A Novel Approach for Detecting and Recognizing Mathematical Symbols

Nusrat Jahan   12201001

Mirza Abdullah Al Nafiz   13101230

Supervised by

Mr. Rubel Biswas
Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Department of Computer Science and Engineering, BRAC University | 2016

TABLE OF CONTENTS

Table of Contents I

List of Figures II

List of Tables III

Declaration IV

Acknowledgement V

Abstract VI

Chapter 1 | Introduction

1.1 Motivation 11

1.2 Problem Statement and Methodology 11

1.3 Goals 12

1.4 Thesis Contribution 12

1.5 Related Work 13

1.6 Thesis Outline 13

Chapter 2 | Literature Review

2.1 Image Segmentation 14

2.1.1 Line segmentation 14

2.1.2 Word Segmentation 15

2.1.3 Character Segmentation 15

2.2 Segmentation Methodologies 15

2.3 Hindrances for Image Segmentation 16

2.4 Template Matching 16

2.4.1 The Cross Correlation Method 17

2.4.2 The Normalized Cross Correlation Method 18

2.5 HOG (Histograms of Oriented Gradients) 18

2.6 SVM (Support Vector Machine) 19

2.7 Preprocessing of Images 19

2.7.1 Binarization 20

Chapter 3 | Methodology

3.1 Acquisition 22
3.1.1 Target Image Acquisition 23
3.1.2 Templates or Patch Images Acquisition 23
3.1.3 Dataset Acquisition 23
3.1.4 HOG Feature Extraction 24
3.3 Preprocessing 25
3.3.1 Binarization 25
3.4 Segmentation 25
3.4.1 Line Segmentation 26
3.4.2 Word Segmentation 26
3.4.3 Character Segmentation 27
3.5 Detection 27
3.5.1 Template matching 27
3.6 Recognition 27
3.6.1 SVM 27

Chapter 4 | Experimental Results

4.1 Segmentation Results 29
4.2 Detection Results 31
4.3 Recognition Results 32

Chapter 5 | Conclusion

5.1 Future Work 33
5.2 Conclusion 33

Reference

LIST OF FIGURES

Figure 2.4 Template Matching Example

Figure 2.6.1a Binarization Example 1

Figure 2.6.1b Binarization Example 2

Figure 3.0 Methodology

Figure 3.1.1 Target image or Input Image

Figure 3.1.2 Patch images for template matching.

Figure 3.1.3 Dataset For Hog Feature Extraction

Figure 3.1.4 Feature Extraction, A portion of the xls_data_for_(

Figure 3.1.1 Binarized input image

Figure 4.1a Line Segmentation on Input image

Figure 4.1b Word Segmentation on Input image

Figure 4.1c Character Segmentation on Input Image

Figure 4.2a Detection of Opening & Closing First Bracket in a line

Figure 4.2b Detection of Opening & Closing First Bracket in a line

Figure 4.6.1a Hog Classification

LIST OF TABLES

Table 4.3 SVM Results

Declaration

We hereby declare that the work shown in this thesis is solely done by us and the results found here are the outcomes of our study and research. The work of any other researcher used in this thesis is mentioned in the reference section. This work neither partially nor wholly has been previously submitted for any other degree program.

Author's Signature | **Signature of Supervisor**

_________________________ | _________________________

Nusrat Jahan | Mr. Rubel Biswas

____

Mirza Abdullah Al Nafiz

Acknowledgement

We would like to express our sincere gratitude to our thesis supervisor Mr. Rubel Biswas for providing us with his continuous guidance and support throughout our thesis work. We would like to thank him for introducing us to many new concepts that were unknown to us and for helping us whenever we had problems regarding our research and experiments. We are grateful for his motivation and inspiration.

We express our gratitude to our parents who have always cooperated with us and encouraged us to excel in our academics work, humanity and thus encouraged us to excel in life.

We would like to thank our classmates and friends for encouraging, inspiring us and for always cheering us up during tough times.

We would like to thank our respected faculty members of the CSE department and the Lab staff for directly or indirectly helping us during our thesis.

Abstract

Extraction and recognition of mathematical characters and equations for documented images is important for many applications, artificially intelligent systems in particular to store or analyze mathematical data. It is vital that efficient and robust approaches are utilized in this sector. Recognition methodologies that are widely used at present work very well for non-mathematical data, but for mathematical data, the results are somewhat unsatisfying. Moreover, precision is important while extracting mathematical texts. In our thesis we have studied a number of text detection algorithms and used image segmentation methodologies, template matching, SVM (Support Vector Machine) to detect and recognize mathematical symbols from a document image.

Chapter 1 | Introduction

Image processing techniques are very popular in the world of text recognition. Numerous techniques have been discovered in the field of image processing that makes the recognition of texts from images an easy task. These techniques are capable of recognizing characters very accurately from images. But for mathematical symbols the same techniques give a very different result.

One of the most popular techniques used for text extraction from images is OCR (Optical Character Recognition). Though OCR works very well for non-mathematical data, its accuracy in recognition drops dramatically for mathematical symbols and characters.

Since detection and extraction of mathematical data from images is of great importance, we have tried to implement a methodology of our own using multiple existing techniques of image processing to detect mathematical symbols from images.

In this thesis, we have considered document images as our input from which we are going to detect the mathematical symbols.

1.1 Motivation

There is a huge collection of printed and handwritten mathematical documents that may have potentially beneficial knowledge for solving complex mathematical problems. These documents have mathematical symbols which normal image processing techniques cannot detect. Our thesis aims to detect and recognize those mathematical data combining image processing techniques like image segmentation along with object recognizing techniques like template matching and SVM (Support Vector Machine). Our work will enable applications to extract and evaluate mathematical expressions from printed texts. Also it will serve as a process to digitalize hard copies of mathematical books and documents.

1.2 Problem Statement and Methodology

One of the problems of detecting texts from images is that it requires a good amount of processing time. This can be a major hindrance in efficiency when working with a huge number of images. One of the ways we can make this better is by using image segmentation.

By image segmentation we can divide the entire image into small regions which reduces the time required for matching or comparison.

Since we are aiming towards finding the mathematical symbols only, our dataset mainly comprises of images of mathematical operators like +, -, ×, ÷, = and parenthesis (,), {,}, [,]. The idea is to find lines, words and character segments that may contain a match with any of the above since an equation must have a “=” sign in between its other coefficients and variables. We aim to detect the symbols using a detection technique called template matching.

