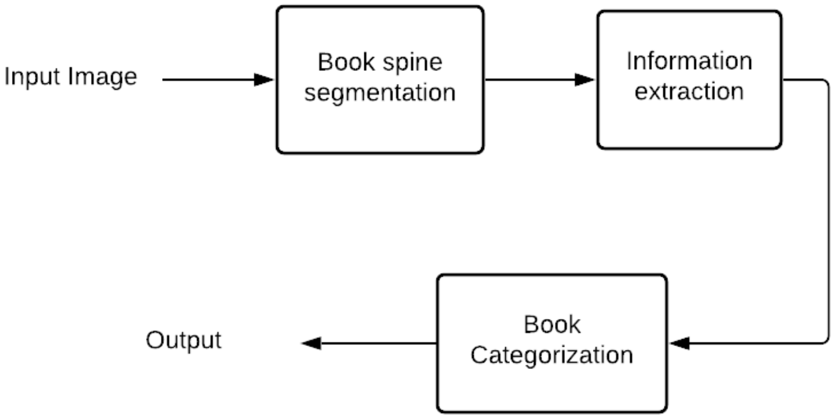


Figure 1.1: System architecture of the proposed framework.

1.3 Difficulties

Many difficulties may arise while developing a automatic book detection system. We also have faced difficulties during the development of this framework. Few difficulties that occurs frequently is given below:

- **Orientation:** Region of interest may be oriented in different ways in an image.
- **Illumination:** Images may be of different illuminations.

- **Noise:** There may be noise in image.
- **Textual styles:** It is troublesome for OCR to recognize texts of distinctive textual styles.

1.4 Applications

There are enormous applications of automatic book detection framework. Some are given below:

- Book recognition can be used to automate libraries, e.g., managing an inventory of books.
- Computer graphics and virtual reality applications may use the segmented book spine images to populate virtual bookshelves with real book spines.
- It can also be implemented in book shops.

1.5 Motivation

Managing huge number of books is quite difficult for human beings. Our proposed system will help librarian to sort book easily and maintaining up-to-date inventory of large set of books. Many methods have been developed to detect only book spine in an image . But Not every method is perfect. So we have proposed automatic book detection method that can efficiently recognize books by reading its title on book spines. So our main motivation of our work is given below:

- To develop an interface to detect books automatically.
- To select the categories of books from a collection of books.
- To extract meta-data of books
- To perform objective and subjective evaluation of the system

1.6 Contribution of the thesis

Research work is conducted to achieve a set of goals whether it is to define a new methodology or to improve the existing ones.In this project, the main focus was given a new methodology for automatic book recognition system.The primary contribution of this thesis is the following:

- We have developed an interface to detect books automatically .
- We have extracted meta-data of books
- We have also developed an interface to select the categories of books from a collection of books

1.7 Thesis Organization

The rest of this report is organized as follows:

- Chapter 2 gives a brief summary of previous research works in the field of automatic book recognition system.
- Chapter 3 gives description of proposed methodology.
- Chapter 4 provides the description of the working data set and analysis of the performance measure for the proposed framework.
- Chapter 5 contains the overall summary of this thesis work and provides some future recommendations as well.

1.8 Conclusion

In this chapter, an overview of our proposed automatic book detection algorithm is given. Along with the difficulties, the summary of the recognition system framework is described in this chapter. The motivation behind this work and contributions are also described here. In the next chapter, background and present state of the system will be provided.

Chapter 2

Literature Review

2.1 Introduction

In recent years there have been done several works for developing ideal book detection and recognition system by overcoming all the challenges. This section provides the brief discussion related to the existing and previous works of book recognition and detection method.

2.2 Related Literature Review

In most of earlier related works, creators centered on a particular portion of the framework. Recognizing books on bookshelves has been considered in many papers. In [1], D. Lee, Y. Chang, J. Archibald, and C. Pitzak develop a framework for matching book-spine images for library shelf-reading process automation. In [2], Quoc and Choi create a system for recognizing books on bookshelves utilizing robots with cameras. They discover individual book spines by identifying the straight lines inside the picture. From the sectioned book spines, they distinguish the content positions utilizing edges and utilize 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. Chen et al. [3] 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. Loechtefeld et al. [4] 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. Crasto et al. [5] show an intelligently bookshelf based on a camera-projector framework where they utilize the color histogram of the book spines to distinguish books. Matsushita et al. [6] 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 [7],they showed a way to recognize books from library bookshelf images. The proposed system consists of many modules like noise removal, detection of bookshelf row and book detection and extraction from bookshelf. Firstly, the noise is removed by filtering and edges are detected. On the image where edge is detected ,the technique counts the the number of pixels of every candidate for horizontal lines and compares the value with a threshold. Horizontal lines segments the image into different portions by extending Elected lines . These individual portions are bookshelf row regions of the bookshelf . Then each of the portion is passed into the book detection module for further Processing.The proposed technique differentiates between the book and non-book regions based on various properties of the shape of the book and detects books arranged in different angular orientation on a shelf. Their proposed technique can extract bookshelf rows and detect individual books from library bookshelf image successfully.

