Fast text image detection

Devadeep Shyam

2015

Devadeep Shyam. (2015). Fast text image detection. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65349>

<https://doi.org/10.32657/10356/65349>

FAST TEXT IMAGE DETECTION

Submitted by
DEVADEEP SHYAM
Supervisor
Prof. Alex C. Kot

School of Electrical & Electronic Engineering

A thesis submitted to Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Master of Engineering

2015

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....Date.....

.....Devadeep Shyam.....

Acknowledgements

First and foremost, I would like to express my sincere thanks to my thesis advisor, Professor Alex C. Kot, for bringing the problem of fast text detection to my attention and allowing me full freedom to explore. His confidence in my abilities to carry out the research work is greatly appreciated. His emphasis on the demonstrability of the ideas has immensely influence my research work. The tireless supports and invaluable inspiration from Prof. Kot cannot be emphasized enough. His sharp insights and excellent guidance; his encouragements in every stage of my research work; his patience and endeavor in improving every single word in papers and patents and his criticism and strict attitude towards the research work have been a resource and model for me to complete this thesis and it will continuously benefit me throughout my lifetime.

I want to thank Miss Wang Yan, PhD student in Nanyang Technological University, who guided me patiently during the project, and encouraged me a lot.

I wish to acknowledge all the panel members of this thesis for their time, valuable comments and supports. I want to thank Professor Wang Lipo, Professor Chan Kap Luk and Professor Ramakrishna Kakarala for the encouragements and for the helpful suggestions. I would like to thank all my subject teachers, especially, Professor Quek Hiok Chai, Professor Bi Guoan , Professor Anamitra Makur, Professor Jiang Xudong and Professor Pina Marziliano, the basic theory and skill that I have learned are the solid foundation of my research work.

My dream towards a master can never be possible without the strong supports from my family. I am indebted to my parents (Rajani Kanta Shyam and Lakhi Shyam) for instilling me the perseverance and hard-working from childhood which benefits me no matter what I choose to do. I am grateful to my brother, Rajdeep Shyam for his inspiration and encouragements, and for his care during the course of my study.

Contents

Acknowledgements	i
Table of contents	ii
List of Figures	iv
List of Tables	vi
Abstract	vii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives and Contributions	2
1.3 Organization of the Thesis	3
2 Literature Review	4
2.1 Text in Images	4
2.1.1 Text in Document Images	4
2.1.2 Text in Scene Images	4
2.1.3 Caption Text in Images	5
2.2 Text Properties in Images	6
2.3 Text Information Extraction	7
2.4 Challenges in Text Detection	9
2.5 Text Detection Techniques	12
2.5.1 Edge-based Methods[47]	12
2.5.2 Texture-based Methods[47]	14
2.5.3 Stroke-based Methods	14
2.5.4 Connected Component-based Methods	16
2.6 Optical Character Recognition (OCR)	17
2.7 Image-based Spam Filtering	18
2.8 Color Contrast	19
2.9 Contrast Adjustment	20
2.10 Periodic Pattern Detection	22
2.10.1 Autocorrelation	22
2.10.2 Discrete Fourier Transform	22
2.11 Morphological Image Processing	23

2.12 Summary	23
3 Text Image Detection by Histogram	24
3.1 Proposed Method	24
3.1.1 Text Extraction	24
3.1.2 Text Distribution Representation	27
3.1.3 Text Images Detection	30
3.2 Text Proportion Estimation	34
3.3 Experiments and Results	35
3.3.1 Experimental setup	35
3.3.2 Threshold value for D_p	38
3.3.3 Experiment results	38
3.4 Discussions	40
3.4.1 Text Detection by SWT	40
3.4.2 Ads Policies for text images	40
3.4.3 Calculating the number of lines	40
3.4.4 Failed images	43
3.4.5 Parameter Analysis	43
4 Text Image Detection by Hough Transformation	46
4.1 Text Alignment Challenge	46
4.2 Type of Text Image with Text Alignment	46
4.3 Text Image Detection for Arbitrary Text Alignment by Hough Transformation	46
4.3.1 Hough line extraction	46
4.3.2 Filtration	47
4.4 Experiments and Results	50
4.4.1 Experimental setup	50
4.4.2 Experiment result	50
4.5 Discussions	50
4.5.1 Failed images	50
4.5.2 Parameter Analysis	50
5 Proposed Text Image Detection System	52
5.1 Proposed Detection System	52
5.2 Experimental Results	53
5.3 Comparison with Existing Techniques	54
5.4 Observations	56
6 Future Work	59
6.1 Text Images of Different Color Contrast	59
7 Conclusions	60
Reference	61

List of Figures

1.1	Some examples of text image	1
2.1	Grayscale document images	5
2.2	Multi-color document images	5
2.3	Images with caption text	6
2.4	Scene text images	6
2.5	TIE System [47]	11
2.6	Some example of challenges in Text Detection	11
2.7	The flowchart of traditional text detection algorithms	13
2.8	Example of a (a)color image (b) gray image	13
2.9	Example of different step by step process for texture segmentation[45,46]	15
2.10	Illustration of finding the line	15
2.11	Pixel is assigned with lowest value	16
2.12	The output result of different stages in the method by Lienhart et al [51,52]	18
2.13	OCR System [47]	19
2.14	Color-wheel	20
2.15	Example of an image using color-wheel	20
2.16	Example of two images from our dataset using color-wheel	21
2.17	Example of histogram equalization	21
2.18	(a) Gray image (b) Result of contrast adjustment	22
2.19	Example of checking an image with a structuring element [18]	23
3.1	Flowchart of text detection method by histogram	25
3.2	(a) Original image (b) Binary image	25
3.3	Example of (a) a text image (b) a non-text image	26
3.4	Binary image of (a) the text image (b) the non-text image	26
3.5	Horizontal connected component	27
3.6	Proposed text extraction from Non-text image example(a) Binary image (b) Horizontal connected components of binary image (c) Obtained text candidate binary image, where $T_1 = 2$, $i = 1, \dots, 12$	27
3.7	Proposed text extraction from Text image example(a) Binary image (b) Horizontal connected components of binary image (c) Obtained text candidate binary image, where $T_1 = 2$, $i = 1, \dots, 12$	28
3.8	Text candidate image of (a) the text image (b) the non-text image	28
3.9	Histogram for (a) text image (b) non-text image	29
3.10	Binary histogram for (a) text image and (b) non-text image	29
3.11	Zoomed (1st 100 rows) binary histogram for text image	30
3.12	Autocorrelation of binary histogram for (a) text image (b) non-text image	32
3.13	First half of the Autocorrelation for (a) text image (b) non-text images	32
3.14	Illustration of Step histogram of maxima	33

3.15	Step histogram of maxima and minima for text image	33
3.16	Step histogram of maxima and minima for non-text image	34
3.17	(a) Original image (b) The result of dilation after Text image detection	35
3.18	(a) Original image (b) The result of dilation after Text image detection	35
3.19	Example of images from our dataset downloaded from internet	36
3.20	Example of text or spam images from our dataset taken from standard dataset[1,7] . .	36
3.21	Example of non-text or ham images from our dataset taken from standard dataset[1,7]	36
3.22	Example of text images created by us with simple background	37
3.23	Example of text images created by us with little complex background	37
3.24	Example of text images created by us with complex background	37
3.25	Accuracy vs threshold for decision point graph of training set	39
3.26	(a) Original image (b) Example of ad policy used by Facebook	41
3.27	Binary histogram, b of a text image	41
3.28	One cycle, f of the binary histogram of Fig. 3.27	41
3.29	Convolution of b and f of a text image	43
3.30	Three examples for text line detection	44
3.31	Two examples of horizontally repeated regular pattern images	44
3.32	D_p vs Images from testing dataset	45
3.33	D_p vs Text font size	45
4.1	Results of (a) original (b) Binary image (c) Histogram (d) Autocorrelation	47
4.2	Two examples of text lines appearing vertically	47
4.3	(a)Rotated caption text while embedding(b)Rotated caption text after embedding . .	48
4.4	Scene text with different text alignment	48
4.5	Skew document text image	48
4.6	Flowchart of detection by Hough transformation	49
4.7	Example of Hough transformation-based method (a) Rotated text image (b) Corre- sponding binary image (c) Corresponding edge map; E (d) Hough lines of the first highest 5 peaks of the Hough Space	49
4.8	False positive results for detection method by Hough transformation	51
4.9	N_L vs Images from testing dataset	51
5.1	Flowchart of proposed detection system	52
5.2	Text candidate binary image generated by (a) Method in[20] (b) Method in[21] (c) Method in[15] (d) Method in[26]	55
5.3	The flowchart for the comparison system	56
5.4	Some positive examples of text images of different languages from our dataset	57
5.5	Some positive examples of text images with document degradation from our dataset .	58
6.1	One example of combined text images	59

List of Tables

2.1	Text properties in images[53]	7
2.2	Published work on TIE in 1994-1998 [47]	8
2.3	Published work on TIE in 1999-2000 [47]	9
2.4	Published work on TIE in 2001-2002 [47]	10
2.5	Published work on TIE in 2004-2012 [53]	10
3.1	Categorization of Training Dataset	38
3.2	Categorization of Testing Dataset	38
3.3	Result for training dataset implemented in MATLAB	39
3.4	Result for testing dataset implemented in MATLAB	39
3.5	Confusion matrix for the testing dataset	40
4.1	Result of text image detection by Hough transformation for testing dataset implemented in MATLAB	50
5.1	Result for the first step of the proposed detection system experimented on testing dataset	53
5.2	Result for the second step of the proposed detection system experimented on testing dataset	53
5.3	Result for the proposed detection system experimented on testing dataset	54
5.4	Confusion matrix for the testing dataset	54
5.5	Text image detection performance comparison of different methods for testing dataset, where Proposed TE is proposed Text Extraction, TDR is Text Distribution Representation, TID is Text Image Detection Step, HLE is Hough Line Extraction and F is the Filtration of the proposed algorithm implemented in MATLAB	57
5.6	Text paragraph image detection performance comparison of different methods, where we use ‘ - ’ if not reported and Comparison is evaluated by the Processing time of Proposed Method to the compared method	58

Abstract

Internet offers a broad platform for people to share information and opinions. Illegal or sensitive commentaries in written form are blocked easily by text filters. However, it is difficult to automatically filter out those articles embedded and propagated via images. Among the large number of images, in order to prohibit the dissemination of those commentaries, detecting whether an image contains a sufficient amount of words provides convenience to the government. In this thesis, we propose a detection system to determine whether an image contains paragraphs or not. First of all, we propose a Histogram based method to filter out the images having text paragraphs in horizontal orientation and then propose a method based on Hough Transformation to detect text paragraphs in arbitrary orientation from the images without paragraphs. To achieve a better performance and detect text images with text of arbitrary orientation on images, we propose the detection system by combining the two proposed methods. To imitate the scenario, we construct a new dataset covering more than 2000 images of with and without paragraphs. Extensive experiments on the dataset demonstrate the effectiveness and practicability of the proposed detection system.

Chapter 1

Introduction

1.1 Motivation

Internet spreads all kinds of messages swiftly every day, even conveys defamation or illegal commentaries. How to prohibit the dissemination of illegal information is non-trivial for the government and the internet service providers. Such type of information in written form can be easily blocked. However, if a person takes a screenshot of the sensitive article and saves it as an image, or embeds the whole article in an image, it is difficult to be detected automatically and in a very fast manner.

Compared to the images without embedded articles, the proportion of images with embedded articles is very small, let alone those embedded with slander or illegal propaganda. Therefore, to narrow the scope of the targeted images, it would be necessary to develop a detector for the government to filter out images embedded with sufficient text. For simplicity, the image embedded with/without sufficient text is termed as text/non-text image in the following discussions. Some examples of text images, taken from internet are shown in Fig. 1.1.

Figure 1.1: Some examples of text image

Automatic text detection in unconstrained environments plays a central role in many computer vision applications as a first crucial step towards text recognition. Over the past years many algo-

rithms have been proposed to address the challenge in automated method to detect and localize the text present on an image. These techniques are pixel basis and each pixel has independent statistics. The advantage of this pixel basis approach is providing pixel-level accuracy and detecting more subtle details of the text. On the other hand, the computational complexity is increased and the higher level classification of the pixels is not considered.

Most of the existing text detection methods target on detecting mainly scene text in images. In Scene text images the intensity or color contrast of the text is almost same as the background and the text may not be put in paragraph or article. All these methods detect text by extracting the features of text like width, height, stroke or texture of text from the image. The Stroke Width Transform, [15] algorithm is one of the recent text detection techniques and has been widely used by many researchers. The Stroke Width Transform algorithm aims to compute the value of each pixel in an image from gray-level scale to the width of a stroke which has the highest possibility containing this pixel.