Another key problem for recognition of texts is the accuracy of matching. A “(” and a “c” can appear to be the same during template matching process. So we had to conduct experiments using various approaches to find a comparatively more accurate method. We decided to use HOG feature extraction and SVM for recognition.

1.3 Goals

In our thesis our main objective is to build a system that will use image processing techniques to detect and recognize mathematical symbols from a document image. We want to improve the processing time by dividing the entire image into lines, words and characters using image segmentation and then use detection techniques like template matching to find the location of the mathematical symbols in the image. We then want to classify the detected symbols using HOG feature extraction and SVM. We also want our algorithm to be accurate, efficient and robust.

1.4 Thesis Contribution

With the help of our thesis work various application softwares will be able to detect mathematical symbols in document images and use them for further evaluation and storage and use them for further analysis.

1.5 Related Work

Dr. Jenila Livingston L.M. ,et.al [1] in their research on “Text Detection From Documented Image Using Image Segmentation ” have used image segmentation for document images to separate lines, words and characters to detect texts.

Gupta Mehul, et.al [2] in their research on “Text-Based Image Segmentation Methodology” have discussed about the general methodologies of image segmentation and have also given a comprehensive analysis on which approach works better for document images.

Dr. S.Vijayarani, et.al [10] in their research on “Template Matching Technique For Searching Words in Document Images” have elaborately explained the process and approaches for template matching. They have also provided a comprehensive analysis on the comparison among the results of those methodologies.

1.6 Thesis Outline

- Chapter 1 describes what we plan to do and what we have achieved through our thesis.
- Chapter 2 provides necessary background research and literature review in image processing techniques like image segmentation, template matching, preprocessing and recognition. We have discussed every topic in details to make clear the underlying concepts of each technique and how they work.
- Chapter 3 provides a detailed step by step description of our methodology in experimentation. It also shows each of the steps with their respective outputs.
- Chapter 4 discusses about the results and evaluation.
- Chapter 5 concludes our thesis with future work and a brief summary.

Chapter 2 | Literature Review

2.1 Image Segmentation

Segmentation refers to a process of partitioning an image into groups of pixels which are homogeneous with respect to some criterion [1]. It splits the entire image into several connected areas or cells.

Image segmentation is the process of segmenting or dividing an image into multiple regions or sets of pixels based on some criterion such as intensity, color, texture etc. It is an important step in object detection, object representation, visualization, and text detection and in the retrieval of specific information from an image. Image segmentation can be broadly classified into two types [3].

1. **Local Segmentation:** It deals with the segmenting sub images which are small windows on a whole image[1]
2. **Global Segmentation:** It deals with the images consisting of relatively large number of pixels and makes estimated parameter values for global segments more robust [1].

During the segmentation process horizontal and vertical segmentation is carried out on the image which produces outputs as lines and either words or characters respectively. There are three main phases of image segmentation.

I. Line Segmentation

II. Word Segmentation

III. Character segmentation

2.1.1 Line segmentation

Line segmentation is the foremost step for text based image segmentation. It includes horizontal scanning of the image, pixel-row by pixel-row from left to right and top to bottom [2] [9] [11] [12] [13]. The level of intensity at each pixel is detected and depending on this value these pixels are grouped into numerous regions. Each region is a part of the entire image and can be extracted for further use or segmentation. In case of document images these are lines from which words and character segmentation is done.

2.1.2 Word Segmentation

Word segmentation is the next phase of image segmentation. It includes vertical scanning of the image, pixel-row by pixel-row from left to right and top to bottom [10] [16]. Each of the horizontal lines obtained after line segmentation is taken and again the level of intensity at each pixel at each line is detected and depending on this value these pixels are grouped into numerous regions. Each region is a part of the entire image and can be extracted for further use or segmentation. In case of document images these are words from which character segmentation is done.

2.1.3 Character Segmentation

Character segmentation is the last stage for text based image segmentation. It is similar to in operations as word segmentation [2] [11] [14] [15]. Character segmentation is fruitful in both the case of segmenting after the line segmentation and word segmentation phases i.e. the input for this phase can be either a line or a word. After character segmentation the output is a group of regions each comprising of the characters of the given text image.

Segmentation is an important stage because the extent one can reach in separation of words, lines, or characters directly affects the recognition rate of the script [10].

2.2 Segmentation Methodologies

The segmentation of a document image depend greatly on the guidelines present in that document. An image with guidelines is less likely to have any skew in its line orientation. Guidelines or border lines also limits the size of the characters to be used thus it becomes easier to segment the image. Also the higher the level of segmentation the greater will be the need of precision. There are a number of approaches for image segmentation as proposed in [15]. In this paper the author has explained the *Pixel Counting Approach*, the *Histogram approach*, the *Smearing Approach*, the *Stochastic Approach* and the *Water Flow Approach*. In the paper “Text-Based Image Segmentation Methodology ”[2] the author has provided a comprehensive analysis on why the histogram approach and the pixel counting approaches are more appropriate for segmenting document images. In our thesis we have adapted the histogram approach for segmenting the images.

2.3 Hindrances for Image Segmentation

There are various factors that hinder the process of text based image segmentation [8] [20].

I. **Quality of the Image:** Presence of noise in the image results in degradation of accuracy and efficiency [24].

II. **Handwritten and Printed Document [2]:** For printed document the distance between the neighboring text lines is precise whereas the same cannot be said about handwritten documents. Thus image segmentation is a big challenge for images with handwritten texts. For printed documents however the task of segmentation is easier and more precise. Cursive text provides additional difficulty during character segmentation, due to the presence of ligatures [2].

III. **Alignment and Type of the Symbols:** For handwritten document if the individual lines are not straight or if there is a presence of skew then the overall complexity for text extraction increases [22] [23].

IV. **Texture of the Image:** presence of texture in the image can be problematic for the image segmentation process.

2.4 Template Matching

Template matching is a simple task of performing a normalized cross-correlation between a template image (object in training set) and a new image to classify [16]. For this method we need two components:

I. Target or source image: The image in which we want to find a match of the template image provided by us.

II. Template image: The image which will be compared to the source image.

Figure 2.4 Template Matching Example

Comparison is made between the source image and template image or images to detect the highest matching area. To locate the matching area, the template image is compared to the source image by sliding it against the source image i.e. the template image is moved by one pixel at a time (left to right, up to down). At each point a value is generated which tells how similar the template is to that particular area of the image. There are a number of approaches to perform template matching in image processing.

I. The Cross Correlation method
II. The Normalized Cross Correlation method
III. The Performance Index method

We have used the Cross Correlation and The Normalized Cross Correlation method for template matching. These methods are described below.

