Text Segmentation and Binarization using the Difference Theoretic Texture Features

Major Project Report Submitted in Partial Fulfillment of the Requirements
for the Award of the degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEM

SUBMITTED BY

KM RACHNA DEVI

(2K12/ISY/14)

UNDER THE GUIDANCE OF

Dr. SEBA SUSAN

(ASSISTANT PROFESSOR)

(Department of Information Technology)

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

June 2014

CERTIFICATE

This is to certify that the project report entitled "***Text Segmentation and Binarization using the Difference Theoretic Texture Features***" is being submitted by ***Km. Rachna Devi (2K12/ISY/14)*** in the partial fulfillment of the requirements for the award of ***"Master of Technology degree in Information System"*** is submitted to the department of Information Technology, Delhi Technological University, New Delhi; is the original work carried out by her under the guidance and supervision.

This project report was not submitted earlier at any other University or Institute for the award of any degree or diploma.

(Project Guide)

Dr. Seba Susan

Assistant Professor

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042.

ACKNOWLEDGEMENTS

First, I would like to express my special gratitude to my family for their constant encouragement and support that helped me in the completion of this project.

I would like to pay my deepest and sincere gratitude to my project guide ***Dr. Seba Susan,*** Assistant Professor, Department of information technology, **Delhi Technological University**, Delhi; for her invaluable support, guidance and continuous inspiration and giving me the opportunity to work on this topic. It would never be possible for us to take this project to this level without her innovative ideas and her relentless support and encouragement.

I am highly thankful to **Professor Dr. O. P. VERMA**, H.O.D., Department of Information Technology, Delhi Technological University, Delhi, for the support provided by him during the entire duration of the degree course and especially in this report.

I am thankful to all the people who are active in this field.

Km. Rachna Devi

Roll No.: 2K12/ISY/14

M. Tech (Information System)

Delhi Technological University

E-mail: rkgit.rachna0502@gmail.com

ABSTRACT

Text Segmentation and Binarization is a very difficult and challenging task in case of unclear document images because of the difference between text and background of the text. Threshold play a vital role for text binarization because it separates the foreground and background of images and very difficult to find a good threshold. Many approaches have been available to find the threshold for text segmentation and binarization. All of them have its own advantage and disadvantage. In this project report, an approach has been proposed to text segmentation and binarization from challenging and sophisticated document images using difference theoretic texture feature. In this proposed algorithm, the adaptive threshold, contrast stretching and gamma correction have used for enhancement and binarization of document images. An adaptive global threshold algorithm merges with contrast stretching and gamma correction to enhance the document image.

The difference theoretic texture feature is used for text extraction from an image. The difference theoretic feature technique has also used to segment text area from document images. This technique provides better results for all unclear and clear images as compare to other techniques. The proposed technique has been implemented using MATLAB 2012. This proposed technique has been tested on five recent document image binarization contest (DIBCO) 2009, 2011 and 2013 and handwritten-DIBCO 2009, 2010, 2011 and 2013 datasets [1] [2] [3] [4] and achieves performance 81.22%, 82.86%, 85.10%, 83.40%, 76.04% respectively. It is also tested on many different challenging document images which is captured by mobile phone. This technique provides better results for all unclear and clear images as compared to other techniques.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

Text segmentation and binarization is an important part in document image processing that is a subfield of digital image processing. The main objective of document image processing is to detect text area in the image. For creating the Algorithm "*Text Segmentation and Binarization using Difference Theoretic Texture Feature* ", studied different research papers and book of image processing. The keywords used in this algorithm are: Otsu's Threshold Method [5], Gamma Correction [6], Contrast Starching [6], Difference Theoretic Texture Feature Technique [7], and some other concept of digital image processing. The document image can be segmented into various area such as Line segmentation, Word segmentation, Character segmentation and text area segmentation. The proposed algorithm is used for binarize and text area segmentation from any type of document images. In proposing technique we focus on the binarization of grayscale document image because in most cases information has lose when we convert grayscale to binary image. Therefore, most of the work has been developed to work on binary images like that Otsu's Threshold [5], Yanowitz's Method [8] [9], Niblack's Method [10] [11], Sauvola's Method [12], Bernsen's method [13] and Post-Yanowitz's Method [14]. To test the proposed technique against to previously exist technique, the predefined datasets of Document Image Binarization Contest (DIBCO) and Handwritten-DIBCO [15] [16] [17] [18] have used. This chapter, discuss about the text segmentation and binarization and some better method of text binarization.

1.1 Overview of Text Segmentation and Binarization

Today is very common that the document image embedded with text and some other things like pictures and so on. Text Segmentation is the process of recognizing the text appearing in document images. Text extraction in images has been used in a large variety of applications such as mobile robot navigation, document retrieving, object identification, vehicle license plate detection, etc. [19]. There are so many popular techniques available for text extraction, such as discrete wavelet transform (DWT) [20] and Haar wavelets [21]. There are two ways for text segmentation in document image, namely region based approach and texture based approach. In the region based approach, each pixel in the image is considered and assigns it to a particular region or object [22]. This approach is basically divided into two subcategories: edge based [23] [24] [25]and connected component based. They are relatively independent of changes in text size and orientation, but having difficulties with complex images with non-uniform backgrounds, for example, if a text string touches a graphical object in the original image, they may form one connected component in the resulting binary image [22]. The main aim of segmentation is to partition the document image into various homogeneous regions such as text block, image block, line and word [26]. In the propose approach, the texture based technique used to extract the text area.

In many cases for improving the quality of text in document image we need to binarize the image. The meaning of Text Binarization of document image is to segment the text (foreground information) from the background of the document. Binarization of document images is a critical step in many document processing workflows, and at the same time it is a good example of a complex analysis problem [15] [16]. For text segmentation and binarization, the best known technique is thresholding. Thresholding is a method to extract foreground information from background. It is used to determine whether a particular pixel belongs to the background and foreground. In image if the pixel value is greater than the threshold value, then it is known as foreground pixel otherwise it is known as background pixel.

There are two types of threshold methods: Single Threshold and Multilevel Threshold. Single Level Threshold can produce binary images (0, 1) (image contains only two pixel value: 0 and 1. In other hand, Multilevel Threshold can produce an image with a range between 0 and 255 of pixel values [17] [18] [27] [28] [29]. There is a different way to use threshold: global and variable thresholding. In Global thresholding [6], the threshold value is constant applicable over an entire image. But in Variable thresholding [6], threshold change over an image. The variable thresholding in which the value of threshold at any point in the image depends on the properties of a neighborhood of that point, called as Local (Regional) thresholding [6]. The variable thresholding in which the threshold value depends on the spatial coordinates themselves is known as dynamic thresholding [6]. Local Threshold is better as compared to Global Threshold. We can calculate the value of threshold by using the average intensity of the image, the histogram of the image and so on. There are different techniques have been proposed through the year for document binarization. The segmentation and binarization technique has also designed and this is better as compare to other techniques. The proposed algorithm removes background information and binarized only text area from document image. Many others algorithm are available to binarize document images. The well-known image binarization algorithms are Yanowitz, Niblck, Sauvola and Post-Yanowitz threshold. These algorithms provide foreground information completely, but they can't able to remove background noise completely when there is no difference between the foreground and background of an image.

1.2 Threshold Techniques

Thresholding is very first step in preprocessing for poor quality images to eliminate the background, foreign bodies. Preprocessing requires for enhancement of contrast between dark and light pixel value to improve the quality of the image. Threshoding are used for separating background information from foreground. The binarization methods of grayscale document images can be divided into two main categories: Local Binarization and Global Binarization. The very best example of Global binarization is the Otsu threshold method [5] that tries to find one threshold value for complete image. The global binarization method is best in the case of scanned documents.

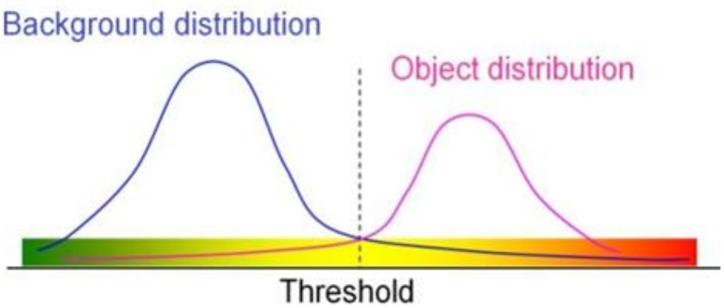


Figure -1.2 Separation of object from background using threshold

1.2.1 Single level Thresholding

In single threshold, we select a threshold value T that change the image into black and white. In segmentation phase, threshold plays a very important role. For local segmentation single level threshold methods were becoming well. Local binarization are able to provide good results in case of old and degrade document image because it computes thresholds individually for each pixel. Suppose that the intensity histogram of image f (x, y) composed of light objects on a dark background in such a way that object and background pixels have intensity values groups into to dominate modes [6]. To extract the object (Text) from the background is to select a threshold T. Any point (x, y) in the image at which $f(x, y) > T$ is called an ***Object Point;*** otherwise, the point is called a ***Background Point*** [6]**.** In mathematical form, the segmented image $g(x, y)$ using a single threshold is given by:

$$g(x,y) = \begin{cases} 1, & if \;\; f(x,y) > T \\ 0, & if \;\; f(x,y) \le T \end{cases} \tag{1.2.1}$$

1.2.2 Multilevel Thresholding

Multilevel thresholding methods were becoming well because a single threshold is not always suitable for global segmentation. In case of Multilevel Threshold, more than one threshold value is used for image segmentation [30]. Multilevel Threshold segment an image called f (x, y) by classifying a point (x, y) as belonging to the background if $f(x, y) <= T_1$, to one object class if $T_1 < f(x, y) <= T_2$ and to another if $f(x, y) > T_2$. The Multilevel Threshold condition with segmented image g (x, y) is given by:

$$g(x,y) = \begin{cases} a, & if \quad f(x,y) \leq T1 \\ b, & if \quad T1 < f(x,y) \leq T2 \\ c, & if \quad f(x,y) \leq T1 \end{cases} \tag{1.2.2}$$

In above equation a, b and c are distinct pixel value in the image.

1.2.3 Multi-Threshold

The Multi Threshold method uses a series of threshold values and computes the total number of blobs or objects in an image for each threshold. The peak threshold values are those with the highest total number of blobs as compared to their threshold neighbors [30].

1.2.4 Otsu's Method for Thresholding

Otsu's method of image segmentation is one of the best methods for threshold selection. Normally, the Otsu's method found the optimal threshold in an image by maximizing the between-class variance of pixel intensity with an exhaustive search. However, with an increase of the number of classes in an image, this method becomes rather inefficient because it requires a large number of iterations to compute the cumulative probability (zeroth-order moment) and the mean (first-order moment) of a class [5]. Otsu method tries to find a single threshold value of image and this threshold is used to assign a pixel value to foreground or background. Otsu method is based entirely on computation performed on histogram of an image easily obtainable in a one dimensional array. It is used to automatically perform histogram shape-based image thresholding or the reduction of a gray level image to a binary image [6]. Algorithm of Otsu's Method is described below-

1. Read any image.
2. Convert it into gray image.
3. Compute the histogram of an image.
4. Compute the normalized histogram (histogram in one dimension) of an image and component of histogram by using $Pi = ni/MN$ where i = 0, 1, 2, 3… L-1.

   Pi=histogram component

   ni=number of pixel with intensity i.

   MN=size of image. And L-distinct intensity level.
5. Calculate the component of the histogram.
6. Compare the intensity value of image with threshold value t and store in c1 and c2.

$$g(x,y) = \begin{cases} c1 = f(x,y), \ if \ \ f(x,y) > t \\ c2 = f(x,y), \ if \ \ f(x,y) \leq t \end{cases} \tag{1.2.3}$$

7. Compute the average intensity of the entire image (global mean)

$$mg = \sum_{i=0}^{L-1} i(Pi) \tag{1.2.4}$$

8. Compute the probability p1 using threshold that a pixel is assigned to class c1.

$$p1 = \sum_{i=0}^{t} Pi \tag{1.2.5}$$

9. Compute the probability p2 using threshold that a pixel is assigned to class c2.

$$p2 = \sum_{i=t+1}^{L-1} Pi \tag{1.2.6}$$

10. Compute mean intensity value m1 of the pixels assigned to class c1.

$$m1 = \sum_{i=0}^{t} i(Pi) \tag{1.2.7}$$

11. Compute mean intensity value m2 of the pixels assigned to class c2.

$$m2 = \sum_{i=t+1}^{L-1} i(Pi) \tag{1.2.8}$$

12. Compute mean m (average intensity) up to the threshold value t.

13. Compute between class variance.

$$mb = p1(m1 - mg)^2 + p2(m2 - mg)^2 \tag{1.2.9}$$

14. Calculate the optimal threshold value l.

$$mo(l) = \max_{0 \le i < L-1} mb(i) \tag{1.2.10}$$

15. Compare the optimal threshold value with intensity value of image

$$g(x,y) = \begin{cases} g = 1, & if \;\; f(x,y) > l \\ h = 0, & if \;\; f(x,y) \le l \end{cases} \tag{1.2.11}$$

CHAPTER-2

LITERATURE REVIEW

In this chapter, some popular related works are discussed. Many numbers of researchers have proposed an algorithm to binarized the images. These algorithms are also known as a threshold algorithm because a threshold is required to convert an input image into a binary image. These local threshold techniques estimate a different threshold value for each pixel according to the grayscale information of the neighboring pixels.

2.1 Yanowitz and Bruckstein [8] [9]

Yanowitz and Bruckstein suggested using the gray-level values at high gradient regions as known data to interpolate the threshold surface of image document texture features. In this method, a threshold surface is constructed by finding the edge points of the smoothed image. The gradient magnitude image is computed and thinned to one pixel-wide line to identify edge points. An iterative interpolation process is employed to get a smooth surface passing through the edge points. The constructed surface is used to threshold an image [31].

In an iterative interpolation process, the interpolated surface is set at image gray scale at the edge points and 0 at the other points. The technique uses adaptive thresholds and the value of that threshold is calculated through the combination of edge analysis, processing with gray level information and the design of the interpolated threshold surface. After that the image is binarized using that threshold value. In this method we calculate threshold value at any pixel by using the following equation:

$$T(x,y) = m(x,y) \times (1 - \frac{k}{100}) \tag{2.1.1}$$

Where T (x, y) is the threshold at pixel (x, y), m (x, y) is the local mean and k is used to determine how much of the total print object boundary is granted as a part of the given object. The main point of Yanowitz's method is, to make a threshold surface depending on the edge point's condition. So the threshold value can change according the gray scale surface of the original image. But in this method, it is very difficult to select edge points. If the edge point cannot select correctly, the threshold surface cannot be constructed properly, otherwise they will contribute much to the threshold surface construction.

2.2 Niblack's Method [10] [11]

It is a local thresholding method based on the calculation of the local mean value and of local standard deviation. In this method we calculate threshold value at any pixel by using the following equation:

$$T(x,y) = m(x,y) + k \times s(x,y) \tag{2.2.1}$$

In this equation T (x, y) is the threshold at pixel (x, y), m (x, y) and s (x, y) are respectively the local mean and standard deviation of the local area. k is used to determine how much of the total print object boundary is granted as a part of the given object. This method provides the facilities to separate object from background of that object. But through this method we cannot remove background noise, situated far away from the objects.

2.3 Sauvola local thresholding [12]

This method is better for classical document images. The Sauvola method for local binarization does quite well, and basic idea of Sauvola is that there is a lot of local contrast, the threshold value should be selected close to the mean value, by an amount proportional to the normalized local standard deviation. The proposed technique uses rapid image surface analysis for algorithm selection and adaptation according to the document contents. The contents are used to select the algorithm type and need for parameterization, if any, and to compute and propose the threshold value for each or every nth pixel (interpretive approach).

The document content is used to guide the binarization process: a pictorial content is subjected to a different type of analysis than a textual content. The degradations, such as illumination and noise, are managed within each algorithm structure to effectively filter out the imperfections. The results of the thresholding processes are combined with a binarized image that can either use a fast option, i.e. to compute binarization for every nth pixel and interpolate the threshold value for the in between pixels, or a pixel by pixel option that computes a threshold value for each pixel separately.

In this method we calculate threshold value at any pixel by using the following equation:

$$T(x,y) = m(x,y) \times (1 + k \times \left({}^{s(x,y)}/_{R} - 1 \right)) \qquad (2.3.1)$$

In this equation T (x, y) is the threshold at pixel (x, y), m (x, y) and s (x, y) are respectively the local mean and standard deviation of the local area. k is used to determine how much of the total print object boundary is granted as a part of the given object. R is the maximum of standard deviation s (x, y).

2.4 Niblack's Method with Post-Processing using Yanowitz [14] [9]

This threshold technique is the combination of two techniques: one is Yanowitz and Bruckstein's method and the second is a Niblack's method. This method provides better results as compare to Yanowitz's method and Niblack's method because Yanowitz's method retrieve all text, but not able to remove noise and Niblack's method is best for removal of noise. So the resultant of both Yanowitz and Niblack provides better results. This method removes the most irrelevant things. In other words Niblack's method with the addition of the postprocessing step of Yanowitz and Bruckstein's method added performed the best, and was also one of the fastest binarization methods [32]. These are the following steps of the post-yanowitz algorithm:

1.) Take an input image.
2.) Apply average filter of (3×3) on input image to remove noise and make it smooth (image1).
3.) Sobel's edge operator [33] is used to calculate the gradient magnitude image of the image1 [34].
4.) Select a threshold value (TP). There is no automatic method to specify TP for a given image, so it is specified by trial and error.
5.) Calculate the average gradient of the edge pixels (the pixels that are 4-connected to the background) for all 4-connected components and remove those components that having an average edge gradient below the threshold TP [34].

CHAPTER-3

RELATED CONCEPTS

This chapter reviews the details about concepts that used to design propose algorithm. These concepts are Gamma Correction, Contrast Stretching, Difference Theoretic Features and Performance Measure.

3.1 Gamma Correction

With Gamma Transformations, you can curve the grayscale components either to brighten the intensity (when gamma is less than one) or darken the intensity (when gamma is greater than one) [6].

The gamma correction can also be defined by using the following expression [6]:

$$G1 = c \times G^{\gamma} \tag{3.1.1}$$

In the above equation, G1 (non -negative) is the output and G (non negative) is the input value. c is a positive constant value and in many cases the value of c is 1. γ is a gamma value. If the value of gamma is greater than one ($\gamma > 1$), increase contrast and if the value of gamma is less than one ($\gamma < 1$) it reduce contrast. If you notice, different display monitors display images at different intensities and clarity [35]. That means, every camera has built-in gamma correction with certain gamma ranges and so a good monitor automatically corrects all the images displayed on it for the best contrast to give users the best experience [35] [36]. Gamma Correction is also known by Power-Law Transformation.

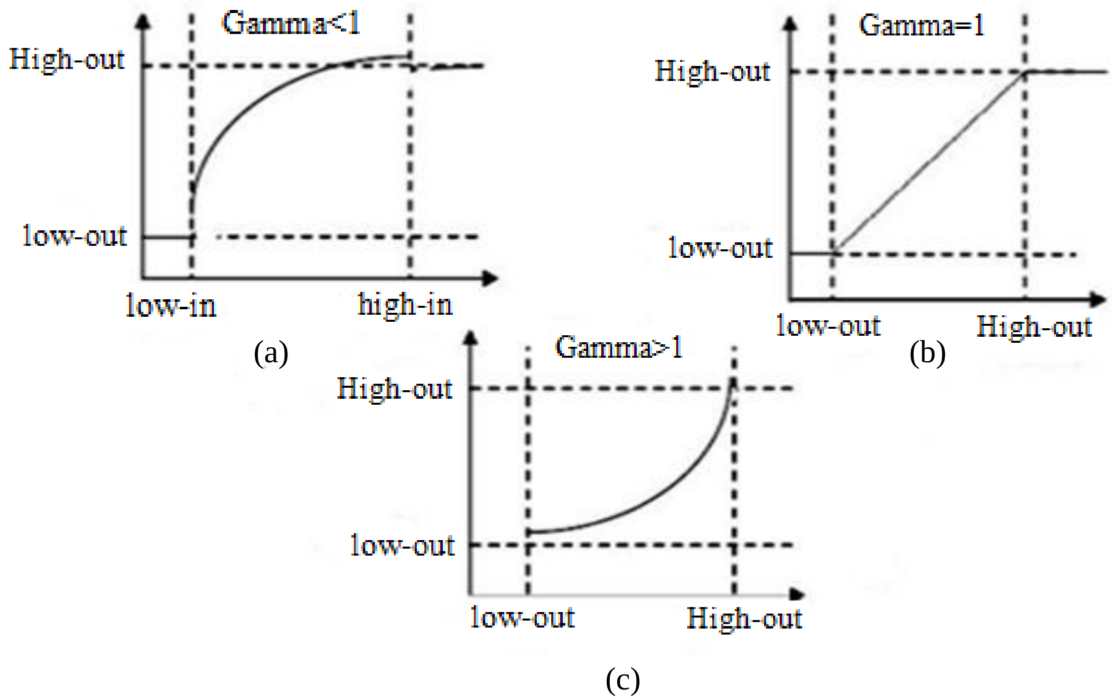


Fig 3.1 All stages of Gamma Correction.

3.2 Contrast- Stretching

Contrast is created by the difference in luminance reflected from two adjacent surfaces [37]. In real perception, contrast is determined by the difference in color and brightness of an object with other objects. Contrast-Stretching transformations increase the contrast between the darks and the lights [38] [6]. Low contrast images may be the cause of wrong setting of the camera, shadow and poor illumination. In other word, contrast stretching is a process that makes more suited result as compare to original image.

The ***contrast- stretching*** has the form [6]:

$$S = 1/(1 + \left(\frac{m}{r}\right)^{E}) \qquad (3.2.1)$$

m - is the mid-line where you want to switch from dark values to light values.

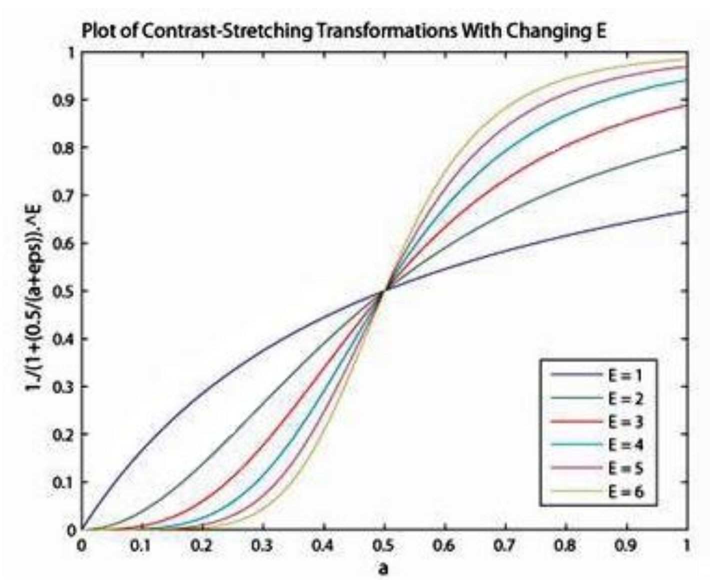
r - ***Input image.***

S - ***Output image***.

E – to Controls the ***slope*** of the function.

In the below figure we can easily understand about contrast stretching. It separates the image into two parts, first one is black and second is white, on the m value, and the transition between these parts is a slope that depends on E value. The slope between black and white parts could be more or less smooth. If the value of E is equal to one (E=1), the stretching became a threshold transformation. If the value of E > 1, the transformation is defined by the curve, which is smoother if the value of E is increased. When E < 1, make the curve become negative. For the grayscale image with low contrast the best method is contrast stretching.

In below figure you can see all mathematic value of the contrast stretching equation:-



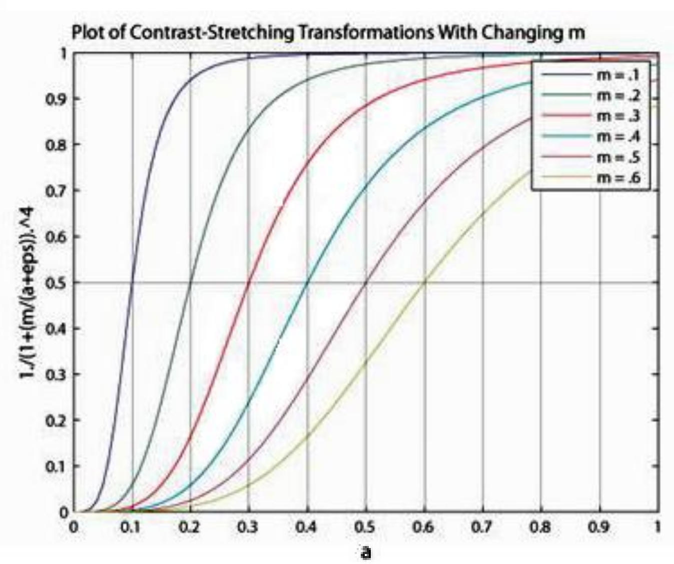


Fig 3.2 Contrast Stretching with variable E.   Fig 3.3 Contrast Stretching with variable m

3.3 Difference Theoretic Texture Features [7]

There are so many techniques available for texture feature extraction. Some best techniques are co-occurrence matrix [39] [40], Markov random fields [41], Gabor filters Q1 [29] [42], Hermite transforms [43], DCT transforms [44] and wavelet representations [45] [46] [47] [48] [49] [50]. Most of these techniques assume that the texture images have the same orientation and scale [7]. But the difference theoretic texture feature technique is better as compared to others because it reduces the feature dimension and provide very good results. It is due to the fact that it is scale, rotation and illumination in variant. This method is superior in performance with an advantage of reduced feature dimension as compare to other method of feature extraction like local binary pattern (LBP) [51], LBP variance (LPBV) [52] method [7]. This method returns a feature set D.

D=[absdiff*H*, absdiff*V*, absdiff*D*, *p*diff*H*, *p*diff*V*, *p*diff*D*, abs*Y*, *pY*, *p*diff*HY*, *p*diff*VY*, *p*diff*DY*]

$$absdiffH = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|f(i,j)-f(i,j+1)|}{M\times N} \tag{3.3.1}$$

$$absdiffV = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|f(i,j)-f(i+1,j)|}{M\times N} \tag{3.3.2}$$

$$absdiffD = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|f(i,j)-f(i+1,j+1)|}{M\times N} \tag{3.3.3}$$

$$pdiffH = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}Ph((f(i,j)-f(i,j+1))Q)}{M\times N} \tag{3.3.4}$$

$$pdiffV = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}Pv((f(i,j)-f(i+1,j))Q)}{M\times N} \tag{3.3.5}$$

$$pdiffD = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}Pd(f(i,j)-f(i+1,j+1)Q)}{M\times N} \tag{3.3.6}$$

$$absY = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|f(i,j)-\mu|)}{M\times N} \tag{3.3.7}$$

$$pY = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} Pg(f(i,j)-\mu)Q)}{M\times N} \quad (3.3.8)$$

$$pdiffHY = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} Phg((f(i,j)-f(i,j+1))Q,(f(i,j)-\mu)Q)}{M\times N} \quad (3.3.9)$$

$$pdiffVY = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} Pvg((f(i,j)-f(i+1,j))Q,(f(i,j)-\mu)Q)}{M\times N} \quad (3.3.10)$$

$$pdiffDY = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} Pdg((f(i,j)-f(i+1,j+1))Q,(f(i,j)-\mu)Q)}{M\times N} \quad (3.3.11)$$

The following parameters are used in above equations represents-

$f(i,j)$ is the pixel intensity at coordinate position *(i, j)*, *i = 1, 2, 3,….,M-1, M* and *j = 1, 2, 3,…,N-1, N. Ph, Pv and Pd* are local probability values obtained from *h(horizontal), v(vertical)* and *d(diagonal)* histogram respectively. *Pg* is global probability obtained from g(global difference) and *Phg*, *Pvg* and *Pdg* are joint probability of local and global differences and the mean value of the image represents using μ. *pY* is the mean frequency of occurrence of global differences obtained by mapping back to each pixel the *Pg*

These are the following steps of difference theoretic texture features [7] -

1.) Find the signed grey level differences both global and local for each pixel in an image.
2.) Find the histograms of signed global and local differences ph, pv, pd and pg.
3.) Find joint histograms of signed global and local differences phg, pvg and pdg.
4.) Map the probability values from the histograms to each pixel in the image based on best match of histogram indices and actual pixel differences.
5.) Find the average of the absolute differences and the probabilities of the signed differences over all the pixels in the image to form an11-dimensional feature set using the equations (3.3.1–11) described above.

3.4 Performance Measure

There are so many popular techniques available for binarization. That's why a performance method used to compare the results of the proposed method with others. The performance is evaluated by using F-Measure (FM) [53]. The image can be divide into four parts: one is True Positive (this part includes only relevant information), second is False Positive (area of unexpected relevant information), third one is True Negative (represents the relevant thing as irrelevant) and the last one is False Negative (includes the unexpected irrelevant information). We can also define TP (True Positive) as is the total number of matched foreground pixels, FP (False Positive) is the total number of misclassified foreground pixels in binarization result as compared to ground-truth and FN (False Negative) is the total number of misclassified background pixels in binarization result as compared to ground-truth [54]. These parts are used to calculate FM. This section reviews how to calculate FM, Precision and Recall.

3.4.1 Precision

Precision is a technique that uses to evaluate the average probability of relevant retrieval. The precision value represents a relevant thing. The Precision is the ratio of correctly detected text to the sum of correctly detected text plus unexpected texts. Unexpected texts in an image are those texts which are actually not a text (foreign bodies, noise), but have been detected by the algorithm as text regions. The correctly detected text is also known as True Positive (TP) and unexpected text is known as False Positive (FP)

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.4.1}$$

If the value of precision is 1 that means the results retrieves by the algorithm was relevant. For example, in case of document image result, if precision rate = 1, then the resulted image contains only text, but we cannot say that all relevant text were retrieved.

3.4.2 Recall

The average probability of complete retrieval is known as Recall. A recall can be defined as the ratio of correctly detected text to the sum of correctly detected text plus texts which are not detected by the algorithm. The texts which have not been detected by the algorithm is known as False Negative (FN). If the value of recall is 1 that means the results retrieves by the algorithm was relevant. For example, in case of document image result, if precision rate = 1, then the resulted image contains only text, but we cannot say nothing about how many irrelevant thing was also retrieved.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{3.4.2}$$

3.4.3 F-Measure

F-Measure is used to measure performance of binary image. It is calculated by using the combination of precision and recall. F-Measure is also known as tradition F-Measure or balanced F-Score or F_1 Measure, because precision and recall are equally weighted. F-Measure can be considered as a harmonic mean of precision and recall.

$$\mathrm{F-Measure} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{3.4.3}$$

CHAPTER-4

PROPOSED TEXT SEGMENTATION AND BINARIZATION USING DIFFERENCE THEORETIC TEXTURE FEATURES

This chapter presents a brief overview of the proposed approach “text segmentation and binarization using the difference theoretic texture feature. The purpose of this algorithm is to find out text area and then binarize the text area. This chapter divides into five sections. The first section presents an input. The second section reviews the preprocessing. Third section represents about feature extraction and matching. Fourth section is text segmentation and binarization and last section presents an analysis.

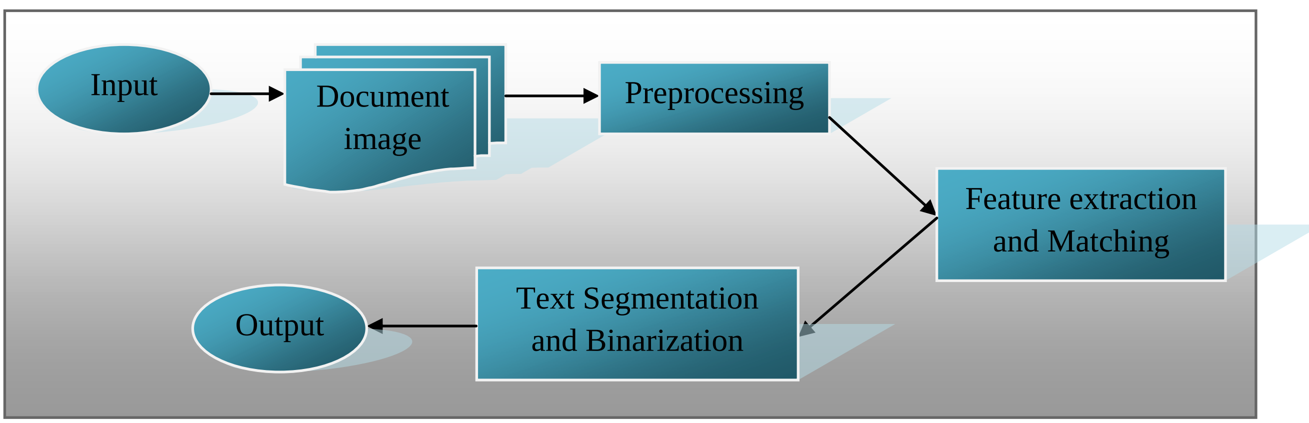


Fig 4.1 Architecture diagram of our method

4.1 Input Image

The proposed algorithm has been tested on more than hundred (100) document images. These document images are taken from well-defined DIBCO datasets and captured from a mobile phone. There are sixty-six (66) images available from DIBCO datasets and thirty-six (36) images from a mobile camera.

The DIBCO series developed by Gatos et al, Tikakis et al and Pratikakis et al are publicly available. Every dataset of DIBCO has an original image and corresponding Ground Truth (GT) image. The first document image binarization contest (DIBCO-2009) dataset contains five (5) handwritten and five (5) machine printed image. The second (DIBCO-2010) dataset contains only ten (10) handwritten image. The DIBCO-2011 and DIBCO-2013 dataset have eight (8) machine printed and eight (8) handwritten images. There are fourteen (14) handwritten image available from the DIBCO-2012 dataset. There are thirty-six (36) document images that captured using mobile phones (Samsung Galaxy S4). These document images are divided into different categories based on noise-

1.) Document images with fingers (10 images)
2.) Document images with pen/pencil (8 images)
3.) Document images with mobile/watch (8 images)
4.) Extra document image (10 images)

Table 4.1: The Input Datasets used for Experimentation.

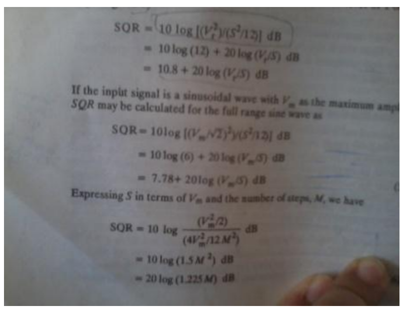
Datasets	Handwritten image	Machine printed image	Total image
DIBCO-2009	5	5	10
DIBCO-2010	10	----	10
DIBCO-2011	8	8	16
DIBCO-2012	14	----	14
DIBCO-2013	8	8	16
Mobile phone	3	33	36

Before using these document images, there is a need to resize the document images with (300×400) to make same size. For proposed algorithm there also need to create a template image of (15×20) size. This template image is used to extract text from document image.

4.2 Preprocessing

The aim of preprocessing is to prepare the input for further processing. It is the first step after reading the input. Preprocessing performs vital role, if the quality of document image is poor and require to remove noise and contrast enhancement between foreground and background of an image. An improve adaptive global threshold algorithm (threshold algorithm with gamma correction and contrast stretching) have used to enhance input image and removal of foreign bodies. The detail preprocessing algorithm is described below with an example (document image captured using mobile phones)-

1) Take an input image and convert it to gray image.



$$SQR = 10 \log [(V_r^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (12) + 20 \log (V_r/S) \text{ dB}$$
$$= 10.8 + 20 \log (V_r/S) \text{ dB}$$

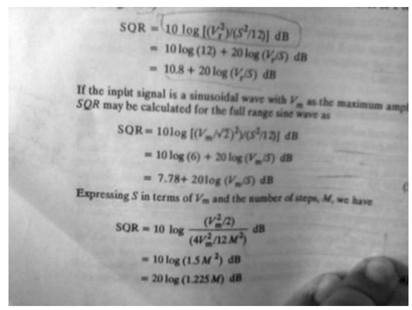
If the inplut signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10 \log [(V_m/\sqrt{2})^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (6) + 20 \log (V_m/S) \text{ dB}$$
$$= 7.78 + 20 \log (V_m/S) \text{ dB}$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10 \log \frac{(V_m^2/2)}{(4V_m^2/12M^2)} \text{ dB}$$
$$= 10 \log (1.5M^2) \text{ dB}$$
$$= 20 \log (1.225M) \text{ dB}$$

Fig 4.2.1 input image



$$SQR = 10 \log [(V_r^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (12) + 20 \log (V_r/S) \text{ dB}$$
$$= 10.8 + 20 \log (V_r/S) \text{ dB}$$

If the inplut signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10 \log [(V_m/\sqrt{2})^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (6) + 20 \log (V_m/S) \text{ dB}$$
$$= 7.78 + 20 \log (V_m/S) \text{ dB}$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10 \log \frac{(V_m^2/2)}{(4V_m^2/12M^2)} \text{ dB}$$
$$= 10 \log (1.5M^2) \text{ dB}$$
$$= 20 \log (1.225M) \text{ dB}$$

Fig 4.2.2 gray image

2) Select an initial threshold value t1 by calculating the using Otsu's method [5].
3) Take second threshold value t2=0.
4) Divide the input image into two clusters that is g1 and g2 by using t1. Cluster g1 includes pixels with gray level value greater than or equal to t1 and cluster g2 includes pixels with gray level value less than t1 [1].

$$g1(x,y) = \begin{cases} 1, & if \;\; f(x,y) \geq t1 \\ 0, & otherwise \end{cases} \qquad (4.2.1)$$

$$g2(x,y) = \begin{cases} 1, & if \;\; f(x,y) < t1 \\ 0, & otherwise \end{cases} \qquad (4.2.2)$$

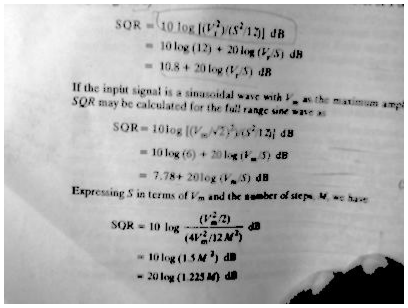


Fig 4.2.3 g1 image

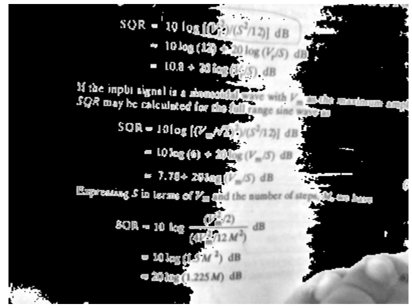


Fig 4.2.4 g2 image

5) Perform gamma correction [6] on both input image g1 and g2 by using the following formula:-

$$g11 = c \times (g1)^{\frac{1}{g}} \quad (4.2.3)$$

$$g21 = c \times (g2)^{\frac{1}{g}} \quad (4.2.4)$$

g=constant (we take 1/g=0.97), if g>1 increase contrast and g<1 reduce contrast

c=1 (constant)

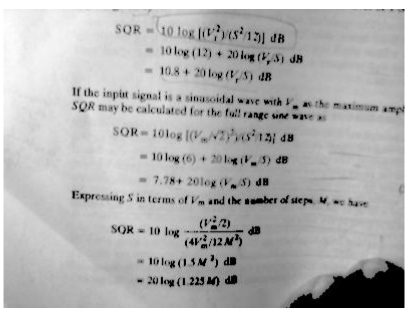


Fig 4.2.5 g11 image

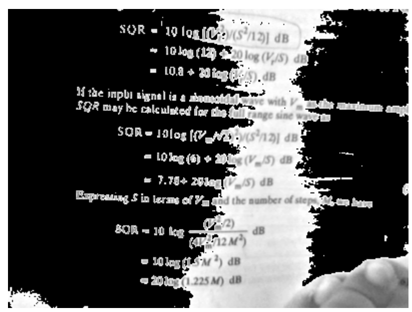


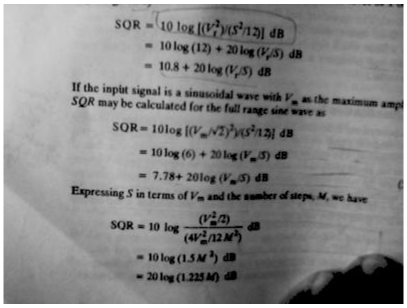
Fig 4.2.6 g21 image

6) Replace t2 by t1.
7) Calculate the average gray level value t1 for pixel in g11.
8) Repeat step 2 to 6 until the difference between t1 and t2 is greater than zero.
9) Perform contrast stretching [6] on both input image g11 and g21 and produce output images g111 and g211 respectively by using the following formula-

$$g111 = 1/(1 + \left(\frac{t1}{g11}\right)^{E}) \quad (4.2.5)$$

$$g211 = 1/(1 + \left(\frac{t1}{g21}\right)^{E}) \qquad (4.2.6)$$

E=constant (we take E=4)



SQR = 10 log [(V_r^2)/(S^2/12)] dB
= 10 log (12) + 20 log (V_r/S) dB
= 10.8 + 20 log (V_r/S) dB
If the input signal is a sinusoidal wave with V_m as the maximum amp
SQR may be calculated for the full range sine wave as
SQR = 10log [(V_m/√2)^2)/(S^2/12)] dB
= 10 log (6) + 20 log (V_m/S) dB
= 7.78+ 20log (V_m/S) dB
Expressing S in terms of V_m and the number of steps, M, we have
SQR = 10 log $\frac{(V_m^2/2)}{(4V_m^2/12M^2)}$ dB
= 10 log (1.5M^2) dB
= 20 log (1.225M) dB

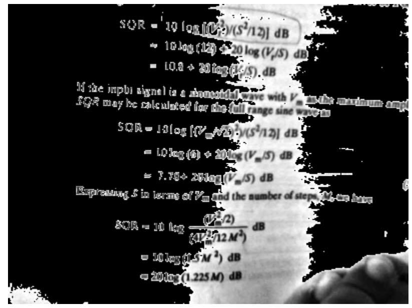


Fig 4.2.7 g111 image

Fig 4.2.8 g211 image

As you can see the results and compare the original image with enhanced image. Original image contains noise like a finger in the corner of the image and some shadow portion. But the enhanced image (i.e. g111 image) has a very less portion of finger nails and improves the quality of the image. After preprocessing the enhanced image (i.e. g111 image) are used for further steps.

4.3 Feature Extraction and Matching

The output of previous section (i.e. enhanced image g111) is used in this section. The algorithm design for feature extraction and matching are described below-

10) Now g111 is the new document image.
11) Creates a template image (t1) in Microsoft word for any size.
12) Resize the template with any value of height (h1=15) and length (l1=20).

Fig 4.3.1 template image

13) Find the difference theoretic texture features [7] of the template.

14) Match the template image with original document image by using block matching:

- Takes the value of the height (h1=15) and length (l1=20) to create blocks in document image for matching with the template.
- Find the distinct texture features of all blocks.
- Match the features of each block (D2) with the texture features of the template (D1) by using the maximum of absolute differences to detect a mismatch.

$$m = \max |D1 - D2| \tag{4.3.1}$$

15) Create an array (A) to store those pixel values where matching is found.

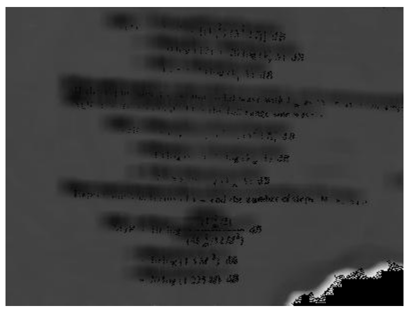


Fig 4.3.2 A image

4.4 Text Segmentation and Binarization

The result produce by previous section (A image) are used to find a text area from document image. The process of finding text area and binarization are described below-

16) Display the array (A) (through this array we can detect the line of the document) and finds the number of lines by comparing the threshold value (calculated using Otsu method [5]) (imgc1).
17) Multiply the original image (g111) with the array A to extract text from lines (imgc2).

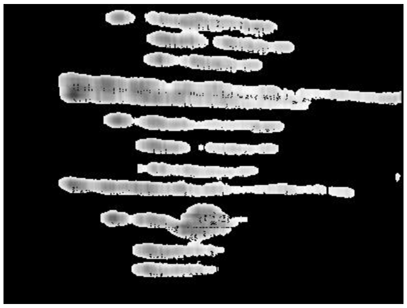


Fig 4.4.1 imgc1 image

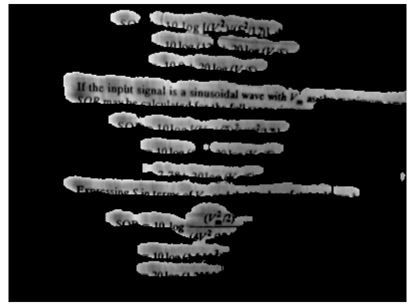


Fig 4.4.2 imgc2 image

18) GET BINARIZED TEXT FROM BLOB:-

- ✓ Converts (imgc2) image into a binary image using <u>Otsu threshold method (imgc3)</u>.
- ✓ Apply average filter window size [3, 3] to imgc3 image (imgc4).

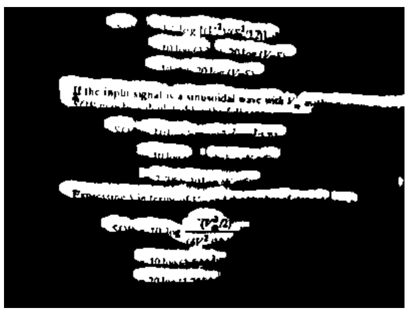


Fig 4.4.3 imgc3 image

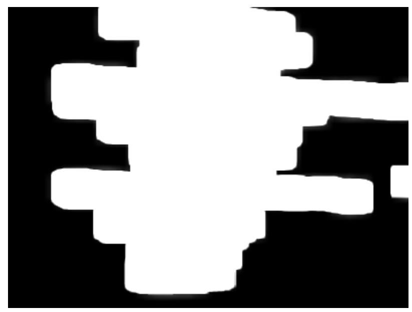


Fig 4.4.4 imgc4 image

- ✓ 1.) Multiply (imgc4) with the original image (input image) (imgc5).

  2.) Convert (imgc5) image into a binary image using yanowitz method [8](imgc6).

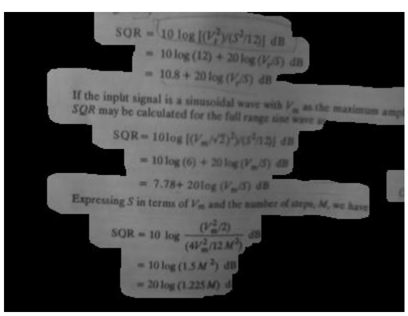


Fig 4.4.5 imgc5 image

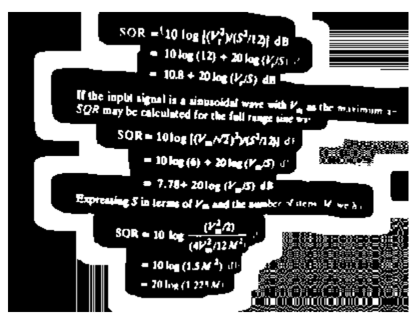


Fig 4.4.6 imgc6 image

  3.) For the purpose of extracting the text, in a loop for each white pixel in dilate image put the value of (imgc6) image into a new black image. (imgc7) (final binary image)

4.) Superimpose the text area in the original image (imgc8) (final text segment image).

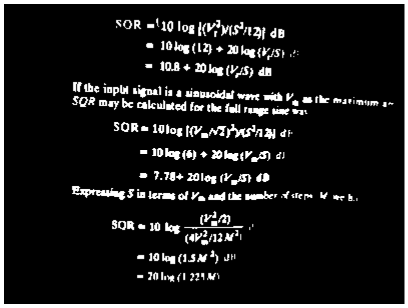


Fig 4.4.7 imgc7 image

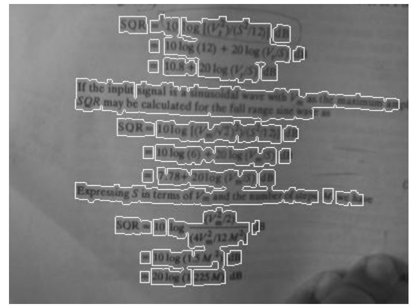
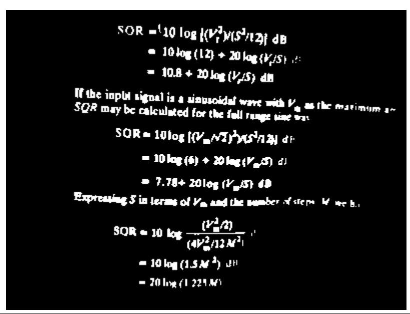
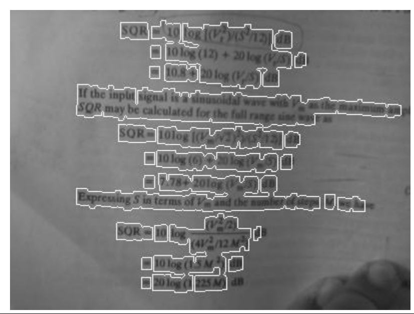
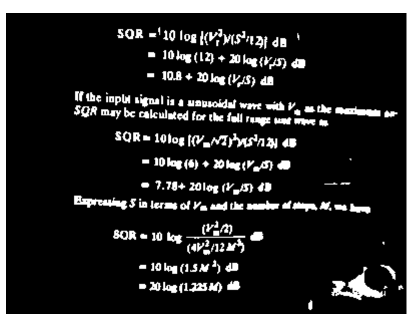
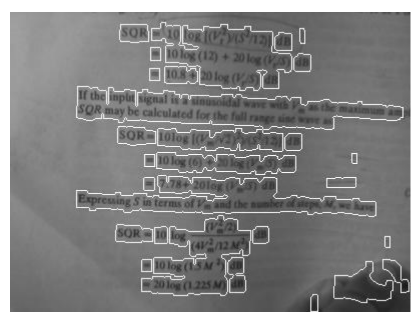
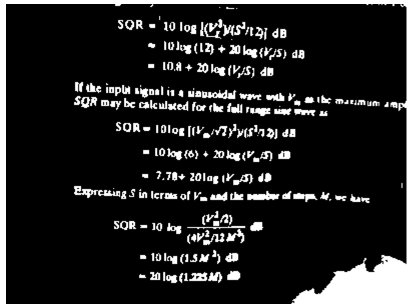
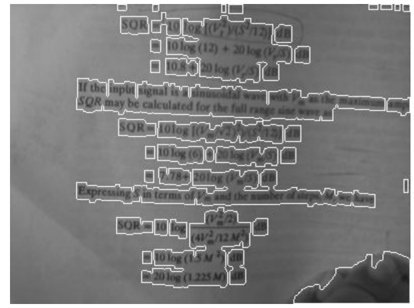
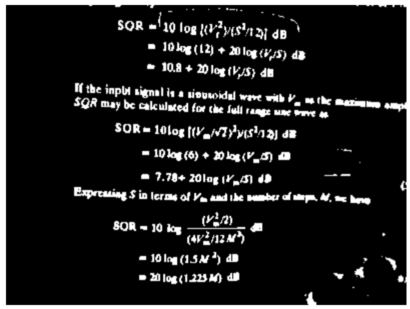
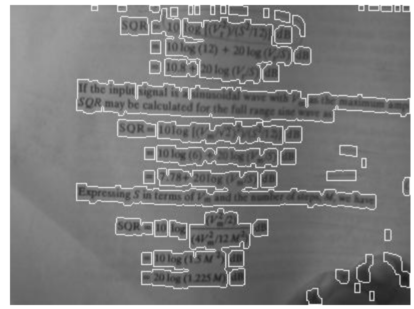
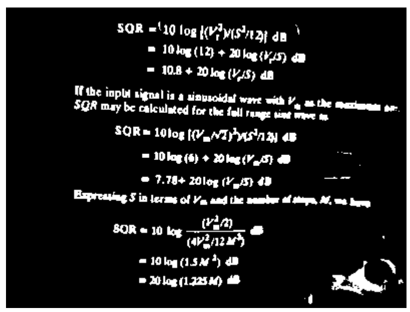
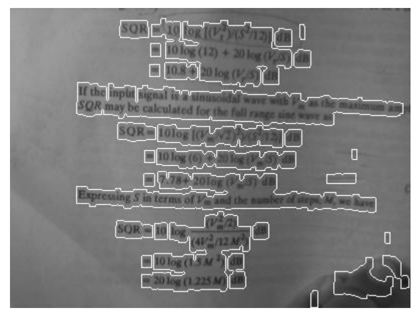


Fig 4.4.8 imgc8 image

The above figure (Fig 4.4.7) and (Fig 4.4.8) are the final output of proposed algorithm “text segmentation and binarization.” This algorithm has been compared with well-known binary algorithm which are Yanowitz’s method, Niblack’s method, Sauvola’s method and Post-Yanowitz’s method.

Table 4.2 Comparison Method

Method	Binary image	Segmented image
Proposed method		
Yanowitz’s method		

Niblack's method		
Sauvola's method		
Post-Yanowitz's method		

As you can see Table 2 which defines the results of proposed and some best comparison methods, the result of proposed algorithm is best because it remove noise and binary only text area. But in case of the comparison method, no one completely removes the noise figure.

The first, best result is provided by our method and second best result come from a Niblack's method. The third best result is of Yanowitz's method and fourth best (last) produces through Sauvola's method. In case of DIBCO series, the proposed method is also evaluated based on precision, recall and F-Measure.

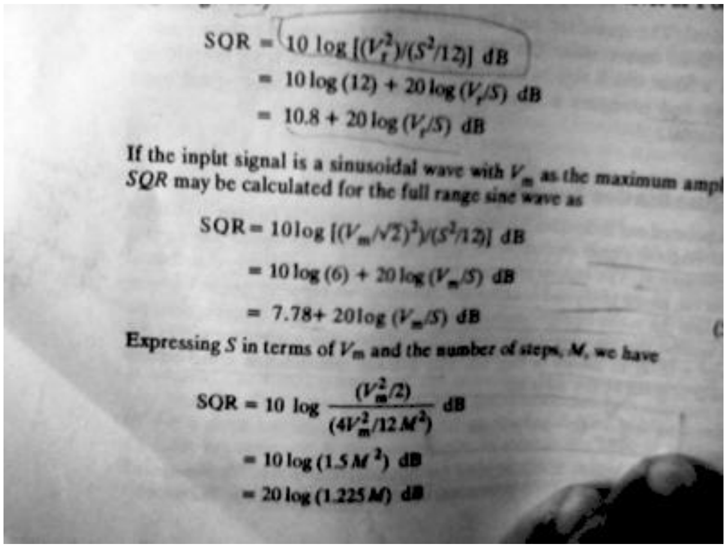
4.5 Analysis

This section review about some user define parameters to create propose algorithm. How and why only these parameters are provided better results as compare to other parameters. These parameters are described below in detail.

4.5.1 Selection of Gamma Value

It is very difficult to select a gamma value to improve the quality of the image. The different value of gamma has tried to make good result for every input image. After trying so many value, the one value is selected which provide good result for every input image. The selected gamma (g) value is 1.039 (i.e. 1/g=0.97) in equation 4.2.3 and 4.2.4. Because the selected gamma value is greater than one (1) that means it is used to increase contrast. So much value has also tried to final 1/g = 0.97. For some value of 1/g, the figures are shown below-

Fig 4.5.1 represents an image when we use (1/g = 0.95). This image have some extra area of finger as compare to Fig 4.5.3 (used in proposed algorithm 1/g = 0.97).



$$SQR = 10 \log [(V_r^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (12) + 20 \log (V_r/S) \text{ dB}$$
$$= 10.8 + 20 \log (V_r/S) \text{ dB}$$

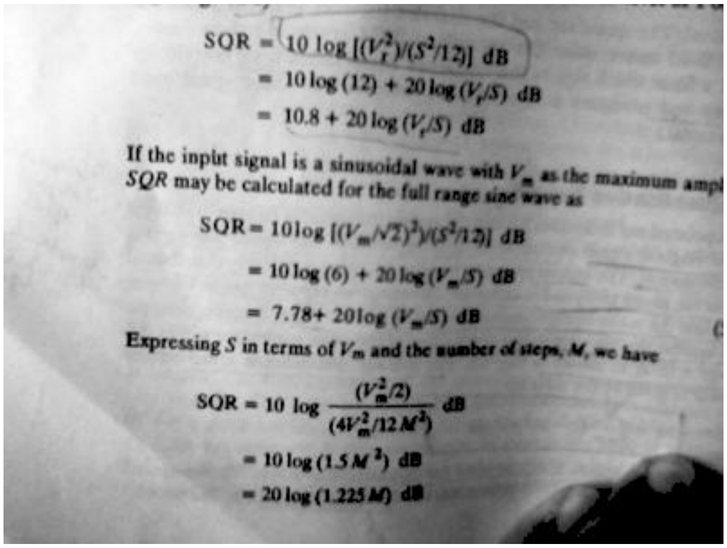
If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10 \log [(V_m/\sqrt{2})^2/(S^2/12)] \text{ dB}$$
$$= 10 \log (6) + 20 \log (V_m/S) \text{ dB}$$
$$= 7.78 + 20 \log (V_m/S) \text{ dB}$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10 \log \frac{(V_m^2/2)}{(4V_m^2/12M^2)} \text{ dB}$$
$$= 10 \log (1.5 M^2) \text{ dB}$$
$$= 20 \log (1.225 M) \text{ dB}$$

Fig 4.5.1 image with 1/g=0.95



$$SQR = 10 \log [(V_r^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (12) + 20 \log (V_r/S) \text{ dB}$$
$$= 10.8 + 20 \log (V_r/S) \text{ dB}$$

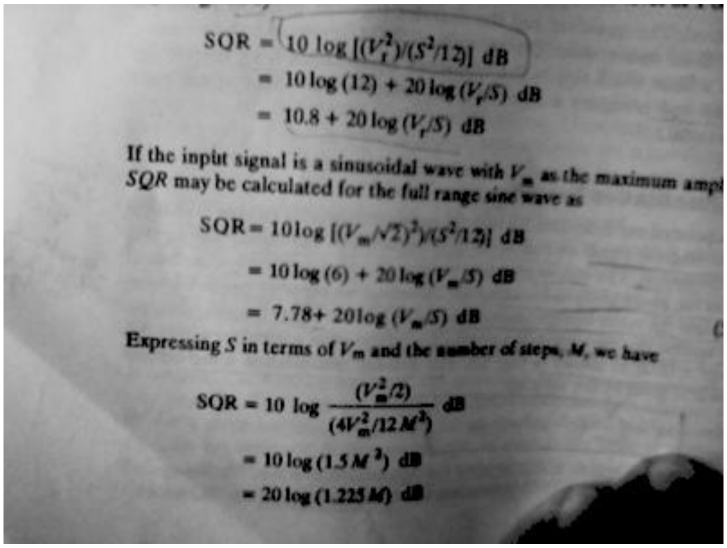
If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10 \log [(V_m/\sqrt{2})^2/(S^2/12)] \text{ dB}$$
$$= 10 \log (6) + 20 \log (V_m/S) \text{ dB}$$
$$= 7.78 + 20 \log (V_m/S) \text{ dB}$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10 \log \frac{(V_m^2/2)}{(4V_m^2/12M^2)} \text{ dB}$$
$$= 10 \log (1.5 M^2) \text{ dB}$$
$$= 20 \log (1.225 M) \text{ dB}$$

Fig 4.5.2 image with 1/g=0.96



$$SQR = 10 \log [(V_r^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (12) + 20 \log (V_r/S) \text{ dB}$$
$$= 10.8 + 20 \log (V_r/S) \text{ dB}$$

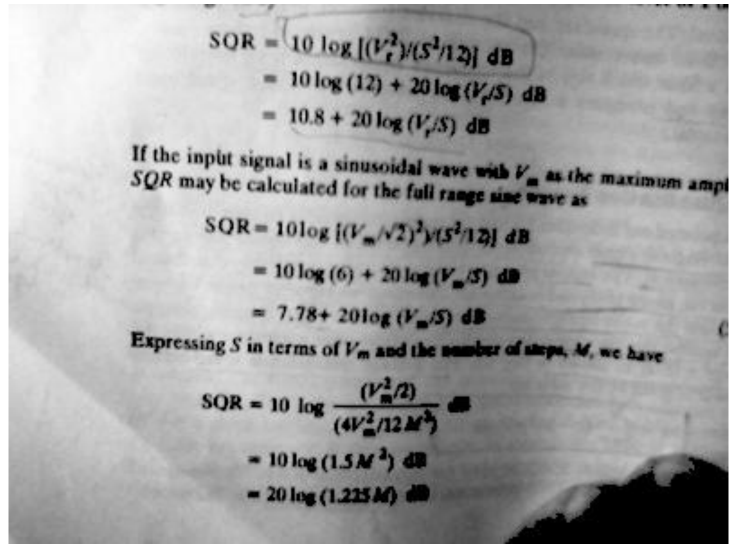
If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10\log [(V_m/\sqrt{2})^2/(S^2/12)] \text{ dB}$$
$$= 10 \log (6) + 20 \log (V_m/S) \text{ dB}$$
$$= 7.78 + 20\log (V_m/S) \text{ dB}$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10 \log \frac{(V_m^2/2)}{(4V_m^2/12M^2)} \text{ dB}$$
$$= 10 \log (1.5M^2) \text{ dB}$$
$$= 20 \log (1.225M) \text{ dB}$$

Fig 4.5.3 image with 1/g=0.97



$$SQR = 10 \log [(V_r^2)/(S^2/12)] \text{ dB}$$
$$= 10 \log (12) + 20 \log (V_r/S) \text{ dB}$$
$$= 10.8 + 20 \log (V_r/S) \text{ dB}$$

If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10\log [(V_m/\sqrt{2})^2/(S^2/12)] \text{ dB}$$
$$= 10 \log (6) + 20 \log (V_m/S) \text{ dB}$$
$$= 7.78 + 20\log (V_m/S) \text{ dB}$$

Expressing S in terms of V_m and the number of steps, M, we have

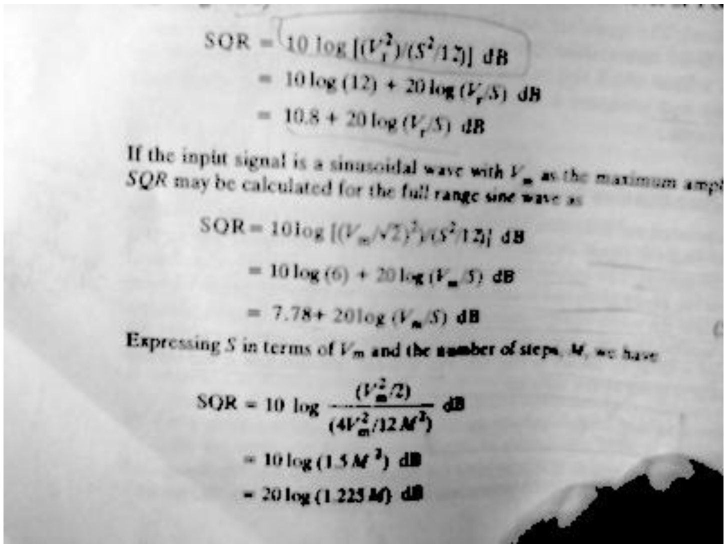
$$SQR = 10 \log \frac{(V_m^2/2)}{(4V_m^2/12M^2)} \text{ dB}$$
$$= 10 \log (1.5M^2) \text{ dB}$$
$$= 20 \log (1.225M) \text{ dB}$$

Fig 4.5.4 image with 1/g=0.99

This result is not good so tried with some other gamma value (i.e. 1/g = 0.96) which shown in Fig 4.5.2. The second figure is better than as compare to first one. After that an idea is come that if the value of 1/g is greater than 0.95, provide better results. Then try with 1/g = 0.97 are shown in Fig 4.5.3. It is better than the second one because it also removes some extra portion of noise and the text is also readable. After that, if the value of 1/g is increase than the result is bad as compared to Fig 4.5.3. As you can see in Fig 4.5.4 with 1/g = 0.99 is greater than 0.97, not able to produce readable text (in last three line db are not good as compare to Fig 4.5.1, Fig 4.5.2 and Fig 4.5.4). That means the saturation point is 1/g = 0.97. That's why 0.97 is selected for the propose algorithm.

