Journal of Signal and Information Processing, 2014, 5, 208-228
Published Online November 2014 in SciRes. http://www.scirp.org/journal/jsip
http://dx.doi.org/10.4236/jsip.2014.54022

Simple and Efficient Text Localization for Compressed Image in Mobile Phone

Jung Hyoun Kim[1], Adrián Canedo-Rodríguez[2], Jung Hee Kim[1], John Kelly[1]

[1]College of Engineering, North Carolina A&T State University, Greensboro, USA
[2]Research Centre on Information Technology, University of Santiago de Compostela, Santiago de Compostela, Spain
Email: kim@ncat.edu, adrian.canedo@usc.es

Received 26 September 2014; revised 20 October 2014; accepted 17 November 2014

Abstract

Extraction of the text data present in images involves Text detection, Text localization, Text tracking, Text extraction, Text Enhancement and Text Recognition. Due to its inherent complexity, traditional text localization algorithms in natural scenes, especially in multi-context scenes, are not implementable under low computational resources architectures such as mobile phones. In this paper, we proposed a simple method to automatically localize signboard texts within JPEG mobile phone camera images. Taking into account the information provided by the Discrete Cosine Transform (DCT) used by the JPEG compression format, we delimitate the borders of the most important text region. This system is simple, reliable, affordable, easily implementable, and quick even working under architectures with low computational resources.

Keywords

Text Localization, Discrete Cosine Transform, Text Extraction, Text Detection

1. Introduction

Text Information Extraction (TIE) is a well differentiated branch on the Pattern Recognition area. Since the early 1960s, when Optical Character Recognition (OCR) was born as one of the first clear applications of Pattern Recognition [1], the detection, localization, extraction, and processing of textual information within images has evolved overcoming technical difficulties, while increasing its application areas. Traditionally, TIE was focused on the analysis of scanned documents, which provided a pseudo-ideal scenario: high resolution, minimal character shape distortion, even and adequate illumination, clear, simple and known backgrounds, minimal blur, and so on. However, this constrained scenario was insufficient for the development of useful applications for general

How to cite this paper: Kim, J.H., Canedo-Rodríguez, A., Kim, J.H. and Kelly, J. (2014) Simple and Efficient Text Localization for Compressed Image in Mobile Phone. *Journal of Signal and Information Processing*, **5**, 208-228.
http://dx.doi.org/10.4236/jsip.2014.54022

users' needs, not limited to document analysis under controlled conditions.

A TIE system's input is a still image or a sequence of images. Its goal is to process this image in order to extract the contained text information, though a defined set of following steps:

1) Text Detection: Determination of the presence of text.
2) Text Localization: Determination of the location of the texts.
3) Text Tracking: In sequences of images, determination of the coherence of the text between frames, to reduce the processing time by not applying all the steps to every frame, and to maintain the integrity of position across adjacent frames.
4) Text Extraction: Binarization of the image by separating the text components (foreground) from the background.
5) Text Enhancement: Increasing of the quality of the binary image, mainly by increasing its resolution and reducing the noise.
6) Text Recognition (OCR): Transformation of the binary text image into plain text using an Optical Character Recognition Engine.

The explosion of Handheld Imaging Devices (HID) over the past years brought the opportunity to develop useful and marketable TIE applications. On contrast with scanners, these devices are small, light, portable, cheap, easily integral with networks, and can capture any image or video in any scenario. As a result, they build a huge market niche, either standalone or embedded on other devices, such as mobile phones. Sign recognition and translation for travelers, automatic license plate recognition for law enforcement, driver assistance systems, assistance for visually impaired persons, or autonomous vehicle navigation represent just a small set of the wide range of possibilities of this new area.

Nevertheless, TIE scenario using HID is far from being as ideal as with scanners. The resolution is lower than in scanned documents, the illumination is very difficult to control, the background and layout of the texts are arbitrary and often complex and the text may present various distortions. The processing techniques needed for TIE systems to overcome these difficulties are usually extremely computationally expensive, so its implementation on HID, which have very low computational resources, is often unfeasible. On the other hand, those methods which are efficient in computational terms are not robust enough to cope with "real world" degradations, such as uneven illuminations, or lighting reflections. As a result, in order to develop useful and marketable TIE applications over these devices, it is necessary to achieve a compromise between simplicity and accuracy.

In this paper, we propose a simple, fast, and accurate algorithm to perform the first step that every TIE system involves: text localization. In order to ensure the usefulness and market viability of our methodology, we have designed it to work in portable devices such as mobile phones, as part of an automatic translation system. Since the computational time is a major issue on this devices, and taking into account that most of the images are stored using a JPEG compressed format [2], we have designed our system to localize the text by using the DCT (Discrete Cosine Transform) coefficients [3]-[5], avoiding the computationally expensive process of uncompressing the image. We define the concept of Horizontal and Vertical Text Energy depending on the frequency information of the image. Using this information, we first roughly delimitate the several text candidate areas, from which we choose the most probable target of the translation. Finally, the boundaries of the chosen candidate are precisely adjusted, and the text is well localized.

In Section 2, we describe an introduction about the general TIE system and with the main challenges for text localization. In Section 3, the details of our methodology will be described including the math modeling. In Section 4, we verify our algorithm, whose results are described and analyzed in Section 5. Finally, we summarize the contents of the paper, and discuss about the contribution of our proposal, and the future research lines in Section 6.

2. Background for Text Localization

In order for a text localization algorithm to be useful, it has to accurately locate the texts in an image, as fast as possible. Text Localization techniques are divided in Region-Based, Texture-Based and Hybrid [6] [7]. Other than the references mentioned, the reader may refer to the papers in the references [8]-[11]. The Region Based Methods are divided into three types: Connected Component, Edge-Based and Hybrid Methods. The Connected Component Method generates connected components and filter out the non-text components for grouping [12]-[17]. However, this method is expensive because of geometrical analysis; especially connect edges, or similar regions. The Edge Based Method mainly focused on high contrast between the text and the back ground

[18]-[20]. This method is affected drastically by noises, uneven lightings, and complex backgrounds. The Texture Based Method can be applied not only to uncompressed images but also to compressed images directly [21]-[29]. From an uncompressed image, variance, color, energy, frequency information, or edge density are usually extracted in order to differentiate foreground and background regions. Some methods used to characterize the image textures, like edge filters, Wavelet Transform, Fast Fourier Transform, variance calculation, or Gabor Filters, may be very expensive in computational terms.

Most digital images are transmitted, processed and stored in compressed form. Particularly, most of images taken by mobile phones are compressed because of limited storages. Specifically, the most widely compressed images are in JPEG format. In this sense, we extract the texture features of the different regions directly from the compressed image format without the need of uncompressing the images which saves a lot of computational time. Discrete Cosine Transform coefficients from JPEG images usually characterize the different textures, and many classification algorithms can be used to separate foreground and background regions [30]-[33]. However, it is not easy to determine which DCT coefficients are better to differentiate text and background textures. In general, this is the fastest method because it does not compress the stored information as well as does not apply expensive spatial image processing operations.

3. Proposed Algorithm

Our algorithm makes two assumptions: First, the images are stored in the JPEG format. We will further consider just the luminance channel. Second, the user centers the image in the text that he wants to process. Our purpose is to delineate the most significant text area, represented by the lower and upper horizontal and vertical boundaries by, respectively.

The JPEG compression standard establishes that each $M \times N$ image $f(x, y)$ as shown in **Figure 1** where $x \in \{0, \cdots, M-1\}$ and $y \in \{0, \cdots, N-1\}$. The image $f(x, y)$ is divided into 8 × 8 blocks $f_{ij}(x, y)$ where $i \in \{0, \cdots, (M/8)-1\}$ $j \in \{0, \cdots, (N/8)-1\}$ and $x, y \in \{0, \cdots, 7\}$ as shown in **Figure 2**. Each block will be encoded by the Discrete Cosine Transform as Equation (1):

$$F_{ij}(k_1, k_2) = \frac{C(k_1)C(k_2)}{4}\left\{\sum_{x=0}^{7}\sum_{y=0}^{7} f_{ij}(x, y)\cos\left(\frac{(2x+1)k_1\pi}{16}\right)\cos\left(\frac{(2y+1)k_Y\pi}{16}\right)\right\}, \tag{1}$$

where $k_1, k_2 \in \{0, \cdots, 7\}$

$$C(k_1) = C(k_2) = \frac{1}{\sqrt{2}}$$

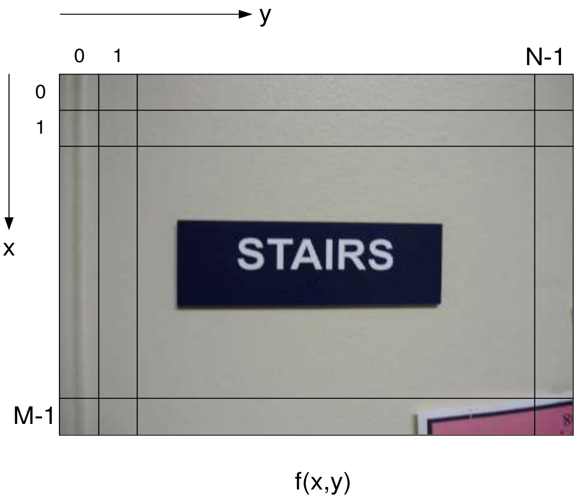


Figure 1. Representation of an image *f*(*x*, *y*).

