Histogram-Based Fast Text Paragraph Image Detection

Devadeep Shyam, Yan Wang, and Alex C. Kot
Rapid-Rich Object Search (ROSE) Lab
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
dshyam@ntu.edu.sg, wang0696@e.ntu.edu.sg, eackot@ntu.edu.sg

***Abstract*—Rumormongers always use long paragraphs to spread slanderous stories so that they can convince readers. Those illegal or sensitive rumors uploaded into the internet can be written on images to by-pass text filters. These images can be detected by existing filters such as OCR, but the detection is very time consuming. To prohibit the dissemination of those commentaries, detecting whether an image contains a sufficient amount of words provides convenience to the government or internet service providers. Because of this, we focus on developing a fast pre-processor algorithm for detecting images embedded with sufficient text, such that the text filters (e.g. OCR) only need to focus on those suspected images. In this paper, we propose a histogram-based fast detection method to determine whether an image contains paragraphs of text or not. Binary histograms are extracted from the converted binary images. Then, due to the periodic pattern of the histograms, a step curve is designed to apply on the autocorrelation of those histograms. The area under the curve is further utilized to differentiate images with paragraphs and those without. To imitate the scenario, we construct a new dataset covering more than 2000 images of with and without paragraphs. The results show the effectiveness of the proposed detection system, which achieves 99.5% in accuracy and 15 millisecond per image in speed implemented in C++.**

I. Introduction

INTERNET spreads all kinds of messages swiftly every day, including defamation or illegal commentaries. Such type of information in text can be easily blocked by text filters. However, a person can take a screenshot of the sensitive article and save it as an image, or overlay the whole article on an image. To explain or convince the whole illegal information, the information is usually put in long write-ups (e.g., stories) on the image. These images can be detected by existing filters such as Optical Character Recognition (OCR). OCR, is a technology that converts different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data. But it is very time consuming. Therefore, we develop a fast algorithm to detect images embedded with long paragraphs which are potentially more harmful. Only detected images with text paragraph will be further analysed by OCR techniques, which saves time. For simplicity, the image embedded with sufficient text (paragraphs, articles) is termed as text paragraph image in the following discussions. Some examples of text paragraph image, taken from internet are shown in Fig. 1.

Fig. 1. Some examples of text paragraph image

In this paper, we first generate a text candidate binary image from the original image. Then we represent the distribution of text by employing autocorrelation. In the final step, we design a step graph to differentiate the periodic pattern of text paragraph images from the noisy pattern of non-text paragraph images.

The major contribution of this paper is to develop a fast algorithm to detect text paragraph images. As text paragraph images occupy a very small portion of all the images which are potentially more harmful, designing an efficient and fast pre-filtering for massive upload of images is necessary. We develop a fast algorithm to detect images embedded with sufficient text, which can be further verified by text filters. This reduces the processing time by eliminating images which do not have sufficient text and thus do not have to be processed by text filters.

The rest of this paper is arranged as follows: Section II discusses the applications and techniques of existing text detection works. Section III presents the proposed text paragraph image detection method in detail. Section IV describes and discusses the experimental results. Section V describes shortcomings and failed results of the proposed method, followed by the conclusion and future work in the last section.

This work was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by a grant from the Singapore National Research Foundation and administered by the Interactive & Digital Media Programme Office at the Media Development Authority.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.76

Fig. 2. Types of text on images (a) Images with caption text (b) Images with scene text (c) Document images

II. Existing Text Detection Techniques

There are mainly three types of text that can be embedded in images: *caption text*, which is artificially overlaid on the image (see Fig. 2(a)); *scene text*, which exists naturally in the image (see Fig. 2(b)); and *text in document images*, which exists in document images with homogeneous background, but affected by illumination, focus or different types of document degradation (see Fig. 2(c)). Existing techniques proposed for detecting text on images are always focusing on one single type of text. On the contrary, our focus is to detect text paragraph images regardless of the types of text. For example,

