# A Robust Page Frame Detection Method for Complex Historical Document Images

Mohammad Mohsin Reza[1], Md. Ajraf Rakib[1], Syed Saqib Bukhari[2] and Andreas Dengel[1,2]

[1]*Department of Computer Science, University of Kaiserslautern, Germany*

[2]*Deutsche Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Kaiserslautern, Germany*

Keywords: Document Analysis, Document Pre-processing, Complex Historical Document Analysis, 16th-19th Century German Books, Document Noise Removal, Page Frame.

Abstract: Document layout analysis is the most important part of converting scanned page images into search-able full text. An intensive amount of research is going on in the field of structured and semi-structured documents (journal articles, books, magazines, invoices) but not much in historical documents. Historical document digitization is a more challenging task than regular structured documents due to poor image quality, damaged characters, big amount of textual and non-textual noise. In the scientific community, the extraneous symbols from the neighboring page are considered as textual noise, while the appearances of black borders, speckles, ruler, different types of image etc. along the border of the documents are considered as non-textual noise. Existing historical document analysis method cannot handle all of this noise which is a very strong reason of getting undesired texts as a result from the output of Optical Character Recognition (OCR) that needs to be removed afterward with a lot of extra afford. This paper presents a new perspective especially for the historical document image cleanup by detecting the page frame of the document. The goal of this method is to find actual contents area of the document and ignore noises along the page border. We use morphological transforms, the line segment detector, and geometric matching algorithm to find an ideal page frame of the document. After the implementation of page frame method, we also evaluate our approach over 16th-19th century printed historical documents. We have noticed in the result that OCR performance for the historical documents increased by 4.49% after applying our page frame detection method. In addition, we are able to increase the OCR accuracy around 6.69% for contemporary documents too.

## 1 INTRODUCTION

There has been a resurgence of interest in Optical Character Recognition (OCR) in recent years mainly for digitizing document to increase re-usability of information. Automatic data processing plays a vital role in the processing of huge documents quickly to make our daily life not only easier but also get more benefit to reuse the information and achieve. Currently, the usage of contemporary and invoice type documents is very high in our daily life but it is also clearly noticeable from the last couple of years that people are showing their interest in the digitization of historical documents. By considering such interests, the government is also started to take necessary steps for digitizing this huge archives. Existing commercial OCR systems (like ABBYY[1] and OmniPage[2]) and open-source OCR systems (like OCRopus[3] and Tesseract[4]) have traditionally been optimized for contemporary documents like books, letters, memos, and other end-user documents. However, OCR engines for digitization of historical archives differ in their requirements from such traditional OCR systems mainly because of complex layouts and ancient character style. In addition, OCR systems traditionally have usually been developed for specific scripts and languages, and it is difficult to train them for old scripts that can give high performance. There were some efforts from the researchers to handle the challenging case of historical document images. One of the known open-source OCR system called anyOCR (Bukhari et al., 2017)

[1] https://www.abbyy.com/

[2] https://www.nuance.com/print-capture-and-pdf-solutions/optical-character-recognition/omnipage.html

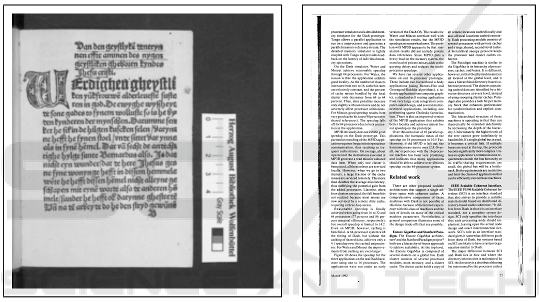
[3] https://github.com/tmbdev/ocropy

[4] https://github.com/tesseract-ocr/tesseract

Reza, M., Rakib, M., Bukhari, S. and Dengel, A.
A Robust Page Frame Detection Method for Complex Historical Document Images.
DOI: 10.5220/0007382405560564
In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, pages 556-564
ISBN: 978-989-758-351-3

is specially developed for Latin script historical documents, but also remarkable for contemporary and semi-structured documents like books, magazines, articles etc. Recently, anyOCR system updated for invoice documents (Reza et al., 2018) by implementing several methods including page frame.