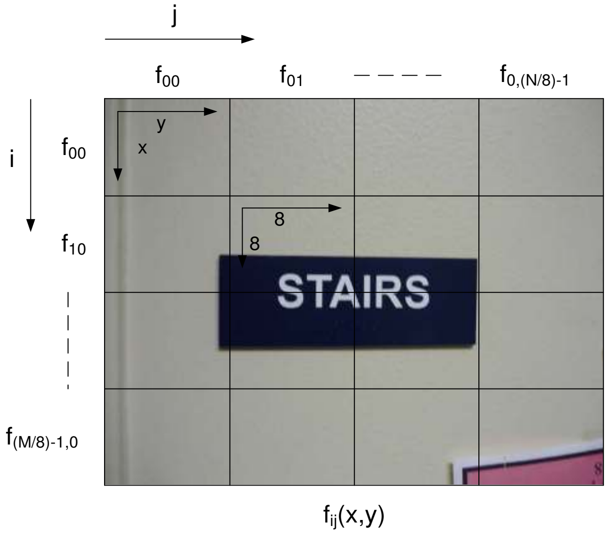


Figure 2. $f_{ij}(x,y)$—Representation of an image $f(x,y)$ divided on 8 × 8 blocks.

for $k_1 = k_2 = 0$, and $C(k_1) = C(k_2) = 1$ otherwise.

The horizontal text area localization is based on the following statement: a DCT block which contains characters will specially present strong high frequency components in both horizontal and vertical directions due to the variations between foreground and background. We define two measures for the degree of text presence for each block: Block Horizontal Text Energy $E_h(i,j)$ in Equation (2), and Block Vertical Text Energy $E_v(i,j)$ in Equation 3 as shown in **Figure 3**.

$$E_h(i,j) = \sum_{k_2=1}^{4} F_{ij}(0,k_2) \tag{2}$$

$$E_v(i,j) = \sum_{k_1=1}^{4} F_{ij}(k_1,0) \tag{3}$$

The Block Horizontal Text Energy of each block (i,j) is the summation of the blue coefficients $\{(0,1),(0,2),(0,3),(0,4)\}$, and the Block Vertical Text Energy is the summation of the green ones $\{(1,0),(2,0),(3,0),(4,0)\}$.

We will then extend the previous definitions considering the accumulated Text Energy in a row by Equations (4) and (5) or a column by Equations (5) and (6).

$$R_h(i) = \sum_{j=c_1}^{c_2} E_h(i,j) \tag{4}$$

$$R_v(i) = \sum_{j=c_1}^{c_2} E_v(i,j) \tag{5}$$

$$C_h(j) = \sum_{i=r_1}^{r_2} E_h(i,j) \tag{6}$$

$$C_v(j) = \sum_{i=r_1}^{r_2} E_v(i,j) \tag{7}$$

where c_1, c_2 represent the column borders, and r_1, r_2 the row borders of a region of interest as shown in **Figure 4**.

Calculation of $R_h(i)$ on an area is defined by the row borders r_1 and r_2. Calculation of $C_h(j)$ on an

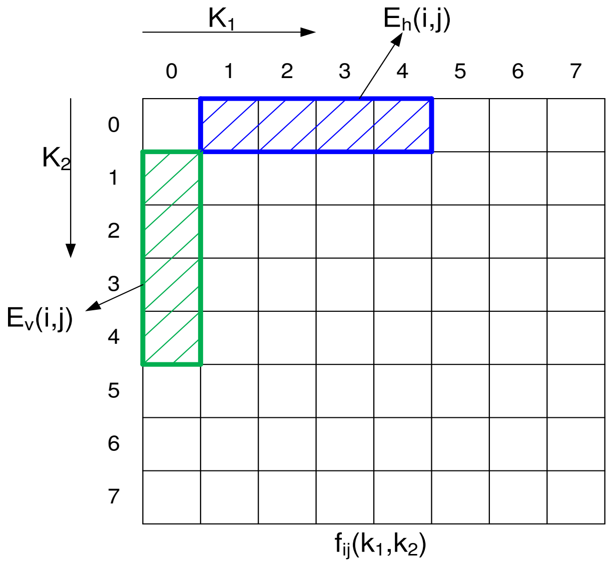


Figure 3. A 8 × 8 block on DCT domain $F_{ij}(x,y)$.

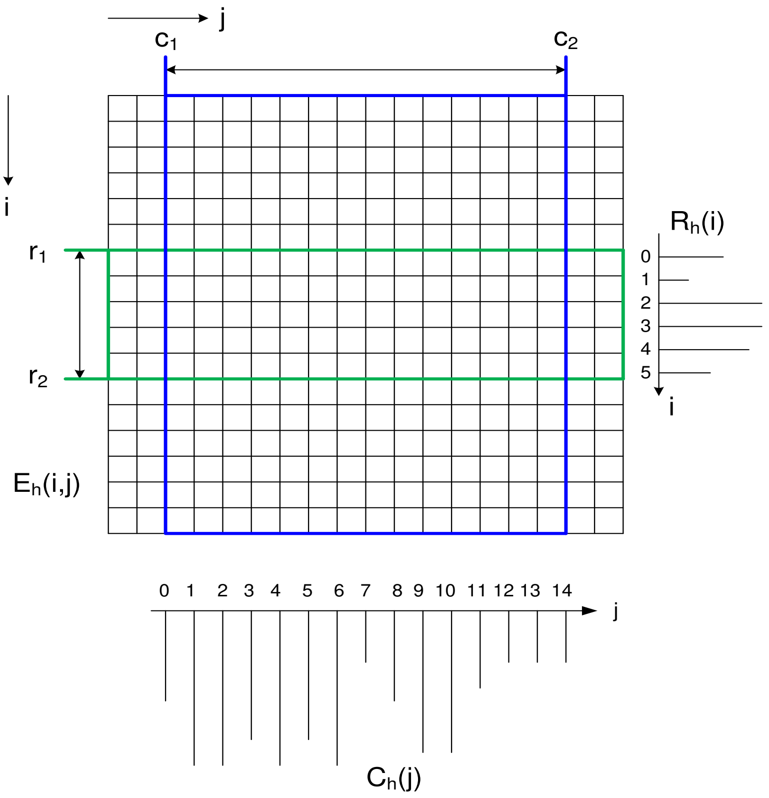


Figure 4. Text energy calculation for the row and the column.

area is defined by the column borders c_1 and c_2. The procedure would be the same to calculate R_v and C_v, replacing E_h with E_v.

Now, considering the text areas are horizontally aligned, the rows of blocks which contain text lines will present higher $R_h(i)$ values $(c_1 = 0, c_2 = (N/8) - 1)$ than the non-text rows $R_h(i)$. At this point, a set of adjacent rows with value over $\text{AVG}(R_h(i))$ in **Figure 4** will be clustered in a group as text region candidate, where

$\text{AVG}\left(R_h\left(i\right)\right)=\frac{1}{M/8}\sum_{i=0}^{(M/8)-1}R_h\left(i\right)$. From these groups, we choose the most significant one, which concentrates higher energy and is closer to the middle of the image by using Equation (8).

$$W_r\left(m\right)=\frac{\sum_{i=r_1(m)}^{r_2(m)}R_h\left(i\right)}{\left|\frac{r_1\left(m\right)+r_2\left(m\right)}{2}-\frac{\left(M/8\right)}{2}\right|} \tag{8}$$

where $\left(r_1\left(m\right),\ r_2\left(m\right)\right)$ are the vertical boundaries of the *m*-th group candidate, and M is the number of rows. After this decision, the desired vertical borders will be $\left(L_v=r_1\right)$ and $\left(U_v=r_2\right)$ as shown in the **Figure 5**. As an example, there are two groups in **Figure 4**, but the groups span from row 30 to 40 is chosen to be the most significant one.

From $E_h\left(i,j\right)$, we calculate $R_h\left(i\right)$ in **Figure 5**. Then, the rows with values above $\text{AVG}\left(R_h\left(i\right)\right)$ are clustered in candidate groups, delimitated by its row boundaries $r_1\left(m\right)$ *and* $r_2\left(m\right)$ for the *m-th* group.

After the elimination of non-text rows we then eliminate the non-text columns of the text candidate region. In order to do this, we calculate $T(n)$ by applying a median filter (window length equal to three) both to $C_h\left(j\right)$ and $C_v\left(j\right)$ within the borders $\left(L_v,U_v\right)$, and then by multiplying both results as in Equation (9). Then, we apply the same median filter to $T\left(n\right)$, so that we will have a smoothed version of the Text Energy accumulated on each column as in Equation (10) as shown in **Figure 6**.

Given the chosen text candidate, we calculate $C_h\left(j\right)$ and $C_v\left(j\right)$ within its borders L_v, and U_v in **Figure 6**. Then, we apply to each of them a median filter with the window size 3, and multiply the result, to which we apply a median filter (window size 3) again.

$$T\left(n\right)=\left\{\frac{1}{3}\sum_{j=n-1}^{n+1}C_h\left(j\right)\right\}\left\{\frac{1}{3}\sum_{j=n-1}^{n+1}C_v\left(j\right)\right\}\quad n\in\left\{0,\cdots,\left(N/8\right)-1\right\} \tag{9}$$

$$S\left(j\right)=\frac{1}{3}\sum_{n=j-1}^{j+1}T\left(n\right)\quad j\in\left\{0,\cdots,N/8\right\} \tag{10}$$

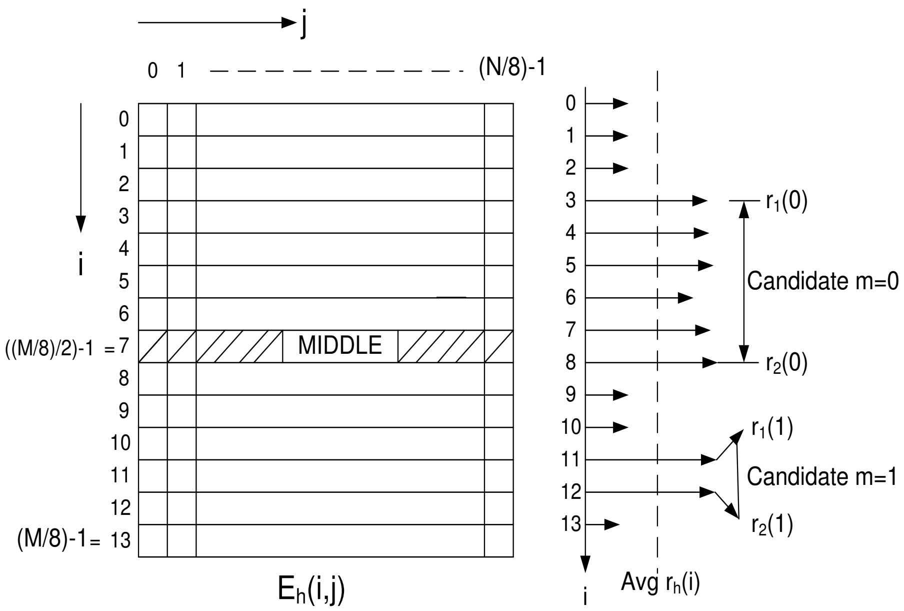


Figure 5. Calculation of the text region candidates.