2.4.1 The Cross Correlation Method

The cross-correlation template matching is motivated by the distance measure (squared Euclidean distance) [10] [17], [18].

$$d^2_{f,t}\,(u,v) = \sum_{x,y}[f(x,y) - t(x-u, y-v)]^2 \quad \text{……….. (I)}$$

f being the input image and *t* being the template image. We get the summation over *x, y* where we take the difference of *f(x, y)* and *t(x-u, y-y)* under the window containing the feature t at *(u, v).*

$$d^{\,2}{}_{f,t}\,(u,v) = \sum_{x,y}\ [f^{\,2}(x,y) - 2f(x,y)t(x-u,y-v) + t^{\,2}(x-u,y-v)] \quad \text{……….. (II)}$$

In equation. (II) The term $\sum_{x,y} f^2(x,y)$ is a constant and if $t^2(x-u,y-v)$ an approximate constant is also then the rest of the equation can be said to be the measure of similarity between the feature and the source image. Thus we have,

$$c(u,v) = \sum_{x,y} f(x,y)t(x-u,y-v) \text{ ……… (III)}$$

2.4.2 The Normalized Cross Correlation Method

A few difficulties may arise during matching using the correlation method. For example, the correlation between feature and an exactly matching region in the image may be less than the correlation between the feature and a bright spot [24]. Again the range of *c (u, v)* is not independent of the size of the feature. The correlation coefficient overcomes these difficulties by normalizing the image and feature vectors to unit length, yielding a cosine-like correlation coefficient [19].

$$\gamma = \frac{\sum_{x,y} f(x,y)-\overline{f}_{u,v})(t(x-u,y-v)-\overline{t})}{\sqrt{\sum_{x,y} (f(x,y)-\overline{f}_{u,v})^2 \sum_{x,y} (t(x-u,y-v)-\overline{t})^2}} \text{ …………… (IV)}$$

Equation. (IV) Is the Normalized cross correlation which is the mean of the feature and the mean of *f(x, y)* in the region under the feature.

2.5 HOG (Histograms of Oriented Gradients)

The histogram of oriented gradients is a feature descriptor. It has it's usages in image processing and computer vision. This approach was first introduced by Navneet Dalal and Bill triggs. They were the teachers for French National Institute for Research in Computer Science and Automation. Their work aims to detect human face. This method counts the occurrence of gradient orientation in localized portion of the image. Rodrigo Minetto and his team used histogram of oriented gradients for text images, where they developed T-HOG [22].

2.6 SVM (Support Vector Machine)

In machine learning support vector machine, is a supervised learning model. Here the data is analyzed for classification and regression analysis. SVM groups the point s according as the training data set. When a new point comes SVM assigns it in one category or another category. For this classification it acts as a non-probabilistic binary linear classifier. Furthermore, we can perform linear non -linear classification using kernel trick. To classify data we need to label the training data. Without labeling support vector machine cannot classify. We can use support vector clustering then.

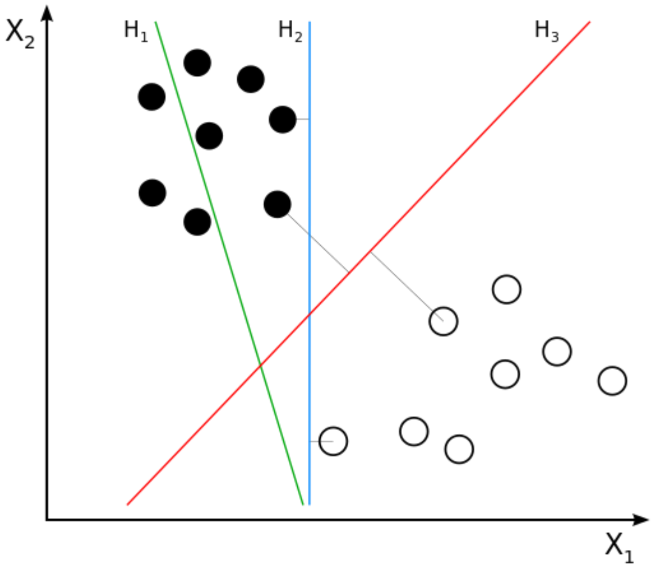


Figure 4.6.1a Hog Classification

In Figure 4.6.1a, H1 tries to classify but not successfully. In next phase H2 classify them but is has a lower margin. H3 classifies them with a large margin.

2.7 Preprocessing of Images

Prior performing segmentation, the target image must be brought into a specific format to simplify the subsequent processing [2]. Whatever be the source of the document image, there is always a difference in the intensity levels and added noise can also be a problem in some case so the image must be preprocessed before the segmentation stage. After preprocessing the image obtained will have a better amount of shape information, high compression and low noise i.e. preprocessing removes any noise that might have been added to the image during acquisition or transmission. It is important to mention that a colored image must be converted in to gray image prior to noise removal. Thus preprocessing is an important step for the better performance of the overall system. The preprocessing [4] [5] [6] includes noise removal [7], binarization, normalization [2].

2.7.1 Binarization

Image binarization is the initial step in many document analysis and recognition system, its objective is to extract the text from image background [20]. It is a Bi-level representation of images in which there are only two levels- a high and a low. This is one of the preprocessing techniques whose accuracy is vital for a successful segmentation and recognition phase. So effective binarization is crucial. There have been many studies in the field of determining accurate methods of binarization. Sezgin and Sankur [21] classified the image binarization methods into six categories: *entropy-based methods histogram-based methods, foreground attribute-based methods, spatial binarization methods clustering-based methods, and locally adaptive methods* [20]. Another popular technique of binarization is image thresholding technique and a lot of methods have been proposed on the basis of this technique. Some of these methods are *Global thresholding, Adaptive mean thresholding, Adaptive Gaussian thresholding* and *Otsu's Thresholding.*