In this paper, we present a page frame detection approach for complex historical document images. Figure 1(a) depicts a complex historical document image whereas Figure 1(b) represents a contemporary document image. The dataset we are using here is quite challenging because of its noisy ruler, damaged pages and other noisy objects around the page boundary. To overcome this problem at first we do the binarization which Bukhari et al. (Bukhari et al., 2017) used in his paper and skew correction (Gari et al., 2017) (Wagdy et al., 2014) of a document image and then apply morphological methods to detect and remove ruler. After removing the ruler we use different approaches to filter noise and detect document content area.



(a) (b)

Figure 1: Historical Vs. Contemporary Image. (a) A sample Historical document image, and (b) A sample Contemporary image.

The rest of the paper is organized as follows. After discussing related work in Section 2, our proposed method discussed in Section 3 for page frame detection for historical documents followed by performance evaluation in Section 4 to compare our result with other systems. Finally, in Section 5 we conclude our work.

## 2 RELATED WORK

Page frame detection for any kind of document is a very challenging task, especially historical document. According to Shafait et al. (Shafait et al., 2008) there are basically two different kinds of noise generate along the border of the document while copying or scanning a book page, one of them is called textual noise (extraneous symbols from the neighbouring page) and another is non-textual noise such as black borders, speckles etc. Textual noise may become a reason of getting an unwanted result after the processing of Optical Character Recognition(OCR). In anyOCR system, Bukhari et al. (Bukhari et al., 2017) are using a method called text-image segmentation that is very effective for removing non-textual noise from a document. Similarly, Shafait et al. (Shafait and Breuel, 2009) are removing textual noise from a document, which is also performing well over contemporary documents. Shafait and Breuel have combined projection profile analysis with connected component removal to implement their method that can recognize the borders of noise areas. However, our target is first to remove both textual and non-textual noise from more challenging historical documents that occur in the page border area and then detect the page frame area based on the content area of the documents. Before implementing our approach, we have investigated a couple of papers to know the contribution of other researchers in the area of page frame detection. So far we do not find enough work related to page frame detection for historical document images but there are few papers available online that mention about page frame detection for general documents.

Shafait et al. (Shafait et al., 2007) applied a method that shows how to detect the page frame with the uses of a geometric matching algorithm that is ignoring the presence of white-space between real page contents and noisy borders. Authors have reduced the error rate of OCR by using this page frame method over UW-III dataset (Phillips, 1996) which is contemporary documents dataset. However, page frame recognition of historical documents is comparatively harder than general documents. So, we have considered historical documents to develop our proposed page frame detection method and finally, we evaluate of this new page frame approach over both historical and contemporary documents of two different datasets.

According to Stamatopoulos et al. (Stamatopoulos et al., 2010), we can detect the page frame of two different page in a single document image. Here authors have used some pre-processing (binarization, noise-removal and image smoothing) technique to get the vertical zones of two pages and after that, they have calculated the horizontal zones of each page. Nonetheless, this work is not exactly like our work but still its related to page frame recognition. So, we have noticed every single document image of this paper and found that the used documents contain less page border noise compare to ours.

As like the work of Stamatopoulos et al. (Stamatopoulos et al., 2010) another page frame method is developed by Bukhari et al. (Bukhari et al., 2012) where they have also taken the help of pre-processing (binarization and text or non-text segmentation) techniques and detect the text lines before applying the straight-line approximation algorithm over the starting and ending points of the ridges to get the page frame regions of documents. DFKI-I dataset (which is camera captured document images) has been used for the evaluation of this page frame approach that contains pages of several technical books. The problem we faced during the analysis of historical document images is still unsolved and this gives us the motivation for developing a new page frame method especially for complex historical document images.

In the paper, Cinque et al. (Cinque et al., 2002) give an excellent explanation about three major problems (print-through, marginal artifacts, and partial extra pages) that we find in most of the documents. They have used a page segmentation method to recognize different sections and the page frame area of the document images. Authors measured the content area of the document image to detect the page frame region of a document image. In the end, the method is tested over UW-III dataset and achieved good results too. However, as our concern about historical documents we planned to test our page frame detection approach over historical documents image dataset, as well as the general documents dataset.