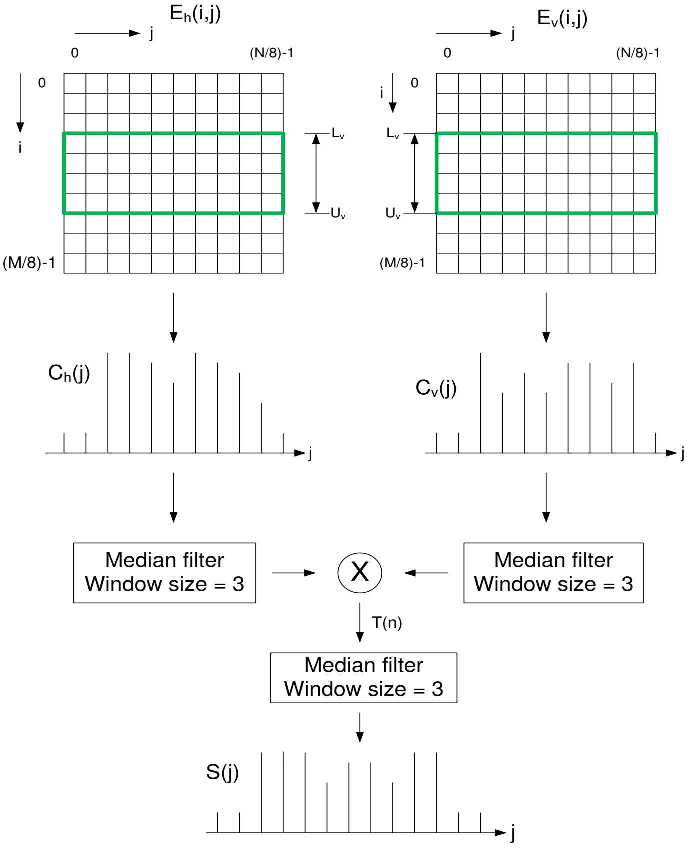


Figure 6. Calculation of *S*(*j*).

At this point, we compute the iterative algorithm described by Equations (11) through (14), **Figure 7** which searches for the background level in $\bar{S}(j)$ ($S(j)$ normalized in [0, 1]). Starting from a_0 as shown in dash line in **Figure 7(a)**, it builds on each iteration n the estimated background group g_n in $\bar{S}(j)$, made up by those columns with values below a_{n-1} which are previous estimation of the background level. It stops at a_{stop}, dotted line in **Figure 7(c)** when the variance of g_n is low enough for this group to be considered background. The $\text{AVG}(g_n)$ is the average value of the group g_n. Just the regions above the line will be considered, from which the middle peak will be chosen.

Initially, a_0 is the average of $\bar{S}(j)$, and g_1 group of rows below a_0. On each iteration n, we calculate the values of g_n by using Equation (12) and a_n by using Equation (13), until the condition on Equation (14) is fulfilled. The rows with values above a_{stop} will form the foreground group.

$$a_0 = \frac{1}{(N/8)} \sum_{j=0}^{(N/8)-1} \bar{S}(j) \tag{11}$$

Repeat:

$$g_n = \left\{ \bar{S}(j) \forall j \middle| \bar{S}(j) < a_{n-1} \right\} \tag{12}$$

$$a_n = \frac{1}{2}\left(a_{n-1} + \text{AVG}(g_n)\right) \tag{13}$$

Until

$$\text{SQRT}\left(\text{VAR}(g_n)\right) \le 0.02 \to a_{\text{stop}} = a_n \tag{14}$$

Figure 8 shows an example of overall text localization based on the proposed procedure. Also, **Figure 9** shows the successful examples of the text localization using our proposed method for the images under various conditions.

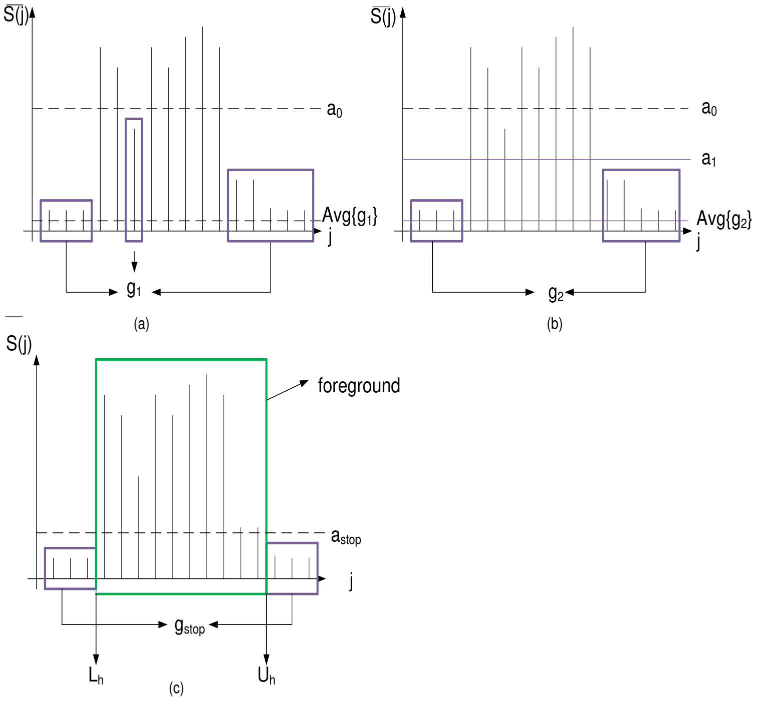


Figure 7. Iterative algorithm.

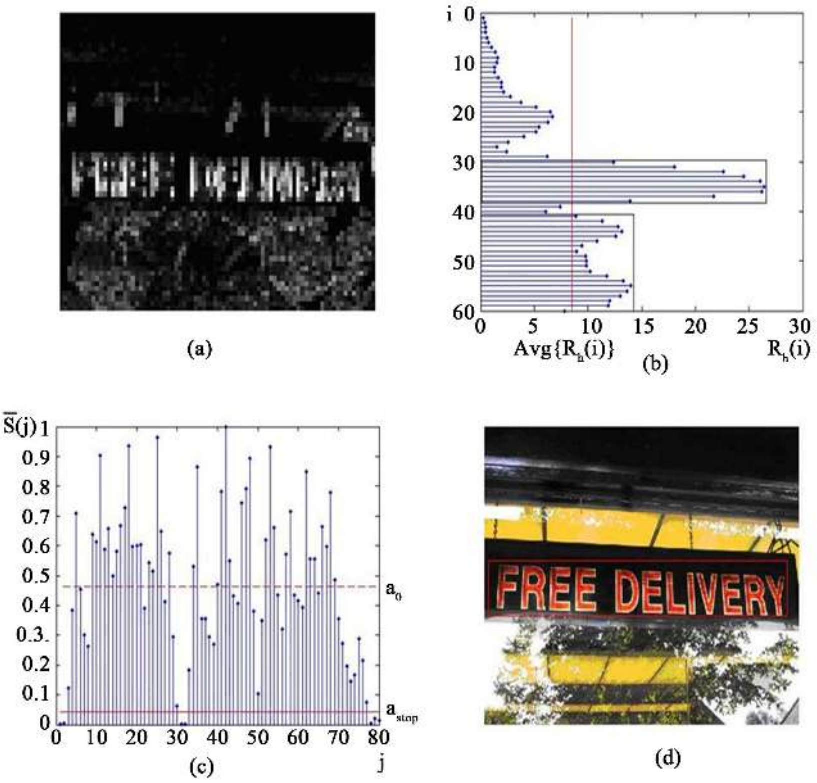
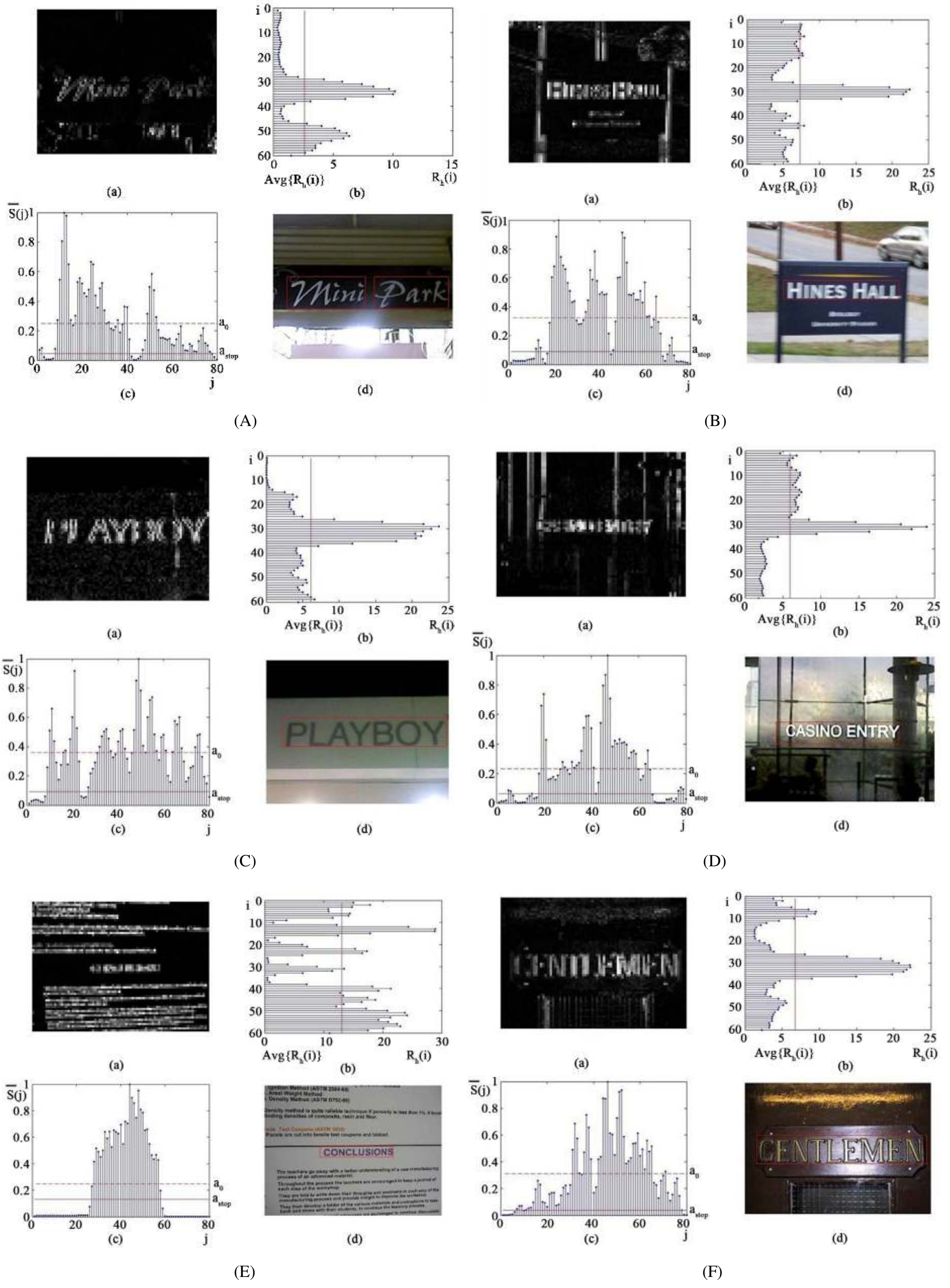