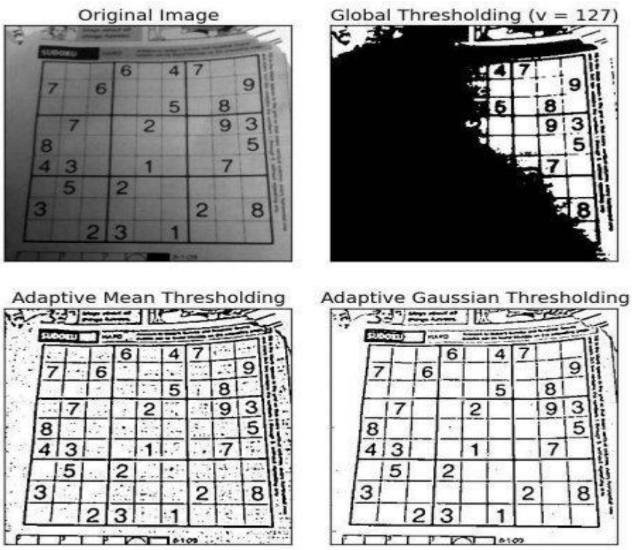


Figure 2.6.1a Binarization Example 1

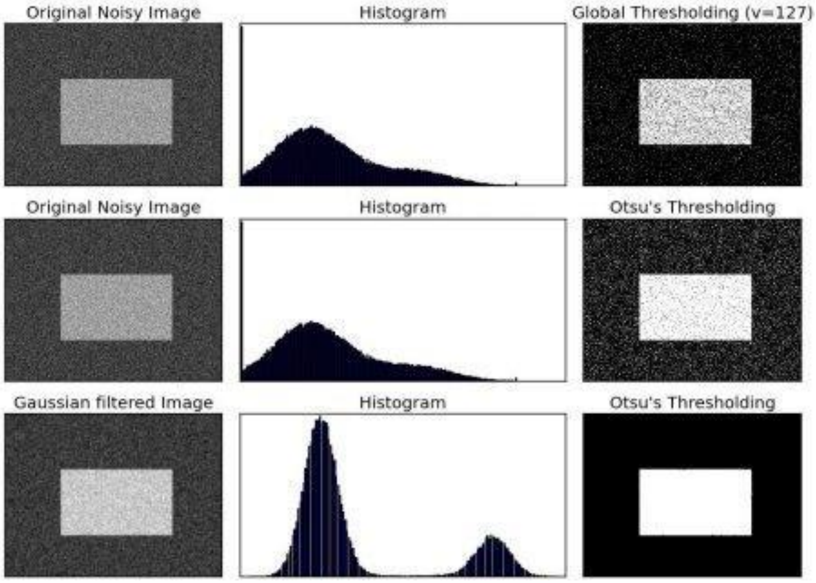


Figure 2.6.1b Binarization Example 2

Chapter 3 | Methodology

In a broader sense our entire thesis methodology is divided into five major stages. These stages are Acquisition, Preprocessing, Segmentation, Detection and Recognition. We have made a GUI (Graphical User Interface) that displays the original image and the resultant image after each of the stages is performed.

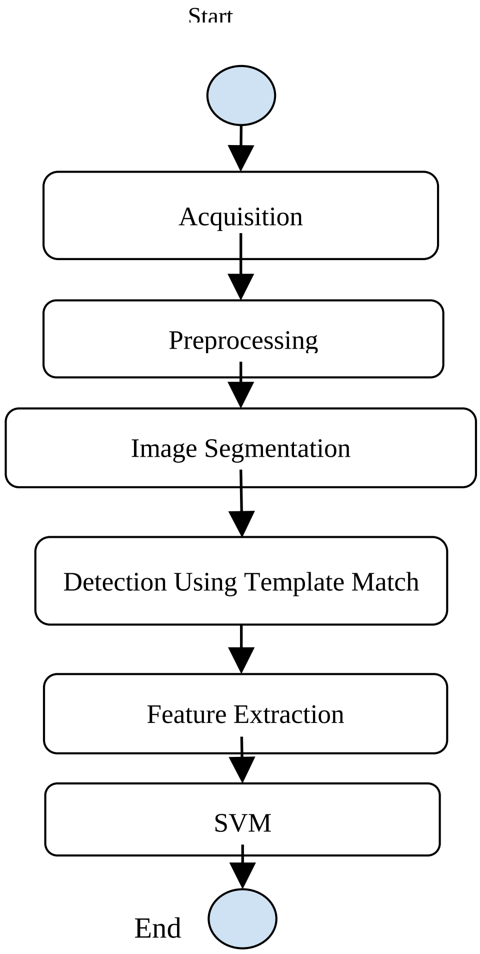


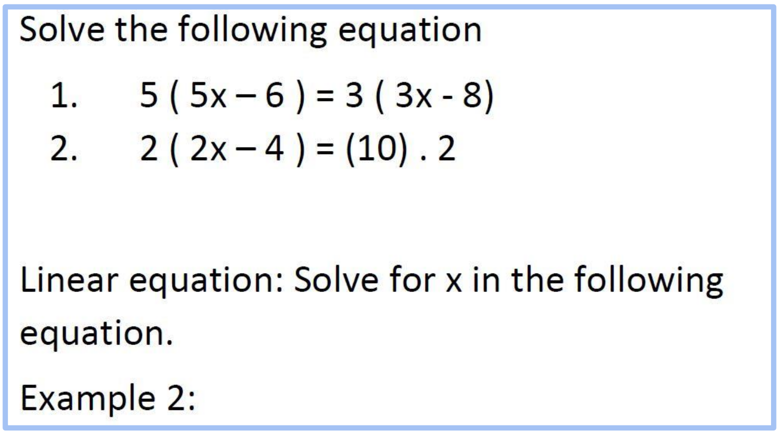
Figure 3.0 Methodology

3.1 Acquisition

In the image acquisition phase we have taken an input image and also we have acquired the required datasets of template images that were later on used for template matching. We have also derived a HOG (histogram of gradients) feature of the images in our dataset for running the SVM (Support vector machine).

3.1.1 Target Image Acquisition

In this stage the image to be read is taken. For online image acquisition the image is taken in the form of document image. The file format can be any image formats such as JPG/JPEG, BMP, PNG, TIFF, GIF etc. For offline sources the image is either scanned with a scanner or a camera. Below is our input or target image from which we want to detect mathematical symbols.


Solve the following equation

1. 5 (5x – 6) = 3 (3x - 8)
2. 2 (2x – 4) = (10) . 2

Linear equation: Solve for x in the following equation.

Example 2:

Figure 3.1.1 Target image or Input Image

3.1.2 Templates or Patch Images Acquisition

We have taken 1 sample for some of the mathematical characters. The mathematical characters were =, (,). The template images are as follows.

Figure 3.1.2 Patch images for template matching.

3.1.3 Dataset Acquisition

We have taken 10 samples for each of the mathematical characters. The mathematical characters were +, -, ×, ÷, =, (,), {,}, [,]. They are 32x32 pixel each. We have considered 10 different font types while preparing the dataset. So the total numbers of samples were 110. The constituent images of our dataset are as follows. We have considered the most popularly used fonts for academic writing. The fonts that were used are given below along with the size in which they were used.