In this thesis, we are aiming at developing an algorithm to automatically detect images embedded with paragraphs, which is the first and key step for blocking transmission of illegal information via images and whether the content of the text is sensitive or not is beyond our concern. Moreover, in our target images the color contrast between the text and the background is high; so that readers can easily read the text on images and will not be distracted unless like some posters, where text is also embedded on images, but with inconsistent manner. Moreover, as millions of images are uploaded in internet every day, where very few may be embedded with articles, let alone sensitive one and all the existing methods are very time consuming. So the existing techniques for text detection followed by OCR are not suitable for the huge amount of images uploaded into the web. Hence a very fast with high performance algorithm is required in server-side to block those small number of target images. So, we propose a detection system to determine whether an image contains paragraphs or not.

Our proposed detection system consists of two steps:

1. Histogram-based method detects whether an image contains horizontal text lines. Images with horizontal text lines are detected as text images, while images without horizontal text lines will be fed into the Hough transformation-based method.
2. Hough transformation-based method detects whether an image flowing through Step1 contains non-horizontal text lines. Images with non-horizontal text lines are detected as text images, while images without non-horizontal text lines are detected as non-text images.

1.2 Objectives and Contributions

Automatic text detection on an image in unconstrained environments plays a central role in many computer vision applications as a first crucial step towards text recognition. Over the past years, many algorithms have been proposed to address the challenges in detecting and localizing the text presented on an image of various conditions of large variations in language, font, color, scale and orientation in complex scenes.

While many algorithms have been proposed on text image detection, the main objectives of this project is:

(i) to achieve an automated and scalable solution for the detection with a high accuracy rate and a fast processing speed.

(ii) to facilitate internet service providers and government agencies to block slander by images.

The main contribution of this project is :

Existing methods are not designed for detecting text in paragraphs or articles. They are time consuming and not considering the properties of text paragraphs. Therefore, these techniques are not suitable for our target images. Hence in this thesis, we propose a method to block images with sufficient text.

We conduct all the experiments on MatLab R2010a, in a desktop of Intel(R) Core(TM) i5-4570 3.20GHz CPU and 8.0GB RAM. Moreover, we implement the proposed method in C# also to check the optimal processing time.

1.3 Organization of the Thesis

The thesis is organized as follows:

- Chapter 2 is the Literature Review which briefly introduces the history of traditional techniques, and also gives explanations about the algorithms that will be applied during this project.
- Chapter 3 describes the Text image detection method of text in horizontal orientation in details.
- Chapter 4 describes the challenges of text alignment or orientation in Text images and another method is proposed to detect Text images with arbitrary text alignment.
- Chapter 5 presents the final detection system by combining the two proposed methods.
- Chapter 6 presents the future works of this project followed by the conclusions in the last chapter.

Chapter 2

Literature Review

This chapter details the literature survey on the existing text detection and recognition techniques and identifies the strength and weakness of the different algorithms available. It also briefly covers the topics which can be used or helpful in the project.

2.1 Text in Images

The images embedded with text contain useful information for many applications, such as; automatic annotation, indexing, and structuring of images. To extract these information from the images includes text localization, text detection and recognition. However, there are lot of challenges in it because of the variations of text font, size, alignment and the low contrast between the text and the background and text in complex background.

Some example of text images are shown in Figures 2.1- 2.4.

There are mainly three types of text that can be embedded in images: caption text, which is artificially overlaid on the image(shown in Fig. 2.3); scene text, where text exists naturally in the image(shown in Fig. 2.4); and text in document images, which exists in document images with homogeneous background, but affected by illumination, focus or different types of document degradation(shown in Fig. 2.1 and 2.2).

2.1.1 Text in Document Images

Document images with complex colored and textured backgrounds are very popular in magazines, advertisements and web pages. For usability, it is very important to build a fast and robust text recognition and detection system for these images. In general approach, binarization by adaptive thresholding is used to extract the text from the document images. But it is very challenging problem to find an optimal threshold to preserve meaningful information and eliminate unnecessary one. Moreover, Texture-based methods also are proposed as a solution to this, where text is assumed to represent a different texture than non-text. In this method, text is considered as a periodic texture, because the characters on a text line form a more or less periodic structure in the horizontal direction, while the text lines, in turn, form a periodic structure in the vertical direction.

2.1.2 Text in Scene Images

Text part of the image captured by camera is Scene text. Some examples of Scene text are street signs, billboards, and text on trucks and writing on shirts. The nature of scene text is unpredictable as compared to caption text, which is more structured. As the nature of scene text is unpredictable, it poses lots of challenges. Out of these, arbitrary orientation is more challenging as it is not as easy as processing straight text lines.

The London Advertiser.

Figure 2.1: Grayscale document images

Figure 2.2: Multi-color document images

2.1.3 Caption Text in Images

The artificially superimposed text on images or videos at the time of editing is Caption text. It is also known as Overlay text or Cut line text, as shown in Fig. 2.3 . It is a powerful source of high-level semantics. For indexing, retrieval and summarization, many text detection techniques have been proposed to detect, segment, and recognize it automatically. There are lots of applications for extracting caption text from images; such as: in sports video is very useful for the creation of sports summary, highlights etc. and in news, advertisement images and video also. It includes text of arbitrary orientation, arbitrary size, moving text, rotating text, growing text, shrinking text.

Figure 2.3: Images with caption text

Figure 2.4: Scene text images

2.2 Text Properties in Images

Text on images have some particular properties or features from background, based on which the text can be extracted from the background of the image. These properties have been used in many text detection techniques. The variations of text in images mainly depend on these properties. The properties are shown in Table 2.1. The main properties of text on images are[53]:

1. Size:

The range of font size variation could be diverse. As Scene text is very uncertain, so it can have different text font and size.

2. Alignment:

Table 2.1: Text properties in images[53]

Property	Sub-classes
Size	the range of font size variation could be diverse
Alignment	Horizontal or vertical Straight line with arbitrary direction
Color	Gray Single or multicolored Text
Edge	Strong edges between the text boundary and background
Distortion	due to changes in camera angles, some text may carry perspective distortions

Usually the text on images lies horizontally, although sometimes it can be appeared in non-planner also. As Scene text is very uncertain, so it can have alignment in any direction and any geometrical deformation.

3. Color:

One of the striking properties of text on images is the uniform color of text. This property is used for text detection based on connected component-based approach. But some images with complex color document images and video images can contain multiple color of text or different colors within one word.

4.Edge:

For readability, most text on images have strong edges between its boundaries and background.

5. Distortion:

Due to changes in camera angles, some text may carry perspective distortions that dramatically affect extraction performance.

2.3 Text Information Extraction

Before proceeding further, we discuss the main problem and challenges of Text Information Extraction. The most challenging problem of TIE is that the input image of TIE can be colored or gray, the text may or may not move etc. Depending on these scenarios TIE can be divided into five categories:

- (i) detection
- (ii) localization
- (iii) tracking
- (iv) extraction and enhancement
- (v) recognition (OCR) (Fig. 2.13).

In TIE system, Text detection basically defines to determine the presence of text on an image or a sequence of images. Text localization is used to determine the location of text in the image and figure out the location by generation some rectangular boxes covering the text. Mainly in a video or sequence of images, text tracking is used to detect text.

In order to facilitate text recognition, the text needs to be extracted from the background. As the extracted text regions usually have low-resolution with non text noises, the extracted text is recognized by OCR technology. Table 2.2, Table 2.3 and Table 2.4 shows some of the published work on TIE.

Table 2.2: Published work on TIE in 1994-1998 [47]

Author	Year	Approach	Features
Ohya et al. [28]	1994	Adaptive thresholding and relaxation operations	Color, scene text localization and recognition
Lee and Kankanhalli [29]	1995	Coarse search using edge information, followed by connected component (CC) generation	Scene text (cargo container), localization and recognition
Smith and Kanade [30]	1995	3×3 filter seeking vertical edges	Caption text, localization
Zhong et al. [31]	1995	CC-based method after color reduction, local spatial variance-based method	Scene text (CD covers), localization
Yeo and Liu [32]	1996	Localization based on large inter-frame difference in MPEG compressed image	Caption text, localization
Shim et al. [33]	1998	Gray level difference between pairs of pixels	Caption text, localization
Jain and Yu [34]	1998	CC-based method after multi-valued color image decomposition	Color (book cover, Web image, video frame), localization
Sato et al. [35]	1998	Smith and Kanades localization method and recognition-based character extraction	Recognition

Table 2.3: Published work on TIE in 1999-2000 [47]

Author	Year	Approach	Features
Chun et al. [36]	1999	Smith and Kanades localization method and recognition-based character extraction	Caption text, localization
Antani et al. [37]	1999	Multiple algorithms in functional parallelism	Scene text, recognition
Messelodi and Modena [38]	1999	CC generation, followed by text line selection using divisive hierarchical clustering procedure	Scene images (book covers, slanted) localization
Wu et al. [39]	1999	Localization based on multi-scale texture segmentation	Video and scene images (newspaper, advertisement) recognition
Hasan and Karam [40]	2000	Morphological approach	Scene text, localization
Li et al. [41]	2000	Wavelet-based feature extraction and neural network for texture analysis	Scene text (slanted), localization, enhancement, and tracking
Lim et al. [42]	2000	Text detection and localization using DCT coefficient and macro block type information	Caption text, MPEG compressed video, localization
Zhong et al. [43]	2000	Texture analysis in DCT compressed domain	Caption text, JPEG and I-frames of MPEG, localization

2.4 Challenges in Text Detection

Automatic text detection on an image in unconstrained environments plays a central role in many computer vision applications as a first crucial step towards text recognition. Over the past years,

Table 2.4: Published work on TIE in 2001-2002 [47]

Author	Year	Approach	Features
Jung [44]	2001	Gabor filter-like multi-layer perceptron for texture analysis	Color, caption text, localization
Chen et al. [45]	2001	Text detection in edge-enhanced image	Caption text, localization and recognition
Strouthopoulos et al. [46]	2002	Page layout analysis after adaptive color reduction	Color document image, localization

Table 2.5: Published work on TIE in 2004-2012 [53]

Author	Year	Features
Ou et al.[54]	2004	Edge intensity, text localization
Kim et al.[55]	2004	Text extraction, hierarchical features
Kim et al.[56]	2005	Image intensities, text localization
Hu and He[57]	2008	Edge detection, text extraction
Bui et al.[58]	2009	Topographic map, convolutional, neural network, text detection
Pan et al.[59]	2010	Text detection, feature extraction, coarse-to-fine, classification
Lee et al. [60]	2010	Edge constraint, text detection
Minetto et al.[61]	2010	Text detection, multi-resolution, image segmentation
Neumann and Matas[62]	2010	Text localization, maximally stable, extremal regions
Pan et al.[63]	2011	Conditional random field, connected component analysis
Chen et al.[14]	2011	Text detection, maximally stable, extremal regions
Le et al.[64]	2011	Text localization, parallel edge feature, mean shift clustering
Pan et al.[65]	2011	Text detection, stroke segmentation, conditional random field
Epshtein et al[15]	2011	Text detection, stroke width transform
Zhou et al.[66]	2011	Multilingual, scene text detection, HOG, MG, LBP
Neumann and Matas[62]	2012	Extremal regions, text localization

many algorithms have been proposed to address the challenges in detecting and localizing the text presented on an image of various conditions, as shown in Fig. 2.6, of large variations in language,

Figure 2.5: TIE System [47]

font, color, scale and orientation in complex scenes.

Figure 2.6: Some example of challenges in Text Detection

The main challenges of text detection on images are: (1) the diversity of the texts and (2) the complexity of the backgrounds. Although text on images appears in a linear form [5] and is expected

to have similarities for uniformity and readability, including similar size, aspect ratio and gaps between the letters and words, but on the other hand, repeated patterns (such as windows and barriers) and random clutters (such as grasses and leaves) may be similar to texts, and thus lead to potential false positives. Moreover, detecting texts of arbitrary orientations in complex background remains another challenge for most practical system.

2.5 Text Detection Techniques

Over the past years, many algorithms have been proposed to detect text presented on an image. The existing text detection techniques on images have been proposed for specific applications including reading license plates, book sorting, visual classification of magazines and books, reading freight train IDs, road sign recognition, detection of danger labels, and reading signs in warehouses, page segmentation and analysis, address block location and recognition, and content-based image/video indexing[73]. The challenging problems of text detection in these applications are dealing with the diversity of text font, size, orientation and language or often image which is affected by variations in scene and camera parameters such as illumination, focus, motion, etc.. Three competitions; ICDAR 2003 [11], ICDAR 2005 [12] and ICDAR 2011 [13] have also been organized for this text localization task and various techniques have been presented. In general, these techniques can be classified into four main categories: edge-based, texture-based, stroke-based and connected component-based. Edge-based methods locate text candidate regions by edge detector and morphology operations [20, 21]. These methods are efficient with images containing simple background, but it fails when images contain complex background with rich edge information. Texture-based methods distinguish the text from background by detecting the distinct textural properties of text. These methods detect text accurately but take longer time [22, 23]. Stroke-based methods detect text by extracting stroke components with similar features. The Stroke Width Transform [15] algorithm is one of the recent stroke-based text detection techniques. It aims to extract text characters with stable stroke widths and considers components with low stroke width variations as text components. As text on an image has similar color and certain size, shape and spatial alignment constraints, connected component-based methods [24,25,26] detect text by extracting a group of neighbouring pixels with similar property. More recently in connected component-based methods, Maximally Stable Extremal Region (MSER) is applied in many text detection techniques [14,26]. By integrating several improvements over traditional MSER based methods and distance metric learning techniques, Yin et al [26] proposed a fast and effective pruning algorithm to detect scene text by extracting MSER as character candidates using the strategy of minimizing regularized variations.