Figure 8. (a) $E_h(i,j)$ representation of the target image; (b) $R_h(i)$ representation of the target image and its mean (line); (c) $\overline{S}(j)$ representation, a_0 (dash line) and a_{stop} (continuous line) for the chosen region; and (d) Text localization results in the original image.



(A) (B) (C) (D) (E) (F)

Figure 9. Examples of successful Text Localization—Locating the text in (A) Uneven lighting condition; (B) Blurred image; (C) Dim light conditions; (D) Dark uneven colored background; (E) Text image; (F) Dark colored Image.

4. Verification of the Methodology

Even though many Text Localization methods are proposed so far as described in Section 2, we considered that none of them met our requirements for implementing a real Scene Text Translation System. Our goal is to develop a fast algorithm to work in a reasonably small time period on a device with low computational resources such as mobile phone, and accurate enough to handle the most common challenges present in natural scene images. On the one hand, some methodologies proved to be very reliable and accurate, but their complexity was prohibitive for our purpose. On the other hand, some performed on a very short time lapse, but they lacked on accuracy. Also, most of them are highly dependent on the application for which they have been designed, so we considered them not able to perform well under the specific features of the input images of our system. For all these reasons, we developed a non-general purpose Text Localization algorithm, fast, reliable and useful on the Scene Text Translation System scenario.

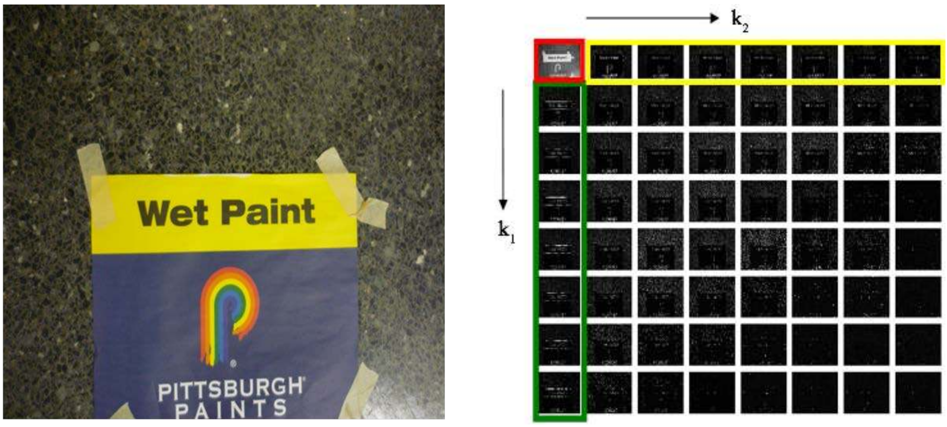
A. Text Localization in the compressed domain

Images are usually stored using the JPEG format on the mobile phones, since they lack of memory space. A methodology that avoids the process of uncompressing the image is on big advantage against those others which have to perform the uncompressing. Moreover, it is almost immediate to extract the DCT coefficients from the JPEG image, so are the frequency features which can be used to extract the text. For all these reasons, and given the outstanding time processing results achieved by some algorithms [11], we have decided to develop a Text Localization algorithm on the compressed domain.

B. Text energy

We have the scene image $M \times N$ as shown in **Figure 10(a)**, and we apply the DCT to each 8 × 8 block. If we want to see the importance of each component (k_1, k_2), we can take each component (k_1, k_2) from each block. With all of them, we will have a matrix $(M/8)\times(N/8)$ with numerical values, containing the (k_1, k_2) components of the (i, j) block on each (i, j) pixel. Thus, we have the $8\times 8 = 64$ small images in **Figure 10(b)**. Each pixel (i, j) of the image (k_1, k_2) in **Figure 10(b)** represents the value of the component (k_1, k_2) on the (i, j) 8×8 DCT block of the scene image. Note that this representation is not related to the algorithm, but gives a useful view of the importance of each single component on the image. As we mentioned before, the frequency components of the image play a key role in our algorithm. Each *8x8* blocks has different frequency information depending on the amount of variations in each block and on their strength. In this regard, blocks with no variations, such as simple background blocks, will present most of their frequency information around the DC component $(k_1 = 0, k_2 = 0)$. Also, blocks with vertical or horizontal variations will concentrate their information on their vertical k_1 and horizontal k_2 components, respectively.

Text characters present variations with respect to the background along several directions, but the horizontal and vertical variations are the stronger ones. This rule does not apply to general backgrounds, where the variations can be concentrated on any direction, not following a specific pattern, like characters do. Consequently, the



(a) (b)

Figure 10. (a) $M \times N$ scene text image; (b) Representation of the importance of each component (k_1, k_2).

horizontal and vertical components should be useful to localize the text blocks. On our experiments, the information of the background blocks did not follow any pattern regarding the coefficients on which it was concentrated, but the information of the text blocks was specially concentrated on the coefficients $\{(0,1),(0,2),(0,3),(0,4)\}$, and $\{(1,0),(2,0),(3,0),(4,0)\}$, from which we defined the Horizontal and Vertical Text Energy in Equation (2) and (3). In the **Figure 10**, we represented the values of each coefficient of a scene text image. It is clear that the selected coefficients above are the best ones in order to differentiate the foreground and the background parts, highlighting the text areas, while hiding the other elements in the image.

In the scene text image representation on the DCT domain in **Figure 10(b)**, each square contains the information of a specific component (k_1,k_2). The DC component, which represents the average value of the block, is the red one. The pure horizontal variations are represented by the yellow components and the pure vertical variations by the green components. Combinations of both are represented by the non-colored components.

C. Text line detection

Usually, Text Localization algorithms try to classify individual 8×8 DCT blocks either as text or as background, assuming that text blocks have high frequency components, while background ones do not. This is true when the foreground-background contrast is high, and the background is clean and simple. However, in more realistic situations, it is necessary to take other characteristics into account, and very more complex classification algorithms have to be used, increasing considerably the computational expenses. On the contrary, our approach just takes into account the layout of the texts to find groups of text blocks instead of individual text blocks, without the need of any complex processing.

First of all, it is clear that text lines will accumulate higher Text Energy along them than non-text lines, even if the background is very complex. Also, uneven lightening and lightening distortions may cause the misclassification of individual blocks, but their influence is insignificant when considering big block sets. For these reasons, we defined the Text Energy in rows in Equations (4) and (5) and columns in Equations (6) and (7). By the **Figure 10**, it is clear that these definitions are a powerful tool to localize the text parts of an image. Text lines accumulate far more Text Energy than background lines, producing remarkable peaks on both histograms. However, in the case of very complex scenes as shown in **Figure 12**, these definitions may cause the detection of false texts. Anyway, it does not represent a problem for the algorithm, because this false detection can be avoided by the selection of the correct text to process explained in the next section.

D. Selection of the text to process

Based on the behaviour of the potential users' of a Scene Text Translation system, we choose just one of the text candidates to be processed: the text that the user wants to translate. The natural behaviour of a user using translation system on his mobile phone would be to focus the image on the target text. However, by just taking into account the distance to the middle of the texts, the system would lack of flexibility, since the user would have to put the target exactly in the middle of the picture. Then, we include the Text Energy as the measure of the importance of the texts. The text will be detected and centred in the image. As a result, further steps including the text extraction will be processed to be translated, avoiding the processing of undesired parts of the image, as well as background parts misclassified as text.

5. Experimental Test and Results

We implemented the algorithm using Matlab 7.6.9 (R2008a) on a Pentium 4 PC (CPU 3.8 GHz) in order to verify its performance. We tested the algorithm with a set of 182 images, with which we tried to reasonably represent the situations for which a user could use the system. The proposed method required a computing time of 0.12 seconds for 0.3 mega pixel pictures using the stated software and PC. We classified the detection results into six groups as shown in **Table 1**.

- The candidate areas contain the whole text region, and they are never bigger than the 1.1 times the desired text area size as shown in **Figure 11**. This group included 146 images (80.2%).
- The candidate areas contain the whole text region, and its size is from 1.1 to 1.25 times the desired text area size. This group included 18 images (9.89%).
- The candidate areas contain the whole text region, and its size is from 1.25 to 1.5 times the desired text area size. This group included 4 images (2.19%).
- The candidate areas do not contain the whole text region, and its size is from 0.9 to 1 times the desired text area size. This group included 9 images (4.94%).