4.5.2 Selection of E-Value in Contrast Stretching

In contrast stretching function, E value is used to control the slope of the function. The contrast stretching is used to increase the contrast between darks and light. The value of E should be good to enhance the image. There are four images Fig 4.5.5, Fig 4.5.6, Fig 4.5.7 and Fig 4.5.8 of different E value that is E = 2, E = 3, E = 4 and E = 6 respectively. These images are shown below and you can compare every image with others to select good result. According to visual perception, Fig 4.5.7 (third image) is a good result as compared to other results because it is readable as well as reduces the shadow portion.



$$SQR = 10\log[(V_r^2)/(S^2/12)]\ dB$$
$$= 10\log(12) + 20\log(V_r/S)\ dB$$
$$= 10.8 + 20\log(V_r/S)\ dB$$

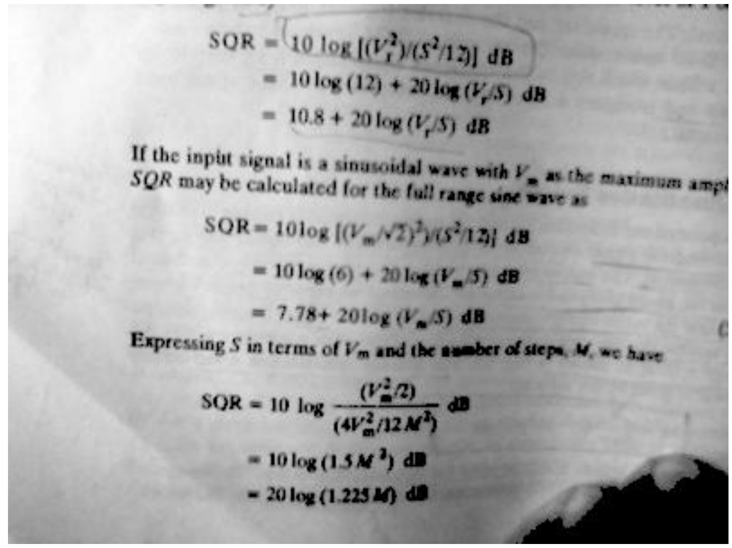
If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10\log[(V_m/\sqrt{2})^2/(S^2/12)]\ dB$$
$$= 10\log(6) + 20\log(V_m/S)\ dB$$
$$= 7.78 + 20\log(V_m/S)\ dB$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10\ \log\frac{(V_m^2/2)}{(4V_m^2/12M^2)}\ dB$$
$$= 10\log(1.5M^2)\ dB$$
$$= 20\log(1.225M)\ dB$$

Fig 4.5.5 image with E=2



$$SQR = 10\log[(V_r^2)/(S^2/12)]\ dB$$
$$= 10\log(12) + 20\log(V_r/S)\ dB$$
$$= 10.8 + 20\log(V_r/S)\ dB$$

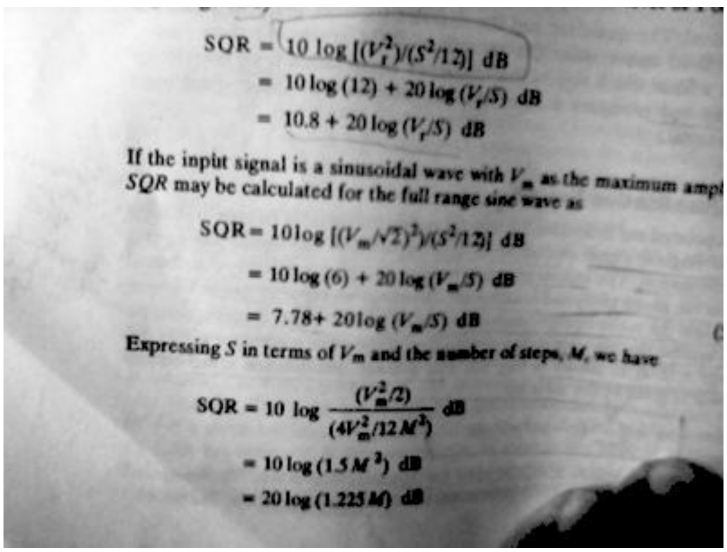
If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10\log[(V_m/\sqrt{2})^2/(S^2/12)]\ dB$$
$$= 10\log(6) + 20\log(V_m/S)\ dB$$
$$= 7.78 + 20\log(V_m/S)\ dB$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10\ \log\frac{(V_m^2/2)}{(4V_m^2/12M^2)}\ dB$$
$$= 10\log(1.5M^2)\ dB$$
$$= 20\log(1.225M)\ dB$$

Fig 4.5.6 image with E=3



$$SQR = 10\log[(V_r^2)/(S^2/12)]\ dB$$
$$= 10\log(12) + 20\log(V_r/S)\ dB$$
$$= 10.8 + 20\log(V_r/S)\ dB$$

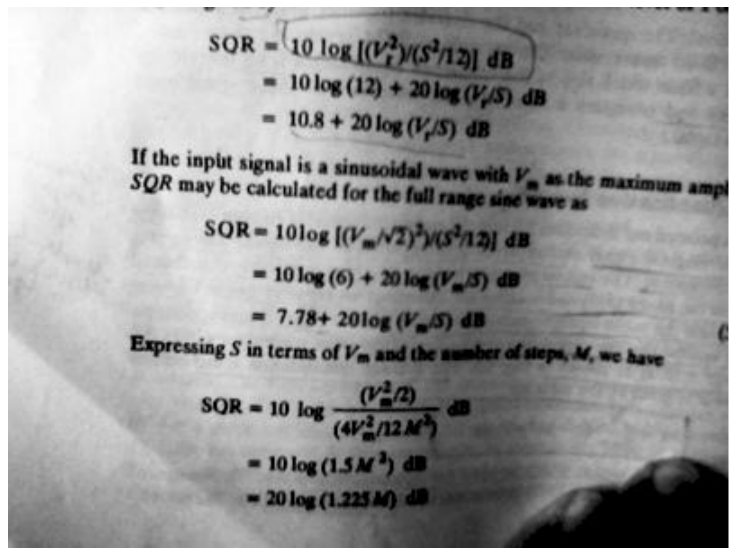
If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

$$SQR = 10\log[(V_m/\sqrt{2})^2/(S^2/12)]\ dB$$
$$= 10\log(6) + 20\log(V_m/S)\ dB$$
$$= 7.78 + 20\log(V_m/S)\ dB$$

Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10\ \log\frac{(V_m^2/2)}{(4V_m^2/12M^2)}\ dB$$
$$= 10\log(1.5M^2)\ dB$$
$$= 20\log(1.225M)\ dB$$

Fig 4.5.7 image with E=4



$$SQR = 10\log[(V_r^2)/(S^2/12)]\ dB$$
$$= 10\log(12) + 20\log(V_r/S)\ dB$$
$$= 10.8 + 20\log(V_r/S)\ dB$$

If the input signal is a sinusoidal wave with V_m as the maximum ampl
SQR may be calculated for the full range sine wave as

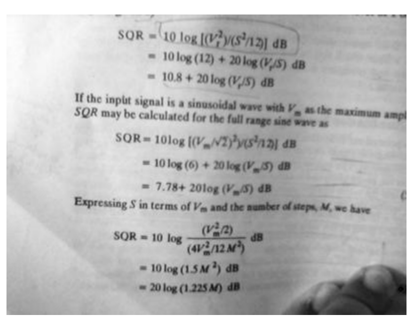
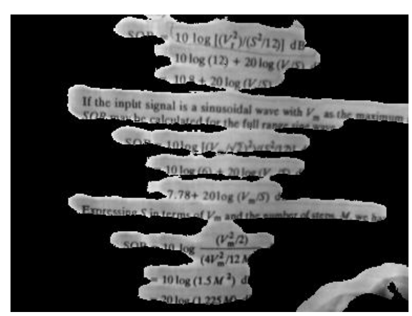
$$SQR = 10\log[(V_m/\sqrt{2})^2/(S^2/12)]\ dB$$
$$= 10\log(6) + 20\log(V_m/S)\ dB$$
$$= 7.78 + 20\log(V_m/S)\ dB$$

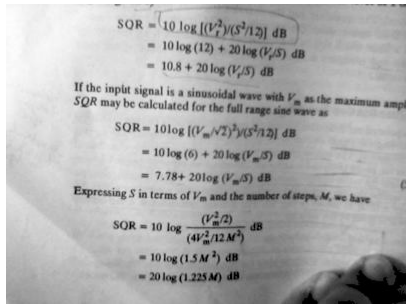
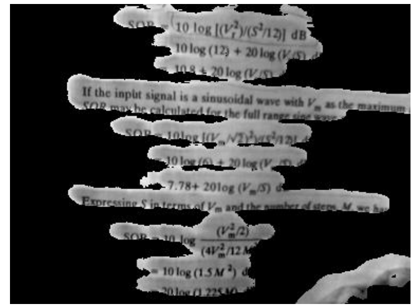
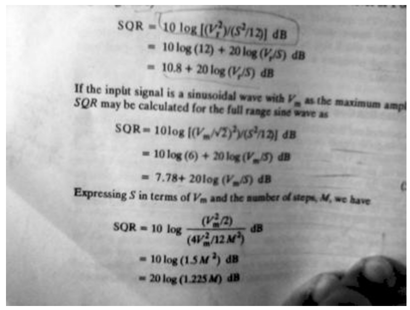
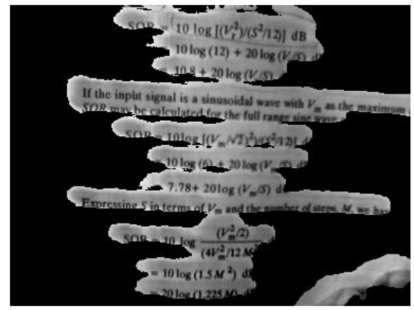
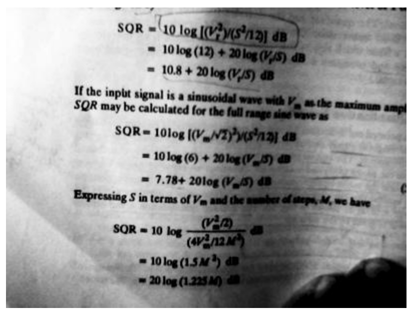
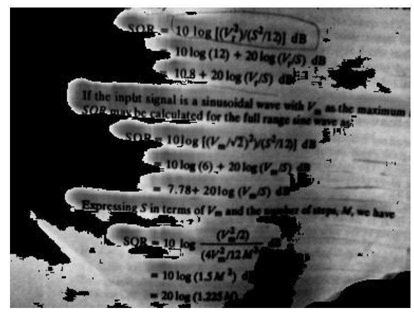
Expressing S in terms of V_m and the number of steps, M, we have

$$SQR = 10\ \log\frac{(V_m^2/2)}{(4V_m^2/12M^2)}\ dB$$
$$= 10\log(1.5M^2)\ dB$$
$$= 20\log(1.225M)\ dB$$

Fig 4.5.8 image with E=6

Table 4.3 Results with different 1/g and E value

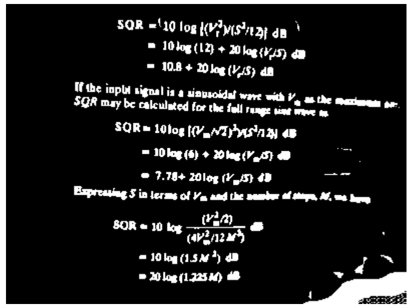
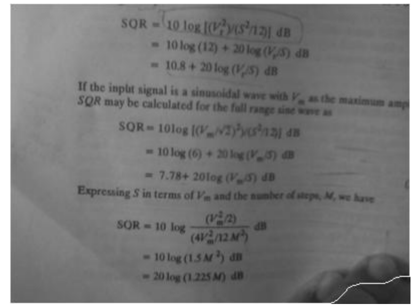
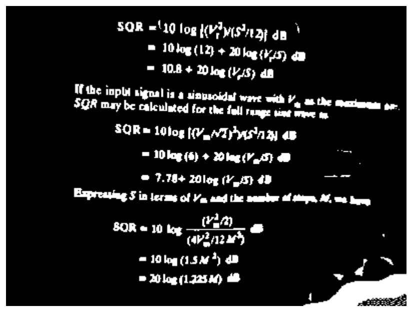
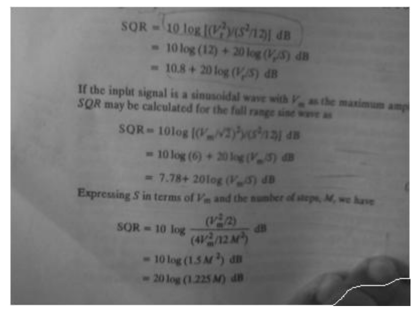
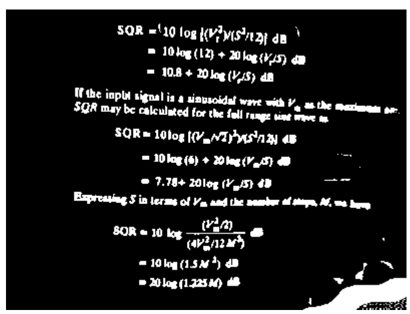
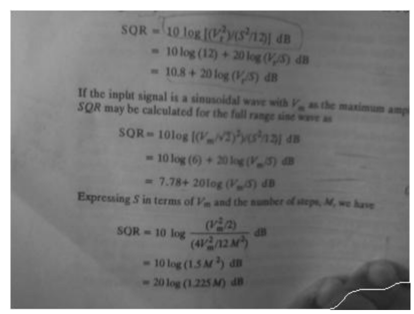
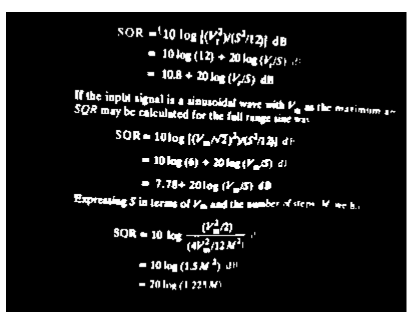
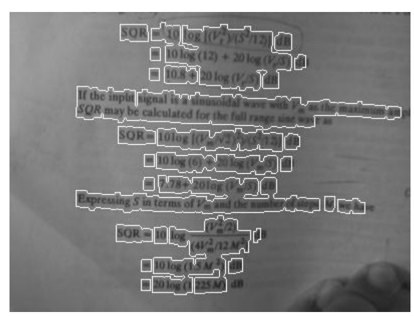
1/g	E	Enhance image	Segmented image
0.8	3		

0.9	3		
0.9	3.5		
0.97	8		

4.5.3 Selection of Template Image

The template requires extracting text from document images. If the features of template are similar to the features of blocks in document image, then it provides high matching score. There are more than 100 document images, and only one template. So it is very difficult to give better result for every image. There are two things requires in case of the template. First one is text of template image and the second one is the size of the selected template image. So many templates have tried with different- different size, but the selected template provides better result in every image. Some templates and corresponding outputs are given below-

Table 4.4 Results with different template image

Template image	Binary image	Segmented image
		
		
		
dB		

<u>SELECTION OF TEMPLATE SIZE</u>: If the template size is similar to character size of input image, then it retrieves almost text area. Currently the selected template size is [15, 20] after trying so many different sizes for template image. For every input image the selected template image size [15, 20] provides good results. Here showing the results with some other good template size to compare with selected size results. In below graphs the red line represents FM, PRECISION and RECALL with template size [15, 20] and others for different template size.

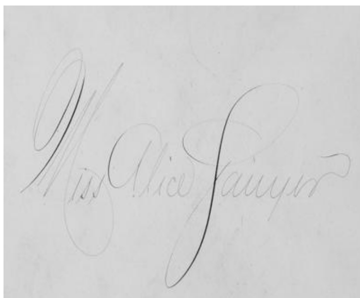


Fig 4.5.9 Img1

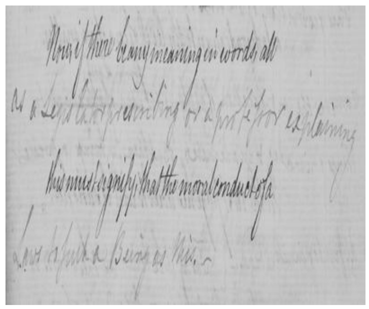


Fig 4.5.10 Img2

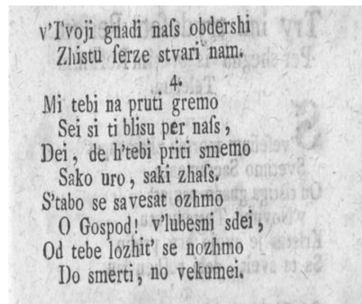


Fig 4.5.11 Img3

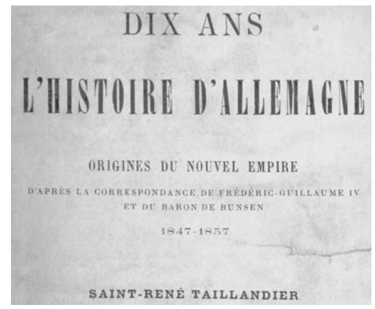


Fig 4.5.12 Img4

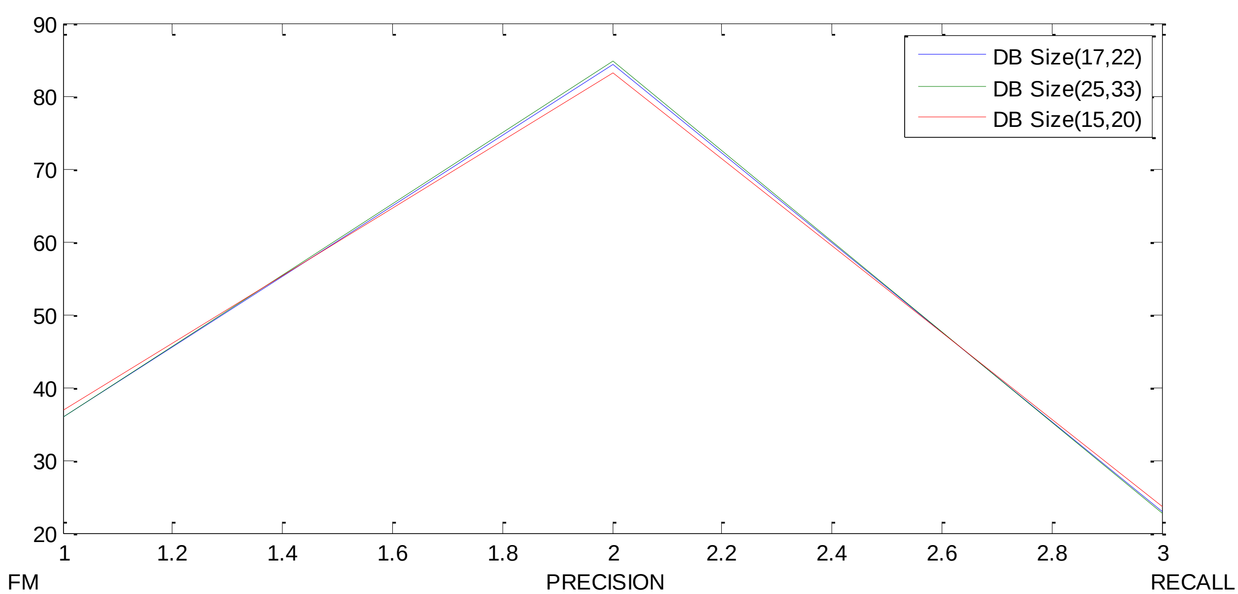


Fig 4.5.13 Results with different template size for Img1

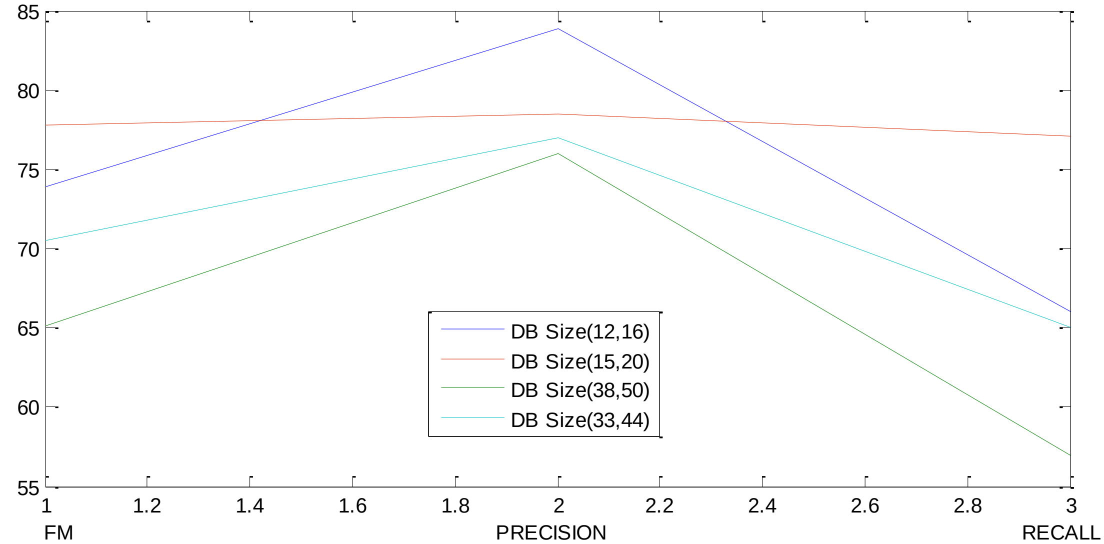


Fig 4.5.14 Results with different template size for Img2

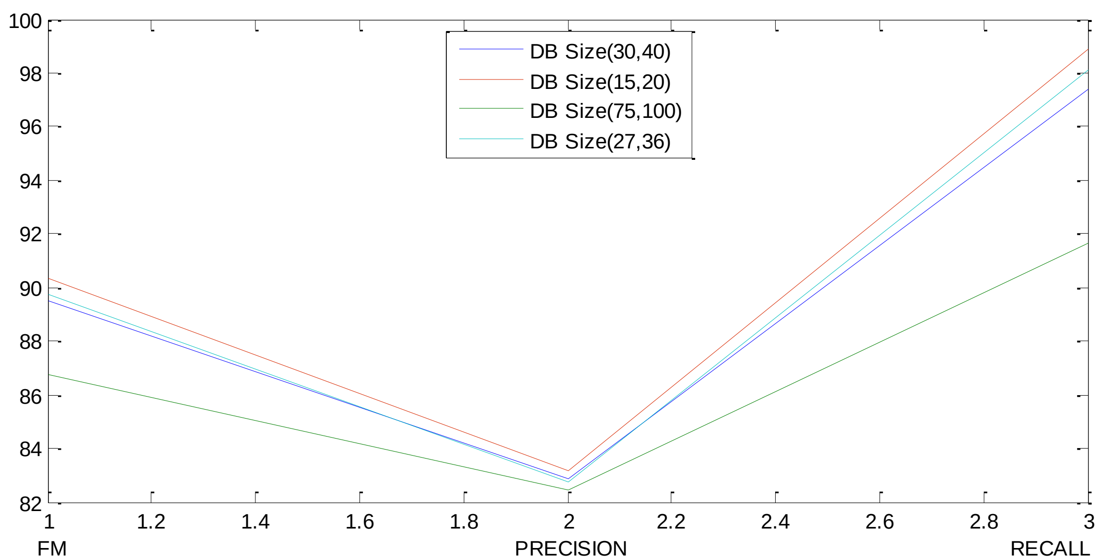


Fig 4.5.15 Results with different template size for Img3

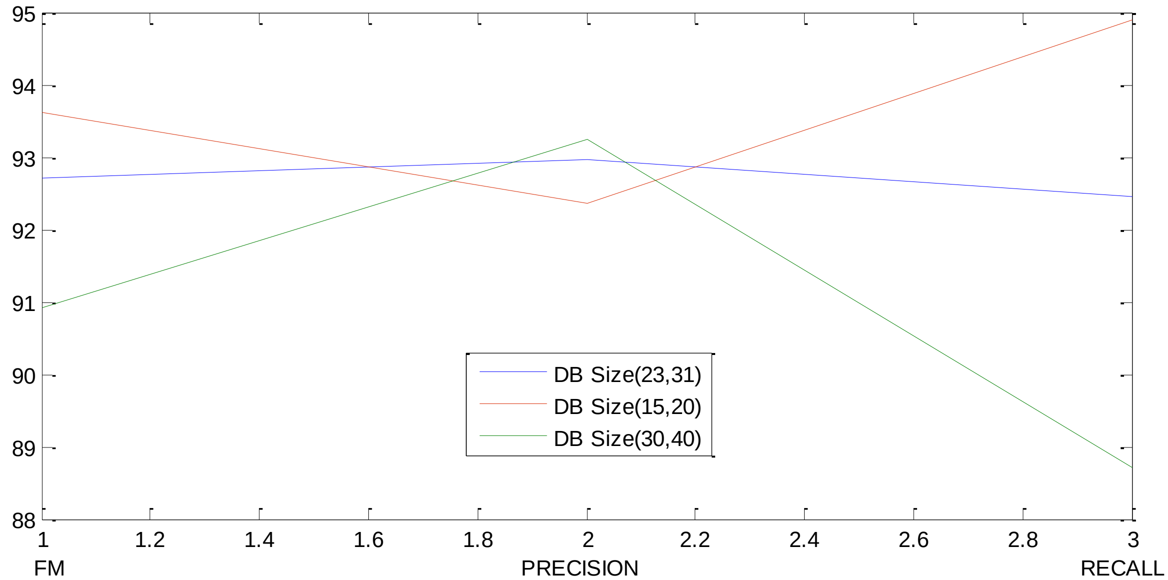


Fig 4.5.16 Results with different template size for Img4

Table 4.5 Results of Img1

Size	FM	Precision	Recall
17, 22	36.01	84.22	22.90
25, 33	35.85	84.87	22.73
15, 20	36.90	83.05	23.72

Table 4.6 Results of Img4

Size	FM	Precision	Recall
23, 31	92.70	92.96	92.45
30, 40	90.92	93.25	88.70
15, 20	93.61	92.36	94.89

Table 4.7 Results of Img2

Size	FM	Precision	Recall
12, 16	73.88	83.87	66.02
38, 50	65.08	75.94	56.94
33, 44	70.51	76.96	65.06
15, 20	77.75	78.47	77.04

Table 4.8 Results of Img3

Size	FM	Precision	Recall
30, 40	89.53	82.84	97.39
75, 100	86.78	82.42	91.63
27, 36	89.77	82.73	98.12
15, 20	90.35	83.15	98.92

CHAPTER-5

EXPERIMENTAL RESULTS & DISCUSSION

This chapter represents the results of the proposed algorithm of “text segmentation and binarization” was tested using most difficult document images. These document images are collected from three different categories: document images captured using a Mobile Phone, document images were selected from DIBCO and H-DIBCO datasets. The first section presents an Experimental setup. The second section presents Mobile Image results and DIBCO dataset results of the proposed algorithm. The fourth section presents a performance analysis, including precision, recall, and F-Measure.

5.1 Experimental Setup

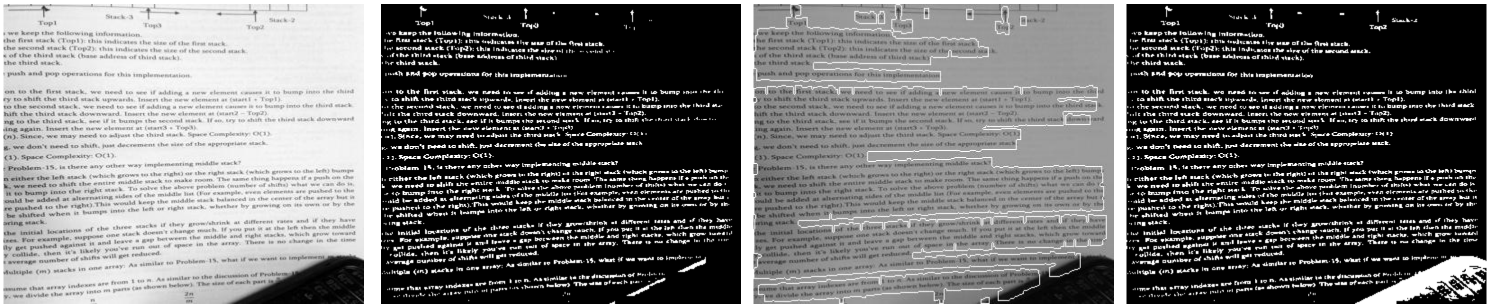
The system with Intel core i3 processor 2.40 GHz, RAM to 4.00 GB and 64-bit operating system and MATLAB 2012 tool are used to implement proposed algorithm. The following parameters are required to design, propose algorithm:

1.) Samsung Galaxy S4 mobile to capture images and DIBCO datasets.
2.) Resize input image with 300×400.
3.) Resize template image with 15×20.
4.) Define gamma value 0.97 (i.e. 1/g = 0.97).
5.) In contrast stretching for control the slope of the function (E), E=4.

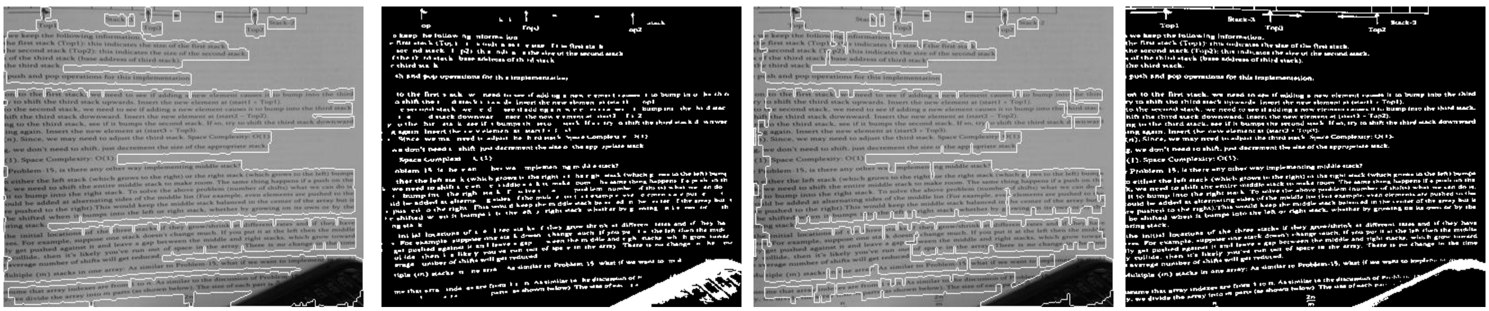
5.2 Proposed Algorithm Results

The results of the proposed method are divided into two categories: Mobile Image Results and DIBCO datasets results.

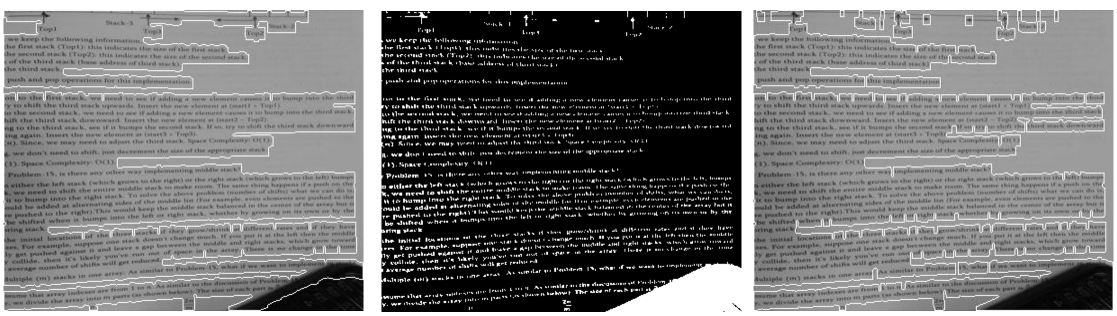
5.2.1 Mobile Image Results



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

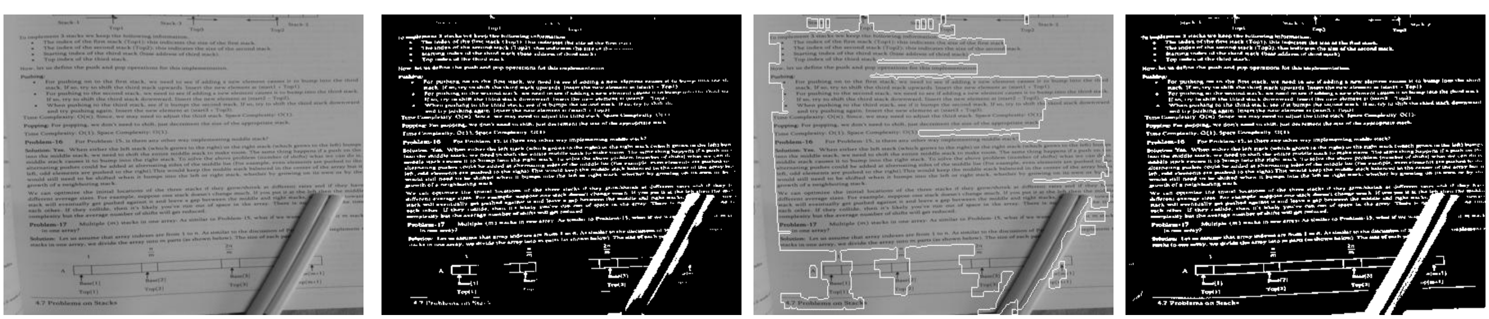


(e) (f) (g) (h)

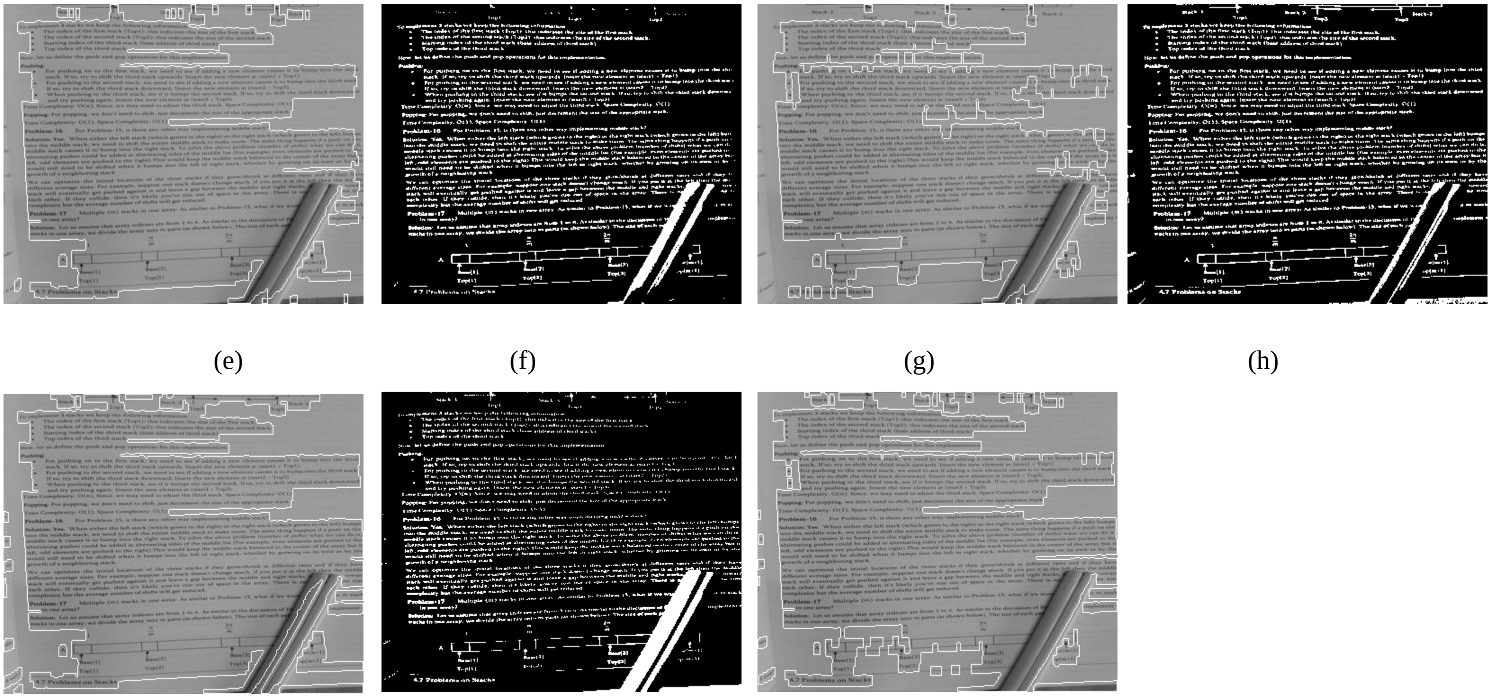


(i) (j) (k)

Fig (5.1)



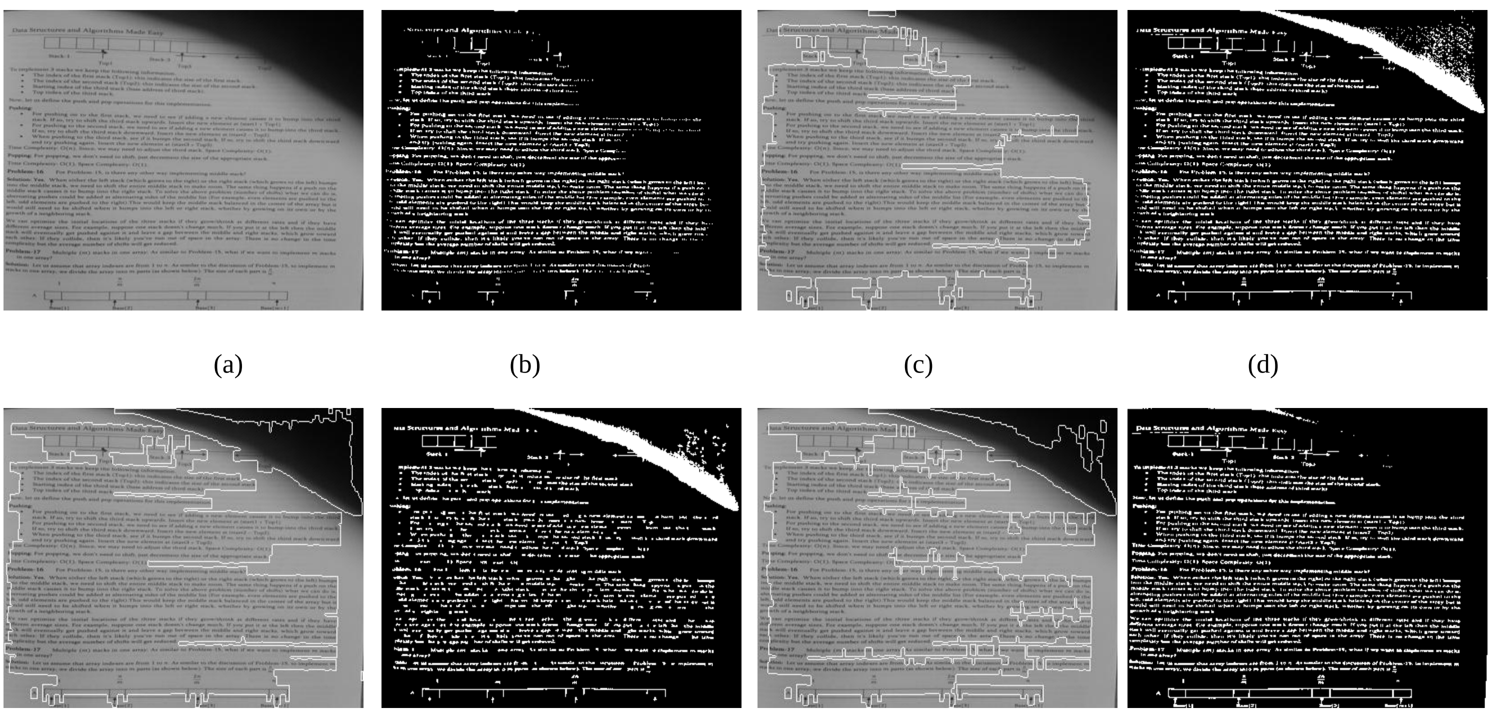
(a) (b) (c) (d)



(e) (f) (g) (h)

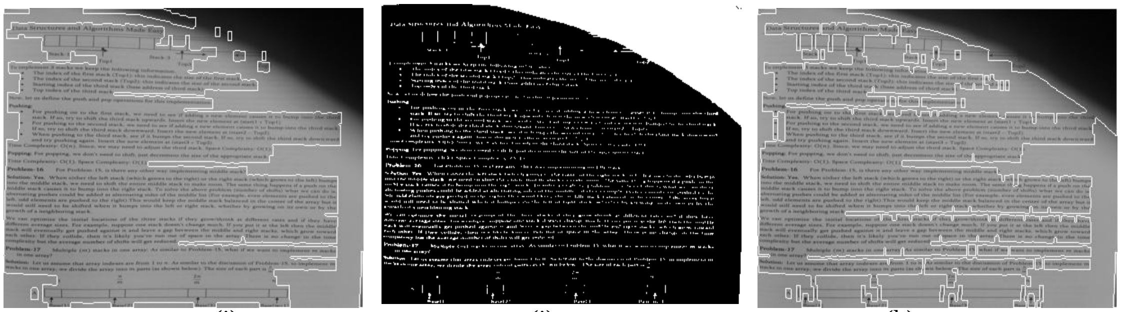
(i) (j) (k)

Fig (5.2)



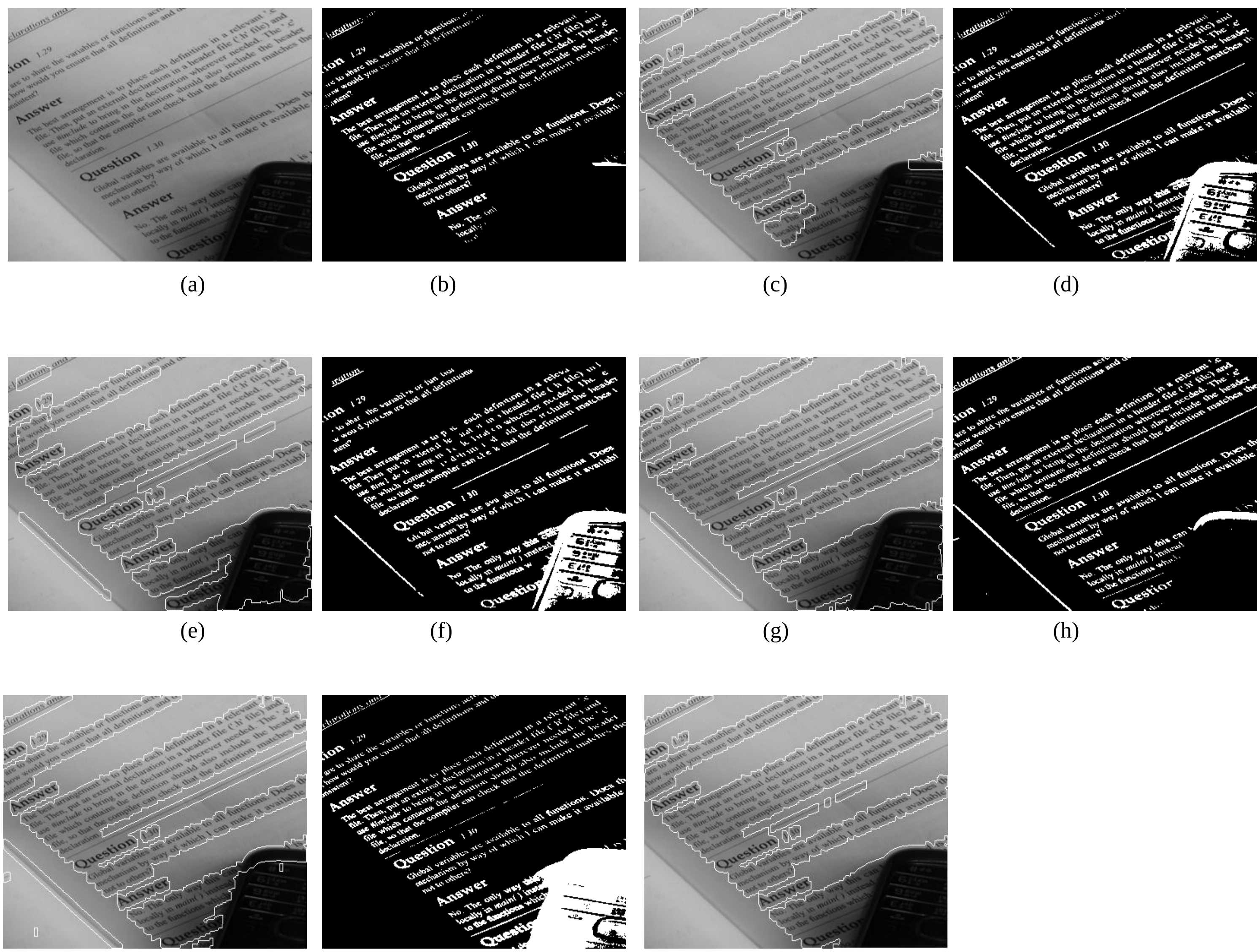
(a) (b) (c) (d)

(e) (f) (g) (h)



(i) (j) (k)

Fig (5.3)

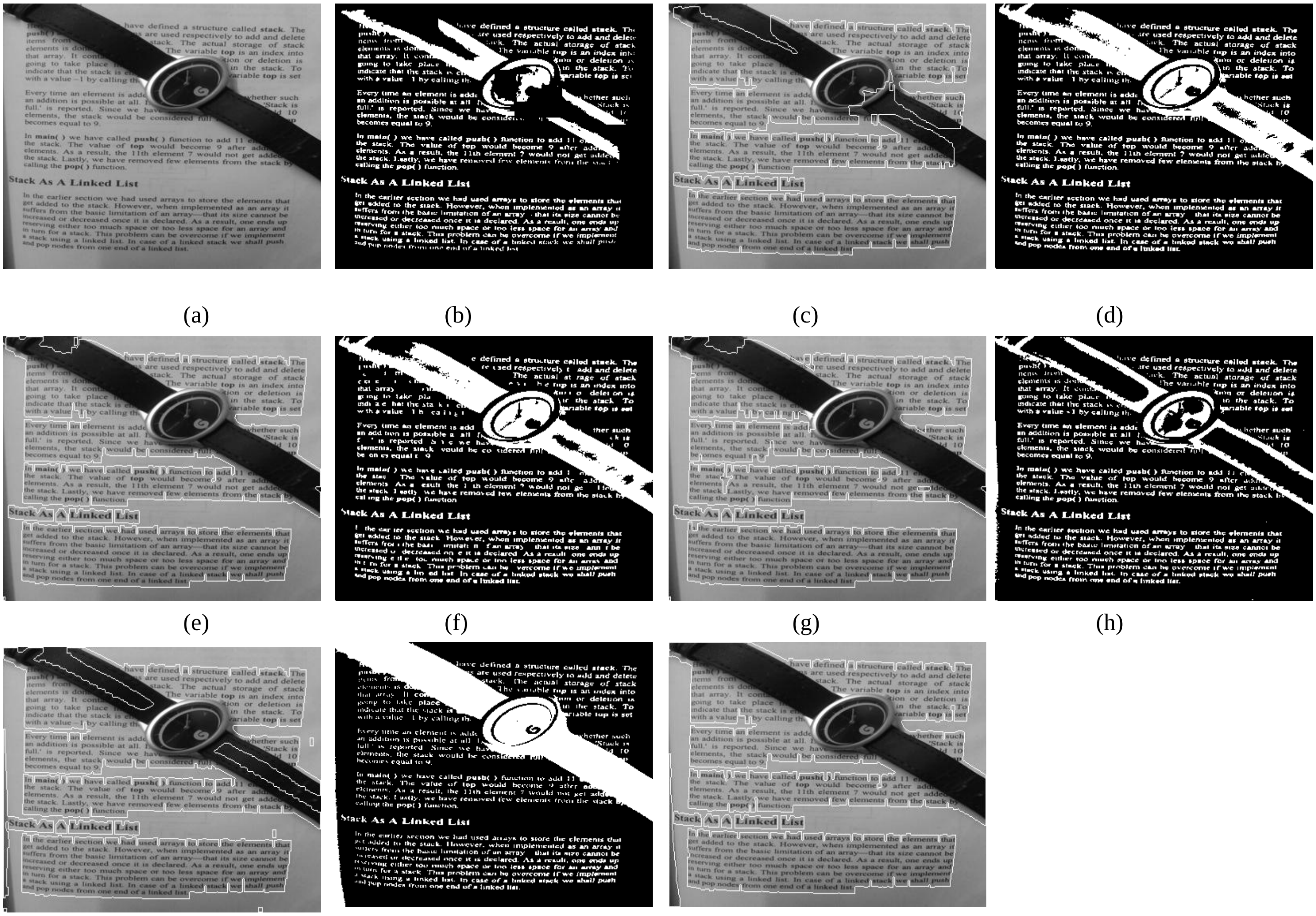


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k)

Fig (5.4)

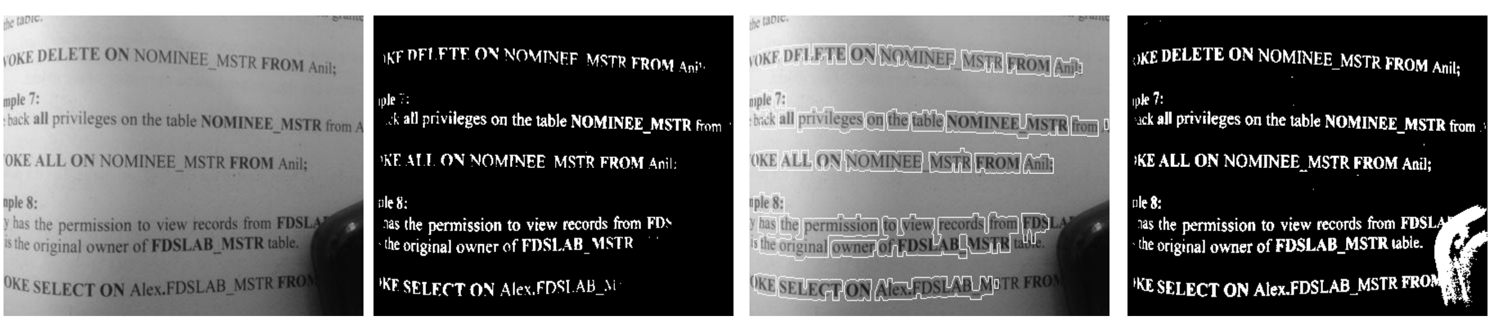


(a) (b) (c) (d)

(e) (f) (g) (h)

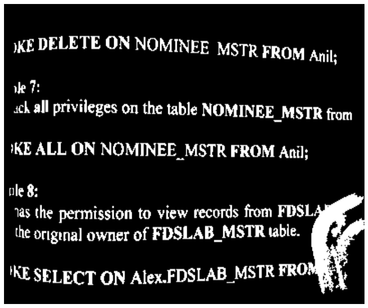
(i) (j) (k)

Fig (5.5)



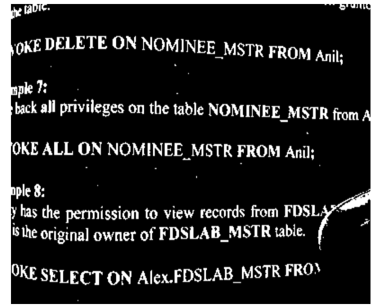
(a) (b) (c) (d)

(e)



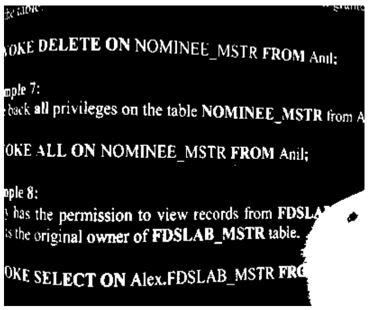
(f)

(g)

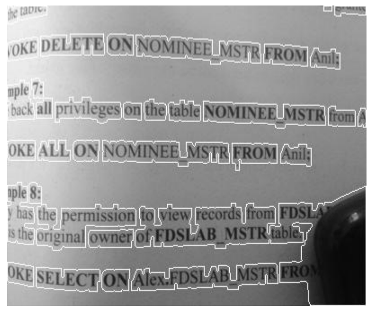


(h)

(i)

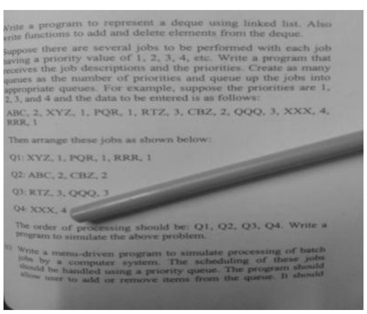


(j)

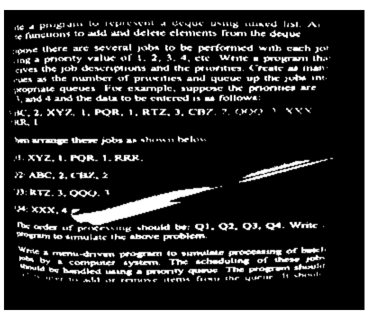


(k)

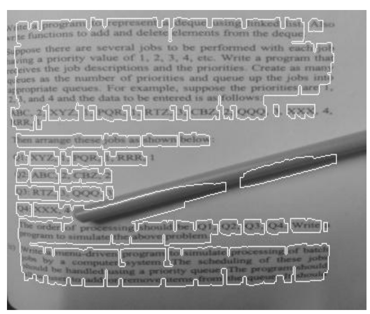
Fig (5.6)



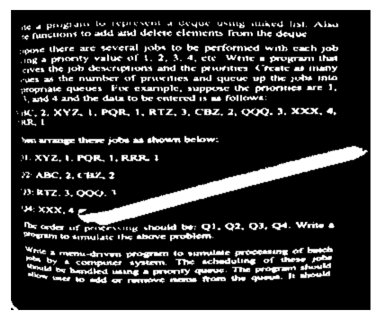
(a)



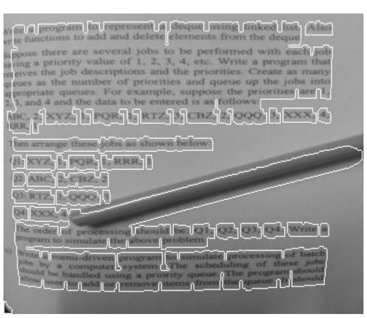
(b)



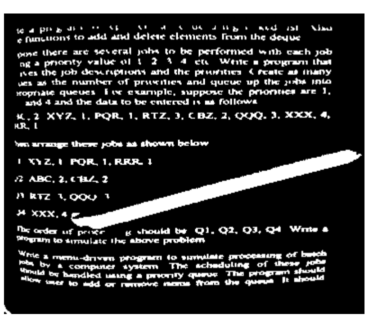
(c)



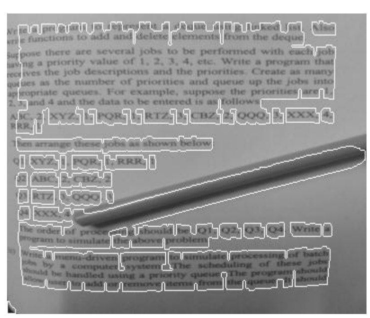
(d)



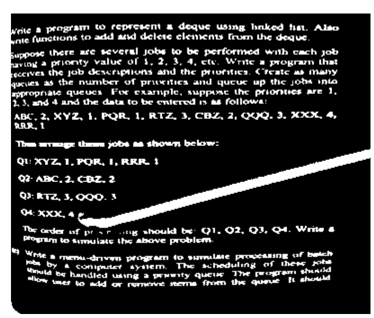
(e)



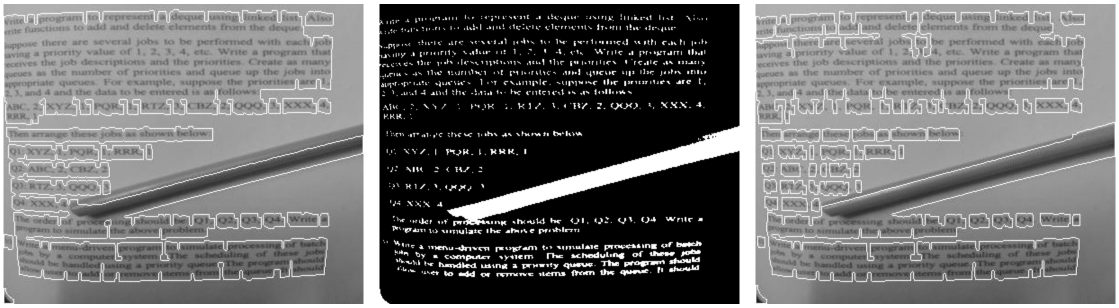
(f)



(g)

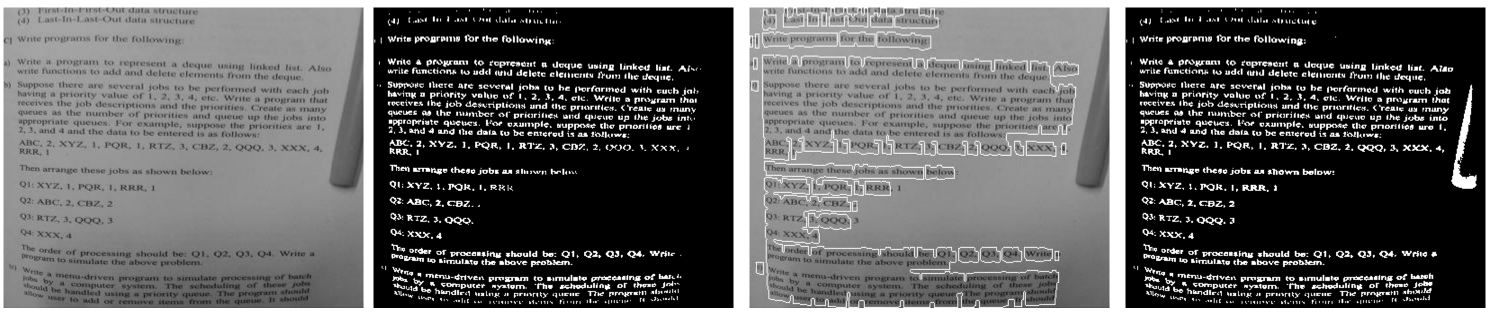


(h)

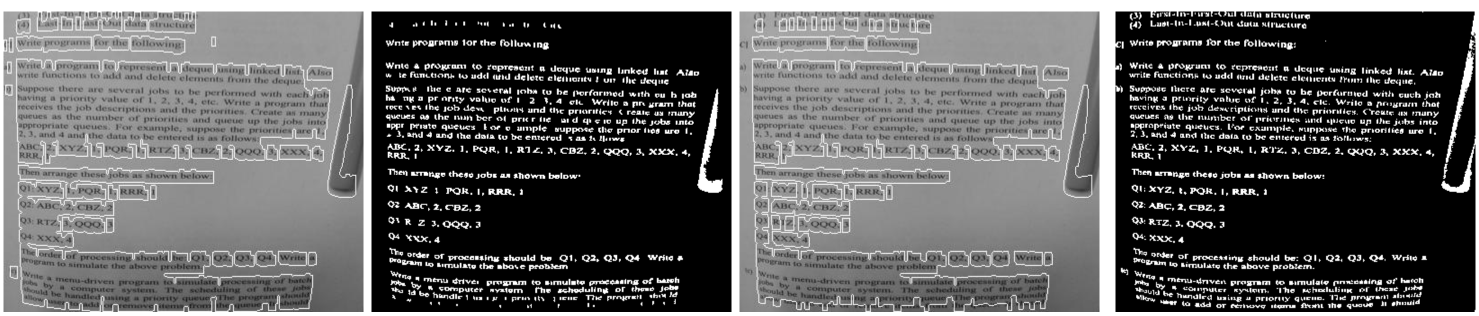


(i) (j) (k)

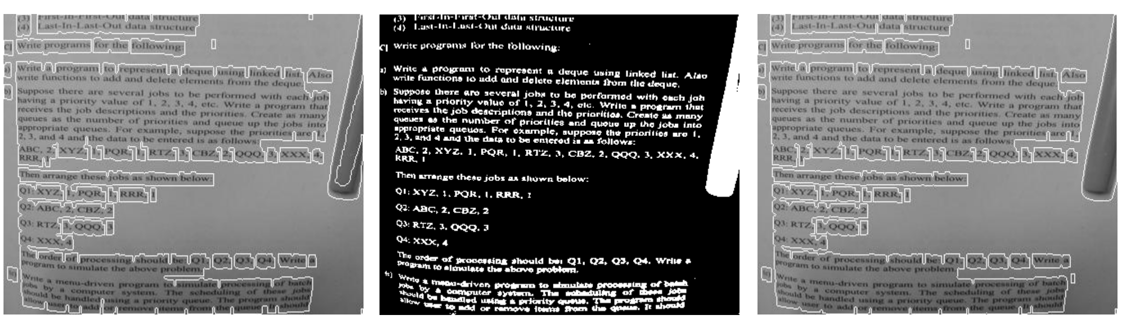
Fig (5.7)



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

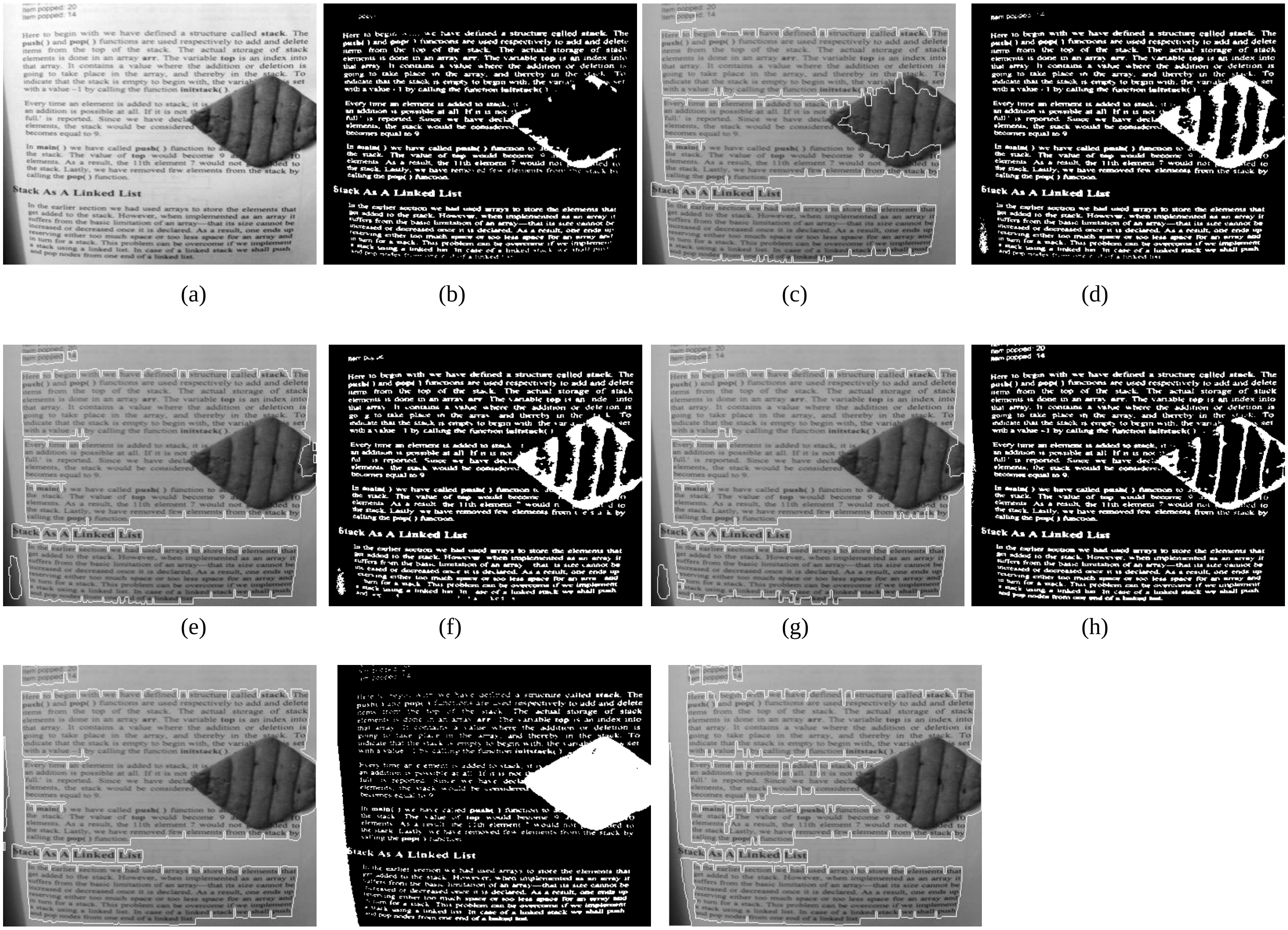


(e) (f) (g) (h)



(i) (j) (k)

Fig (5.8)