Existing techniques proposed for detecting text on images are always focusing on one single type of text as mentioned in section 2.1 because of different properties. Moreover, as most of the existing text detection techniques are used in character recognition and localization, the methods of these techniques are mainly pixel-based. The advantage of this pixel-based approach is providing pixel-level accuracy and detecting more subtle details of the text. But the computational cost is high.

Among these techniques, most of them use some coarse-to-fine steps to localize text eventually. These steps can be represented in a flowchart as shown in Fig. 2.7. The first step is to select candidate text regions by using a proper approach, after which, its common to use a coarse classifier to group letter candidates and filter out the rest as noises. The final move is clustering all the letter candidates into text lines, and removing the isolated parts by using a fine classifier.

2.5.1 Edge-based Methods[47]

Edge-based methods mainly target the image with high contrast between the text and the background. As we have mentioned in Section 2.1, most text on images have strong edges between its boundaries and background, so in these methods text are extracted by detecting the edges and grouped the edges based on text features. After that non text regions are eliminated by some filters. Mainly Canny,

Figure 2.7: The flowchart of traditional text detection algorithms

Sobel operators are used to detect text edges and after that a morphological or smoothing operators are used for grouping.

Smith and Kanade [30] implement a horizontal differential filter of dimension 3×3 and apply on the input image to detect the threshold and find vertical edges. After generating the edge map, small edges are eliminated and neighbouring edges are grouped together by bounding box. To filter out nontext regions the aspect ratio, size of each bounding box is considered.

Hasan and Karam [40] proposed a morphological technique for text extraction. The input image is converted from RGB scale to gray scale by the equation:

$$Y = (0.299R + 0.587G + 0.114B) \quad (2.1)$$

where R, G, and B are the red, green, and blue components, respectively. After the gray scale conversion, edges are identified by a morphological operator. After that an optimum threshold is evaluated to obtain a binary edge image. Dilation is used to group spatially close edges and erosion is used to remove or eliminate small components or non-text regions. As text has fixed size, aspect ratio, thickness and gray-level homogeneity, non-text regions can be filtered out from the gray image. Many researchers have implemented this approach. Fig. 2.8(a) shows an input image and 2.8(b) shows gray-scale image of the color image.

Figure 2.8: Example of (a) color image (b) gray image

Another very popular edge-based method is Chen et al. [45]. In this method, Canny operator is used to detect edge from the input image. After that one small window for one edge point is designed to estimate scale and orientation. From this scale information text is enhanced and text edges are

connected by dilation to convert into clusters. After that some filters are applied on the clusters to filter out the non-text regions depending on the aspect ratio of horizontal-vertical and height. For scale estimation two Gabor filters are used ; an edge-form filter and a stripe-form filter and generate input features and after that a neural network is used on the output of these filters. An appropriate scale is decided to enhance the edge information.

2.5.2 Texture-based Methods[47]

In texture-based methods, text are detected by extracting the distinct textural properties of text from the background. To detect textural properties of text regions in an input image, some filters are used; such as Gabor filter, Wavelet, FFT, spatial variance etc.

Among the texture-based methods, Zhong et al. [31] first converts the input image into gray image and use the spatial variation in the image to detect text regions with high variance. To compute the spatial variance for neighbouring pixels a horizontal window is designed of dimension 1×21 . Canny edge detector is applied and horizontal edges are identified and small edges are joined into longer lines. The edges with opposite directions are merged together to get the text line from the edge map.

Another similar method to Zhong et al. [31], Park et al. [48] has proposed to localize car licence plate. In this method, a time delay neural network (TDNN) is used as a texture classifier in the HSI color space and the horizontal variance is used for the detection. A small window is designed on the input image and HSI color values are received by two TDNNs, used as horizontal and vertical filters. From this window it is detected text and non-text regions. License plates are located by bounding boxes after these filters.

Sin et al. [49] frequency features of text by calculating the number of edge pixels in horizontal and vertical directions Fourier spectrum to detect text regions in real scene images. It is assumed that text regions are rectangular and rectangles are detected after edge detection followed by the Hough transform.

Mao et al. [50] apply Wavelet transform to detect text. In this method, local energy variations are extracted at different scales by Harr Wavelet decomposition. After this, the image is converted to binary image by getting the threshold from the local energy variation and text regions are detected from the binary image at several scales by merging the connected components depending on the geometric text attributes such as size and aspect ratio.

Wu et al. [45, 46] used a multi-scale texture segmentation scheme to segment out the text from the image. The segmentation of text includes nine second-order Gaussian derivatives. After extracting candidate text regions a non-linear transformation is applied to the image and for each pixel local energy estimation is calculated. Depending on this estimation pixels are clustered using the K-means algorithm. This texture based method is divided into 5 steps:

- (i) stroke generation
- (ii) stroke filtering
- (iii) stroke aggregation
- (iv) chip filtering
- (v) chip extension.

The process time in this method is 10 seconds for a 320240 sized image and it misses text with small sizes. The text localization rate was higher than 90%, but experimented on the text having more than 10 pixels, while the false positive rate was 5.6 %.

Fig. 2.9 shows examples of different steps of this method.

2.5.3 Stroke-based Methods

The stroke of a text is a continuous part of an image that forms a band of nearly constant width. The Stroke-based method aims to compute the value of each pixel in an image from gray-level scale to the width of a stroke which has the highest possibility containing this pixel.

Figure 2.9: Example of different step by step process for texture segmentation[45,46]

Stroke Width Transform

The Stroke Width Transform, [15] algorithm is one of the recent text detection techniques and has been widely used by many researchers. To recover stroke from this method, the edges are computed

Figure 2.10: Illustration of finding the line

using Canny edge detector on the gray image of the input and calculated the gradient direction dp of each edge pixel. If a pixel p lies on one edge of a stroke, its gradient direction dp must be approximately perpendicular to the orientation of that stroke. Along that gradient direction, if another edge pixel q is detected, and the gradient direction of that pixel, dq , is roughly opposite to dp , this line with two ends p and q can be considered as a transect of a stroke. This process of finding

the line can be illustrated by Fig. 2.10.

After finding the line the value of each pixels between p and q along the direction dp , is the distance between the two ends p and q , which indicates the width of that stroke. This can also be shown in the Fig. 2.10 above. If for an edge pixel p there is no matching pixel q or dq is not the opposite of dp then the ray is discarded. After finding all the lines, each line is represented by a group of pixels and a value which indicates the distance between the two ends. But if a pixel is involved in two transects, the value assigned to it should be the lower one. This can be explained by using the Fig. 2.11.

Figure 2.11: Pixel is assigned with lowest value

From Fig. 2.11, we can see that the red pixel is involved with one horizontal and one vertical connected component passing through it. The length of the shorter connected component is the actual width of this stroke. Finally the output of the SWT is an image where each pixel contains the width of the most likely stroke it belongs to.

2.5.4 Connected Component-based Methods

In the connected component-based methods, text are extracted by grouping small components into successively larger components. To identify the components, a geometrical analysis is used to filter out non-text regions or components based on spatial arrangement.

Connected component-based method is very popular in the field of text detection, as it is relatively simple. This method can be divided into four stages:

- (i) Pre-processing, eg. noise reduction, color clustering etc.
- (ii) Connected Component generation
- (iii) Removing the non-text components
- (iv) Component clustering.

After getting the connected components of the probable or candidate text regions, many filters are designed to filter out the non-text regions based on the text properties, because of this several thresholds are defined. Moreover these values are dependent on images or sequence of images.

Ohya et al. [28] presented a text detect and recognize method based on connected component-based approach, mainly for scene text. Four stages below are mentioned in this method;

- (i) Conversion to binary image by local thresholding
- (ii) Detection of text candidate regions by gray-level difference
- (iii) Compare the text candidate regions with the standard text pattern from the dataset
- (iv) update the similarities by relaxation operation.

This method has a lot of disadvantages such as text alignment, the color of the text (monochrome) and documents with variations in illumination. According to this method the recall was 85.4% and accuracy rate was 66.5%.

Lee and Kankanhalli [29] proposed a connected component-based method to detect and recognize text on images of different size, font, color and various lighting conditions. In this method edge of the input image is detected before the connected components detection. The text edges is identified from the edge map by detecting the difference between adjacent pixels. After that local threshold is used to select each text candidate.

Zhong et al. [31] proposed a connected component-based method based on color reduction. The color histogram in the RGB color space of the input image is extracted and quantized by using the peaks in the histogram. The main theory of the detection is that text occupies the same color domain and significant portion of the image. As text occupies a fixed size and shape, some filters were designed to eliminate or remove non-text regions based on aspect ratio, size, area, diameter and spatial alignment. This method was mainly implemented on CD images and book cover images.

Shim et al. [33] proposed a connected component-based method based on the homogeneity of intensity of text regions in the input image. In this method the input image is converted to gray image and then pixels with similar gray levels are grouped into clusters. Non-text regions are removed based on the size, significantly large regions and text candidate regions are extracted based on the gray level contrast. Then text are verified based on the size, area, fill factor, and contrast of the text candidate regions. Then text strings are extracted from neighbouring text regions. The processing time per image of this method is 1.7 seconds and miss rate range is 0.29% to 2.68% .

As text on images has the same color and size, Lienhart et al. [51, 52] proposed an algorithm to extract text based on connected component of text regions with the same and similar color. This method mainly extracts caption text in video or image sequence. The accuracy of this method experimented on 2247 images with caption text is 86% to 100% . This accuracy is high as because, this method is implemented only on images with caption text. Fig. 2.12 shows an example of this method.

2.6 Optical Character Recognition (OCR)

Optical character recognition (OCR) is the process of converting scanned or photograph images of typewritten or handwritten text (numerals, letters, and symbols), into machine-encoded/computer-readable text. Moreover, many multimedia sources provide images and video that contains visible text that can potentially be detected, segmented, and recognized by OCR. These types of recognition are very important for advanced multimedia information retrieval systems.

There are three stages in OCR system as shown in Fig. 2.13:

1. document digitization
2. character/word recognition
3. output distribution

In the first stage of this process, the document are captured by scanner and converted into document images. Because of the recent scanners of advanced technology, document images can be captured with high resolution. So modern feature based OCR methods require image spatial resolutions of at least 200dots per inch (dpi) The second (and the most interesting) stage is responsible for character and/or word recognition in document images. The process involves four operations:

1. optional image analysis: image quality assessment, text line detection, word and character extraction, etc.
2. optional image enhancement : removing speckle and other image noise, filling holes and breaks, etc.
3. character/word recognition usually based on their shapes and other features
4. optional contextual processing to limit the feature search space

In the third (final) stage, the output interface communicates the OCR results to the outside world. For example, many commercial systems allow recognition results to be placed directly into spread sheets, databases, and word processors. Other commercial systems use recognition results directly in further automated processing and when the processing is complete, the recognition results are

Figure 2.12: The output result of different stages in the method by Lienhart et al [51,52]

discarded. In any case, the output interface is vital to the OCR systems because it communicates results to the user and program domains outside of the OCR system.

2.7 Image-based Spam Filtering

To escape from email filters some sensitive or illegal, fake information can be spread through images by embedding the information in text format in the image and attaching the image in the email. This kind of images is known as image-based spam or spam image.

In recent years, spam phenomenon is continuously growing. So, to block this kind of spam mail is necessary for the internet service providers. Currently, there are several modules in anti spam filters mainly aiming at different features of the mails. Many research have been proposed to design these modules by categorising text mainly in the body field of the email and analyse the information after recognizing the text on the mail.

Several methods [1][2] have been proposed to block or detect spam images by extracting text from the image with some tools like Optical character recognition (OCR) which could then be processed simultaneously while analysing the text in the body of the email or some text detection techniques in images can be used to take out the information from the image. Some researchers[3,4,5] have proposed to detect spam images by detecting generic low level image features, such as number of

Figure 2.13: OCR System [47]

colors, text area, image aspect ratio, color coverage etc.

Moreover, some works [6][7][8][9] also tried detecting spam images by identifying adversarial attempt on the images. To make this process of detection faster, Dredze et al. [10] introduce Just in Time (JIT) feature extraction by using JIT decision tree. This method extract features from the image with some classifiers and it is very fast with high accuracy.

From this literature survey, we have found that the main aim of these methods is to block spam images which contain sensitive or illegal words by extracting the whole text from the image with some tools. But in our project we have targeted the text images by detecting the paragraph or article on the images without checking the sensitivity of the words on the images. As we have targeted different types of images as compared to spam images, so the aim of this thesis is also different from these aforementioned techniques. This technique can also be employed as the initial step to filter out all text images for sensitive word detection step.