There is an excellent work done by Stamatopoulos et al. (Stamatopoulos et al., 2007) on border detection and noise removal for historical documents by using the combination of projection profiles and connected component labelling process. Authors of that paper mainly work on camera captured historical documents where writers introduced a method that can automatically recognize the document borders and splits not only the black borders but also the noisy text areas from the neighbouring pages. Finally, to verify the recognized noisy text areas authors have also used signal cross-correlation. Nevertheless, the historical documents we have used for the development of this proposed page frame recognition method is more challenging as compared to the used document images by Stamatopoulos et al. (Stamatopoulos et al., 2007).

However, the purpose of this paper is to introduce a method that may help others to work on more complex historical documents which contain multiple page borders as well as other unexpected object shapes like a ruler, knife, and so on.

## 3 PAGE FRAME DETECTION METHOD FOR HISTORICAL DOCUMENTS

From the overview of the related work, a reader can get a very good idea about the contribution of different researchers in page frame detection. With due respect to all the researchers, we want to mention that most of the research has been done based on contemporary documents where the researchers have also found good accuracy in OCR after detecting the page frame and removing the noises from the borders but there is very less work done for historical document page frame detection. If we apply the methods that are developed based on contemporary documents then we get less accuracy in the OCR.

In our related work, we have mentioned an excellent work of historical documents page frame detection method which is introduced by Stamatopoulos et al. (Stamatopoulos et al., 2007). This method can recognize the page border from camera captured documents and remove the noises that occur from the neighbouring page of the document images. Instead of this work, we do not find any other work on historical documents page frame detection. Due to that reason, we have implemented our own method that can remove the noisy ruler or other objects which has a shape like a ruler and after removing that ruler or object our method can recognize the page frame area of the document images which is the goal of our work.

German 16th-19th Century historical documents also contain the above mentioned problem. Removing those noise from a historical document image is not an easy task because that does not handle by other pre-processing methods like binarization or existing page frame detection. The ruler detection is a very challenging task because ruler contains value along with text which also considers as text noise and needs to remove. We have tried different approaches to detect and remove the ruler from images. Our two different page frame approaches are shown in Figure 2. Before detecting page frame, we have decided to use the image binarization and deskewing technique implemented by Bukhari et al. (Bukhari et al., 2017). The binarization technique uses a percentile based binarization method that is suitable for different types of grayscale document images from properly scanned to camera-captured having non-uniform illuminations, low contrast.

Figure 3(a) depicts the view of the original image and Figure 3(b) represents the image after using percentile binarization method of the anyOCR system (Bukhari et al., 2017). After applying the percentile binarization technique we have used the contours function of OpenCV library (Itseez, 2015) to detect the shape of the ruler in the document images and fill with white background. Figure 4(b) shows the detected ruler shape from Figure 4(a) document image and Figure 5 illustrates the document image after removing the ruler with white pixels.

As we already get the document image without the noisy ruler, now we can use text area detection method to get the text area of the document image. Here, we have used another function of OpenCV called morphological transformation (Itseez, 2014) to detect the text area of the document image. We get the outline of each text objects by using the morphological gradient. Then we merge all the necessary text boundary box as column wise that may help us to get the page frame area of the document image. Finally, we get our desired content area and crop the area from the document image for further use of optical character recognition. Additionally, we have also made another approach of page frame detection by using the line segment detector (LSD) (G. Randall and Morel, 2008) where we detect the text boundary line of a document image. Figure 8 and 10 shows the page boundary area over different historical document images and Figure 9 and 11 represents the final output of our page frame detection method.

## 3.1 Noisy Ruler Removal

This is the first stage for our page frame method. We apply our method over a binary image:

$$I(x,y) = \{0,1\}\ 0 \le x < I_x, 0 \le y < I_y \qquad (1)$$

In the noisy ruler removal step, we detect ruler using OpenCV contours (Itseez, 2014). Contours is a curve joining all the continuous points along the boundary, having same colour or intensity. It is a very useful method for shape analysis and object detection. This method basically detects all the shape of text or non-text object.