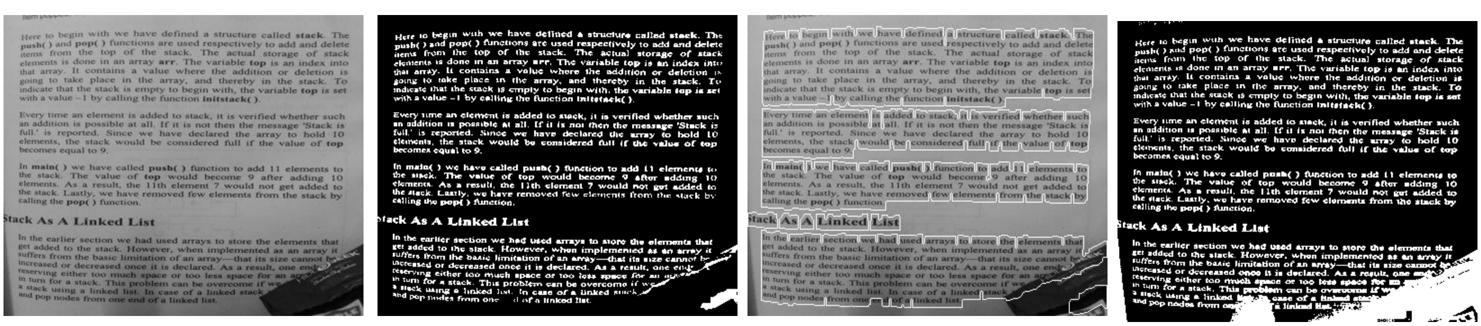


(a) (b) (c) (d)

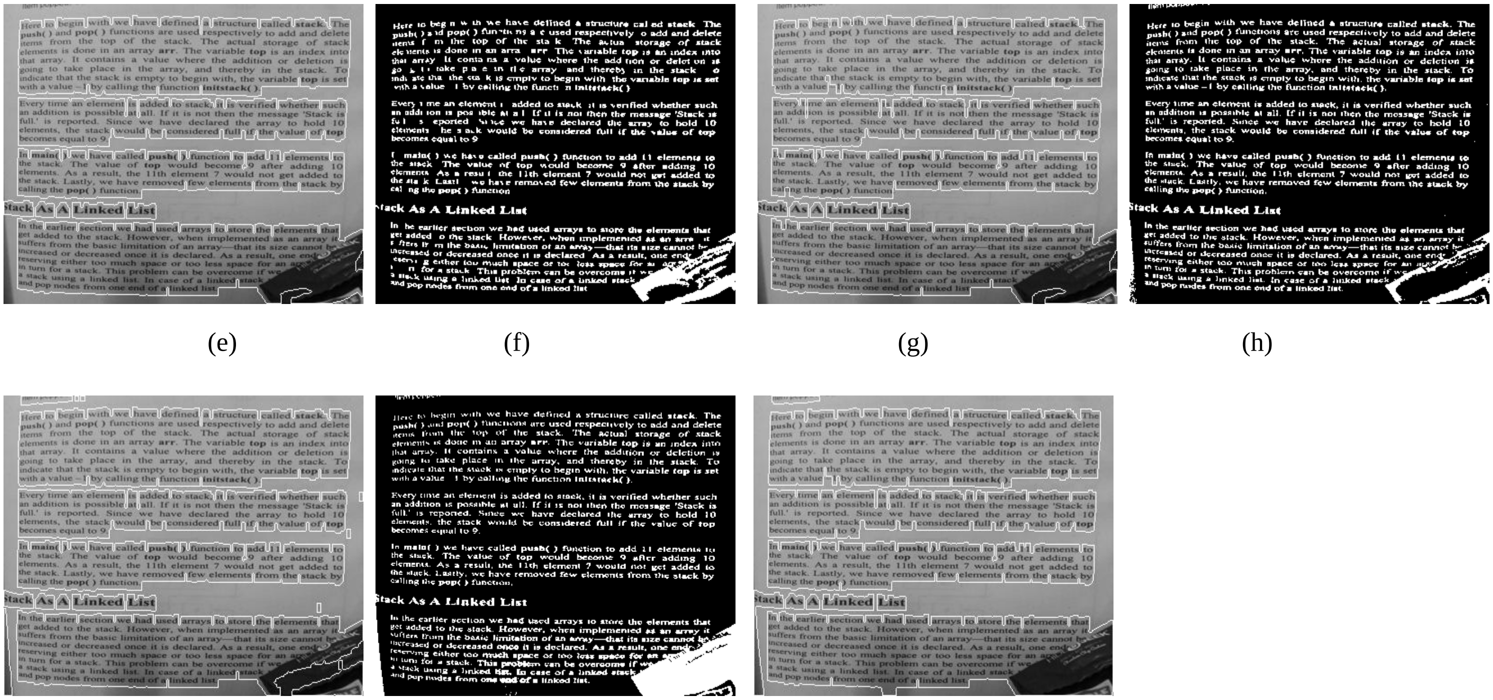
(e) (f) (g) (h)

(i) (j) (k)

Fig (5.9)



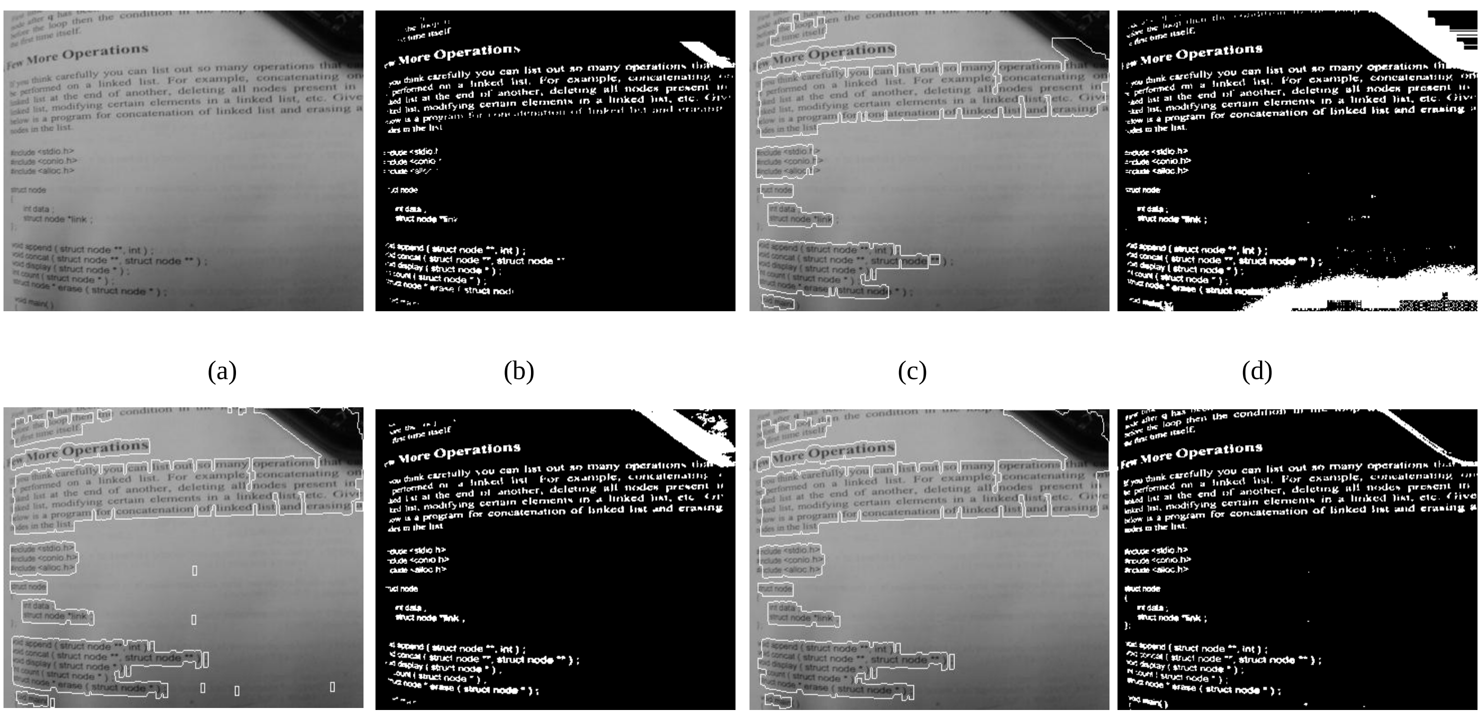
(a) (b) (c) (d)



(e) (f) (g) (h)

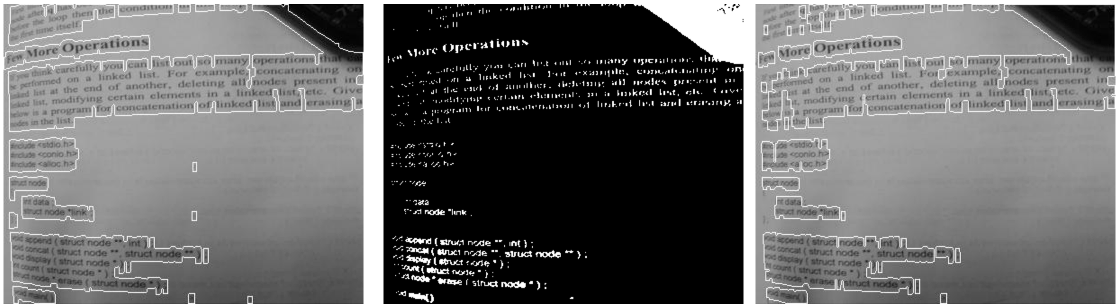
(i) (j) (k)

Fig (5.10)



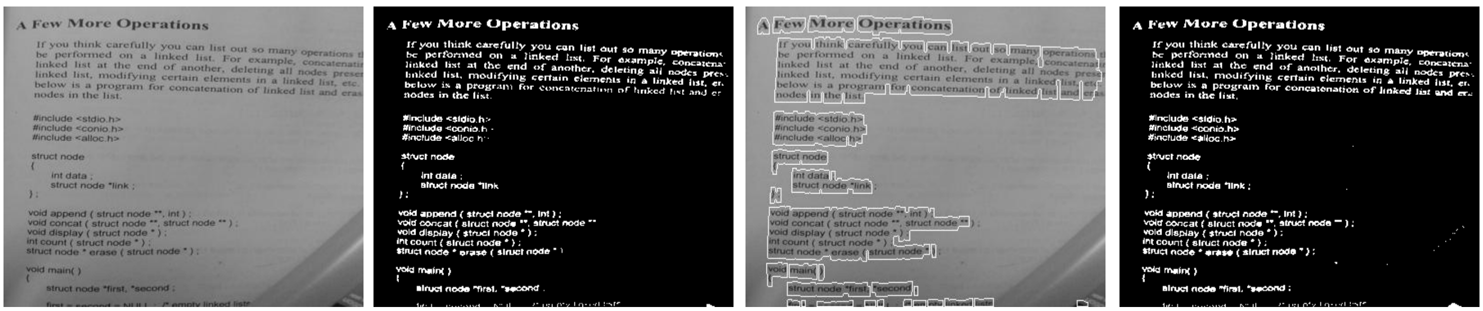
(a) (b) (c) (d)

(e) (f) (g) (h)

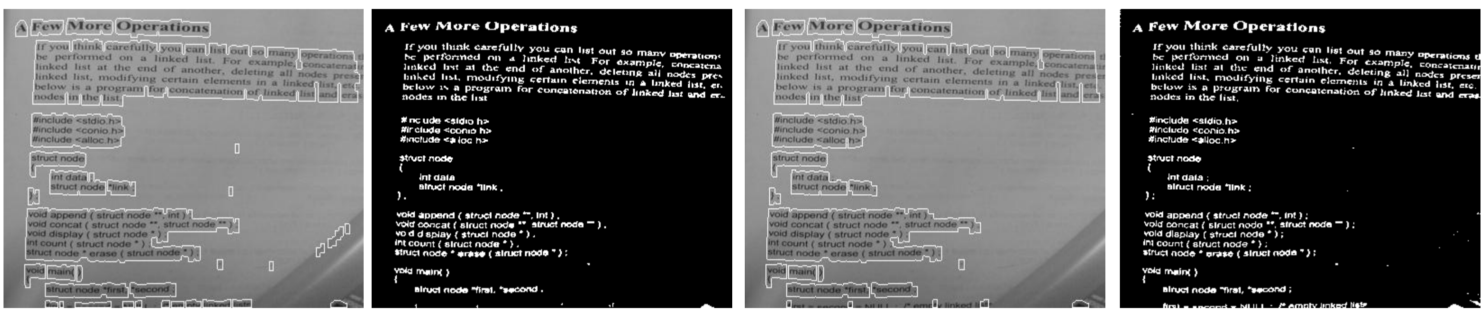


(i) (j) (k)

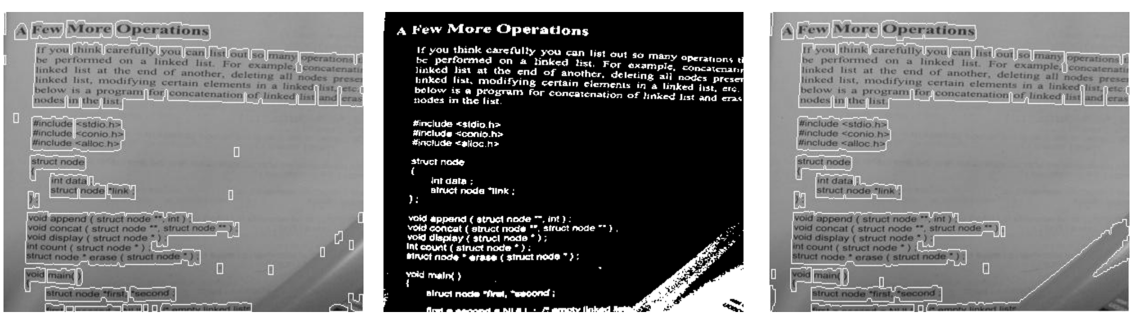
Fig (5.11)



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



(e) (f) (g) (h)



(i) (j) (k)

Fig (5.12)

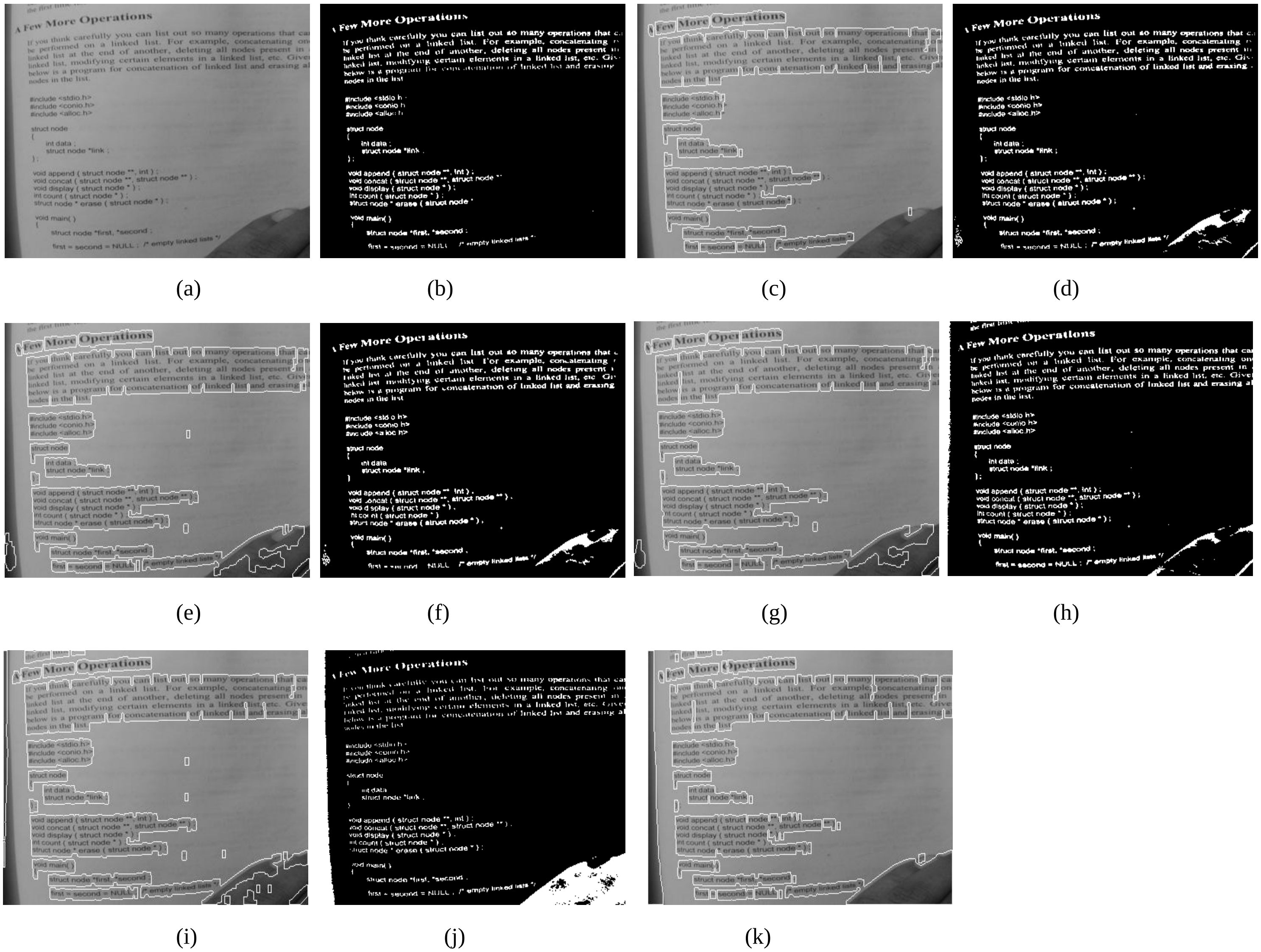
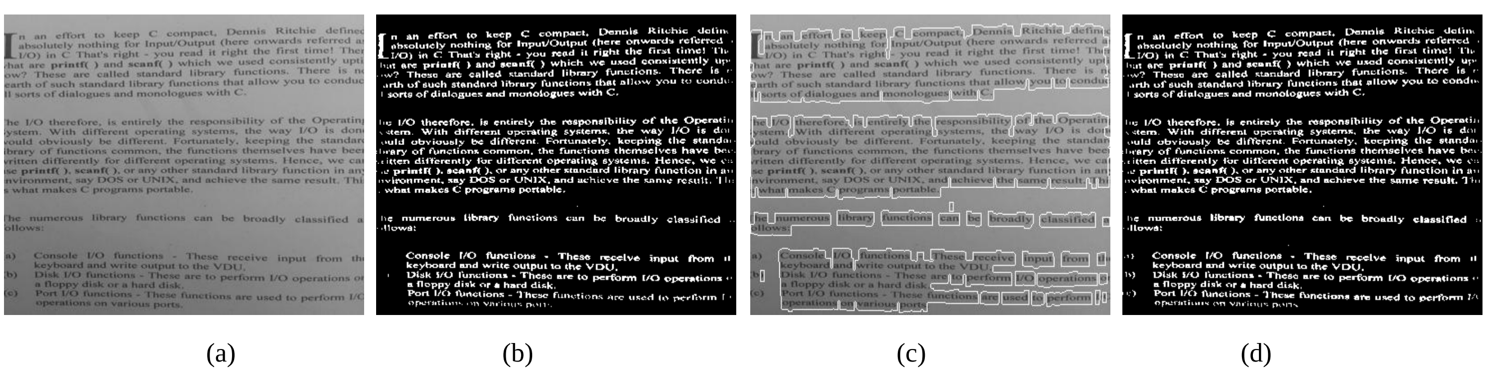
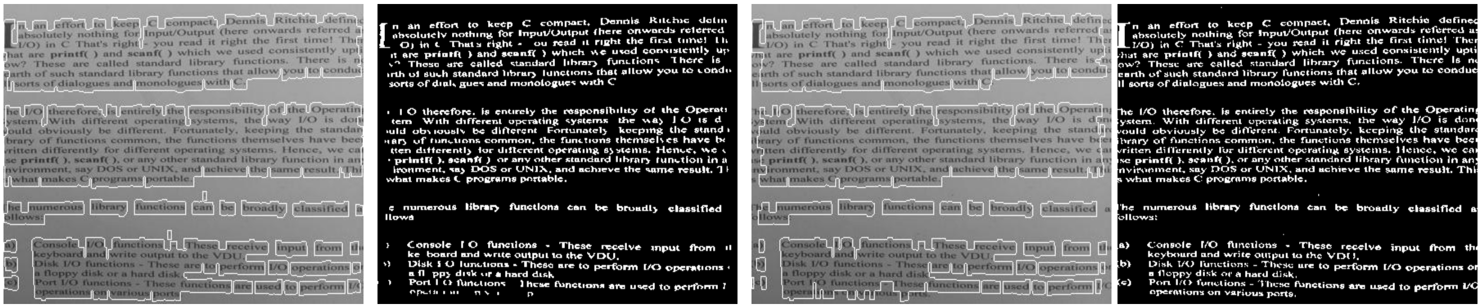
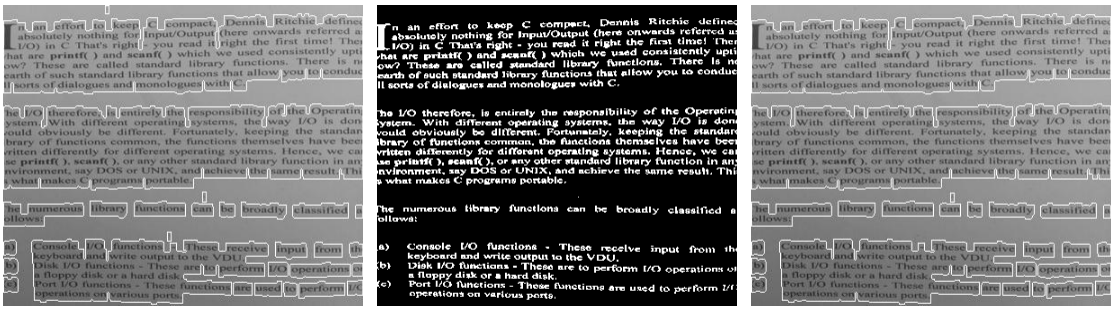


Fig (5.13)



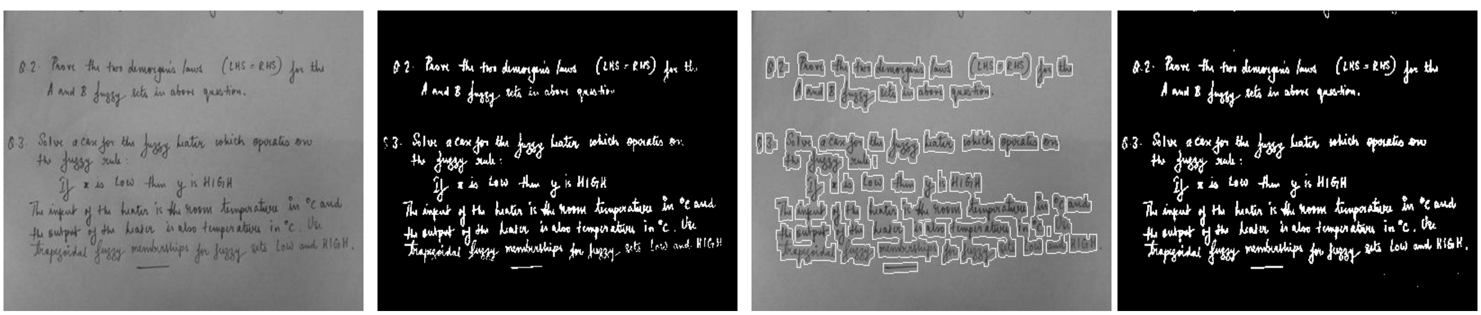


(e) (f) (g) (h)

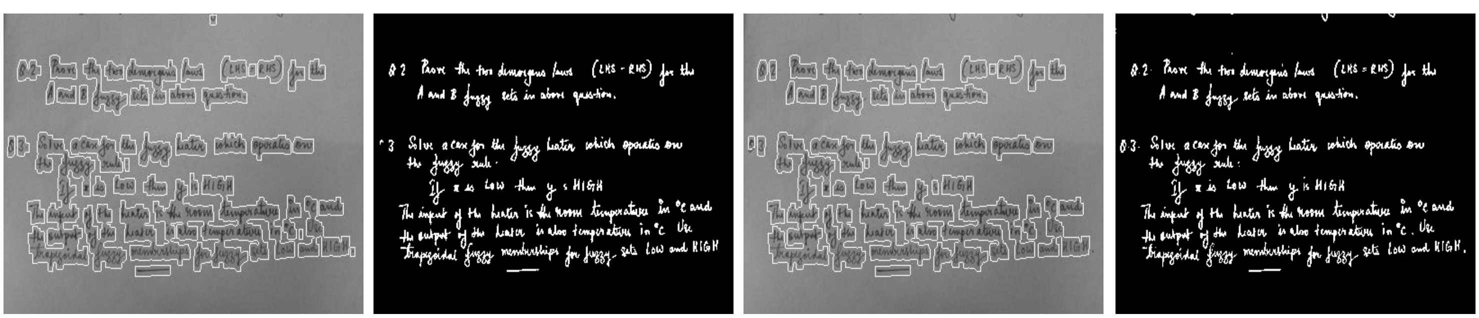


(i) (j) (k)

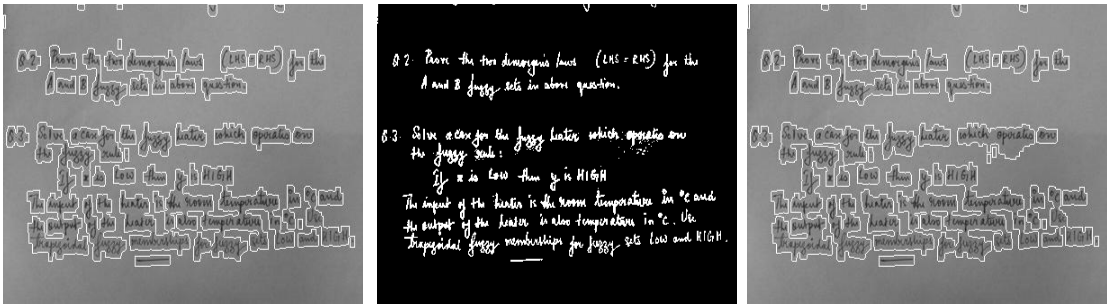
Fig (5.14)



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

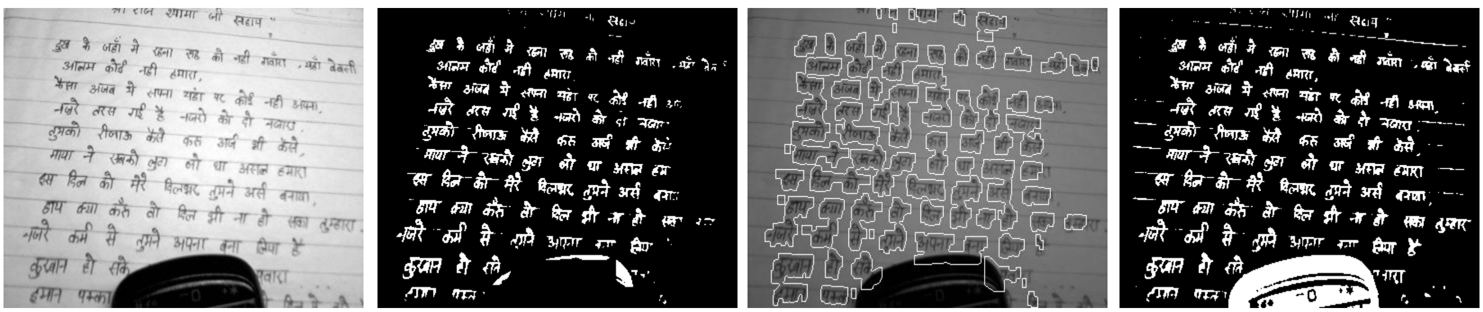


(e) (f) (g) (h)

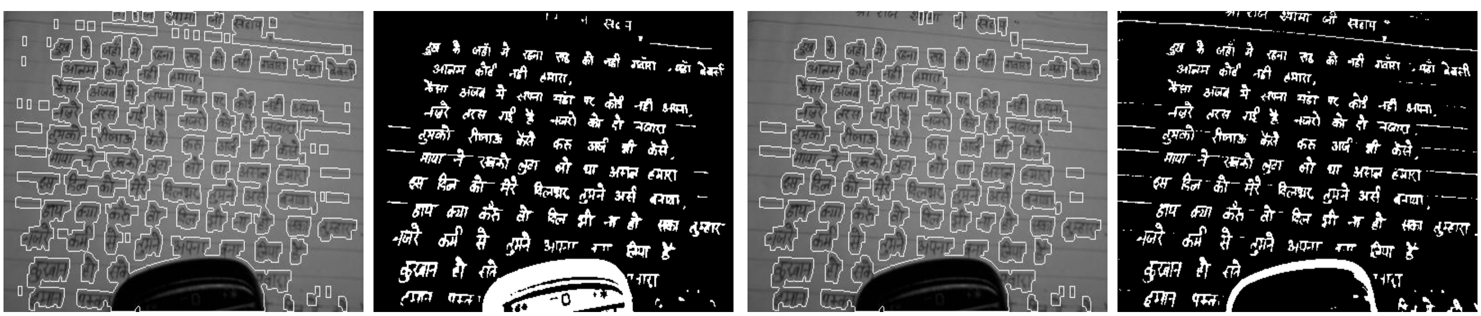


(i) (j) (k)

Fig (5.15)



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



(e) (f) (g) (h)

(i) (j) (k)

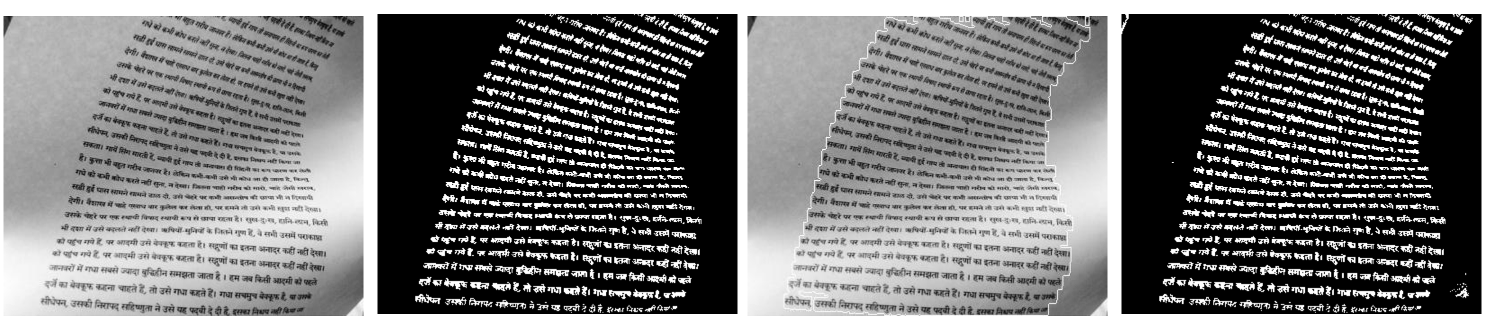
Fig (5.16)

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

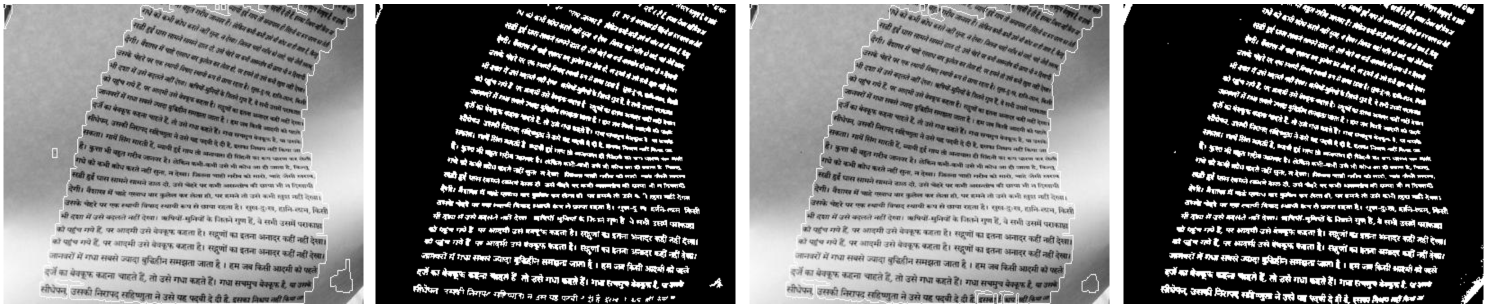
(e) (f) (g) (h)

(i) (j) (k)

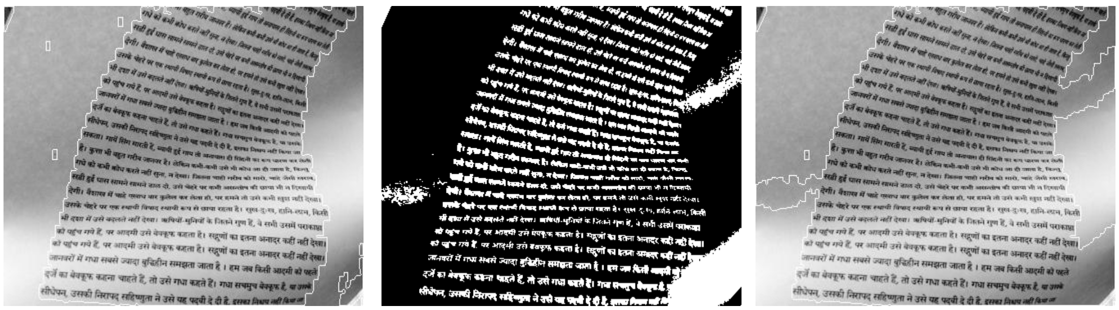
Fig (5.17)



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

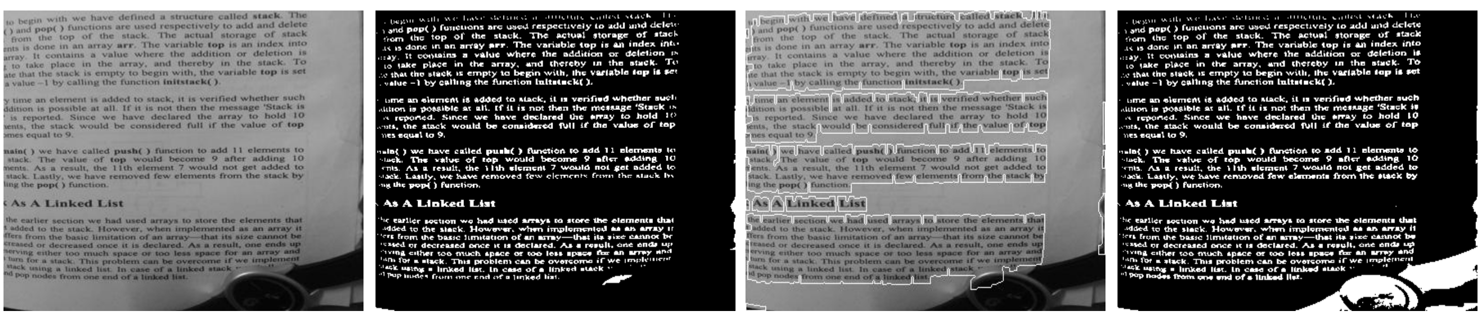


(e) (f) (g) (h)

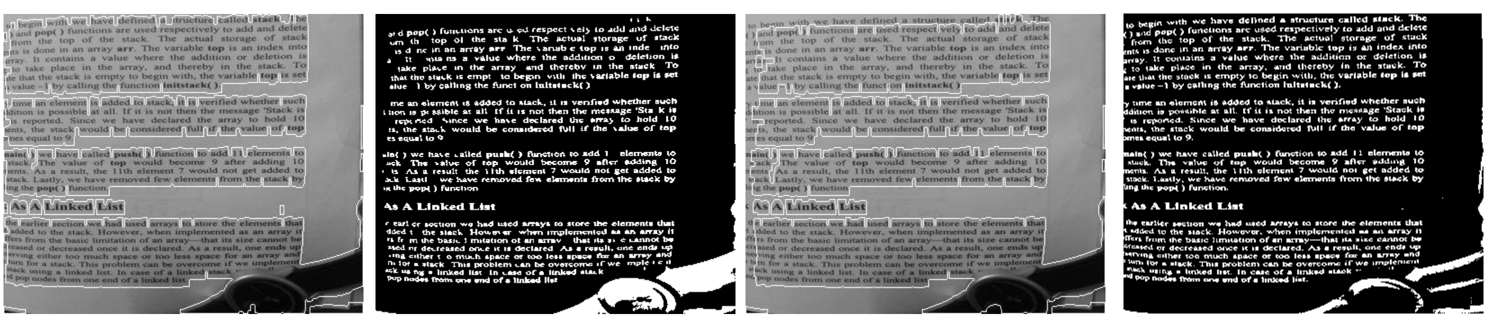


(i) (j) (k)

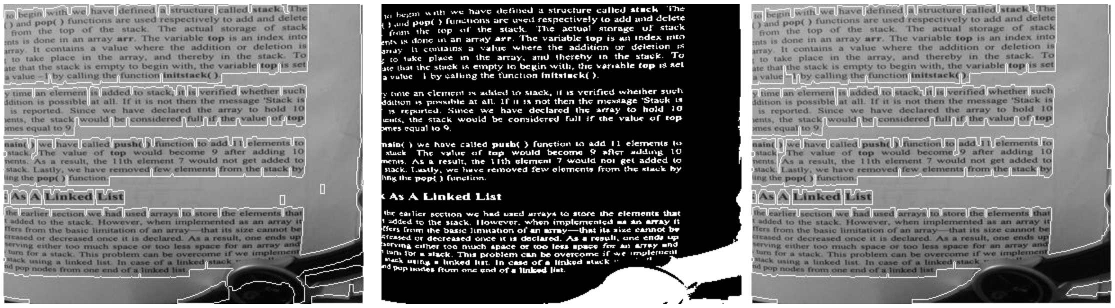
Fig (5.18)



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

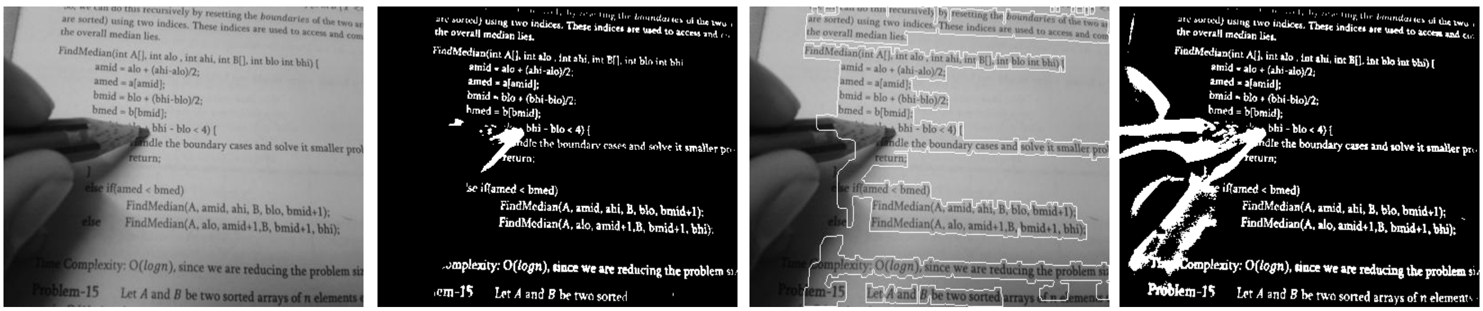


(e) (f) (g) (h)

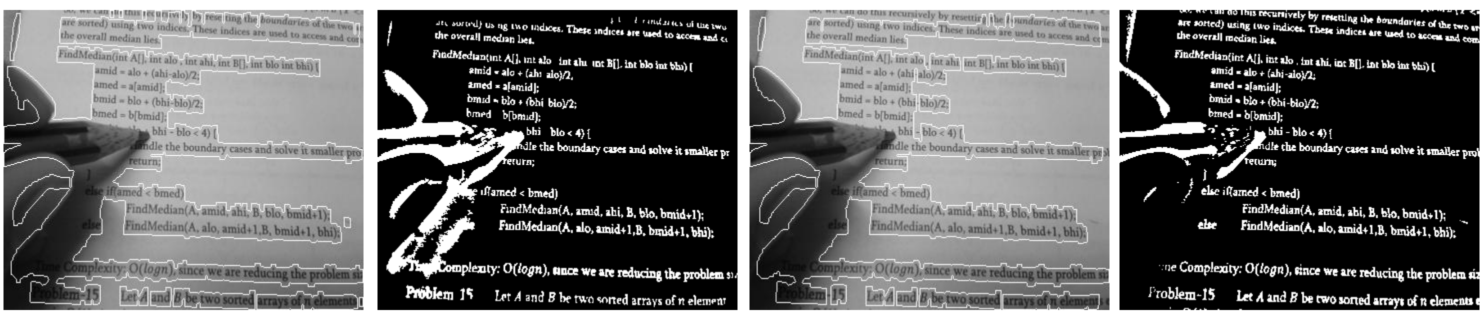


(i) (j) (k)

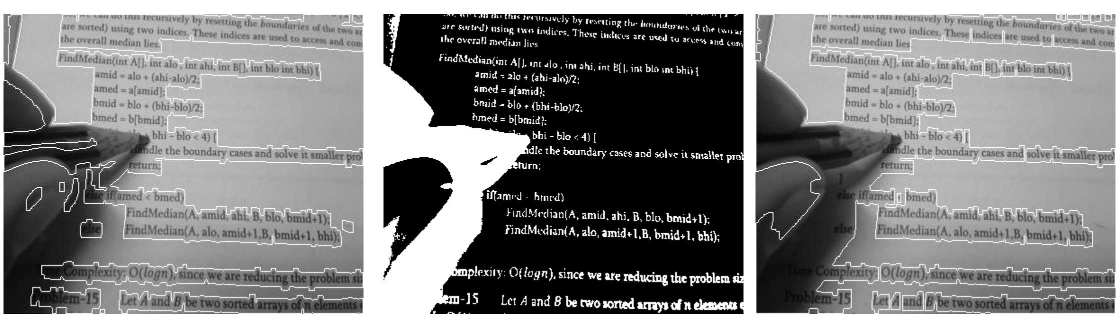
Fig (5.19)



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

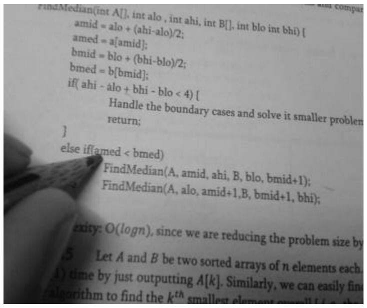
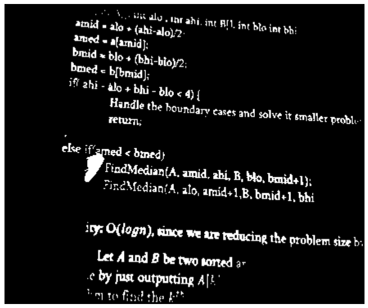
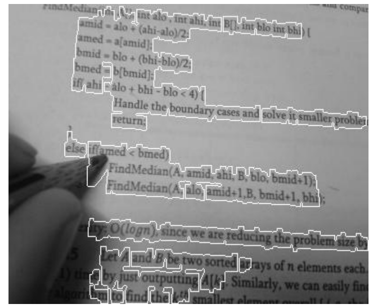
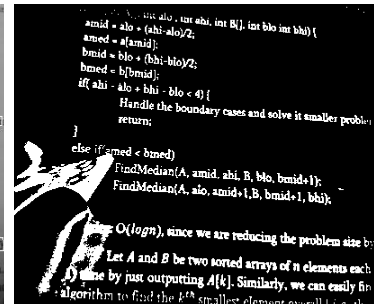


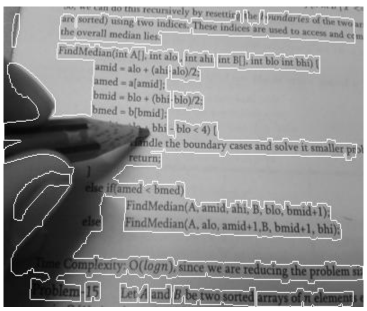
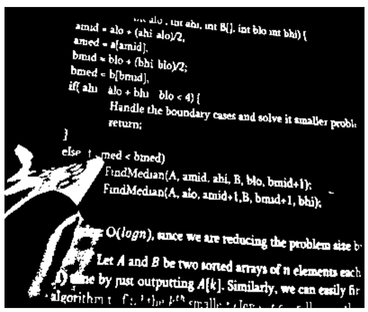
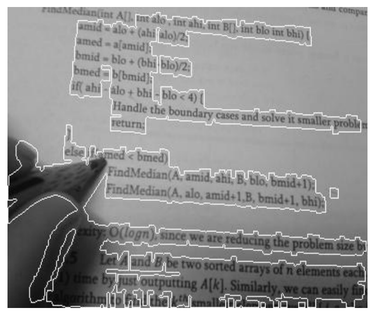
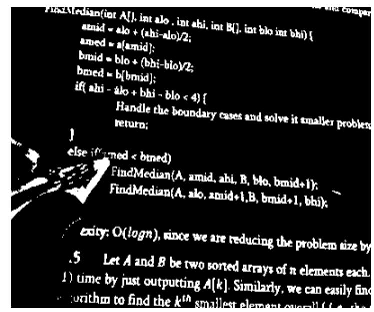
(e) (f) (g) (h)



(i) (j) (k)

Fig (5.20)


(a)

(b)

(c)

(d)


(e)

(f)

(g)

(h)

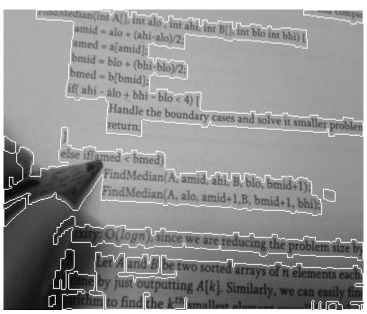
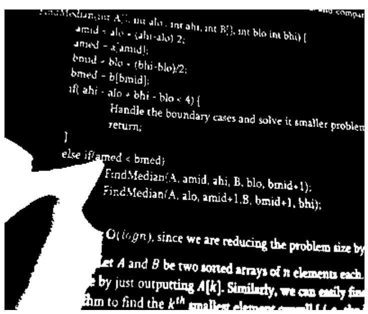
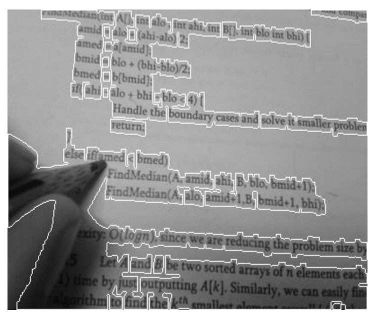
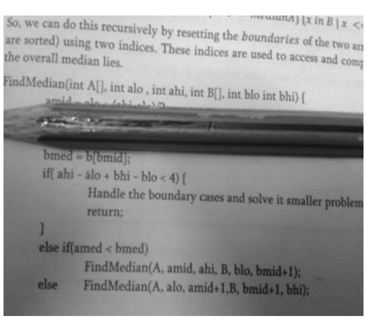
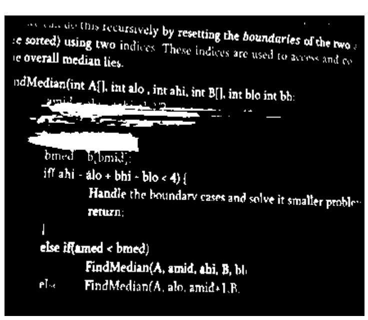
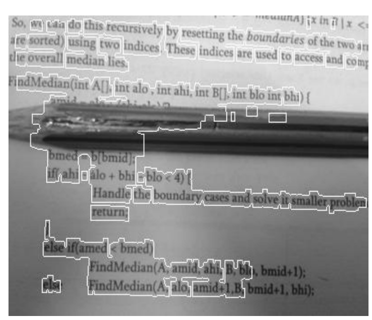
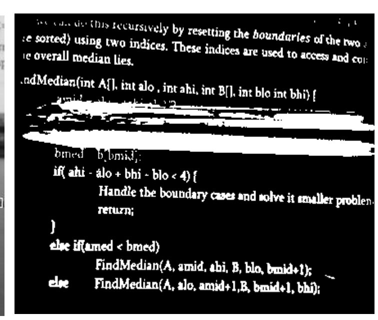
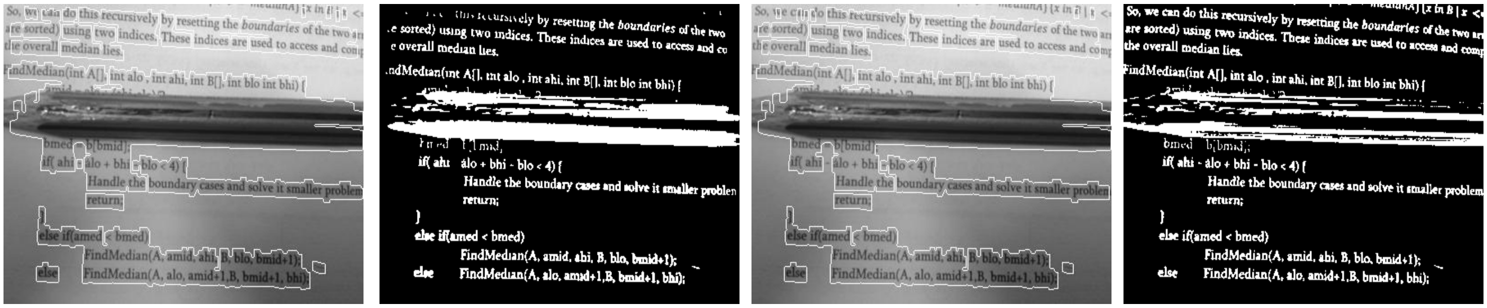
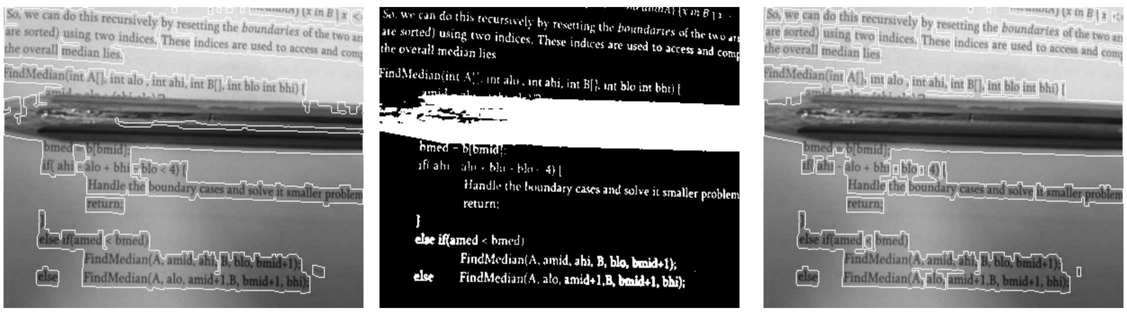

(i)

(j)

(k)

Fig (5.21)


(a)

(b)

(c)

(d)

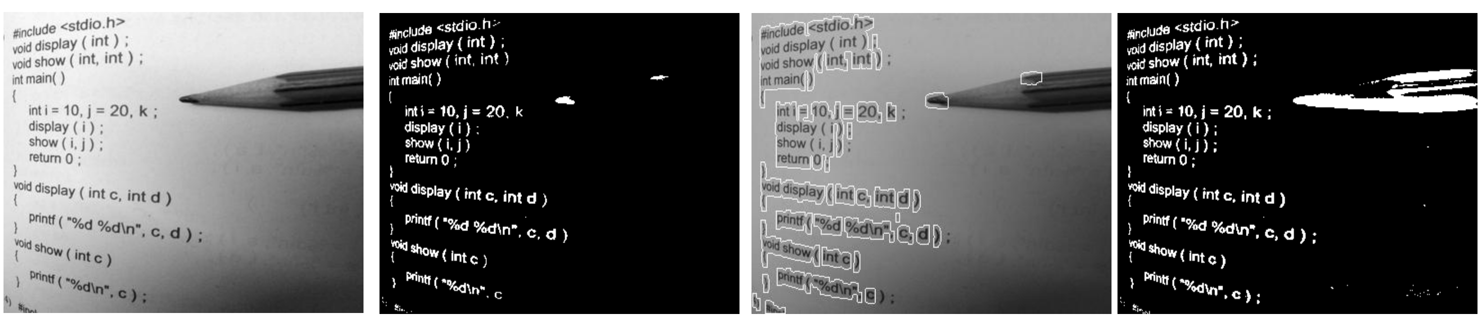


(e) (f) (g) (h)

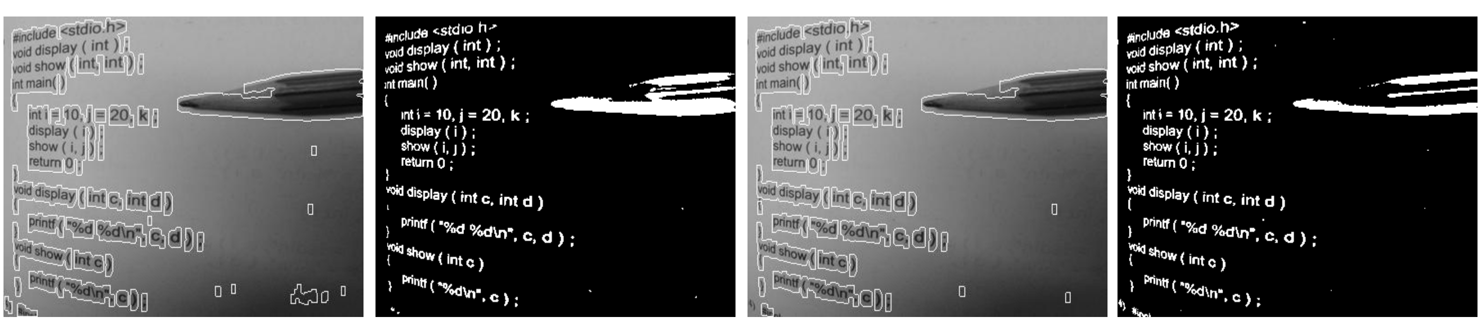


(i) (j) (k)

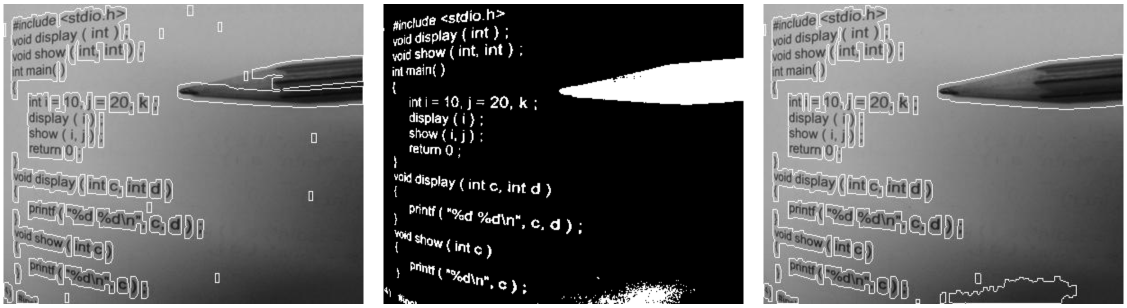
Fig (5.22)



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

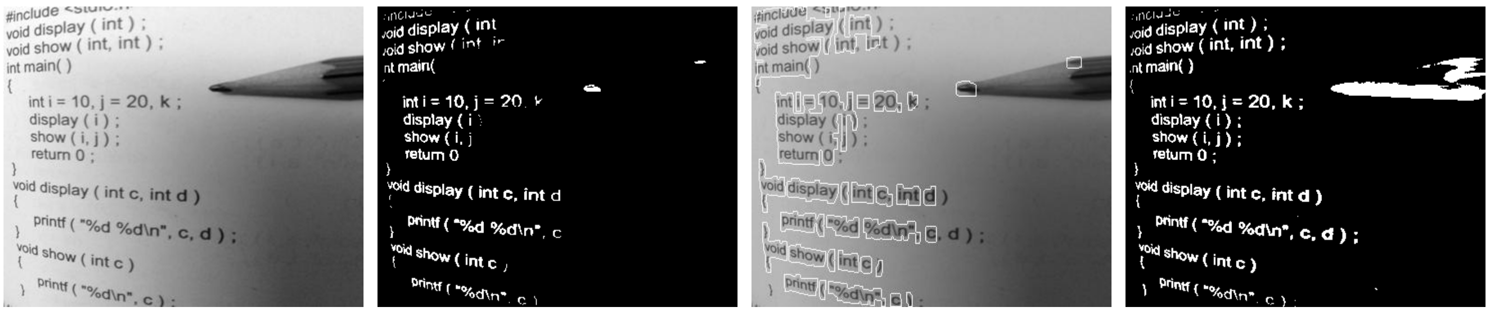


(e) (f) (g) (h)

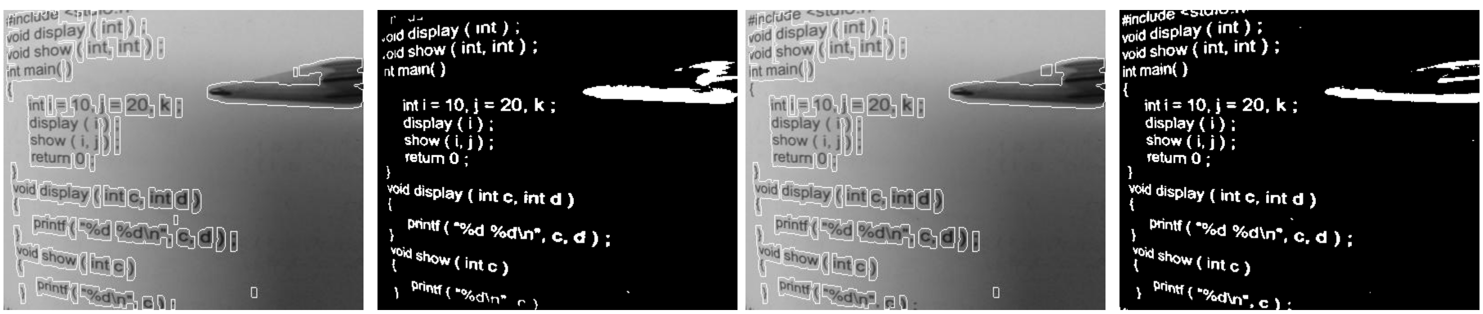


(i) (j) (k)

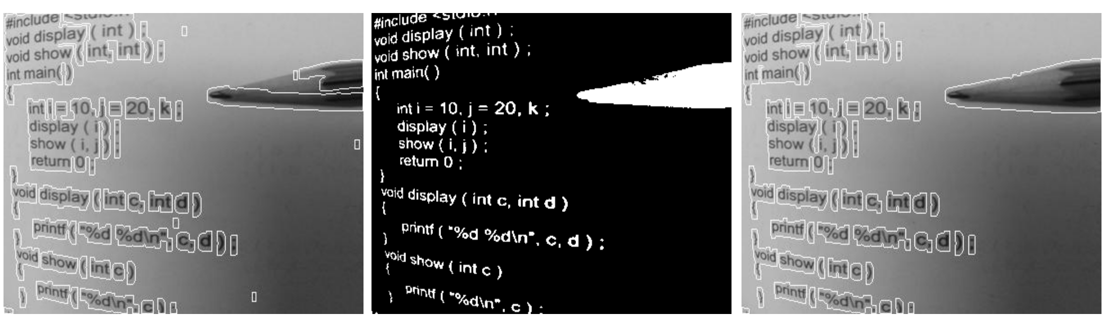
Fig (5.23)



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

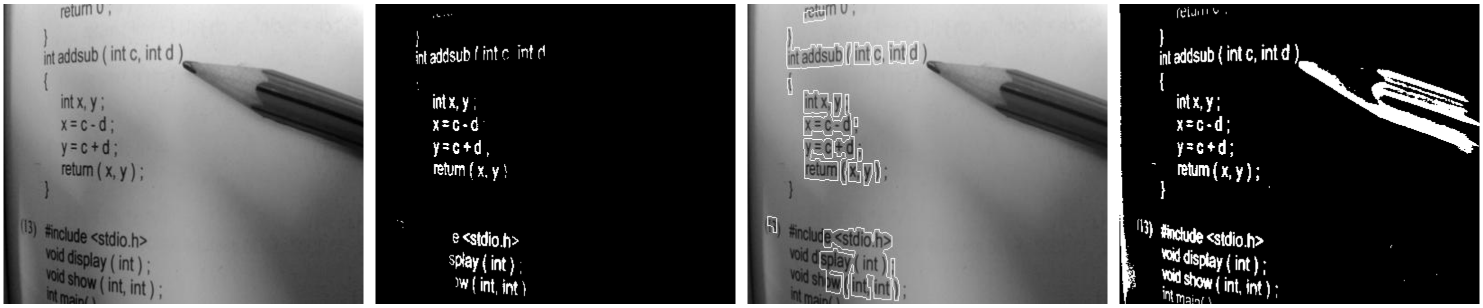


(e) (f) (g) (h)

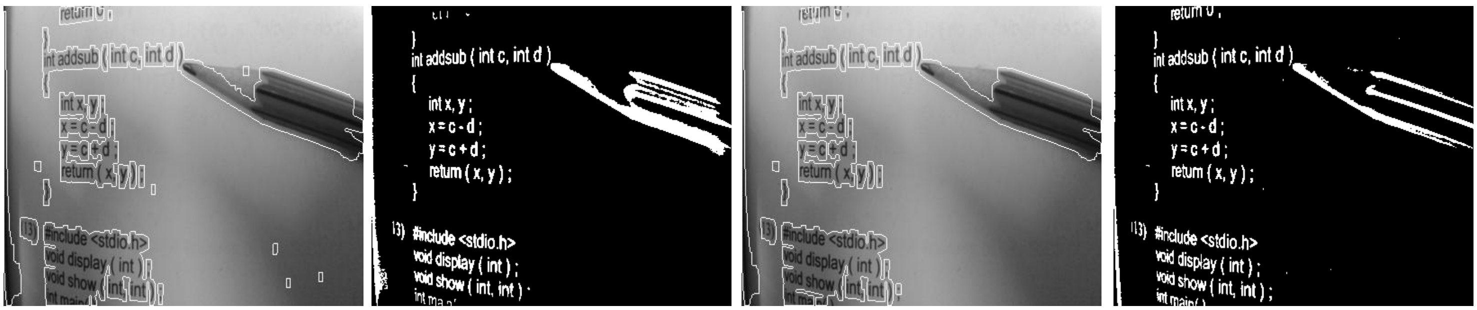


(i) (j) (k)

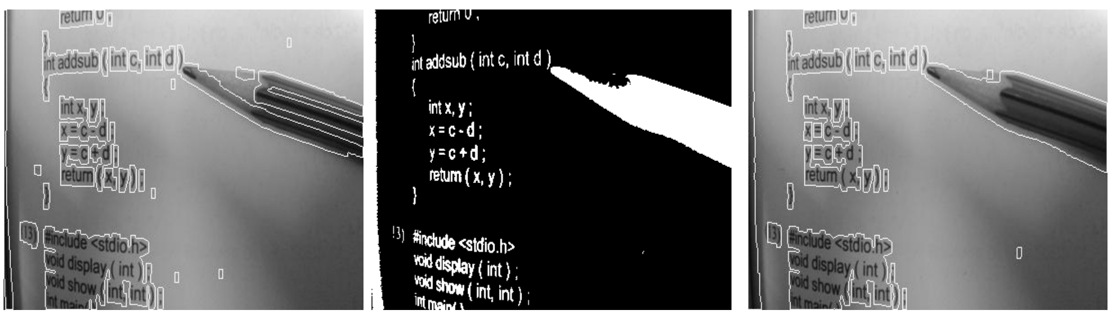
Fig (5.24)



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

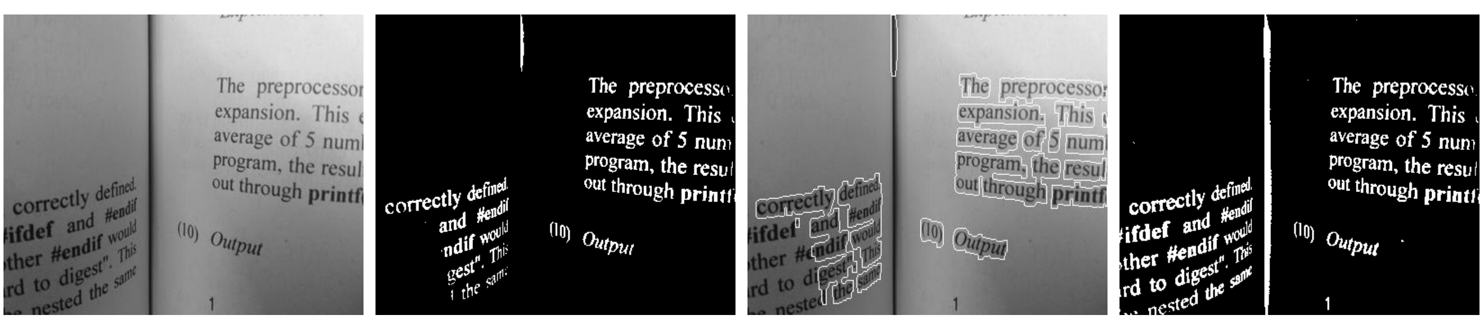


(e) (f) (g) (h)