2.8 Color Contrast

Contrast of an object means the features of the object , such as color, luminance that can be distinguished from the background of the image. In practical applications, mainly the contrast is determined by the color and brightness of the object in the same field of view. In the case of contrast of text in an image, the text is easily readable if the color contrast between the text and background is high because the human visual system is more sensitive to contrast. Contrast is also the difference between the shading of the printed material on a document and the background on which it is printed [16].

There are many theories of color contrast of foreground and background in an image, but to consider different colors of text on images we use a simple color contrast concept for our understanding; ie color wheel. The color wheel - primary colors are: red, blue and yellow. Every other color is made

up from these colors. The colors opposite to each other on the wheel are considered "complimentary" colors. The figure of color wheel is shown in Fig. 2.14.

Figure 2.14: Color-wheel

According to this concept an image has good color contrast if the colors of the background and the object are almost opposite of each other in the color-wheel. In the image in Fig. 2.15, if the colors are almost opposite of each other then the color contrast is good. The image is loud and grabs attention.

Figure 2.15: Example of an image using color-wheel

For creating the dataset, we also used the concept of color wheel to embed text on images. Two examples from our dataset are shown in Fig. 2.16 to show the usage of color wheel concept in embedded text and background of these images.

2.9 Contrast Adjustment

It is very difficult to get an image with the right contrast. It is very necessary in many applications, such as in art, detection, recognition, or improving photographs. In our project also, the right contrast of the image is very important to extract the text from the background.

Histogram processing is one answer for computationally adjusting the contrast, and can also be applied to a number of problems involving color balance or tone transfer. This section focuses on different methods for post-processing an image to modify its underlying histogram, affecting contrast and color balance.

This part will cover histogram equalization and an advanced technique called adaptive histogram equalization, both intended for increasing the contrast of an image. The cumulative distribution

Figure 2.16: Example of two images from our dataset using color-wheel

function (CDF) of an image histogram does not form a straight line; meaning that the pixel values are not equally likely to occur (since the CDF is the integral of the PDF). Most images do not have flat CDFs. The increase of the contrast can be done by having a flat CDF. The process of flattening the CDF is called histogram equalization. As an example:

Figure 2.17: Example of histogram equalization

The resulting image has particularly dark islands and a bright sky, which is not visually appealing, but the detail within the buildings is improved significantly [17].

We used the existing advanced version of histogram equalization, adaptive histogram equalization, with the assumption that the image varies significantly over its spatial extent. The algorithm divides the image into smaller tiles and applies histogram equalization to each tile. After that it interpolates

the results. An example is shown in Fig. 2.18 from our dataset.

Figure 2.18: (a) Gray image (b) Result of contrast adjustment

2.10 Periodic Pattern Detection

Usually, a paragraph has a fixed gap between two consecutive lines and the height of each letter is constant. We observe the periodic or quasi periodic behaviour of the occurrence of text lines in text images. Autocorrelation and Fourier transformation are well known in the computer music literature, originally applied in the regime of pitch perception to do fundamental frequency estimation. They are also applied for analysis of rhythmic and metric structures.

2.10.1 Autocorrelation

Autocorrelation finds the correlation of a signal against different versions of itself time-shifted by various amounts. Each time-shift amount is called a lag time. The output of an autocorrelation is the correlation amount as a function of lag time. The maximum value will always be at a lag of zero, since a signal is always perfectly correlated with an exact copy of itself. Other peaks in the autocorrelation indicate lag times at which the signal is relatively highly correlated with itself; these can be interpreted as periods at which the signal quasi-repeats. In other words, autocorrelation is based on the idea that a quasi-periodic signal will resemble itself in the time domain when time-shifted by duration (nearly) equal to the period.

2.10.2 Discrete Fourier Transform

In mathematics, the discrete Fourier transform transfer a finite number of uniformly spaced samples into the sequence of coefficients of a finite combination of complex sinusoids, ordered by their frequencies, that has the same number of samples. Namely, the discrete Fourier transform converts the samples in their original domain to the frequency domain. In image processing, these samples can be the values of pixels along a row or a column of an image. Since the discrete Fourier transform deals with a finite number of data, it can be implemented with a computer, and a fast Fourier Transform algorithm is usually employed.

The list of N complex numbers $X_0, X_1, \dots, X_{N-1}$ is converted into a sequence of complex numbers with length N according to the discrete Fourier transform formula:

$$X_k = \sum_{n=0}^{N-1} (x_n e^{-i2\pi kn/N})$$

2.11 Morphological Image Processing

It is very difficult to convert a colored image to a binary image by using a threshold, as there are lots of distortion in the image due to the texture and noise. To get an appropriate threshold by reducing the noises, morphological image processing is used on the image[18]. It is basically a sequence of non-linear operations based on the morphological features of the input image. It is not related to the numerical value of the pixels, it depends on the relative or cumulative ordering of the pixels. Because of this it can be used on binary images. To use morphological operations on gray image, the absolute value of the pixels should not be used. Instead, one structuring element is used to detect the entire image and compare the neighbourhood of pixels. An example for such operation on three different position is shown in Fig. 2.19.

Figure 2.19: Example of checking an image with a structuring element [18]

The result of morphological operation applied on a binary image is also a binary image and the structuring element with the following properties is a small matrix of pixels with values 0 or 1.

1. The size of the structuring element is the dimension of the matrix.
2. The pattern of ones and zeroes implies the shape of the structuring element.
3. An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

2.12 Summary

This Chapter gives the basic introduction of text types that can be embedded on images. It also gives a concept of color contrast and contrast adjustment for embedding text on images such that the text is easily readable. In this chapter, a comprehensive review of the existing algorithms for text detection, recognition and localization has been presented. Text detection techniques employing various design philosophies and exploiting different characteristics and type of text have been presented in detail. It also gives explanations about the spam images and some algorithms to filter out these spam images. Moreover, it gives a broad understanding of periodic and quasi periodic signals. It also gives a brief overview of morphological operations in image processing which can be used to estimate text proportion in text images. With all these basic understandings, we can proceed to the methodology in the next chapter.

Chapter 3

Text Image Detection by Histogram

3.1 Proposed Method

Fig. 3.1 shows our proposed text image detection by histogram to detect text image of text in horizontal direction. This method consists of three parts: text extraction, text distribution representation and text image detection. Text extraction is applied to get a binary image from the input image to differentiate text from the background. After that a histogram extraction technique is proposed to represent the text distribution. Finally in order to detect text paragraph, we detect periodic pattern of text distribution in the histogram by a newly designed step graph method from autocorrelation concept.

3.1.1 Text Extraction

For a faster speed, we normalize each input image by resizing its height to 500 pixels while remaining the ratio unchanged. In order to extract potential text pixels and separate them from the background, we first convert the color image into the gray image and then apply contrast adjustment by using an existing advanced version of histogram equalization, adaptive histogram equalization for contrast improvement to the gray image [8]. Contrast adjustment on the gray image is used to enhance the contrast between potential foreground text and background image. The enhancement is done in order to set an optimum threshold level to differentiate it into two classes and convert to the binary image.

After color contrast enhancement, we convert the gray image into a binary image to distinguish potential foreground text pixels from the background. We employ Otsus method [27] for the binary conversion and set 0 to pixels of the foreground and 1 to those belonging to the background (as shown in Fig. 3.2 (b)). The foreground of the binary image contains both text potential regions and false positive text potential regions (appearing big dark regions as we can observe in Fig. 3.2(b)). Therefore, we propose an algorithm to remove the false positive text regions and extract the potential text regions. Several text detection techniques have been proposed using machine learning tools [1]. There is still a growing interest in developing advanced method in this field. But the main problem in obtaining efficient and robust text detection using machine learning tools is to lower the computational processing speed for recognizing text with high accuracy. In order to solve this, we propose a localization or verification scheme that extracts text blocks from the foreground of an image with a fast speed and a low rejection rate.

Fig. 3.3 is an example of two target images of text and non-text images. After resizing and contrast adjustment, the binary images of the inputs of text and non-text images are shown in Fig. 3.4.

An important cue for the text is that it appears in a line form [5]. Text on a line is expected to have similarities for uniformity and readability, including similar stroke width, letter width, height

Figure 3.1: Flowchart of text detection method by histogram

Figure 3.2: (a) Original image (b) Binary image

and spaces between the letters and words. Because of the spaces between words, the number of the pixels that could be grouped together to form a connected component is limited.

Let I_b be the binary image and $I_b(x, y)$ denotes the pixel value having x and y as the coordinates in the binary image I_b . Here we define the continuous connected component of I_b as a set of horizontal connected pixels denoted as S_i^H (as shown in Fig. 3.5). In S_i^H , i indicates the i^{th} connected component of all horizontal connected component.

Therefore, we define the sets S^H as: $S^H = (S_1^H, S_2^H, \dots, S_{n_1}^H)$ where, n_1 is the number of components in S^H .

Next, we propose to remove the big regions or dark regions with pixel value as 0 in binary images. After removing the big dark regions, we get the partial dark regions or part of text regions due to

Figure 3.3: Example of (a) a text image (b) a non-text image

Figure 3.4: Binary image of (a) the text image (b) the non-text image

the spaces between texts. It also includes some small non text regions. But, because of the constant size of text in a paragraph and spaces between text in horizontal direction, partial text regions are extracted in consistent manner. Whereas, the small non text regions are extracted in a distorted or inconsistent manner. We term the obtained image after removing the big dark regions as text candidate binary image.

Let I_b be the binary image with the height of H and width of W . $I_b(x, y) \in \{0, 1\}$ denotes the intensity of the pixel located at the coordinates (x, y) in I_b , where $x = 1, \dots, W$ and $y = 1, \dots, H$. We define the continuous horizontal connected component of I_b as a set of horizontal connected pixels denoted as S_i^H (see Fig. 3.5). Here, i indicates the i^{th} horizontal connected component in the image. S_i^H can be defined as

$$S_i^H = \{(u_{i1}, v_i), (u_{i2}, v_i), \dots, (u_{iq}, v_i)\} \quad (3.1)$$

where, (u_{ij}, v_i) indicates the coordinate of the j^{th} pixel in S_i^H and q is the total number of pixel in S_i^H .

Let I_T be the text candidate binary image and it's pixel value is represented as:

$$I_T(x, y) = \begin{cases} 1 & \text{if } \exists i : (x, y) \in S_i^H; x = u_{ij}; \\ & (u_{iq} - u_{ij}) \geq T_1 \\ I_b(x, y) & \text{otherwise} \end{cases} \quad (3.2)$$

An example is shown in Fig. 3.6 and Fig. 3.7 to illustrate the process for non-text and text

Figure 3.5: Horizontal connected component

Figure 3.6: Proposed text extraction from Non-text image example (a) Binary image (b) Horizontal connected components of binary image (c) Obtained text candidate binary image, where $T_1 = 2$, $i = 1, \dots, 12$.

examples respectively. In our implementation, we set $T_1=5$ to extract small partial text regions of all text font size. Fig. 3.8(a) and (b) show the text candidate images of the example images.

3.1.2 Text Distribution Representation

Based on our observation, we find that the most striking feature of the text line in a paragraph is the uniform distribution structure of its layout, i.e. text lines have similar length and height. Moreover, the gaps between two subsequent lines are constant through an article. Inspired by this observation,

Figure 3.7: Proposed text extraction from Text image example (a) Binary image (b) Horizontal connected components of binary image (c) Obtained text candidate binary image, where $T_1 = 2$, $i = 1, \dots, 12$.

Figure 3.8: Text candidate image of (a) the text image (b) the non-text image

we propose a histogram-based method to represent the text distribution accumulated row by row from the text candidate binary image. The histogram; $\mathbf{n} = (n(1), n(2), \dots, n(H))$ of the entire image can be estimated as:

$$n(y) = W - \sum_{x=1}^W I(x, y), y = 1, 2, \dots, H \quad (3.3)$$

The histogram of the images are illustrated in Fig. 3.9, where the x-axis represents the indices

Figure 3.9: Histogram for (a) text image (b) non-text image

Figure 3.10: Binary histogram for (a) text image and (b) non-text image

of rows in the sample image, and the y-axis computes the number of potential text pixels (with the value of 0) on each specific row. In order to remove the noise and differentiate the text lines from the gap between consecutive text lines, the histogram is further converted into a binary one; $\mathbf{b} = (b(1), b(2), \dots, b(H))$ which is denoted by:

$$b(y) = \begin{cases} 1 & \text{if } n(h) \geq M \\ 0 & \text{otherwise} \end{cases}, y = 1, 2, \dots, H \quad (3.4)$$

where an adaptive threshold M is decided by

$$M = \frac{\sum_{y=1}^H n(y)}{H}$$

We consider mean value of the original histogram as the threshold for the binary histogram conversion to find an adaptive threshold decided by the major lengths of potential text lines. As shown in Fig. 3.10, we plot the values of the binary histogram of the images and the x-axis represents the indices of rows in the binary image, and the y-axis shows the binary value, where value 1 indicates text lines and value 0 indicates gaps. . For a clear view, the zoomed plot of the first 100 rows of the binary histogram of the text image is shown in Fig. 3.11. As we can observe that the value of the rows which have text line is 1 and non-text is 0.