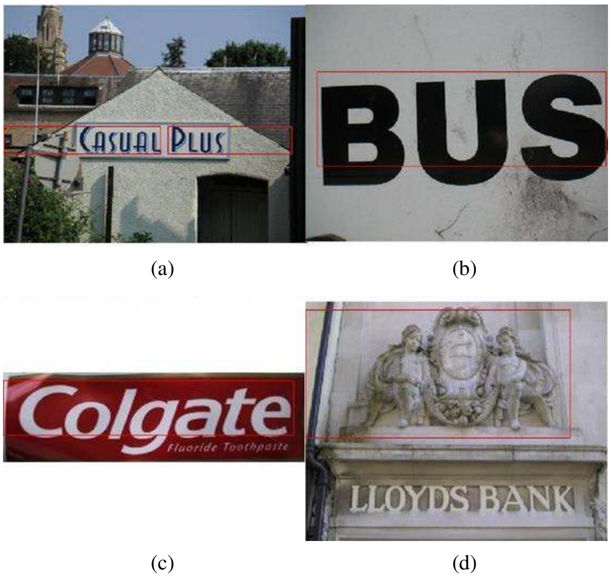


Figure 11. Examples of text localization error: (a) Area excess; (b) Area shortage; (c) Area excess & shortage; and (d) Failure.

Table 1. Results of text area localization.

	Success	Partial success			Failure	
Size of the text	1.1	1.1 to 1.25	1.25 to 1.5	0.9 to 1	0.9 to 0.2	Text cannot be localized
No of images	146	18	4	9	2	3
Percentage	80.2	9.89	2.19	4.94	1.10	1.65

- The candidate areas do not contain the whole text region, and its size is from 0.9 to 0.2 times the desired text area size. This group included only 2 images (1.10%).

The text area cannot be localized. This group included only 3 images (1.65%).

It is possible to classify the results in three groups: success, failure, and partial success. We consider a result to be successful when the text is completely localized, and the boundaries do not cut any foreground or select any excessive background. On the other hand, a result will be considered as a failure when the foreground is not localized, or the localization cuts out a significant part of the text. Finally, we consider it to be partial success when the text is localized, but there exists a small area excess and/or shortage comparing to the ideal boundaries.

Figures 11(a)-(c) show some examples of partial success in locating the text. **Figure 11(d)** is an example of failure, where the text is not localized because its contrast with the background is very smooth, plus the fact that is not well centered on the image, so the system assumes that the user does not want to process it.

6. Performance Evaluation

Most of the research areas in Computer Vision and Pattern Recognition face several difficulties in performance evaluation. Some of the difficulties in performance evaluation for the text localization algorithms are as follows: [5]

- There is no specific public domain database of images containing text to compare two text localization methods. So, we have to evaluate our own database for the performance evaluation of our algorithm.
- The performance results of text localization algorithm are different in different algorithms. Some algorithms extract the characters in the image, some other algorithm extract all the text in the image, while others extract the important text region in the image. Here we extract the important text from the image. So, it is very difficult to compare most of the algorithms.
- The output results format for text localization algorithms is different for different algorithms.

- The performance of different algorithms usually is to be compared using pixel by pixel, rectangle by rectangle by rectangle or character by character comparison. But it is very difficult to compare all the algorithms in the same way shown above.

A. Text data in different angles under normal conditions:

Consider an image under normal conditions with plain background as shown in **Figure 12**. We already mentioned above that the text has to be centered for successful text localization for our algorithm. But here we considered an image with same text and background under different circumstances such as when the text is zoom in or text is not centered or distorted or text is not present in the image.

In **Figures 12(a)-(d)**, the text is zoom in and the text localization is completely successful. In **Figure 12(d)**, the text is localized in two parts. Since there is the gap between the texts, the meaning of the text may change but the localization is successful because the complete important text is localized. In **Figure 12(e)**, the text is in the center of the image is in diagonal shape. The important text in the image is localized, but the localization of unnecessary background on both sides of the text in the image makes the result as partially successful. In **Figure 12(f)**, the localization is unsuccessful because the frequency caused by noise in the image is more than the text data present in the image. So, the localization is unsuccessful. In **Figure 12(h)**, the text present in the image is not clear. The localized text area is more than the text area present in the image. So, the localization of the image is partially successful. In **Figure 12(i)**, there is no text in the image. Due to the small energy generated by the background the unwanted background is localized. Anyway the images with no text are not useful for the algorithm.

In **Figure 13**, the localization is unsuccessful because the frequencies generated by the unwanted background dominated the text in the image in both vertical and horizontal directions.

We already made two exceptions that the images are stored in JPEG format and the text in the image has to be centered for successful text localization of our algorithm. We can see from the above example that if the important text is centered in the image then most of the times the results are either successful or partially successful.

B. Images under three different conditions:

In **Figure 14**, the text localization of the same image is shown under three different conditions such as natural, flash light and dark conditions. Under normal conditions the localization of the important text is perfect. But under flash light and dark conditions only the localization is not completely successful. In case 1: under lighting conditions there is too much of light which ignored the important text and caused the noise and in dark light condition even the background is entirely dark the text in the image is of different color and so the localization is successful. In case 2: under flash light the localization is successful but under dark light condition the algorithm

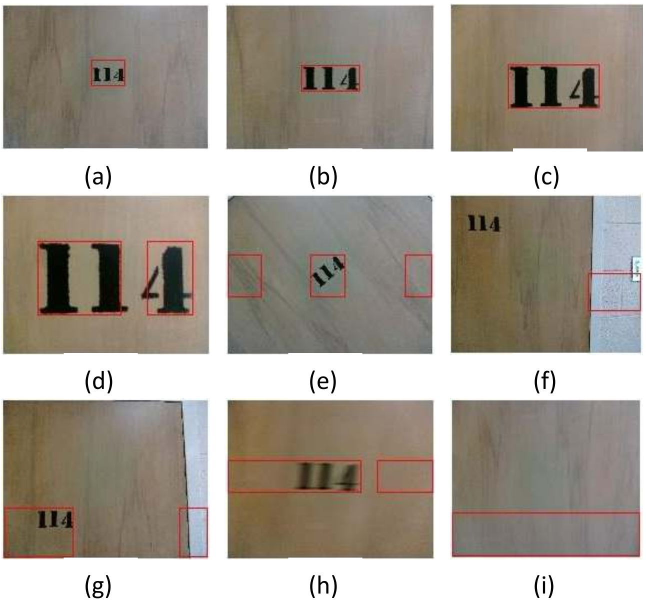
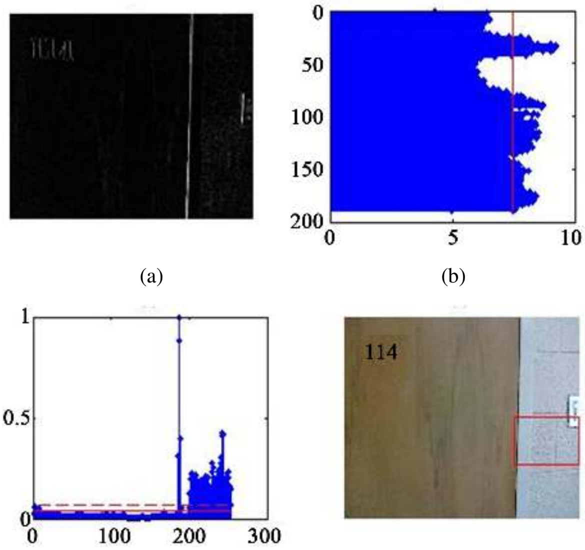


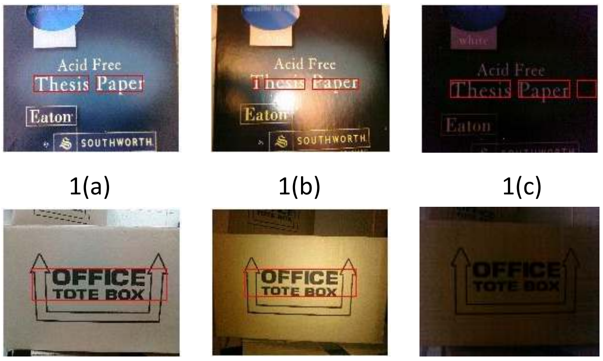
Figure 12. Text localization for the same text in different angles of the image.



(a) (b)

(c) (d)

Figure 13. Example of unsuccessful text localization for the **Figure 12(f)**.



1(a) 1(b) 1(c)

2(a) 2(b) 2(c)

Figure 14. Images in 1(a) and 2(a) [natural condition], 1(b) and 2(b) [flash light], 1(c) and 2(c) [dark light].

cannot able to detect the boundaries of the text due to the color of the text which is as dark as the image. So, the algorithm cannot able to detect the boundaries of the image causing the failure result.

C. Results of text localization under different conditions:

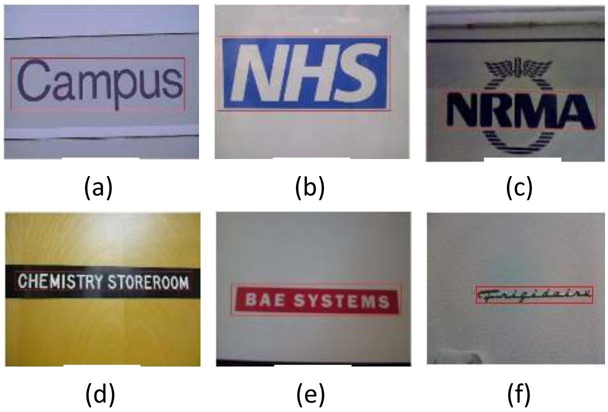
The algorithm we produced here works under "real world" conditions such as natural or normal conditions, dark colored background, complex colored background, commercial sign board, dim light and blurred images. Now we are going to examine the success rate of our algorithm under these conditions.