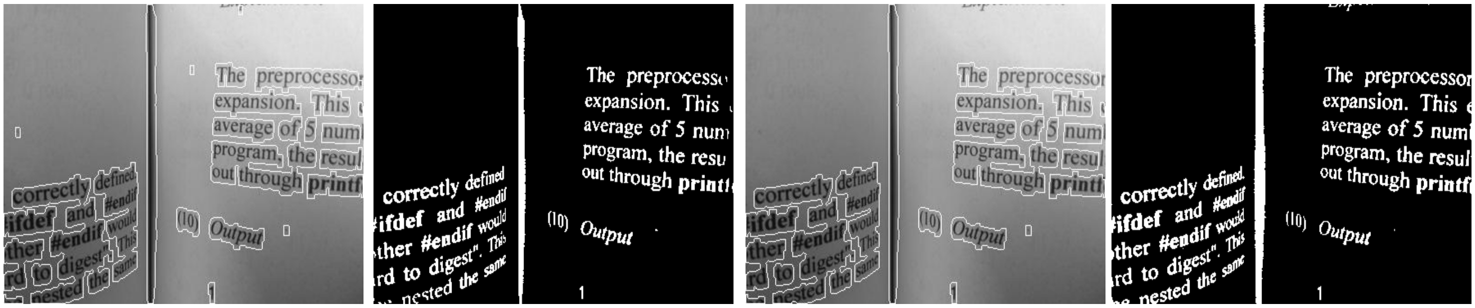


(i) (j) (k)

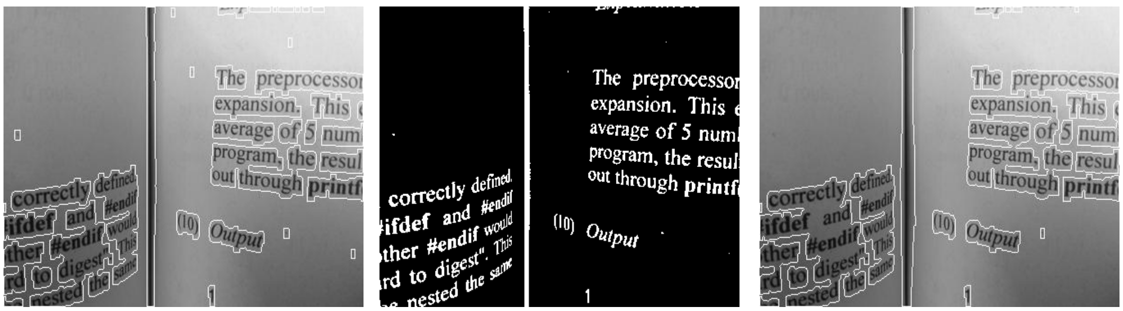
Fig (5.25)



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

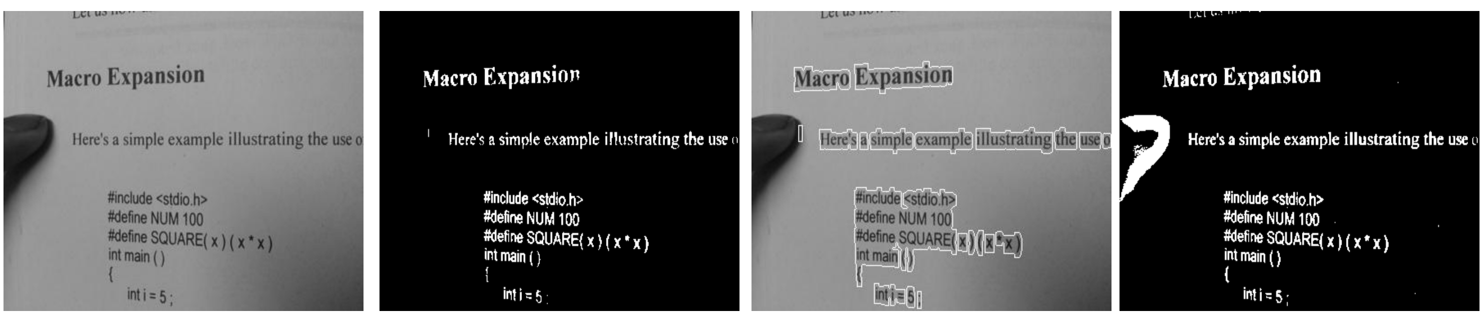


(e) (f) (g) (h)

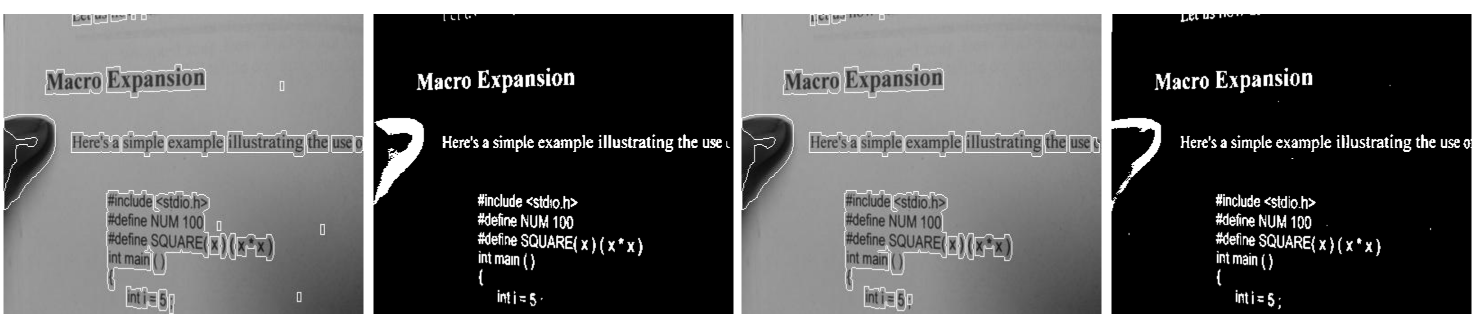


(i) (j) (k)

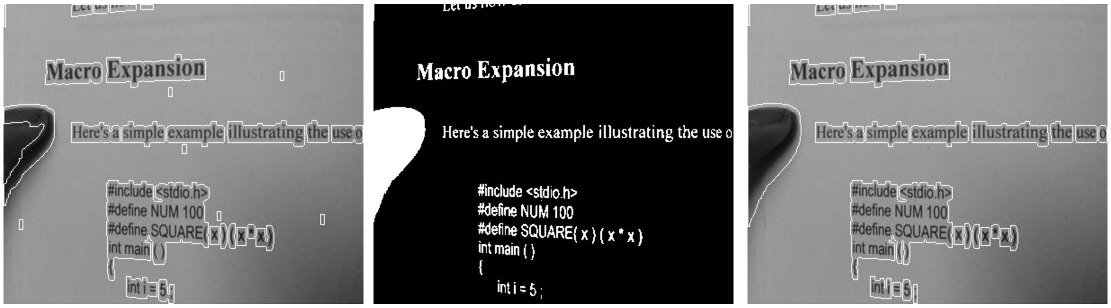
Fig (5.26)



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

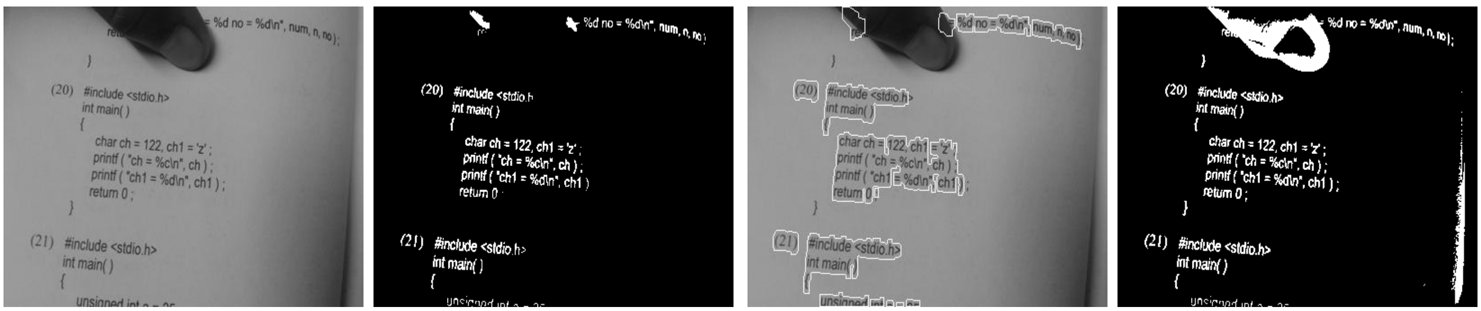


(e) (f) (g) (h)

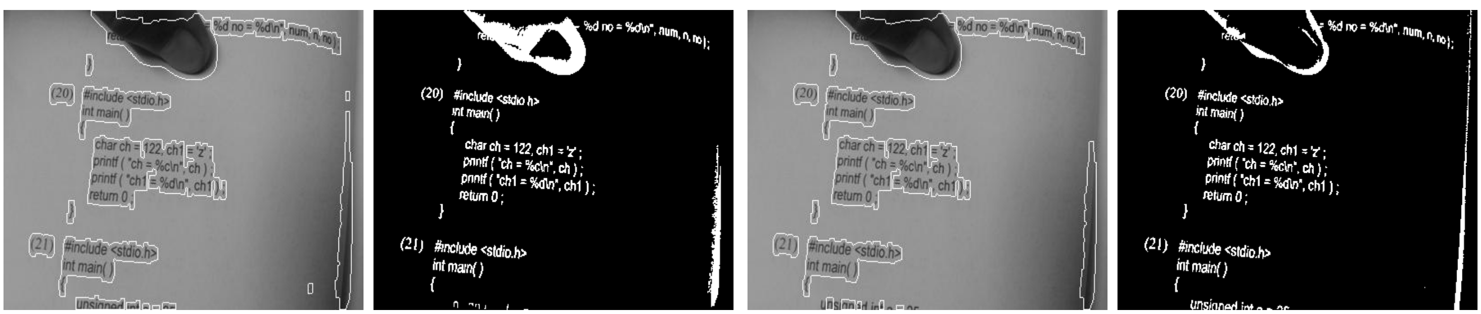


(i) (j) (k)

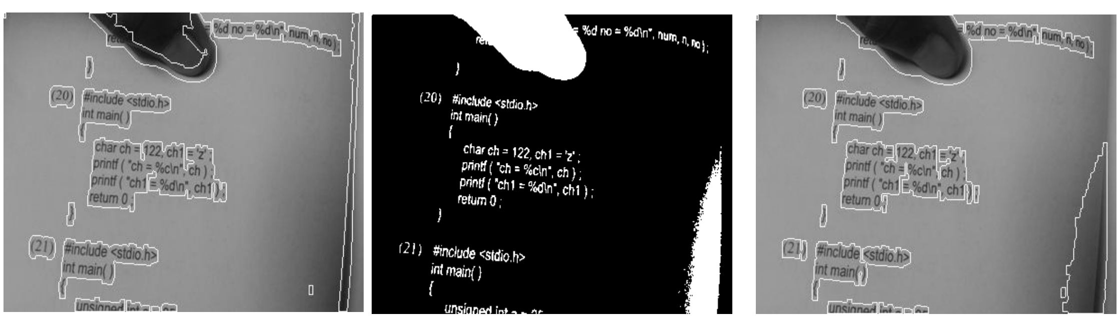
Fig (5.27)



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

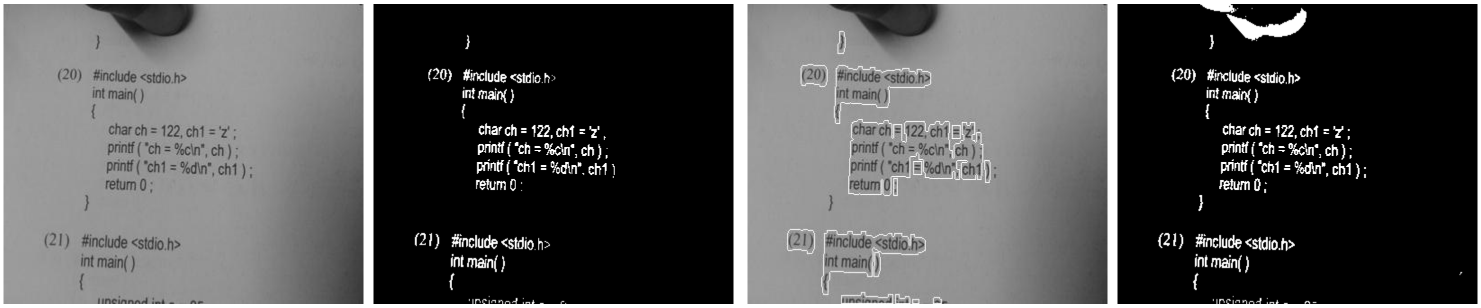


(e) (f) (g) (h)

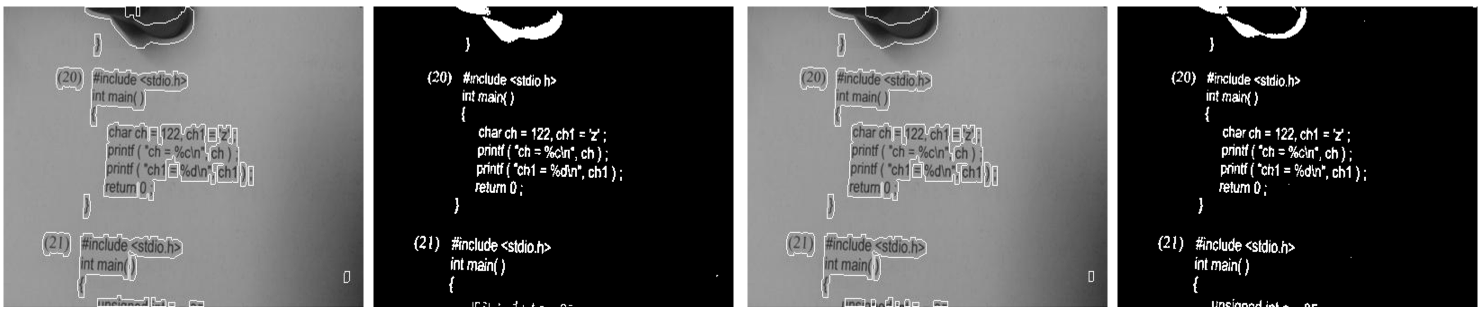


(i) (j) (k)

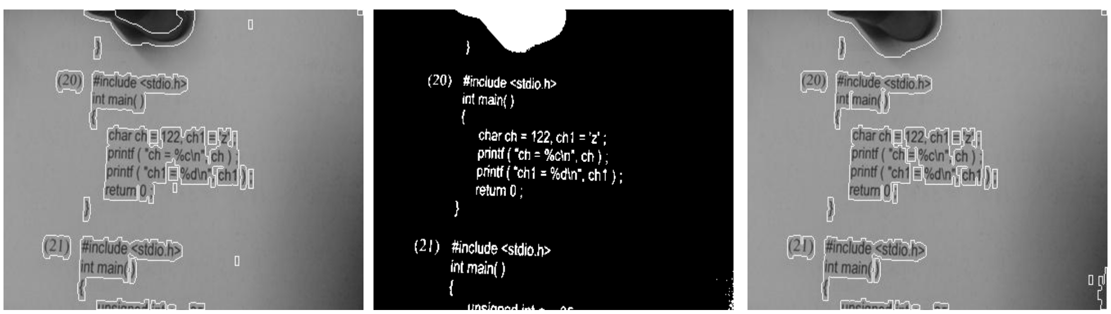
Fig (5.28)



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

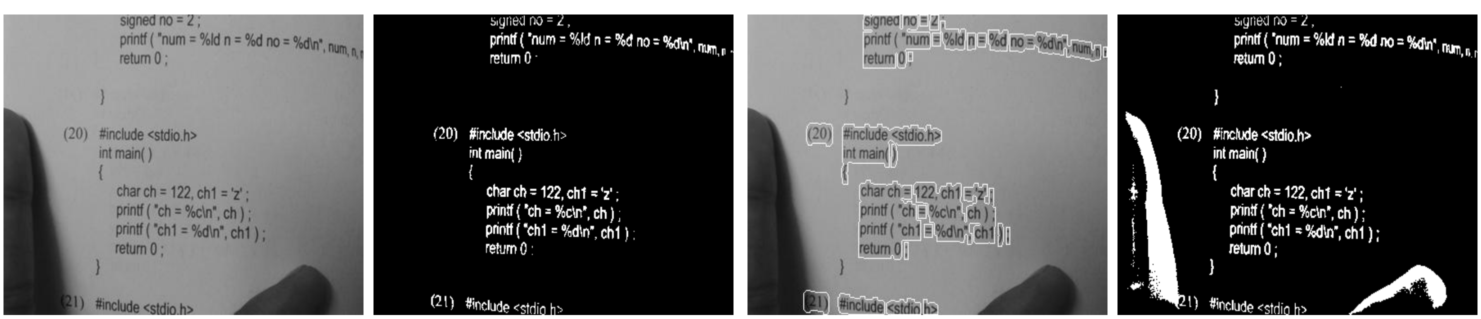


(e) (f) (g) (h)

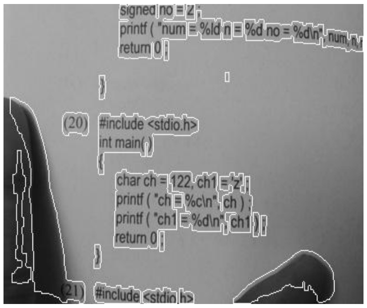


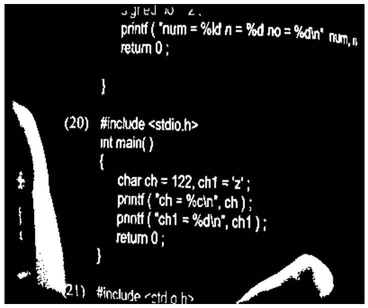
(i) (j) (k)

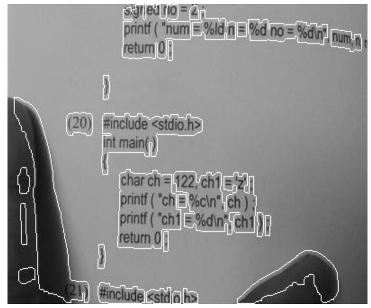
Fig (5.29)

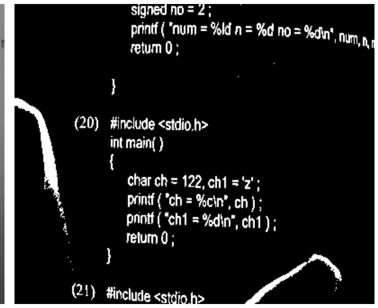


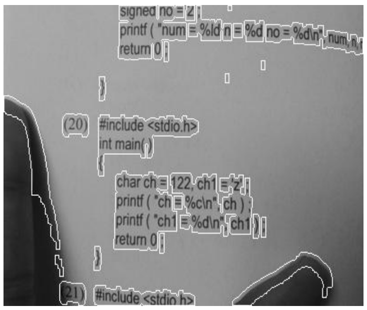
(a) (b) (c) (d)

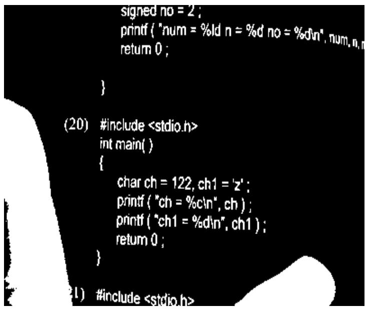

(e)


(f)


(g)


(h)


(i)


(j)

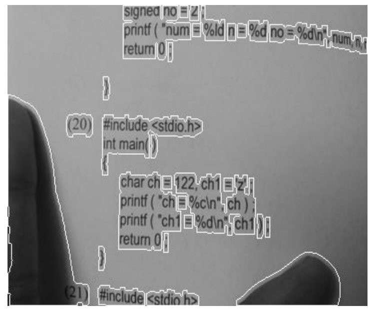
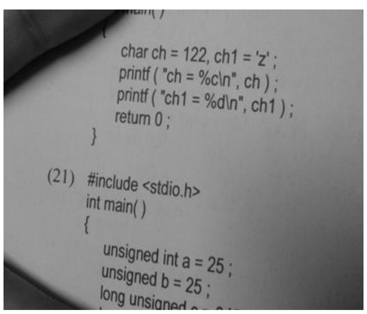
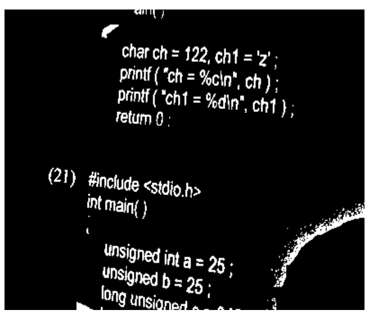
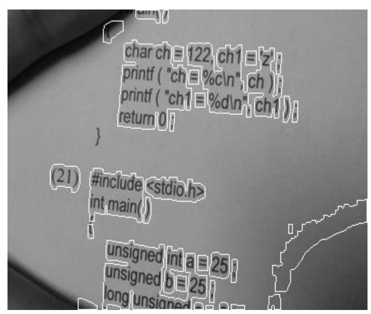
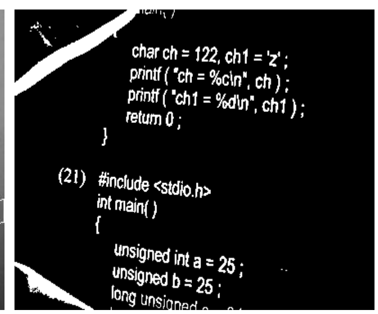

(k)

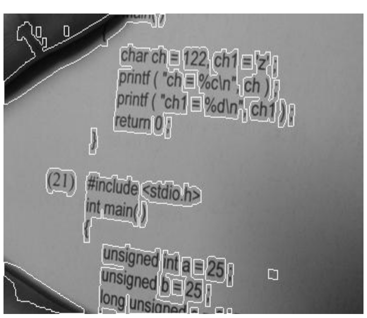
Fig (5.30)

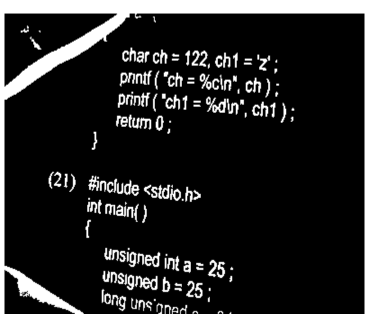

(a)

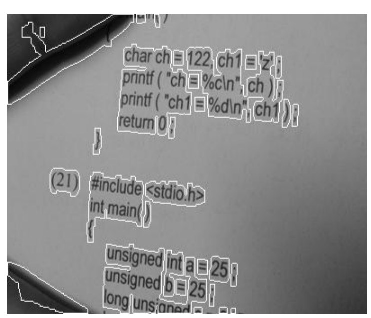

(b)

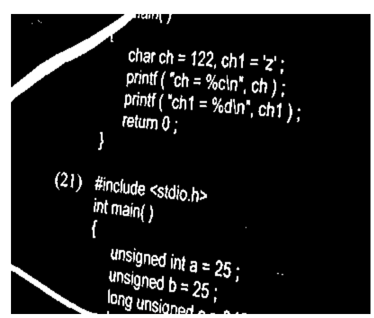

(c)

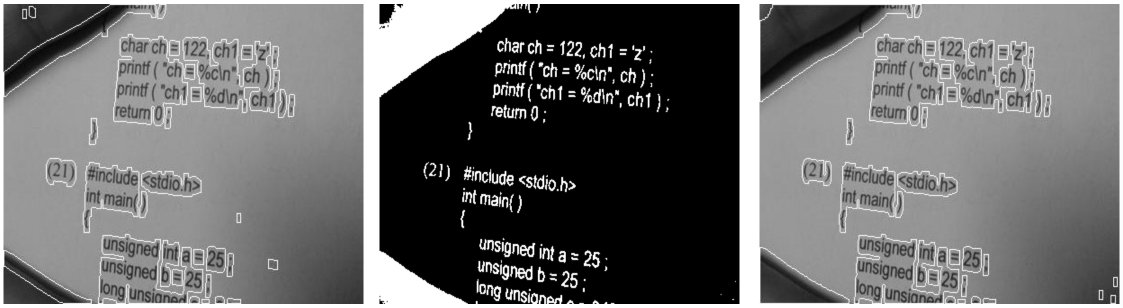

(d)


(e)


(f)

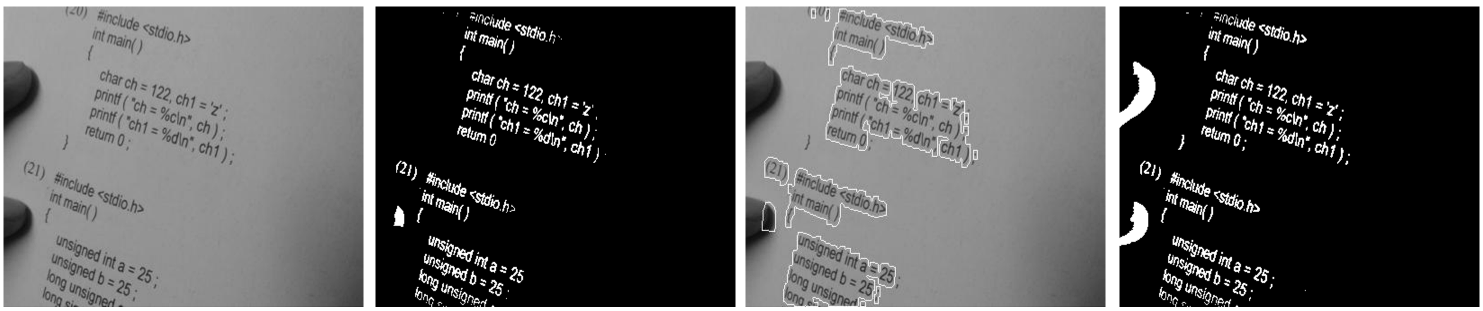

(g)


(h)

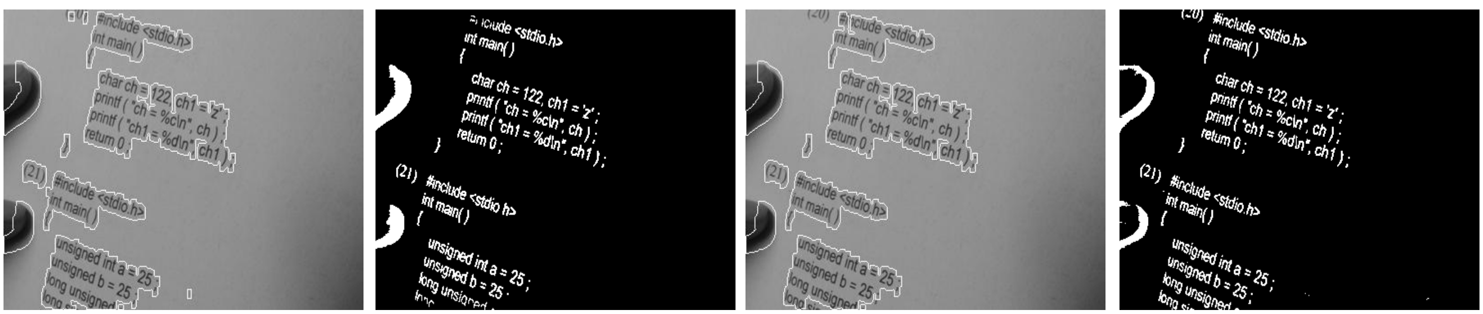


(i) (j) (k)

Fig (5.31)



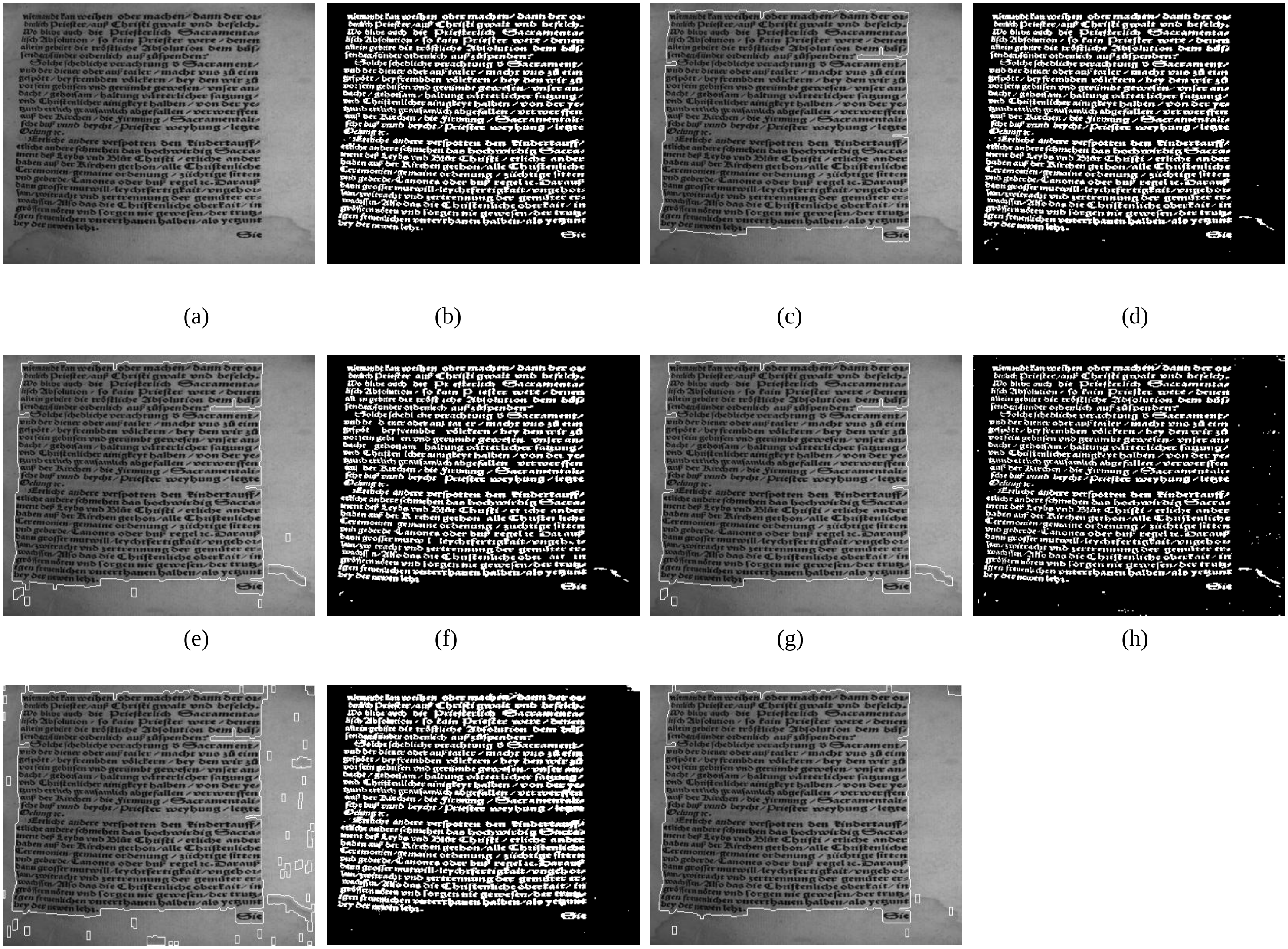
(a) (b) (c) (d)



(e) (f) (g) (h)

(i) (j) (k)

Fig (5.32)

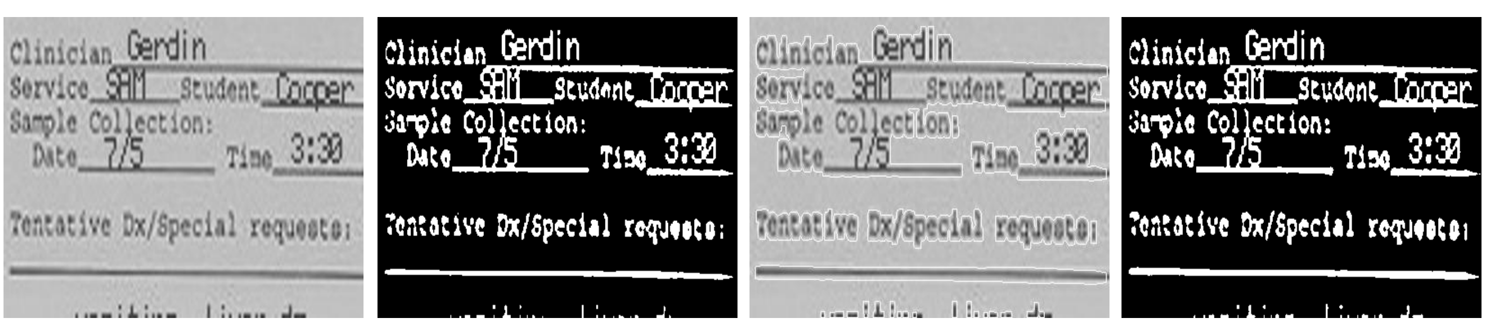


(a) (b) (c) (d)

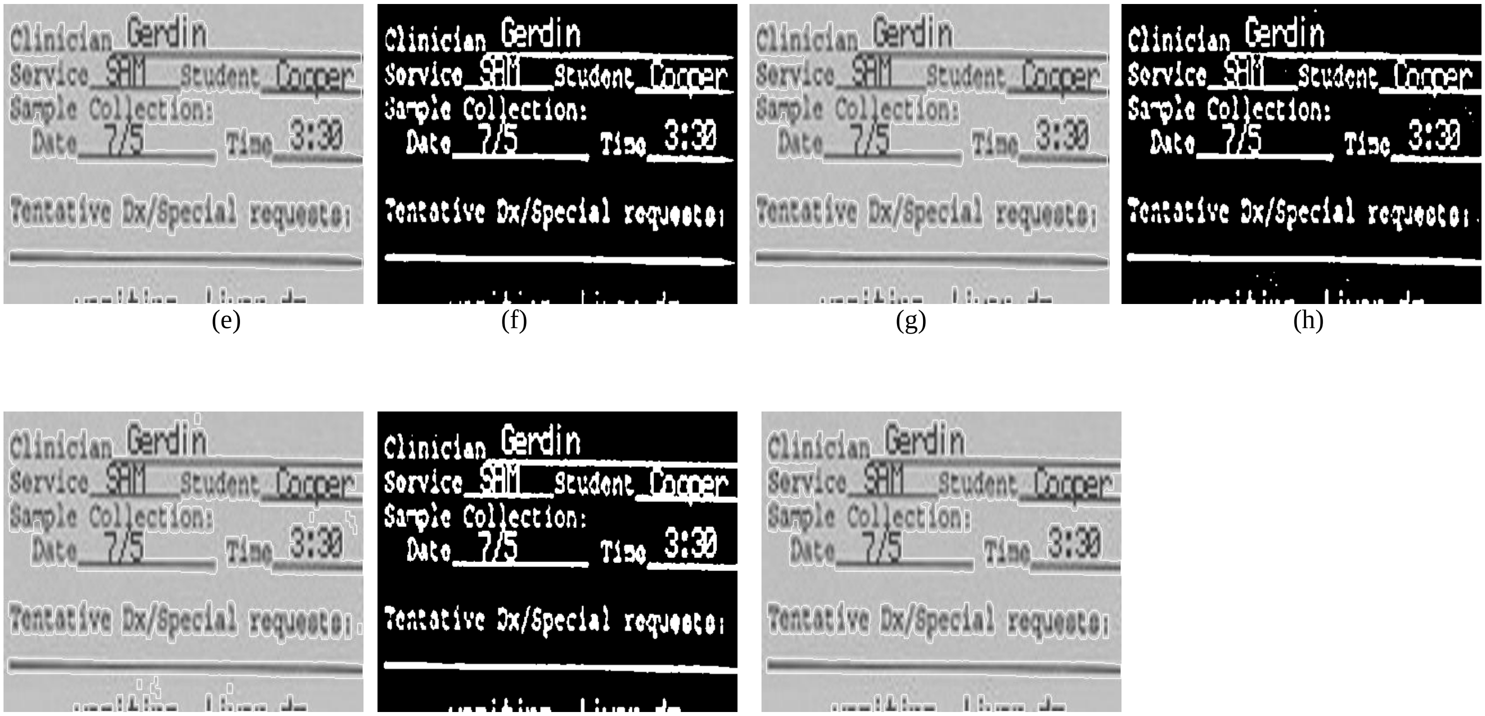
(e) (f) (g) (h)

(i) (j) (k)

Fig (5.33)



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



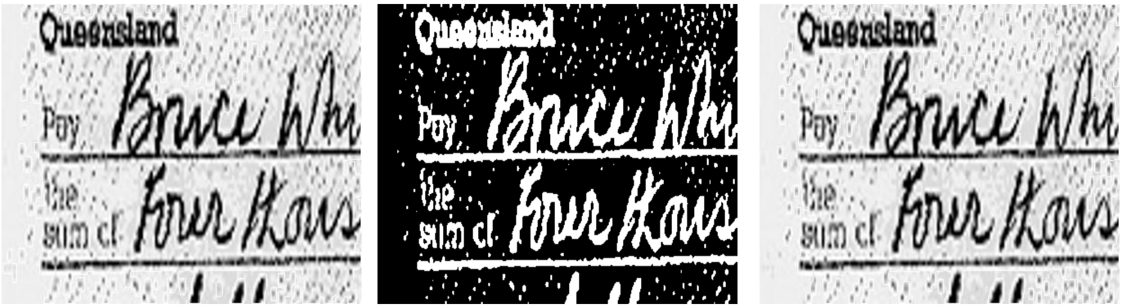
(e) (f) (g) (h)

(i) (j) (k)

Fig (5.34)

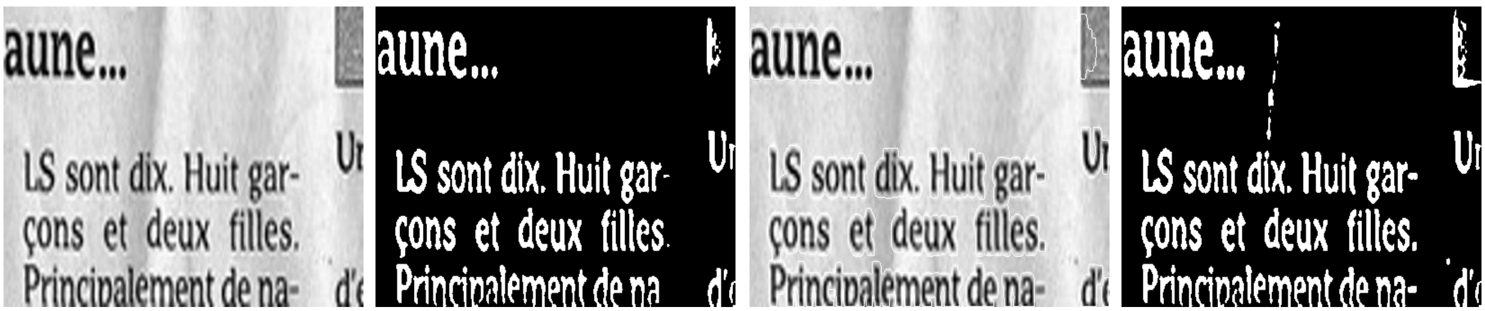
(a) (b) (c) (d)

(e) (f) (g) (h)

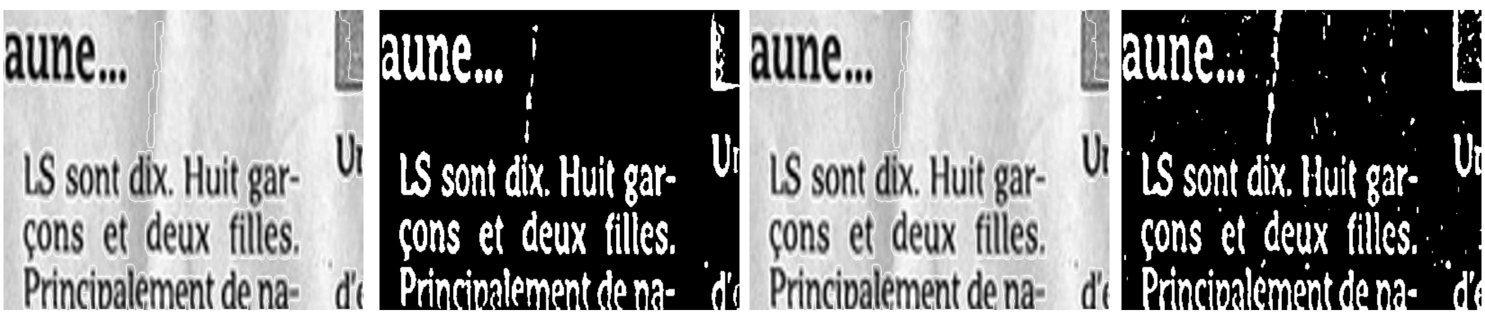


(i) (j) (k)

Fig (5.35)



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



(e) (f) (g) (h)

(i) (j) (k)

Fig (5.36)

In above figures the following parameters are used-

(a) - Original Image, (b) - Binary Image using Proposed Algorithm, (c) - Text Segment Image using Proposed Algorithm, (d) - Binary Image using Yanowitz’s Method, (e) - Text Segment Image using Yanowitz’s Method, (f) - Binary Image using Post Yanowitz’s Method, (g) - Text Segment Image using Post Yanowitz’s Method, (h) - Binary Image using Niblack’s Method, (i) - Text Segment Image using Niblack’s Method, (j) - Binary Image using Sauvola’s Method, (k) - Text Segment Image using Sauvola’s Method.

There are 36 mobile images that describe above with experimental results are divided into different classes based on irrelevant information-

Class1: Document images with finger are reviewed in figure 5.3, 5.13, 5.20, 5.21, 5.27, 5.28, 5.29, 5.30, 5.31, 5.32.

Class2: Document images with pen/pencil are reviewed in figure 5.2, 5.7, 5.8, 5.12, 5.22, 5.23, 5.24, 5.25.

Class3: Document images with the mobile / watch are reviewed in figure 5.1, 5.4, 5.5, 5.6, 5.10, 5.11, 5.16, 5.19.

Class4: Extra document images are shown in figure 5.9, 5.14, 5.15, 5.17, 5.18, 5.26, 5.33, 5.34, 5.35, 5.36.

The main aim of propose method is to remove irrelevant information and binarized only text area of an image. In case of Mobile images, out of 36 results 27 results (i.e. Fig 5.1, 5.3, 5.4, 5.6, 5.8, 5.9, 5.10, 5.12, 5.13, 5.15, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.32, 5.33, 5.36) is very good using proposed algorithm. The rest of them are also good using our method as compared to other well known methods. The rank number is given to every method based on visualization that are shown in below table-

Table 5.1 Rank of Methods

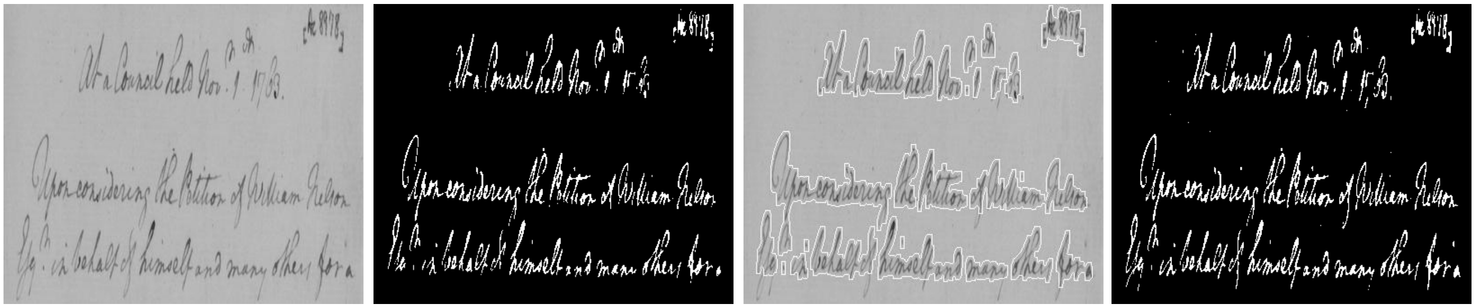
Image	Proposed Method	Method (1)	Method (2)	Method (3)	Method (4)
5.1-5.4	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}
5.5	2^{nd}	4^{th}	3^{rd}	1^{st}	5^{th}
5.6-5.7	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}
5.8	1^{st}	3^{rd}	2^{nd}	4^{th}	5^{th}
5.9-5.11	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}
5.12-5.13	1^{st}	3^{rd}	2^{nd}	4^{th}	5^{th}
5.14-5.15	Same	Same	Same	Same	Same
5.16	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}
5.17	Same	Same	Same	Same	Same
5.18	1^{st}	3^{rd}	2^{nd}	4^{th}	5^{th}
5.19-5.25	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}
5.26	1^{st}	4^{th}	2^{nd}	5^{th}	3^{rd}
5.27-5.32	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}
5.33	1^{st}	4^{th}	3^{rd}	5^{th}	2^{nd}
5.34-5.35	Same	Same	Same	Same	Same
5.36	1^{st}	4^{th}	3^{rd}	2^{nd}	5^{th}

Method 1 - Yanowitz, Method 2 - Post Yanowitz, Mehod 3 - Niblack, Method 4 – Sauvola.

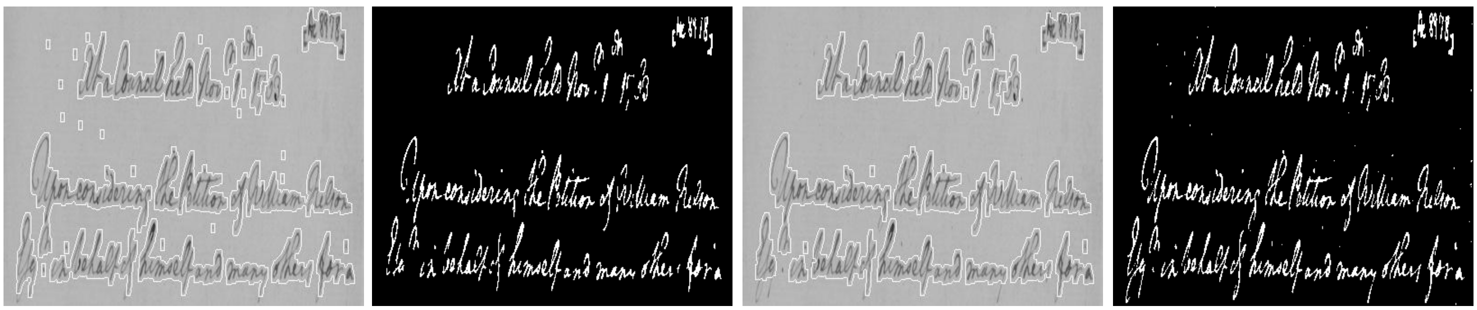
First best method is Proposed Method, Second best method is Niblack's Method, Third best method is Post Yanowitz's Method, fourth best method is Yanowitz's Method and fifth best method is Sauvola's Method.

5.2.2 DIBCO Dataset Results

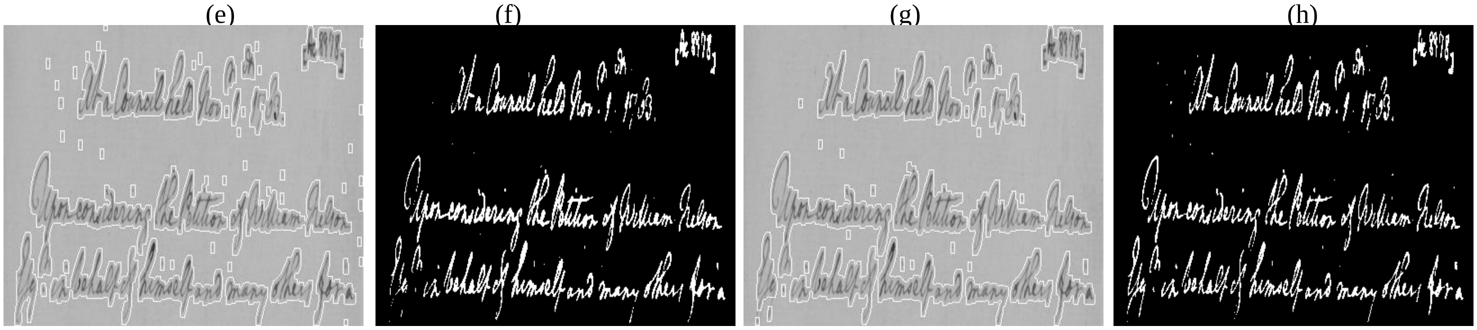
5.2.2.1 DIBCO-2009 Dataset Results



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

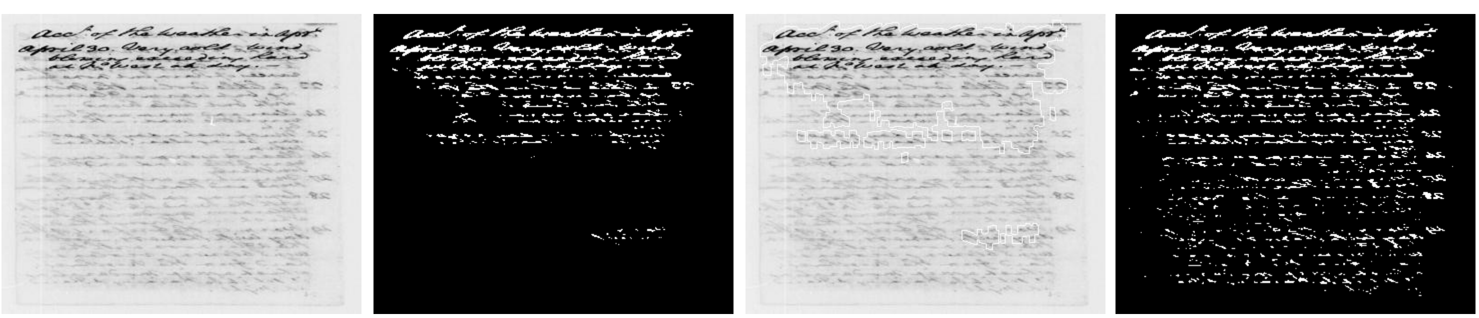


(e) (f) (g) (h)



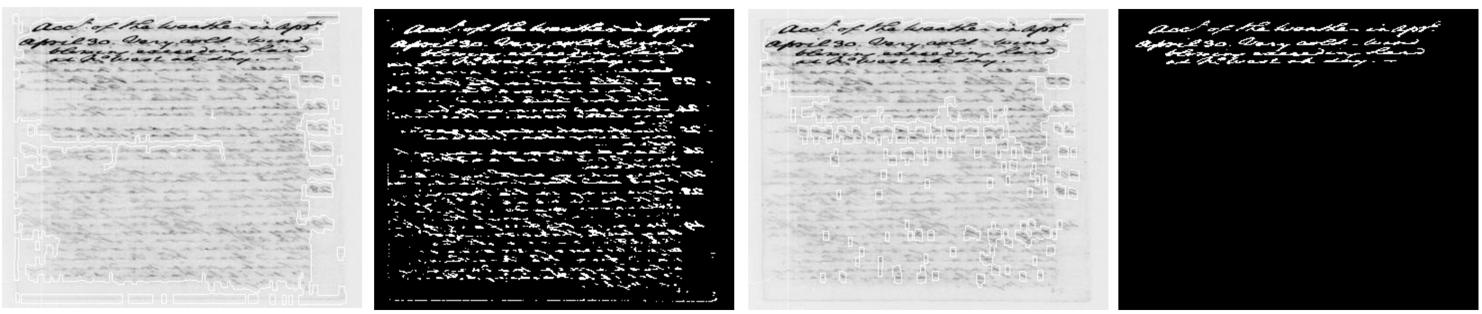
(i) (j) (k) (l)

Fig (5.37)



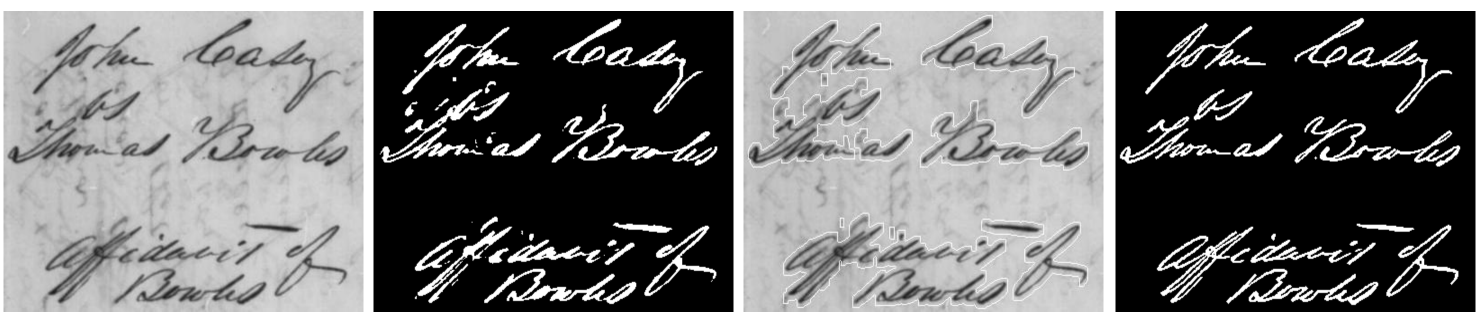
(a) (b) (c) (d)

(e) (f) (g) (h)

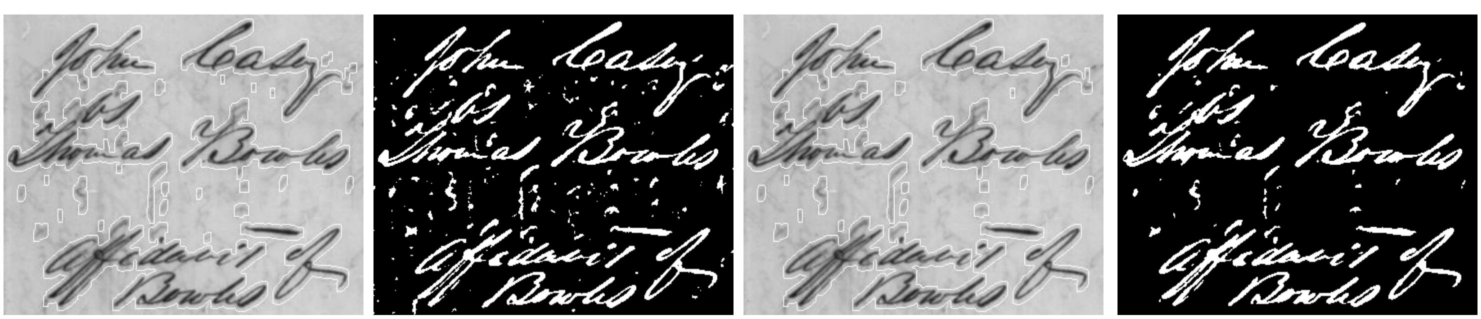


(i) (j) (k) (l)

Fig (5.38)



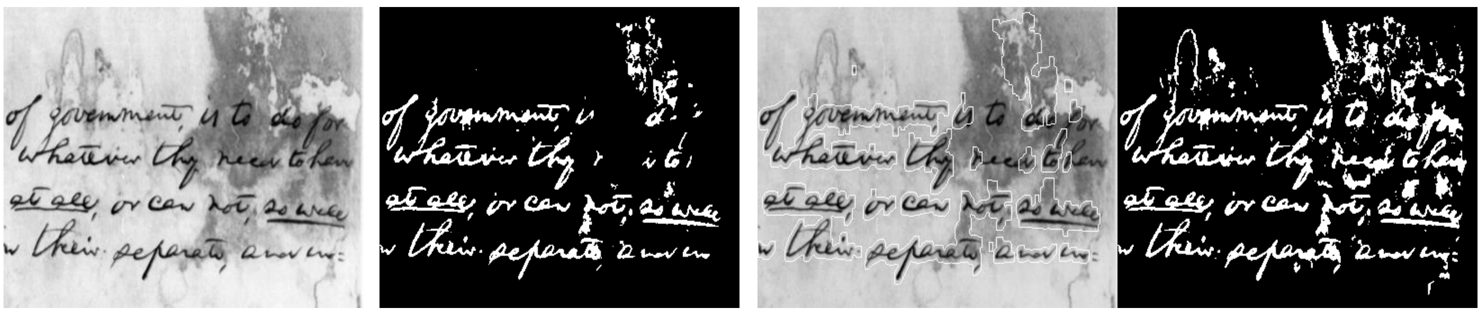
(a) (b) (c) (d)



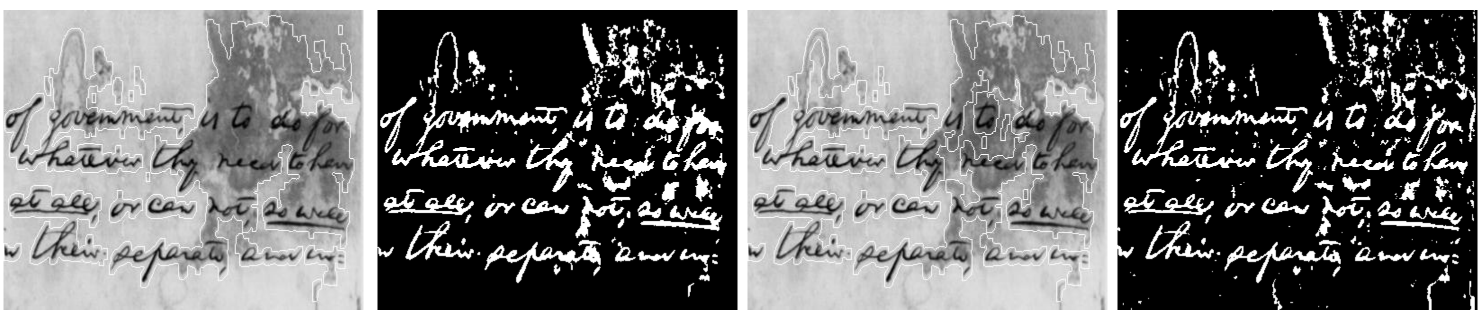
(e) (f) (g) (h)

(i) (j) (k) (l)

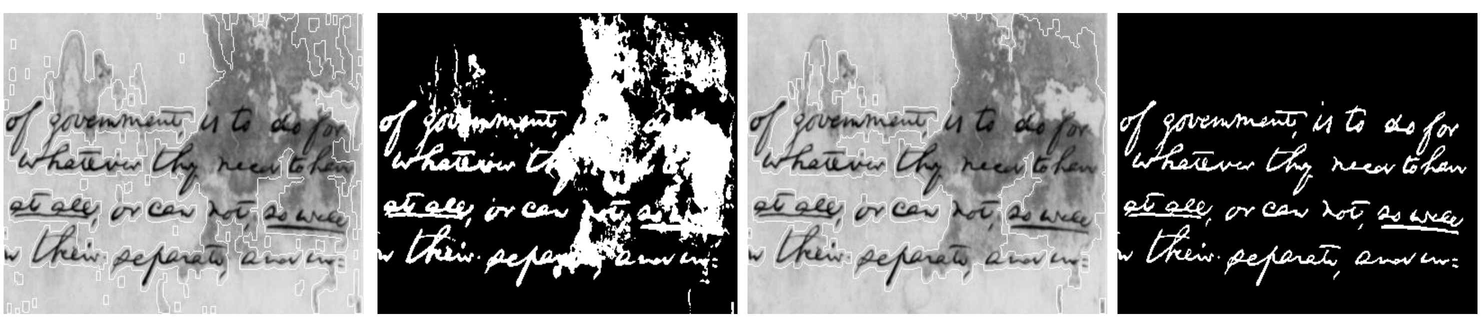
Fig (5.39)



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

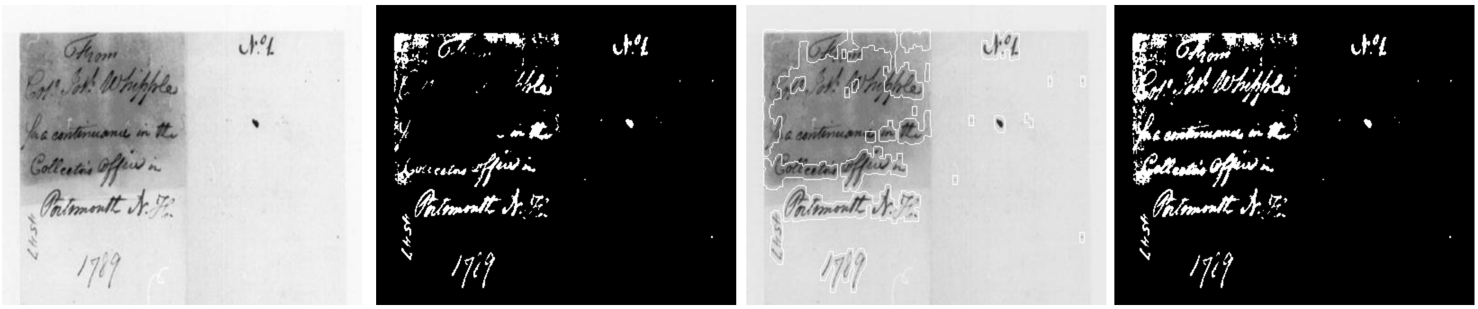


(e) (f) (g) (h)

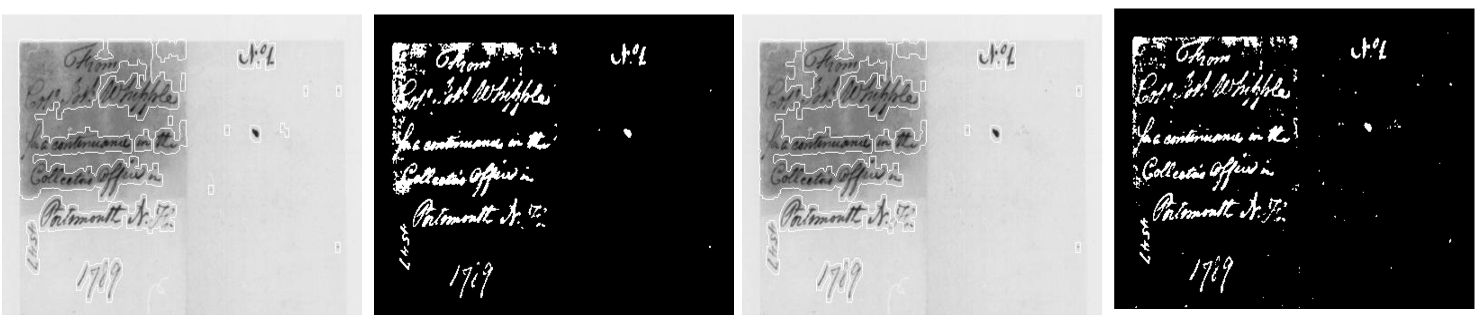


(i) (j) (k) (l)

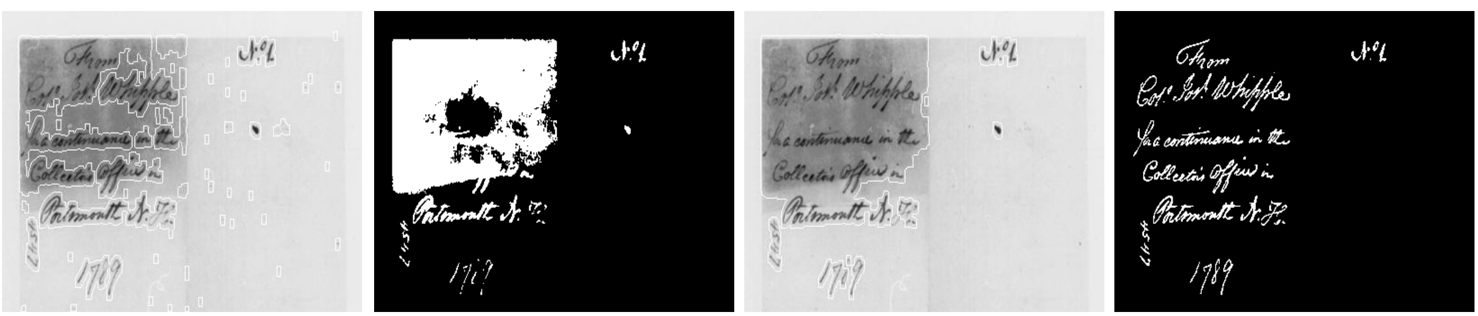
Fig (5.40)