In [8], this paper represents a method on book detection and title recognition from multi-row bookshelf to automate inspection of bookshelves in libraries and bookshops. In this method each books spine are segmented and individual books are recognized from multi-row bookshelf images. At first individual rows are separated by detecting horizontal edges and later separated row images are segmented into vertical lines to segment books into specific book regions. Unwanted objects are removed using region properties by converting book spines into binary images and finally extracts titles from individual book spines. Then the letters and symbols of the title are segmented and extracted by using bounding box and connected component region. Separated character images are matched with the data set images by applying template matching. As a result, their developed new method recognizes the title.

In [9],this paper focuses on the performance of read heads under the off-track reading condition when the reader is under the influence of the recorded magnetization patterns on the medium.It analyzes how the read head behavior is impacted by magnetic field due to various data patterns.The analysis is focused on the micro-magnetic modeling of the state of magnetization in read sensor considering its external magnetic fields due to both the hard bias and the media magnetization pattern. The effects of various magnetization patterns on media are analyzed and then the evaluation of the effect of thermal magnetic agitation of the electromagnetic precession of magnetization happens.

In [10], this paper a method is proposed that segments books on the bookshelves in images that are taken from general viewpoints.Firstly a set of book spine candidates is established by using a shape dependent active contour.A subset of these candidates are selected by using spatial constraints on the assembly of spine candidates. It formulates the selection problem as the maximal weighted independent set (MWIS) of a graph. The segmented book spines can rendered in computer graphics applications or can be used by recognition systems (e.g., library automation).It can also assist users in bookshelf reorganization or to create a bookshelf with a tidier look by modifying the image.

In [11], they focus on automating bookshelf inspection in libraries and bookstores by a novel technique to extract individual book titles from bookshelf images. This technique consists of two steps. Firstly the component, such as book boundary and spine, and the local slant angle are optimally estimated at each horizontal position of a bookshelf image, and then the individual book images are isolated. Secondly the book title is extracted from each book image isolated by the first step. A projection histogram of its edge map is used in this case.

In [12], a technique is proposed to extract the title areas from book cover images. A typical book cover image basically can contain text, pictures, diagrams , complex and irregular background. In addition, the high variability of character features such as thickness, font, position, background and tilt of the text also makes the text extraction task more complicated. So their proposed method is a method of two steps efficient that uses Histogram of Oriented Gradients and color information to find the title regions.Firstly,they perform text localization to find the title candidates. Then refinement process is performed to find the sufficient components of title areas.

2.3 Conclusion

In this chapter we have discussed previously developed method of automatic book detection system. We have come to know about various approaches to spine segmentation, text extraction from an image. We have also found out some method's advantage and flaws.

Chapter 3

Methodology

3.1 Introduction

Automatic book detection and recognition system is a well-smeared problem within the area of computer vision and image processing. any methods have been developed to solve this problem. In this chapter we will discuss about our proposed methodology to solve this problem,experimental setup and it's implementation.

3.2 Diagram/Overview of Framework

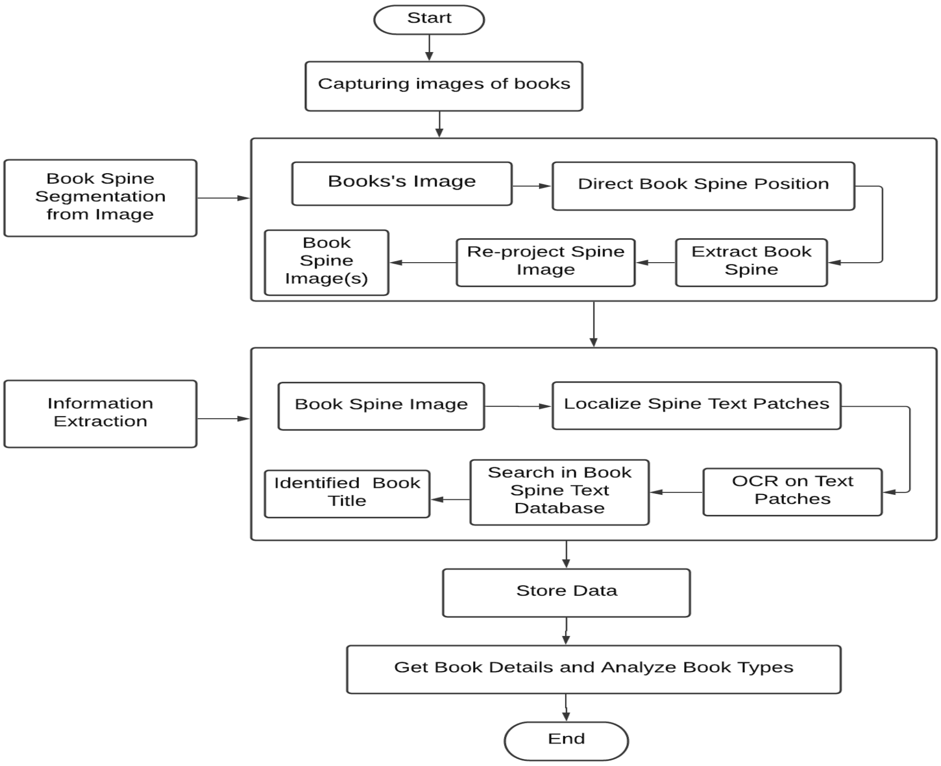

Start
Capturing images of books
Book Spine Segmentation from Image
Books's Image
Direct Book Spine Position
Book Spine Image(s)
Re-project Spine Image
Extract Book Spine
Information Extraction
Book Spine Image
Localize Spine Text Patches
Identified Book Title
Search in Book Spine Text Database
OCR on Text Patches
Store Data
Get Book Details and Analyze Book Types
End

Figure 3.1: System architecture of the proposed framework.