OpenCV contours perform better over binarize image. Before applying contours, a threshold and canny edge detection (Canny, 1986) should be applied over the binary image. Algorithm 1 is used to find parent rectangles by removing child rectangles from the document image.

A ruler is basically a rectangular object which is laid on the boundary side. We consider rectangle area between 30% to 1% of the document image area and height:width ratio is 10:3 or vice-versa to assume as a ruler. We filter out rest of the rectangles which does not match with the above threshold value. We then make ruler area as a filled white background. The main idea to remove ruler is to avoid text noise inside of the ruler which make easy to detect main text content.

Algorithm 1: Find parent rectangles by removing child rectangles.

```
Data: rect
Result: Find parent rectangles
for $i$ in $1 : len(rect)$ do
    $x_1, y_1, w_1, h_1 = rect[i]$;
    for $j$ in $i+1 : length(rect)$ do
        $x_2, y_2, w_2, h_2 = rect[j]$;
        if $(x_1 < x_2)$ and $((x_1 + w_1) > (x_2 + w_2))$
          and $(y_1 < y_2)$ and $((y_1 + h_1) >$
          $(y_2 + h_2))$ and $(h : w <> 10 : 3)$ then
            remove $rect[j]$;
```

## 3.2 Detect Text-area

In this stage, we use OpenCV morphological transformation (Itseez, 2014) to detect text area of the document image. We do morphological gradient to get the outline of the objects Figure 6(a) and then apply morphological close transformation over the inverse of the document. Morphological close transformation use to closing small holes (Figure 6(b)) inside the foreground objects to make symbols intact, clean and thin object shape as like Figure 6(c).

We used OpenCV contours with kernel size [10,1] to get text bounding area which is shown in Figure 7. We filter out bounding area where the black pixel contains less than 45%, width of the box is W $>$ 90% $>$ 15 and height of the box is H $>$ 50% $>$ 15 (where W is document width and H is document height) to precise text bounding area.

## 3.3 Page Frame Detection using Text Bounding Box

We merge all text boundary box column wise if the X-axis of one boundary box lay on the X-axis of other boundary box and the box area is larger than 5% of the total image area usually not in the page border area.

We may get more than one column boundary box as shown in Figure 8. This column boundary box may contain noisy text area coming from the neighbouring area or part of the main page content area. To distinguish this, we apply some heuristic geometrical calculation to decide whether marge or remove those columns to get the main page frame area. We merge columns if the column width is below 55% of original image width and pixel space between columns is less than 4% of the original image width.