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

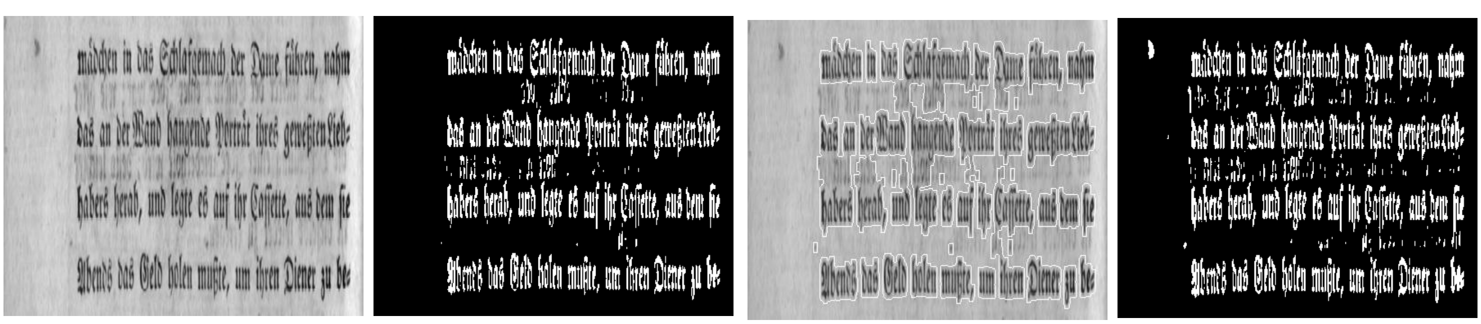


(e) (f) (g) (h)

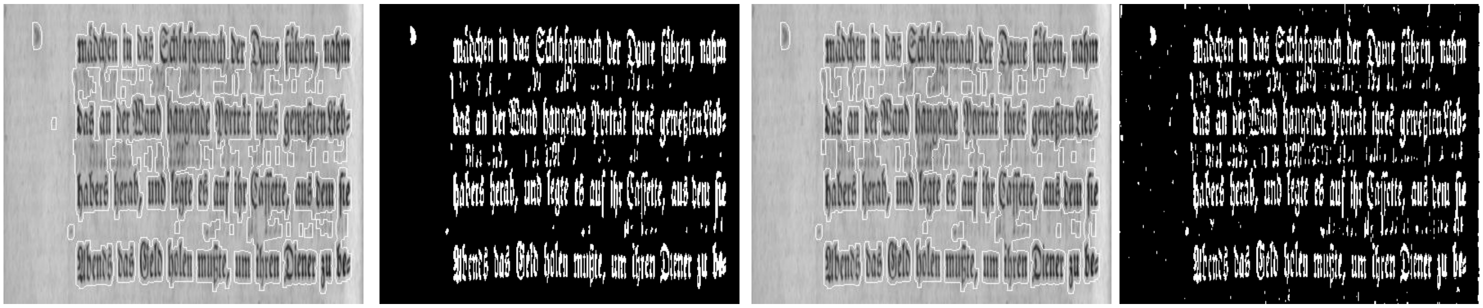


(i) (j) (k) (l)

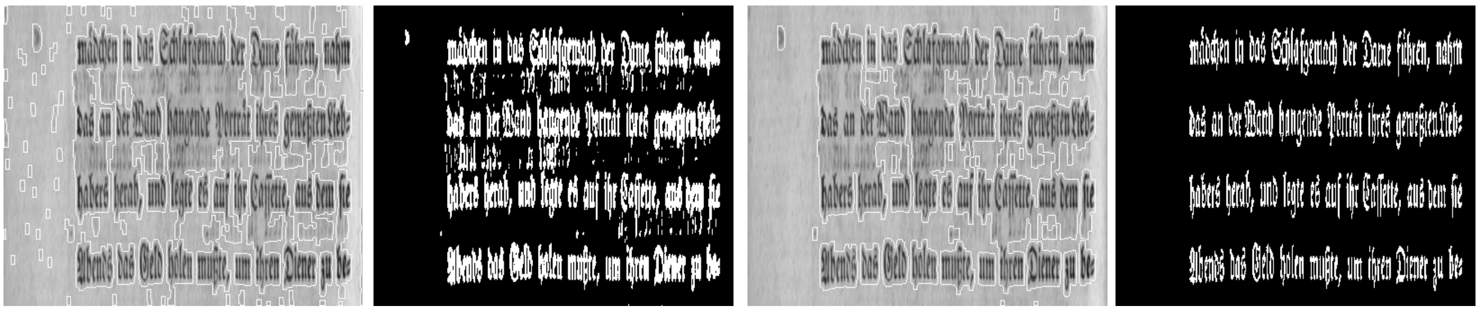
Fig (5.41)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.42)

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

(e) (f) (g) (h)

(i) (j) (k) (l)

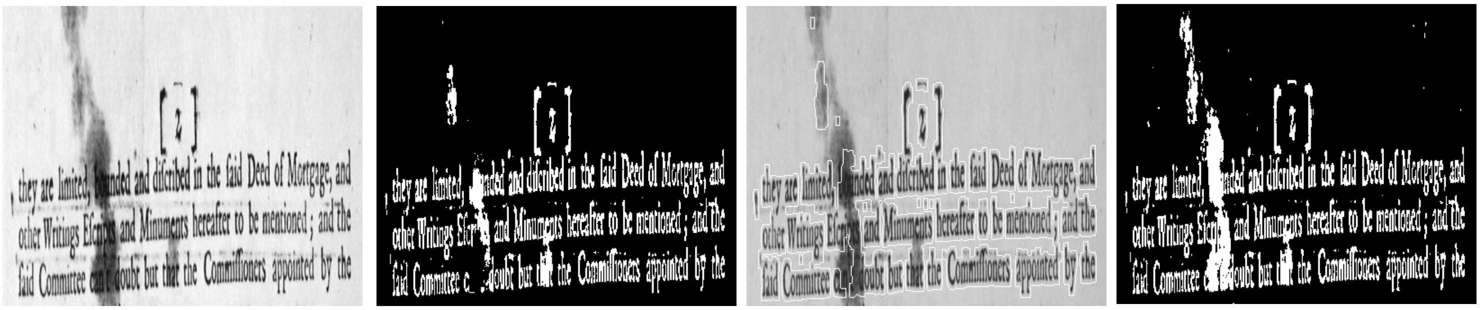
Fig (5.43)

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

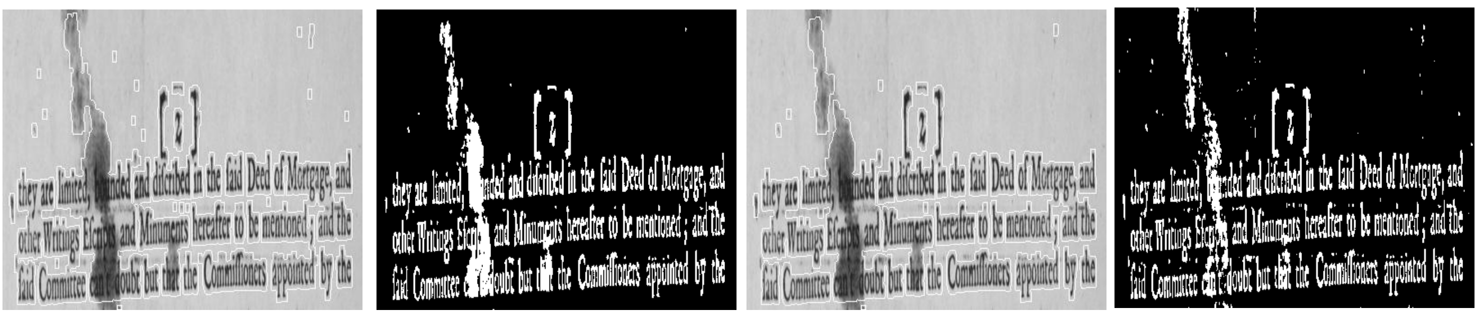
(e) (f) (g) (h)

(i) (j) (k) (l)

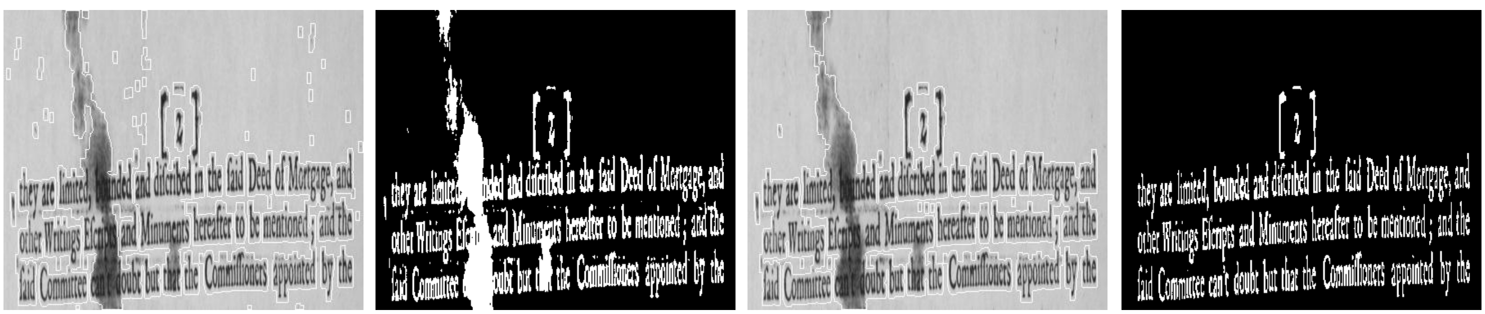
Fig (5.44)



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

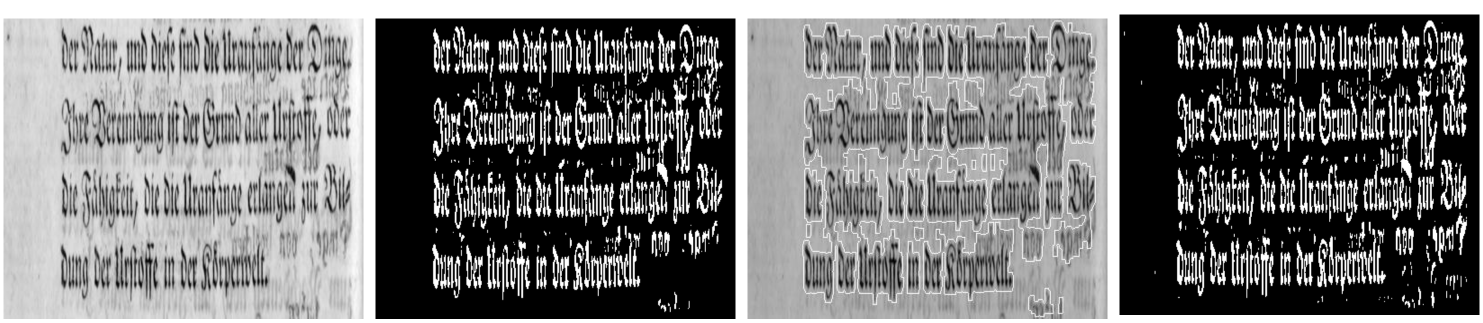


(e) (f) (g) (h)

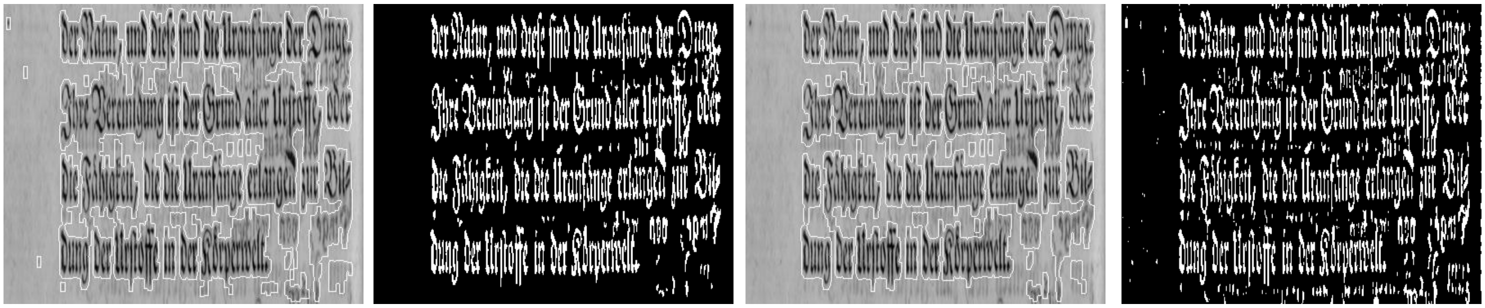


(i) (j) (k) (l)

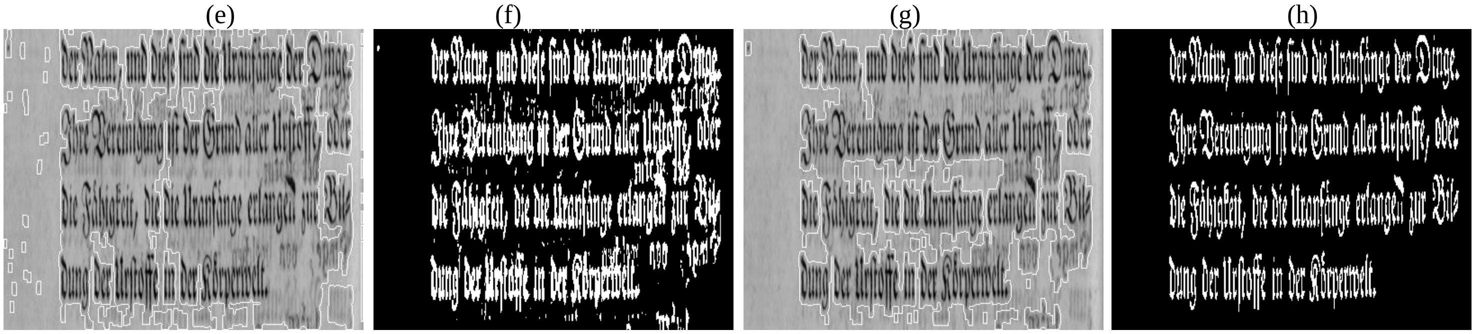
Fig (5.45)



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



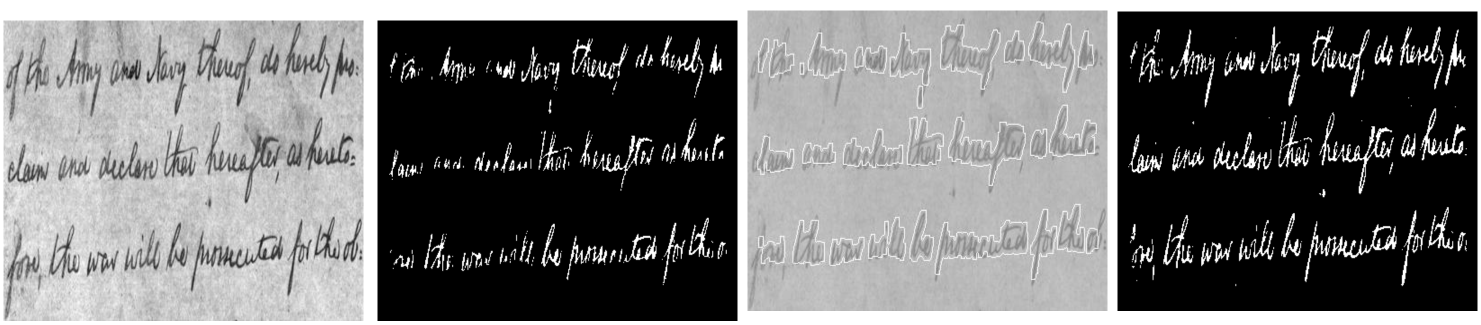
(e) (f) (g) (h)



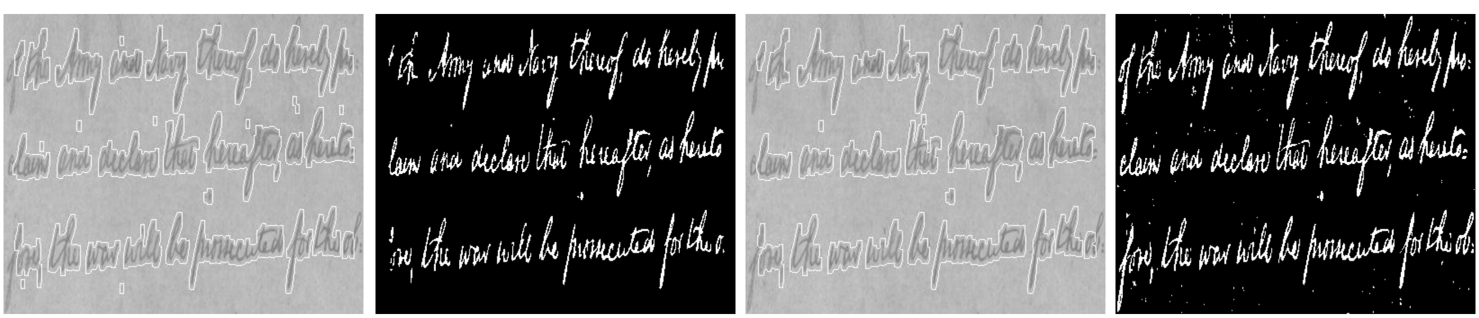
(i) (j) (k) (l)

Fig (4.46)

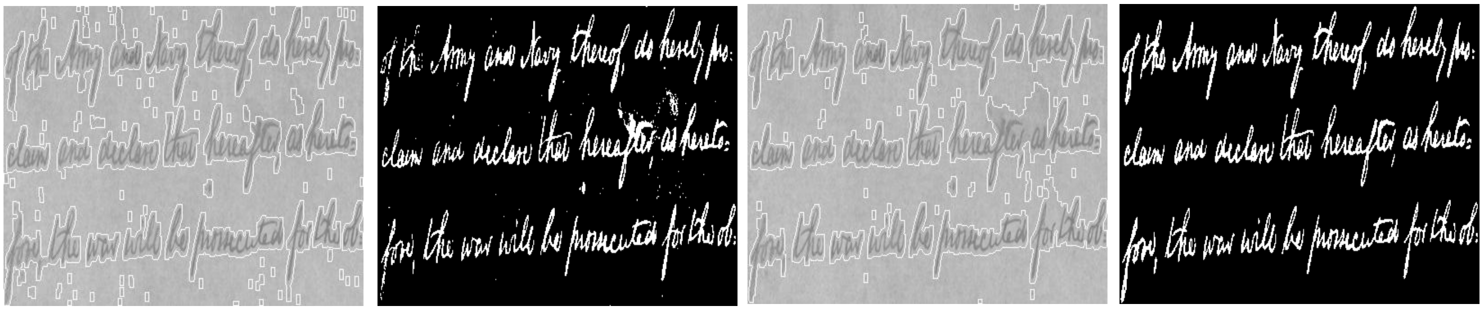
5.2.2.2 DIBCO-2010 Dataset Results



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

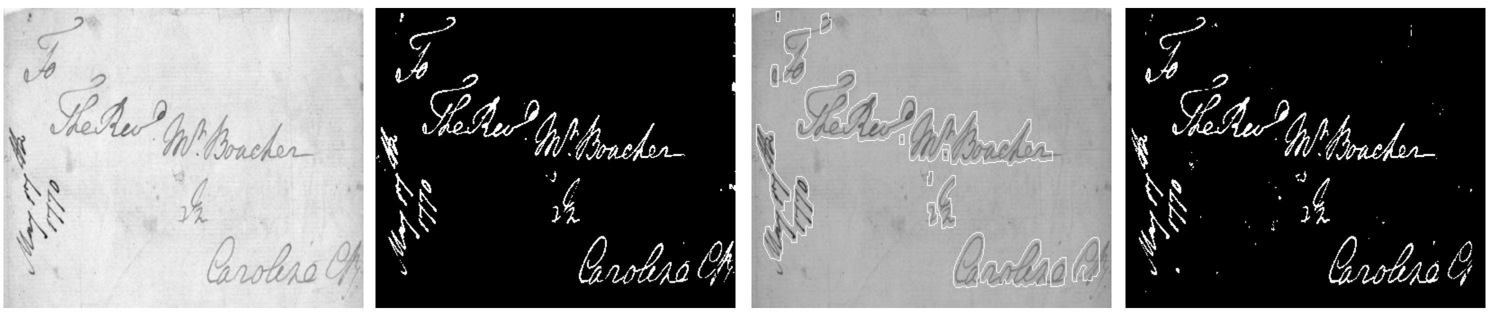


(e) (f) (g) (h)



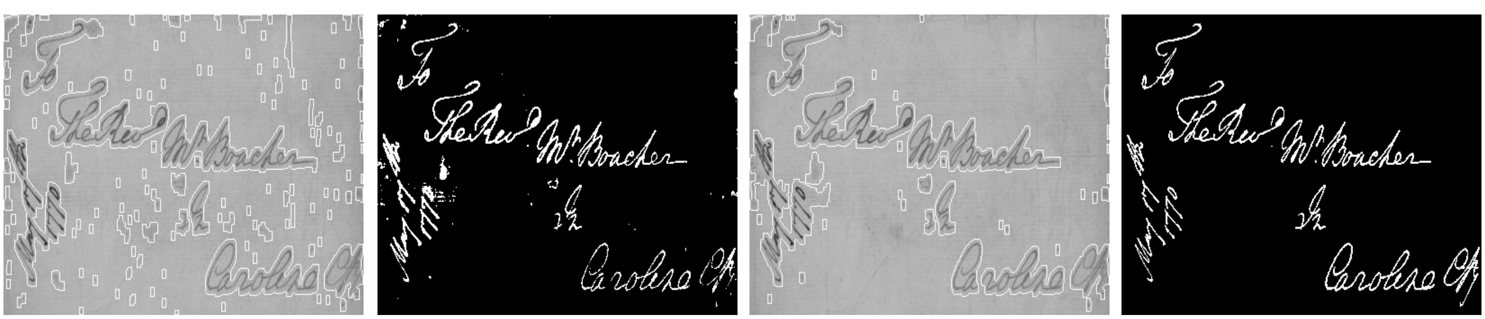
(i) (j) (k) (l)

Fig (5.47)



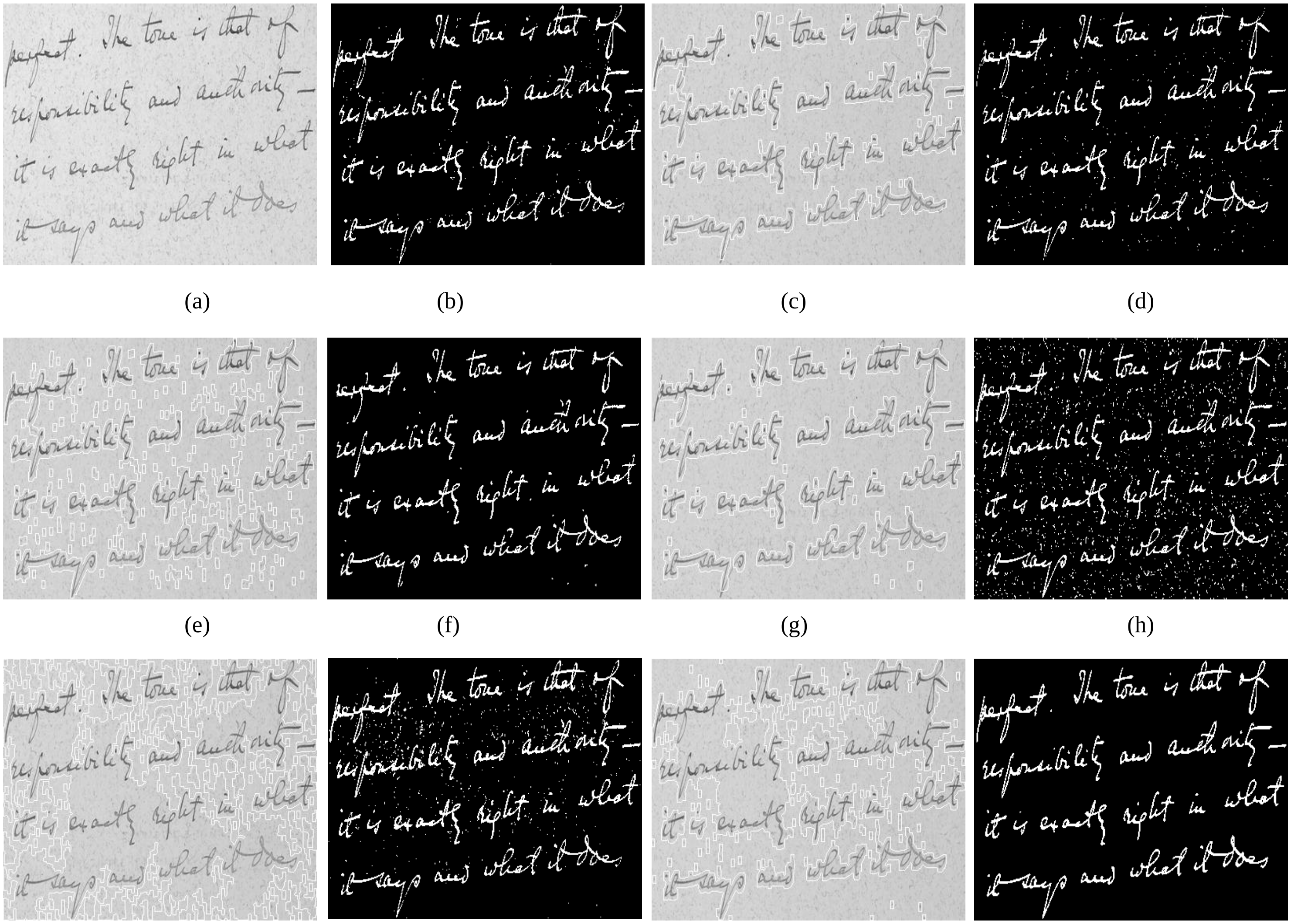
(a) (b) (c) (d)

(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.48)

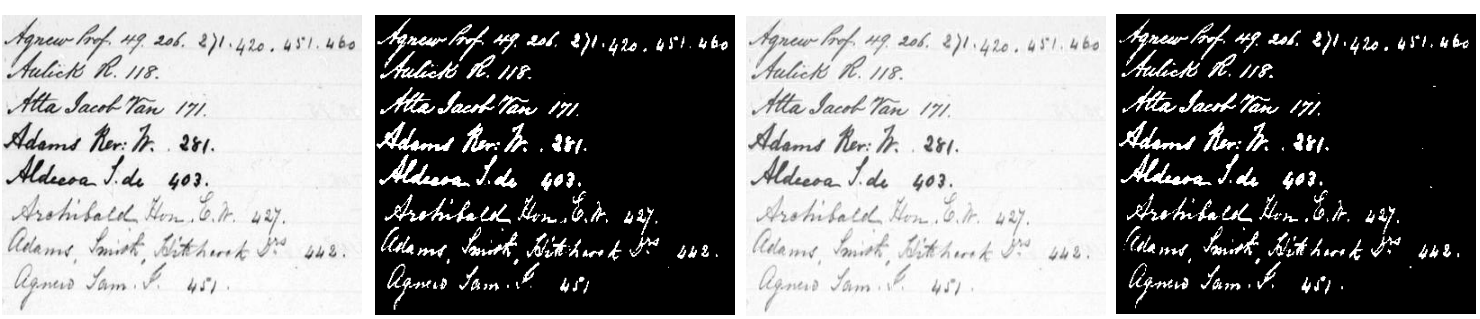


(a) (b) (c) (d)

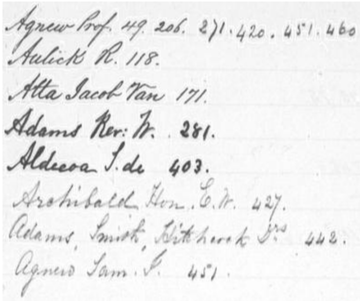
(e) (f) (g) (h)

(i) (j) (k) (l)

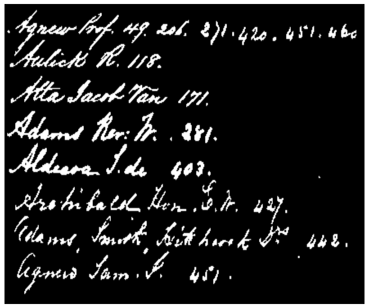
Fig (5.49)



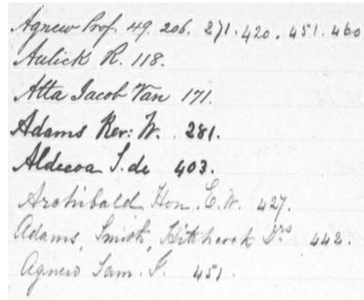
(a) (b) (c) (d)



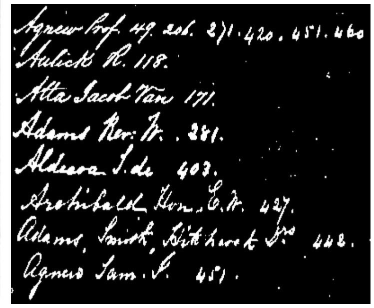
(e)



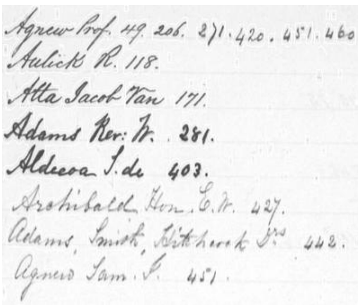
(f)



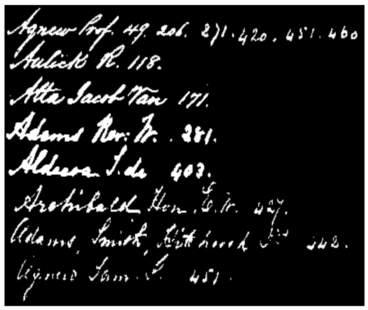
(g)



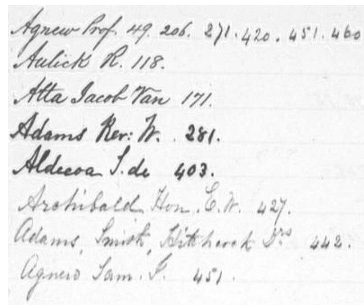
(h)



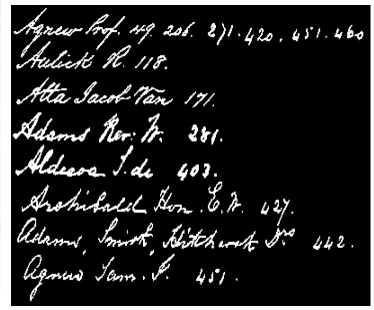
(i)



(j)

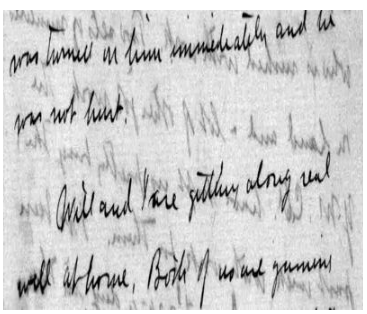


(k)

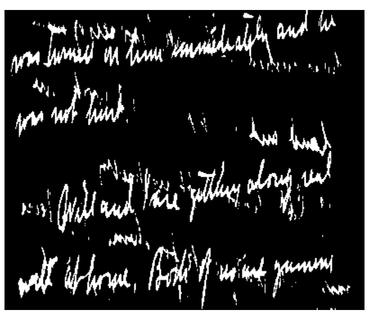


(l)

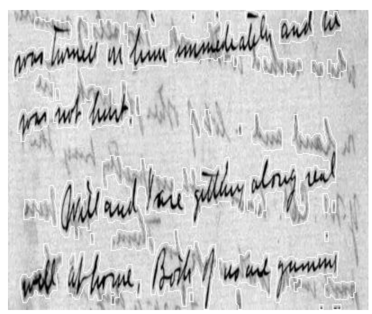
Fig (5.50)



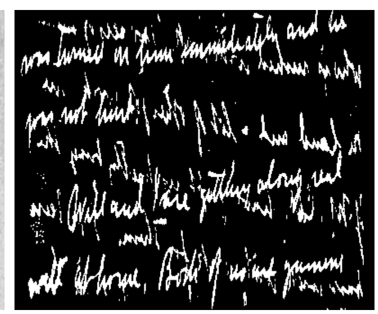
(a)



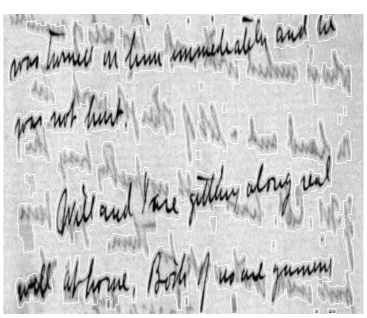
(b)



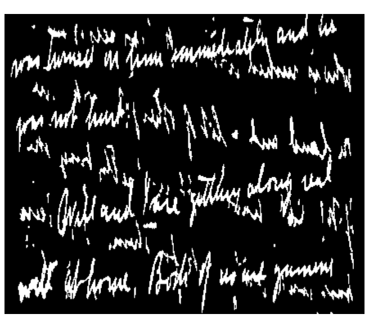
(c)



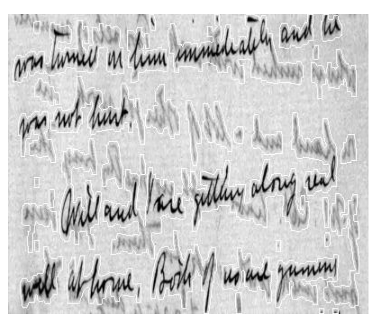
(d)



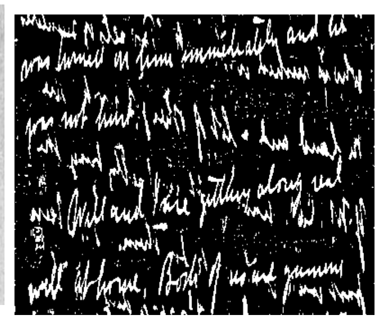
(e)



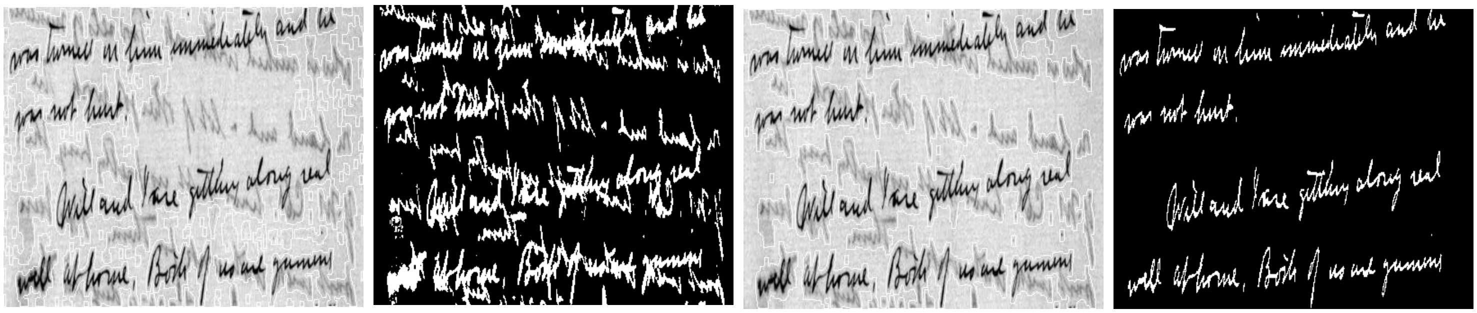
(f)



(g)

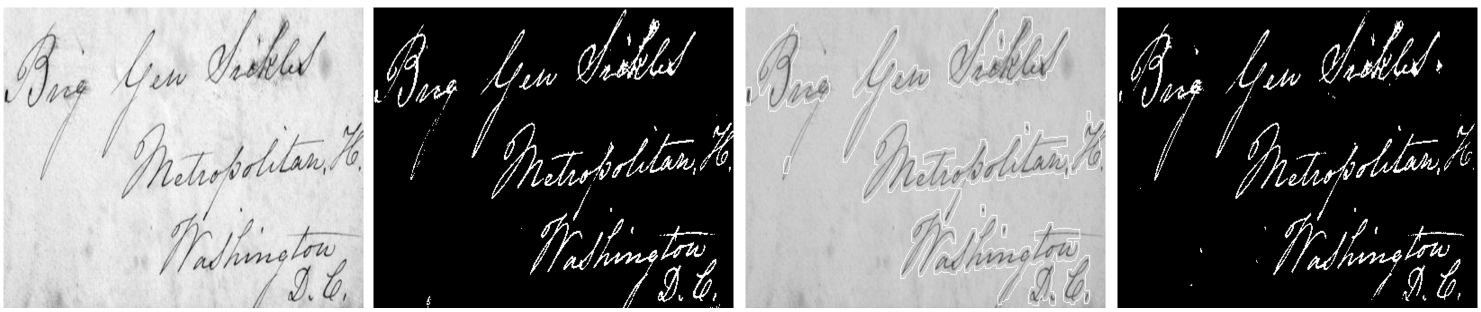


(h)

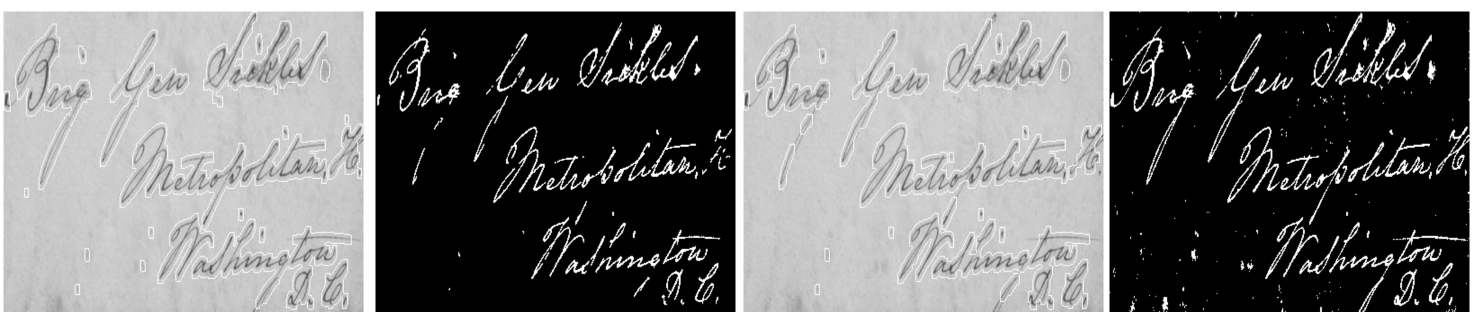


(i) (j) (k) (l)

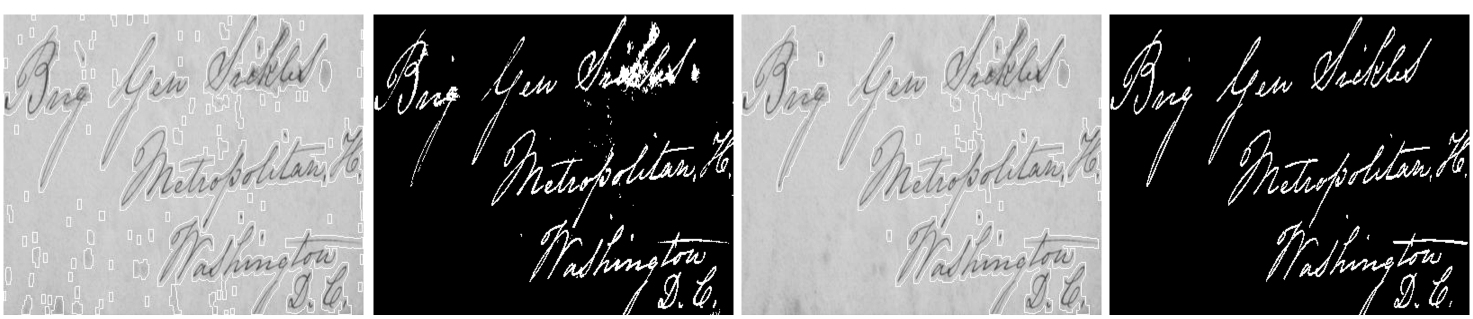
Fig (5.51)



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

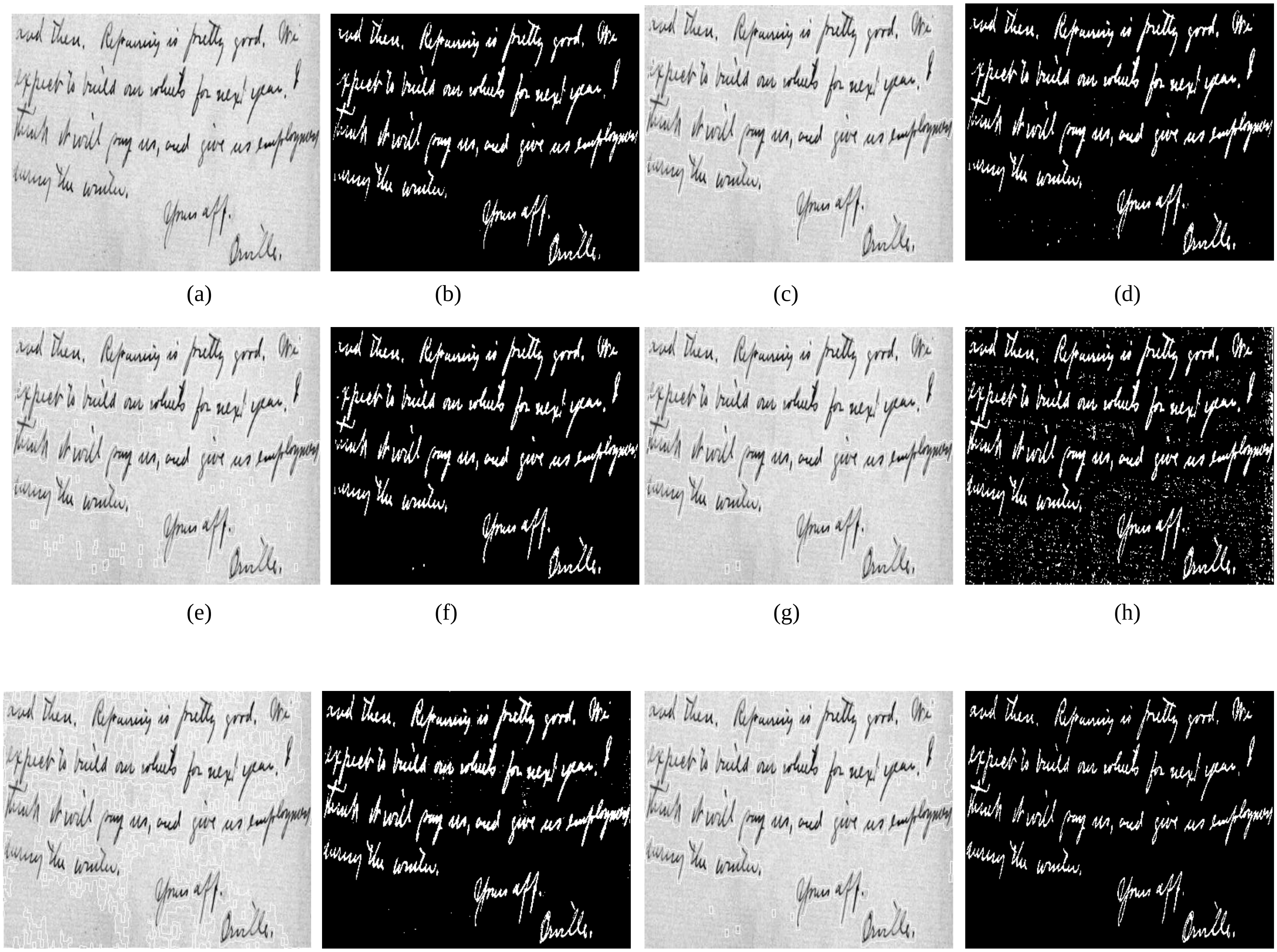


(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.52)

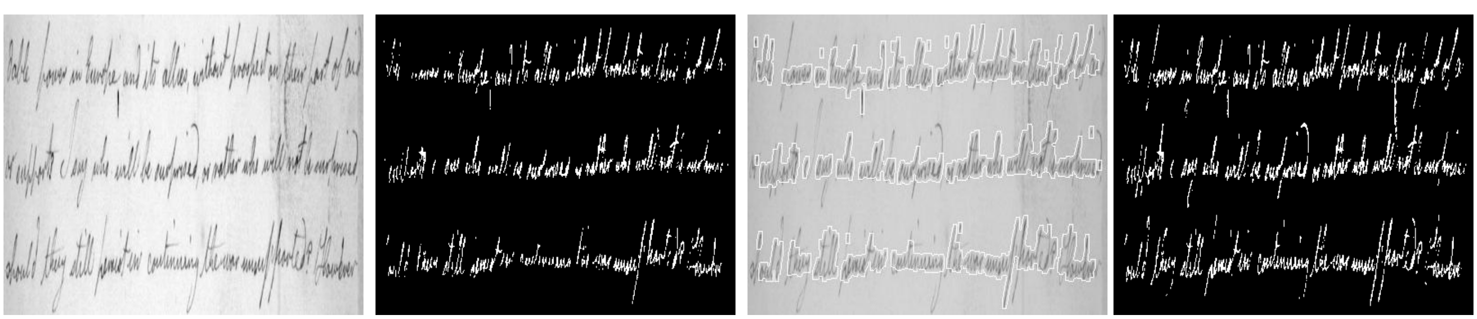


(a) (b) (c) (d)

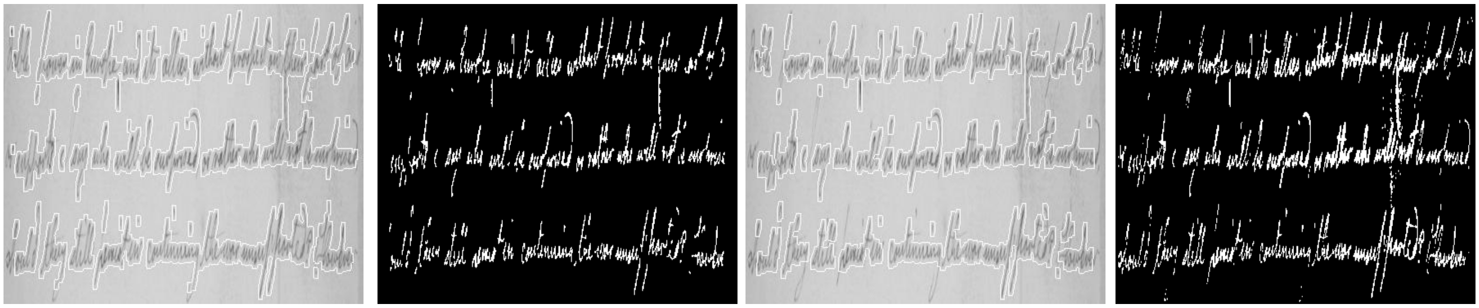
(e) (f) (g) (h)

(i) (j) (k) (l)

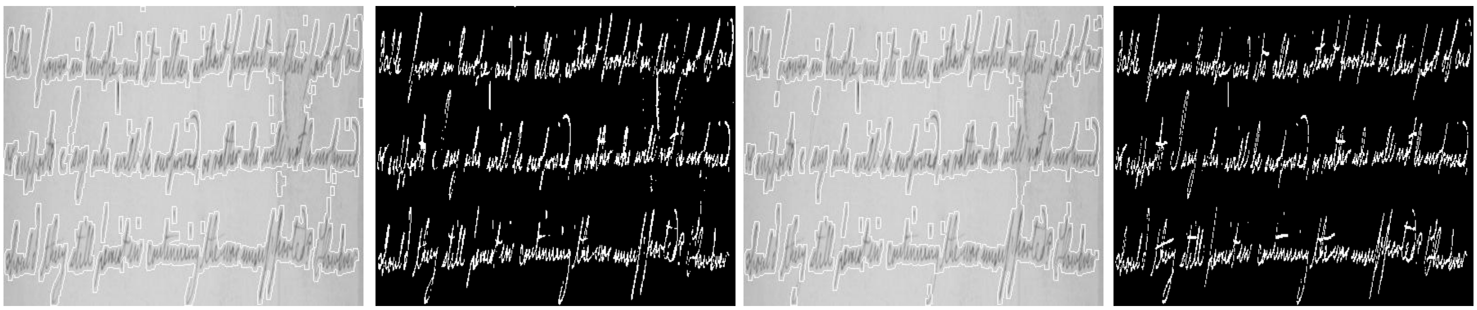
Fig (5.53)



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

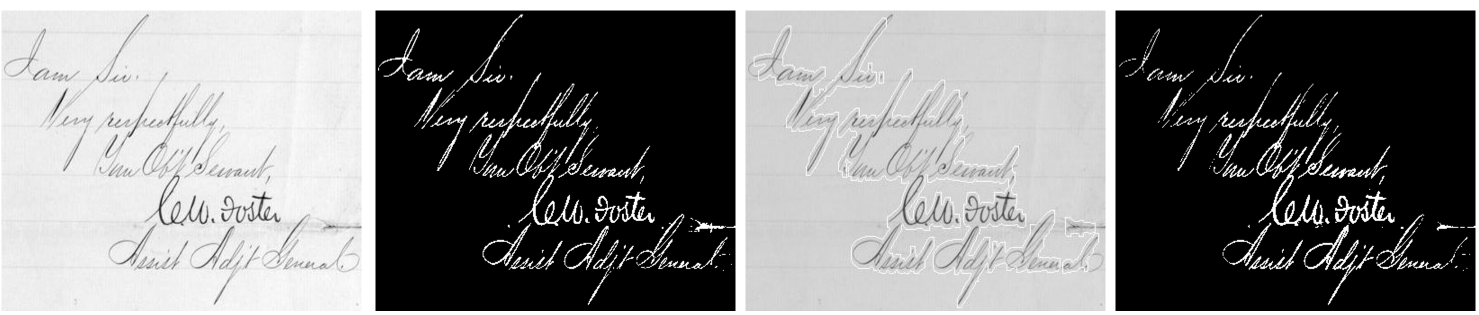


(e) (f) (g) (h)

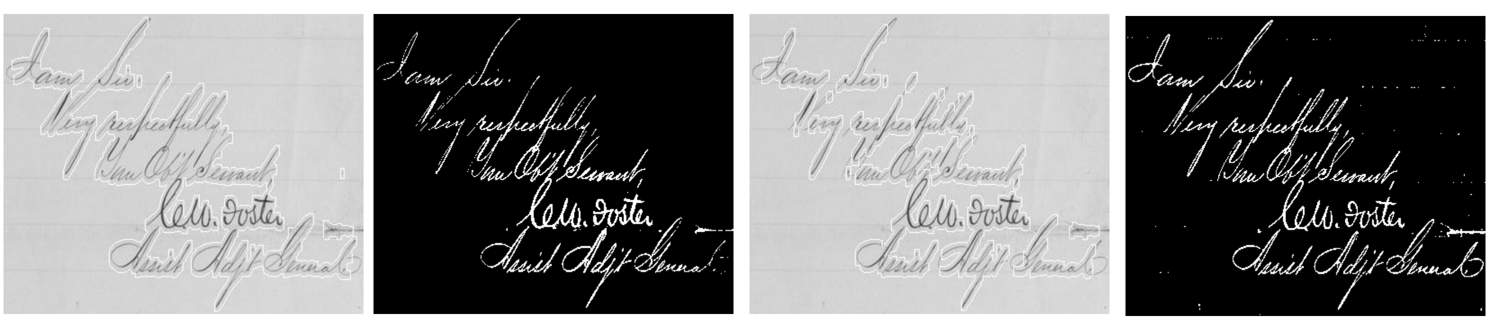


(i) (j) (k) (l)

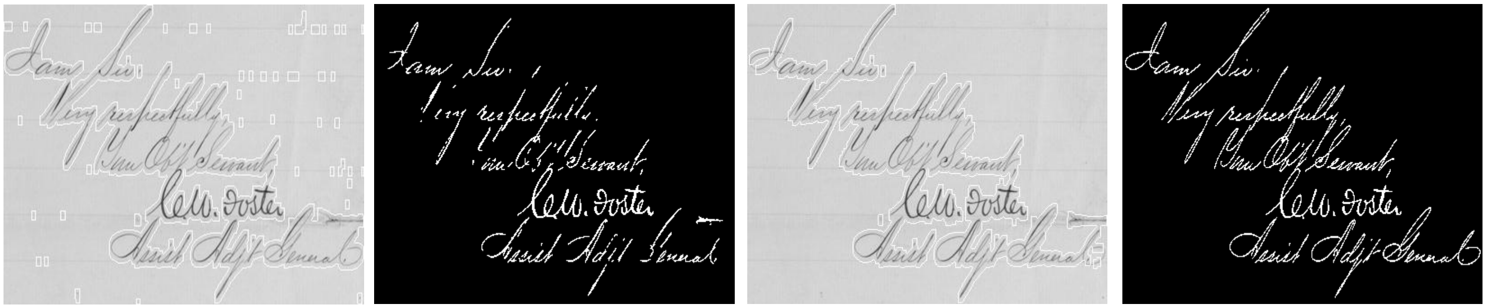
Fig (5.54)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.55)

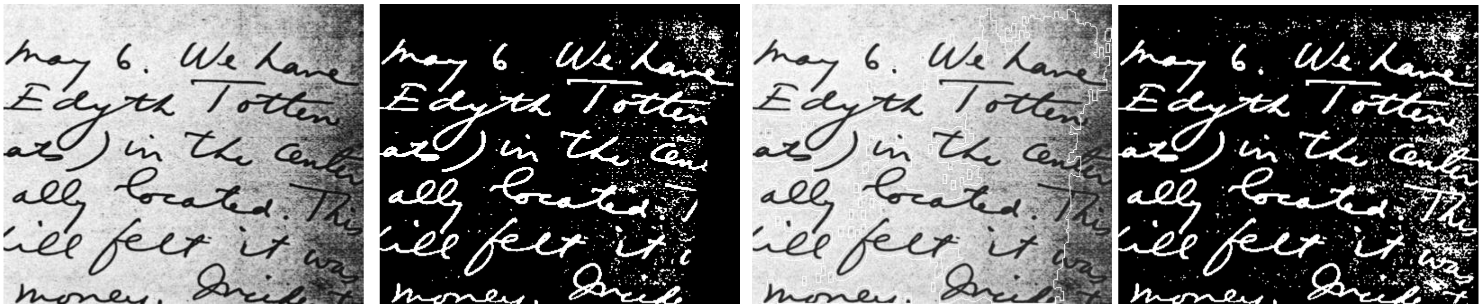
(a) (b) (c) (d)

(e) (f) (g) (h)

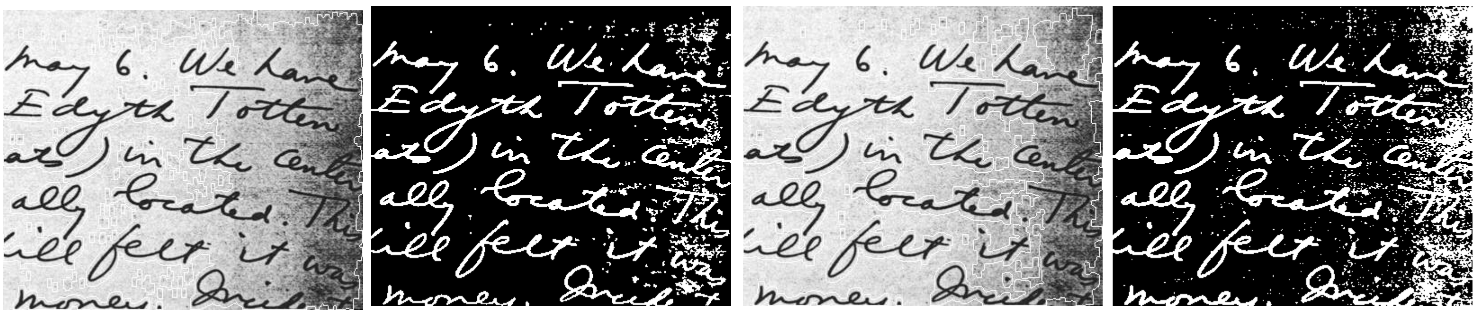
(i) (j) (k) (l)

Fig (5.56)

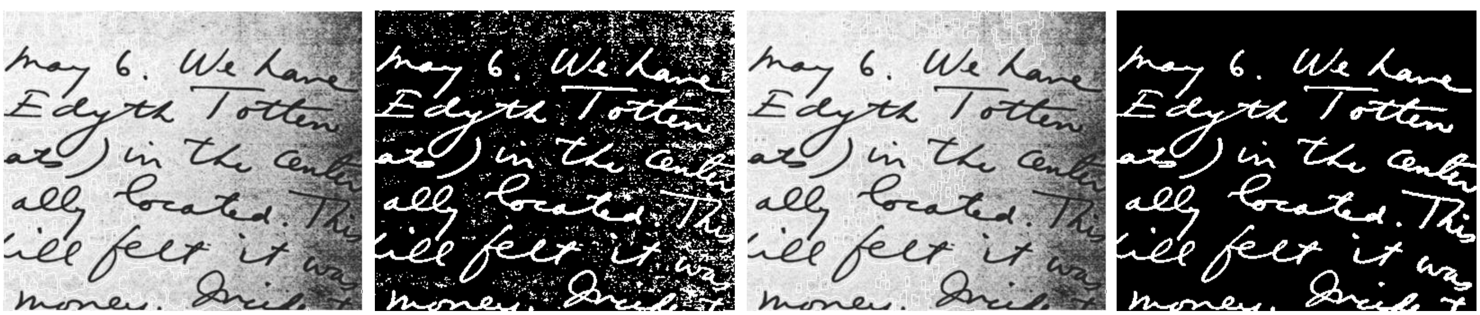
5.2.2.3 DIBCO-2011 Dataset Results



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

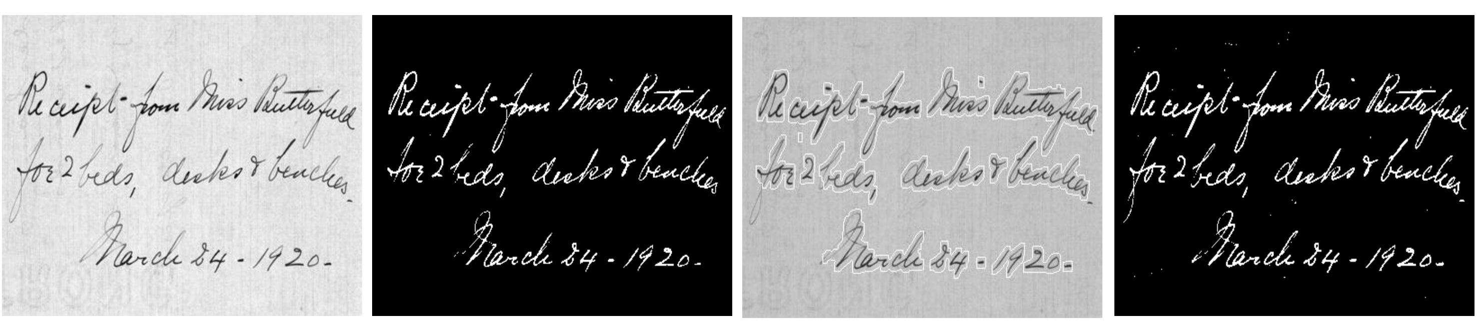


(e) (f) (g) (h)

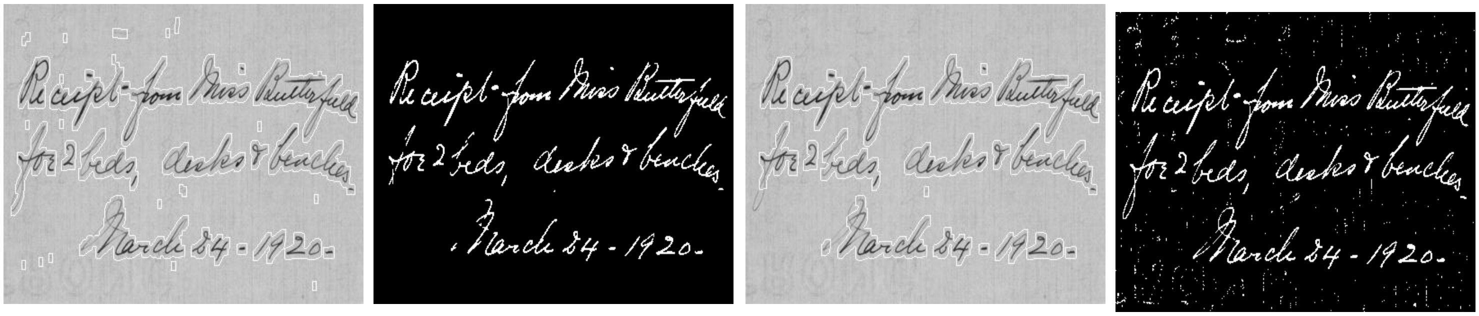


(i) (j) (k) (l)

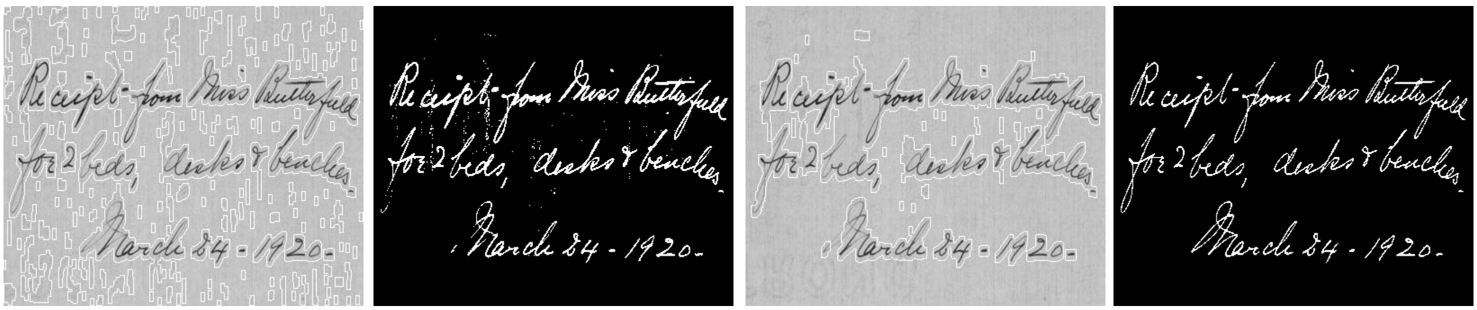
Fig (5.57)



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

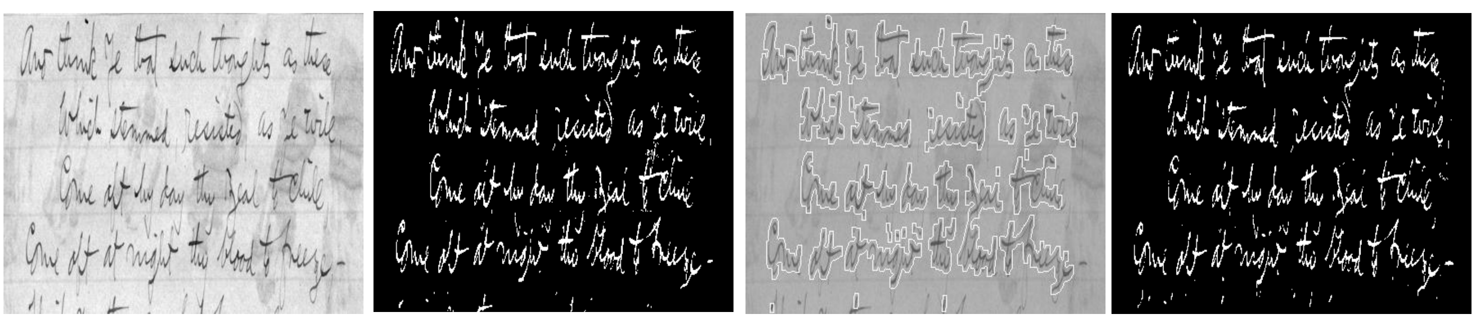


(e) (f) (g) (h)