Our proposed framework is depicted in Figure 3.1. The framework is comprised of the following major steps: (1) Book Spine Segmentation (2) Information Extraction (3) Book's Type Categorization.

3.3 Detailed Explanation

3.3.1 Data Pre-processing

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.

3.3.2 Edge Detection

An image processing technique for detecting the boundaries of objects within images is edge detection. It operates by detecting brightness discontinuities. In fields like image processing, computer vision, and machine vision, edge detection is used for image segmentation and data extraction. To detect edges, we used the Canny edge detection algorithm in our proposed process. The stages it has followed are given below.

3.3.2.1 Noise removal

Since edge detection is susceptible to noise in the image, first step of our proposed system is to remove the noise in the image with a 3x3 Gaussian filter.

3.3.2.2 Finding Intensity Gradient of the Image

The image that has been smoothened is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction (G_x)

and vertical direction (G_y). From these two images, we can find edge gradient and direction for each pixel.

3.3.2.3 Non-maximum Suppression

After obtaining the gradient magnitude and direction, the image is fully scanned to eliminate any unnecessary pixels that do not form the edge. For this, each pixel is tested to see whether it is a local maximum in its neighborhood in the gradient direction. Now we will take a look at the picture below:

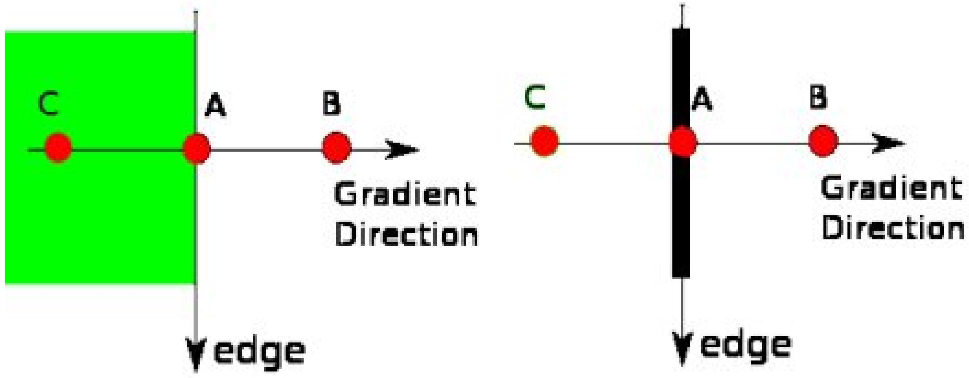


Figure 3.2: Non maximum suppression.

Point A is perched on the precipice of something large (in vertical direction). Normal to the edge is the gradient direction. The directions of points B and C are in a gradient. To see if point A forms a local limit, it's compared to points B and C. If this is the case, it will be considered for the next stage; if not, it will be suppressed (put to zero).

In other words, we'll get a binary picture with "thin edges" as a result.

3.3.2.4 Hysteresis Thresholding

This stage determines which edges are genuine and which are not. We'll need two threshold values, minVal and maxVal, for this. Any edges with an intensity gradient greater than maxVal are certain to be edges, whereas those with an intensity gradient less than minVal are certain to be non-edges, and should be discarded. Based on their connectivity, those that fall between these two levels are known as edges or non-edges. Whether they have a connection to "sure-edge" pixels, they are considered to be part of edges. Otherwise, they are also discarded. See the image below:

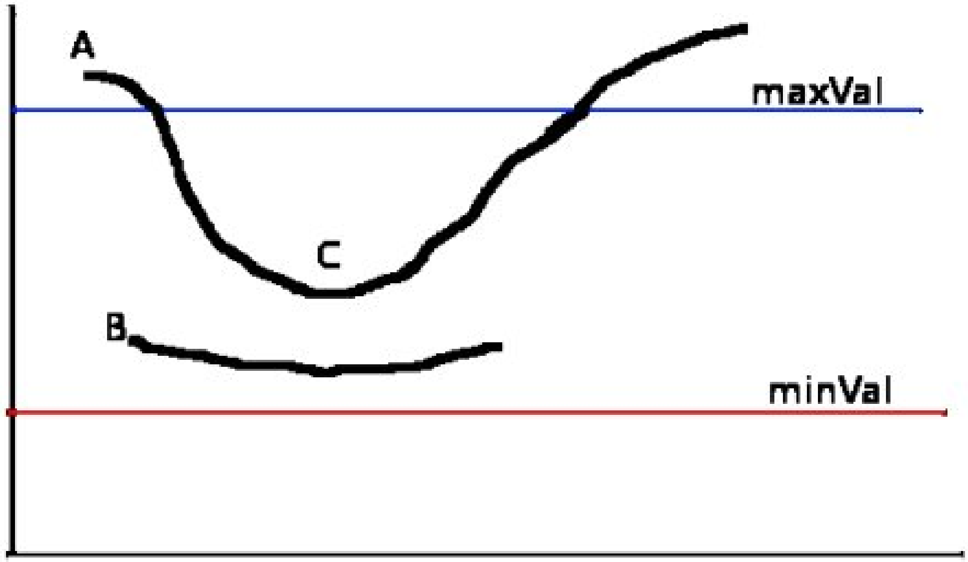


Figure 3.3: Hysteresis Thresholding.