Figure 3.11: Zoomed (1st 100 rows) binary histogram for text image

3.1.3 Text Images Detection

After the text distribution representation step, in order to extract the features of a text paragraph, we detect the periodic or quasi periodic pattern of text lines in the paragraph. As binary histogram of text image has a quasi periodic signal in the space domain, we compute the autocorrelation on binary histogram to differentiate the images with text paragraph and non-text. The autocorrelation of binary histogram is represented by a vector $\mathbf{c}$ of dimension $1 \times H$.

$$\mathbf{c} = (c(1), c(2), \dots, c(H)) \quad (3.5)$$

where,

$$c(y) = \sum_{j=1}^{H-y} (b(j) \times b(j+y)), y = 1, 2, \dots, H$$

Fig. 3.12 illustrates the result of applying autocorrelation on the binary histogram for a text image and a non-text image. As depicted, the x-axis represents the indices of rows in the sample image, and the y-axis represents the correlation coefficients of the binary histogram. As shown in Fig. 3.12, the autocorrelation for text image contains significant peaks and troughs due to the periodic or quasi periodic pattern of text lines. Moreover, the autocorrelation for the non-text image is smoother than the autocorrelation for the text image without any significant peaks or troughs. In order to differentiate these two, we take out all the maxima and minima from $\mathbf{c}$ represented by two matrix $\mathbf{P}$ and $\mathbf{Q}$ (as shown in equation 5 and 6) of dimension $2 \times N'$ and $2 \times n'$ respectively, where N' and n' are the total number of all maxima and minima points. The whole procedure for detecting extrema is summarized in Algorithm 1.

$$\mathbf{P} = \begin{bmatrix} p^1(1) & p^1(2) & \dots & p^1(N') \\ p^2(1) & p^2(2) & \dots & p^2(N') \end{bmatrix}^T \quad (3.6)$$

$$\mathbf{Q} = \begin{bmatrix} q^1(1) & q^1(2) & \dots & q^1(n') \\ q^2(1) & q^2(2) & \dots & q^2(n') \end{bmatrix}^T \quad (3.7)$$

The half plot of the autocorrelation for the example images are shown in Fig. 3.13.

The typical characteristics of peaks in autocorrelation of text images are employed to select the final candidate text lines and the periodic pattern of text lines should satisfy the following constraints: it contains peaks due to the correlation of a text line with a text line and trough due to the correlation of the gap between two consecutive text lines with a text line. Fig. 3.14 illustrates two consecutive (i.e., i^{th} and $(i+1)^{th}$) maxima points; $P(i)$ and $P(i+1)$ with coordinates (x_i, y_i) and (x_{i+1}, y_{i+1})

Algorithm 1 Detection of maxima and minima of a vector

```

1: procedure EXTREMA(c)
Require: Vector c with length more than 1
2:   Initialize:  $y, h', x', y', i, j \leftarrow 0$  ▷ Initializing indices
3:    $H \leftarrow \text{length of } c$ 
4:   repeat
5:     if  $c(y+2) - c(y+1) \neq 0$  then
6:        $h' \leftarrow h' + 1.$ 
7:        $d^1(h') \leftarrow c(y+2) - c(y+1).$ 
8:        $d^2(h') \leftarrow y+2.$ 
9:        $y \leftarrow y+1.$ 
10:    end if
11:  until  $y = H - 2.$ 
12:  repeat ▷ Calculating maxima in c
13:    if  $(0 < d^1(x'+1))$  and  $(d^1(x'+2) < 0)$  then
14:       $i \leftarrow i+1.$ 
15:       $p^1(i) \leftarrow d^2(x').$ 
16:       $p^2(i) \leftarrow c(p^1(i)).$ 
17:       $\mathbf{P}(i) \leftarrow (p^1(i), p^2(i)).$ 
18:       $x' \leftarrow x'+1.$ 
19:    end if
20:  until  $x' = h' - 2.$ 
21:  repeat ▷ Calculating minima in c
22:    if  $(d^1(y'+1) < 0)$  and  $(0 < d^1(y'+2))$  then
23:       $j \leftarrow j+1.$ 
24:       $q^1(j) \leftarrow d^2(y').$ 
25:       $q^2(j) \leftarrow c(q^1(j)).$ 
26:       $\mathbf{Q}(j) \leftarrow (q^1(j), q^2(j)).$ 
27:       $y' \leftarrow y'+1.$ 
28:    end if
29:  until  $y' = h' - 2.$ 
30:  return P, Q ▷ All maxima and minima points of c
31: end procedure

```

Figure 3.12: Autocorrelation of binary histogram for (a) text image (b) non-text image

Figure 3.13: First half of the Autocorrelation for (a) text image (b) non-text images

respectively, where $x_i = p^1(i)$, $y_i = p^2(i)$ and $x_{i+1} = p^1(i+1)$, $y_{i+1} = p^2(i+1)$. The characteristics of peaks signify that $(x_{i+1} - x_i)$ is almost equal to the gap between i^{th} and $(i+1)^{th}$ text-lines of the text image. Following this computation, we define two step histograms of maxima and minima by two vectors; $\mathbf{r}$ and $\mathbf{s}$, respectively. $\mathbf{r} = (r(1), r(2), \dots, r(H))$ is represented by:

$$r(i) = \begin{cases} 0 & \text{for } 1 \leq i < p^1(1) \\ p^2(k) & \text{for } p^1(k) \leq i < p^1(k+1), \\ & k = 1, 2, \dots, N' - 1 \\ p^2(N') & \text{for } p^1(N') \leq i \leq H \end{cases} \quad (3.8)$$

Figure 3.14: Illustration of Step histogram of maxima

Figure 3.15: Step histogram of maxima and minima for text image

Similarly, $\mathbf{s} = (s(1), s(2), \dots, s(H))$ is represented by:

$$s(i) = \begin{cases} 0 & \text{for } 1 \leq i < q^1(1) \\ q^2(k) & \text{for } q^1(k) \leq i < q^1(k+1), \\ & k = 1, 2, \dots, n' - 1 \\ q^2(n') & \text{for } q^1(n') \leq i \leq H \end{cases} \quad (3.9)$$

Fig. 3.15 illustrates the two step histogram for the autocorrelation of a text image shown in Fig. 3.13(a) and Fig. 3.16 for the non-text image is shown in Fig. 3.13(b).

Fig. 3.15 and Fig. 3.16 show that the area under the graph of the two step histograms for the

Figure 3.16: Step histogram of maxima and minima for non-text image

non-text image are almost the same (as shown in Fig. 3.16) which is different from the text image (as shown in Fig. 3.15). Therefore, we compute the area difference under these graphs. Let $A(\mathbf{r})$ and $A(\mathbf{s})$ indicate the area under the step histogram graphs for $\mathbf{r}$ and $\mathbf{s}$ respectively and are computed by:

$$A(\mathbf{r}) = (p^2(N') \times (H - p^1(N'))) + \sum_{k=1}^{N'-1} p^2(k) \times (p^1(k+1) - p^1(k)) \quad (3.10)$$

$$A(\mathbf{s}) = (q^2(n') \times (H - q^1(n'))) + \sum_{k=1}^{n'-1} q^2(k) \times (q^1(k+1) - q^1(k)) \quad (3.11)$$

The final decision point for detecting text image is given by:

$$D_p = \frac{(A(\mathbf{r}) - A(\mathbf{s}))}{(W \times M_X(\mathbf{r}))} \times 100\% \quad (3.12)$$

where $M_X(\mathbf{r}) = r(H)$. D_p increases with the increase of the number of periodic text lines in the image. Therefore, in order to make a decision on whether the image is text image or not, a user-friendly threshold T can be offered and modified by users. Finally, an image is verified as a text image if the condition; $D_p \geq T$ is satisfied. In our implementation, we obtain a default value of T from the training set.

3.2 Text Proportion Estimation

From the decision point we can detect text images, but we cannot calculate the proportion of text in an image. We extend our work to calculate the proportion of text after detecting a text image. We used dilation, one of the basic operations in mathematical morphology to segment out the text portion from the binary image of the text image. The output of a text image after dilation is shown in Fig. 3.17.

In this result, we can observe that the white portion is the text. So we have calculated the percentage of white portions in this binary image of the text image as the percentage of text potential

Figure 3.17: (a) Original image (b) The result of dilation after Text image detection

Figure 3.18: (a) Original image (b) The result of dilation after Text image detection

regions. The percentage of text of the above example is 80.27%. As from this step the text percentage can be calculated, we have given the user an option for further filtering by putting a desired percentage threshold for text.

Moreover, sometimes the text of two images may be same, but the size of the paragraph may be different. Using this method we can easily figure out that both have same proportion of text. One of the examples is shown in Fig. 3.18. The percentages of text in the images are 33.52% and 34.15%, respectively.

3.3 Experiments and Results

3.3.1 Experimental setup

To the best of our knowledge, we are the first to address this problem. So there is no standard dataset for testing. In order to evaluate the proposed algorithm, we build a dataset which mainly contains

- (i) Text and non-text images downloaded from Internet(as shown in Fig. 3.19)

- (ii) Text images which are spam images with short messages(as shown in Fig. 3.20) and non-text or ham images taken from standard dataset[1,7] (as shown in Fig. 3.21)
- (iii) Text on images embedded by us.

Figure 3.19: Example of images from our dataset downloaded from internet

Figure 3.20: Example of text or spam images from our dataset taken from standard dataset[1,7]

Figure 3.21: Example of non-text or ham images from our dataset taken from standard dataset[1,7]

Our dataset contains 2217 images covering 773 text images and 1544 non-text images. Among 773 text images, we embed text paragraphs on 152 natural images taken from internet. The embedded

articles are taken from several internet publications (e.g. Wikipedia, Google) across a variety of disciplines. The background of these images are divided into three category as:

- (i) Simple background(as shown in Fig. 3.22)
- (ii) Little complex background(as shown in Fig. 3.23)
- (iii) Complex background(as shown in Fig. 3.24)

Figure 3.22: Example of text images created by us with simple background

Figure 3.23: Example of text images created by us with little complex background

Figure 3.24: Example of text images created by us with complex background

Table 3.1: Categorization of Training Dataset

Category	Source	Number of image	Total Image
Text Image	Created by us	52	52
Non-Text Image	Downloaded from Internet	102	102

Table 3.2: Categorization of Testing Dataset

Category	Source	Number of image	Total Image
Text Image	Public Dataset[1,7]	621	721
	Created by us	100	
Non-Text Image	IEEE IFS-TC 2013 Dataset	650	1442
	Public Dataset[1,7]	742	
	Downloaded from Internet	50	

These images contain many instances of text paragraph on images, varying font size and other characteristics which make the dataset more diverse. For this implementation, we assume that all text lines are horizontal and embed accordingly. From these 152 images, we take randomly 52 text images for training dataset and 100 for testing dataset. Moreover, our testing dataset also contains 621 standard text images taken from the standard dataset; such as Giorgio Fumeras dataset [1,7] and these are spam images with short messages.

It should be noted that, although the dataset concentrates on text images, it also includes a large variety of non-text images in order to provide a broader range of data for true negative experiments. We download 152 non-text images from internet and take randomly 102 images in training dataset and remaining 50 images in testing dataset. Our dataset also includes some non-text images from standard dataset; such as 650 non-text images from IEEE IFS-TC Image Forensics Challenge 2013 dataset and 742 ham images from Giorgio Fumeras dataset [1,7]. We put all 1392(650+742) non-text images of standard datasets in our testing dataset in order to verify our method.

Overall our training dataset contains 52 text images and 102 non-text images and our testing dataset contains 721 text images and 1442 non-text images. The summary of the training and testing datasets are shown in Table 3.1 and 3.2 respectively.

3.3.2 Threshold value for D_p

Ideally, the threshold for D_p can be set such that there will be no rejection of true text images. In order to increase the accuracy of the implementation, first of all, we calculate the D_p for all the images of the training dataset and after that we compute the accuracy for the training set by changing the threshold for D_p from 1% to 70%. Fig. 3.25 illustrates the plot of accuracy vs threshold for D_p . We take out the threshold value with maximum accuracy from the graph i.e. 10% as the default threshold for the D_p to detect the text images. The performance of the proposed method is evaluated by keeping 10% as the threshold value for D_p and test on our testing dataset.

3.3.3 Experiment results

We take 10% as the default threshold value for the decision point and test the training dataset of 52 text images and 102 non-text images. The results of the training dataset implemented in MATLAB are shown Table 3.3. The proposed method is implemented on our testing dataset and the corresponding results are listed in Table 3.4(implemented in MATLAB). Table 3.4 show that our proposed method is able to achieve a high accuracy with a low processing time. In order to conclude the whole performance of our proposed algorithm, we define this implementation of the testing dataset, as 721 positive instances and 1442 negative instances for text image detection and

Figure 3.25: Accuracy vs threshold for decision point graph of training set

Table 3.3: Result for training dataset implemented in MATLAB

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Image	52	2	99.6%	0.16
Non-Text Image	102	1	99%	0.10
Total	154	3	99.8%	0.12

Table 3.4: Result for testing dataset implemented in MATLAB

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Image	721	2	99.7%	0.15
Non-Text Image	1442	4	99.7%	0.09
Total	2163	6	99.7%	0.11

formulate a 2×2 contingency table or confusion matrix, as shown in Table 5.6. From the confusion matrix, the high sensitivity (99.7%) and high specificity (99.7%) conclude that proposed algorithm can correctly identify a text image as well as excluding a non-text image.