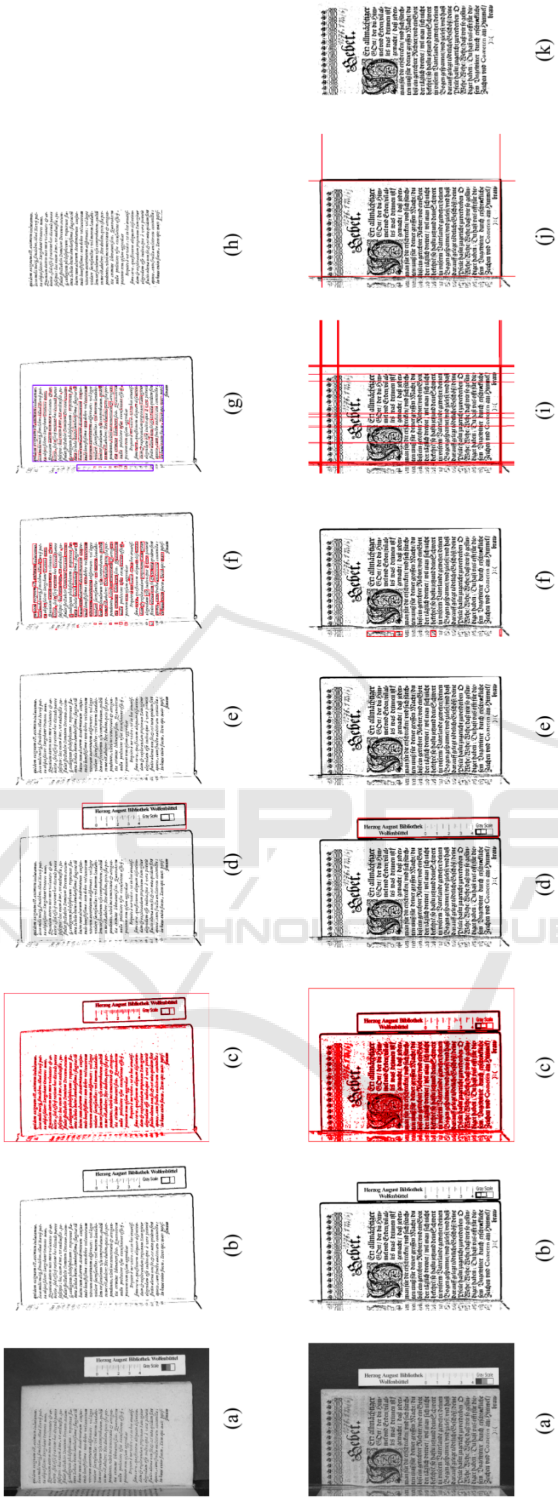
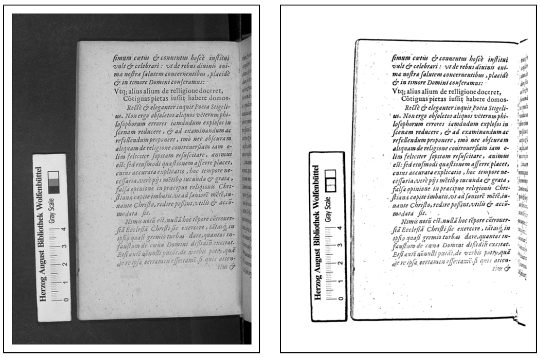
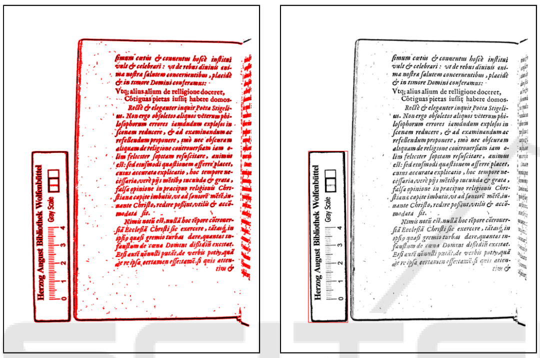


Figure 2: Complete processing steps of our page frame method over two different documents which perform two different approaches. (a) Original historical document image, (b) Image after binarization process, (c) Apply contours to detect the shape of every object, (d) Detect ruler after filtering object area, (e) Remove ruler from the document, (f) Detect text area boundary, (g) Detect page boundary area over the document image, (h) Final page frame by cropping boundary area through Text Bounding Box, (i) Line detection using LSD , (j) Line detection by filtering in border region, and (h) Final page frame by cropping boundary area through LSD.



(a) (b)

Figure 3: Document binarization. (a) Original historical document image, (b) Binarized image.



(a) (b)

Figure 4: Apply OpenCV contours and geometric algorithm to detect ruler of the document: (a) Apply contours to detect shape of every object, (b) detect ruler after applying geometric algorithm.

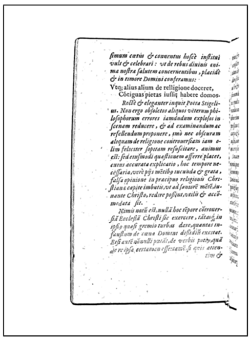
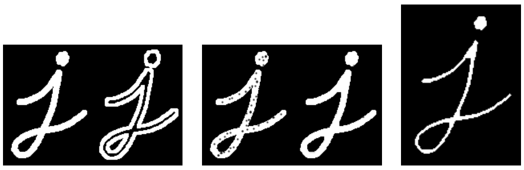


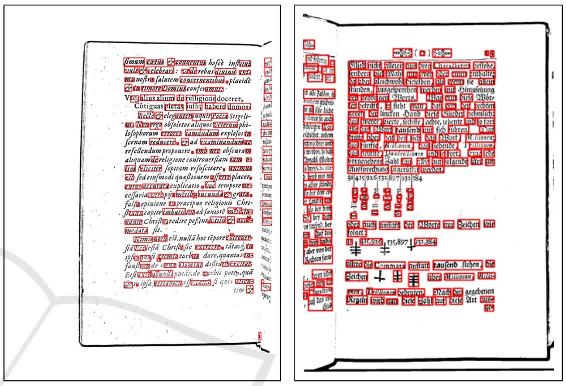
Figure 5: Remove ruler from the document.