Since edge A is greater than maxVal, it is referred to as a "sure-edge." Despite the fact that edge C is below maxVal, it is connected to edge A, so it is called a true edge, and we get the complete curve. Edge B, on the other hand, is bound to no "sure-edge" and thus is discarded, despite the fact that it is above minVal and in the same region as edge C. As a result, it is important that we choose minVal and maxVal accordingly to get the correct result.

This stage also removes small pixels noises on the assumption that edges are long lines.

So what we finally get is strong edges in the image.

3.3.3 Hough Line Transform

In image analysis, computer vision, and digital image processing, the Hough transform is a feature extraction technique. The technique's aim is to use a voting method to locate imperfect instances of objects within a given class of shapes.

A line can be written as y = mx+c or in parametric form as $\rho = x\cos\theta + y\sin\theta$, where ρ is the perpendicular distance from the origin to the line and θ is the angle created by this perpendicular line and the horizontal axis determined in counter-clockwise rotation. (The direction depends on how the coordinate system is represented.) OpenCV makes use of this representation).

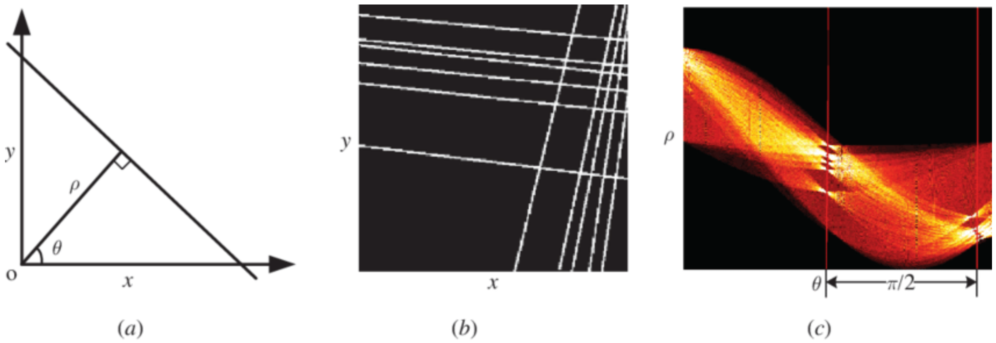


Figure 3.4: Hough line transform

So, if a line passes below the origin, it has a positive rho and an angle less than 180. If it's moving over the root, the angle is less than 180, and the rho is negative, rather than greater than 180. Vertical lines will have a 0 degree angle, and horizontal lines will have a 90 degree angle.

Let's look at how the Hough Transform works for lines. These two terms can be used to represent any line (ρ, θ). So it first creates a 2D array or accumulator (to hold the values of two parameters) that is initially set to 0. Let rows represent the rho and columns represent the theta. The size of the array is determined by the level of accuracy required. If we want the accuracy of angles to be one degree, we will need 180 columns. The maximum distance possible for ρ is the image's diagonal length. Taking one pixel accuracy as an example, the number of rows can be the image's diagonal length.

Let's consider a 100x100 image with a horizontal line running through the center. Take the line's first point. We already know its (x,y) coordinates. Now, in the line equation, enter the values theta = 0,1,2,....,180 and calculate the rho. For each (ρ, θ) pair, increment the value in our accumulator in the corresponding (rho, theta) cells by one. As a result, in the accumulator, the cell (50,90) = 1 along with some other cells.

Take the second point on the line now. Carry out the same steps as described above. Increase the values in the cells corresponding to (ρ, θ) we have obtained. Cell (50,90) = 2 this time. We are actually voting on the (ρ, θ) values. We repeat this process for each point on the line. The cell (50,90) will be incremented or voted up at each point, while other cells may or may not be voted up.

As a result, the cell (50,90) will have the most votes at the end. So, if we search the accumulator for the maximum number of votes, we get the value (50,90), which means that there is a line in this image at 50 degrees from the origin and at an angle of 90 degrees. The presence of bright spots at certain locations indicates that they are the parameters of possible lines in the image.