Table 3.5: Confusion matrix for the testing dataset

	Condition positive	Condition negative	
Test outcome positive	TP=719	FP=4	Precision= 99.4%
Test Outcome negative	FN=2	TN=1438	Negative prediction value=99.9%
	Sensitivity/Recall= 99.7%	Specificity= 99.7%	Accuracy =99.7%

3.4 Discussions

3.4.1 Text Detection by SWT

The Stroke Width Transform, [15] algorithm is one of the recent text detection techniques and has been widely used by many researchers to detect text mainly in scene text images. The Stroke Width Transform technique extracts text based on the assumption that words consist of roughly constant strokes. Therefore, compared with others, this technique can be used to detect text regardless of the font, size or language. Another advantage of the SWT technique is that the text can be detected even if the background is complex. After doing the experiment, the accuracy of the experiment is quite same with our proposed method, but with low speed. The processing time per image was 3.47 sec. But the aim of our work is to design a very efficient with high performance algorithm to block those small amounts of images which contains sensitive information.

3.4.2 Ads Policies for text images

Nowadays, many publishing companies are coming up with many new advertisement policies for text images to evaluate practical, useful ads and block illegal ads. As compared with these ad policies, like Facebook, Google etc. our method could work better. For example, the advertisement policy for text images by Facebook. According to this policy, images in ads can include text, but the quantity or proportion of text on the whole image should not be more than 20% text in the image. This policy is based on detecting the quantity of text and it does not detect the sensitivity or the purpose of the embedding of text on images. One example of such filter is shown in Fig. 3.26. This image is rejected by Facebook, as the quantity of text as calculated by their algorithm in this image is 32%. But there is nothing on the image to be blocked. Our proposed method can be more useful in this type of ad policies for the text images because it is message dependent rather than the quantity. Moreover the threshold to categorize text images can be changed by the company depending on the requirement of the ad policy. The example is non-text according to our method.

3.4.3 Calculating the number of lines

To further improve the functionality of the algorithm, our proposed method is extended to calculate the number of text lines in document images and text images with simple background after detecting a text image. Suppose, we have detected an image I as text image and let the binary histogram of I denote as $\mathbf{b}$. Ideally, the height of text lines within a paragraph (sequence of continuous 1s in $\mathbf{b}$) is almost the same. Moreover, the gap between the consecutive text lines (sequence of continuous 0s in $\mathbf{b}$) is almost the same as the height of the text line within a paragraph (sequence of continuous 1s in $\mathbf{b}$). In order to represent the periodicity of text line and the gap between consecutive text lines,

Figure 3.26: (a) Original image (b) Example of ad policy used by Facebook

we create another binary histogram denoted by $\mathbf{f}$ of dimension $1 \times 2H_M$ and computed by equation below, where H_M is the average height of text lines (sequence of continuous 1s in $\mathbf{b}$).

Figure 3.27: Binary histogram, $\mathbf{b}$ of a text image

Figure 3.28: One cycle, $\mathbf{f}$ of the binary histogram of Fig. 3.27

$\mathbf{f} = (f(1), f(2), \dots, f(2H_M))$ The whole procedure for detecting $\mathbf{f}$ is summarized in Algorithm 2. Fig. 3.27 illustrates binary histograms $\mathbf{b}$ and Fig. 3.28 illustrates one cycle of Fig. 3.27, $\mathbf{f}$ of a text image. In order to estimate the number of text lines or the number of cycles in $\mathbf{b}$, we convolute $\mathbf{f}$

Algorithm 2 Detection of one cycle of binary histogram

```

1: procedure CYCLE(b)
2:   H  $\leftarrow$  length of b
3:   Initialize: y, h', y', i, sum  $\leftarrow$  0 ▷ Initializing indices
4:   loop:
5:   if  $b(y+2) - b(y+1) \neq 0$  then
6:     h'  $\leftarrow$  h' + 1.
7:      $g^1(h') \leftarrow b(y) - b(y-1)$ .
8:      $g^2(h') \leftarrow y$ .
9:     y  $\leftarrow$  y + 1.
10:    goto
11:    close;
12:  end if
13:  y  $\leftarrow$  H - 2.
14:  summation:
15:  if  $(g^1(y'+1) = 1)$  and  $(g^1(y'+2) = -1)$  then
16:    i  $\leftarrow$  i + 1.
17:     $v(i) \leftarrow g^2(y'+1) - g^2(y')$ .
18:    y'  $\leftarrow$  y' + 1.
19:    sum  $\leftarrow$  sum + v(i).
20:    goto
21:    close;
22:  end if
23:  y'  $\leftarrow$  h' - 2.
24:  cycle:
25:  for h  $\leftarrow$  1 to  $\lfloor \frac{sum}{i} \rfloor$  do
26:    f(h)  $\leftarrow$  1.
27:  end for
28:  for h  $\leftarrow$   $\lfloor \frac{sum}{i} \rfloor + 1$  to  $2 \times \lfloor \frac{sum}{i} \rfloor$  do
29:    f(h)  $\leftarrow$  0.
30:  end for
31:  return f
32: end procedure

```

over $\mathbf{b}$. The convolution of $\mathbf{b}$ and $\mathbf{f}$ is denoted by $\mathbf{b}^* = (b^*(1), b^*(2), \dots, b^*(H))$ and computed as:

$$b^*(y) = \sum_{j=1}^{2H_M} (b(y-j) \times f(j)) \text{ and } y = 1, 2, \dots, H$$

Fig. 3.29 illustrates the convolution of the binary histograms shown in Fig 3.27.

Figure 3.29: Convolution of $\mathbf{b}$ and $\mathbf{f}$ of a text image

The characteristics of peaks in convolution of $\mathbf{b}$ and $\mathbf{f}$ for text images, are employed to select the number of text lines as the periodic pattern of text lines should satisfy the following constraints: it contains peaks due to the correlation of a text line or sequence of 1s in $\mathbf{b}$ with sequence of 1s in $\mathbf{f}$ and trough due to the correlation of the gap between two consecutive text lines or sequence of 0s in $\mathbf{b}$ with sequence of 1s in $\mathbf{f}$. In order to estimate the number of text lines, we compute the number of peaks by calculating the maxima in the convolution graph which is approximately equal to the number of text line in a text image with simple background. This step will be efficient to estimate the number of text lines in document images or handwritten images. For example, two document images and a text image with simple background are shown in Fig. 3.30. From this step we find correctly the number of text lines of these images as 25, 32 and 19 respectively.

3.4.4 Failed images

Horizontally repeated regular pattern

Our method will not work on the images with repeated regular graphical design pattern in horizontal direction; as shown in Fig. 3.31. As we have focused on accumulating number of dark pixels horizontality after the conversion to binary image, these kinds of images will give false positive results.

3.4.5 Parameter Analysis

Threshold for D_p ; T

We apply the proposed histogram-based method on randomly selected 100 text paragraph and 100 non-text paragraph images from the testing images, and plot the D_p value for each image in Fig. 3.32. It can be observed that the D_p values are discriminatively distributed for text paragraph and non-text paragraph images. More specifically, text paragraph images are with larger D_p values than non-text paragraph images, which fall above certain values to the threshold $T \in [8\%, 10\%]$ in Fig. 3.32.

To analyse the change on the value of D_p w.r.t. the font size and the number of text lines, we newly construct a text paragraph image dataset. In order to evaluate the influence of actual text font size on the value of D_p without any resizing of image as discussed in Section 3.1.1, each image is created with fixed height size as 500 pixels. We vary the number of text lines from 2 to 8, and vary the text font size from 30 to 100 pixels. Fig. 3.33 visually shows the D_p value changes with the increasing text font size. It can be observed that for a certain number of text lines, D_p value is

Figure 3.30: Three examples for text line detection

Figure 3.31: Two examples of horizontally repeated regular pattern images

positively related with the font size. Likewise, given a specific font size, D_p increases when adding more text lines. Our default threshold for D_p would fail if the image only contains a small portion of text, such as 2 to 5 text lines with font size of 30. With such small portion, the image is less likely to tell a persuasive story, which is concordant with our assumption.

Figure 3.32: D_p vs Images from testing dataset Figure 3.33: D_p vs Text font size

Chapter 4

Text Image Detection by Hough Transformation

4.1 Text Alignment Challenge

According to the proposed method mentioned in section 3, we have not considered the orientation of the text and we assume that text appears horizontally. However, if a person tilts the image such that there is no horizontal text pattern, then the histogram based detection method can not detect these types of text images. For example, the text image shown in Fig. 4.1(a) and its histogram shown in Fig. 4.1(c), the periodic pattern does not exist anymore. Besides, an arbitrary orientation of text paragraph makes the problem more complex and challenging. For example some text lines appear vertically, as depicted in Fig. 4.2.

4.2 Type of Text Image with Text Alignment

We have mentioned in section 2.1 that normally, there are three types of text in an image. They are; Caption text, Scene text and normal text. In caption text, the text can be embedded in different orientation (as shown in Fig. 4.3(a)) or after embedding the text, the text image can be rotated by some angle (as shown in Fig. 4.3(b)). In scene text, the text images can be captured in any direction with different text orientations, as shown in Fig. 4.4. The text in document images can also be in different orientation, which is commonly known as skew document images. Many research have been proposed to solve the orientation or degradation of skew document images. Two examples of skew document text images are shown in Fig. 4.5.

4.3 Text Image Detection for Arbitrary Text Alignment by Hough Transformation

In order to detect text images with text of arbitrary orientation from the non-text images, we implement the second method. This method consists of two parts: Hough line extraction and Filtration as shown in Fig. 4.6.

4.3.1 Hough line extraction

In general, the Hough transformation is mainly used to identify lines in an image or positions of arbitrary shapes, most commonly, circles and ellipses. It is a feature extraction technique used in computer vision and image processing [16].

Figure 4.1: Results of (a) original (b) Binary image (c) Histogram (d) Autocorrelation

Figure 4.2: Two examples of text lines appearing vertically

First of all, for the input image (see Fig. 4.7(a)) we apply the Canny edge detector on its binary image (see Fig. 4.7(b), obtained from Section 3.1.1) to get the edge map (see Fig. 4.7(c)). Then, we apply Hough transformation on the edge map and extract the Hough lines from the image.

4.3.2 Filtration

Hough lines contain both potential text lines and false positive text lines. In this part, we propose an algorithm to extract the potential text lines. It is expected that in text images the most significant Hough lines would be the text line. Therefore, in order to identify the potential text lines, we extract the first β major Hough lines by choosing the first β peaks having the maximum value in the Hough

Figure 4.3: (a)Rotated caption text while embedding(b)Rotated caption text after embedding

Figure 4.4: Scene text with different text alignment

Figure 4.5: Skew document text image

Space of the Hough transformation. In our implementation, we obtain a default value of β as 5. Fig. 4.7(d) shows the corresponding potential text lines.

In order to get the final decision, we extract the line segments of each potential text lines, and detect the corresponding angle of each line segments with the horizontal axis of the image. The set of the angles of the line segments is denoted as $\theta = (\theta_1, \theta_2, \dots, \theta_L)$. Here, L is the number of the line segments in the image and θ_j is the angle of j^{th} line segment. We take the Bag-of-words by

Figure 4.6: Flowchart of detection by Hough transformation

counting the angle frequency and denote the most frequent value as N_L . As N_L increases with the increase of the number of periodic text lines in the image, we verify a text image if the condition $N_L \geq T_L$ is satisfied. In our implementation, we set $T_L = 50$ because it achieves the maximum detection accuracy of the training dataset.

Figure 4.7: Example of Hough transformation-based method (a) Rotated text image (b) Corresponding binary image (c) Corresponding edge map; E (d) Hough lines of the first highest 5 peaks of the Hough Space

Table 4.1: Result of text image detection by Hough transformation for testing dataset implemented in MATLAB

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Image with text alignment	178	17	90.4%	0.09
Text Image without text alignment	721	81	88.8%	0.11
Non-Text Image	1442	20	98.6%	0.08
Total	2341	118	94.9%	0.09

4.4 Experiments and Results

4.4.1 Experimental setup

In order to evaluate the proposed method, we build a dataset for text images with different text orientations, which mainly contains text images rotated manually by an angle of 5 degree to 90 degree with an interval of 5 degree. This dataset contains 178 text images. Moreover, we include some scene text images in this dataset by taking pictures in different angles of a document and some skew document images also.