1) *Caption text:* Mariano et al.[11] propose a text detection method for locating horizontal, uniform-colored caption text in video frames. Chen et al.[12] propose a new method for detecting and also recognizing caption text in complex images and video frames. Li et al.[13] propose a video text extraction scheme using key text points.
2) *Scene text:* Epshtein et al.[14] aim to extract text characters in natural scenes with stable stroke widths and considers components with low stroke width variations as text components. Yin et al.[15] detect text in natural scene images by extracting a group of neighboring pixels with similar properties such as color,certain size, shape and spatial alignment constraints. Koo et al.[16] propose to detect scene text by extracting connected components in images by using the maximally stable extremal region algorithm.
3) *Text in document images:* Bukhari et al.[1] extract text line from document images based on the use of a convolution of isotropic Gaussian filter with line filters. Instead of binarization or multi-oriented Gaussian blurring of an image as in the conventional methods, Seong et al.[6] use integral image and design filters that are proper to detect text regions on the integral image of document images. Koo et al.[7] propose a new approach to the estimation of document states such as interline spacing and text line orientation, which facilitates a number of tasks in document image processing. Kumar et al.[8] propose a novel scheme for the extraction of textual areas of a document image using globally matched wavelet filters. Li et al.[9] develop an algorithm for segmenting document images into four classes: background, photograph, text, and graph. They classify these by distribution patterns of wavelet coefficients in high frequency bands. As existing OCR modules developed for various Indian scripts can handle text only single-column documents Chaudhuri et al.[10] propose an algorithm to analyse a page layout of the Indian document images.

Moreover, the existing text detection techniques on images have been proposed for specific applications including page segmentation and analysis, address block location and recognition, license plate location and recognition, and content-based image/video indexing. These methods deal with the diversity of text font, size, orientation and language or images affected by variations in scene and camera parameters such as illumination, focus, motion etc. 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. However, they are time consuming and most of these methods are not considering the properties of text paragraphs. Hence in this paper, we propose a pre-processing detection system to automatically block images embedded with sufficient text in a fast manner, such that given an input image, we propose a histogram-based method to filter out images embedded with text paragraph from non-text paragraph images.

III. Proposed Methodology

Fig. 3 shows our proposed text paragraph image detection system. It consists of three parts: text extraction, text distribution representation and text paragraph image detection. The text extraction is applied to get a binary image from the input image to differentiate text from the background. A histogram extraction technique is then 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 autocorrelation-based step curve method.

A. Text Extraction

For an input image with height more than 500 pixels, we resize its height to 500 pixels with the image aspect ratio unchanged, as shown in Fig. 4. 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 color contrast adjustment[4]. This is used to enhance the contrast

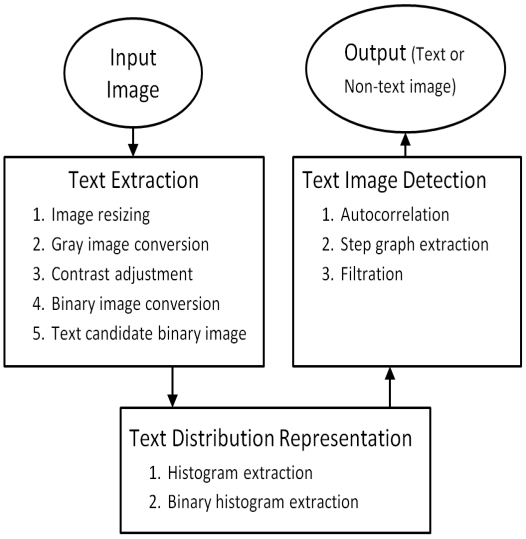

Fig. 3. Algorithm proposed for text paragraph image detection

between potential foreground text and background image. It facilitates the following process of getting the potential text regions from the text paragraph image.

After color contrast enhancement, we convert the gray image into a binary one to distinguish potential foreground text pixels from the background. We employ Otsu's method [2] for the binary conversion and set 0 to pixels of the foreground and 1 to those belonging to the background (as shown in Fig. 4 (b)).