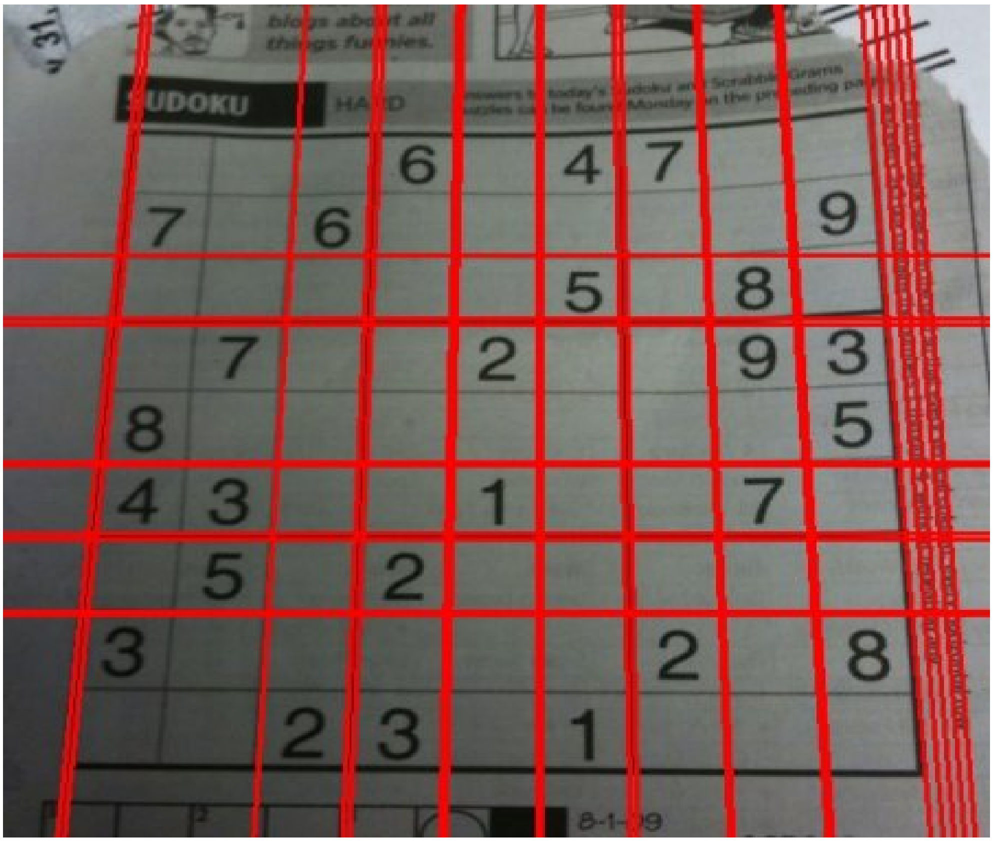


Figure 3.5: Line detected using hough line transform

This is how the hough transform works for lines. It's simple, and we might be able to implement it on our own using Numpy.

3.3.4 Book Spine Segmentation Module

Generally books are organized vertically in bookshelves, so the purpose 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.

First Canny edge detection is performed on image. Then, using an erosion operation, tiny and undesirable regions are removed. As a final point, an image with all vertical edges is obtained, where all vertical lines are indicated as the boundaries of the book spines. Then Hough line transform is applied 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.

The spines of books are then removed 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 line segments.

$$width = x_{i+1} \text{ - } x_i \tag{3.1}$$

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 separated from the original image using the width and length. This segmentation procedure results in a set of segmented book spine images which will be input to the next module.

Algorithm 1: Book Spine Segmentation

```
Result: Segmented individual book spine
begin;
pre-processed image;
compute edges with Canny edge detection;
apply Hough line transform where parameter is canny image, threshold value
  and theta;
draw vertical lines according to houghlines;
according to vertical lines crop images;
for each segmented images do
    Insert them into an array
end
print array;
end
```

3.3.5 Tesseract OCR

Tesseract is an open-source text recognition engine that has been developed by Google since 2006 under the Apache 2.0 license. Tesseract was regarded as one of the most accurate open-source OCR engines in 2006. We can use it directly or through an API to extract printed text from images.The best part of it is that it supports a large number of languages. Tesseract can be made compatible with various programming languages and frameworks by using wrappers. It can recognize text from a board document, or it can be used to recognize text from a single line text image. The visual representation of the Tesseract OCR architecture is shown below.

When it comes to the Tesseract 4.00, it has a text line recognizer configured in its new neural network subsystem. Nowadays, a Convolutional Neural Network (CNN) is typically used to recognize an image containing a single character.

Tesseract treats the input image as a page of text in segments by default. If we want to capture a small region of text from an image, we can use Tesseract's

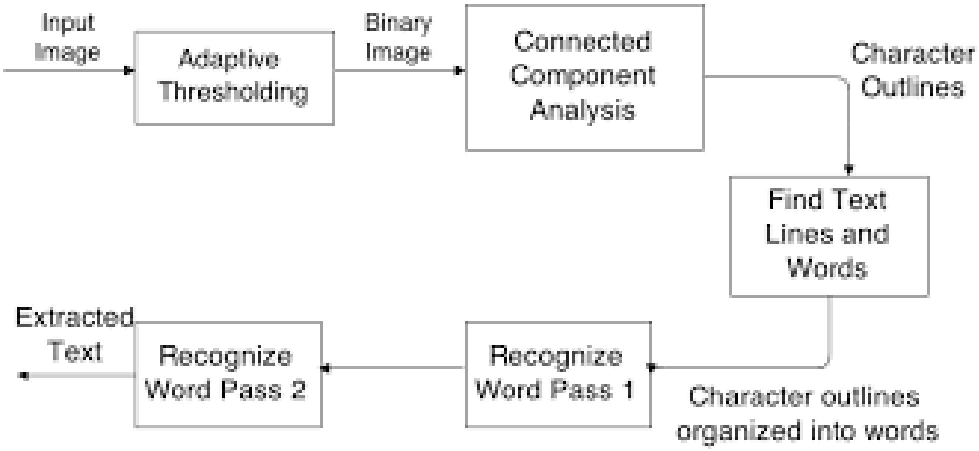


Figure 3.6: Tesseract-OCR-engine-architecture

different segmentations. weu can do this by giving it the –psm mode. Tesseract fully automates page segmentation but does not detect orientation or script.The following are the various Tesseract configuration parameters:

Page Segmentation Mode (–psm): By configuring this, we can tell Tesseract how to split an image into text segments. There are 11 modes for command-line help. We can select the one that best meets our needs.