- Sans Serif - size 20
- Times New Roman - size 20
- Century Schoolbook - size 20
- Ariel - size 20
- Bell MT - size 20
- California FB - size 20
- Calisto MT - size 20
- Cambria - size 20
- calibri - size 20

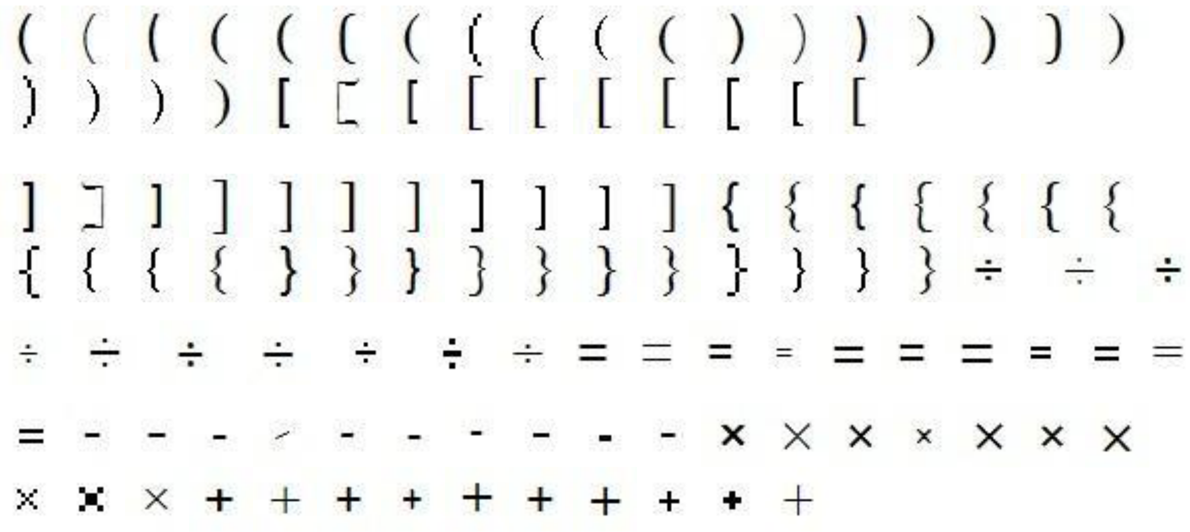


Figure 3.1.3 Dataset for Hog Feature Extraction

3.1.4 HOG Feature Extraction

We have used HOG feature extraction algorithm to extract features of all 110 sample images. Here use have used 1×N Vector for feature extraction.

	A	B	C	D	E	F	G	H	I	J
1	0.014803	0.013033	0.011382	0.009789	0.032356	0.013404	0.016273	0.017391	0.02635	0.052292
2	0.03372	0.028175	0.02448	0.005199	0.017515	0.005143	0.013858	0.018215	0.026254	0.066238
3	0.013719	0.018339	0.015412	0.008802	0.012235	0.007281	0.015435	0.014757	0.012422	0.033831
4	0.020196	0.009034	0.014281	0.008867	0.014299	0.019823	0.02462	0.023988	0.024471	0.048561
5	0.035363	0.018331	0.014831	0.006861	0.011587	0.009079	0.018354	0.02387	0.031556	0.043495
6	0.016086	0.012241	0.013865	0.009869	0.013084	0.004367	0.00725	0.016066	0.02822	0.054925
7	0.024272	0.01978	0.015957	0.01165	0.017606	0.006981	0.00827	0.020888	0.02934	0.038298
8	0.0046	5.69E-05	0.000701	0.00047	0.002999	0.000125	0.000721	0.000457	0.005099	0.030109
9	0.004151	0.000393	0.00033	0.000162	0.004314	4.62E-05	0.000886	0.000366	0.004144	0.06439
10	0.021534	0.010522	0.01892	0.010437	0.03605	0.018502	0.017277	0.014723	0.021041	0.055519

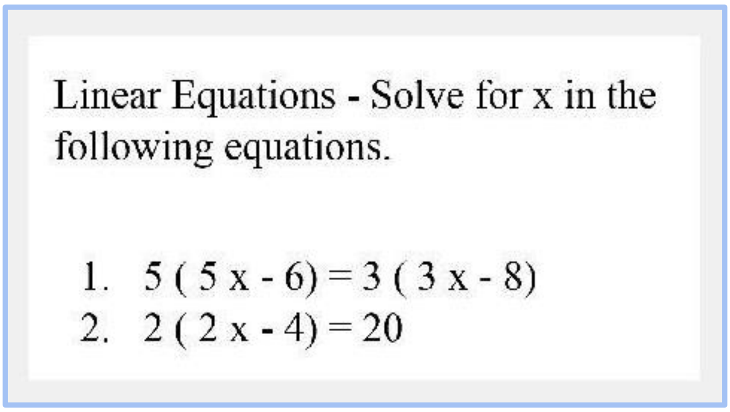
Figure 3.1.4 Feature Extraction, A portion of the xls_data_for_(

3.3 Preprocessing

Before moving on to the segmentation phase we have preprocessed the image for better results.

3.3.1 Binarization

We have used thresholding based method to binarize the input image, the template images and also the images of the dataset. The binarized version of our input image is given below.


Linear Equations - Solve for x in the following equations.

1. 5 (5 x - 6) = 3 (3 x - 8)
2. 2 (2 x - 4) = 20

Figure 3.1.1 Binarized input image

3.4 Segmentation

We have used the histogram based approach to consecutively segment the image into its constituent lines, words and characters. Since we are working on a document image we have considered threshold values for the border line and word gap which helps us separate the lines and the words respectively.

3.4.1 Line Segmentation

We carried out line segmentation on our input image. We used the average frequency of gray level in the image. Where we would find the shades of white as most common frequency we marked that area as line space. Similarly when we found black as most common frequency we marked that area as line.

Algorithm for Line Segmentation:

I. Calculate horizontal histogram and save it in a matrix. Take variable name newLine and set it to 0. Take a list named lineList.

II. Scan the level of grayness for each row and perform the later operation till the last row

III. ***if*** newLine is equal to 0 then

if the level of grayness falls below 250

save the row no. in a variable and set newLine as 1

Else continue

IV. ***Else***

If the level of grayness is above 250 then

save the row no. in a variable and set newLine and add the start, end of this line to a the lineList.