The foreground of the binary image contains both text potential regions and false positive text potential regions appearing big dark regions as shown in Fig. 4(b). Therefore, we remove the big dark regions. After removing the big dark regions, we get the partial regions or part of text regions due to 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 or part of text regions are extracted in consistent manner. Whereas, the small non-text regions are extracted in a distorted or inconsistent manner. For simplicity, we term this map 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 $x = 1, ..., W$ and $y = 1, ..., H$. We define the continuous horizontal connected component of I_b as a set of horizontal connected pixels denoted as S_i^H (see Fig. 5). Here, i indicates the i^{th} horizontal connected component in the image. S_i^H can be defined as

$$S_i^H = \{(u_{i1}, v_i), (u_{i2}, v_i), \ldots, (u_{iq}, v_i)\} \quad (1)$$

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 ($q = 4$ in Fig. 5).

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

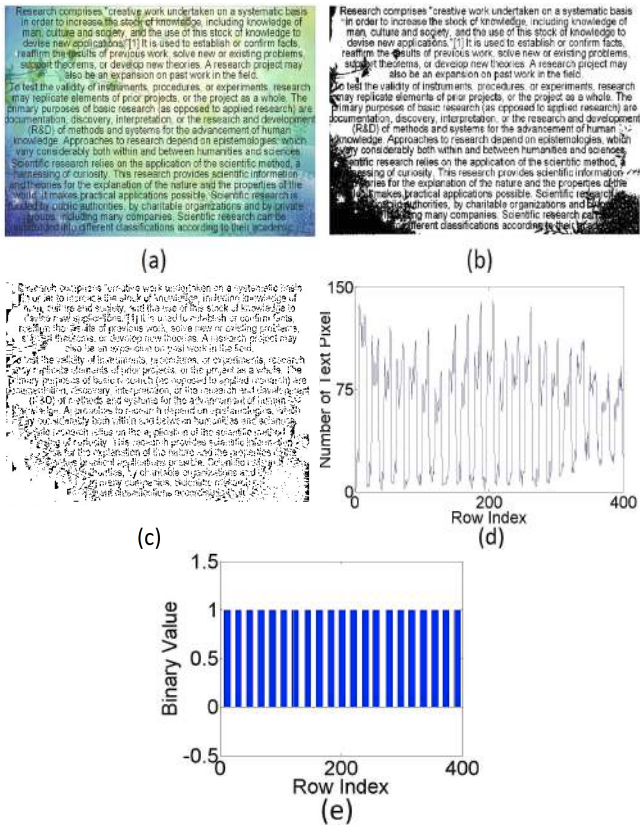

Fig. 4. (a) Original text paragraph image (b) Binary image (c) Text candidate binary image (d) Histogram (e) Binary histogram

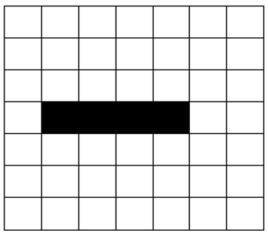

Fig. 5. Illustration of horizontal connected component (each block represents a pixel)

$$I_T(x, y) = \begin{cases} 1 & \text{if } \exists i: \ (x, y) \in S_i^H; x = u_{ij}; \\ & (u_{iq} - u_{ij}) \geq T_1 \\ I_b(x, y) & \text{otherwise} \end{cases} \quad (2)$$

An example is shown in Fig. 6 to illustrate the process. In our implementation, we set T_1=5 to extract small partial text regions of all text font size. Fig. 4(c) show the text candidate images of the example image.

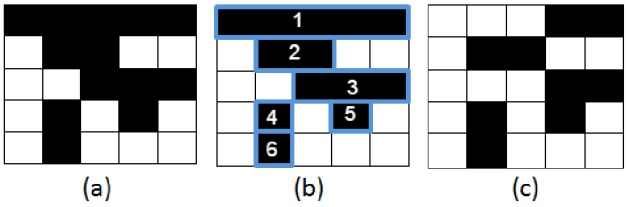

Fig. 6. (a) Binary image (b) Horizontal connected components (i.e., S_i^H) of binary image (c) Obtained text candidate binary image, where $T_1 = 2$, $i = 1, ..., 6$.