Engine Mode (–oem): Tesseract has few engine modes, each with a different level of performance and speed. Tesseract 4 has added a new LSTM neutral net mode that works the best.

3.3.6 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 uniform block of text. Tesseract gives the output as a dictionary. This dictionary has content of the input image such as its detected text location, height, width, position information, 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 title from the spine content, the information will be put away in a array. Recognizing book category will be done by analyzing this extracted texts.

Algorithm 2: Text extraction from image

```
Result: Extracted 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 condition then
            append word to a list ;
        else
            do nothing;
        end
    end
end
join all elements of the list;
append the resultant text to another list;
print the list;
end
```

3.3.7 Books Categorization

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.

Algorithm 3: Books Categorization

```
Result: 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
         $book_titleinbyFuzzyStringMatching$if score>max then
            max=score ;
            z=index ;
        else
            do nothing;
        end
        if max>0 then
            extract book's category name from database and store it in a list;
        else
            do nothing;
        end
    end
end
print the list;
end
```

3.3.8 Implementation

In this section we will discuss the implementation step by step that how we implement our system with input images and the outputs. 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 3.7 shows a sample input image.

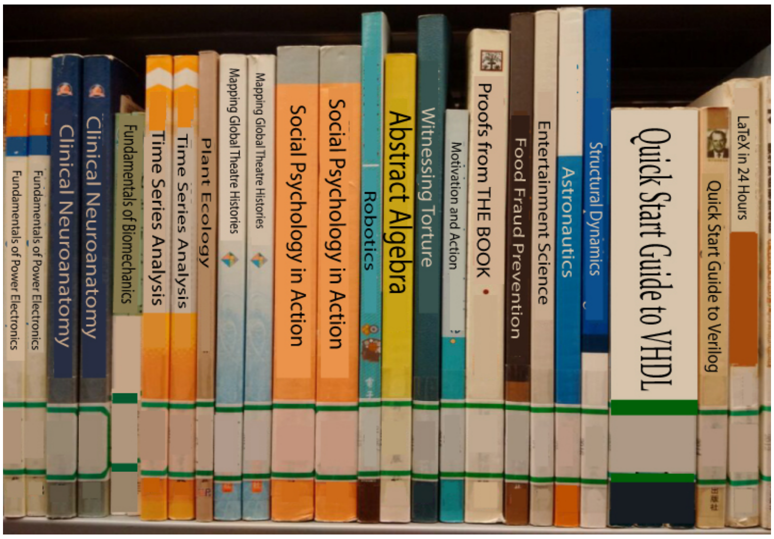


Figure 3.7: Input image

At first image of books is taken by a camera and uploaded it to the system. Vertical lines are the most important feature to distinguish between 2 adjacent books. The first stage is computation of edges using Canny edge detector. Then Hough Transform is used to detect lines that resemble book edges. We draw lines according to hough lines. Figure 3.8 shows vertical boundary lines of the edge detected images.

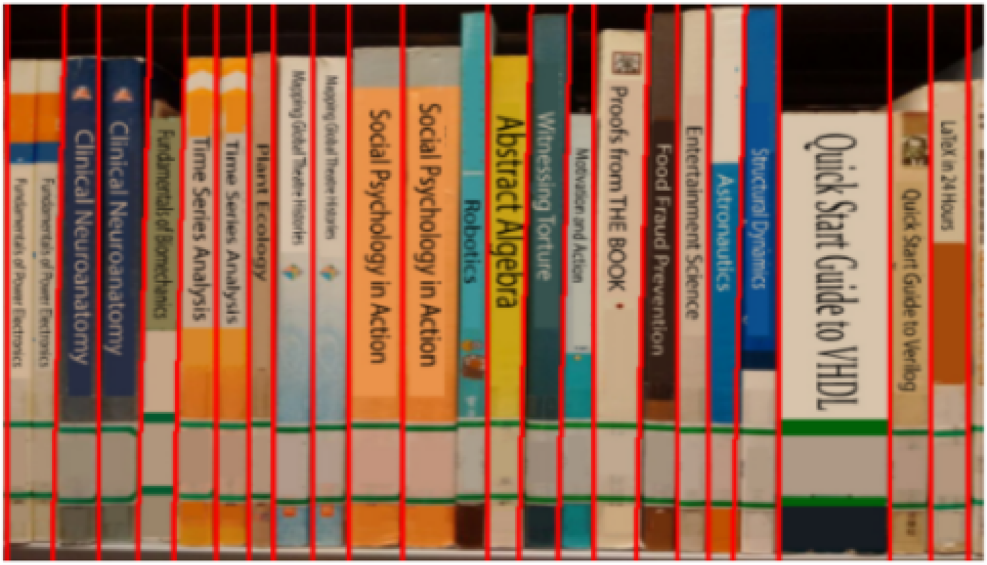


Figure 3.8: Vertical boundary line as detected edge

In the case where a book spine is not fully found due to cropping, we extend the boundary lines to have the same length as the boundary lines of spines that were correctly detected. Then we crop the image according to boundary lines shown in Figure 3.9.