4.4.2 Experiment result

The proposed method is implemented on our testing dataset with the dataset of text alignment and the corresponding results are listed in Table 4.1 (implemented in MATLAB). Table 4.1 shows that though the accuracy is less than the proposed method mentioned in section 3, but this method is faster and can solve the text alignment problem with 94.9% accuracy. We can observe that the proposed method to solve the text alignment is able to achieve a low processing time, but with less accuracy as compared to the method mentioned in section 3.

4.5 Discussions

4.5.1 Failed images

In the proposed detection method by Hough transformation, we detect text images based on the linear characteristic of text edges. So the non-text images which have similar linear edges of objects may give us false positive results. Some examples of this type of false positive results are shown in Fig. 4.8.

4.5.2 Parameter Analysis

Threshold for N_L ; T_L

In this set of experiments, we discuss the parameter N_L , and analyse the changes of N_L with various images.

First of all, we apply the proposed Hough transformation-based method on randomly selected 100 text paragraph and 100 non-text paragraph images from the testing images, and plot the N_L value for each image in Fig. 4.9. It can be observed that the N_L values are discriminatively distributed for text paragraph and non-text paragraph images. More specifically, text paragraph images are with

Figure 4.8: False positive results for detection method by Hough transformation

larger N_L values than non-text paragraph images, which fall above certain values to the threshold $T_L \in [30, 60]$ in Fig. 4.9.

Figure 4.9: N_L vs Images from testing dataset

Chapter 5

Proposed Text Image Detection System

5.1 Proposed Detection System

Fig. 5.1 shows our proposed text image detection system. It consists of two steps: detection by Histogram mentioned in section 3 and detection by Hough transformation based method mentioned in section 4.

Figure 5.1: Flowchart of proposed detection system

Table 5.1: Result for the first step of the proposed detection system experimented on testing dataset

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Image with text of horizontal orientation	721	2	99.7%	0.15
Text Image with text of arbitrary orientation	178	170	4.5%	0.25
Non-Text Image	1442	4	99.7%	0.09
Total	2341	176	92.5%	0.11

Table 5.2: Result for the second step of the proposed detection system experimented on testing dataset

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Image with text of horizontal orientation	721	81	88.8%	0.09
Text Image with text of arbitrary orientation	178	17	90.4%	0.11
Non-Text Image	1442	20	98.6%	0.08
Total	2341	118	94.9%	0.09

5.2 Experimental Results

In order to evaluate the structure of the proposed detection system, we conduct two experiments on the testing dataset including the text images of text in arbitrary orientation and test individually for the first step (Section 3) and for the second step (Section 4) of the proposed algorithm. The corresponding results are listed in Table 5.1 and Table 5.2 respectively. Observations of these tables show that the first step or detection by Histogram based method detects well for text of horizontal orientation. Whereas the second step or detection by Hough Transformation based method outperforms the first step in terms of detecting text of arbitrary orientation; except the horizontal orientation with a faster speed. The possibility to embed text of horizontal orientation is a lot higher than any arbitrary orientation. On the contrary, as because the processing time per image by the second step is fast and detects text of any arbitrary orientation, it is important to pre-process or filter out the text images with text of horizontal orientation by the first step with higher accuracy to achieve a better performance. After the filtration of text images with text of horizontal orientation, we target the text images with text in arbitrary orientation by the second step and get the final text image detection system as shown in Fig. 5.1.

In order to show the effectiveness of the proposed detection system as shown in Fig. 5.1 over the proposed detection by Histogram and Hough transformation based methods, the proposed detection system is implemented on our testing dataset including the text images with text of arbitrary directions. The corresponding results are listed in Table 5.3. It shows that our proposed system is able to achieve a high accuracy with a low processing time. Moreover, because of the variety of text images in our dataset, Table 5.3 also shows that the proposed detection system is language and text font, size independent as long as there are text lines in the same direction. It can also detect three types of text embedded on images as mentioned in Section 2.

Table 5.3: Result for the proposed detection system experimented on testing dataset

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Image with text of horizontal orientation	721	2	99.7%	0.15
Text Image with text of arbitrary orientation	178	17	90.4%	0.35
Non-Text Image	1442	20	98.6%	0.12
Total	2341	39	98.3%	0.15

Table 5.4: Confusion matrix for the testing dataset

	Condition positive	Condition negative	
Test outcome positive	TP=880	FP=20	Precision=97.8%
Test Outcome negative	FN=19	TN=1422	Negative prediction value=98.6%
	Sensitivity/Recall=97.8%	Specificity=98.6%	Accuracy=98.3%

In order to conclude the whole performance of our proposed algorithm, we define this implementation of the testing dataset, as 899 positive instances and 1442 negative instances for text image detection and formulate a 2×2 contingency table or confusion matrix, as shown in Table 5.4. From the confusion matrix, the high sensitivity (97.8%) and high specificity (98.6%) conclude that proposed algorithm can correctly identify a text image as well as exclude a non-text image.

5.3 Comparison with Existing Techniques

To show the advantages of the proposed detection system, we compare it with four existing methods in [20, 21, 15, 26] on the testing dataset by recall, precision, F-measure, accuracy and time complexity per image. Because these four methods are dealing with text detection in natural scene images or document analysis problems, we cannot directly apply their methods for our problem. For fair comparison, we adopt their methods to locate the candidate text regions and generate the text candidate binary image. The text candidate binary images of Fig. 3.3(a) generated by these four existing methods are shown in Fig. 5.2 respectively. After generating the text candidate binary image from these four existing methods, we apply proposed text distribution representation step (Section 3.1.2) and text image detection step (Section 3.1.3). After that Hough line extraction and filtration steps(Section 4) are applied on the text images with text of arbitrary orientation and non-text images to make a final comparison. The flowchart of this comparison is shown in Fig. 5.3.

Pietikainen et al.[20] proposed a simple algorithm to detect text in document images with complex colored and textured backgrounds. It used a texture-based method by detecting the edge information of the image. The main advantage of their method is that the edge representation of their method can also be used for other image interpretation tasks.

Figure 5.2: Text candidate binary image generated by (a) Method in[20] (b) Method in[21] (c) Method in[15] (d) Method in[26]

Chen et al.[21] proposed a new text detection and recognition method. Their method mainly targets images and videos embedded with text in complex background. Their method mainly consists of two stages. In the first stage the text is detected and localized. In the second step, after detecting text lines, text is recognised by OCR technology.

Epshtein method[15] is already discussed in Section 2.

Yin et al.[26] proposed an accurate and robust method for detecting texts in natural scene images. The main advantage of their method is the speed. The algorithm is very fast and extracts Maximally Stable Extremal Regions (MSERs) as the main property of text is that it has minimum regular variations. By a novel self-training distance metric learning algorithm, text candidate regions extracted from MSERs are grouped together. Non text regions are eliminated by some text classifiers based on the posterior probabilities of text candidates.

Table 5.5 compares the text image detection performance of different methods respectively. It is clear from Table 5.5 that recall, precision, F-measure and the accuracy of the proposed algorithm is higher than those of the methods used from [20, 21, 15, 26], and the time complexity per image of the proposed algorithm is lower than the methods used from [20, 21, 15, 26], which indicates that the proposed algorithm detects text image faster than the other four methods.

Text detection from document images has been studied for many years. It is widely termed as Document Layout Analysis or simply Layout Analysis, which aims to extract text line, paragraphs, columns, title, etc. Therefore, those Text Line Detection can be used to solve the text paragraph image detection problem. To evaluate and compare the processing time of our proposed system with other's work, we choose the existing text line detection methods [67,68,69,70,71,72]. They deal with different document images, and the processing time is reported in their works. For comparison, we process the same input datasets of [67,68,69] and similar datasets of [70,71,72](because their datasets are not available) in our proposed detection system. Table 5.6 compares the performance of these six methods with the proposed detection system. It shows that the time complexity of the proposed method is less compared to the six methods, which achieves 424, 3.2, 12.3, 404, 732.3 and 10.3 times faster respectively.

Figure 5.3: The flowchart for the comparison system

5.4 Observations

Our proposed detection system is able to achieve 98.3% accuracy of text image detection, with a time cost 0.15 second per image under MATLAB. As our dataset is diverse and the accuracy of the proposed detection system is high as shown in Table 5.4 and 5.5, we can conclude that the proposed detection system is also language, font type and size independent as long as there are space gaps between text lines. To show language independency of the proposed algorithm, some positive examples of text images with different languages from our dataset; such as English, Chinese, Hindi are shown in Fig 5.4. Some positive example of document images with different challenges; such as documents affected by illumination, focus or different types of document degradation of the proposed algorithm are shown in Fig. 5.5.

Table 5.5: Text image detection performance comparison of different methods for testing dataset, where Proposed TE is proposed Text Extraction, TDR is Text Distribution Representation, TID is Text Image Detection Step, HLE is Hough Line Extraction and F is the Filtration of the proposed algorithm implemented in MATLAB

Methods	Precision	Recall	F-Measure	Accuracy	Time per image (Sec)
Pietikainen Method[20]-TDR- TID-HLE-F	93.3%	57.7%	71.3%	82.2%	1.22
Chen Method[21]- TDR-TID -HLE-F	82.7%	79.9%	81.3%	85.9%	0.59
Epshtein Method[15]-TDR- TID-HLE-F	97.6%	93.9%	95.7%	96.8%	3.53
Yin Method[26]- TDR-TID -HLE-F	91.7%	92.9%	92.3%	94.1%	0.53
Proposed TE- TDR-TID -HLE-F	97.8%	97.8%	97.8%	98.3%	0.15

Figure 5.4: Some positive examples of text images of different languages from our dataset

Table 5.6: Text paragraph image detection performance comparison of different methods, where we use ‘ - ’ if not reported and Comparison is evaluated by the Processing time of Proposed Method to the compared method

Methods	Machine Specification	Code	Time per image (Sec)	Time per image (Sec) for proposed method	Comparison (times)
Bukhari[67]	2.5GHz 40GB RAM Linux	-	204.00	0.48	424
Seong[68]	2.66GHz 4GB RAM	C++	1.17	0.28	3.2
Hyung[69]	2.66GHz 4GB RAM	C++	3.73	0.28	12.3
Kumar[70]	-	MATLAB	158.00	0.39	404
Acharyya [71]	-	MATLAB	286.00	0.39	732.3
Chaudhuri[72]	Pentium 3 700MHz	C++	1.35	0.12	10.3

Figure 5.5: Some positive examples of text images with document degradation from our dataset

Chapter 6

Future Work

In our future work, we can focus on addressing the following problems:

6.1 Text Images of Different Color Contrast

We have mentioned in the section 3.1 that we target those images which have consistent or at least similar color contrast difference between text and the background. Because of this, if multiple text images are combined horizontally with different color contrast, then our proposed method will not work on these kinds of images. An example of this type of image is shown in Fig. 6.1. In order to detect text images with different color contrasts, we can develop a text segmentation algorithm to extract text features separately and make a final decision based on all segmented parts.

Figure 6.1: One example of combined text images

Chapter 7

Conclusions

In this thesis, we propose a text detection system to differentiate images embedded with sufficient text (termed as text images) and other images. Since the number of text images is much less when compared with other images through the internet, this technique could highly reduce labor work in blocking defamation or illegal commentaries which are conveyed by images.

Our technique is composed of two steps. The first step is a histogram based method, where we filter out the text images with text in horizontal direction. The second step is a detection method based on Hough transformation, where we target the text images with text of arbitrary direction.

We demonstrate the effectiveness of proposed technique on a newly built dataset. This dataset is composed of text and non-text images. One fourth of the text images are created by embedding text in natural images, and the remaining text images are taken from some standard datasets or downloaded from the internet. Non-text images are downloaded from the internet. Our technique is able to achieve 98.3% accuracy of text image detection, with a time cost 0.15 second per image under MATLAB.

