

PAPER • OPEN ACCESS

Image based book cover recognition and retrieval

To cite this article: Kalyani Sukhadan *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042088

View the [article online](#) for updates and enhancements.

Related content

- [Analysis on Main Influence Factors for Ultrasonic Flaw Detection of Steel Weld](#)
Tao Du, Jiandong Sun, Dong Liu et al.
- [Behavioral features recognition and oestrus detection based on fast approximate clustering algorithm in dairy cows](#)
Fuyang Tian, Dong Cao, Xiaoning Dong et al.
- [Development of the Keystroke Dynamics Recognition System](#)
E A Kochegurova, E S Gorokhova and A I Mozgaleva

Image based book cover recognition and retrieval

Kalyani Sukhadan, V Vijayarajan, A Krishnamoorthi and D Geraldine Bessie Amali

School of Computer Science and Engineering, VIT University, Vellore-632014, India

E-mail: vijayarajan.v@vit.ac.in

Abstract. In this we are developing a graphical user interface using MATLAB for the users to check the information related to books in real time. We are taking the photos of the book cover using GUI, then by using MSER algorithm it will automatically detect all the features from the input image, after this it will filter bifurcate non-text features which will be based on morphological difference between text and non-text regions. We implemented a text character alignment algorithm which will improve the accuracy of the original text detection. We will also have a look upon the built in MATLAB OCR recognition algorithm and an open source OCR which is commonly used to perform better detection results, post detection algorithm is implemented and natural language processing to perform word correction and false detection inhibition. Finally, the detection result will be linked to internet to perform online matching. More than 86% accuracy can be obtained by this algorithm.

1. Introduction

As online shopping is developing day by day, people now a days prefer online shopping for getting their favourite books .As the online shopping is developing, still some people likes to see their favourite books at the shop itself. However at some point of time we still wanted to visit the real book store and see the book to get the information. When people visit bookstores, they show keen interest in finding the information about the book in which they are actually interested. Generally, people look for more specifications of the particular book, and while gathering information they are also comparing the prices of the book from the different shops. In this way, we focused at building a GUI that will give the feeling of both real store shopping and online shopping. So, the aim of the project is to give full data of the book to the customer including their comments and prices. This will help people to go for better choices while purchasing books. The next aim of the project is to implement it on the mobile devices and help to be a better reader.

2. Related Work

There are lots of research and development going on this topic from last two decades. Smith.R helps us reviewed the development of tesseract OCR engine [1] which has been very inspiring and it make us to think about OCR combined with necessary pre-processing steps. The work by Patel.C also lead us to study using OCR in detection of neural texts [2] they have compared Tesseract OCR is more better when compared with large image database, and analysed the properties of large image database. This analysis gives us an inspiration to develop more techniques which will further increase the



accuracy of Tesseract OCR. To extract features from images from text recognition several ways are present WT Ho et.al used [3]. An introduction of an idea of using Maximally Stable Extremely Region (MSER) was put forward by Chen H et.al [4]. They have been giving excellent result using this proposed method by using mobile document database. After some online results MATLAB R2016bis giving the whole example on extracting text from natural images. After all the literature research, we decided to use MSER combined with geometric a stroke width filtering to design our algorithm. To convert this book cover recognition system as a centralized one with cloud set up, the work in [6] could be useful. The IR sensor which was used to monitor whether the object is empty or filled used in [7] could help to update the central repository as well the display devices used in book cover recognition system.