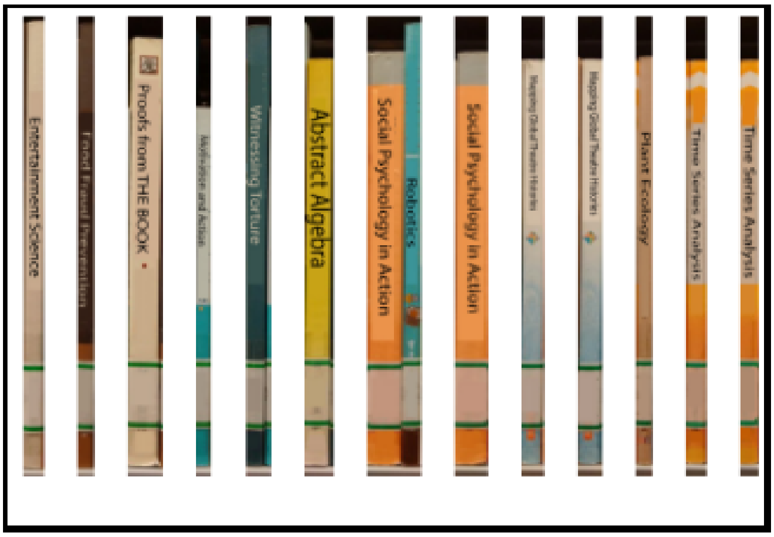
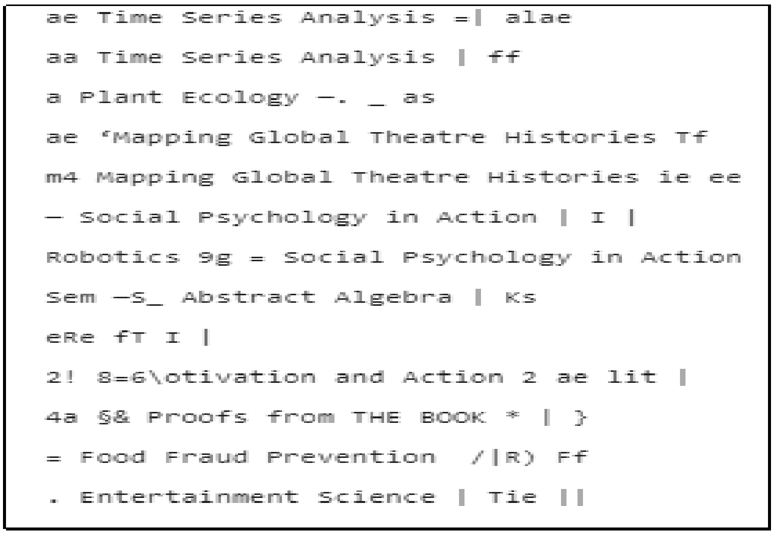


Figure 3.9: Extracted individual book's spine

For the extraction of text, segmented images are passed to an OCR engine for recognition. Words that are recognized from each local text patch, as shown in Figure 3.10, are used as keywords to search a book spine text database.



```
ae Time Series Analysis =| alae
aa Time Series Analysis | ff
a Plant Ecology —. _ as
ae ‘Mapping Global Theatre Histories Tf
m4 Mapping Global Theatre Histories ie ee
— Social Psychology in Action | I |
Robotics 9g = Social Psychology in Action
Sem —S_ Abstract Algebra | Ks
eRe fT I |
2! 8=6\otivation and Action 2 ae lit |
4a §& Proofs from THE BOOK * | }
= Food Fraud Prevention  /|R) Ff
. Entertainment Science | Tie ||
```

Figure 3.10: extracted book's 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.

3.4 Conclusion

This chapter gives a brief overview of our proposed method for book detection and recognition. Our proposed method was tested with various images containing books oriented in horizontally or vertically. The experimental analysis of the proposed framework is discussed in the next chapter.

Chapter 4

Results and Discussions

4.1 Introduction

In the previous chapter, a detailed explanation of the proposed framework for book detection and recognition was given. This chapter examines the performance of the proposed framework. This framework was implemented on a machine having Intel Core i5 processor and 8GB RAM. The input image samples were taken in various illumination condition, book orientation etc. Python 3.6.7(version) is used for developing it. OpenCv libraries are used for image processing and Tesseract OCR is used for text extraction.

4.2 Dataset Description

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

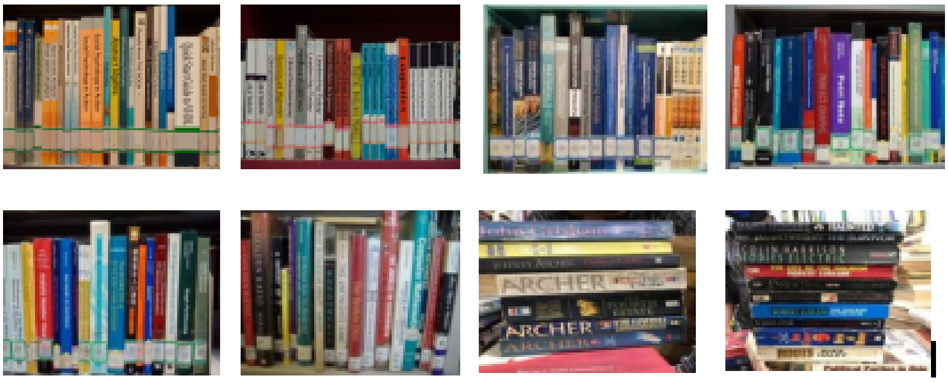


Figure 4.1: Sample input data set

For categorization of books, a database containing book details is used. In total, 145351 samples were collected for analysis. There are many features including book title, author, edition, copyright year, language, category etc. After extracting book title from book spine, title is compared with the each row of this dataset. When the title is matched, system extract the category of that book from the database. If a book's information is missing in the database, the book's data are added to the database.