B. 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 almost constant through an article. Therefore, we propose a histogram-based method to represent the row-wise text distribution accumulated row by row from the text candidate binary image. The histogram of the entire image is denoted as; $\boldsymbol{n} = (n(1), n(2), \ldots, n(H))$ and it's $y^{th}(y = 1, 2, ..., H)$ value is computed by:

$$n(y) = W - \sum_{x=1}^{W} I(x, y) \tag{3}$$

The histogram of a text paragraph image is illustrated in Fig. 4(d), where the x-axis represents the indices of rows in the sample image, and the y-axis computes the number of potential text pixels (with the value of 0 in I) on each specific row. Then 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; $\boldsymbol{b} = (b(1), b(2), \ldots, b(H))$ and it's $y^{th}(y = 1, 2, ..., H)$ value is computed by:

$$b(y) = \begin{cases} 1 & \text{if } n(y) \geq B \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where, an adaptive threshold B is decided by

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

We set the mean value (i.e. B) of the histogram (i.e. $\boldsymbol{n}$) as the threshold for the binarization, which aims to binarize the histogram based on the major lengths of potential text lines. As shown in Fig. 4(e), we plot the values of the binary histogram of the example image 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.

C. Text Paragraph Image Detection

In order to extract the features of a text paragraph, we detect the periodic or a quasi periodic pattern of text lines within a paragraph. As binary histogram of text paragraph image is quasi periodic, we compute the autocorrelation on binary histogram to differentiate the images with text paragraph from non-text. The autocorrelation of binary histogram is represented by a vector; $\boldsymbol{c} = (c(1), c(2), \ldots, c(H))$ of dimension $1 \times H$ and computed by:

$$c(y) = \sum_{j=1}^{H-y} (b(j) \times b(j + y)), y = 1, 2, ..., H \tag{5}$$

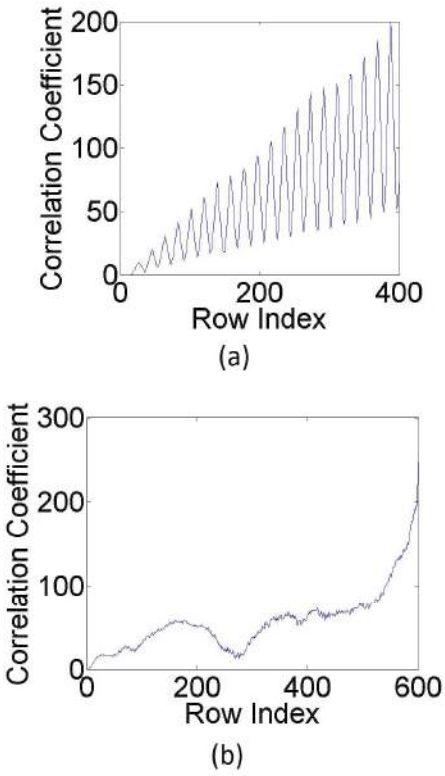

Fig. 7. Autocorrelation of binary histogram for (a) text paragraph image and (b) non-text paragraph image

Fig. 7 illustrates the result of applying autocorrelation on the binary histogram for a text paragraph image and a non-text paragraph image. As shown in Fig. 7, the autocorrelation for text paragraph image contains significant peaks and troughs due to the periodic or quasi periodic pattern of text lines. Moreover, the autocorrelation for the non-text paragraph image is smoother than the autocorrelation for the text paragraph image without any significant peaks or troughs. In order to differentiate these two, we search the local maxima and minima from the first index of $\boldsymbol{c}$ to the last and use two matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ to represent them separately:

$$\boldsymbol{P} = (\boldsymbol{p}(1) \quad \boldsymbol{p}(2) \quad \ldots \quad \boldsymbol{p}(N))^T \tag{6}$$

$$\boldsymbol{Q} = (\boldsymbol{q}(1) \quad \boldsymbol{q}(2) \quad \ldots \quad \boldsymbol{q}(M))^T$$

Here, N and M are the total number of all maxima and minima points, respectively. We define the i^{th} $(i = 1, ..., N)$ maxima by:

$$\boldsymbol{p}(i) = (p_1(i) \quad p_2(i)) \tag{7}$$

where, $p_1(i) = k$ and $p_2(i) = c(k)$; if $c(k-1) < c(k)$ and $c(k+1) \leq c(k); k = 2, ..., H-1$.