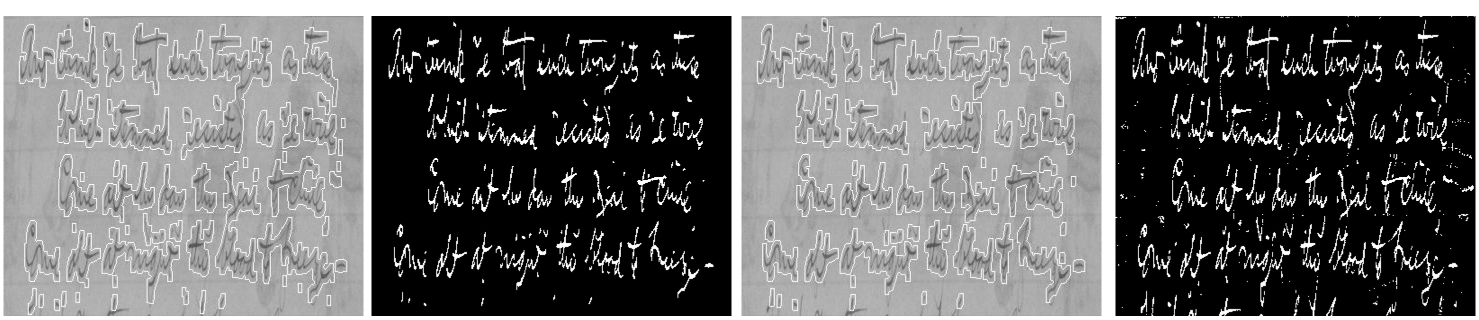


(i) (j) (k) (l)

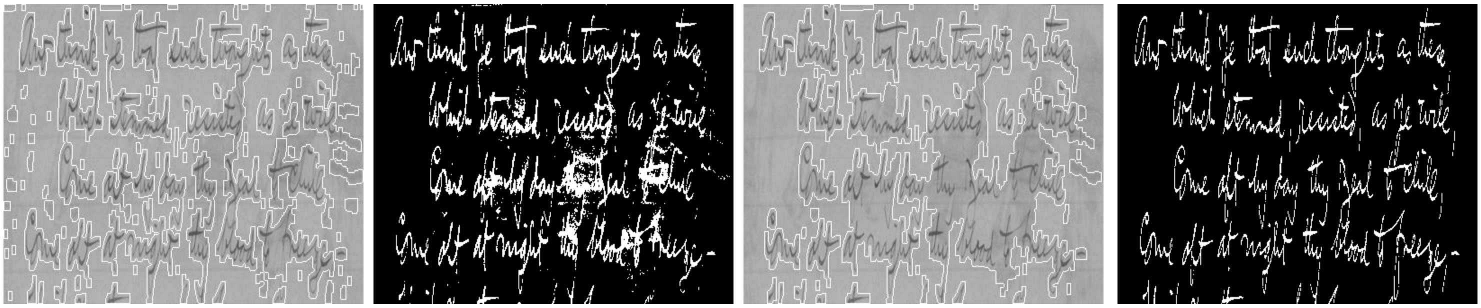
Fig (5.58)



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

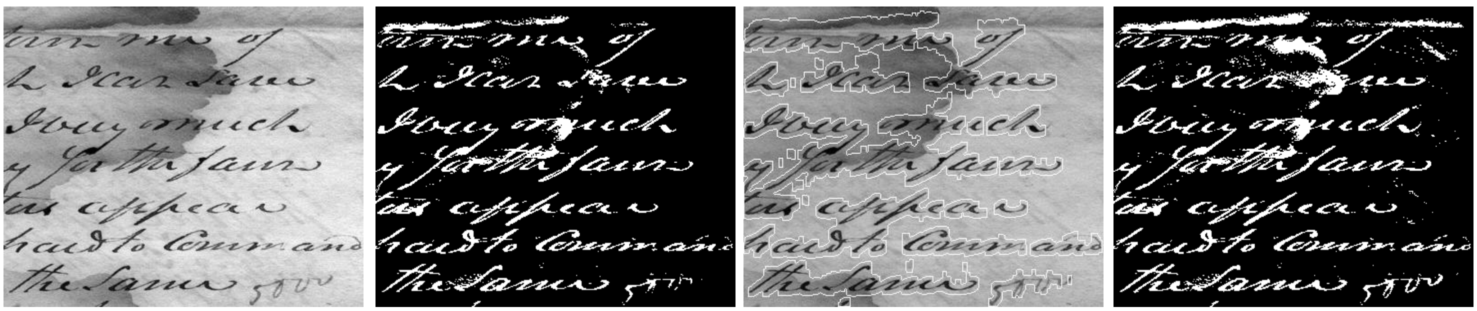


(e) (f) (g) (h)

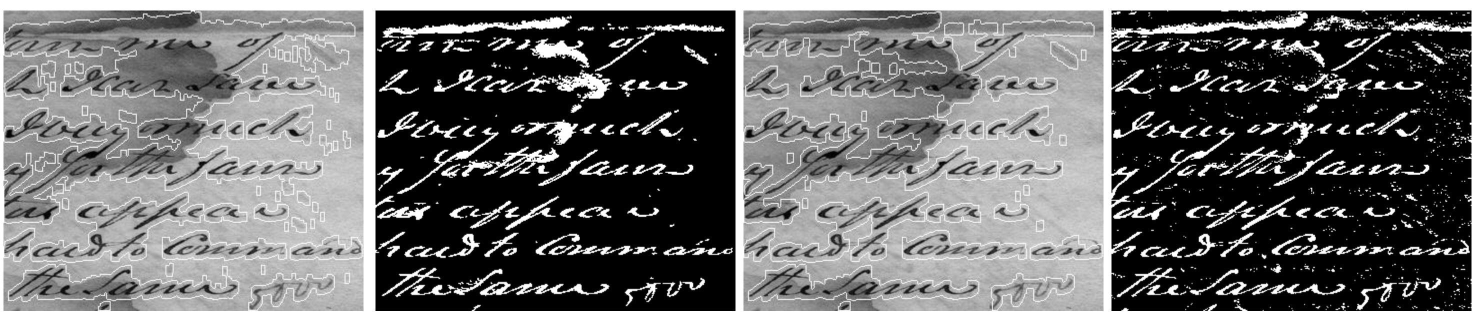


(i) (j) (k) (l)

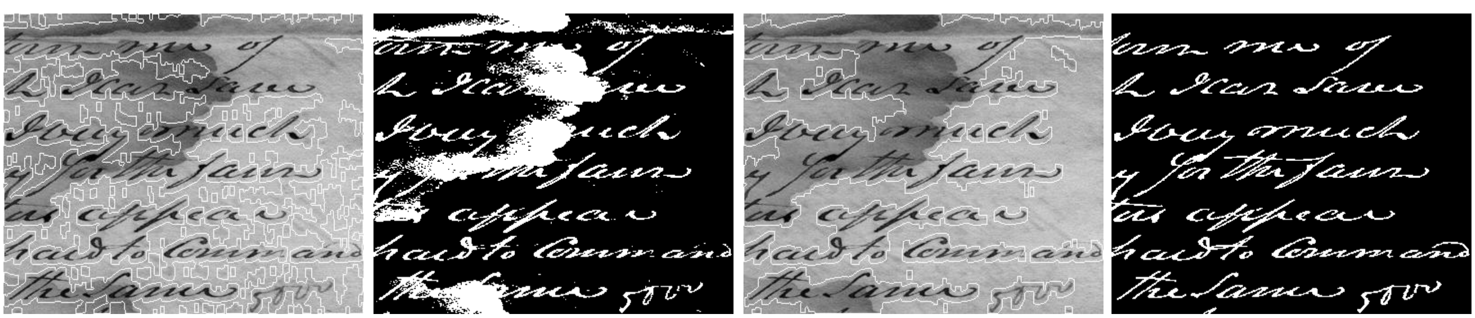
Fig (5.59)



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

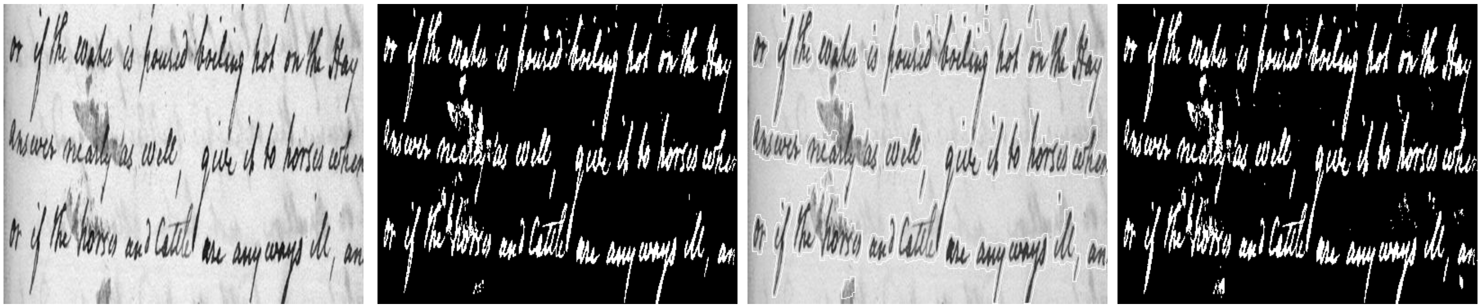


(e) (f) (g) (h)

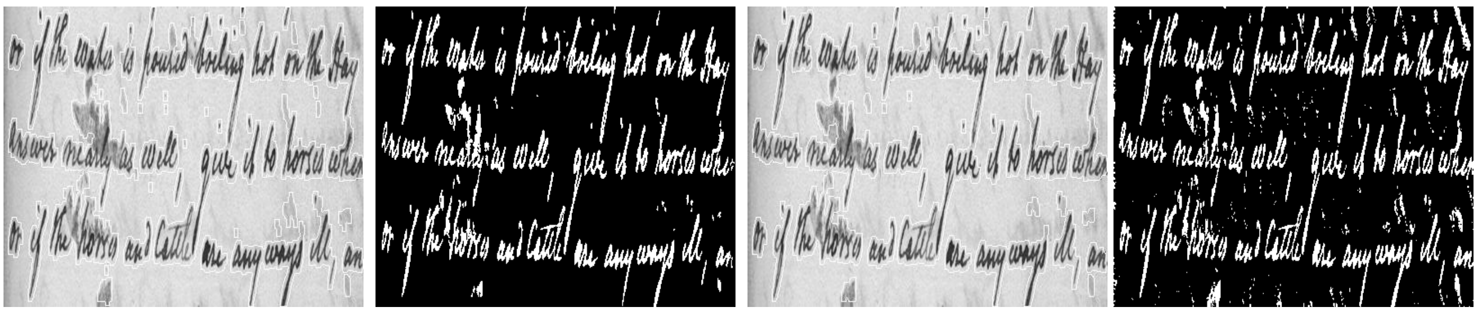


(i) (j) (k) (l)

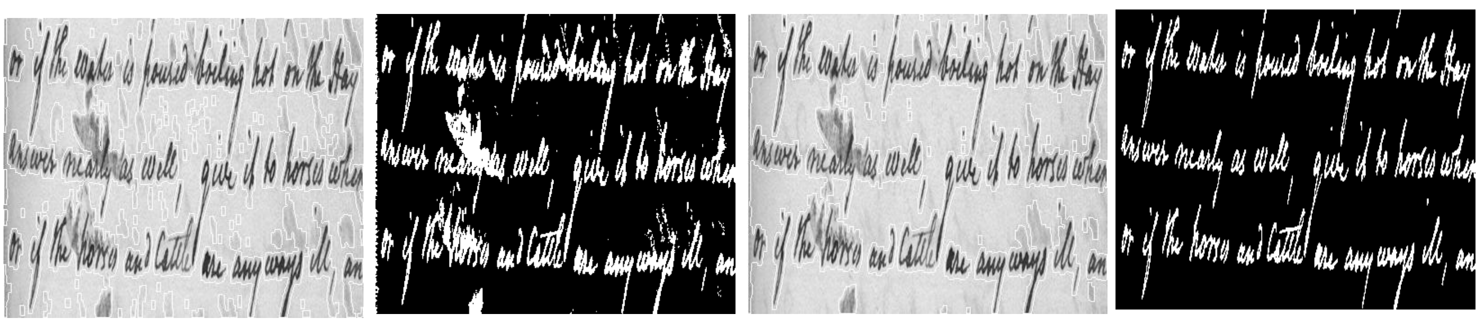
Fig (5.60)



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

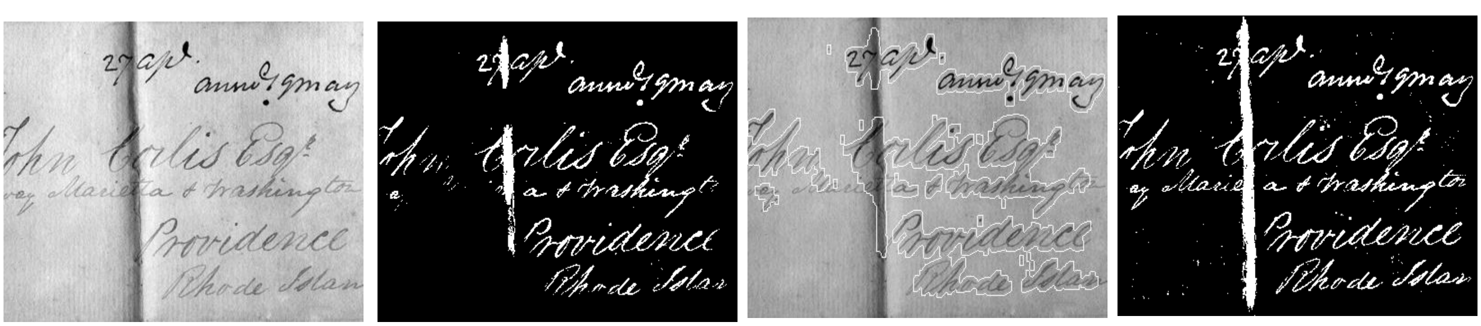


(e) (f) (g) (h)

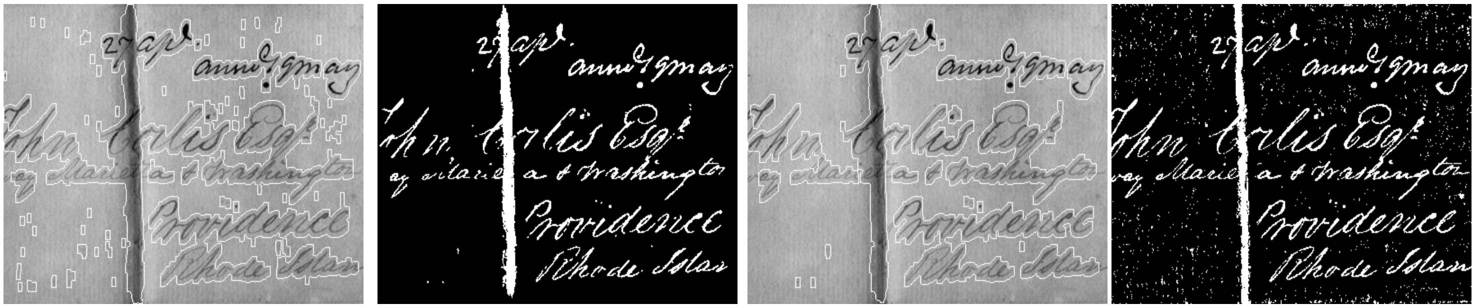


(i) (j) (k) (l)

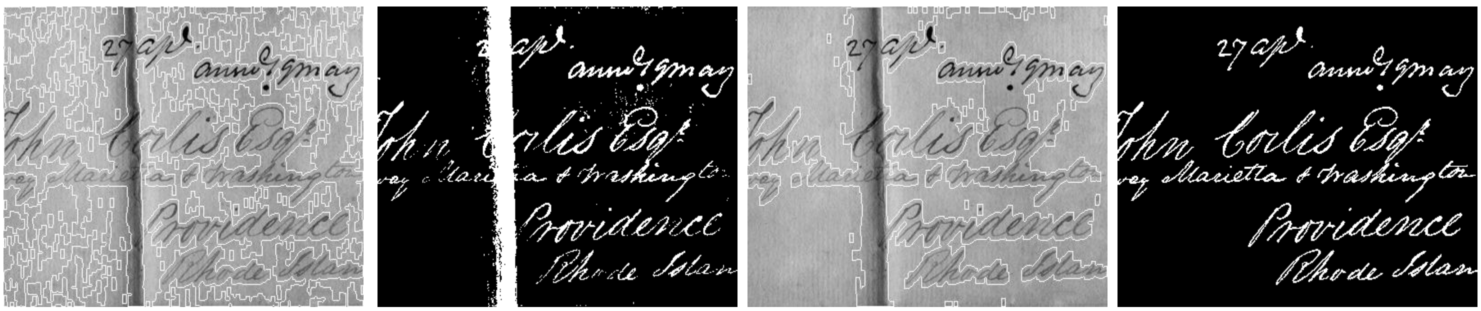
Fig (5.61)



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

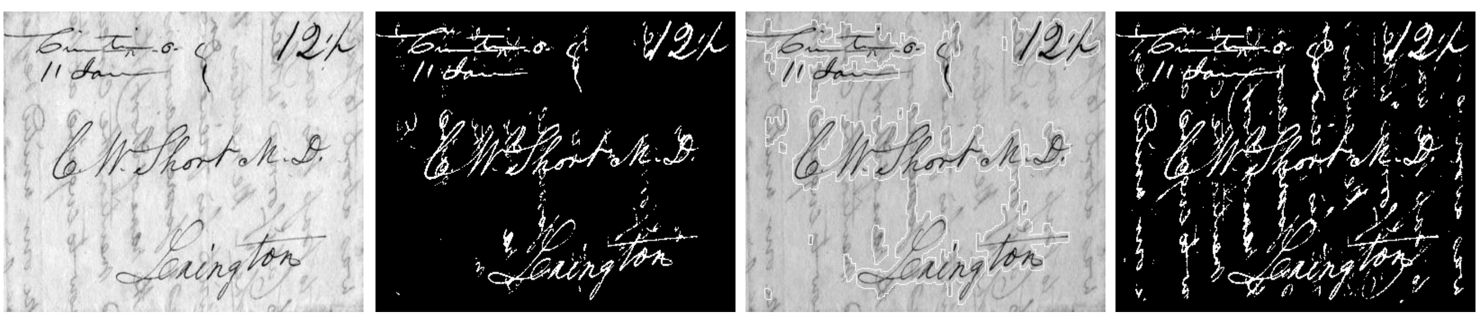


(e) (f) (g) (h)

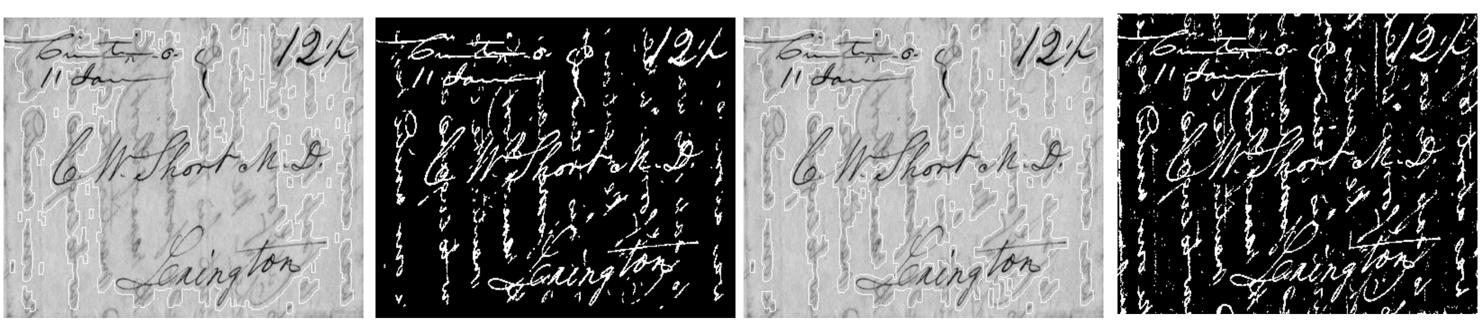


(i) (j) (k) (l)

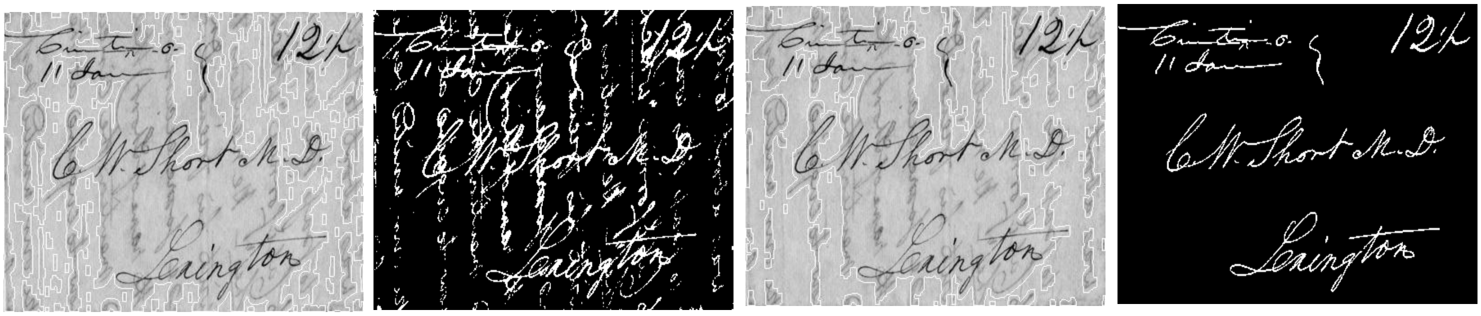
Fig (5.62)



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

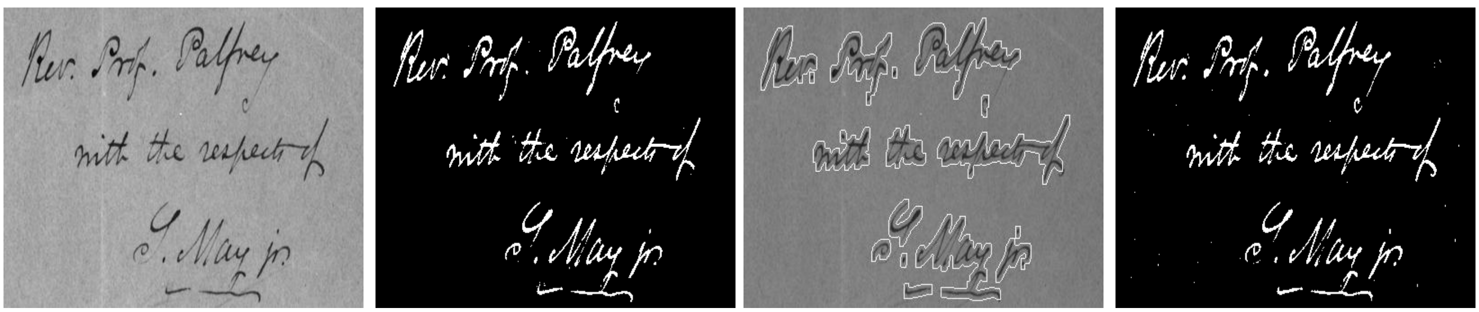


(e) (f) (g) (h)

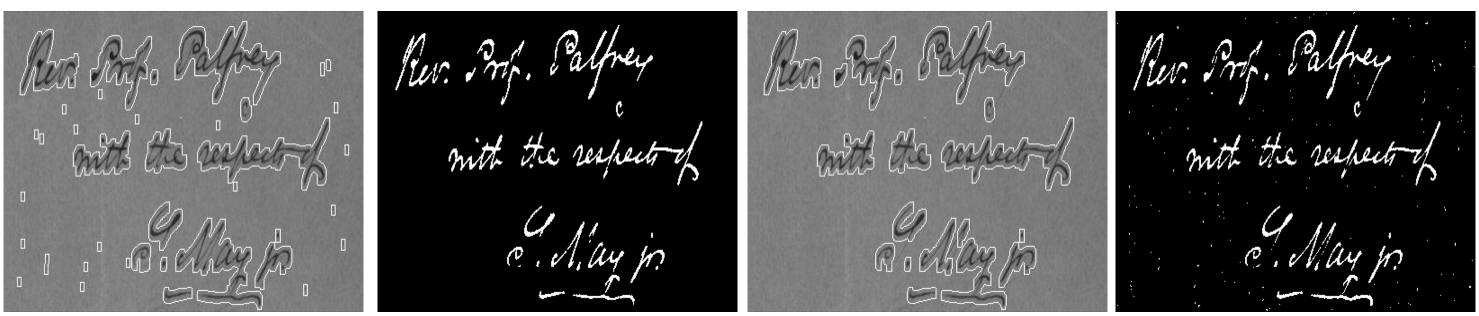


(i) (j) (k) (l)

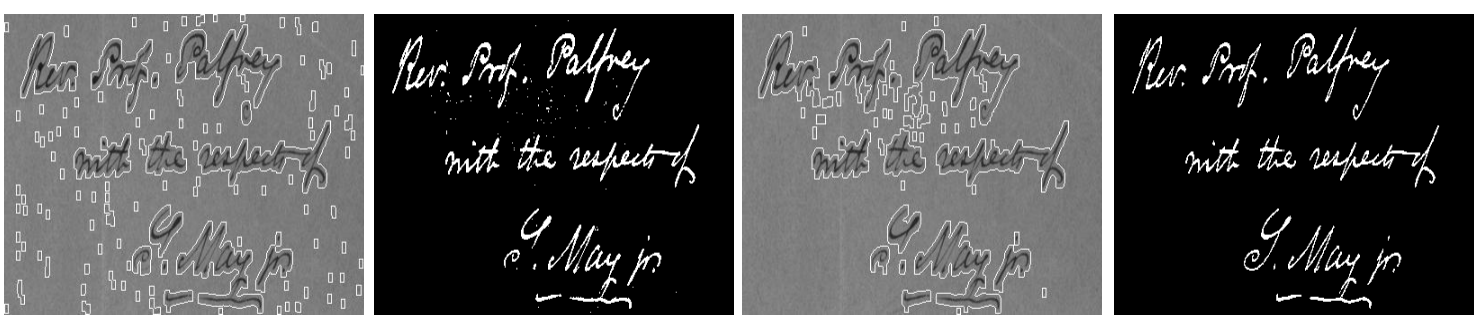
Fig (5.63)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

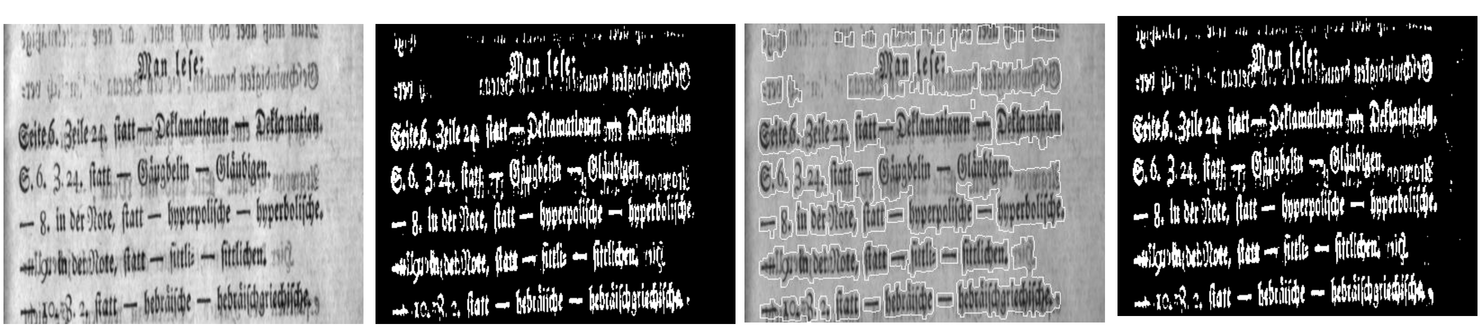
Fig (5.64)

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

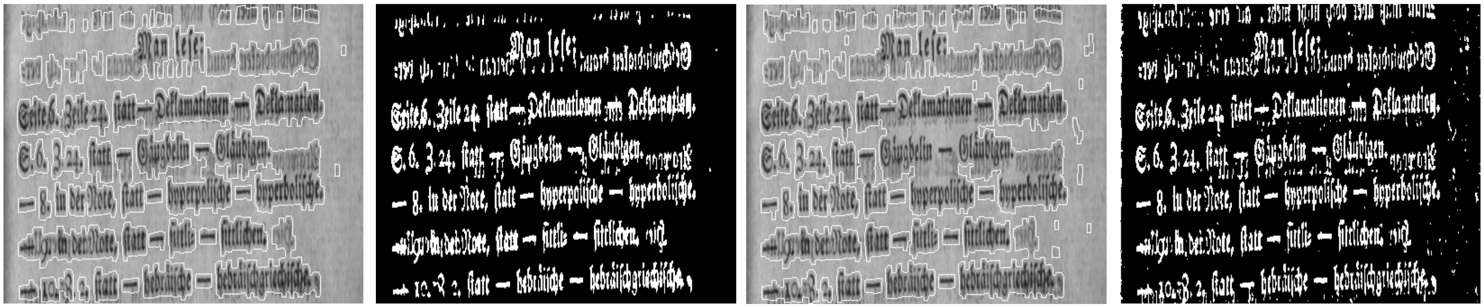
(e) (f) (g) (h)

(i) (j) (k) (l)

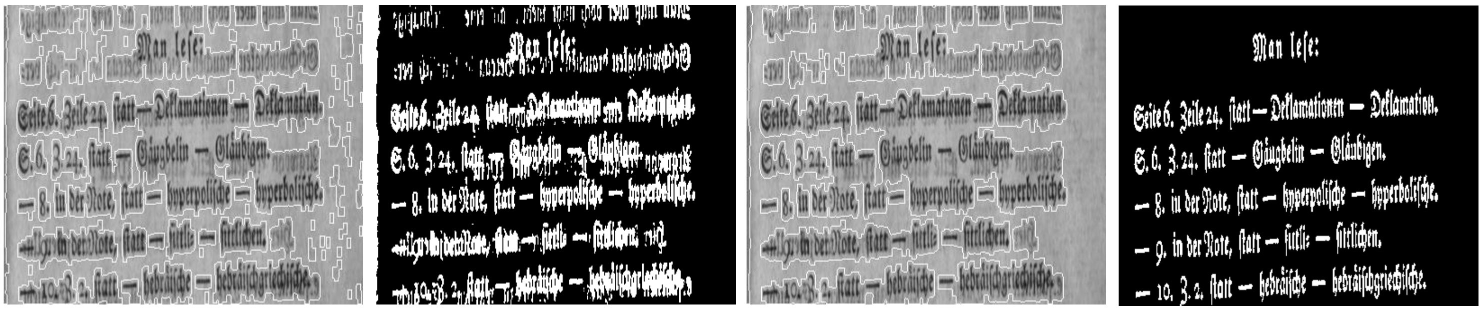
Fig (5.65)



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

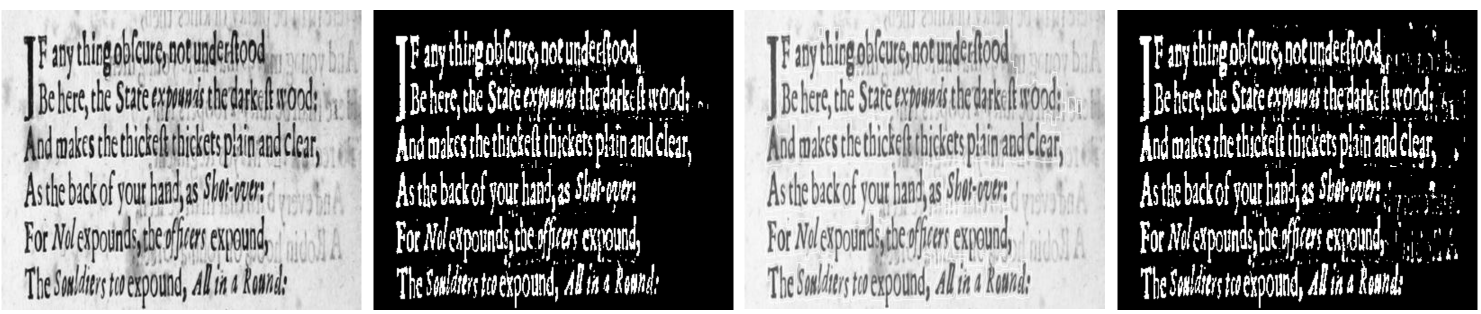


(e) (f) (g) (h)

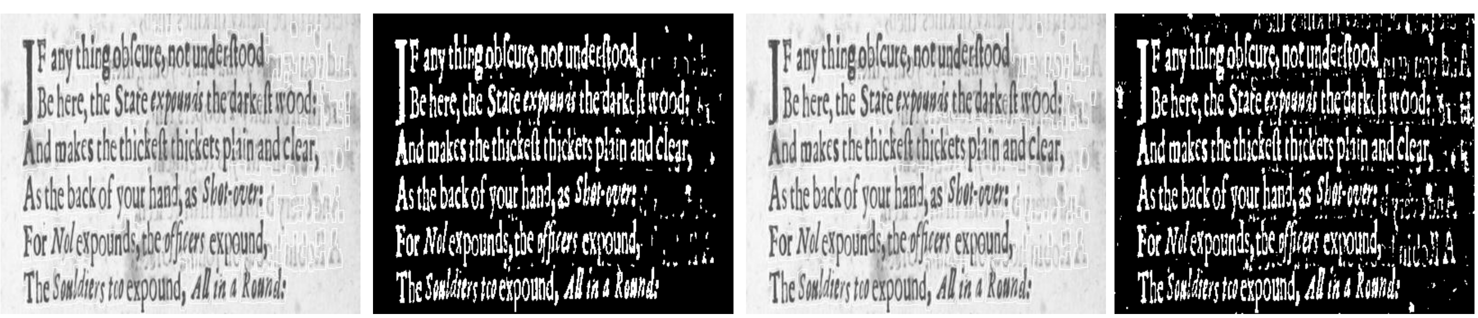


(i) (j) (k) (l)

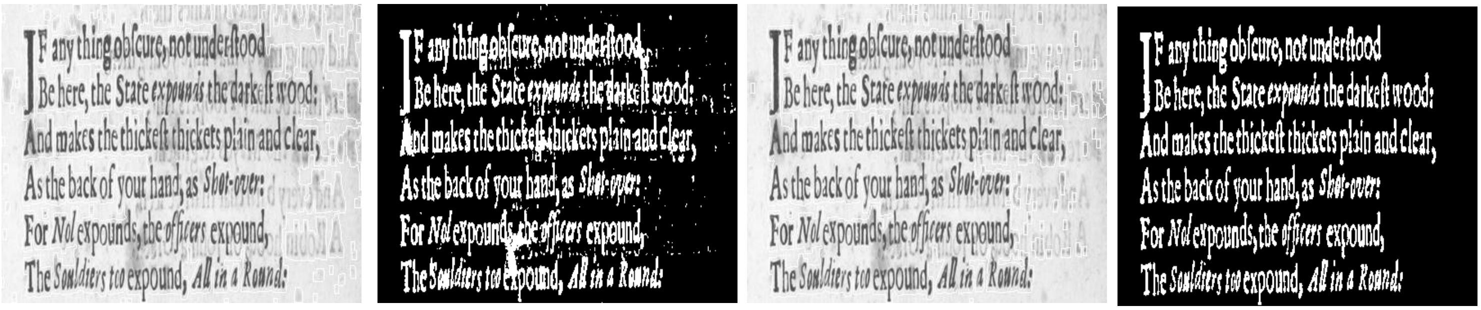
Fig (5.66)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

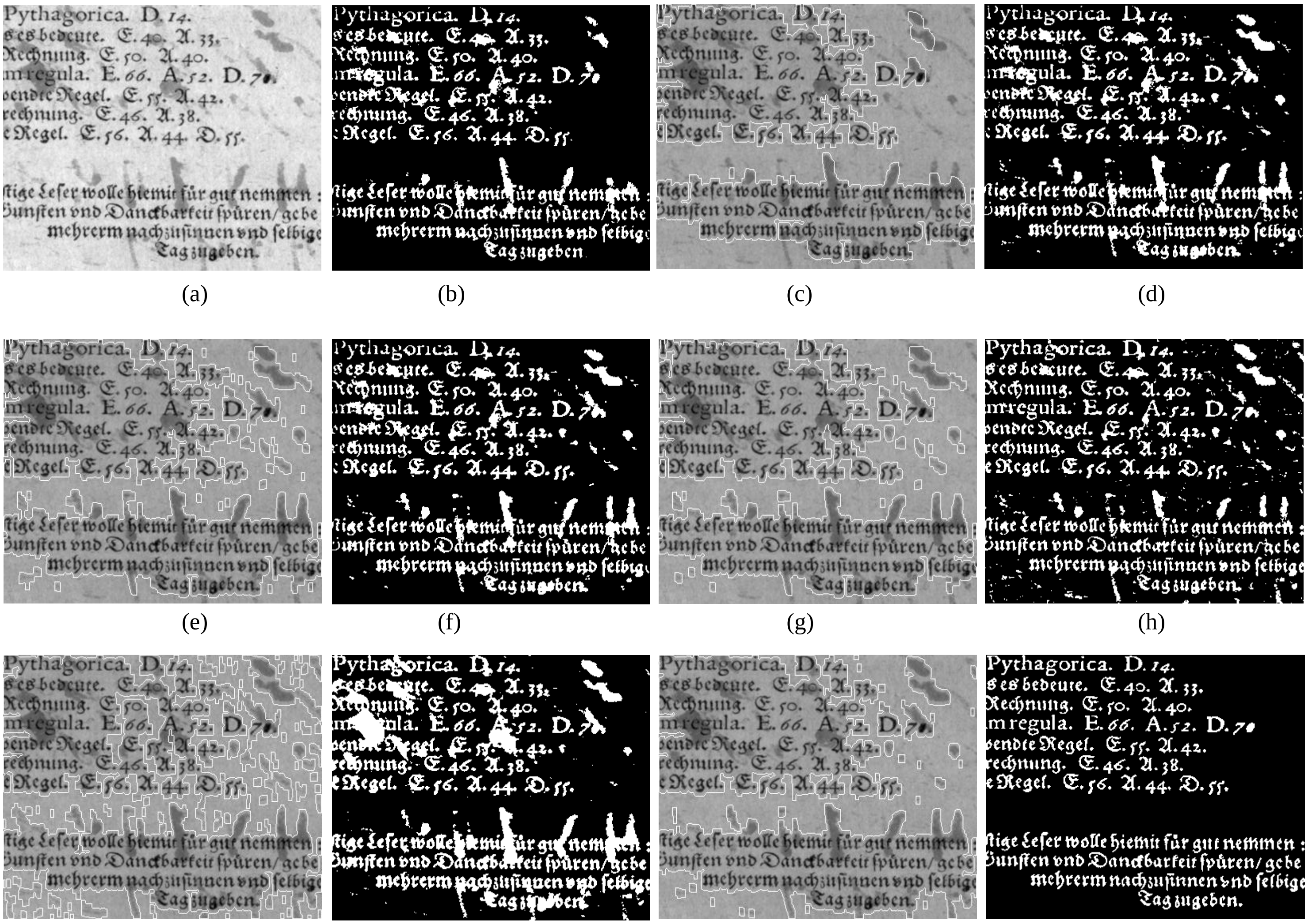
Fig (5.67)

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

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig (5.68)

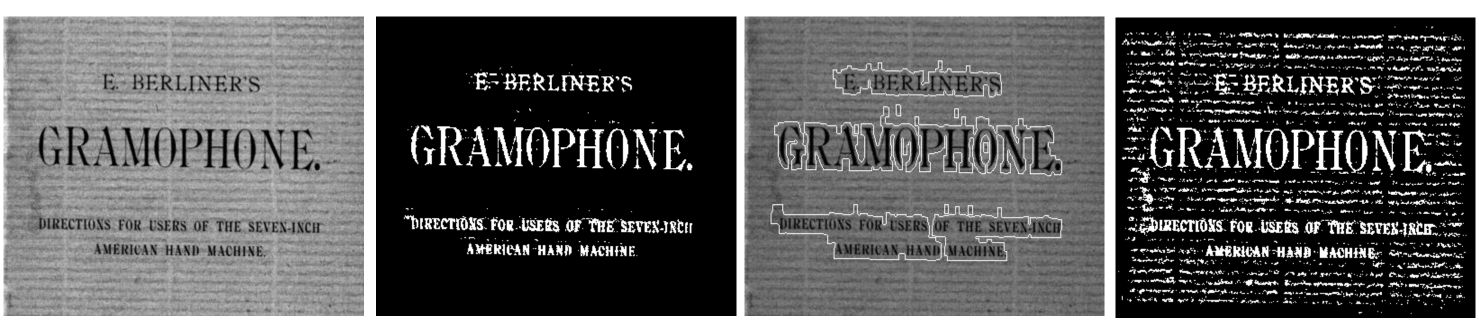


(a) (b) (c) (d)

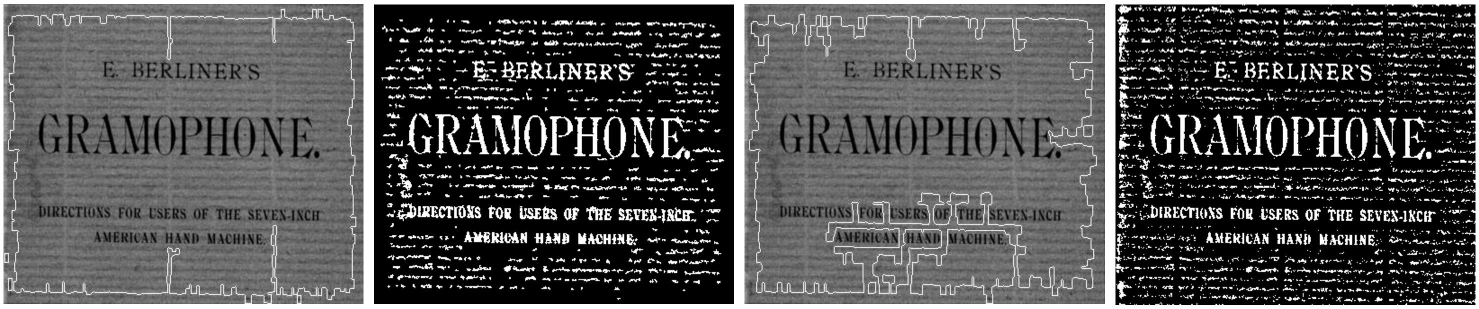
(e) (f) (g) (h)

(i) (j) (k) (l)

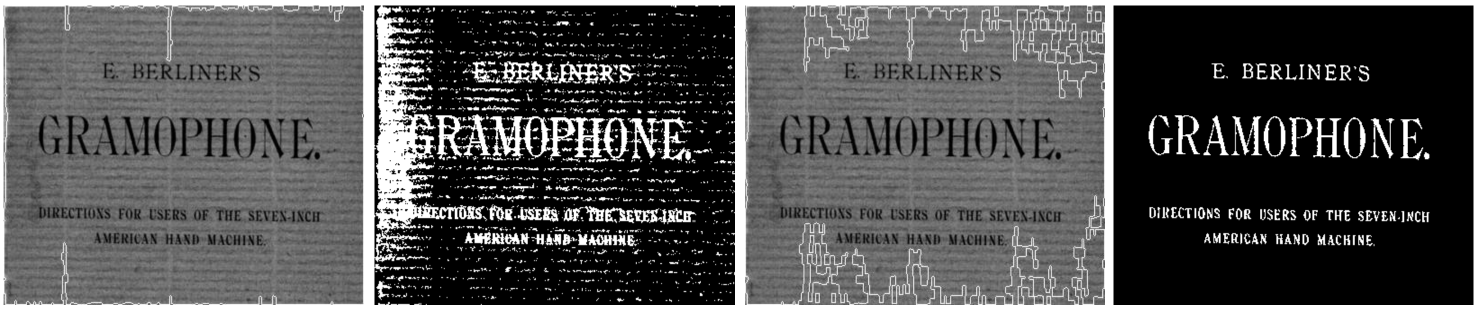
Fig (5.69)



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



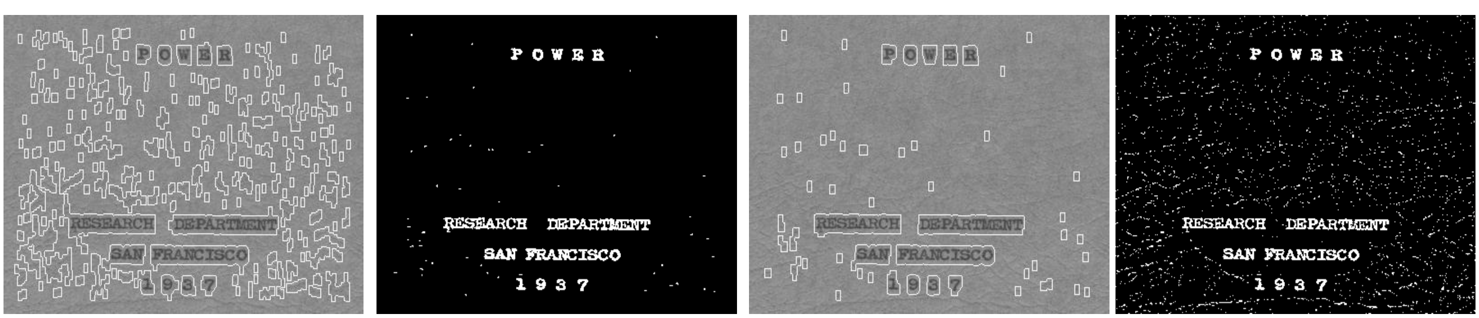
(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.70)

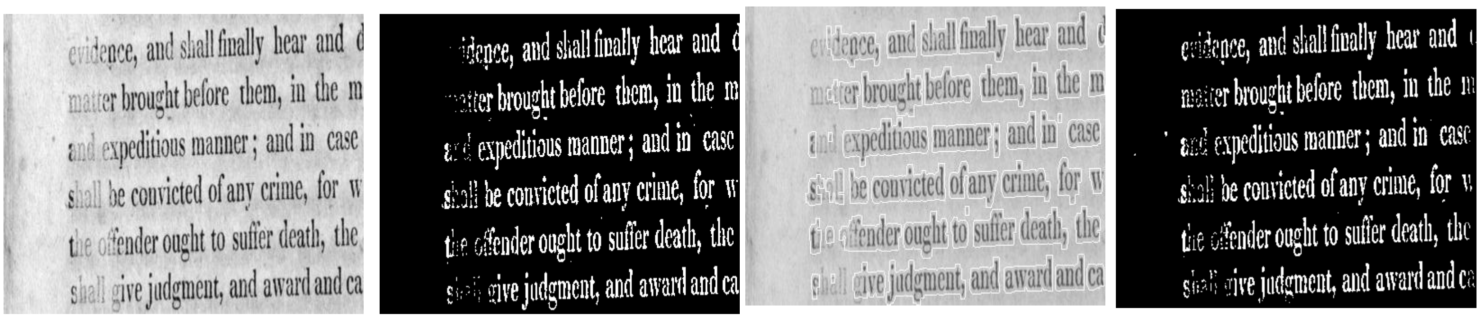
(a) (b) (c) (d)



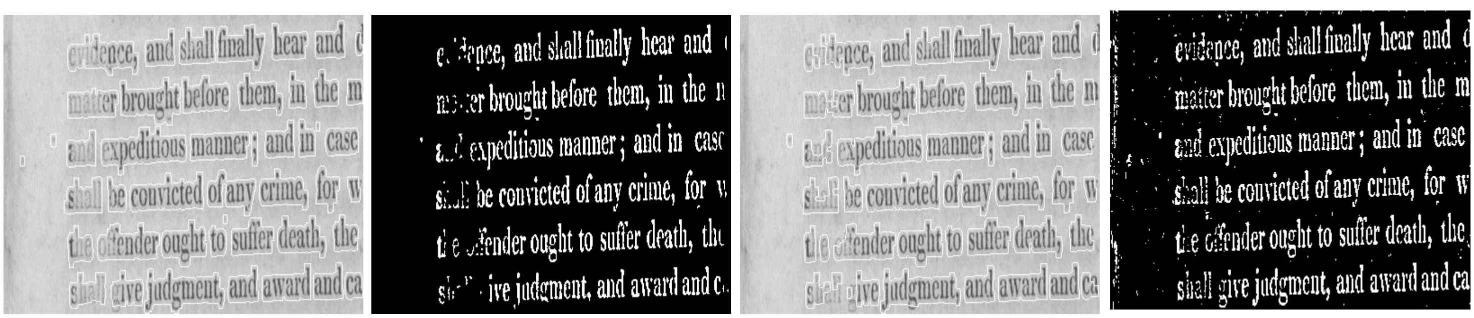
(e) (f) (g) (h)

(i) (j) (k) (l)

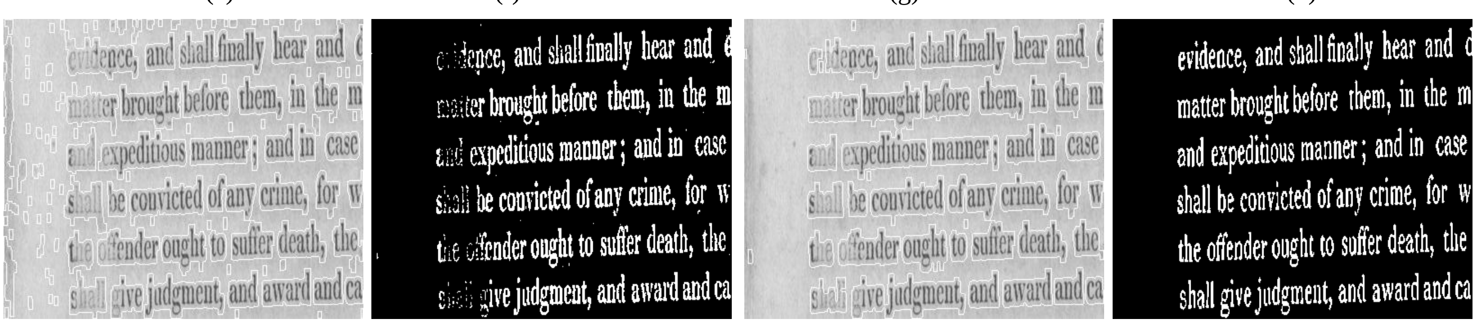
Fig (5.71)



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



(e) (f) (g) (h)

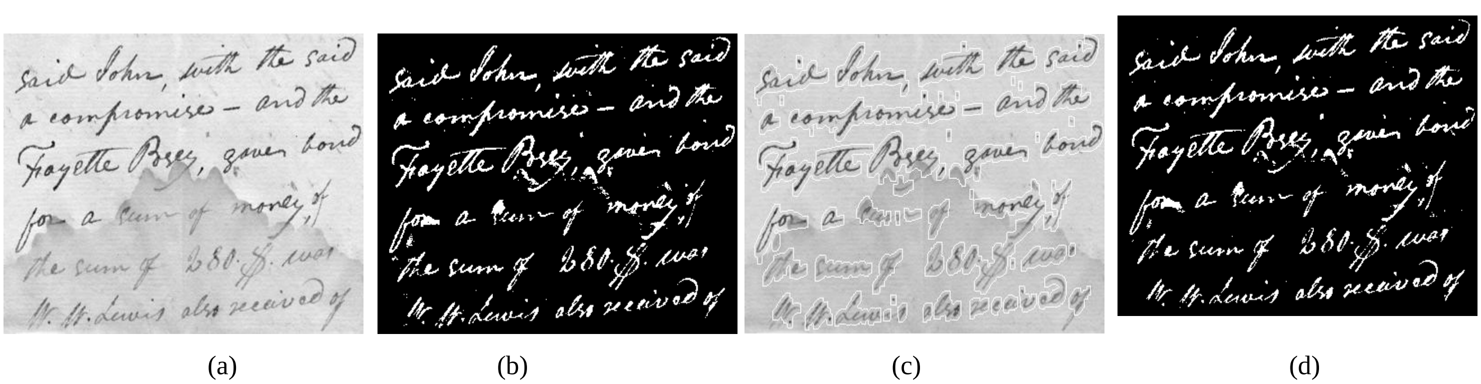


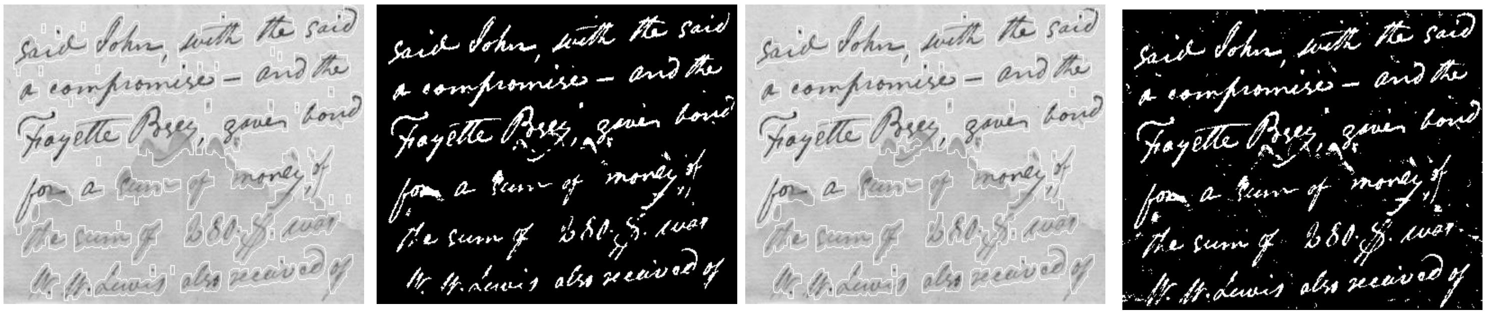
(i) (j) (k) (l)

Fig (5.72)

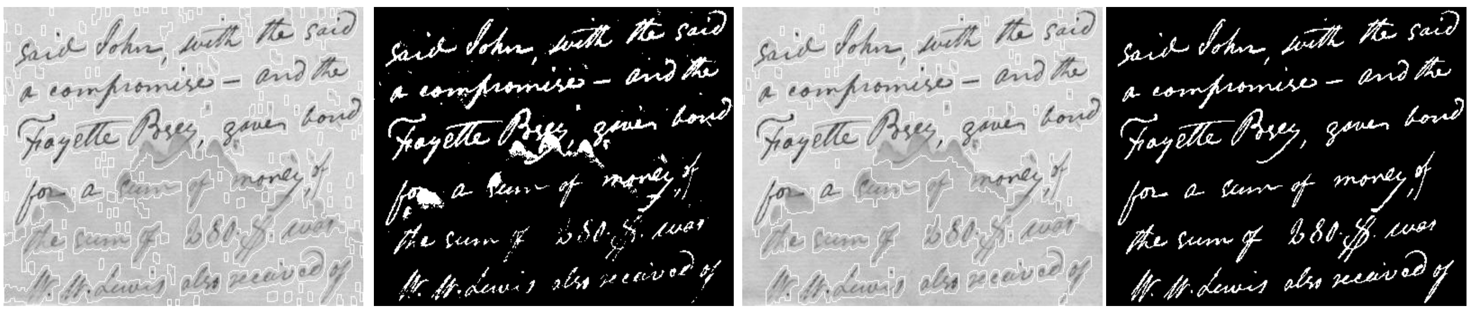
5.2.2.4 DIBCO-2012 Dataset Results

Fig (5.73)



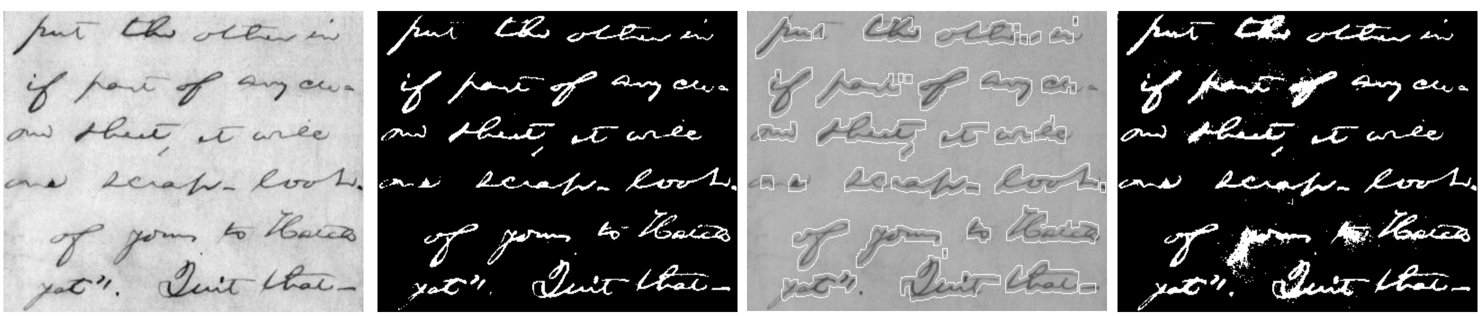


(e) (f) (g) (h)

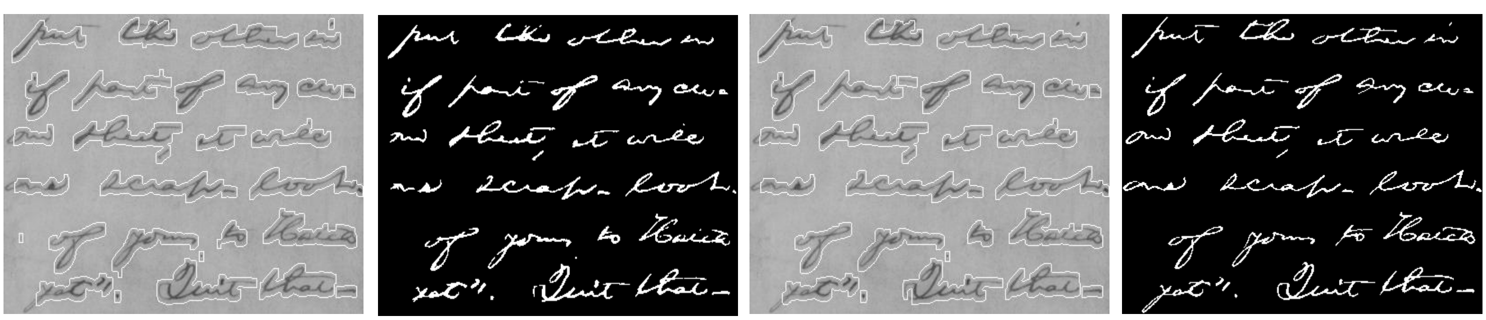


(i) (j) (k) (l)

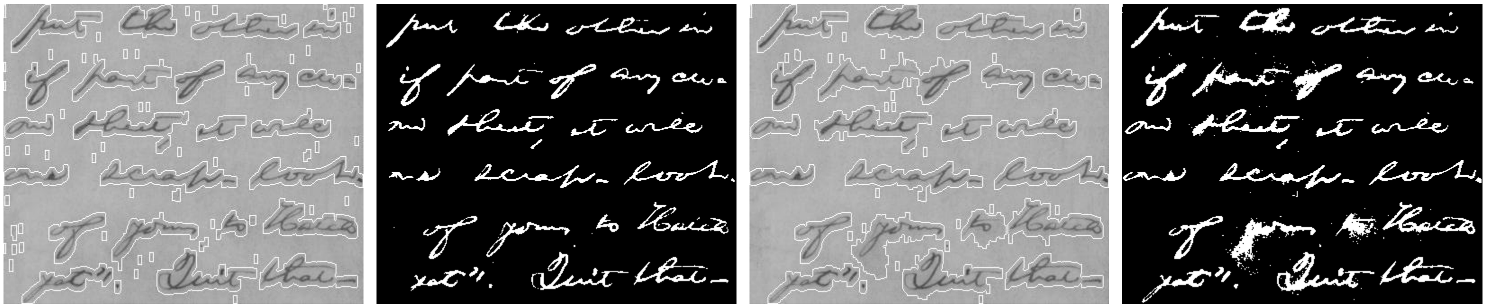
Fig (5.74)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

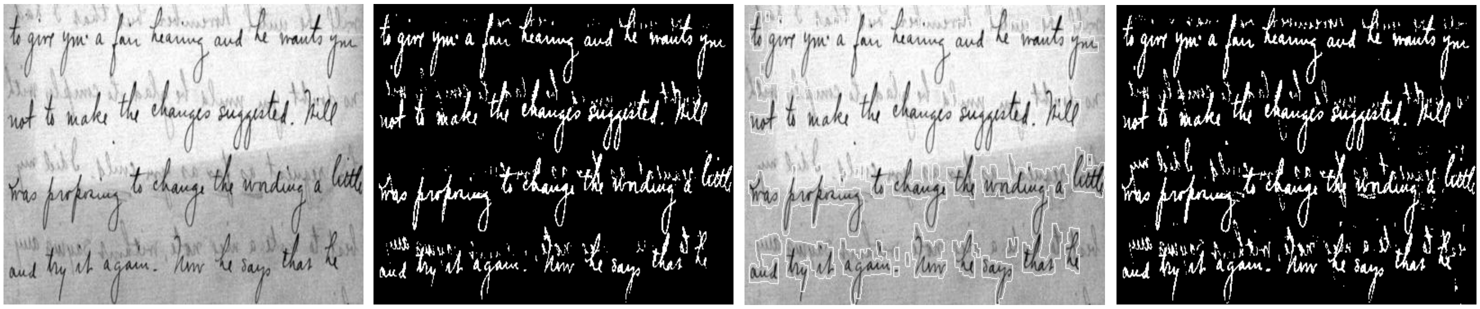
Fig (5.75)

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

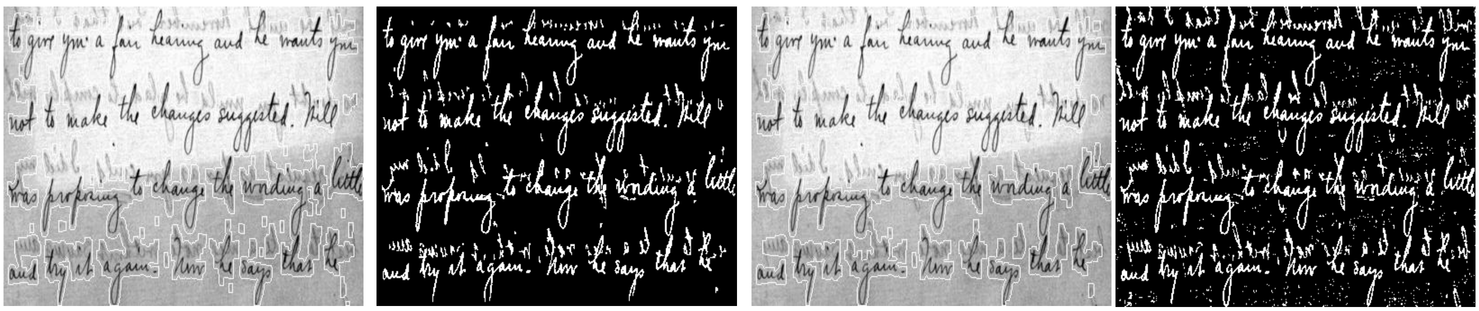
(e) (f) (g) (h)

(i) (j) (k) (l)

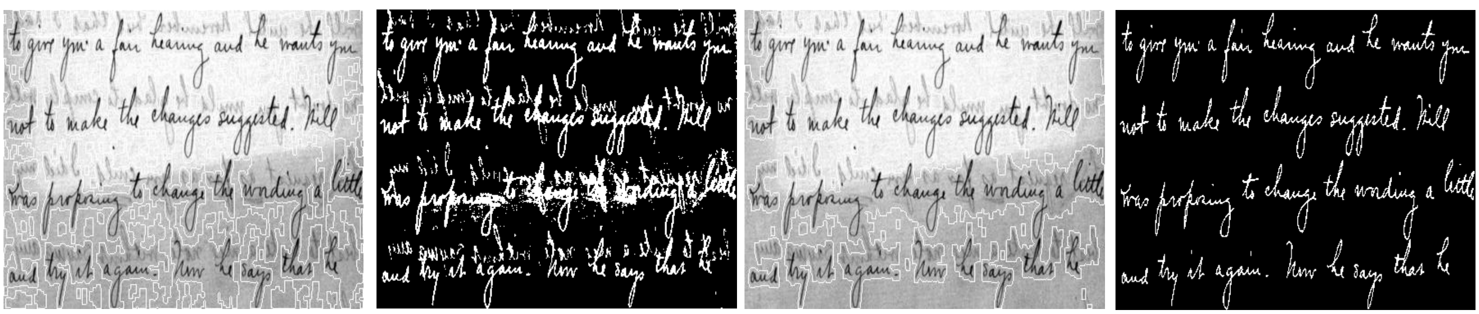
Fig (5.76)