1. Normal Condition:

Figure 15 shows the images with successful text localization under normal conditions. Normal conditions means the text is centered in the image; back ground is plain or not complicated with colors and with natural light conditions. These kinds of images always have high success rate. Our algorithm perfectly suits for these kinds of images. Based on the experiment involving 84 images under normal or perfect conditions, we achieved an accuracy rate of 87%. Some images are partial success due to area excess or area shortage or both with a success rate of 13%. There are no failures in these kinds of images.

2. Commercial Sign Board Images:

Commercial signboard images are one of the important applications for our algorithm. From **Figures 15(a)-(f)** are the commercial sign images with lighting. Even there are so many back ground colors at the back; the main text is highlighted more due to the light present at the back of the text. The background noise is ignored and it is



(a) (b) (c)

(d) (e) (f)

Figure 15. Successful text localization in normal conditions.

very easy to localize the text in these kinds of images. Apart from normal condition images, these kinds of images also have high success rate.

Figures 16(g)-(l) are the commercial sign images under natural light condition. These kinds of images don't have the success rate as commercial sign images with lighting. Images from **Figures 16(g)-(j)** are successful text localization because the background is not complex and the data is easily identified. Consider images (k) and (l) where the background is too complex. The frequencies generated by the complex background in both horizontal and vertical direction are more than the text present in the image. So, instead of the text the background part of the image is localized and hence there is a failure in the localization of these images. In the below **Figure 16(l)**, we can see the reasons for the failure of the localization of an image. The frequency generated by the background noise is more than the text present in the image. Experiments were performed on 48 commercial signboard images and we achieved a success rate of 75% and partial success rate of 19% achieving a total success rate of 94%. Algorithm failed to localize text in 3 images due to more noise in the background than the text present in the foreground.

In **Figure 17**, you can see that the noise generated by the background is more than the text present in the image.

3. Dark Colored Background Images:

Text localization of dark colored back ground images are always successful because of the darkness in the background of the image, the text is clearly visible and the localization of the important text in the image becomes easier.

Text localization of dark colored back ground images are always successful because of the darkness in the background of the image, the text is clearly visible and the localization of the important text in the image becomes easier. In **Figure 18**, we can see that all the images are with dark colored back ground where the text is clearly visible. The frequencies generated by the text are more than the background presented in the image and so localization of the text becomes easier. Only the **Figure 18(f)** has localized more area than the text area presented in the image. Even though localization is partially successful, the important text in the image is localized successful. We got all these images from ICDAR 2003 dataset and achieved a high success rate. Based on the experiments using 30 images under dark colored background conditions we achieved a high success rate of 86.67% and partial success rate of 13.33% and achieving an overall success rate of 100%.

4. Complex Background Images:

Complex background images are nothing but the images with too many things in the background. If the important text is centered, then these kinds of images are successful or partially successful otherwise they have high chances of failure. **Figure 19** shows the examples of successful text localization with complex background. Only the **Figure 19(d)** and **Figure 19(f)** are partially successful because of the excess area localized in the image. We conducted experiments on 36 images and achieved a success rate of 77.78%. Some images are partial success with a success rate of 16.67% achieving an overall success rate of 94.44%. The algorithm failed to localize text in 2 images due to external noise provided by the background.

5. Multi Text Images:

Multi text images in **Figure 20** are the images with the text all over the image. Sometimes important text in

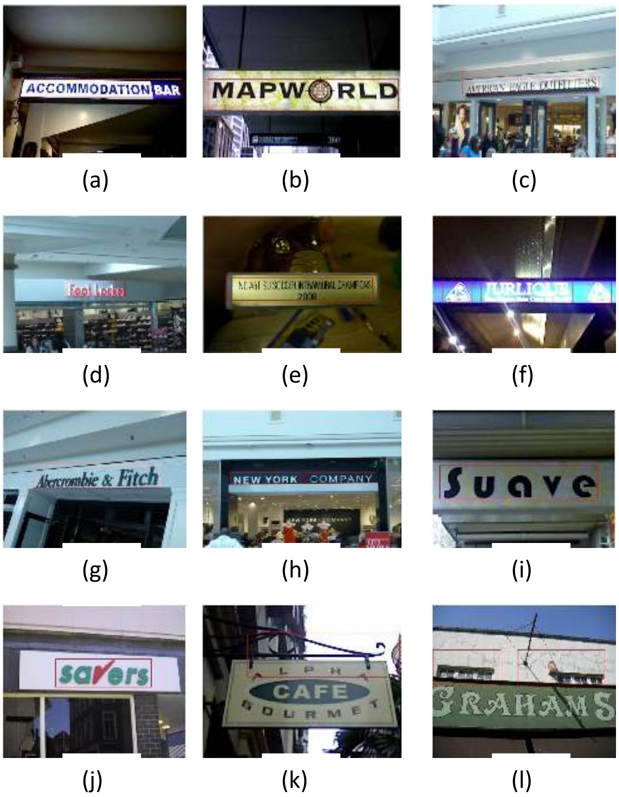


Figure 16. Commercial sign board images (a)-(f) under light conditions, (g)-(l) under natural light condition.

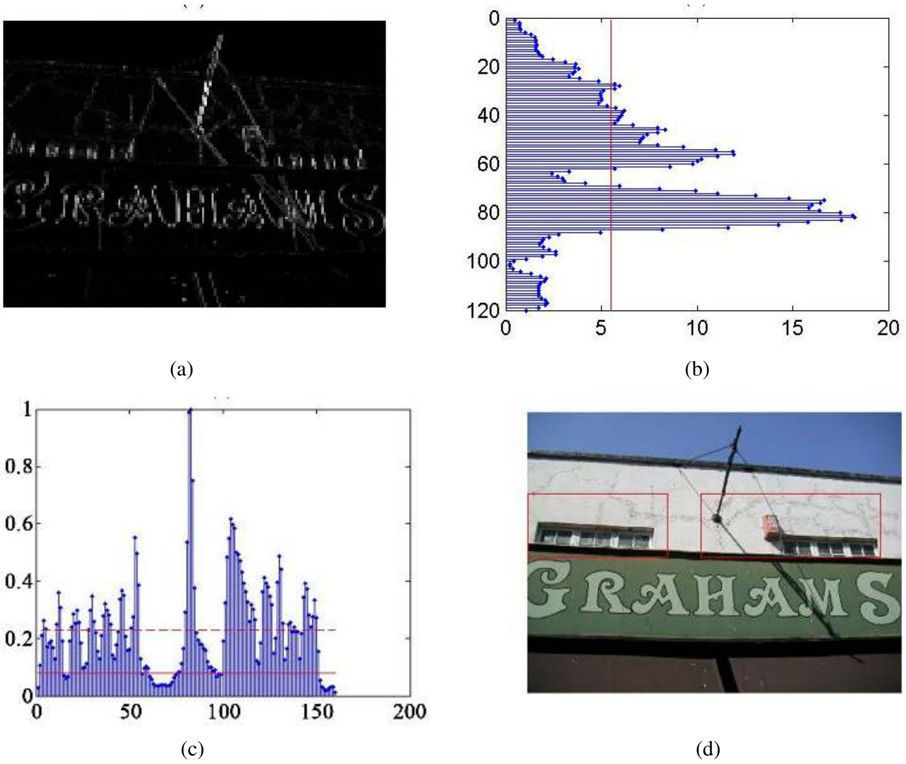
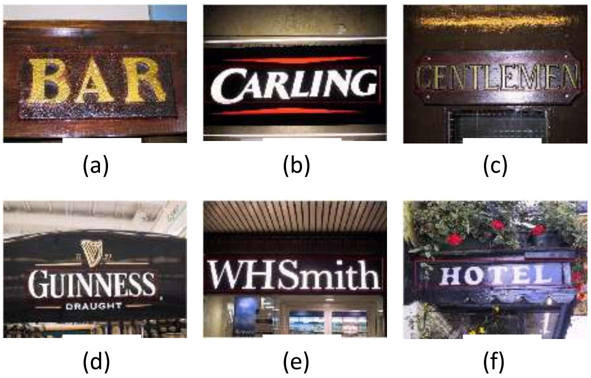


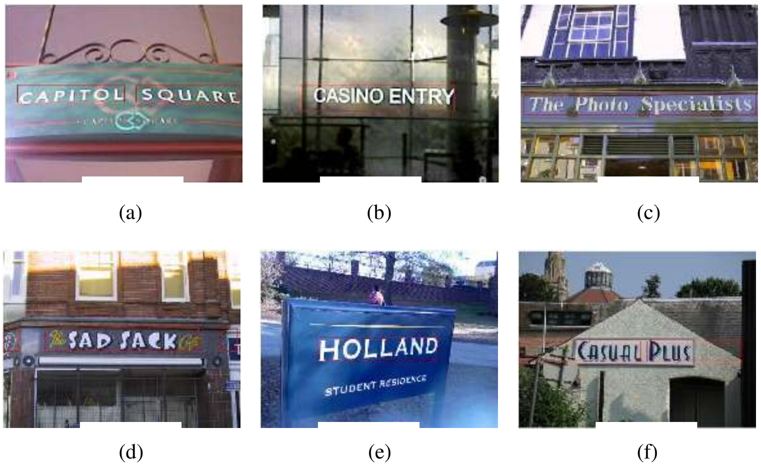
Figure 17. Example of unsuccessful text localization for the **Figure 16(l)**.



(a) (b) (c)

(d) (e) (f)

Figure 18. Successful text localization for dark colored back ground images.



(a) (b) (c)

(d) (e) (f)

Figure 19. Successful text localization for complex background images.