Similarly, we define the j^{th} $(j = 1, ..., M)$ minima by:

$$\boldsymbol{q}(j) = (q_1(j) \quad q_2(j)) \tag{8}$$

where, $q_1(j) = k$ and $q_2(j) = c(k)$; if $c(k) < c(k-1)$ and $c(k) \leq c(k+1); k = 2, ..., H-1$.

$p_1(i)$ and $p_2(i)$ represent the row index and correlation coefficient of the i^{th} maxima and $q_1(j)$ and $q_2(j)$ represent the row index and correlation coefficient of the j^{th} minima. Noted that $p_1(1) < p_1(2) < ... < p_1(N)$ and $q_1(1) < q_1(2) < ... < q_1(M)$.

The typical characteristics of the peaks in the autocorrelation of the text paragraph images are employed to select the final candidate text lines. 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 troughs due to the correlation of the space between two consecutive text lines with a text line. Fig. 8 illustrates two consecutive (i.e.,i^{th} and $(i+1)^{th}$) maxima points; $\boldsymbol{p}(i)$ and $\boldsymbol{p}(i+1)$. Following this computation, we propose two step histograms of maxima and minima represented by two vectors: $\boldsymbol{r}$ and $\boldsymbol{s}$, respectively. $\boldsymbol{r} = (r(1), r(2), \ldots, r(H))$ is computed by:

$$r(i) = \begin{cases} 0 & \text{for } 1 \leq i < p_1(1) \\ p_2(k) & \text{for } p_1(k) \leq i < p_1(k+1), \\ & k = 1, 2, \ldots, N-1 \\ p_2(N) & \text{for } p_1(N) \leq i \leq H \end{cases} \tag{9}$$

Similarly, $\boldsymbol{s} = (s(1), s(2), \ldots, s(H))$ is computed by:

$$s(i) = \begin{cases} 0 & \text{for } 1 \leq i < q_1(1) \\ q_2(k) & \text{for } q_1(k) \leq i < q_1(k+1), \\ & k = 1, 2, \ldots, M-1 \\ q_2(M) & \text{for } q_1(M) \leq i \leq H \end{cases} \tag{10}$$

Fig. 9(a) illustrates the two step histogram for the auto-correlation of a text paragraph image shown in Fig. 7(a) and Fig. 9(b) for the non-text paragraph image as shown in Fig. 7(b). Fig. 9(a) and (b) show that the area under the curve of the two step histograms for the non-text paragraph image are almost the same (as shown in Fig. 9(b)) which is different from the text paragraph image (as shown in Fig. 9(a)). Therefore, we compute the area difference under these curves. Let $A(\boldsymbol{r})$ and $A(\boldsymbol{s})$ indicate the area under the step histogram curves for $\boldsymbol{r}$ and $\boldsymbol{s}$, respectively which are computed by:

$$A(\boldsymbol{r}) = (p_2(N) \times (H - p_1(N))) + \sum_{k=1}^{N-1} p_2(k) \times (p_1(k+1) - p_1(k)) \tag{11}$$