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

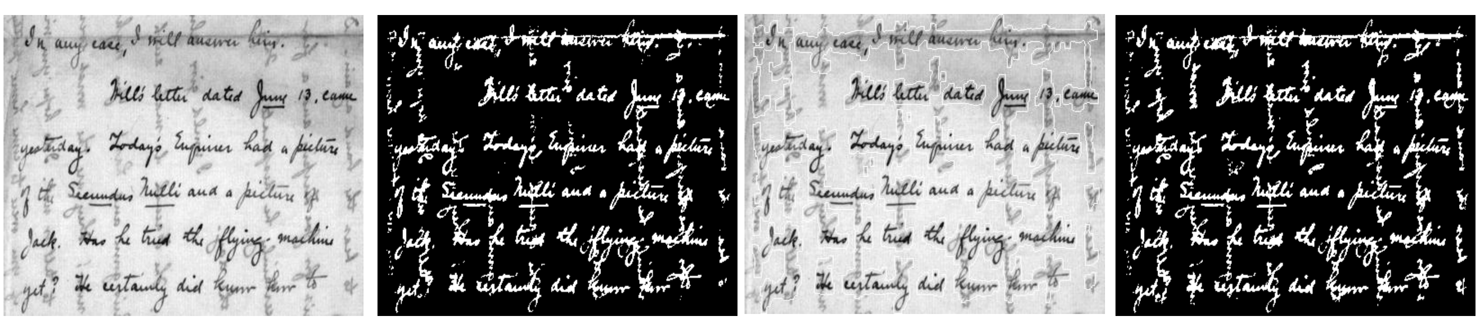


(e) (f) (g) (h)

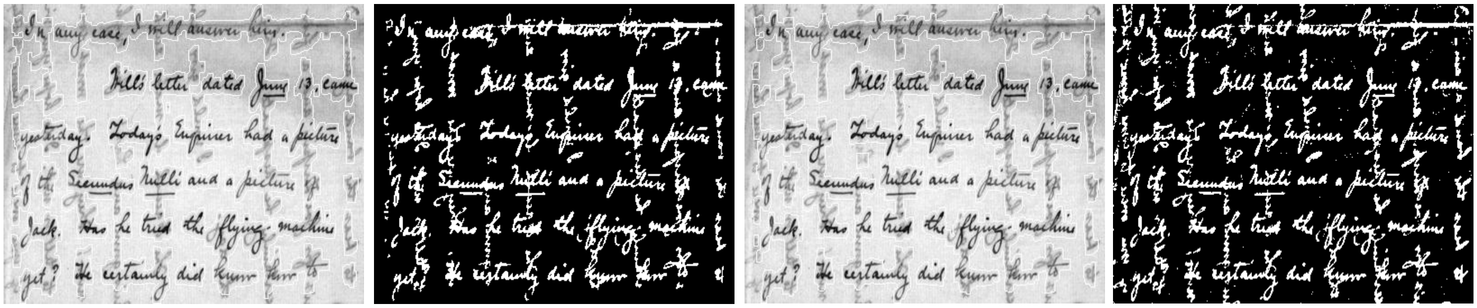


(i) (j) (k) (l)

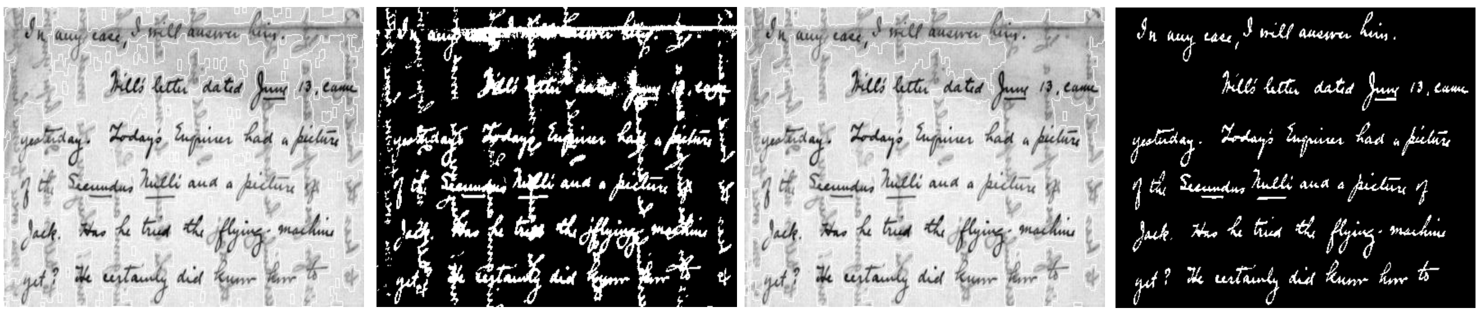
Fig (5.77)



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

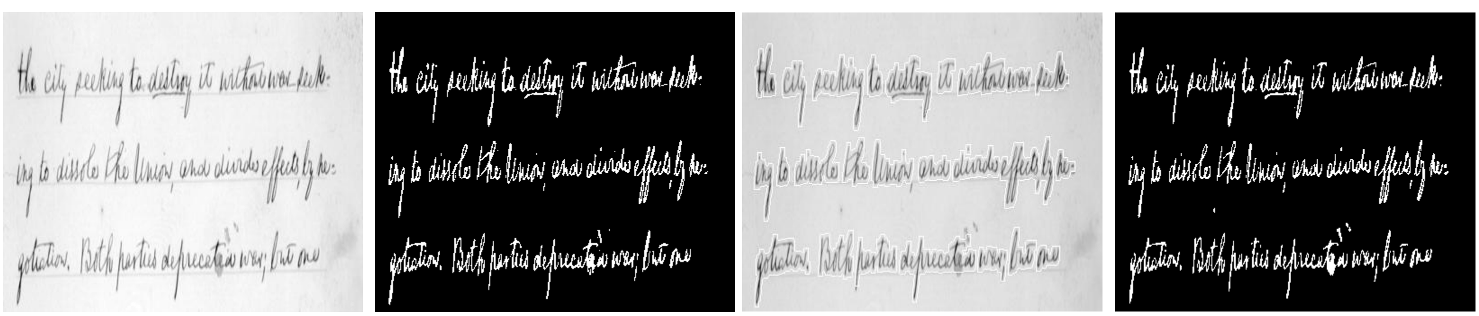


(e) (f) (g) (h)

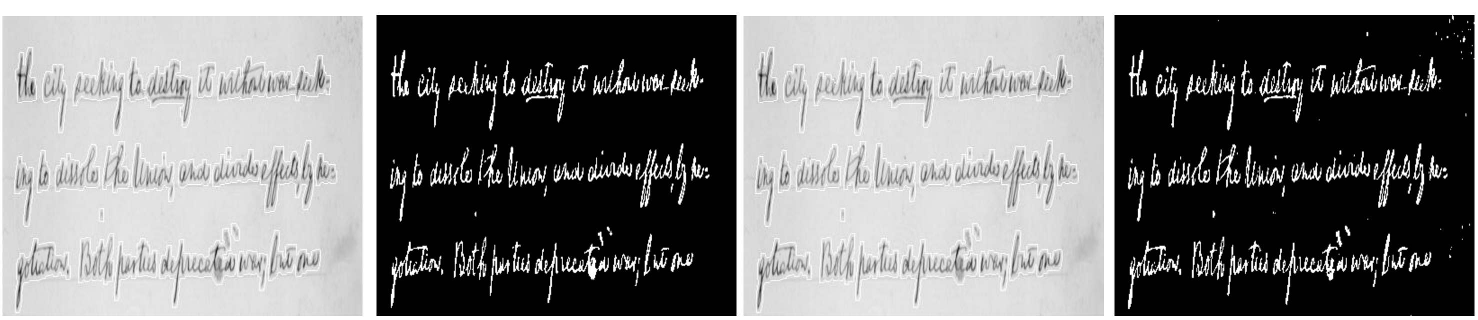


(i) (j) (k) (l)

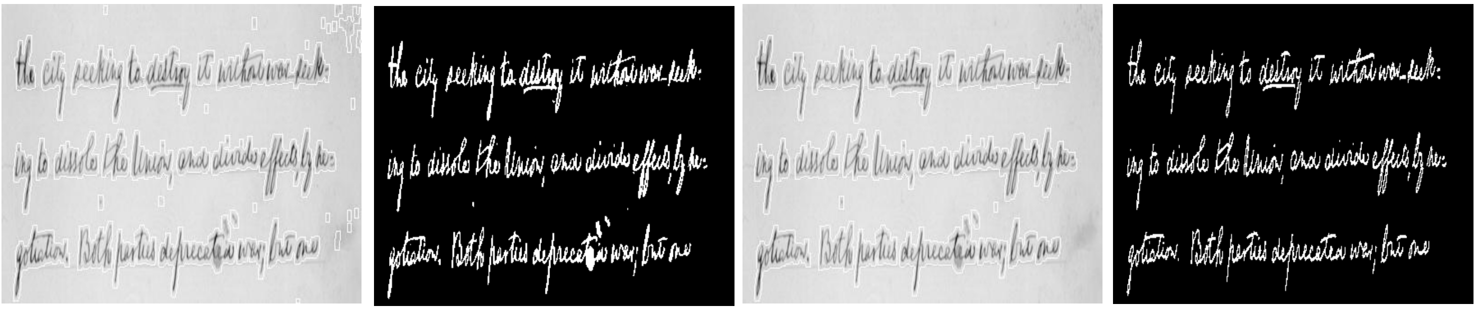
Fig (5.78)



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

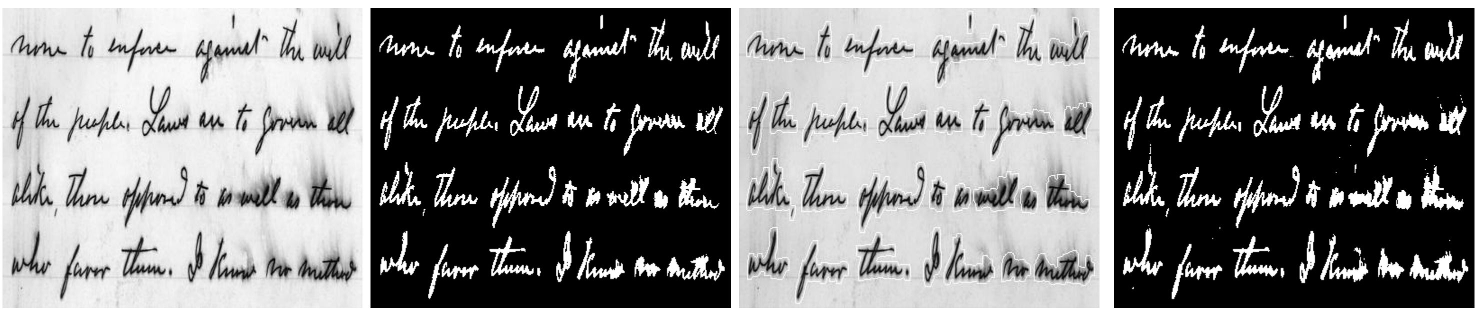


(e) (f) (g) (h)

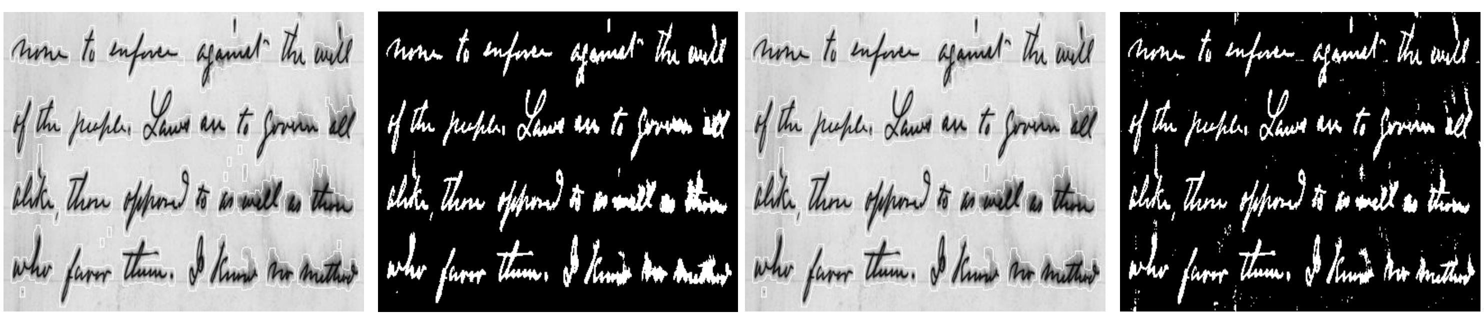


(i) (j) (k) (l)

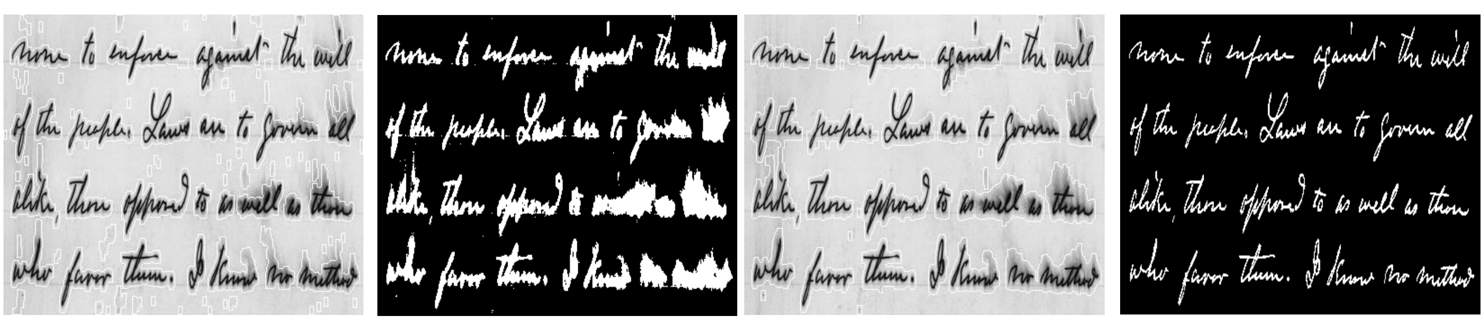
Fig (5.79)



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

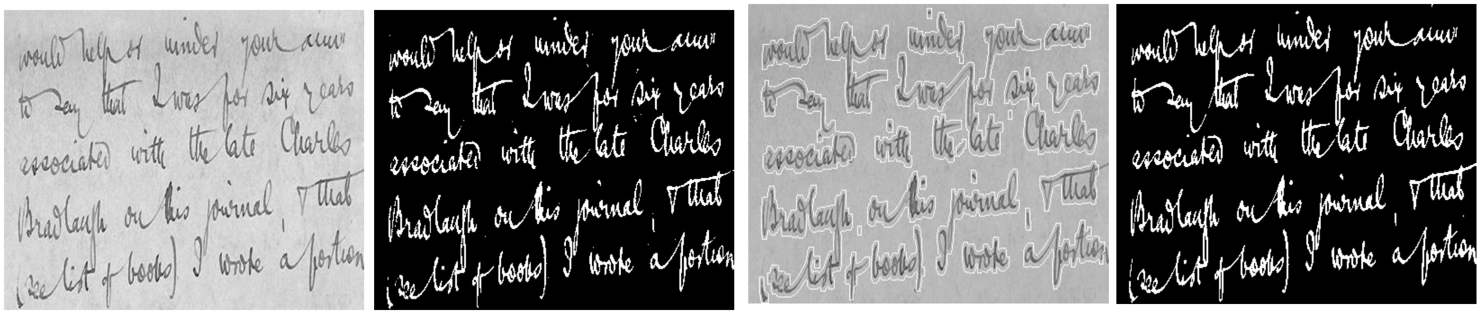


(e) (f) (g) (h)

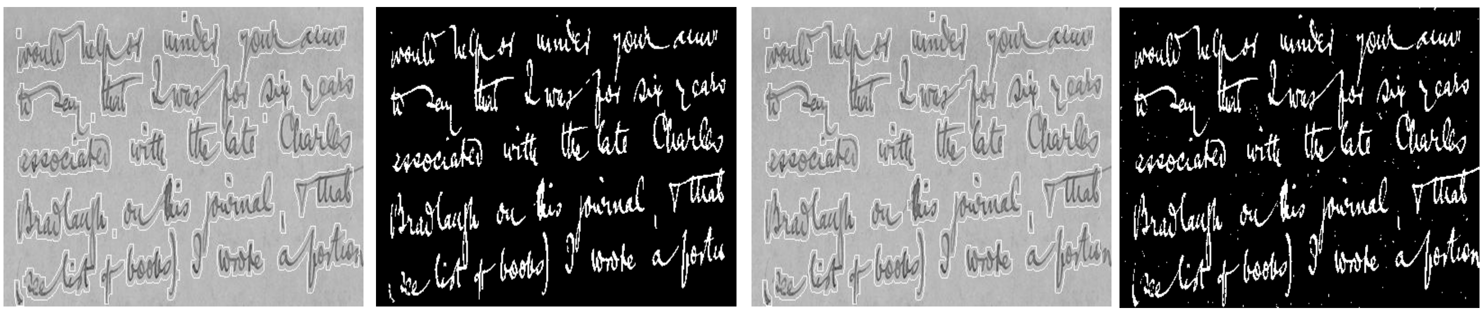


(i) (j) (k) (l)

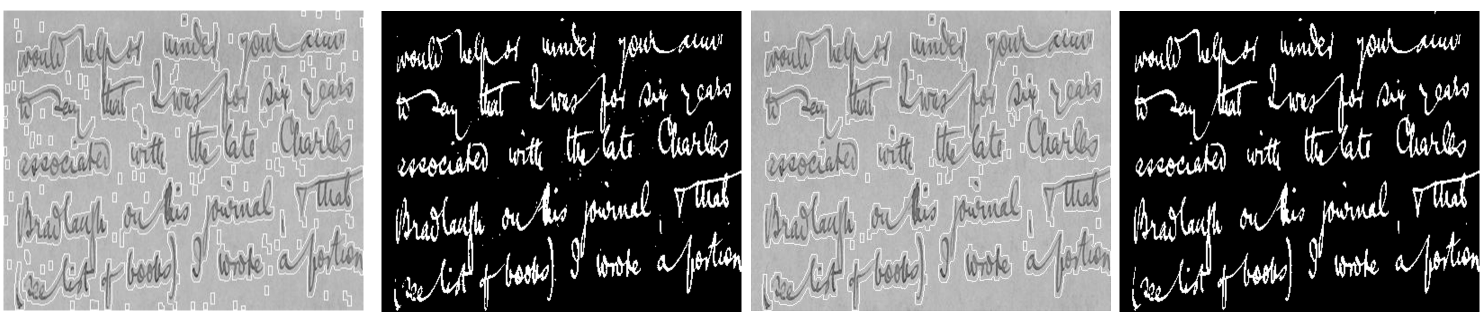
Fig (5.80)



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

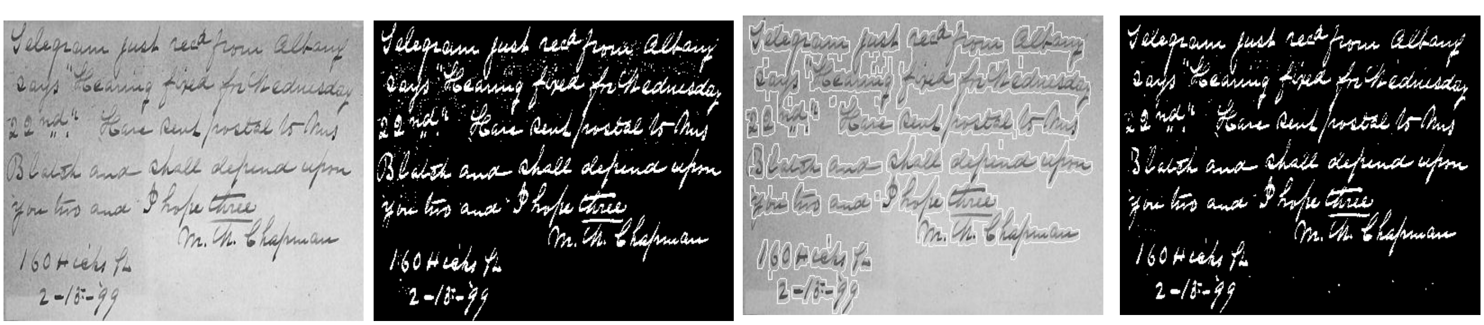


(e) (f) (g) (h)

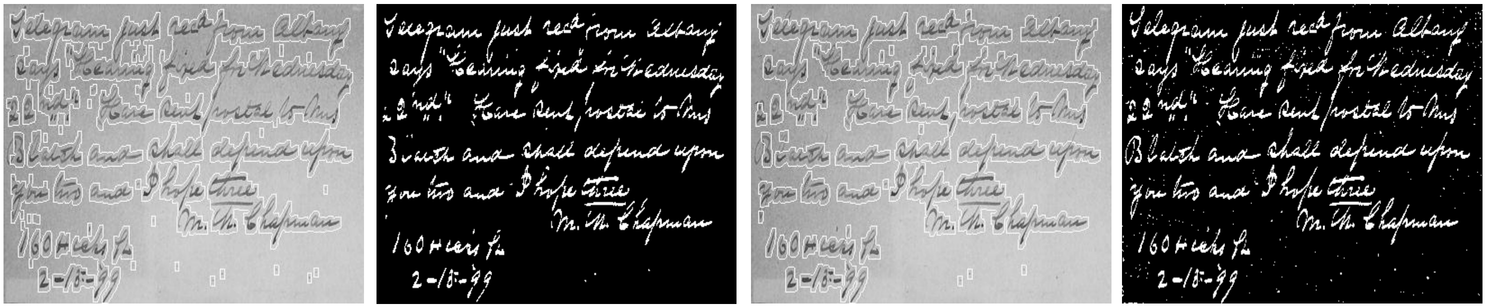


(i) (j) (k) (l)

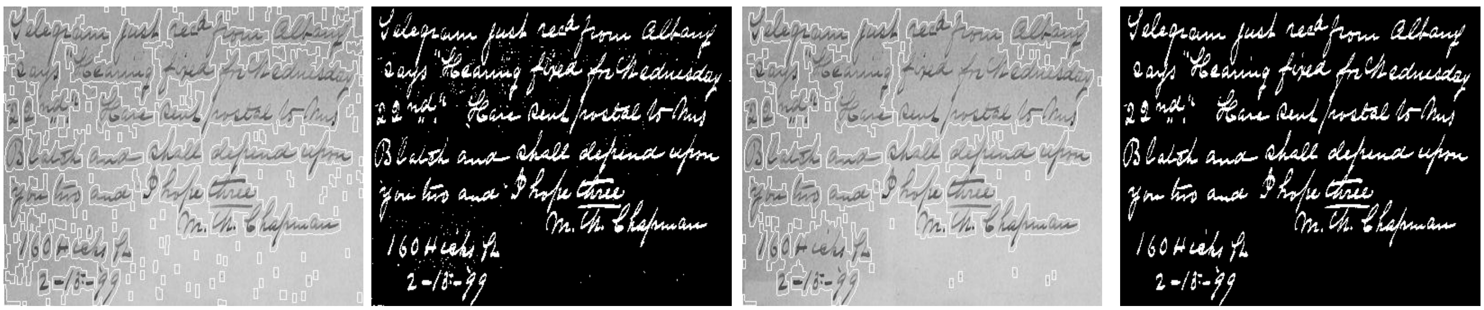
Fig (5.81)



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

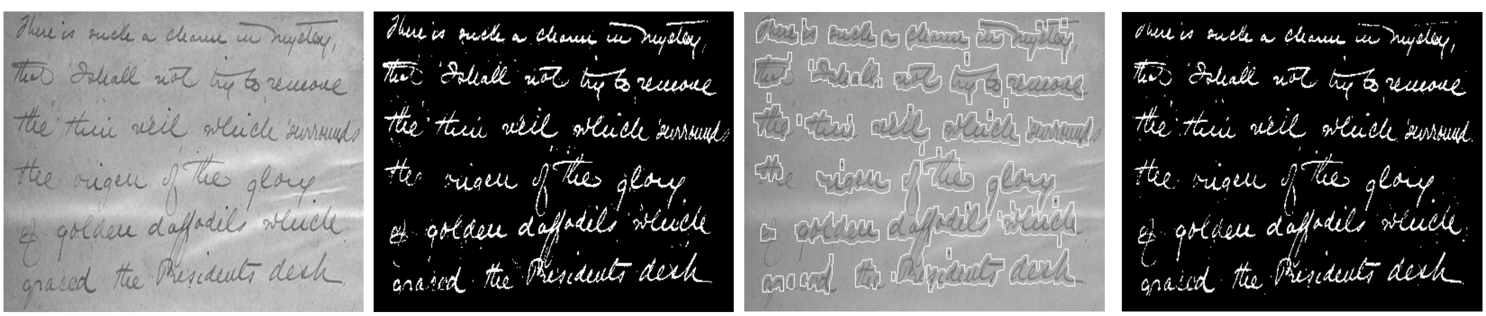


(e) (f) (g) (h)

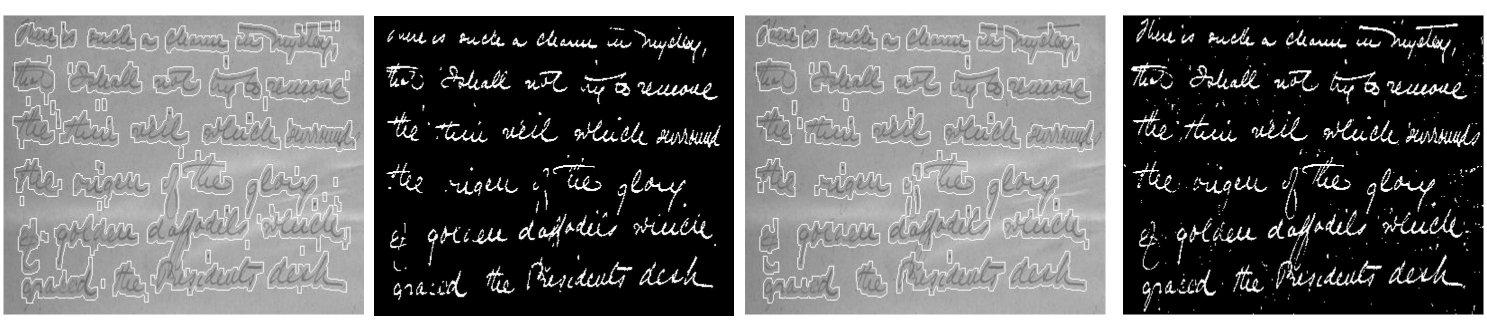


(i) (j) (k) (l)

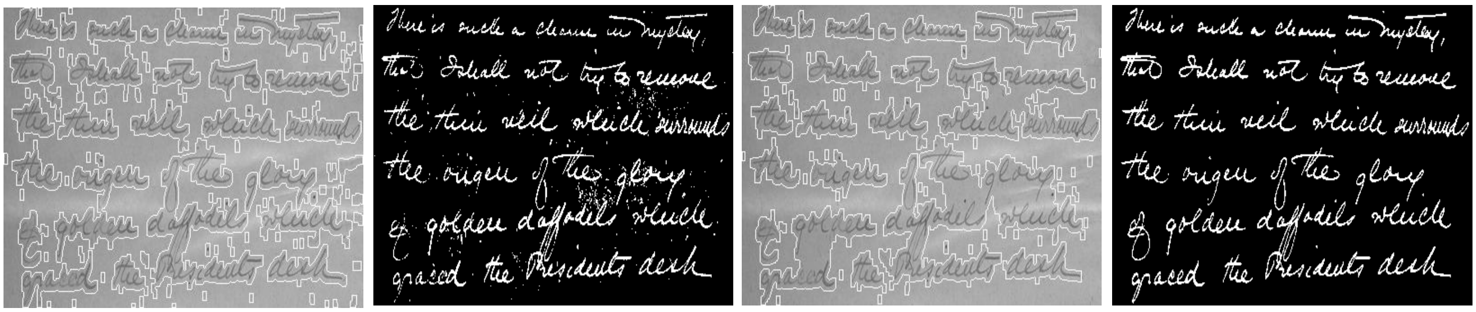
Fig (5.82)



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

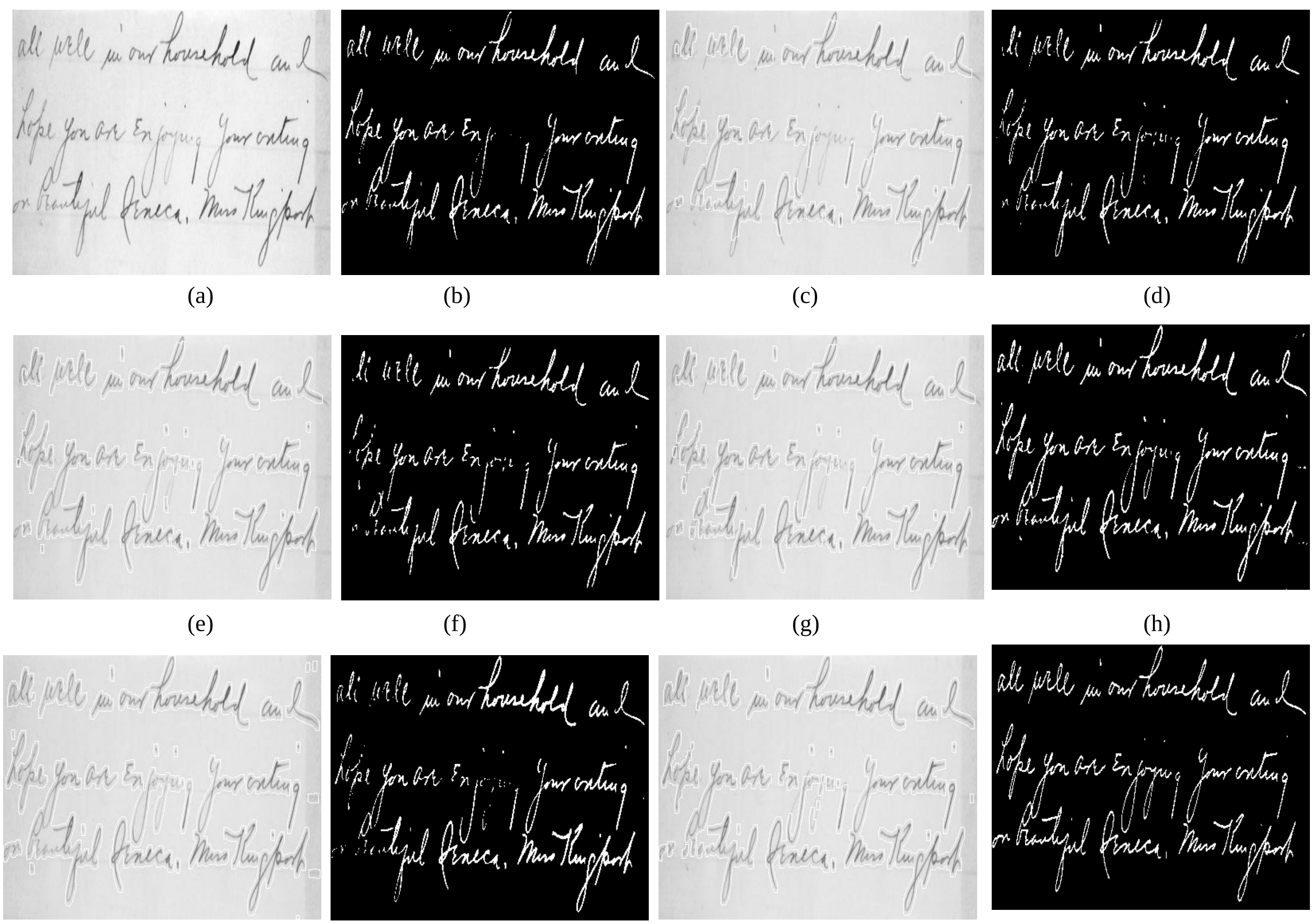


(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.83)

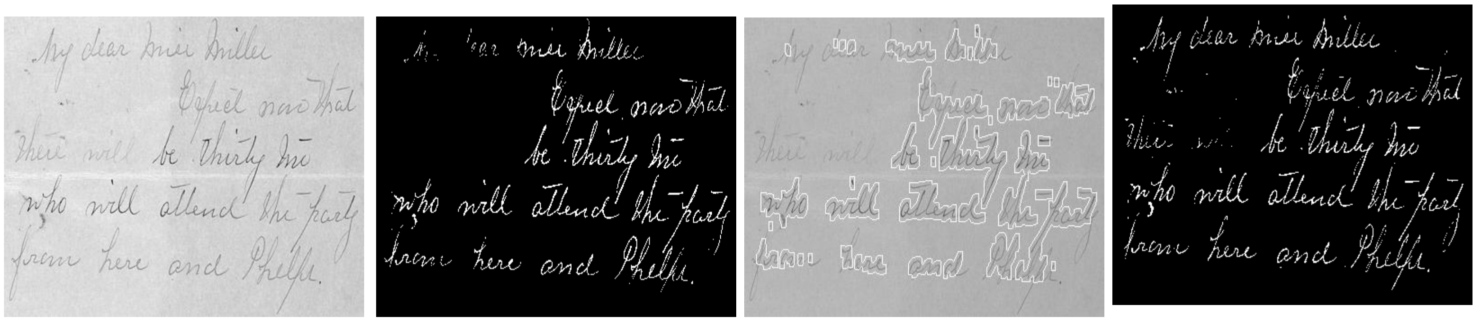


(a) (b) (c) (d)

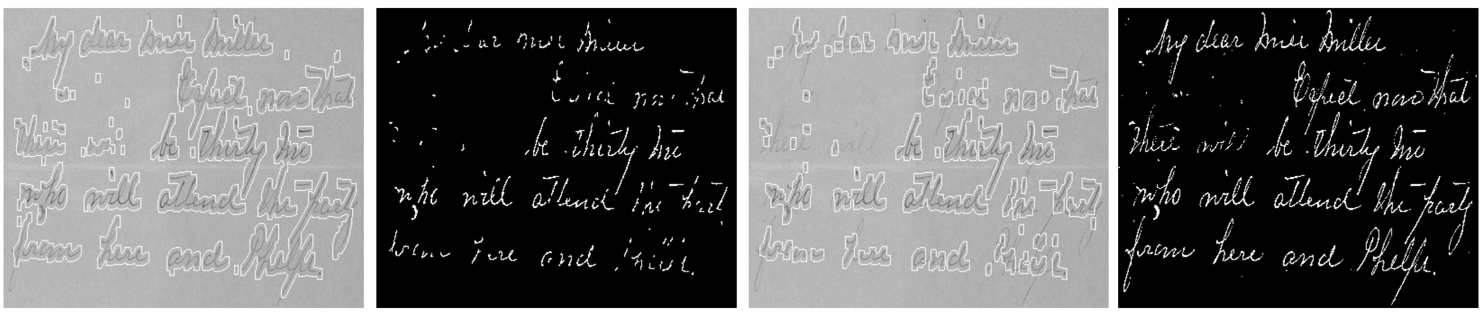
(e) (f) (g) (h)

(i) (j) (k) (l)

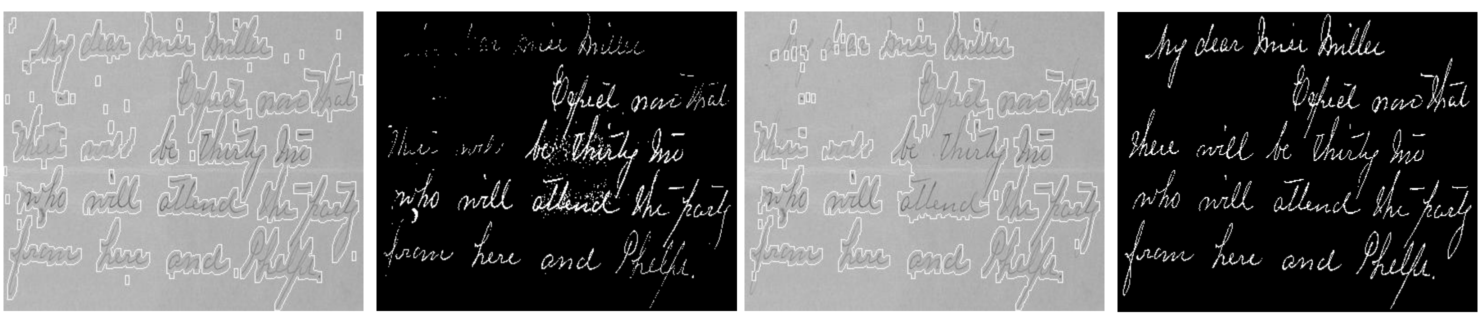
Fig (5.84)



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

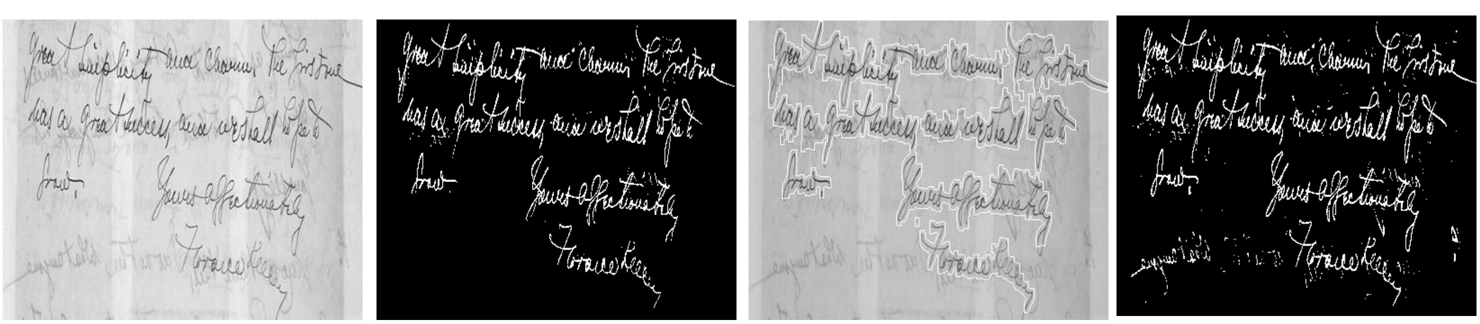


(e) (f) (g) (h)

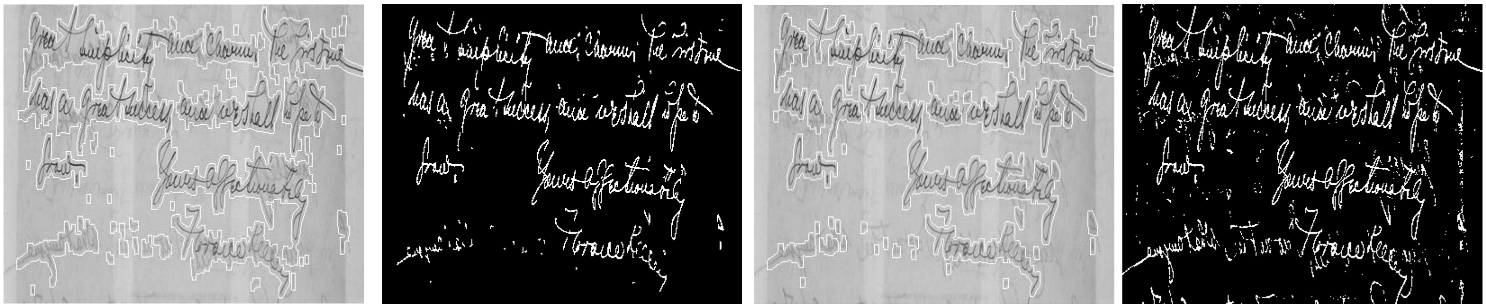


(i) (j) (k) (l)

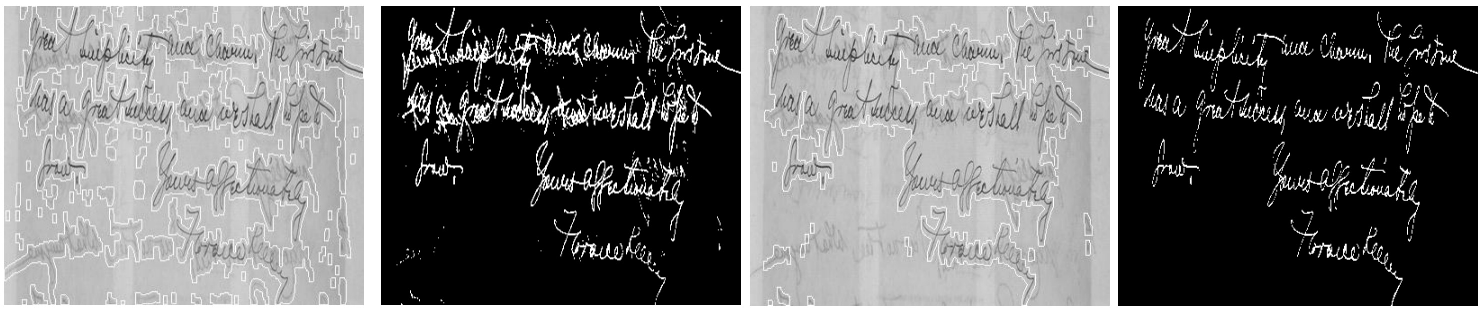
Fig (5.85)



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



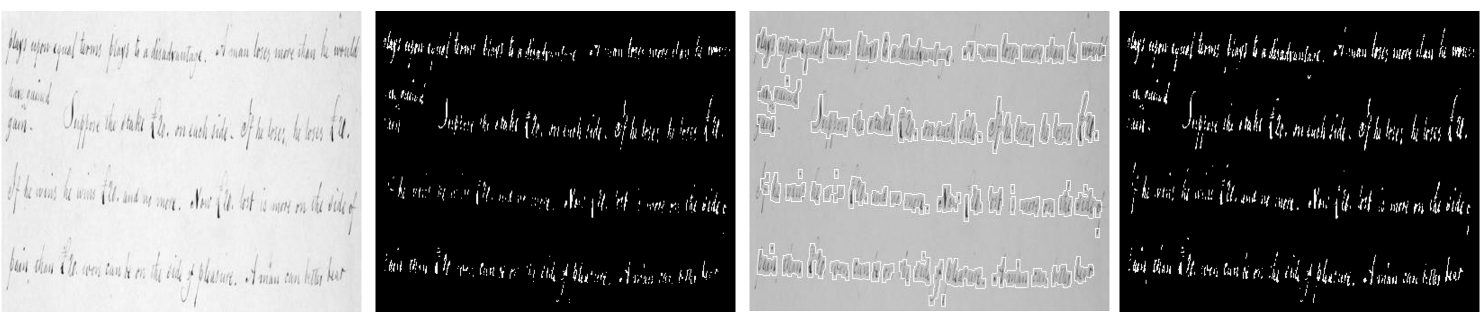
(e) (f) (g) (h)



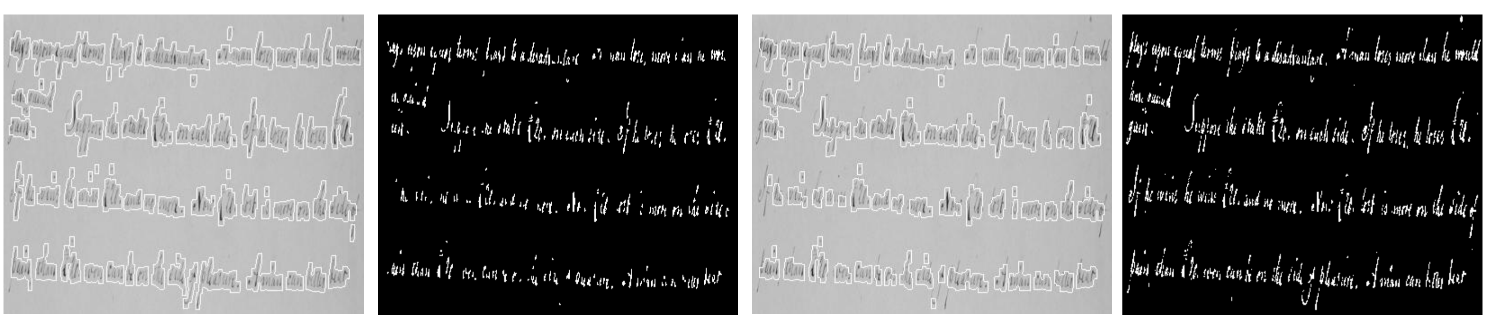
(i) (j) (k) (l)

Fig (5.86)

5.2.2.5 DIBCO-2013 Dataset Results



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



(e) (f) (g) (h)

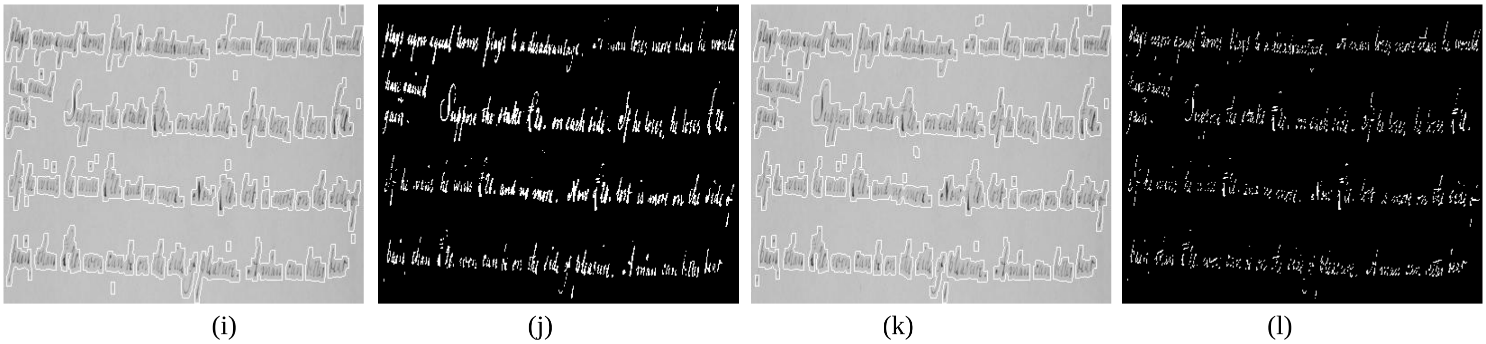
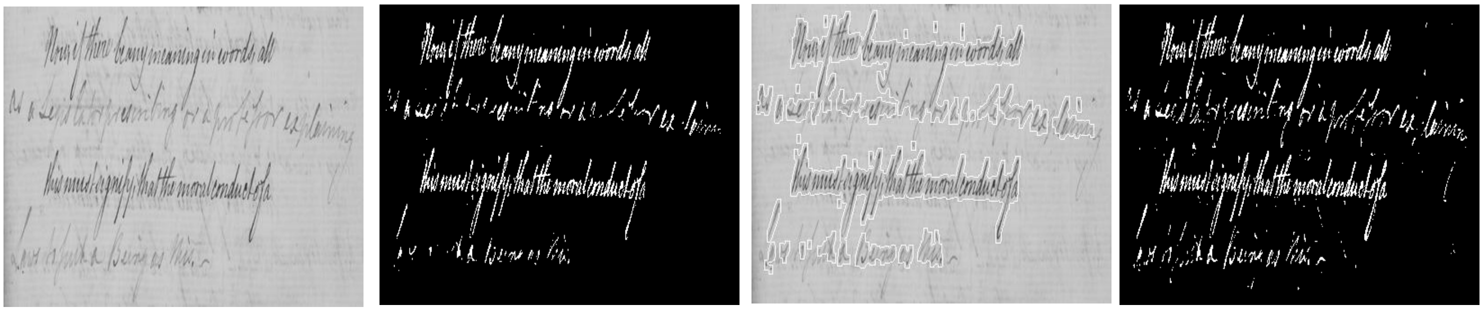
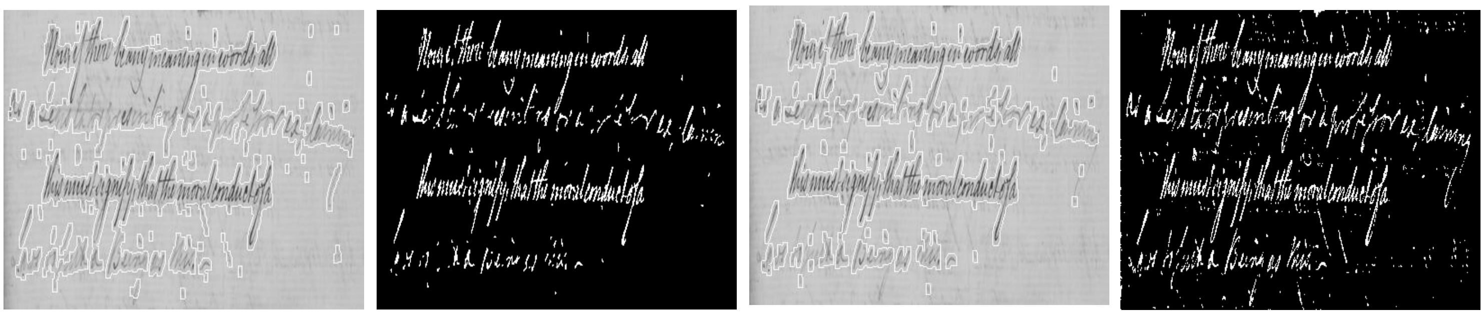


Fig (5.87)

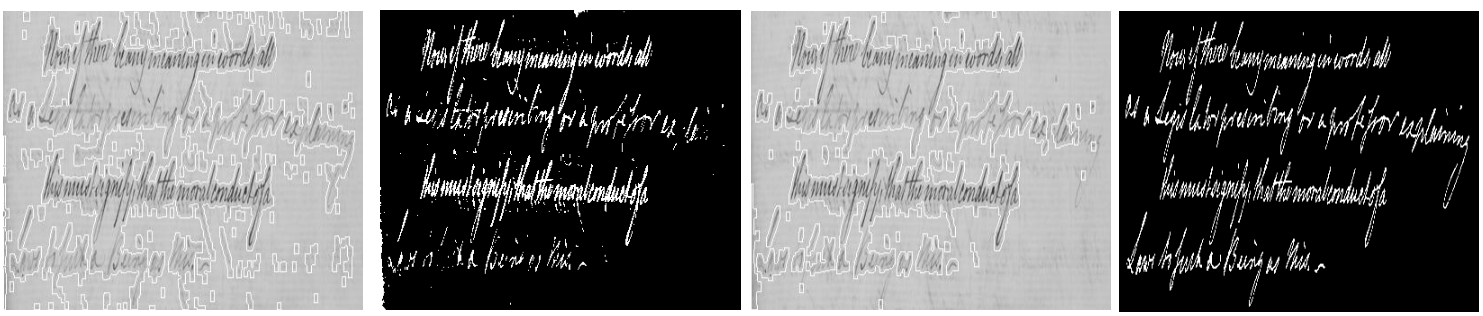
Fig (5.88)



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

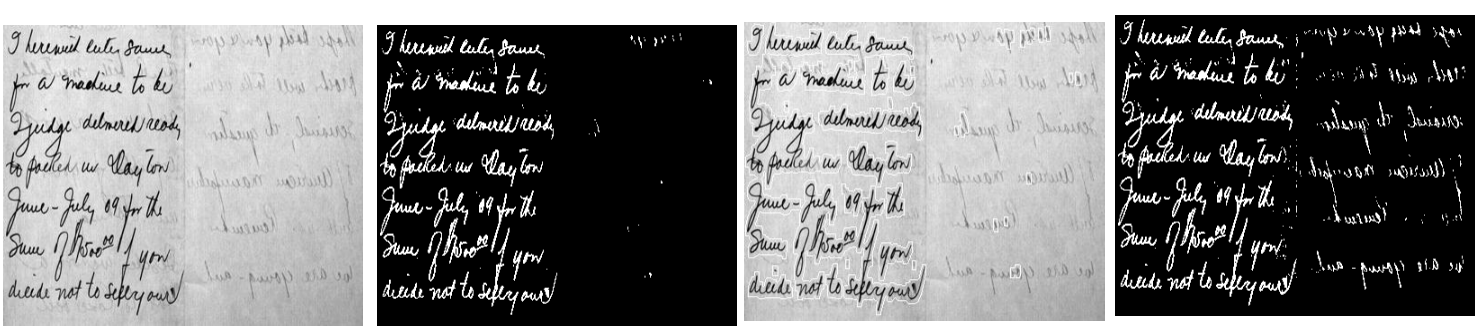


(e) (f) (g) (h)

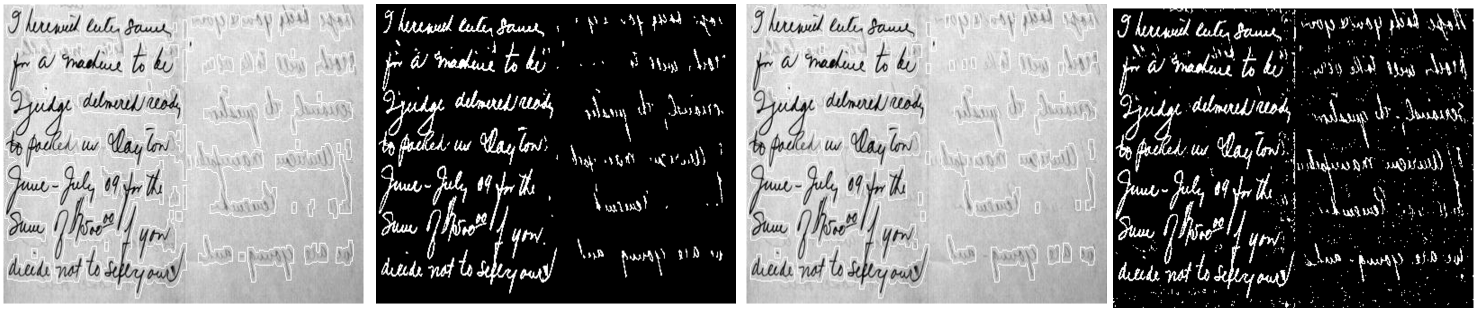


(i) (j) (k) (l)

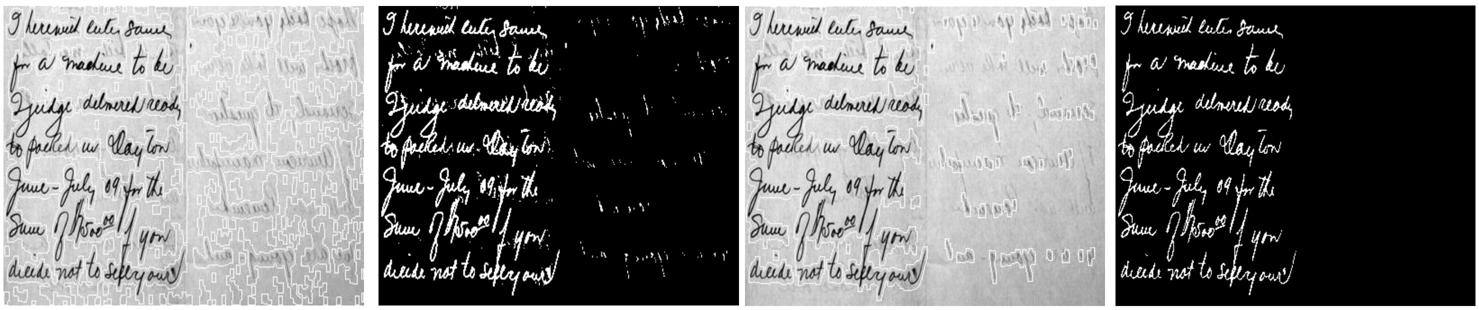
Fig (5.89)



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

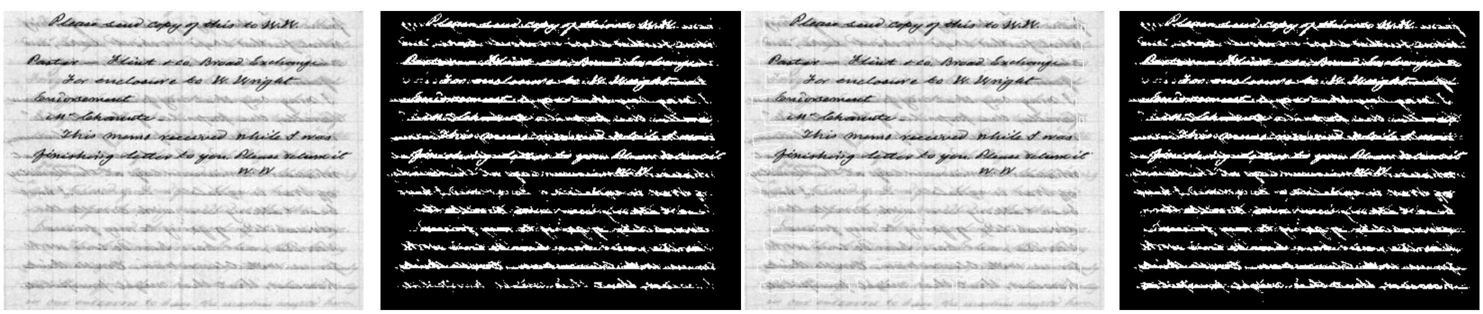


(e) (f) (g) (h)

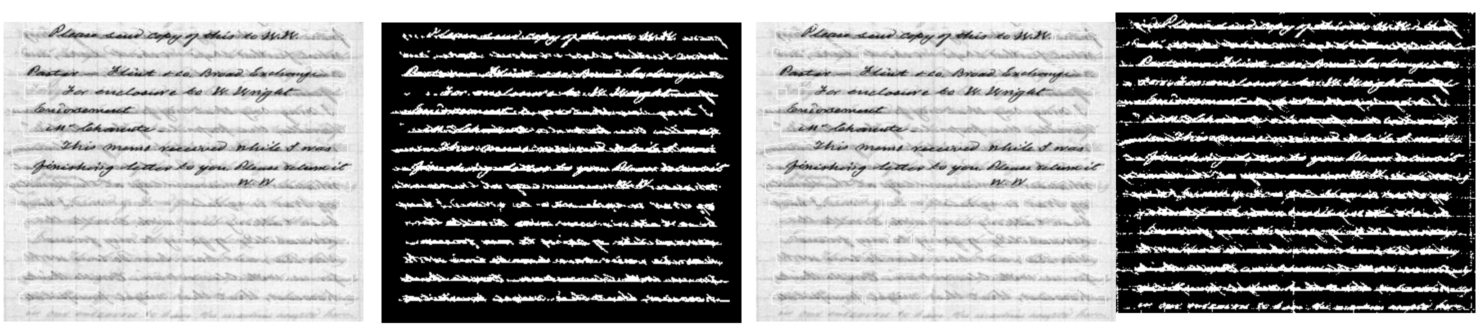


(i) (j) (k) (l)

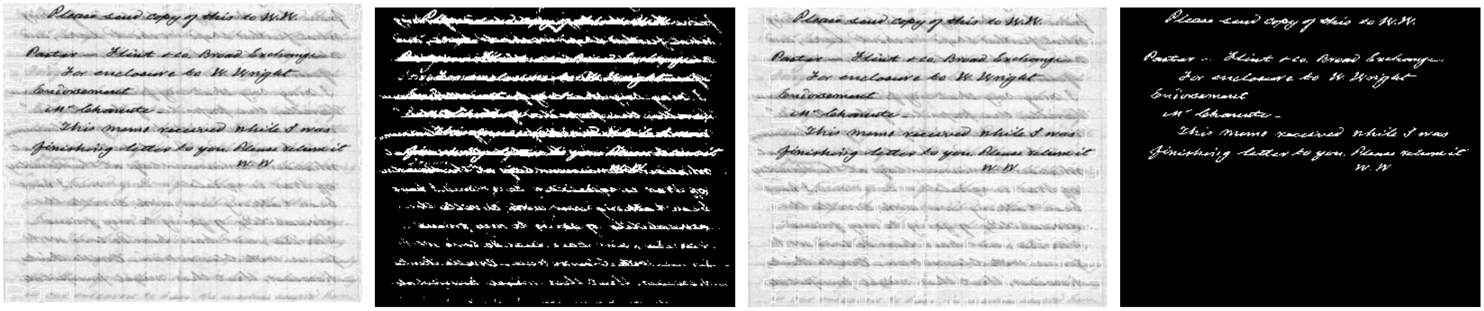
Fig (5.90)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

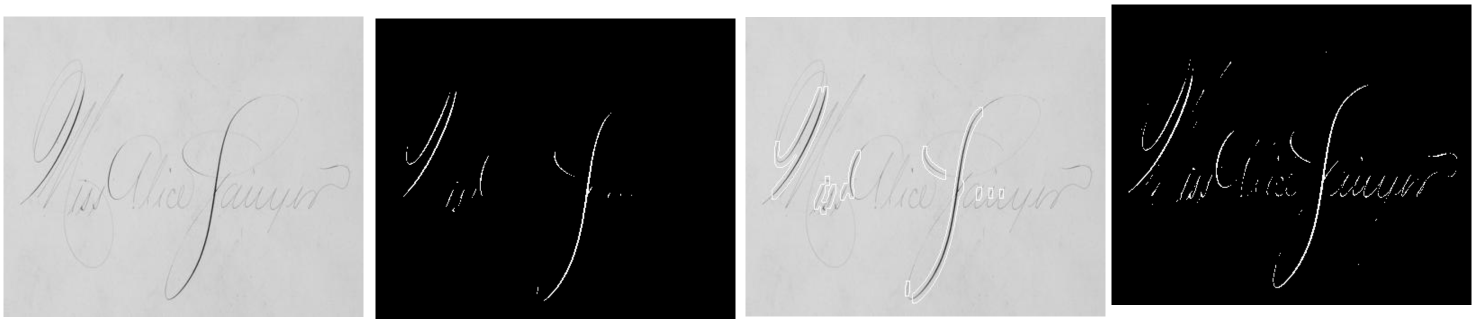
Fig (5.91)

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

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig (5.92)

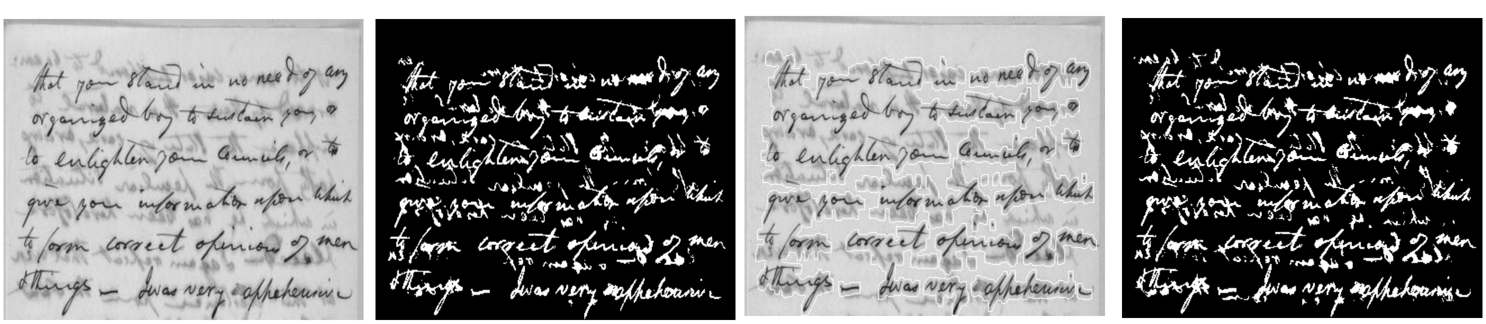


(a) (b) (c) (d)

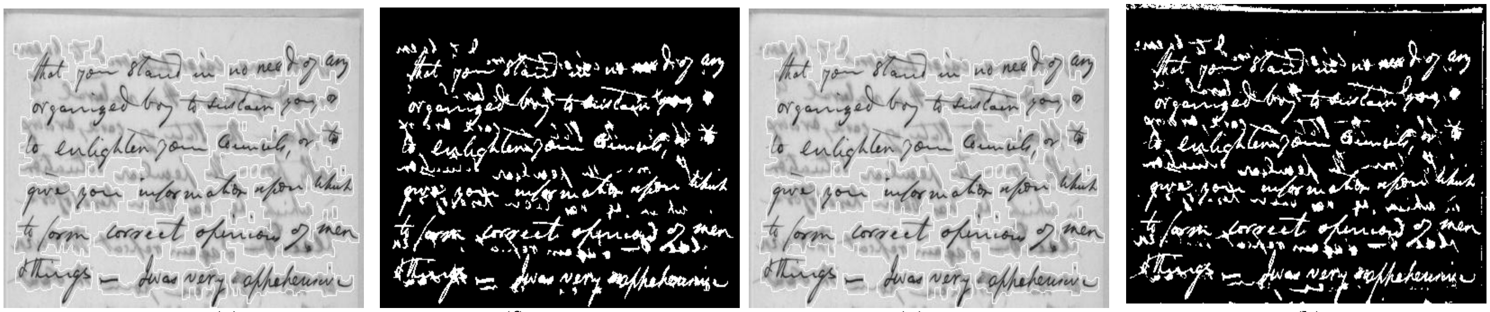
(e) (f) (g) (h)