3.4.2 Word Segmentation

We carried out word segmentation on our input image. Here along with finding out horizontal histogram we also find out vertical histogram. After that when we have the matrix of a particular line, then we try to bind the word area. Here are two important concepts. Our function has an option for minimum space between two characters. Point to be noted minimum space between two characters cannot be less than 1pixel. Another concept is how long it will look for next character of the word. Setting this value too high will cause merge of two words. These values need to be fine-tuned depending on the font size, word space etc. After setting this two value the algorithm gives us location of word in a single line.

3.4.3 Character Segmentation

We carried out line segmentation on our input image. To segment the image to characters first we need to find out the lines in the image. Horizontal and vertical histogram and above described algorithm gives us location of a line. In a line we work with the vertical histogram. Our algorithm scans the histogram and saves the location of those pixel where histograms are less than 250. That's indicates there is character. When we first get a pixel which has lower histogram value than 250 we save the location as the start position of the image and after that when we get a higher value than 250 then we save the location as the end position of the character. At last we get a dynamic size list which holds the characters of all lines.

3.5 Detection

First we try to detect mathematical symbols from images. Here, we try to detect symbols in three different way. One is from lines then comes words and lastly we try to detect symbols from character. Our basic idea is to filter out image part that does not contain mathematical symbols. For detection we have used Template matching.

3.5.1 Template matching

Template matching is used for detection purpose. We filter out image part that does not contain mathematical symbols. Here we used normalized cross correlation co efficient.

3.6 Recognition

This is the final stage of our implementation process. We have used SVM to classify our matched symbols.

3.6.1 SVM

After getting feature matrices from HOG we use those values to train our SVM. We have used 11 class which actually is 11 mathematical symbols stated earlier. For each symbols there are 5 rows of data set. HOG generates $1 \times N$ size matrix which is our feature matrix. We use this matrix to train our SVM.

After training there are our 5 sets of data for each symbol. SVM tries to recognize those images. Results are included in the next section.

Chapter 4 | Experimental Results

The experimentation results can be shown in three phases. First the result for the segmentation, second for the detection phase using Template matching and the third and final phase is recognition using SVM.