After getting page frame area coordinate using this approach, we crop the page to finalize the page frame shown in Figure 9 which is later used for OCR.



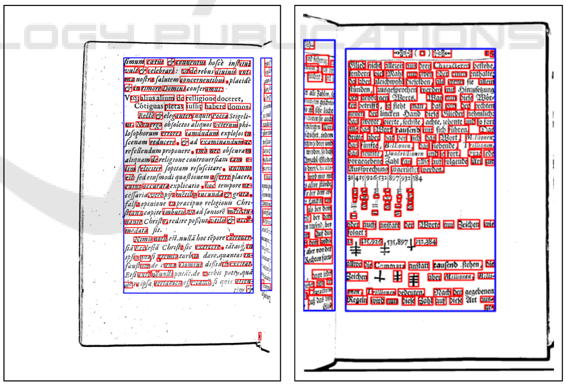
(a) (b) (c)

Figure 6: Apply different morphological transformation over the document to make clean and intact each of the character or objects: (a) morphological Gradient operation gives the outline of the object, (b) morphological Close operation close small holes inside the object, (c) apply both operations over each object to get the clean thin shape.



(a) (b)

Figure 7: Detect text area boundary over two different documents:(a) sample document one, (b) sample document two.

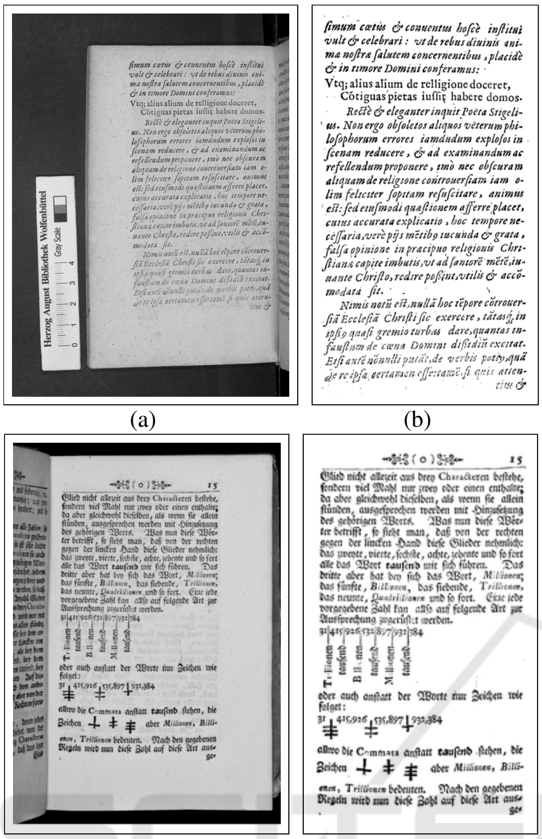


(a) (b)

Figure 8: Detect page boundary over two different documents:(a) sample document one, (b) sample document two.

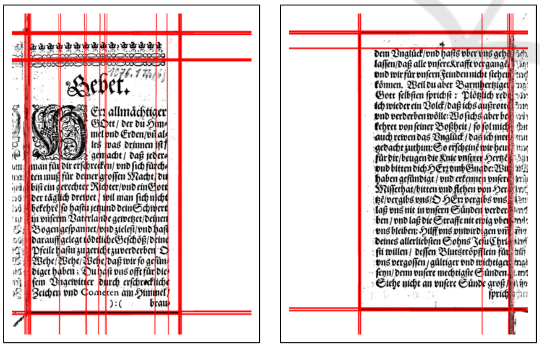
## 3.4 Page Frame Detection using Line Segment Detector