3. Algorithms and Methodologies

The below processing pipeline Figure 1, is going to show how we extract the book information given in the cover of the book and can get prices related to it or information related to the book from the internet. First step is to capture the image which is taken from the webcam and use as the input. Next step is to make the book cover horizontal for this some pre-processing methods are conducted. Then, for finding the bounding box MSER method is used. Then the information such as Book title, author name, will be extracted from the book cover by using OCR method. Lastly on the search from the internet prices of the book will be showed and obtained as a keyword. Later we will discuss each part in detail in the upcoming sections

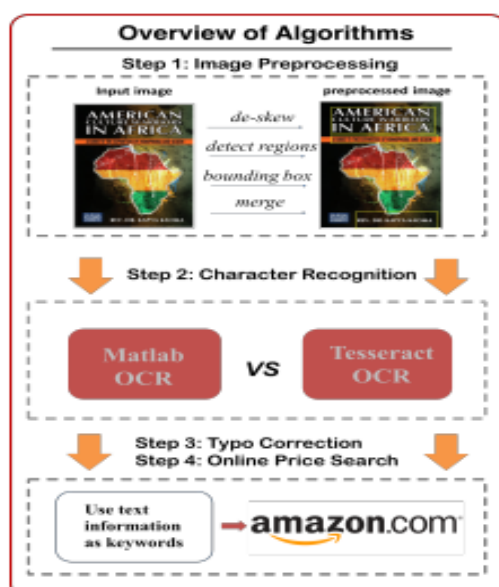


Figure 1. Algorithm Pipeline

In this we are developing a graphical user interface using MATLAB for the users to check the information related to books in real time. We are taking the photos of the book cover using GUI, then by using MSER algorithm it will detect all the properties from the given image, after this it will bifurcate areas which doesn't have text which will be on the basis of morphological difference between text area and non-text area. We implemented a text character alignment algorithm which will improve the accuracy of the original text detection. We will also have a look upon the built in MATLAB OCR recognition algorithm and an open source OCR which is commonly used to provide good results, post detection algorithm is implemented and natural language processing to perform

word correction and false detection inhibition. At last the detection result will be linked to internet to perform online matching.

Pseudo code:

```

Select parameters
Detect candidate text regions using MSER.
    detectmserfeatures ();
    plot (mser region, 'showPixelList', 'true', 'showEllipses', 'false');
    Remove non-text region based on basic geometric properties.
        mserStats=regionProps(mserconncomp 'Bounding
        Box', 'Eccentricity', 'solidity', 'extent', 'Euler', 'Image');
Complete aspect ratio using bounding box data
    Bbox=vertcat(mserstats, bounding box);
Remove region
    mserStats(filterIdx)=[];
    mserRegion(filterIdx)=[];
Remove non-text region based on stroke width variations
    For j=1; num 1(mserStats);
Merge text region for final detection result
    Bboxes=vertical (mserstats.boundary Box);
Expand the boundary box to the small amount.
Clip the bounding boxes to be within the small image bound.
Show the expanded bounding box.
    expandedBoxes_fil=expandedBboxex
filter for the lowest bound
check online.
End();

```

3.1 Image Processing

3.1.1 Image Capture: According to the sources of images from our day to day lives, we implement two ways to capture images. First is to load the image in the database or we can also take an image from the webcam. Another way is to use MATLAB GUI that will automatically preview and take snapshots of book cover images in a natural environment which can be further taken as input. After this, we allow users to give the default parameters in order to perform better results in detections.

3.1.2 Skew Correction: There are some disadvantages of the image which is taken and it will enhance the working of the text recognitions, limitations like geometrical distortion caused by a digital camera or the angle in which the photo is taken. Therefore, we have to correct the rotational problem. There are some edge detection algorithms that we can use such as Sobel edge detection which will find the edges of the book cover, and those edges will occur like in the form of a straight line, then for skew correction we use Hough transform. This will remove the wrong correction from skew correction lines that will be there in the cover of the book. We only take 3 Hough peaks.

3.1.3 MSER: To find the text from the cover of the book, we have to identify parts of the book cover that are having text. MSER is one such algorithm which will help us to detect the useful regions from the image which will basically contain text. Actually, MSER will give the region where we get high image.