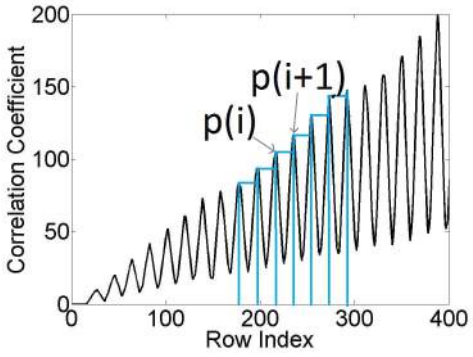

Fig. 8. Illustration of Step histogram of maxima

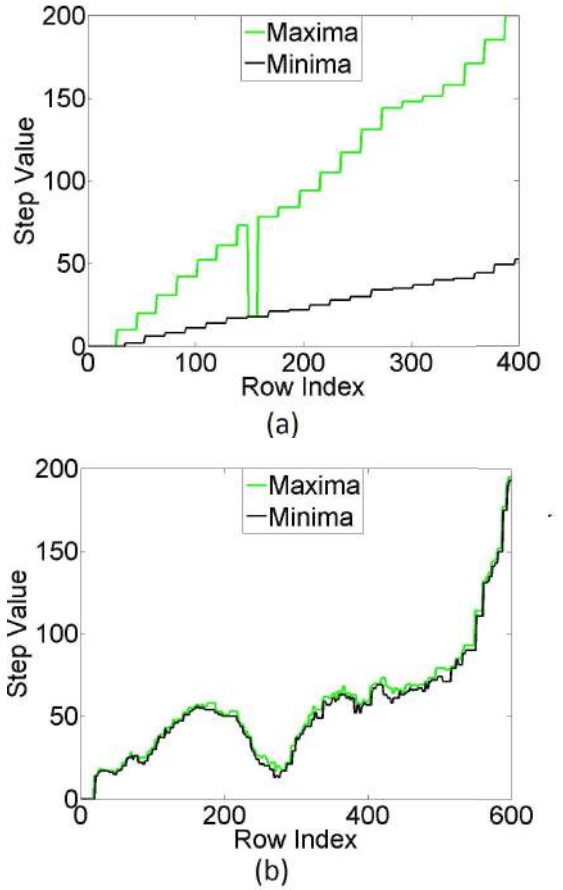

Fig. 9. Step histogram of maxima and minima for (a) text paragraph image (b) non-text paragraph image

$$A(\boldsymbol{s}) = (q_2(M) \times (H - q_1(M))) + \sum_{k=1}^{M-1} q_2(k) \times (q_1(k+1) - q_1(k)) \tag{12}$$

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

$$D_p = \frac{(A(\boldsymbol{r}) - A(\boldsymbol{s}))}{(W \times M_X(\boldsymbol{r}))} \times 100\% \quad (13)$$

where $M_X(\boldsymbol{r}) = r(H)$. D_p increases with the increase of the number of periodic text lines in the image. It is proportional to the text proportion of the image. Therefore, in order to make a decision on whether the image is text paragraph image or not, a threshold T can be set by the user based on their requirement for text proportion. Finally, an image is verified as a text paragraph image if the condition; $D_p \geq T$ is satisfied. We set a default value of $T = 8\%$ as it achieves the maximum detection accuracy of the training dataset.

IV. Experimental Results

A. Dataset Construction

In order to evaluate the proposed algorithm, we build a dataset which contains 2217 images covering 773 text paragraph images and 1544 non-text paragraph images. These images are collected from different sources, which are synthetic, downloaded from the internet, or taken from some public datasets.

Among 773 text paragraph images, we manually embed text paragraphs on 152 natural images retrieved from the internet and the rest of the 621 text paragraph images are taken from a public dataset, Giorgio Fumera [3]. For each embedded text paragraph image, the proportion of text on the image is more than 50%. The embedded articles are taken from several internet publications (e.g. Wikipedia, Google) across a variety of disciplines. These synthetic text paragraph images contain a mixture of simple and complex background, including many instances of text paragraph on images, varying font size and other characteristics. For this implementation, we assume that all text lines are horizontal and embedded accordingly. Images taken from the standard dataset are spam images with short messages embedded on these images in text format. 52 text paragraph images are randomly selected for training and the rest are for testing. For non-text paragraph images, 152 images are downloaded from Google, Flicker as well as other search engines, and the remaining 1392 images are retrieved from IEEE IFS-TC Image Forensics Challenge 2013 dataset [5] and Giorgio Fumera's dataset [3]. Among these images, 100 images are used for training and 1442 images are for testing.

We conduct the experiments on Visual Studio 2012 implemented in C++, in a desktop of Intel(R) Core(TM) i5-4570 3.20GHz CPU and 8.0GB RAM. For the comparison of our method with other existing methods, we also conduct our experiment in MATLAB R2012b on the same desktop.