4.1 Segmentation Results

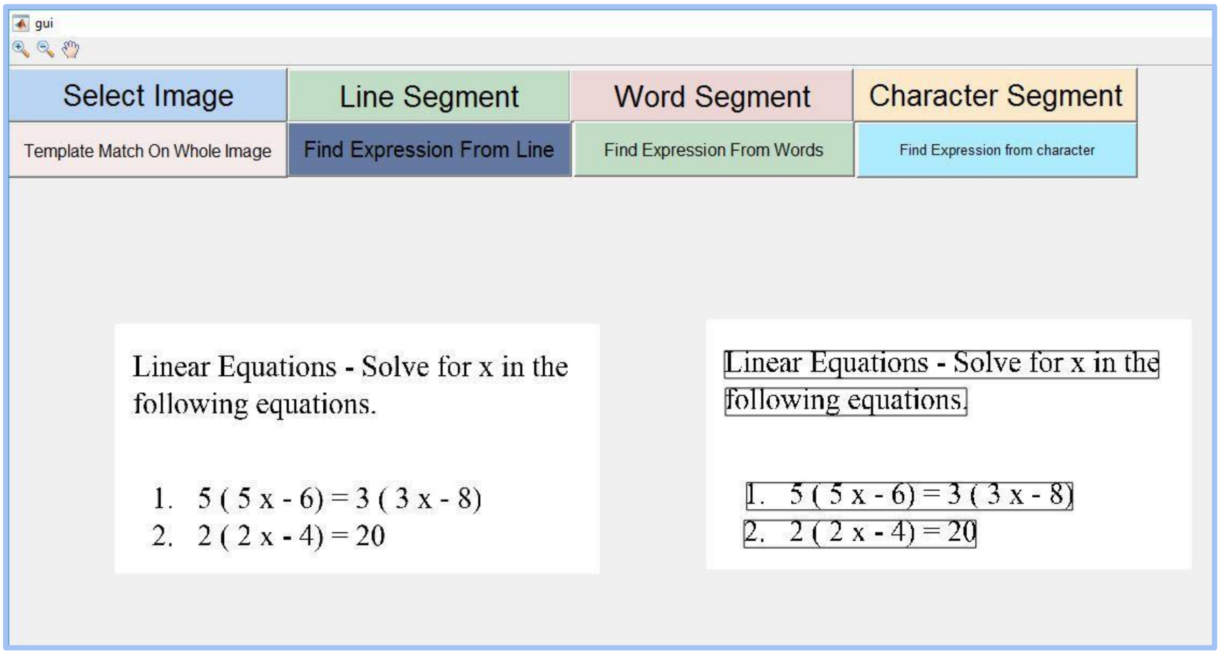


Figure 4.1a Line Segmentation on Input image

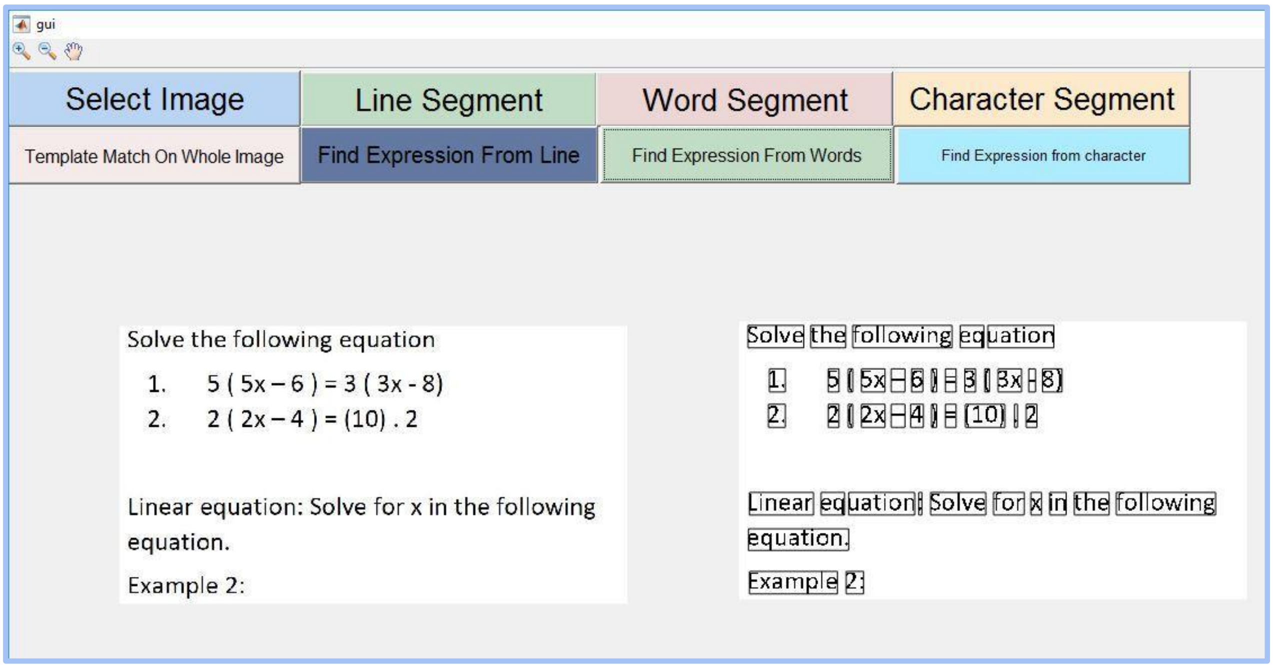


Figure 4.1 b Word Segmentation on Input image

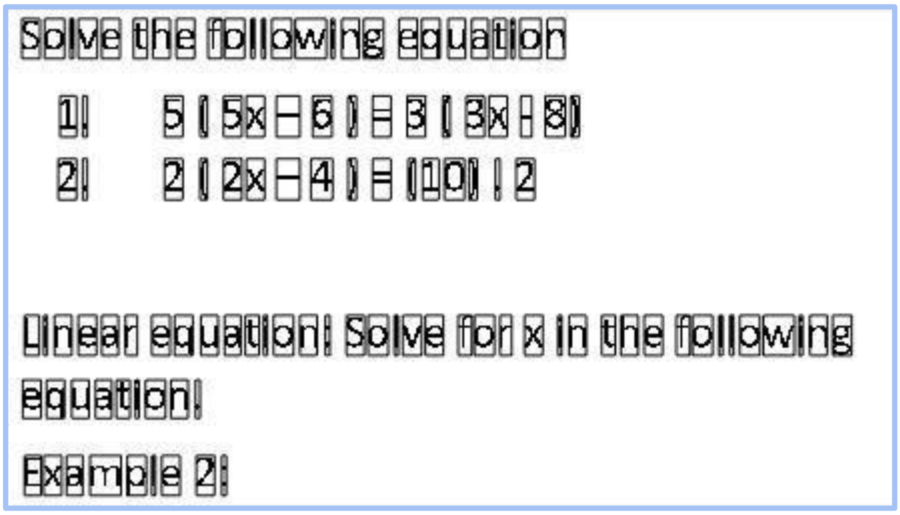


Figure 4.1c Character Segmentation on Input Image

4.2 Detection Results

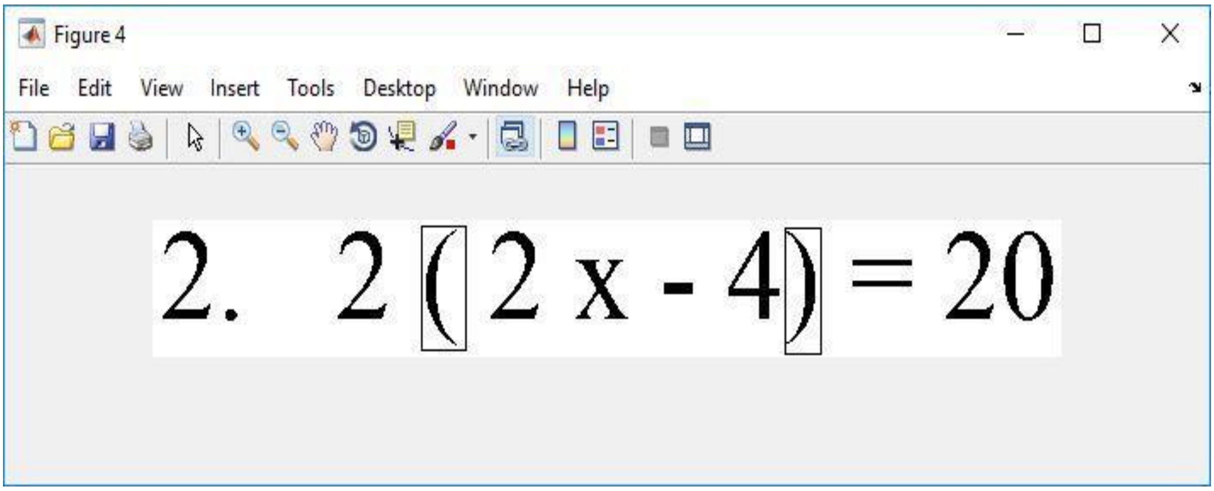


Figure 4.2a Detection of Opening & Closing First Bracket in a line

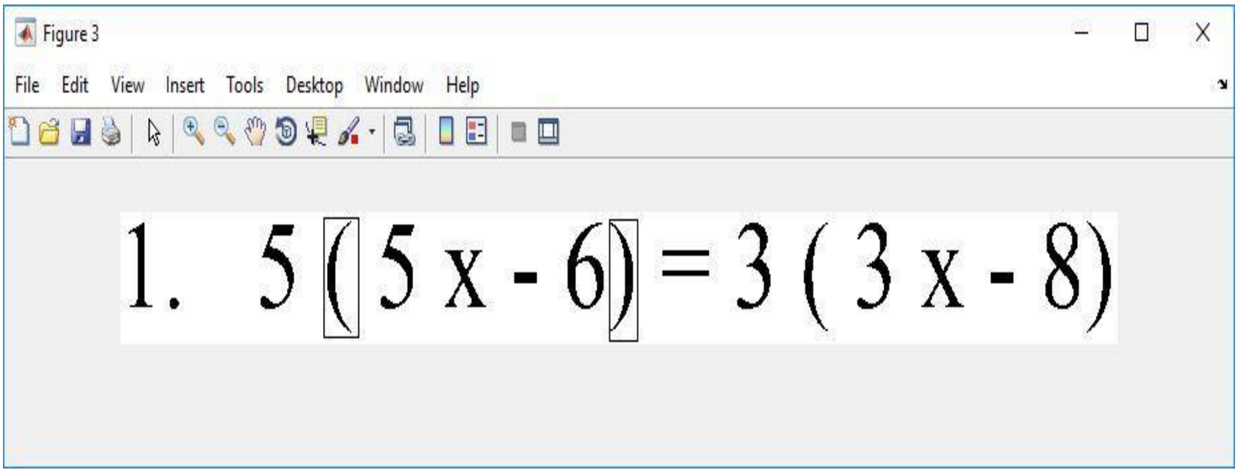


Figure 4.2b Detection of Opening & Closing First Bracket in a line

4.3 Recognition Results

The results of the SVM are given below.