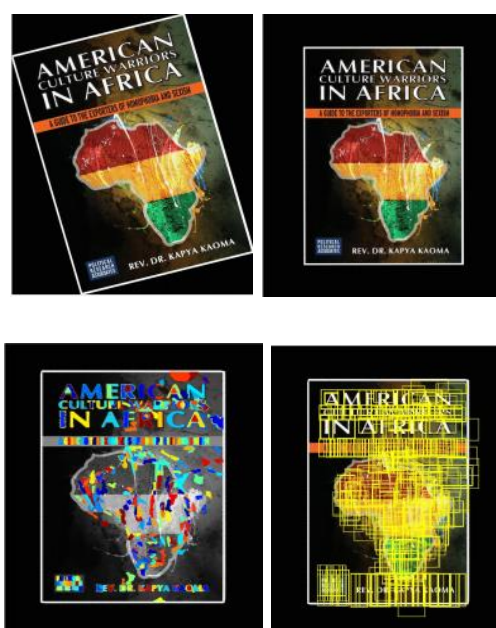
3.1.4 Morphological filtering: The next step after application of MSER is to differentiate between non text region that will be based on geometric difference the properties are: Aspect ratio, Eccentricity, Euler number, Extent and Solidity stroke width is another parameter to differentiate the image between script and non-script region. Positional Filtering: If we take a look at a book cover, the characters are well aligned. So if we are using real character boxes the bounding boxes should be well aligned. Here to group aligned bounding boxes we can use histogram based on their vertical position. Based on the vertical position we used histogram to group aligned bounding boxes. They tend to be low peaks or individual value on the histogram, therefore to filter out non-text features we can set up group threshold.

3.1.5 Bounding Boxes Merging: we will be performing better Optical Character recognition for that we need to perform detection on single text bounding boxes, then we need to combine single character into words. Matlab has implemented an algorithm for combining bounded boxes. But the drawback is it does not take a single text lines in the merging boxes and it can also decrease the overall accuracy for the recognition, so to overcome this we have developed a another method of combining which will allow individual lines inside merged bounding boxes, who is going to increase the recognition accuracy.

4. Optical Character Recognition (OCR)

We are using two different OCR functions for the betterment of optical character recognition function. One is OCR function in Matlab and another one is open source OCR from Google [5].But they have non identical performance across different book cover. In next chapter we will be seeing the detailed comparisons between these OCR functions.

Incorrect detection inhabitation and words auto correction. Because there is a variation in font sizes and styles, so there might be a chance that both OCR functions might get incorrect detection. After the OCR,we seprate out the incorrect detections by taking the word confidence, and then we implement automatic word (typo) correction by getting the English word database and check the words detected with the words in dictionary. The pre-processing result is shown in Figure 2.



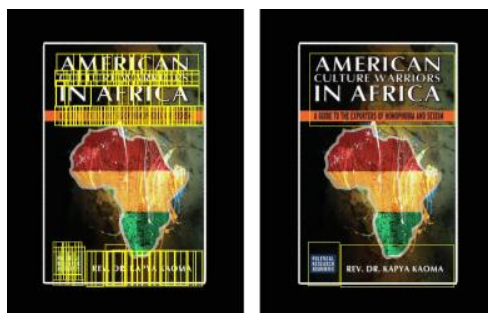


Figure 2. Results for preprocess

(a) input image (b) de-skewed image (c) detect regions with MSER (d) find bounding box (e) remove non-aligned bounding box (f) merge bounding box

4.1. Matlab OCR vs. Tesseract

Tesseract is an optical character engine and is considered as one of the most accurate open source OCR engines. Given below the comparisons of these two methods. The accuracy comparison is shown Table 1.

Table 1. Accuracy Comparison

Accuracy	Test accuracy	Online searching matching
Matlab OCR	57.3%	78%
Tesseract OCR	78.2%	86%

We downloaded 100 test images from Google. This will check the robustness of the system. There should be different design styles of the front book cover. Basically, we can see from the tables (TABLE 1 and Fig.3) that MATLAB OCR lacks of the accuracy compared with Tesseract OCR, time is saved at the time of processing. There might be a chance where we cannot get the exact result. Apart from this if still we cannot extract the same book name, we can still be able to find the details of the book online by using the robustness of the online searching and the information provided. Therefore if the Customer is more concerned about the time, Matlab OCR would be a better option. The time cost factor is shown in Figure 3.