We have also used another approach to detect page frame area. This method applies automatically when our first method does not satisfy. In this method, we detect the boundary line by using the line segment detector (LSD) (von Gioi et al., 2010) which gives



(a) (b) (c) (d)

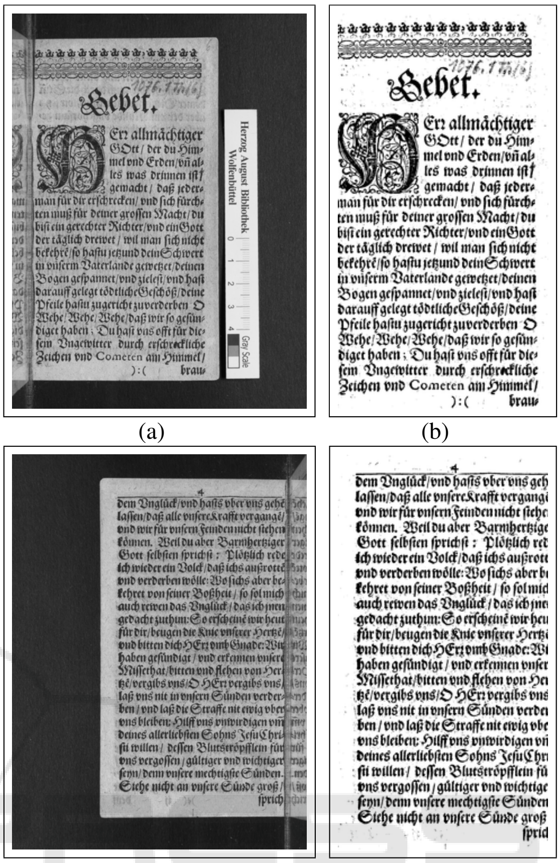
Figure 9: Page frame detection using our text bounding box method: (a)(c) Original image, (b)(d) after applying our method.



(a) (b)

Figure 10: Border line detection over different documents using LSD : (a) a sample document one, (b) a sample document two.

important information about the geometric content of images. Line segments can be used as low-level features to extract information from images. In our approach, we first apply LSD to detect all possible lines in the image. Then we consider the lines which have the length more than 45 pixels and lay down on the border region. The border region considered as 25% from left, 25% from right, 40% from the top and 40% from the bottom. If three lines from left, right, top and bottom found closely (a maximum 15-pixel difference between lines) then we consider it as boundary line as shown in Figure 10. Finally, we crop the image to get page frame which is shown in Figure 11(b) and 11(d).



(a) (b) (c) (d)

Figure 11: Another approach of our page frame detection using Line Segment Detector: (a)(c) Original image, (b)(d) after applying our method.

## 4 EXPERIMENT RESULTS

For the evaluation of our newly introduced page frame detection method, we used 16th to 19th-century OCR-D historical dataset consists of 124 scanned document images. Out of this 124 images we have used 10 images for the training purpose (i.e., parameter optimization) and 114 images for the testing purpose. We have mentioned earlier that this page frame detection method mainly developed for handling the difficulties of historical document images, but we also want to check the performance of this new method over the contemporary dataset. So, we have taken 74

Table 1: Performance evaluation on OCR-D historical documents before and after applying page frame detection method [training images 10 and test images 114].

| **Dataset** | **Before** | **After** |
|---|---|---|
| Train | 78.15% | 84.84% |
| Test | 78.98% | 83.47% |

Table 2: Performance evaluation on UW-III contemporary documents before and after applying page frame detection method [training images 14 and test images 60].

| **Dataset** | **Before** | **After** |
|---|---|---|
| Train | 87.79% | 94.01% |
| Test | 84.75% | 91.44% |

images from UW-III dataset as contemporary documents where we used 14 images for training and 60 images for testing.

There is no existing page frame evaluation method available in the community. Due to that reason, we decided to evaluate by considering OCRed text output performance of the Tesseract OCR system. Tesseract Fraktur language model has used to do the OCR over the OCR-D historical document images. Similarly, English + German language model has used for UW-III document images. We used the Levenshtein distance algorithm to check the OCR accuracy of a generated text file. First of all, we calculate the accuracy of the OCRed text output over the original noisy image and then we again calculate the accuracy over the same images after removing noise by using our page frame method.