B. Experiment Results

We take $T = 8\%$ as the threshold value for D_p and apply our method on the testing dataset. Table I gives the detection accuracies and computational complexities for text paragraph images, non-text paragraph images. It can be observed that the overall accuracy is approaching 100% and the processing time per image is around 15 millisecond only based on C++ implementation. 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 paragraph image detection and formulate a 2 × 2 contingency table or confusion matrix, as shown in Table II. From the confusion matrix, the high sensitivity (99.7%) and high specificity (99.4%) conclude that proposed algorithm can correctly identify a text paragraph image as well as exclude a non-text paragraph image.

TABLE I. Result for the testing dataset

Category	Number of images	Wrong output	Accuracy	Time per image (Sec)
Text Paragraph Image	721	2	99.7%	0.024
Non-Text Paragraph Image	1442	9	99.4%	0.011
Total	2163	11	**99.5%**	**0.015**

TABLE II. Confusion matrix for the testing dataset, where TP, FP, FN and TN short for True Positive, False Positive, False Negative and True Negative, respectively

	Condition positive	Condition negative	
Test outcome positive	TP=719	FP=9	Precision= 98.8%
Test Outcome negative	FN=2	TN=1433	Negative prediction value=99.9%
	Sensitivity/Recall= 99.7%	Specificity= 99.4%	Accuracy= 99.5%

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 [1,6,7,8,9,10]. They deal with different document images, and the processing time is reported in their works. For comparison, we process the same input datasets of [1,6,7] and similar datasets of [8,9,10](because their datasets are not available) in our proposed detection system. Table III 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 550, 4.9, 17.7, 437.7, 793.4 and 11.3 times faster in MATLAB implementation respectively. Moreover, in C++ implementation the time complexity of the proposed method is 2913, 30.6, 99.8, 1974, 3574 and 32.8 times faster respectively.

C. Parameter Analysis

1) Threshold for D_p ; T: In this set of experiments, we discuss the parameter D_p, and analyse the changes of D_p with various font sizes and text lines.

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

TABLE III. TEXT PARAGRAPH IMAGE DETECTION PERFORMANCE COMPARISON OF DIFFERENT METHODS, WHERE WE USE ' NA ' IF NOT REPORTED AND COMPARISON IS EVALUATED BY THE PROCESSING TIME OF PROPOSED METHOD TO THE COMPARED METHOD

Methods	Machine Specification	Code	Time/image (Sec)	Proposed Method in **MATLAB** Time/image(Sec)	Proposed Method in **C++** Time/image(Sec)
Bukhari[1]	2.5GHz, 40GB RAM, Linux	NA	204.00	0.37	.07
Seong[6]	2.66GHz, 4GB RAM	C++	1.17	0.20	**.04**
Hyung[7]	2.66GHz, 4GB RAM	C++	3.73	0.20	**.04**
Kumar[8]	NA	MATLAB	158.00	**0.36**	.08
Acharyya [9]	NA	MATLAB	286.00	**0.36**	.08
Chaudhuri[10]	Pentium 3, 700MHz	C++	1.35	0.11	**.04**