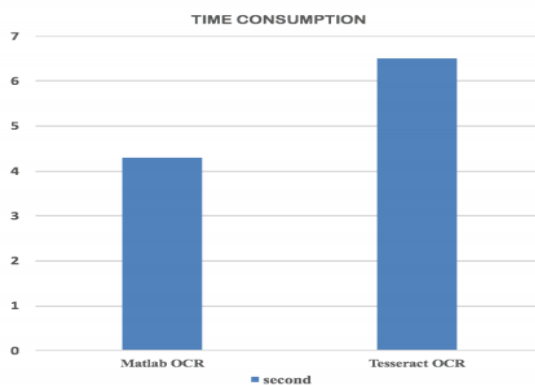


Figure 3. Time cost on Mac Book Pro (Retina, 13-inch, late 2012)

4.2 Robustness to Rotation

Further we need to check two OCR algorithms robustness on rotation, in that we will generate a data set consists of 10 images of the front cover of the book with different rotation angles. So we are more concerned about rotation effect instead of OCR detection effects. The data we chose has better OCR detecting abilities. Further we will calculate the accuracy by counting the total number of correct characters that are outputted by both the algorithms divided by the total no of characters present on the book cover, and the result is shown below. When it comes to rotation it is suggested that both MATLAB built-in-OCR and Tesseract OCR performs relatively robust. While making a comparison the accuracy of open source Tesseract OCR is more related with the rotational angle, whereas the MATLAB built-in OCR is more. The comparison of MATLAB OCR and opensource tesseract is shown in Figure 4.

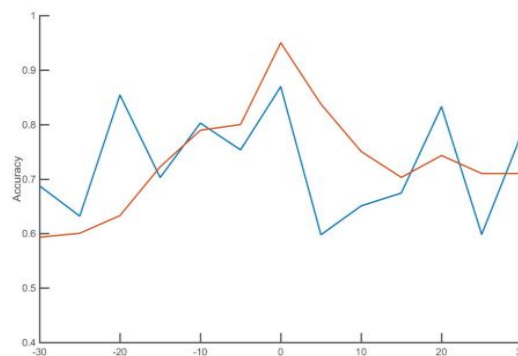


Figure 4. The comparisons between MATLAB OCR and open source Tesseract rotation-invariant. Also we have already specified that Tesseract OCR is slightly better. But in case of rotation MATLAB OCR is slightly robust, so it will be better to put more effort into integrate de-skew algorithm in MATLAB OCR and text recognition algorithm in Tesseract OCR to better the accuracy of OCR algorithm.

4.3 Online Price Searching

In this Project, we have connected to the amazon online website to search the details of the book, this can be more extended to other various online sites and find the average price by price sorting. Following example shows that if the extracted is correct then we can find the book online successfully. (Figure 5 and Figure 6)



Figure 5. Successful online search example-1

4.4 Analysis of unsuccessful test cases

In example Figure 7, we can see the text detection failed and cannot able to detect the information from the give input image and thus cannot find the image online, as is it by default that to detect the book name, it is difficult for the OCR system to separate out the book information. one of the reason behind this is that even if the book front has the better quality after processing. The image may come up with the low contrast because in the background there is the man's picture popping out. The problem with OCR is that it finds difficulty in differentiating the text as well as images. Even if having the separate regions of character, instead of words OCR will detect some single characters