(i) (j) (k) (l)

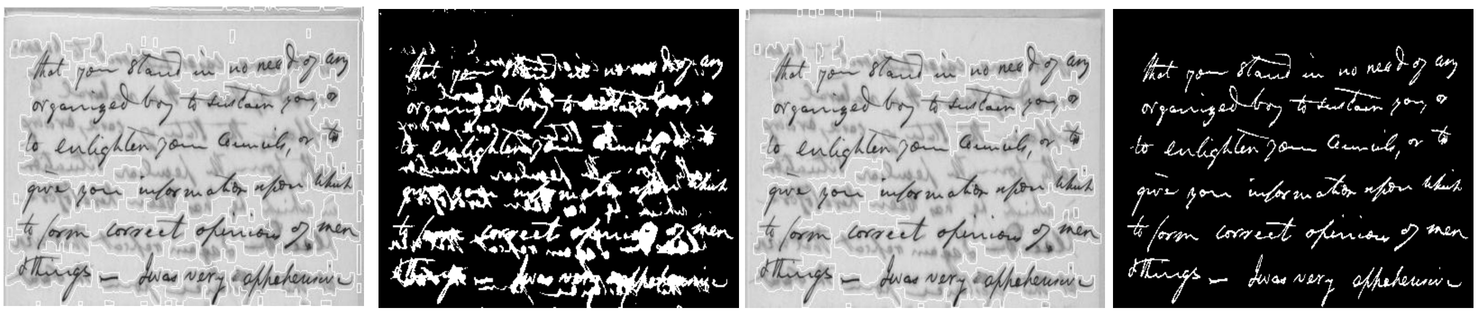
Fig (5.93)



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

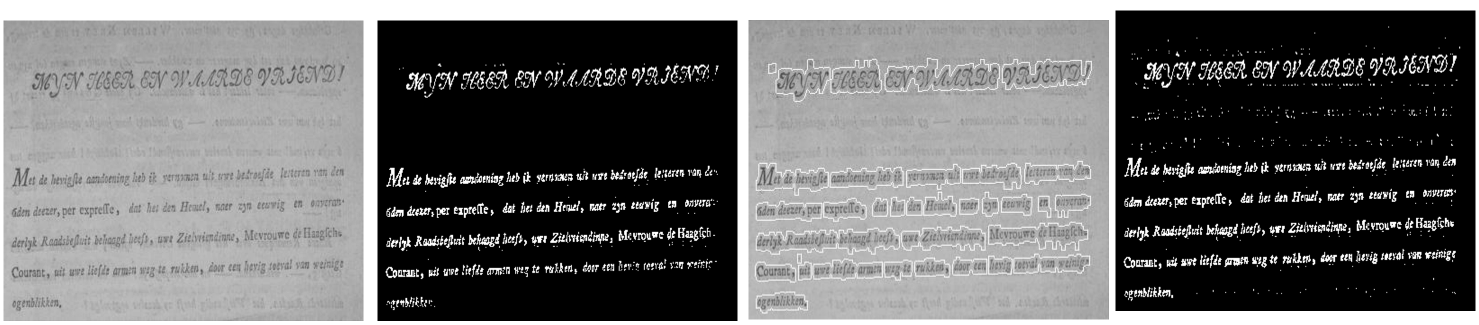


(e) (f) (g) (h)

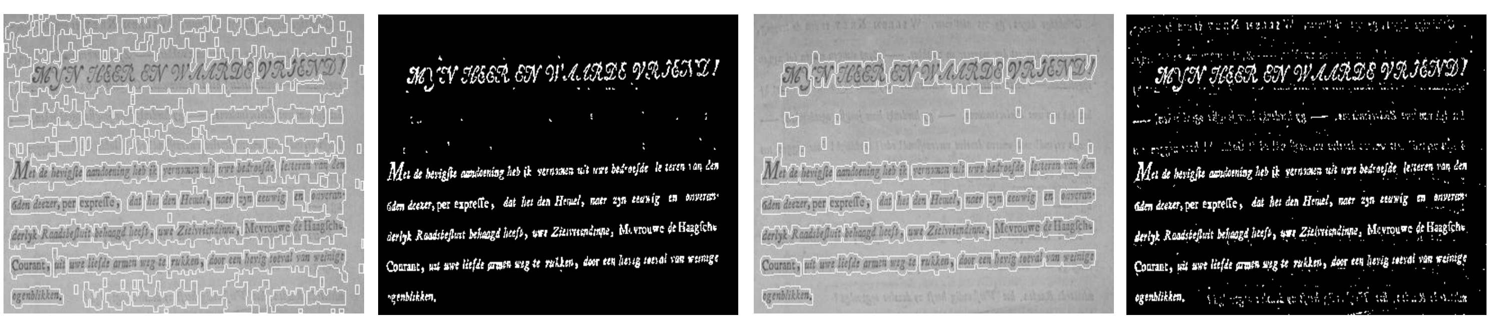


(i) (j) (k) (l)

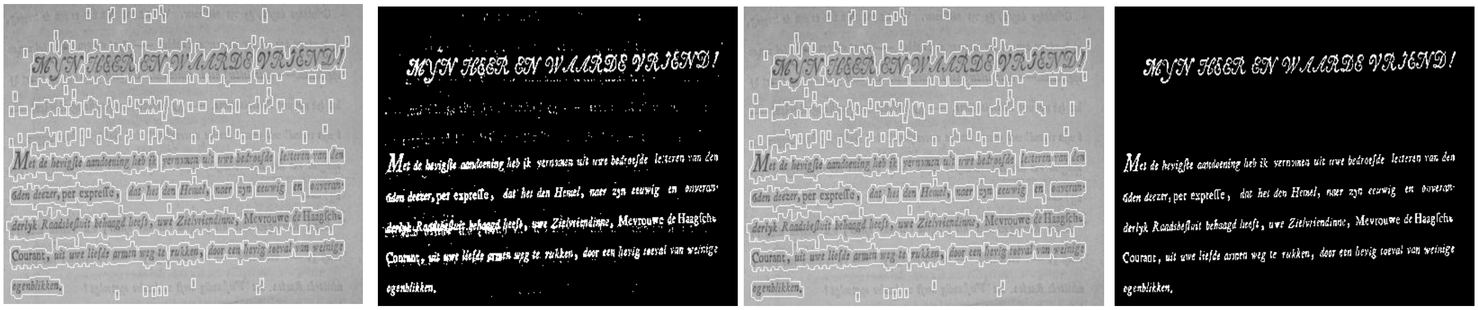
Fig (5.94)



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

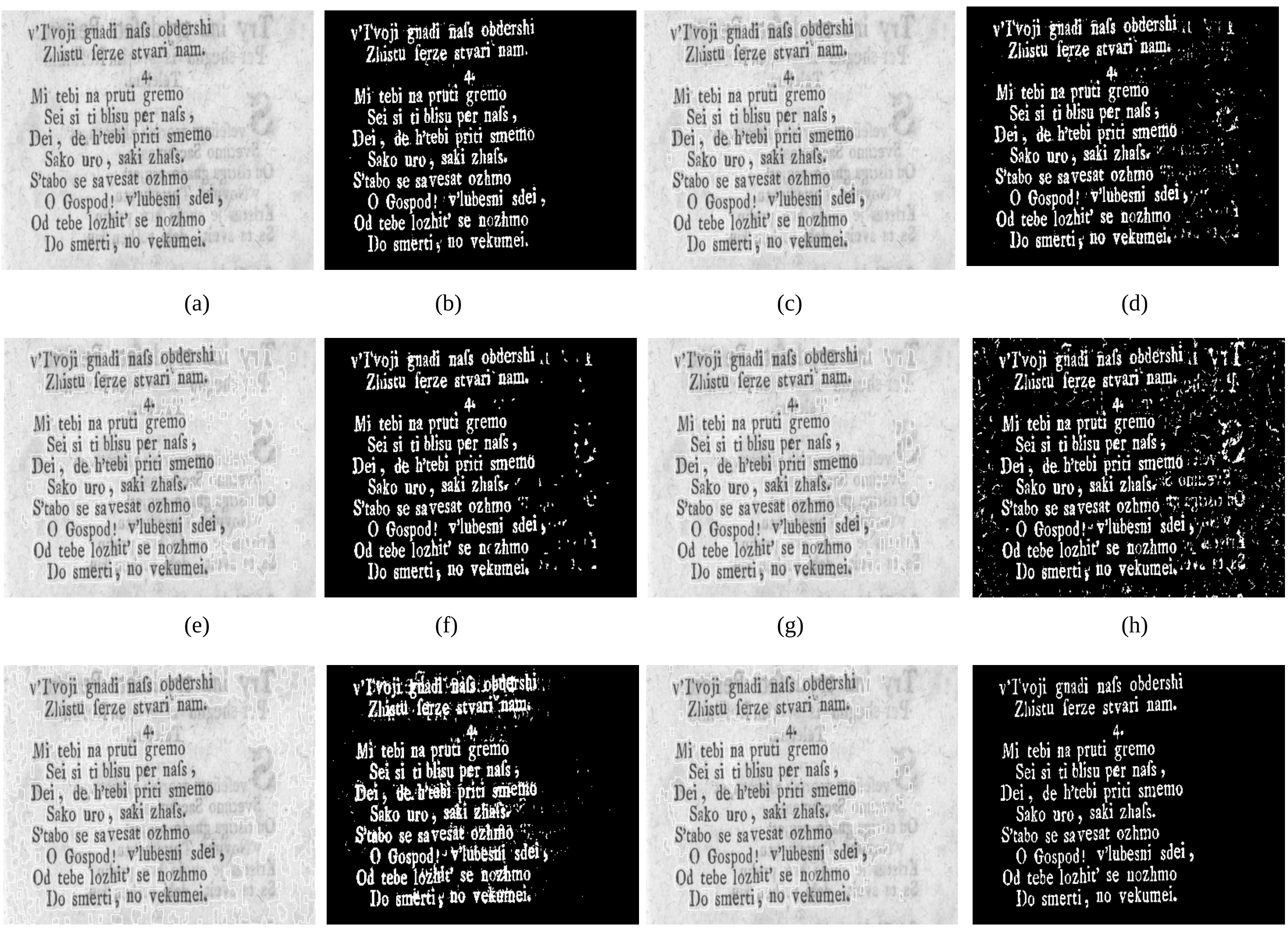


(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.95)

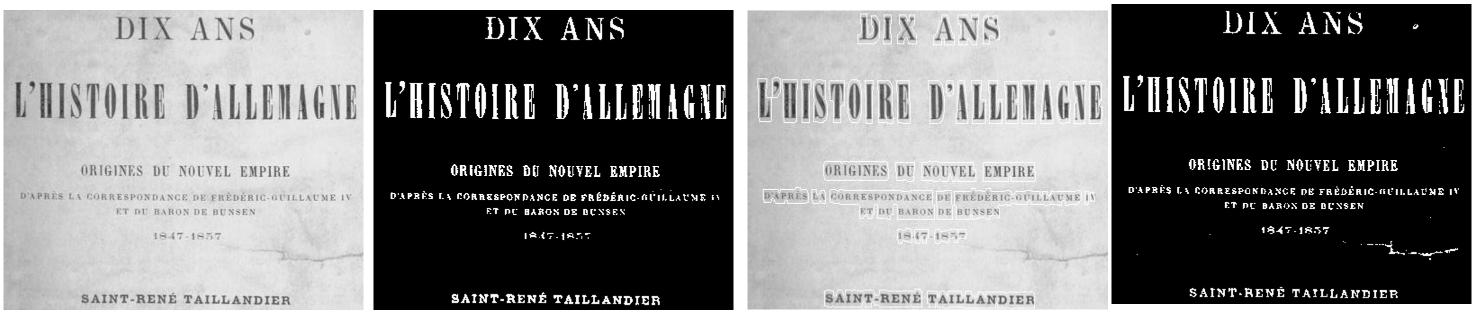


(a) (b) (c) (d)

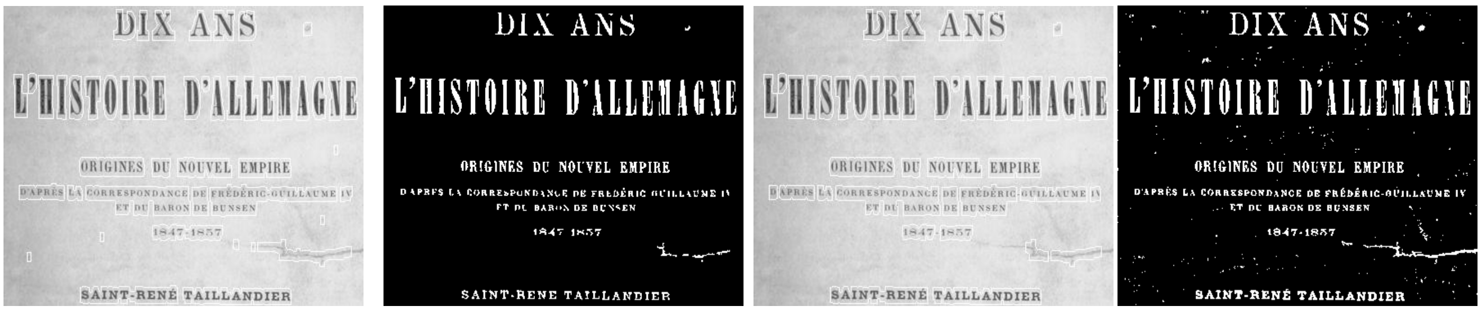
(e) (f) (g) (h)

(i) (j) (k) (l)

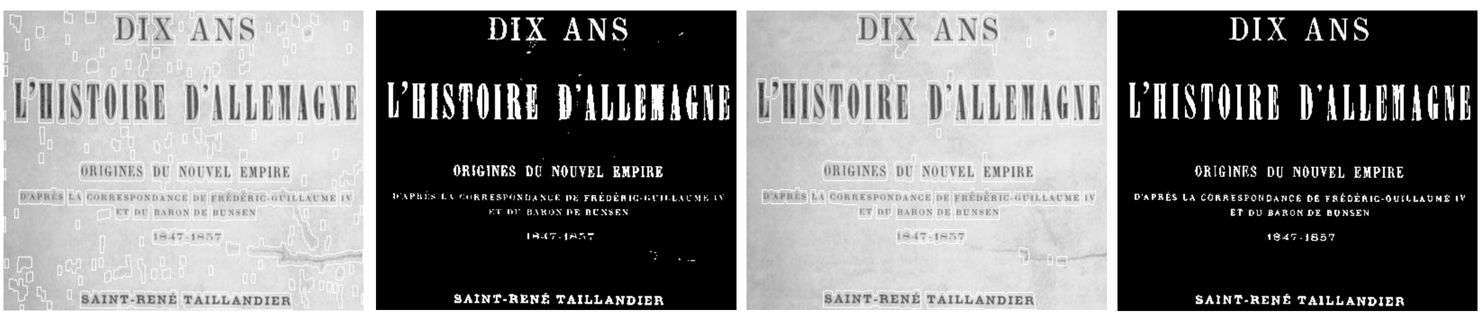
Fig (5.96)



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

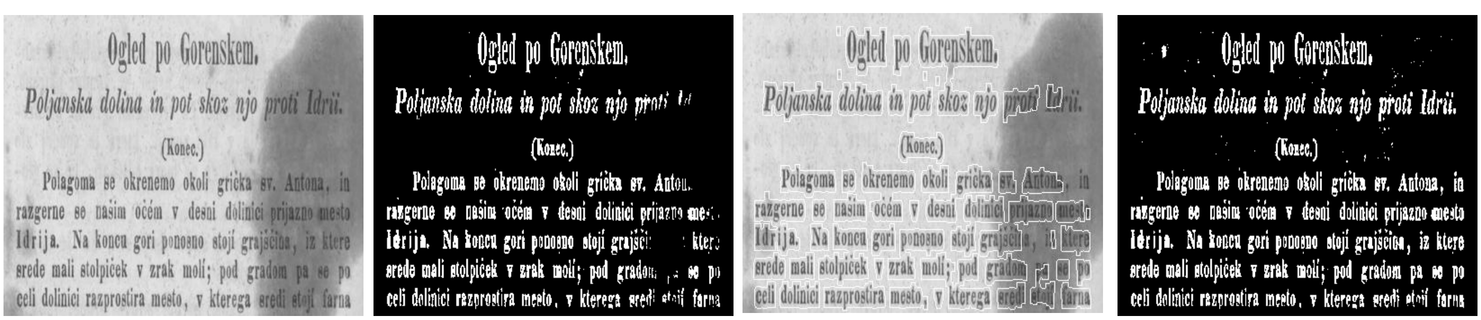


(e) (f) (g) (h)

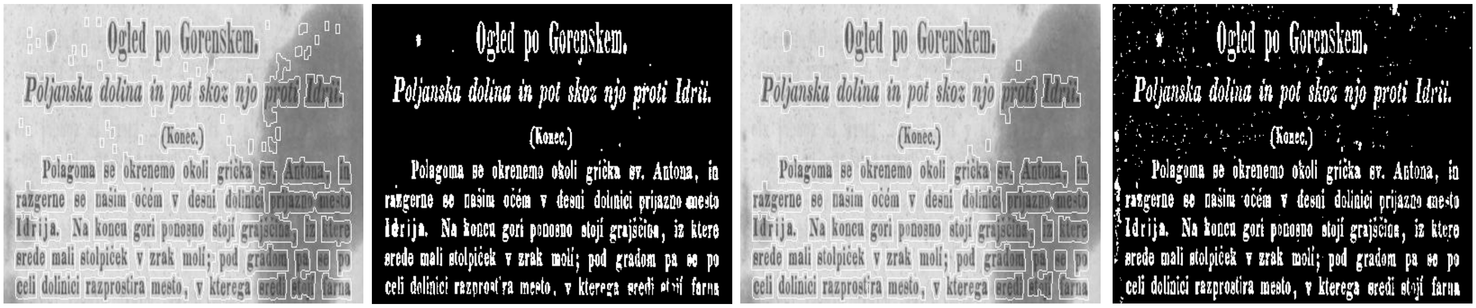


(i) (j) (k) (l)

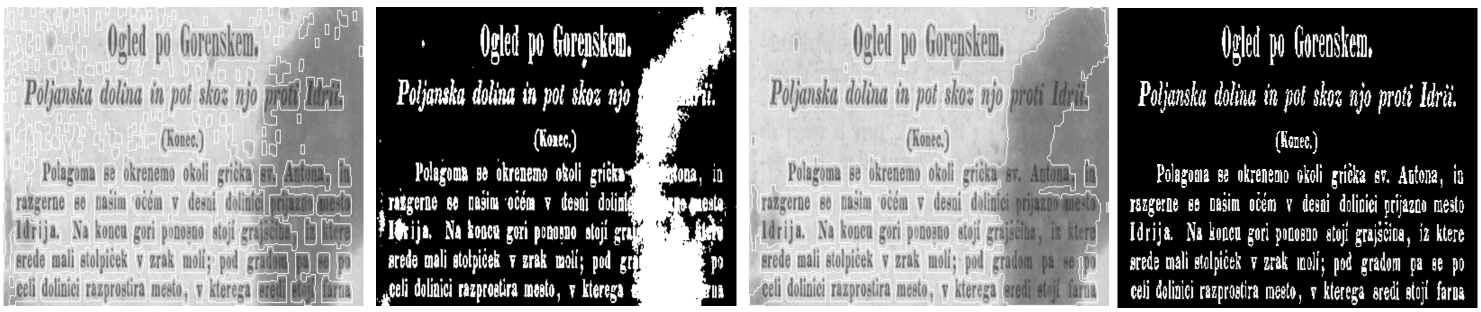
Fig (5.97)



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

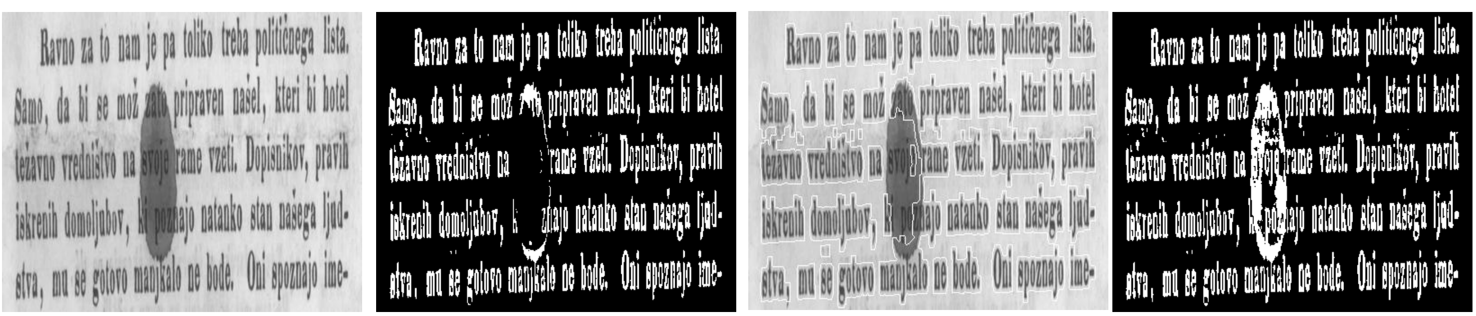


(e) (f) (g) (h)

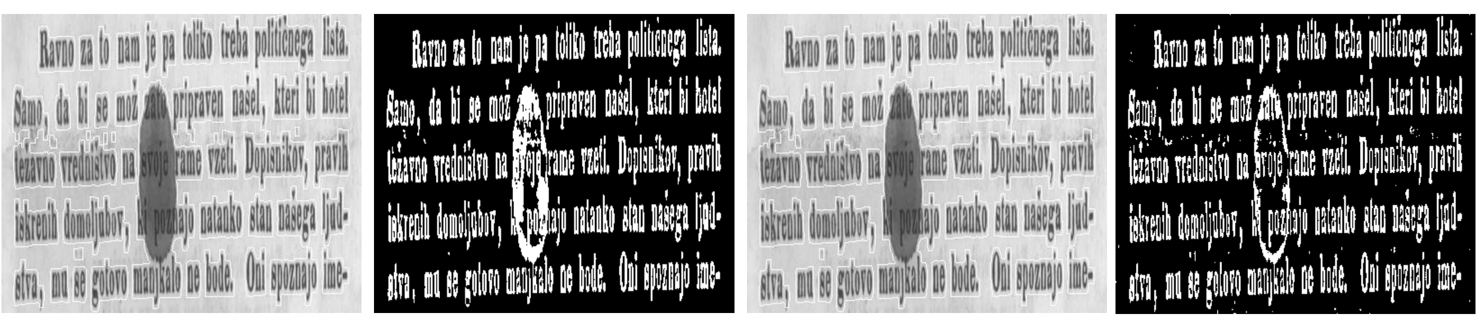


(i) (j) (k) (l)

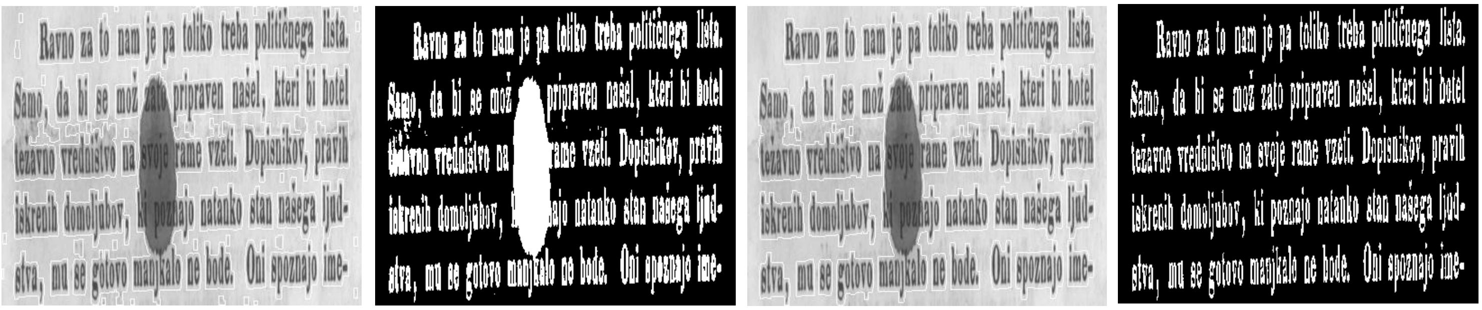
Fig (5.98)



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

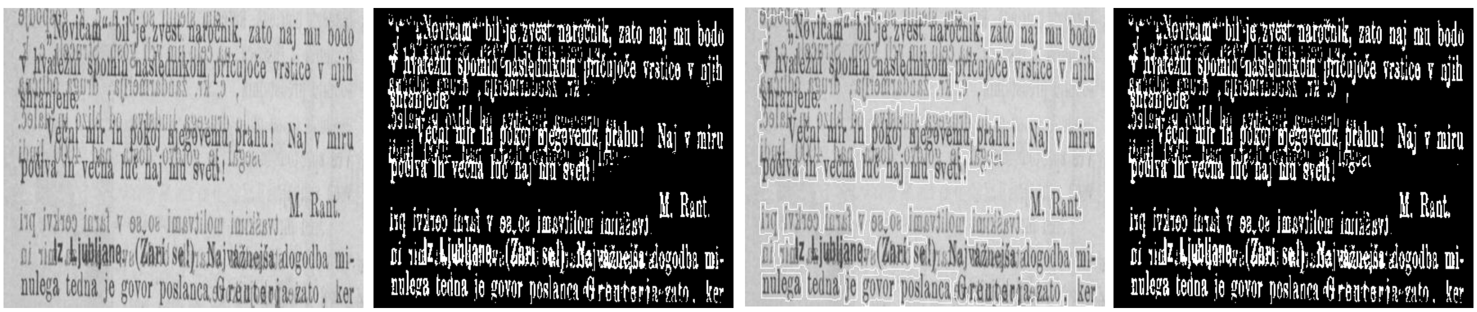


(e) (f) (g) (h)

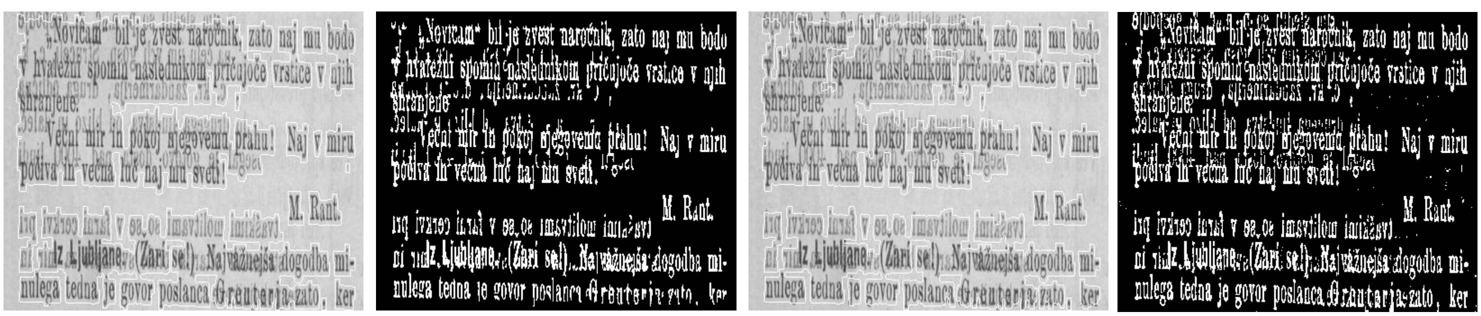


(i) (j) (k) (l)

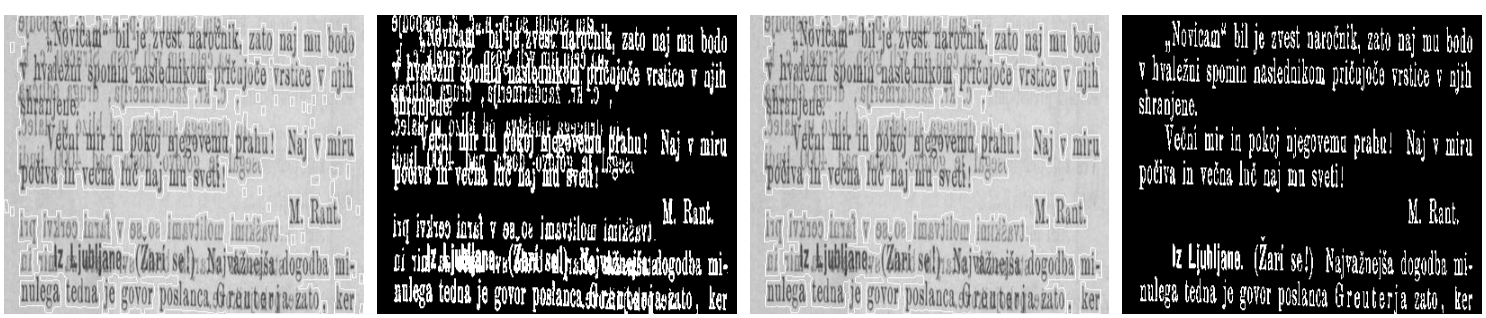
Fig (5.99)



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

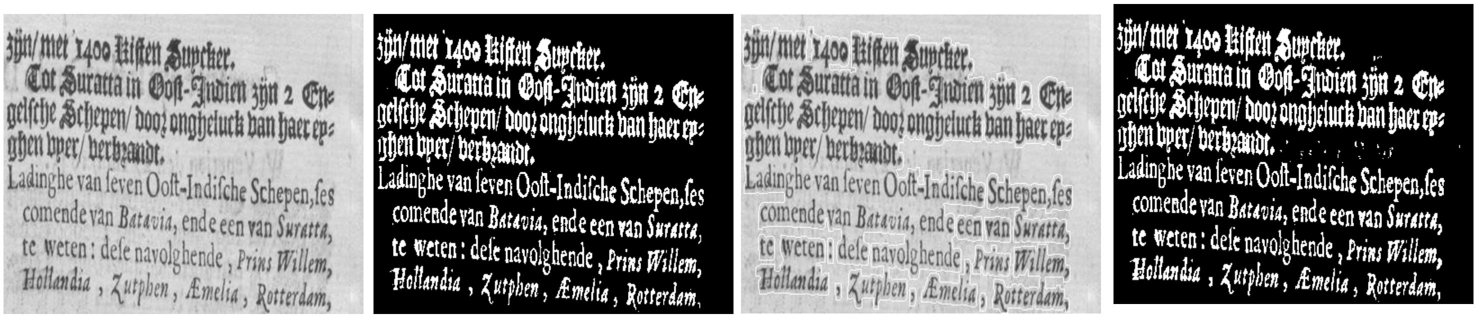


(e) (f) (g) (h)

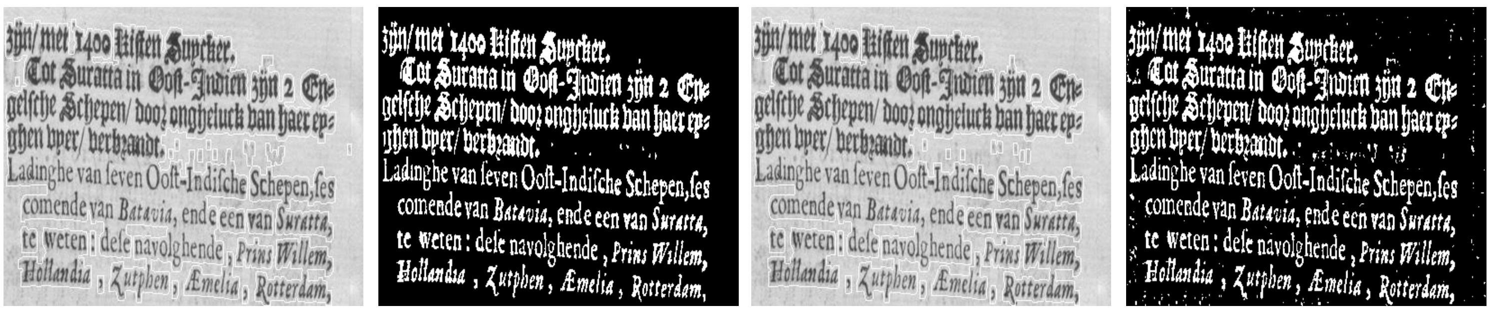


(i) (j) (k) (l)

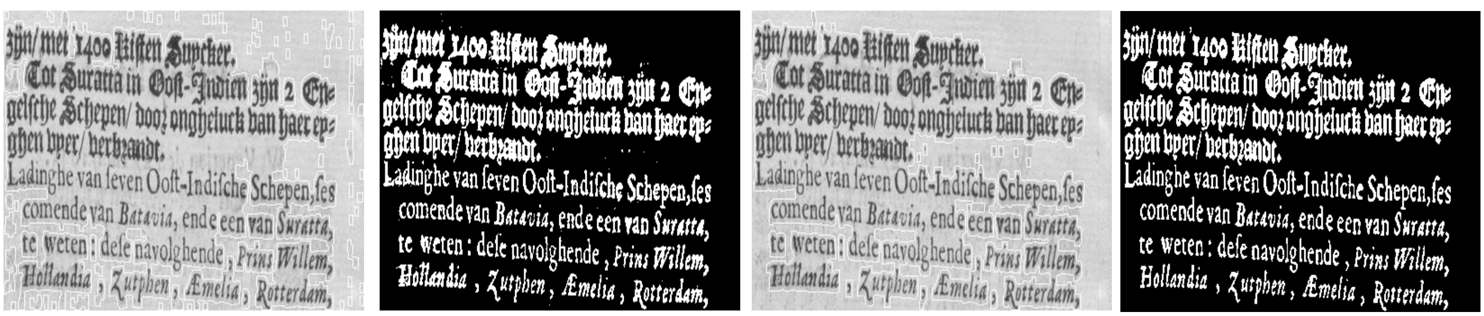
Fig (5.100)



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

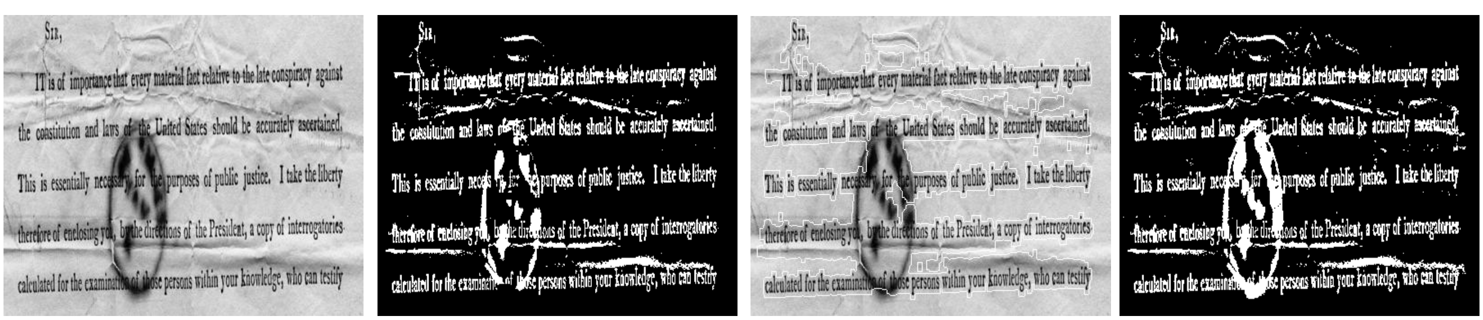


(e) (f) (g) (h)

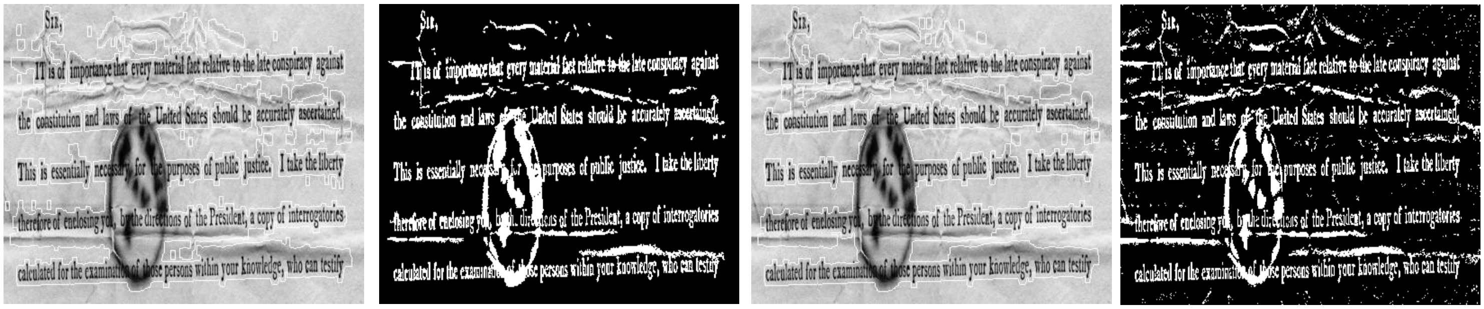


(i) (j) (k) (l)

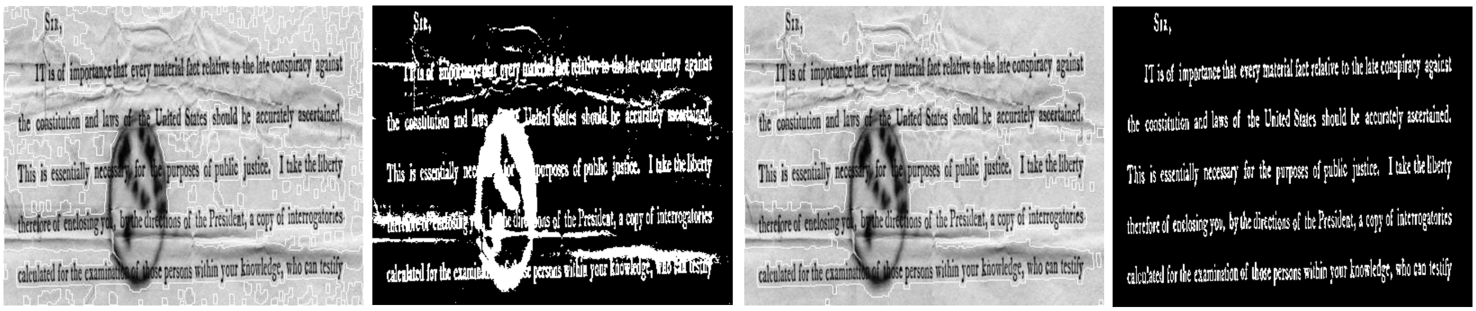
Fig (5.101)



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



(e) (f) (g) (h)



(i) (j) (k) (l)

Fig (5.102)

In above figures the following parameters are used-

(a) - Original Image, (b) - Binary Image using Proposed Algorithm, (c) - Text Segment Image using Proposed Algorithm,

(d) - Binary Image using Yanowitz's Method, (e) - Text Segment Image using Yanowitz's Method,

(f) - Binary Image using Post Yanowitz's Method, (g) - Text Segment Image using Post Yanowitz's Method,

(h) - Binary Image using Niblack's Method, (i) - Text Segment Image using Niblack's Method,

(j) - Binary Image using Sauvola's Method, (k) - Text Segment Image using Sauvola's Method.

(l) – Ground Truth (GT) image.

There are 66 images that describe above with experimental results are taken from five well-known competition dataset DIBCO datasets.

According to visualization out of 66 printed and handwritten results 53 results produced by our method are very good.

5.3 Performance Analysis

According to the evaluation results are shown in below Table I, II, III, IV and V propose algorithm achieves the highest scores in F-Measure and in Precision.

Table I: Evaluation Results of DIBCO-2009

N	P1	R1	FM1	P2	R2	FM2	P3	R3	FM3	P4	R4	FM4	P5	R5	FM5
1	84.46	90.02	87.15	86.95	88.85	87.89	84.38	91.73	87.90	74.76	95.02	83.68	86.67	85.02	85.84
2	42.94	99.45	59.98	26.26	99.56	41.56	17.49	99.04	29.73	25.39	99.89	40.49	28.89	99.30	44.76
3	79.90	91.08	85.13	72.29	95.55	82.31	68.87	93.33	79.26	51.93	99.81	68.32	72.42	95.48	82.37
4	66.89	83.16	74.14	48.02	98.28	64.51	50.07	96.70	65.98	27.09	99.63	42.60	48.36	98.23	64.81
5	54.92	61.14	57.86	53.07	95.90	68.33	60.86	93.90	73.86	18.82	94.94	31.41	53.32	95.09	68.33
6	82.52	93.34	87.60	76.37	95.24	84.77	71.28	90.27	79.66	58.58	99.69	73.80	77.68	95.02	85.48
7	93.42	92.92	93.17	90.27	95.14	92.64	86.12	91.99	88.96	83.97	99.62	91.13	90.93	95.09	92.96
8	96.61	94.51	95.55	95.62	93.46	94.53	86.13	89.86	87.96	92.20	98.94	95.45	97.20	93.43	95.28
9	79.68	94.66	86.52	69.67	98.19	81.50	73.41	95.24	82.91	59.44	99.83	74.52	70.10	98.15	81.79
10	78.10	93.56	85.13	75.26	94.03	83.60	70.94	88.97	78.94	61.38	99.60	75.96	76.64	93.83	84.37
Avg.	75.94	89.38	81.22	69.37	95.42	78.16	66.955	93.10	75.51	55.35	98.69	67.73	70.22	94.86	78.59

Table II: Evaluation Results of DIBCO-2010.

N	P1	R1	FM1	P2	R2	FM2	P3	R3	FM3	P4	R4	FM4	P5	R5	FM5
1	96.01	76.82	85.35	98.26	70.15	81.86	94.13	83.30	88.39	90.38	87.03	88.67	98.41	68.14	80.52
2	88.33	90.54	89.42	89.10	88.11	88.60	78.99	93.42	85.60	77.56	89.09	82.90	89.69	85.60	87.60
3	92.63	84.63	88.45	91.39	82.81	86.89	70.94	87.68	78.43	81.57	90.94	86.00	95.04	80.34	87.07
4	82.01	94.37	87.76	84.98	92.00	88.35	82.53	94.63	88.17	82.52	87.46	84.92	84.91	91.20	87.94
5	54.82	98.75	70.50	41.67	99.17	58.68	36.97	98.92	53.83	30.53	99.96	46.77	42.48	99.04	59.45
6	87.23	84.45	85.82	87.73	80.85	84.15	80.78	87.10	83.82	76.41	90.12	82.70	87.42	75.93	81.27
7	81.62	93.07	86.97	78.12	94.51	85.54	61.69	96.74	75.34	71.11	97.89	82.38	78.47	93.66	85.40
8	85.94	55.87	67.72	83.35	74.83	78.86	76.96	86.17	81.30	73.95	84.36	78.82	82.73	68.00	74.65
9	83.39	88.95	86.08	87.13	84.79	85.94	82.73	92.71	87.44	79.32	85.21	82.16	85.36	71.65	77.91
10	80.76	80.29	80.52	85.40	76.07	80.47	83.94	79.11	81.45	64.94	81.79	72.40	78.66	85.03	73.17
Avg.	83.27	84.77	82.85	82.71	84.32	81.93	74.96	89.97	80.37	72.82	89.38	78.77	82.31	81.85	79.49

Table III: Evaluation Results of DIBCO-2011.

N	P1	R1	FM1	P2	R2	FM2	P3	R3	FM3	P4	R4	FM4	P5	R5	FM5
1	80.04	85.40	82.63	71.24	95.85	81.73	64.01	96.07	76.83	59.52	98.04	74.32	76.88	95.85	85.32
2	87.99	89.45	88.71	84.29	94.90	89.28	74.35	96.38	83.94	71.34	96.83	82.15	84.32	91.96	87.97
3	82.25	82.12	82.18	86.81	79.02	82.73	80.26	86.19	83.12	58.10	91.80	71.17	86.63	74.50	80.11
4	74.39	86.83	80.13	62.09	89.89	73.45	62.08	89.35	73.26	41.03	92.93	56.93	63.54	88.21	73.87
5	83.46	93.04	87.99	80.44	94.87	87.06	72.61	93.62	81.79	64.05	99.30	77.87	81.18	94.68	87.41
6	79.17	75.46	77.27	66.30	86.45	75.05	62.09	88.87	73.10	48.18	82.52	60.84	66.43	83.52	74.01
7	62.21	93.93	74.85	37.56	96.16	54.02	29.47	96.06	45.10	31.83	97.77	48.03	38.96	94.34	55.14
8	85.66	91.42	88.45	87.43	91.80	89.56	88.32	88.14	88.23	79.20	96.42	86.92	87.72	89.14	88.42
9	94.58	88.37	91.37	78.98	89.49	83.91	67.50	86.14	75.69	78.14	99.42	87.50	81.44	89.41	85.24
10	68.64	94.41	79.49	65.56	95.03	77.59	60.80	89.80	72.51	49.30	99.50	65.93	66.65	94.89	78.30
11	94.42	90.03	92.17	89.74	91.08	90.41	79.95	87.91	83.74	81.18	98.35	88.95	91.16	91.04	91.10
12	90.59	95.63	93.04	75.74	96.90	85.03	68.26	95.15	79.49	60.97	99.90	75.73	78.66	96.61	86.72

13	75.54	92.00	82.96	67.40	93.51	78.34	70.60	93.68	80.52	56.46	99.54	72.05	67.75	93.29	78.50
14	77.06	96.18	85.56	32.14	98.63	48.48	32.49	95.49	48.48	25.23	99.79	40.27	36.80	98.38	53.57
15	89.21	88.35	88.78	68.61	95.55	79.87	44.41	91.37	59.76	49.90	98.77	66.30	82.22	95.55	88.39
16	91.57	81.14	86.04	93.15	79.04	85.52	88.70	82.00	85.22	84.57	86.29	85.42	93.08	77.07	84.32
Avg.	82.30	88.98	85.10	71.71	91.76	78.87	65.36	91.01	74.42	58.68	96.07	71.27	73.96	90.52	79.89

Table IV: Evaluation Results of DIBCO-2012.

N	P1	R1	FM1	P2	R2	FM2	P3	R3	FM3	P4	R4	FM4	P5	R5	FM5
1	71.43	93.47	81.08	71.37	93.74	81.04	60.18	93.71	73.29	51.44	98.94	67.69	70.22	86.96	77.70
2	79.47	92.05	85.30	82.35	90.43	86.20	77.83	93.95	85.13	71.17	88.16	78.76	82.57	88.68	85.51
3	81.12	94.40	87.26	81.87	92.01	86.64	80.95	96.86	88.19	59.54	98.64	74.26	81.80	91.22	86.25
4	86.03	91.77	88.81	81.18	93.95	87.10	79.89	92.91	85.91	74.70	97.93	84.75	81.16	93.29	86.80
5	63.06	96.73	76.35	51.67	98.96	67.89	43.70	99.41	60.72	40.00	99.81	57.11	53.63	98.81	69.53
6	48.71	98.42	65.17	43.86	99.06	60.80	40.51	99.40	57.56	35.18	99.57	51.99	44.24	98.71	61.10
7	81.28	87.77	84.40	82.04	86.47	84.20	79.70	87.00	83.19	73.63	94.56	82.79	81.76	84.64	83.17
8	68.61	98.81	80.99	59.78	99.89	74.80	65.47	99.68	79.03	40.89	100.0	58.05	59.91	99.89	74.90
9	95.98	90.36	93.09	96.78	88.52	92.47	95.16	89.55	92.27	91.68	95.47	93.69	96.79	86.98	91.63
10	89.42	93.94	91.63	95.33	87.49	91.24	92.17	90.28	91.22	89.40	94.47	91.86	95.77	83.14	89.01
11	90.95	87.91	89.40	95.78	82.91	88.88	91.87	88.12	89.95	84.91	89.52	87.15	96.60	78.61	86.68
12	84.98	83.08	84.02	87.17	82.28	84.66	81.04	93.10	86.65	78.74	83.53	81.07	86.66	78.22	82.22
13	86.77	68.88	76.80	90.91	71.00	79.73	86.43	86.93	86.68	75.43	69.14	72.15	88.05	46.16	60.57
14	73.97	95.45	83.35	64.35	98.66	77.89	52.53	99.21	68.69	49.33	99.04	65.85	68.20	93.68	78.93
Avg.	78.70	90.93	83.40	77.46	90.38	81.68	73.38	93.57	80.60	65.43	93.48	74.79	77.66	86.35	79.57

Table V: Evaluation Results of DIBCO-2013.

N	P1	R1	FM1	P2	R2	FM2	P3	R3	FM3	P4	R4	FM4	P5	R5	FM5
1	66.47	67.76	67.11	58.10	83.49	68.52	50.41	91.45	64.99	44.81	94.23	60.73	58.87	71.05	64.39
2	90.25	89.14	89.70	81.63	94.11	87.42	76.34	95.01	84.66	68.07	98.93	80.65	83.68	92.31	87.78
3	78.47	77.04	77.75	81.32	76.64	78.91	72.33	84.03	77.75	69.85	80.75	74.90	81.27	68.31	74.23
4	73.87	99.78	84.89	54.77	99.85	70.74	46.04	99.97	63.05	59.97	100	74.97	57.11	99.62	72.60
5	19.93	99.82	33.23	18.45	99.90	31.15	13.97	100	24.52	19.50	100	32.63	19.06	99.13	31.97
6	58.52	98.35	73.38	58.47	98.36	73.34	50.32	99.41	66.82	47.06	99.76	63.95	59.10	95.26	72.95
7	83.05	23.72	36.90	82.40	44.04	57.40	80.06	71.82	75.71	71.15	56.99	63.29	78.21	30.07	43.44
8	49.89	99.01	66.35	44.49	99.61	61.50	40.35	99.62	57.44	31.44	99.88	47.83	45.28	99.21	62.18
9	80.78	91.80	85.94	73.70	98.52	84.32	60.31	98.70	74.87	65.31	98.97	78.69	77.34	96.18	85.74
10	83.15	98.92	90.35	75.20	99.63	85.71	62.88	99.05	76.93	63.37	99.99	77.57	78.04	99.51	87.47
11	92.36	94.89	93.61	92.62	91.36	91.99	84.58	94.76	89.38	86.54	95.26	90.69	92.89	90.31	91.58
12	86.16	88.46	87.29	83.69	96.52	89.65	81.75	96.43	88.48	44.46	99.50	61.46	84.09	96.21	89.74
13	89.54	92.76	91.12	82.22	97.63	89.27	87.04	93.83	90.31	68.56	99.90	81.31	82.37	97.58	89.34
14	61.84	95.01	74.92	60.84	96.18	74.53	58.27	96.87	72.77	46.09	99.94	63.09	63.22	94.76	75.85
15	93.74	92.46	93.10	93.19	91.58	92.38	91.33	88.46	89.87	86.13	97.44	91.44	93.42	91.49	92.45
16	56.26	96.47	71.08	49.65	98.59	66.04	47.93	96.58	64.06	40.70	99.52	57.77	50.57	98.28	66.77
Avg.	72.76	87.83	76.05	68.17	91.62	75.17	62.74	94.12	72.60	57.06	95.06	68.81	66.03	88.70	74.28

In the above Tables the following notification are used to represents-

P1- Precision of our method, R1- Recall of our method, FM1- F-Measure of our method. P2- Precision of Yanowitz's method, R2- Recall of Yanowitz's method, FM2- F-Measure of Yanowitz's method

P3- Precision of Niblack's method, R3- Recall of Niblack's method, FM3- F-Measure of Niblack's method

P4- Precision of Sauvola's method, R4- Recall of Sauvola's method, FM4- F-Measure of Sauvola's method.

P5- Precision of Post Yanowitz's method, R5- Recall of Post Yanowitz's method, FM5- F-Measure of Post Yanowitz's method

Red color- 1st Highest value, Green color- 2nd Highest value, Blue color- 3rd Highest value, Yellow color- 4th Highest value, Purple color- 5th Highest value.

The proposed algorithm has been tested on DIBCO dataset and compares the performance of new method with well defined Yanowitz's method, Niblack's method, Sauvola's method and Post-Yanowitz's method. The Ground Truth (GT) images are used to calculate corresponding Precision, Recall and F-Measure of every image. The average F-Measure value for dataset images are considered as evaluation. The proposed technique achieves best FM and Precision as compare to well-defined techniques.

The Average FM achieves from proposed algorithm in five recent DIBCO 2009, 2010, 2011, 2012 and 2013 datasets [1] [2] [3] [4] are 81.22%, 82.96%, 85.10%, 83.40%, 76.04% respectively and Average Precision are 75.94%, 83.279%, 82.298%, 79.70%, 72.77% respectively. The average recall achieves by our method is more than 80% in all datasets. It is very difficult to achieve a good precision and recall simultaneously. As you can see the tables define above, the proposed algorithm produce a good precision and recall simultaneously.

In DIBCO 2009, 2011 and 2013 dataset, the best Average Recall are 98.69%, 96.07% and 95.06% respectively, achieved by Sauvola's method. In DIBCO 2010, 2012 dataset, the Niblack's method provides best average Recall (i.e. 89.97%, 93.57% respectively). Out of 66 document images our method provides 46 best results (i.e. 69.70%), Yanowitz's method achieve 14 best results (i.e. 21.21%), 12 best results are provided by Niblack's method (i.e. 18.18%) and Sauvola's method provides 3 best results (i.e. 4.54%).

CHAPTER-6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This project report presents a new Text Segmentation and Binarization Approach using the Difference Theoretic Texture Feature. It is very difficult to differentiate between foreground and background of an image using a threshold algorithm when the intensity value of background is something similar to foreground. In some cases background may be quite complex. The propose method achieves the best result in removal of foreign bodies and binaries only the text area of an image. The method first enhances the document image, extract text using the Difference Theoretic Texture Feature and then binarize the text area.

The proposed algorithm has evaluated based on two challenging datasets that includes more than 100 document images: one is most popular Document Image Binarization Contest (DIBCO) dataset (66 images) and the second is created by me using Samsung Galaxy S4 Mobile Phone (36 images). The second dataset created using mobile phone represents a collection of varied document images that contain text and noise (noise is like fingers, pen, pencil, mobile, shadow, watch and so on). The results of both datasets were reviewed in previous section. The Average FM value is used to evaluate the proposed algorithm. The maximum performance, efficiency has obtained using the proposed technique with an average FM which is above than 75% in all DIBCO datasets and with an average precision which is above than 70%. The presented algorithm was compare to more well defined binarization techniques. The major advantage of my method is to differentiate the text and other foreign bodies from the document images.

However, as you can see experimental results of all techniques the best result produces by our technique in term of F-Measure and it retains only the text area from good as well as complicated document images.

6.2 Future Work

In future, the work will be for improving the time and space complexity of the algorithm by using another concept of image processing and make a better classification technique. The proposed technique provides a good recall, but not best, and in future also trying to improve recall value. To create this algorithm the template size is initially defined for every image. An automatic template size will also try to make according to input image which able to improve performance. This technique of text binarization can also be extended to recognition of characters from text area.

CHAPTER-7

REFERENCES

[1] Gatos, B.; Ntirogiannis, K.; Pratikakis, I., "ICDAR 2009 Document Image Binarization Contest (DIBCO2009)," in *10th International Conference on*, vol. 1375, pp. 26-29.

[2] Pratikakis, I.; Gatos, B.; Ntirogiannis, K., "H-DIBCO 2010 - Handwritten Document Image BinarizationCompetition," in *Frontiers in Handwriting Recognition (ICFHR)*, vol. 727, 2010, pp. 16-18.

[3] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. 2011, "ICDAR 2011 Document Image Binarization Contest (DIBCO 2011)," in *2011 International Conference on Document Analysis and Recognition (ICDAR '11)*, 2011.

[4] Pratikakis, I.; Gatos, B.; Ntirogiannis, K., "ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012)," in *2012 International Conference on*, vol. 817, 2012, pp. 18-20.

[5] OTSU N., "A Threshold Selection Method from Gray-level Histograms," *IEEE Trans. Syst. Man Cybern*, pp. 62-66, 1979.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing. Reading.*: MA: Addison-Wesley, 1992.

[7] Seba Susan, Madasu Hanmandlu, "A non-extensive entropy feature and its application to texture classification," *IET Image Process,* pp. 214-225, 2013.

[8] S. D. Yanowitz and A. M. Bruckstein, "A NEW METHOD FOR IMAGE SEGMENTATION," *IEEE Trans.*, pp. 270-275, 1988.

[9] "http://www.mathworks.in/matlabcentral/fileexchange/40266-yanowitz-local-image thresholding,".

[10] N. Kapur, P. K. Sahoo, and A. K.C. Wong, "A new method for graylevel picture thresholding using the entropy of the histogram," *Computer Vision Graphics Image Processing*, pp. 273-285, 1985.

[11] "http://www.mathworks.in/matlabcentral/fileexchange/40266-niblack-local-ithresholding.,".

[12] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition 33(2)*, pp. 225-236, 2000.

[13] J. Bernsen, "Dynamic thresholding of gray level images," *Proc. Intl. Conf. on Pattern Recognition*, pp. 1251-1255, 1986.

[14] Øivind Due Trier and Torfinn Taxt, "Evaluation of Binarization Methods for Document Images".

[15] B. Gatos, I. Pratikakis, and S. Perantonis, "Adaptive degraded document image binarization," *Pattern Recognition, vol. 39*, pp. 317-327, 2006.

[16] R. Rowley-Brooke, F. Piti´e, and A. Kokaram, "A ground truth bleedthrough document image database: Theory and practice of digital libraries," *Lecture Notes in Computer Science, P. Zaphiris, G. Buchanan, E. Rasmussen, and F. Loizides, Eds. Springer Berlin / Heidelberg*, vol. 7489, pp. 185-196, 2012.

[17] Abdullah, S.N.H.S., et al., "An evaluation of classification techniques using enhanced Geometrical Topological Feature Analysis," in *2nd Malaysian Joint Conference on Artificial Intelligence (MJCAI 2010)*, Malaysia, 2010, pp. 12-22.

[18] Abdullah, S.N.H.S., et al., "Multi-threshold approach for license plate recognition system," in *International Conference on Signal and Image Processing WASET Singapore August 25-27*, 2010, pp. 1046-1050.

[19] Neha Gupta, V.K. Banga, "Image Segmentation for Text Extraction," in *2nd International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2012)*, Singapore, 2012, pp. 182-185.

[20] Chung-Wei Liang and Po-Yueh Chen, "DWT Based Text Localization," *International Journal of Applied Science and Engineering*, pp. 105-116, 2004.

[21] S.Audithan, RM. Chandrasekaran, "Document Text Extraction from Document Images Using Haar Discrete Wavelet Transform," *European Journal of Scientific Research ISSN 1450-216X*, vol. 36, 2009.

[22] Ankush Gautam, "Segmentation of Text From Image Document," *Ankush Gautam et al, / (IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 4, pp. 538-540, 2013.

[23] M. Pietikäinen and O. Okun, "Text Extraction from Grey Scale Page Images by Simple Edge Detectors," *Proc. of the 12th Scandinavian Conf. On Image Analysis, Bergen, Norway*, pp. 628-635, 2001.

[24] Jie Xi, Xian-Sheng Hua, Xiang-Rong Chen, et al., "A Video Text Detection and Recognition System," *Proc. of ICME 2001, Waseda University, Japan*, pp. 1080-1083, 2001.

[25] Neha Gupta, V.K. Banga, "Image Segmentation for Text Extraction," in *ICEECE'2012*, 2012.

[26] Santosh, Dr. Jenila Livingston L.M., "Text Detection From Documented Image Using Image Segmentation," *INTERNATIONAL JOURNAL OF TECHNOLOGY ENHANCEMENTS AND EMERGING ENGINEERING RESEARCH*, vol. 14, pp. 144-148.

[27] PirahanSiah, F., S.N.H.S. Abdullah, and S. Sahran, "Adaptive image segmentation based on peak signal-to-noise ratio for a license plate recognition system," in *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on. 2010.*

[28] PirahanSiah, F., S.N.H.S. Abdullah, and S. Sahran, "Comparison single thresholding method for handwritten images segmentation," in *Pattern Analysis and Intelligent Robotics (ICPAIR), 2011 International Conference on. 2011*.

[29] Bovik, A., Clark, H., Geisler, W.S., "Multi-channel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Mach. Intell*, pp. 55-73, 1990.

[30] Farshid Pirahansiah, Siti Norul Huda Sheikh Abdullah and Shahnorbanun Sahran, "Peak Signal-to-Noise Ratio based on Threshold Method for Image Segmentation," *Journal of Theoretical and Applied Information Technology*, vol. 57, nov 2013.

[31] http://ses.library.usyd.edu.au/bitstream/2123/699/2/adt-NU20051223.18554203chapter2.pdf,.

[32] "http://www.mathworks.in/matlabcentral/fileexchange/40266-sauvola-local-image-thresholding,".

[33] William K. Pratt,.: John Wiley & Sons, 1991, pp. 501-504.

[34] "http://www.mathworks.com/matlabcentral/fileexchange/41786-remove-ghosts-from-binarized-images/content/deghost.m,".

[35] Raman Maini and Himanshu Aggarwal, "A Comprehensive Review of Image Enhancement," *JOURNAL OF COMPUTING*, vol. 2, no. 3, March 2010.

[36] "http://funnotes.net/tofpages/TopicOfFortnight.php?tofTpcFl=topicoffortnight22,".

[37] Chumman Lal Chandrakar, Monisha Sharma, S K Singhai, Sscet Bhilai, Gec Bilaspur, "SURVEY OF IMAGE CONTRAST ENHANCEMENT METHODS," *International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD)*, pp. 56-63, 2012.

[38] http://www.cs.uregina.ca/Links/class-info/425/Lab3/lesson.html,.

[39] Haralick, R.M., Shanmugan, K., Dinstein, I., "Textural features for image classification," *IEEE Trans. Syst. Man Cybern*, pp. 610-621, 1973.

[40] Idrissi, N., Martinez, J., Aboteydine, D., "Selecting a discriminant subset of co-occurrence matrix features for texture based image retrieval," *Int. Symp. Visual Computing (ISVC)*, pp. 696-703, 2005.

[41] Cross, G.R., Jain, A.K., "Markov random field texture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 25-39, 1983.

[42] Idrissa, M., Acheroy, M., "Texture classification using Gabor filters," *Pattern Recognit. Lett.*, pp. 1095-1102, 2002.

[43] Estudillo-Romero, A., "Escalante-Ramirez, B.: “Rotation-invariant texture features from the steered Hermite transform," *Pattern Recognit. Lett.*, 2011.

[44] Bai, C., Kpalma, K., Ronsin, J., "Color textured image retrieval by combining texture and color features," in *Signal Processing Conf. (EUSIPCO)*, 2012.

[45] Unser, M., "Texture classification & segmentation using wavelet frames," *IEEE Trans. Image Process.*, pp. 1549-1560, 1995.

[46] Kim, S.C., Kang, T.J., "Texture classification and segmentation using wavelet packet frame and Gaussian mixture model," *Pattern Recognit.*, pp. 1207-1221, 2007.

[47] Avci, E., "Comparison of wavelet families for texture classification using wavelet packet entropy adaptive network based fuzzy inference system," *Appl. Soft Comput.*, pp. 225-231, 2008.

[48] Laine, A., Fan, J., "Texture classification by wavelet packet signatures," *IEEE Trans.*

Pattern Anal. Mach. Intell., pp. 1186-1191, 1993.

[49] Avci, E., "An expert system based on wavelet neural network adaptive norm entropy for scale invariant texture classification," *Exp. Syst. Appl.*, pp. 919-926, 2007.

[50] Pun, C.M., Lee, M.C., "Log-polar wavelet energy signatures for rotation and scale invariant texture classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 590-603, 2003.

[51] Ojala, T., Pietikaenen, M., Maenepae, T., "Multi-resolution gray scalev and rotation invariant texture classification with LBP," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 971-987, 2002.

[52] Guo, Z., Zhang, L., Zhang, D., "Rotation invariant texture classification using LBP variance with global matching," *Pattern Recognit.*, pp. 706-719, 2010.

[53] Powers, David M W (2007/2011)., "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies 2 (1)*, pp. 37-63.

[54] Reza Firsandaya Malik, Saparudin, Intan Septyliana, "An Analysis of Adaptive Approach for Document Binarization," *Computer Engineering and Applications*, vol. 2, pp. 185-194, March 2013.

PUBLICATION

[1] Seba Susan, K.M. Rachna Devi. "Enhancement of Document Images Captured by Mobile Phones using the Adaptive Threshold Technique." *MECON-2013 International Conference.* Noida, 2013. 70-71.