Reference

- [1] G. Fumera, I. Pillai, F. Roli. “Spam filtering based on the analysis of text information embedded into images”, *Journal of Machine Learning Research (special issue on Machine Learning in Computer Security)*, vol. 7, pp. 2699-2720, 12/2006.
- [2] B. Biggio , G. Fumera , I. Pillai , F. Roli. “A survey and experimental evaluation of image spam filtering techniques”, *Pattern Recognition Letters*. Volume 32, Issue 10, 15 July 2011, Pages 1436-1446, ISSN 0167-8655
- [3] H. B. Aradhye , G. K. Myers, J. A. Herson. “ Image analysis for efficient categorization of image-based spam e-mail”. *Int. Conf. on Document Analysis and Recognition*, pp. 914918/2005.
- [4] M. Dredze, R. Gevartyahu, A. Elias-Bachrach. “ Learning fast classifiers for image spam” ,*4th Conf. on Email and Anti-Spam (CEAS)/2007*.
- [5] C.-T. Wu, K.-T. Cheng, Q. Zhu, Y.-L. Wu. “Using visual features for anti-spam filtering”, *IEEE Int. Conf. on Image Processing*, Vol. III.pp. 501504/2005.
- [6] Q. Liu, Z. Qin, H. Cheng, M. Wan. “ Efficient modeling of spam images”, *Int. Symp. on Intelligent Information Technology and Security Informatics*. IEEE Computer Society, pp. 663666/2010.
- [7] B. Biggio, G. Fumera, I. Pillai, F. Roli. “Image Spam Filtering Using Visual Information”, *14th Int. Conf. on Image Analysis and Processing (ICIAP 2007)*, Modena, Italy, IEEE Computer Society, pp. 105–110, 10/09/2007.
- [8] F. Roli, B. Biggio, G. Fumera, I. Pillai, R. Satta. “Image Spam Filtering by Detection of Adversarial Obfuscated Text”, *Workshop on Neural Information Processing Systems (NIPS)*, Whistler, British Columbia, Canada, 08/12/2007.
- [9] B. Biggio, G. Fumera, I. Pillai, F. Roli. “Improving Image Spam Filtering Using Image Text Features”, *Fifth Conference on Email and Anti-Spam (CEAS 2008)*, Mountain View, CA, USA, 21/08/2008.
- [10] M. Dredze, R. Gevartyahu, A. Elias-Bachrach. “Learning Fast Classifiers for Image Spam”, *4th Conference on Email and Anti-Spam (CEAS)*, California, 2007.
- [11] “ICDAR 2003 robust reading competitions”, *International Conference on Document Analysis and Recognition*, 2003, pp. 682-68.
- [12] “ICDAR 2005 text locating competition results”, *International Conference on Document Analysis and Recognition*, 2005. pp. 80-84.
- [13] “ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images”, *International Conference on Document Analysis and Recognition*, 2011, pp. 1491-1496.
- [14] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. “Robust text detection in natural images with edge-enhanced maximally stable extremal regions”, *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 26092612.
- [15] B. Epshtein, E. Ofek, and Y. Wexler. “Detecting text in natural scenes with stroke width transform”, *Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2963-2970.
- [16] [http : //en.wikipedia.org/wiki/Color_wheel](http://en.wikipedia.org/wiki/Color_wheel)

- [17] <http://imageprocessingblog.com/histogramadjustmentsinmatlabpartieequalization/>
- [18] N. Efford. "Digital Image Processing: A Practical Introduction Using Java™."; Pearson Education, 2000; Chapter 11 "Morphological image processing".
- [19] S. Lu, B. Su, C. L. Tan. "Document image binarization using background estimation and stroke edges", *International Journal on Document Analysis and Recognition(IJDAR)* 2010, 13:303314
- [20] M. Pietikainen and O. Okun. "Edge-Base Method for Text Detection from Complex Document Images", *International Conference on Document Analysis and Recognition*, 2001.
- [21] D. Chen, J.-M. Odobez, H. Bourlard. "Text detection and recognition in images and video frames", *Pattern Recognition*, 2004.
- [22] Z. Li, G. Liu, X. Qian, D. Guo, H. Jiang. "Effective and efficient video text extraction using key text points", *Image Processing IET*, 2011, 5(8):591-598.
- [23] M. Zhao, S. Li, J. Kwok. "Text detection in images using sparse representation with discriminative dictionaries", *Image and Vision Computing*, 2010, 28: 1590-1599.
- [24] H. I. Koo and D. H. Kim. "Scene text detection via connected component clustering and nontext filtering", *IEEE transactions on image processing*, 2013, 22(6):2296-2305.
- [25] V.Y. Mariano, R. Kasturi. "Locating uniform-color text in video frames", in *Proc. Intl Conf. Pattern Recognition*, 2000, 539-542.
- [26] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao. "Robust Text Detection in Natural Scene Images", *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2014, 970-983.
- [27] M. Sezgin and B. Sankur. "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic Imaging*, 2004, 13 (1): 146-165. doi:10.1117/1.1631315.
- [28] J. Ohya, A. Shio, and S. Akamatsu. "Recognizing Characters in Scene Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16 (2) (1994) 214-224.
- [29] J. C.M. Lee, and A. Kankanhalli. "Automatic Extraction of Characters in Complex Images", *International Journal of Pattern Recognition Artificial Intelligence*, 9 (1) (1995) 67-82.
- [30] M.A. Smith and T. Kanade. "Video Skimming for Quick Browsing Based on Audio and Image Characterization", *Technical Report CMU-CS-95-186*, Carnegie Mellon University, July 1995.
- [31] Y. Zhong, K. Karu and A. K. Jain. "Locating Text In Complex Color Images", *Pattern Recognition*, 28 (10) (1995) 1523-1535.
- [32] B.L. Yeo, and B. Liu. "Visual Content Highlighting via Automatic Extraction of Embedded Captions on MPEG Compressed Video", *Proc. of SPIE*, 1996, pp.142-149.
- [33] J. C. Shim, C. Dorai, and R. Bolle. "Automatic Text Extraction from Video for Content-based Annotation and Retrieval", in *Proc. of International Conference on Pattern Recognition*, Vol. 1, 1998, pp. 618-620.
- [34] A. K. Jain, and B. Yu. "Automatic Text Location in Images and Video Frames", *Pattern Recognition*, 31 (12) (1998) 2055-2076.
- [35] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith. "Video OCR for Digital News Archive", in *Proc. of IEEE Workshop on Content based Access of Image and Video Databases*, 1998, pp. 52-60.
- [36] B. T. Chun, Y. Bae, and T.-Y. Kim. "Automatic Text Extraction in Digital Videos using FFT and Neural Network", in *Proc. of IEEE International Fuzzy Systems Conference*, 1999, Vol. 2, pp. 1112-1115.
- [37] S. Antani, U. Gargi, D. Crandall, T. Gandhi, and R. Kasturi. "Extraction of Text in Video", *Technical Report of Department of Computer Science and Engineering, Penn. State University*, CSE-99-016, August 30, 1999.

- [38] S. Messelodi and C.M. Modena. "Automatic Identification and Skew Estimation of Text Lines in Real Scene Images", *Pattern Recognition*, 32 (1992) 791-810.
- [39] V. Wu, R. Manmatha, and E. M. Riseman. "TextFinder: An Automatic System to Detect and Recognize Text in Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21 (11) (1999) 1224-1229.
- [40] Y. M. Y. Hasan and L.J. Karam. "Morphological Text Extraction from Images", *IEEE Transactions on Image Processing*, 9 (11) (2000) 1978-1983.
- [41] H. Li, D. Doerman, and O. Kia. "Automatic Text Detection and Tracking in Digital Video", *IEEE Transactions on Image Processing*, 9 (1) January (2000) 147-156.
- [42] Y. K. Lim, S. H. Choi, and S.W. Lee. "Text Extraction in MPEG Compressed Video for Content-based Indexing", in *Proc. of International Conference on Pattern Recognition*, 2000 pp. 409-412.
- [43] Y. Zhong, H. Zhang, and A. K. Jain. "Automatic Caption Localization in Compressed Video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, (4) (2000) 385-392.
- [44] K. Jung. "Neural network-based Text Location in Color Images", *Pattern Recognition Letters*, 22 (14) December (2001) 1503-1515.
- [45] D. Chen, K. Shearer, and H. Bourlard. "Text Enhancement with Asymmetric Filter for Video OCR", in *Proc. of International Conference on Image Analysis and Processing*, 2001, pp. 192-197.
- [46] C. Strouthpoulos, N. Papamarkos, and A.E. Atsalakis. "Text Extraction in Complex Color Document", *Pattern Recognition*, 35 (8) (2002) 1743-1758.
- [47] K. Jung, K. I. Kim, A. K. Jain. "Text information extraction in images and video: a survey", *Pattern Recognition* 37(5): 977-997 (2004)
- [48] S. H. Park, K. I. Kim, K. Jung, and H. J. Kim. "Locating Car License Plates using Neural Networks", *IEE Electronics Letters*, 35 (17) (1999) 1475-1477.
- [49] B.- K. Sin, S.-K. Kim, and B.-J. Cho. "Locating Characters in Scene Images using Frequency Features", in *Proc. of International Conference on Pattern Recognition*, 2002, Vol. 3, pp. 489-492.
- [50] W. Mao, F.-I. Chung, K.K.M. Lam, and W.-c. Sun. "Hybrid Chinese/English Text Detection in Images and Video Frames", in *Proc. of International Conference on Pattern Recognition*, 2002, Vol. 3, pp. 1015-1018.
- [51] R. Lienhart and F. Stuber. "Automatic Text Recognition In Digital Videos", in *Proc. of SPIE*, 1996, pp. 180-188.
- [52] R. Lienhart and W. Effelsberg. "Automatic Text Segmentation and Text Recognition for Video Indexing", *Technical Report TR-98-009, Praktische Informatik IV, University of Mannheim*, 1998.
- [53] H. Zhang, , K. Zhao , Y-Z. Song and J. Guo. "Text extraction from natural scene image: A survey", *Neurocomputing*, 2013, Vol. 122, pp. 310-323.
- [54] W. Ou, J. Zhu, C. Liu, " Text location in natural scene", *Journal of Chinese Information Processing* (2004).
- [55] K.C. Kim, H.R. Byun, Y.J. Song, Y.W. Choi, S.Y. Chi, K.K. Kim, Y.K. Chung, " Scene text extraction in natural scene images using hierarchical feature combining and verification", in: 2004 Proceedings of the *17th International Conference on Pattern Recognition*, ICPR 2004, IEEE, vol. 2, 2004, pp. 679682.
- [56] J.S. Kim, S.C. Park, S.H. Kim, " Text locating from natural scene images using image intensities", in: 2005 Proceedings of the *Eighth International Conference on Document Analysis and Recognition*, IEEE, 2005, pp. 655659.
- [57] D.T. Hu, X.H. He, "Effective Algorithm of Text Extraction in Natural Scenes", *Ph.D. Thesis*, 2008.

- [58] T.D. Bui, W. Pan, C.Y. Suen, "Text detection from natural scene images using topographic maps and sparse representations", in: 2009 *IEEE International Conference on Image Processing*, 2009.
- [59] Y.F. Pan, C.L. Liu, X. Hou, "Fast scene text localization by learning-based filtering and verification", in: 2010 17th *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2010, pp. 2269-2272.
- [60] S.H. Lee, M.S. Cho, K. Jung, J.H. Kim, "Scene text extraction with edge constraint and text collinearity", in: 2010 *Proceedings of 20th ICPR*, pp. 3983-3986.
- [61] R. Minetto, N. Thome, M. Cord, J. Fabrizio, B. Marcotegui, "A Multiresolution System for Text Detection in Complex Visual Scenes", in: *Proceedings of 2010 IEEE 17th International Conference on Image Processing (ICIP)*, vol. 1, 2010, pp. 3861-3864.
- [62] L. Neumann, J. Matas, "A method for text localization and recognition in realworld images", *Computer Vision/ACCV*, 2010, 2011, pp. 770-783.
- [63] Y. Pan, X. Hou, C. Liu, "A hybrid approach to detect and localize texts in natural scene images", *IEEE Transactions on Image Processing*, 20 (2011) 800-813.
- [64] H.P. Le, N.D. Toan, S.C. Park, G.S. Lee, "Text localization in natural scene images by mean-shift clustering and parallel edge feature", *5th International Conference on Ubiquitous Information Management and Communication*, ACM, 2011, p. 116.
- [65] Y.F. Pan, Y. Zhu, J. Sun, S. Naoi, "Improving scene text detection by scaleadaptive segmentation and weighted CRF verification", *2011 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2011, pp. 759-763.
- [66] G. Zhou, Y. Liu, Q. Meng, Y. Zhang, "Detecting multilingual text in natural scene", *2011 1st International Symposium on Access Spaces (ISAS)*, IEEE, 2011, pp. 116 - 120
- [67] S.S.Bukhari, F.Shafait, T.M. Breuel. "Text-Line Extraction Using a Convolution of Isotropic Gaussian Filter with a Set of Line Filters". *International Conference on Document Analysis and Recognition(ICDAR)*, 2011 pp. 579 - 583.
- [68] J.H. Seong, J. Bora, I.C. Nam. "Fast text line extraction in document images". *IEEE International Conference on Image Processing(ICIP)*, 2012, pp. 797 - 800
- [69] I.K. Hyung, I.C. Nam. "State Estimation in a Document Image and Its Application in Text Block Identification and Text Line Extraction". *European Conference on Computer Vision(ECCV)*, 2010, pp 421 - 434.
- [70] S.Kumar, R. Gupta, N. Khanna, S. Chaudhury. "Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model". *IEEE Transactions on Image Processing(TIP)*, 2007, Volume:16 , Issue: 8 , pp. 2117 - 2128.
- [71] M. Acharyya and M. K. Kundu, "Multiscale segmentation of document images using M -band wavelets". *International Conference Computer Analysis of Images and Patterns, (CAIP)*, 2001, pp. 510 - 517.
- [72] A.R. Chaudhuri, A.K. Mandal, B.B. Chaudhuri, "Page layout analyser for multilingual Indian documents". *Language Engineering Conference*, 2002, pp. 24 - 32.
- [73] J. Liang, D. Doermann and H. Li, "Camera-based analysis of text and documents: a survey". *International Journal of Document Analysis and Recognition (IJDA)*, 2005, Volume 7, Issue 2-3, pp. 84-104.