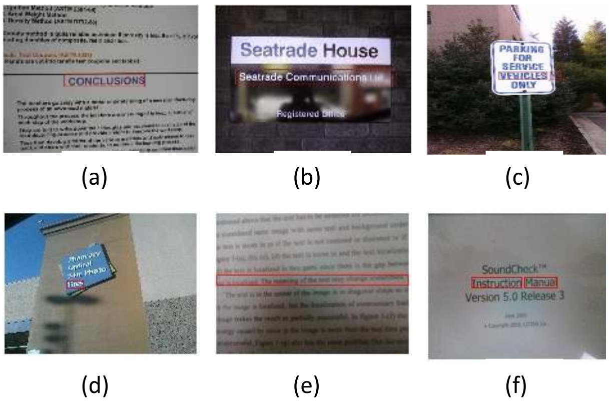
the images cannot be localized because our algorithm concentrates on the text in the center of the image. Anyway our algorithm doesn't deal with text document. It deals with mostly commercial signboard or tags. These kinds of images are given least preference in our algorithm. There can be more than one important text in the image but if you can able to center the important text in the image, then the algorithm is able to localize the important text in the image. The success rate of the images depends on the localization of the text in the center of the image. We took around 6 images and for all the images the text in the center of the image is localized successfully and achieved a success rate of 100%.

6. Dim Light or Blurred Images:

Success ratio of text localization for dim light or blurred images is very low as shown in **Figure 21**. In dim light images the text is very hard to determine from the background due to very low light. If the background and text both are dark then the localization for these kinds of images are even harder. If the text is blurred in the image the text can be localized but the information can't be clearly visible. Blurred images are the images taken while travelling, shaky camera, disturbance etc., Based on the experiment on 35 images; we achieved a success rate 62.38%. The partial success rate is 25.71%, achieving a total success rate of 88.09%. Algorithm failed to localize text in around 4 images causing a failure rate of 11.91% which is high when compared to other conditions.

7. ICDAR 2003 Images Dataset:

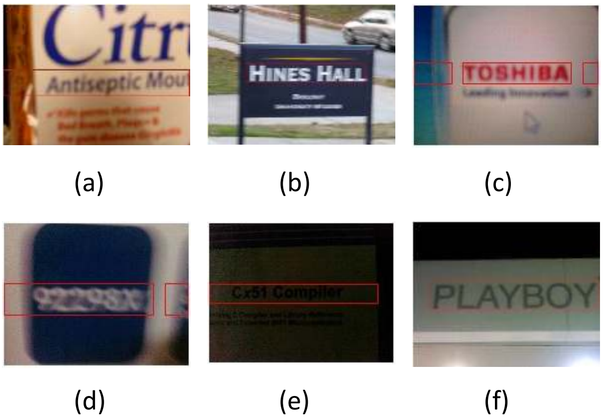
We tested our algorithm with ICDAR 2003 dataset, a public domain database which contains 83 images. In **Figure 22**, we can see the successful text localization of our algorithm for ICDAR 2003 dataset. We achieved an accuracy of 72.28% and a partial success rate of 15.66% achieving a total success rate of 87.94%. Algorithm failed to recognize text in 10 images at 12.06% rate. **Table 2** presents the success rate of our algorithm under different conditions.



(a) (b) (c)

(d) (e) (f)

Figure 20. Successful text localization for multi text images.



(a) (b) (c)

(d) (e) (f)

Figure 21. Localization for Dim light or blurred images.

Table 2. Results of text localization in different conditions.

	Total No. of Images	Success		Partial Success		Failures	
		No. of Images	%	No. of images	%	No. of Images	%
Perfect or Normal Conditions	84	73	87	11	13	-	-
Commercial Sign Board Images	48	36	75	9	18.7	3	6.25
Dark colored back ground	30	26	86.7	4	13.3	-	-
Complex Background Images	36	28	77.8	6	16.7	2	5.5
Multi Text Images	6	6	100	-	-	-	-
Dim Light or Blurred Images	35	22	62.9	9	25.7	4	11.4
ICDAR Images	83	60	72.3	13	15.6	10	12.1
Over All Images	303	239	78.9	46	15.2	18	5.9

7. Summaries and Conclusions

In this paper, we proposed a simple, fast, and accurate Text Localization algorithm to be part of a mobile phone based translator of text embedded on scene text images. Among all the possible types of input documents of a Text Information Extraction system, scene text documents are, by far, the most challenging, due to the multiple degradations that they suffer, and the absence of any known pattern both on the image scenario, and on the text characteristics. Existing algorithms did not fit our purpose. On the one hand, those accurate enough to be part of

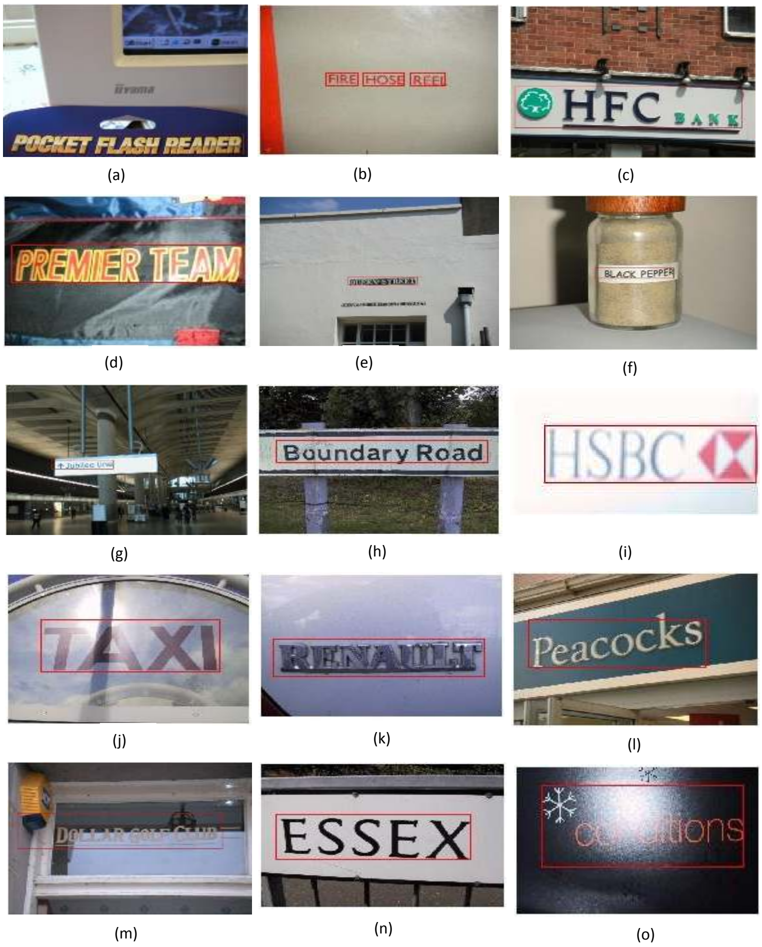


Figure 22. Successful text localization of ICDAR 2003 dataset.

a reliable system were too expensive in computational terms to be implemented on a mobile phone. On the other hand, those who were simple enough to perform fast even on a device with reduced computational resources, were not reliable enough to be considered as a good option. Moreover, most of them were developed for specific applications different from automatic language translation. Therefore, we developed a fast and reliable algorithm to be part of our scene text translation system.

In most of these devices, images are stored using the JPEG compressed format. As our goal is to perform a system as fast as possible, we use this compressed information in order to localize the text, without the need of decompress the image. The JPEG format divides the image in 8 × 8 blocks, and calculates and stores the Discrete Cosine Transform (DCT) of each of them. Thus, using the DCT coefficients it is possible to treat each block in the frequency domain, and decide whether it contains text based on this information or not. We observed that text blocks concentrate their information on some particular coefficients, based on which we define the Text Energy of each block, as the sum of the values of these coefficients. Instead of classifying the blocks in text or non-text blocks, as other algorithm do, we calculate the Text Energy contained by whole rows and columns. Using this approach, the difference between foreground and background areas becomes more evident, and therefore the accuracy of the localization increases considerably. In order to speed up the processing of further steps, we just select the text that the user wants to translate, on which the picture is supposed to be focused. We ensure that the system does not translate unnecessary texts on which the user is not interested, reducing the processing time. After a fine adjustment of the boundaries of the selected area, the text is successfully localized.

We compared our algorithm with different algorithms as shown in **Table 3**. Most of the algorithms are based upon recognizing the characters. Our experiment is on text line detection and we achieved more success rate when compared to other algorithms. We conducted experiments on 303 images in various conditions and achi-

Table 3. Comparison of different text localization algorithms.

Algorithm	No. of images	Text line detection	
		True detection	False detection
Smith and Kanade	63	30	9
V. Wu, Manmatha, Riseman	48	4139	267
Kim	50	107	17
Our algorithm	303	285	18

eved an accuracy of 78.9%. Some images are partially success due to small amount of area excess or area shortage or both with a success rate of 15.2% thus by achieving overall success rate of 94.1% which is more in accuracy than any other algorithm.

Using the Text Energy concept, the algorithm delineates the text region in a rectangular shape. We have done experiments on different images including characters of various sizes, different colors, under uncontrollable lighting conditions and achieved an accuracy of 80.2% in a processing time of 0.12 seconds. Table 2 presents the success rate of images under different conditions.

In conclusion, we presented a simple and affordable text localization algorithm for mobile phones. It shows high accuracy rates under very different "real world" conditions and works in a very short amount of time, so it is adequate to be implemented on a device with low computational resources. We plan to integrate it on a successful automatic language translator for mobile phones, for which we have also developed a Text Extraction algorithm under the same constraints: high accuracy and reduced processing [33].

References

[1] Liang, J., Doermann, D. and Li, H.P. (2005) Camera-Based Analysis of Text and Documents: A Survey. *International Journal of Document Analysis and Recognition* (*IJDAR*), **7**, 84-104.

[2] Wallace, G.K. (1991) The JPEG Still Picture Compression Standard. *Communications of the ACM*, **34**, 30-44. http://dx.doi.org/10.1145/103085.103089

[3] Candrall, D.J. (2001) Extraction of Unconstrained Caption Text from General-Purpose Video. Thesis in Computer Science and Engineering, The Pennsylvania State University, University Park.