We also tried to use ABBYY online OCR system with Fraktur language model but failed to do the OCR for all the training dataset. Surprisingly, only one document performed OCR with Fraktur model and another one document with the combination of English and German language model over original document images. We also used that system after applying the new page frame method and found that only two documents generate the OCRed text with Fraktur and three documents with English and German model. The anyOCR system performs well for Latin typescript documents and it does not contain the Fraktur language model. So, we could not test our page frame detection method with this anyOCR system. Table 1 shows that our page frame method increases the OCRed text accuracy from 78.98% to 83.47% for historical documents. Similarly, it also performed well over contemporary document images by improving the accuracy from 84.75% to 91.44% (Table 2).

# 5 CONCLUSION

In this paper, we proposed an advanced page frame detection method for historical document images. This mechanism is good enough to remove various kind of noises from page boundary. For the evaluation of our implemented page frame detection method, we used different OCR systems to check the accuracy of optical character recognition before and after applying this newly developed page frame approach. We have used two different datasets to test our mechanism. In both cases, we achieved better accuracy after applying our page frame method. Moreover, the proposed page frame detection method is able to improve the OCR accuracy up to 4.49% for the historical and 6.69% for the contemporary document images.

# REFERENCES

Bukhari, S. S., Kadi, A., Ayman, J. M., and Dengel, A. (2017). anyocr: An open-source ocr system for historical archives. In *ICDAR*.

Bukhari, S. S., Shafait, F., and Breuel, T. M. (2012). Border noise removal of camera-captured document images using page frame detection. In Iwamura, M. and Shafait, F., editors, *Camera-Based Document Analysis and Recognition*, pages 126–137, Berlin, Heidelberg. Springer Berlin Heidelberg.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698.

Cinque, L., Levialdi, S., Lombardi, L., and Tanimoto, S. (2002). Segmentation of page images having artifacts of photocopying and scanning. *Pattern Recognition*, 35(5):1167 – 1177. Handwriting Processing and Applications.

G. Randall, J. Jakubowicz, R. G. v. G. and Morel, J. (2008). Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:722–732.

Gari, A., Khaissidi, G., Mrabti, M., Chenouni, D., and Yacoubi, M. E. (2017). Skew detection and correction based on hough transform and harris corners. In *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pages 1–4.

Itseez (2014). *The OpenCV Reference Manual*, 2.4.9.0 edition.

Itseez (2015). Open source computer vision library.

Phillips, I. T. (1996). UW-III english/technical document image database manual. *User's Reference Manual for the UW English/Technical Document Image Database III*.

Reza, M. M., Rakib, M. A., Bukhari, S. S., and Dengel, A. (2018). A high-performance document image layout analysis for invoices. In *DAS2018*.

Shafait, F. and Breuel, T. M. (2009). A simple and effective approach for border noise removal from document images. *2009 IEEE 13th International Multitopic Conference*, pages 1–5.

Shafait, F., van Beusekom, J., Keysers, D., and Breuel, T. M. (2007). Page frame detection for marginal noise removal from scanned documents. In *SCIA*.

Shafait, F., van Beusekom, J., Keysers, D., and Breuel, T. M. (2008). Document cleanup using page frame detection. *International Journal of Document Analysis and Recognition (IJDAR)*, 11:81–96.

Stamatopoulos, N., Gatos, B., and Georgiou, T. (2010). Page frame detection for double page document images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 401–408, New York, NY, USA. ACM.

Stamatopoulos, N., Gatos, B., and Kesidis, A. (2007). Automatic borders detection of camera document images. In *2nd International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil*, pages 71–78.

von Gioi, R. G., Jakubowicz, J., Morel, J., and Randall, G. (2010). Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732.

Wagdy, M., Faye, I., and Rohaya, D. (2014). Document image skew detection and correction method based on extreme points. In *2014 International Conference on Computer and Information Sciences (ICCOINS)*, pages 1–5.