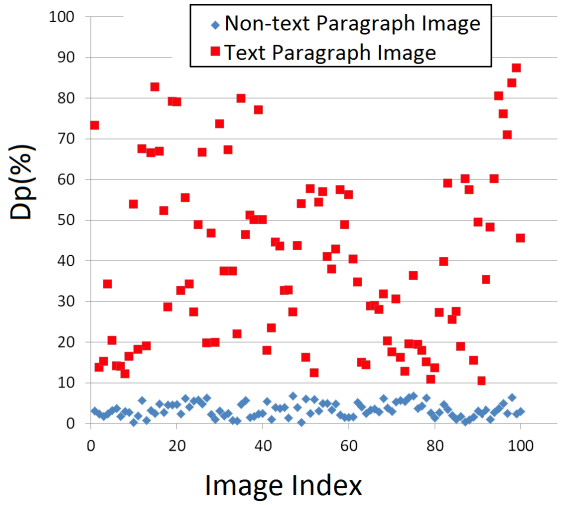

Fig. 10. D_p vs Images in testing dataset

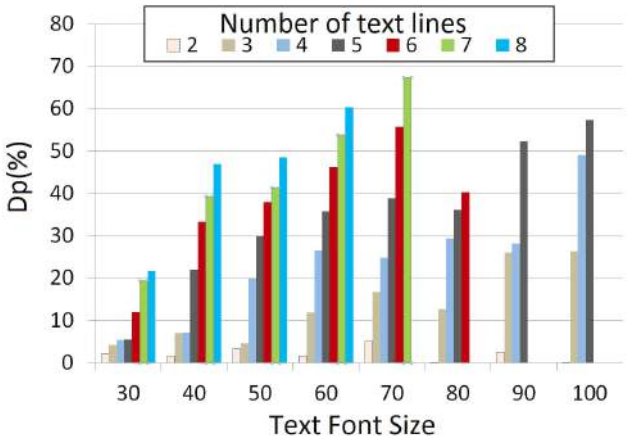

Fig. 11. D_p vs Text font size

each image in Fig. 10. 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. 10.

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 III-A, 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. 11 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 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.

V. FAILED RESULTS

In this section, we analyse the shortcomings and failed results of the proposed method.

A. *Horizontally-repeated regular pattern*

Our method does not work on the images with horizontally-repeated regular graphical design pattern; as shown in Fig. 12. As we focus on accumulating number of dark pixels horizontality after the conversion to binary image, these kinds of images give false positive results.

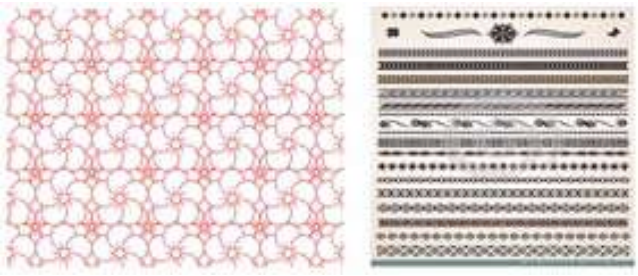

Fig. 12. Two examples of horizontally repeated regular pattern images

VI. Conclusion and Future Work

In this paper, we propose a text detection method to differentiate images embedded with sufficient text (term as text paragraph images) and other images. Since the quantity of text paragraph images is much smaller compared with non-text paragraph images through the internet, this technique can highly reduce government's or internet service provider's labor in blocking defamation or illegal commentaries which are conveyed by images. After text paragraph image detection, the government agencies or internet service providers only need to focus on a small portion of suspected images. We propose to first generate a text candidate binary image from the original image. Then we represent the distribution of text by employing autocorrelation. In the final step, we design a step graph to differentiate the periodic pattern of text paragraph images and the smooth while noisy pattern of non-text paragraph images. We conduct the experiment on a newly built dataset and we achieve a 99.5% accuracy in text paragraph image detection with a time cost of 15 millisecond per image under C++ implementation. Our future work will be extended to the detection of non-horizontally oriented text paragraph.

References

[1] 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.

[2] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic Imaging*, 2004, vol. 13 (1), pp. 146-165.

[3] 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)*, 2006, vol. 7, pp. 2699-2720.

[4] http://imageprocessingblog.com/histogram-adjustments-in-matlab-part-ii-equalization/

[5] http://ifc.recod.ic.unicamp.br/fc.website/index.py

[6] 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

[7] 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.

[8] 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.

[9] 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.

[10] A.R. Chaudhuri, A.K. Mandal, B.B. Chaudhuri,"Page layout analyser for multilingual Indian documents". *Language Engineering Conference*, 2002, pp. 24 - 32.

[11] V. Y. Mariano and R. Kasturi. "Locating uniform-color text in video frames" . Proc. *International Conference Pattern Recognition*, 2000, 539-542.

[12] D. Chen, J.-M. Odobez, H. Bourlard."Text detection and recognition in images and video frames", Pattern Recognition, 2004.

[13] 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.

[14] B. Epshtein, E.Ofek, and Y.Wexler, "Detecing text in natural scenes with stroke width transform", *Computer Vision and Pattern Recognition, (CVPR)*, 2010, pp.2963-2970.

[15] X. C. Yin, X. Yin, K. Huang, and H. W.i Haoi. " Robust Text Detection in Natural Scene Images", *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2014, 970-983.

[16] H. 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.