[4] Shiratori, H., Goto, H. and Hobayashi, H. (2006) An Efficient Text Capture Method for Moving Robots Using DCT Feature and Text Tracking. 18*th International Conference on Pattern Recognition*, **2**, 1050-1053. http://dx.doi.org/10.1109/ICPR.2006.243

[5] Antani, S., Gargi, U., Crandall, D., Gandhi, T. and Kasturi, R. (1999) Extraction of Text in Video. Technical Report of Department of Computer Science and Engineering, CSE-99-016, The Pennsylvania State University, University Park.

[6] Jung, K., Kim, K.I. and Jain, A.K. (2004) Text Information Extraction in Images and Video: A Survey. *Pattern Recognition*, **37**, 977-997. http://dx.doi.org/10.1016/j.patcog.2003.10.012

[7] Mancas-Thillou, C. and Gosselin, B. (2007) Natural Scene Text Understanding. In: Obinata, G. and Dutta, A., Eds., *Vision Systems*: *Segmentation and Pattern Recognition*, I-Tech Education and Publishing, Vienna, 307-332.

[8] Shim, J.C., Dorai, C. and Bolle, R. (1998) Automatic Text Extraction from Video for Content-Based Annotation and Retrieval. *Proceedings of International Conference on Pattern Recognition*, **1**, 618-620.

[9] Jain, A.K. and Yu, B. (1998) Automatic Text Location in Images and Video Frames. *Pattern Recognition*, **31**, 2055-2076. http://dx.doi.org/10.1016/S0031-3203(98)00067-3

[10] Lim, Y.-K., Choi, S.-H. and Lee, S.-W. (2000) Text Extraction in MPEG Compressed Video for Content-Based Indexing. *Proceedings of the* 15*th International Conference on Pattern Recognition*, **4**, 409-412.

[11] Zhong, Y., Zhang, H.J. and Jain, A.K. (2000) Automatic Caption Localization in Compressed Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 385-392. http://dx.doi.org/10.1109/34.845381

[12] Ohya, J., Shio, A. and Akamatsu, S. (1994) Recognizing Characters in Scene Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 214-224. http://dx.doi.org/10.1109/34.273729

[13] Lee, C.M. and Kankanhalli, A. (1995) Automatic Extraction of Characters in Complex Images. *International Journal of Pattern Recognition Artificial Intelligence*, **9**, 67-82. http://dx.doi.org/10.1142/S0218001495000043

[14] Messelodi, S. and Modena, C.M. (1992) Automatic Identification and Skew Estimation of Text Lines in Real Scene

Images. *Pattern Recognition*, **32**, 791-810. http://dx.doi.org/10.1016/S0031-3203(98)00108-3

[15] Zhong, Y., Karu, K. and Jain, A.K. (1995) Locating Text in Complex Color Images. *Pattern Recognition*, **28**, 1523-1535. http://dx.doi.org/10.1016/0031-3203(95)00030-4

[16] Kim, E.Y., Jung, K., Jeong, K.Y. and Kim, H.J. (2000) Automatic Text Region Extraction Using Cluster-Based Templates. *Proceedings of International Conference on Advances in Pattern Recognition and Digital Techniques*, Calcutta, 418-421.

[17] Hase, H., Shinokawa, T., Yoneda, M. and Suen C.Y., (2001) Character String Extraction from Color Documents. *Pattern Recognition*, **34**, 1349-1365. http://dx.doi.org/10.1016/S0031-3203(00)00081-9

[18] Smith, M.A. and Kanade, T. (1995) Video Skimming for Quick Browsing Based on Audio and Image Characterization. Technical Report CMU-CS-95-186, Carnegie Mellon University, Pittsburgh.

[19] Lee, S.-W., Lee, D.-J. and Park, H.-S. (1996) A New Methodology for Gray-Scale Character Segmentation and Recognition. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, **18**, 1045-1050. http://dx.doi.org/10.1109/34.541415

[20] Hasan, Y.M.Y. and Karam, L.J. (2000) Morphological Text Extraction from Images. *IEEE Transactions on Image Processing*, **9**, 1978-1983. http://dx.doi.org/10.1109/83.877220

[21] Park, S.H., Kim, K.I., Jung, K. and Kim, H.J. (1999) Locating Car License Plates Using Neural Networks. *IEEE Electronics Letters*, **35**, 1475-1477. http://dx.doi.org/10.1049/el:19990977

[22] Wu, V., Manmatha, R. and Riseman, E.M. (1999) Text Finder: An Automatic System to Detect and Recognize Text in Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, 1224-1229. http://dx.doi.org/10.1109/34.809116

[23] Wu, V., Manmatha, R. and Riseman, E.R. (1997) Finding Text in Images. *Proceedings of the* 2*nd ACM International Conference on Digital Libraries*, Philadelphia, 23-26 July 1997, 3-12.

[24] Jain, A.K. and Bhattacharjee, S. (1992) Text Segmentation Using Gabor Filters for Automatic Document Processing. *Machine Vision and Application*, **5**, 169-184. http://dx.doi.org/10.1007/BF02626996

[25] Jung, K. (2001) Neural Network-Based Text Location in Color Images. *Pattern Recognition Letters*, **22**, 1503-1515. http://dx.doi.org/10.1016/S0167-8655(01)00096-4

[26] Sin, B., Kim, S. and Cho, B. (2002) Locating Characters in Scene Images Using Frequency Features. *Proceedings of International Conference on Pattern Recognition*, **3**, 489-492.

[27] Mao, W., Chung, F., Lanm, K. and Siu, W. (2002) Hybrid Chinese/English Text Detection in Images and Video Frames. *Proceedings of International Conference on Pattern Recognition*, **3**, 1015-1018.

[28] Jung, K., Kim, K., Kurata, T., Kourogi, M. and Han, J. (2002) Text Scanner with Text Detection Technology on Image Sequence. *Proceedings of International Conference on Pattern Recognition*, **3**, 473-476.

[29] Jain, A.K. and Zhong, Y. (1996) Page Segmentation Using Texture Analysis. *Pattern Recognition*, **29**, 743-770. http://dx.doi.org/10.1016/0031-3203(95)00131-X

[30] Yeo, B.-L. and Liu, B. (1996) Visual Content Highlighting via Automatic Extraction of Embedded Captions on MPEG Compressed Video. *Proceedings of SPIE*, **2668**, 142-149.

[31] Gargi, U., Crandall, D., Antani, S., Gandhi, T., Keener, R. and Kasturi, R. (1999) A System for Automatic Text Detection in Video. *Proceedings of International Conference on Document Analysis and Recognition*, Bangalore, 20-22 September 1999, 29-32.

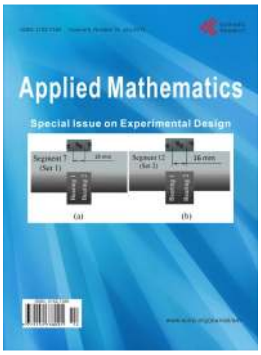
[32] Kim, H.K. (1996) Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database. *Journal of Visual Communication and Image Representation*, **7**, 336-344.

[33] Canedo-Rodriguez, A., Kim, J.H., Kim, S.H., Kelly, J., Kim, J.H., Yi, S., Veeramachaneni, S.K. and Blanco-Fernandez, Y. (2012) Efficient Text Extraction Algorithm using Color Clustering for Language Translation in Mobile Phone. *Journal of Signal and Information Processing* (*JSIP*), **3**, 228-237. http://dx.doi.org/10.4236/jsip.2012.32031

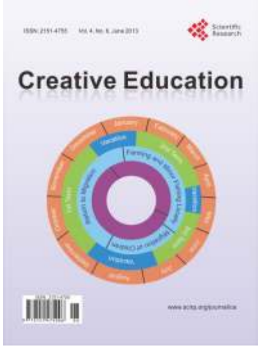
Advances in Bioscience and Biotechnology

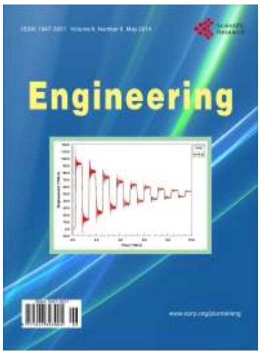
American Journal of Analytical Chemistry

American Journal of Plant Sciences

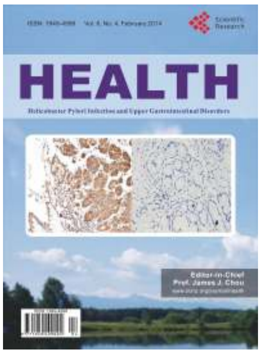

Applied Mathematics
Special Issue on Experimental Design

Agricultural Sciences
Special Issue on Research on Rice


Creative Education

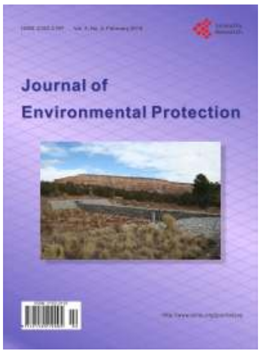

Engineering

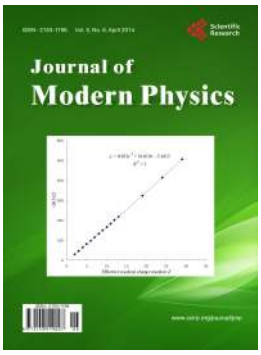
Food and Nutrition Sciences

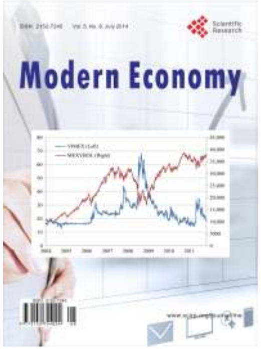

HEALTH

Journal of Computer and Communications

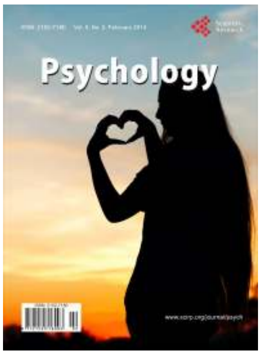
Journal of Cancer Therapy


Journal of Environmental Protection


Journal of Modern Physics


Modern Economy

Natural Science


Psychology