Scenarios	**No of Symbols**											**Total Symbols**	**No of Symbols Recognized**
	(	)	{	}	÷	=	_	×	+	[	]		
Word File	6	6	4	4	2	2	1	2	3	1	1	32	29
Pdf	5	5	2	2	1	2	1	1	1	0	0	20	17

Table 4.3 SVM Results

Chapter 5 | Conclusion

5.1 Future Work

In this thesis we have managed to work on the detection and extraction of the mathematical symbols from an image using our proposed technique. In future we wish to do the following.

- Bring more efficiency to the proposed algorithms.
- Bring better accuracy in the recognition phase.
- Implement algorithm for handwritten text documents.
- Taking consideration of more mathematical symbols like differential operator, integral operator, functions etc.

5.2 Conclusion

The importance of mathematical knowledge and its application is enormous in fields namely Engineering, Computer science, applied physics, Electrical Engineering and so on. We need technologies to preserve this knowledge and use it for the development of mankind. Our thesis is a mere initiative towards that process.

Reference

[1] Santosh, Dr. Jenila Livingston L.M., “Text Detection From Documented Image Using Image Segmentation”, International Journal Of Technology Enhancements And Emerging Engineering Research, Vol 1, issue 4 144 ISSN 2347-4289.

[2] Gupta Mehula, Patel Ankitab , Dave Namratac , Goradia Rahuld, and Saurin Shethe, “Text-Based Image Segmentation Methodology”, International Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 6, December 2015.

[3] Dr.-Ing. Igor Tchouchenkov, Prof. Dr.-Ing. HeinzWörn, “Optical Character Recognition Using Optimisation Algorithms”, Proceedings of the 9th International Workshop on Computer Science and Information Technologies CSIT 2007, Ufa, Russia, 2007.

[4] B. M. Sagar, G. Shobha and P. Ramakanth Kumar,Converting printed Kannada text image file to machine editable format using Database,International Journal of Computers, 2, 2008, 173–175.

[5] S. N. Srihari, V. Govindaraju and A. Shekhawat, "Interpretation of Handwritten Addresses in US Mailstream" in proceedings second International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, IEEE Computer Society Press, pp. 291-294.

[6] M. Maloo and K. V. Kale, "Gujarati Script Recognition: A Review" in International Journal of Computer Science Issues, Vol8, July 2011

[7] S. B. Patil, Neural Network based bilingual OCR system: experiment with English and Kannada bilingual document, International Journal of Computer Applications, 13, 2011, 6–14.

[8] Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti,"Text Line Segmentation Based on Morphology and Histogram Projection", in 10th International Conference on Document Analysis and Recognition, 2009.

[9] M. Thungamani and P. Ramakhanth Kumar, "A Survey of Methods and Strategies in Handwritten Kannada Character Segmentation" in International Journal of Science Research, Vol 01, issue 01, June 2012, pp. 18-23.

[10] Dr. S.Vijayarani and Ms. A.Sakila , Template Matching Technique for Searching Words in Document Images" in International Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 6, December 2015

[11] Nafiz Arica and Fatos T. Yarman-Vural"An Overview of Character Recognition Focused on OffLine Handwriting" in IEEE Transaction on Systems, Man, and Cybernetics, May 2001

[12] D. Brodićand Z. Milivojević,"A New Approach to Water Flow Algorithm for Text Line Segmentation" in Journal of Universal Computer Science, vol. 17, no. 1,2011.

[13] Z. Razak, K. Zulkiflee, R. Salleh, M. Yaacob and E. Mohd, Tamil: A real-time line segmentation algorithm for an offline overlapped handwritten jawi character recognition chip, Malaysian Journal of Computer Science, 20, 2007, 171–182.

[14] K. A. Kluever, Study report character segmentation and classification, http://www.tipstricks.org/example.asp, 2008, 1–21.

[15] T. V. Ashwin and P. S. Sastry, A font and size-independent OCR system for printed Kannada documents using support vector machines:Sadhana, 27, 2002, 35–58.

[16] Luke Cole, Lance Cole, David Austin, Visual Object Recognition using Template Matching: Robotic Systems Lab, RSISE 2, Australian National University, ACT 0200, Australia, pg 2.

[17] Yufan Wang, Qiuze Yu, Wenxian Yu, ―An Improved Normalized Cross Correlation algorithm for SAR Image Registration‖, IEEE IGARSS, 2012.

[18] Vanderbrug, G.J, Rosenfeld, ―Two-Stage Template Matching,‖ IEEE Transactions on Computers, Vol. 60, Issue 11, 1977.

[19] C. Saravanan, M. Surender "Algorithm for Face Matching Using Normalized Cross-Correlation" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-4, April 2013

[20] Rashmi Saini, Document Image Binarization Techniques, Developments and Related Issues: A Review, International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 7, April 2015.

[21] P.K. Sahoo, S. Soltani, ―A survey of thresholding techniques‖, Computer Vision Graphics Image Processing (CVGIP). Vol. 41, pp. 233–260. 1988.

[22] R. Manitto,N. thome,M cord,N. J. Leite, J stolfi. T-HOG: An effective gradient-based descriptor for single line text regions.Pattern Recognition.

[23] Benjamin P. Berman, Richard J. Fateman, Optical Character Recognition for Typeset Mathematics, Computer Science Division, EECS Department University of California at Berkeley.

[24] K. Junga, K.I. Kimb, A.K. Jain, "Text information extraction in images and video: a survey", Pattern Recognition, 2004, pp. 977-997.

[25] Sergey Milyaev, Olga Barinova, Tatiana Novikova, Pushmeet Kohli, Victor Lempitsky, Image binarization for end-to-end text understanding in natural images , Microsoft Research, Cambridge, UK ‡Skolkovo Institute of Science and Technology, Moscow, Russia.

[26] Content-based Multimedia Information Retrieval: State of the Art and Challenges, Michael Lew, et al.,ACM Transactions on Multimedia Computing, Communications, and Applications, pp. 1–19, 2006.

[27] N. Venkateswara Rao, A. Srikrishna, B. Raveendra Babu and G. R. M. Babu, An efficient feature extraction and classification of handwritten digits using neural networks, International Journal of Computer Science, Engineering and Applications, 1, 2011, 47–56.

[28] Zhu Xiaoyan, Shi Yifan,"New Algorithm for Handwritten Character Recognition", Beijing, China.