4.3 Impact Analysis

Our proposed method has immense impact. Our proposed system will lessen human's effort to maintain huge number of books,documents as it is not easy to maintain a lot of books manually.

4.3.1 Social and Environmental Impact

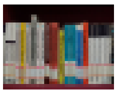
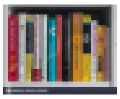
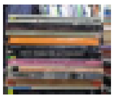
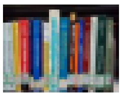
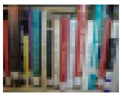
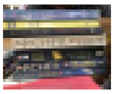
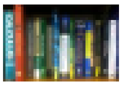
Our proposed method has immense social impact. Our model can be used in library, book shop etc. for managing huge number of books. It can also be used in managing huge official files. Our system will also be very helpful in building virtual library. Our method has no environmental issues.

4.4 Performance Evaluation

We have evaluated our proposed method under different environmental situations and illuminations. We also consider the book orientation whether they are oriented vertically or horizontally. To evaluate the efficiency of our proposed system, we calculate the accuracy with the following equation.

$$Accuracy = \frac{No.of corrected correspondence}{No.of correspondence} * 100\% \tag{4.1}$$

Table 4.1: Performance analysis for proposed system

Sample	No. Of books	Correctly detected books	Book spine segmentation accuracy	Text extraction accuracy	Book type categorization 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%
	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%

4.5 Discussion

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.

4.6 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 [7] only spine segmentation is accomplished. In [8]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.

4.7 Conclusion

In this chapter, we have discussed about the evaluation of performance of our proposed system. We have also discussed the challenges we have faced during the construction of our system. Comparison with other existing frameworks to our system is also discussed in this chapter.

Chapter 5

Conclusion

5.1 Conclusion

In this paper, a new algorithm is introduced for a modern system to recognize 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.

5.2 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. They are described in the following:

- 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.

References

[1] D.-J. Lee, Y. Chang, J. K. Archibald and C. Pitzak, 'Matching book-spine images for library shelf-reading process automation,' in *2008 IEEE International Conference on Automation Science and Engineering*, IEEE, 2008, pp. 738–743 (cit. on p. 5).

[2] N.-H. Quoc and W.-H. Choi, 'A framework for recognition books on bookshelves,' in *International Conference on Intelligent Computing*, Springer, 2009, pp. 386–395 (cit. on p. 5).

[3] D. M. Chen, S. S. Tsai, B. Girod, C.-H. Hsu, K.-H. Kim and J. P. Singh, 'Building book inventories using smartphones,' in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 651–654 (cit. on p. 5).

[4] M. Löchtefeld, S. Gehring, J. Schöning and A. Krüger, 'Shelftorchlight: Augmenting a shelf using a camera projector unit,' in *Adjunct Proceedings of the Eighth International Conference on Pervasive Computing*, Citeseer, 2010, pp. 1–4 (cit. on p. 5).

[5] D. Crasto, A. Kale and C. Jaynes, 'The smart bookshelf: A study of camera projector scene augmentation of an everyday environment,' in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, IEEE, vol. 1, 2005, pp. 218–225 (cit. on p. 6).

[6] K. Matsushita, D. Iwai and K. Sato, 'Interactive bookshelf surface for in situ book searching and storing support,' in *Proceedings of the 2nd Augmented Human International Conference*, 2011, pp. 1–8 (cit. on p. 6).

[7] M. I. Jubair and P. Banik, 'A technique to detect books from library bookshelf image,' in *2013 IEEE 9th International Conference on Computational Cybernetics (ICCC)*, IEEE, 2013, pp. 359–363 (cit. on pp. 6, 27).

[8] N. Tabassum, S. Chowdhury, M. K. Hossen and S. U. Mondal, 'An approach to recognize book title from multi-cell bookshelf images,' pp. 1–6, 2017 (cit. on pp. 6, 27).

[9] Z. Liu, Z. Yuan, C.-L. Ong and S. Ang, 'Influence of data patterns on reader performance at off-track reading,' *IEEE Transactions on Magnetics*, vol. 50, no. 11, pp. 1–4, 2014 (cit. on p. 7).

[10] L. Talker and Y. Moses, 'Independent book spine segmentation,' in *IEEE Winter Conference on Applications of Computer Vision*, IEEE, 2014, pp. 453–460 (cit. on p. 7).

[11] E. Taira, S. Uchida and H. Sakoe, 'Book boundary detection and title extraction for automatic bookshelf inspection,' in *10th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, 2004, pp. 232–237 (cit. on p. 7).

[12] Y. Do, S. H. Kim and I. S. Na, 'Extraction from book cover images using histogram of oriented gradients and color information,' *International Journal of Contents*, vol. 8, no. 4, pp. 95–102, 2012 (cit. on p. 8).