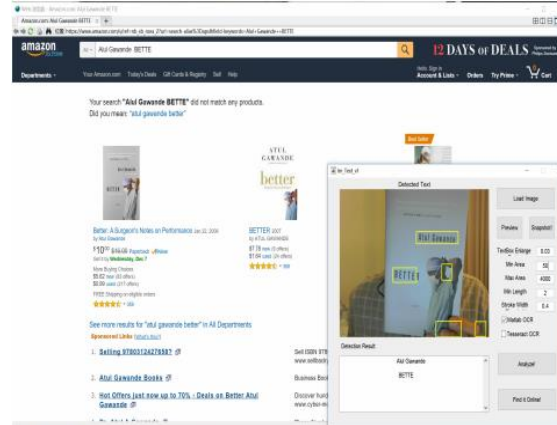


Figure 6. Successful online search example-2

Another failed attempt shown in Figure 7 is for an real-time environment. As we can see in the figure that few of detected bounding boxes reached the edged of the book cover, Since OCR performs script details in bounding box, it will affect the detecting result for the false bounding box this type of problem occur while implementing in 2014b. Because 2014b does not use the graph function. The main idea of using graph function is that it will merge all the lapping bounding boxes and also the bounding boxes which are there outward of the book edge and that can be eliminates by bounding box removal.

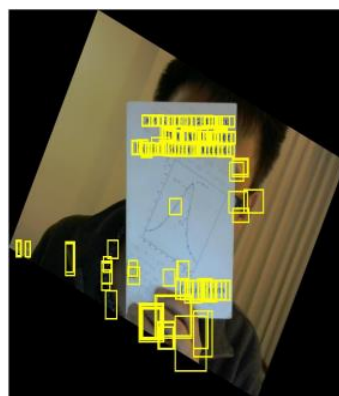


Figure 7. Image cannot be recognized example

5. Conclusion and future work

In this paper, we have developed a MATLAB GUI in which it will detect and recognize the book cover to make the book shopping easier and can take the experiences of both online and offline shopping. With finding the need of adding pre-processes and post-processes ways to make the

performance of OCR function more better, and at the end increases the character recognition. We have built a GUI which is platform-independent, strong, fast and Specific and it will provide good user communication experience. The reason behind developing this project is to give the users a better experience of the book shopping to the users. But eventually it ends up detecting other script based objects such as CD covers and newspapers. In future, the system can be taken further to be applicable on different devices and mobile devices with good flexibility. Even though our text detection systems have good accuracy there is many ways to improve it further. The previous version of the software is not applicable on MATLAB version before 2016A, due to some unavailable inbuilt functions that does not run in the former MATLAB versions. Further we can build our own MSER detection algorithm and bounding boxes merging algorithm. However, it may affect the robustness of the new algorithm as compared to the MATLAB built in functions. We can also make some efforts on to tune the algorithm so will get more compatible to user text detector for all MATLAB versions. Our given approach raises up a number of new ideas, such as Hough rotation, positional bounding boxes filtering and auto word correction. We can also make a comparison between OCR function and open source OCR function in terms of their accuracy. As we talked before their performances varies according to the different datasets. Further we can take the advantage of both of them and try some implementation. Also studied above the text detector will incorrectly recognize the non-text features as text and it cannot be recovered.

References

- [1] Smith R 2007 An overview of the Tesseract OCR engine
- [2] Patel C, Patel A and Patel D 2012 Optical character recognition by open source OCR tool tesseract:Acase study *International Journal of Computer Applications* **55**(10)
- [3] Ho W T, Lim H W and Tay Y H 2009 Two-stage license plate detection using gentle Adaboost and SIFT-SVM. In *Intelligent Information and Database Systems ACIIDS2009 First Asian Conference on IEEE* 109-114
- [4] Chen H, Tsai S S, Schroth G, Chen D M, Grzeszczuk R and Girod B 2011 Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *2011 18th IEEE International Conference on Image Processing* 2609-2612
- [5] Google open source Tesseract OCR <https://code.google.com/archive/p/matlab-tesseractocr/downloads>
- [6] Kannadasan R, Sivashanmugam G, Vijayarajan V, Prabakaran N, Krishnamoorthy A and Naresh K An Integrated Approach for Configuring Hadoop Clusters by Ambari on Horton sandbox
- [7] Kannadasan R, Krishnamoorthy A, Prabakaran N, Naresh K, Vijayarajan V and Sivashanmugam G (2016) RFID Based Automatic Parking System, *Australian Journal of Basic and Applied Sciences* 10(2) 186-191